ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DEPARTMENT OF INDUSTRIAL ENGINEERING

Second Cycle Degree in
AEROSPACE ENGINEERING

Modeling and Control of a Gyrodyne Aircraft Using Neural-Network Based Control Allocation

Supervisor
Prof. Paolo Castaldi

Student
Abhilash Vijaya Kumar

Academic Year 2025/2026

To my mother, Anuradha
without whom this journey
would never be possible.

Abstract

As vertical lift vehicles gain prominence in industry, there is an increased need for studying different configurations. Although, an infinite number of configurations can be designed based on different requirements, only a few make it past the drawing board. In this thesis, a gyrodyne configuration is modelled, simulated and an appropriate control system is developed in a simulation based environment.

This work comprises of developing mathematical models based on fundamental aerodynamic principles. The model is further validated and control laws are then designed based on the implemented model depending on the requirements set early in the design process.

The thesis focuses on modelling, trim procedure, linearisation, development of control system. A novel approach to control allocation is taken to map the most efficient method to map desired moments to effector commands. A proof of concept simulation of the aircraft and comparisons based on multiple approaches are presented.

Acknowledgements

When I first arrived in Italy, I had left my comfort zone for the first time in 22 years. I remember being insecure, completely alone and terrified. I feel now that I am a completely different person and if not for the support and love I received, it would not be possible. I would like to express my gratitude to those who made me a better person and more so, an engineer.

I would like to sincerely thank my supervisor, Dr. Paolo Castaldi for his guidance, who I could dearly rely upon when I ran into dead-ends.

I would like to thank my mother, Anuradha for allowing me to pursue my dreams, prioritizing my daily bread over hers and letting me figure out life.

I am truly grateful for Shivani, for the incredible patience and understanding, for all the tantrums I threw throughout the years and many more to come.

A warm thank you to the people at Sky Eye Systems for providing me with support, and valuable experiences. A special mention to Mr. Tomasso Nannini who always answered my emails and questions.

Of course, my friends Lorenzo Ricci, Michail Moschopoulos who welcomed me and made me feel at home. I consider myself very lucky to have them around me.

Finally, heartfelt thanks to everyone who helped me one way or another.

Contents

List of Figures

List of Tables

Nomenclature and Conventions

Acronyms

VTOL	Vertical Take-Off and Landing
MIL	Model-in-Loop
CFD	Computational Fluid Dynamics

Latin Letters

a	Blade lift curve slope $[1/rad]$
a_0	Blade coning angle $[rad]$
BL, WL, STA	Buttline, Waterline and Stationline $[m]$
c	Chord length $[m]$
C_D	Drag co-efficient
C_L	Lift co-efficient
$C_{L_{max}}$	Maximum lift co-efficient
C_T	Thrust co-efficient
g	Gravitational acceleration $[m/s^2]$
n_b	Number of blades
L, M, N	Rolling, pitching and yawing moment $[Nm]$
n_b	Number of blades
T	Thrust $[N]$
Q	Torque $[Nm]$

R	Rotor radius $[m]$
Re	Reynolds Number
p, q, r	Roll, pitch, yaw rates of body c.g axes $[rad/s]$
p_H, q_H, r_H	Roll, pitch, yaw rates in hub frame $[rad/s]$
v_i	Induced velocity at rotor disk $[m/s]$
u_H, v_H, w_H	Longitudinal, lateral and vertical velocity in hub frame $[m/s]$

Greek Letters

α	Angle of attack $[rad]$
β	Sideslip angle $[rad]$
β_w	Rotor sideslip angle $[rad]$
δ	Equivalent rotor blade profile drag co-efficient
ρ	Air density $[kg/m^3]$
Ω	Angular rate $[rad/s]$
ϵ	Hinge offset ratio, e/R
λ	Inflow ratio
θ_0	Blade root collective pitch $[rad]$
θ_t	Total blade twist, root minus tip $[rad]$
θ_s	Longitudinal cyclic pitch $[rad]$
θ_c	Lateral cyclic pitch $[rad]$
μ	Rotor advance ratio

Subscripts

$c.g$	Centre of Gravity
$c.p$	Centre of Pressure
H	Rotor hub axis system
MR	Main Rotor
TR	Tail Rotor

PR	Propeller
W	Hub-wind axis system

Conventions

All angles follow right hand rule convention, the simulations are performed in radians [rad] while graphs are illustrated in degrees [°] for the readers to have a feel for these numbers.

Chapter 1

Introduction

There has been an increased interest in expanding the envelope of vertical lift vehicles, especially in range, endurance, velocity and angle of attack. Out of the vertical lift vehicles, gyrodyne configuration is the least studied configuration. Although this configuration is the oldest, gyrodyne's concept was tested as early as 1960's. Prominence of gyrodyne faded away with increase in fixed wing, and helicopters. In recent years, with resurgence in applications for vertical take off and landing (VTOL's), it can be seen that the gyrodyne concept can offer the best of both worlds i.e., fixed wing and rotary aircraft's.

Much of the recent focus has been on tilt-rotors and multi-rotors as they offer significant performance but still lack advancements in safety. These configurations can operate as helicopter and transition to cruise as fixed wing aircraft. It is expected that the use of VTOL's is only going to increase and it will be a fundamental mode of transportation in the future. The flight envelope for a traditional helicopters are heavily limited by the aerodynamic effects on the main rotor. In high velocity flights, the rotor blade can experience high drag on the advancing side, whereas on the retreating side, the rotor experiences flow separation and reverse inflow which reduces thrust and severely limits forward velocity. Fixed wings aircraft's, since the advancement of jet engines have been capable of flying at transonic speeds, the lack of ability to vertically take off and land has left a gap in mobility.

Predicting the performance and understanding the behaviour of aircraft's especially for different configurations is of paramount importance, especially in aerospace industry. A designer must be able to actively change a parameters and see the effect of it, in the past, this was limited to final few design options. Now with the computational advancements, mathematical models can be made and numerical simulations can be used to study the behaviour in early design stages. In addition, using Model-in-Loop (MIL) techniques, the aircraft's operational envelope can be validated and tested which can help reduce cost significantly. Apart from the above mentioned advantages, the scope of MIL goes beyond design and development i.e., different control methodologies can be implemented.

The gyrodyne is an over actuated aircraft with multiple control effectors capable of generating similar forces and moments. While this redundancy offers greater control flexibility and improved performance, it also introduces the challenge of distributing control commands among the available actuators. This problem is addressed through control allocation techniques. In this thesis, a neural network based control allocation framework is developed to efficiently map the desired control inputs to the actuators and is evaluated within a Model-in-Loop simulation environment.

1.1 Gyrodyne

A gyrodyne is a type of VTOL aircraft, that is a combination of fixed wing with rotor-like system. The rotor is driven by an engine during vertical take-off and landings. During forward flight, the the rotor is unpowered and free spinning, like an autogyro. The lift for the aircraft is provided by the wings, the thrust by a propeller which is driven by the engine. A typical gyrodyne can be seen below:

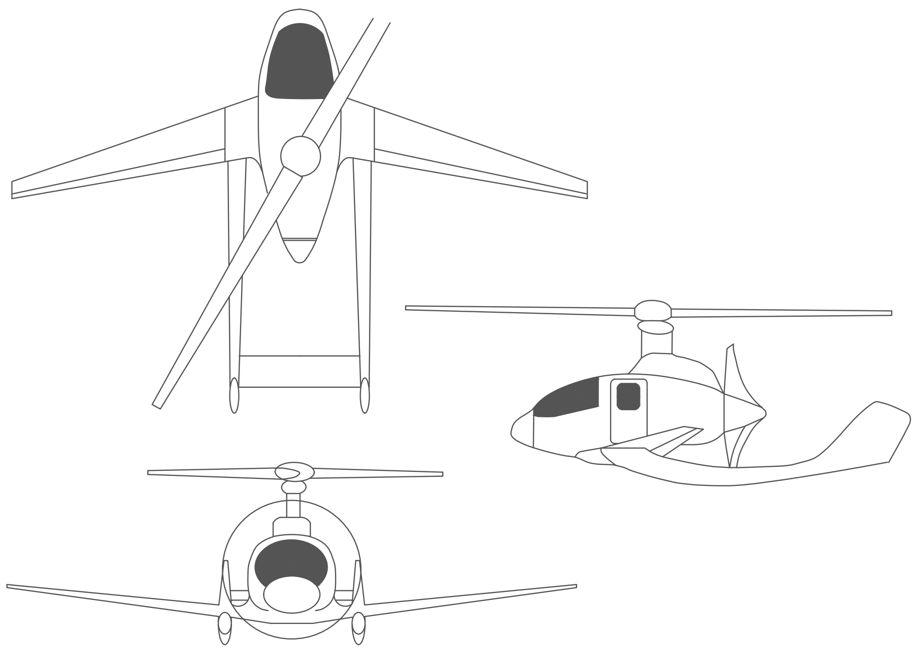

Figure 1.1: Generic Gyrodyne Configuration

There are different types of gyrodyne configuration, the configuration is usually decided based on the size and application for intended use. For example, the earliest configuration involved addition of auxiliary wings to helicopter models (McDonnell XV-1, Fairey Rotodyne). On the Sikorsky S-61F, an additional tail rotor driven by the engine is fitted which is used to provide the necessary anti-torque. The power from the engine is divided between the propeller and the tail rotor.

The most recent configuration of gyrodyne is the CarterCopter. The CarterCopter pioneered the Slowed Rotor Compound (SRC) where the rotor stability is achieved through the combination of the rotor tip weights' location ahead of the blade center line. This led it to achieve flights with advance ratio, $\mu = 1$. In the preliminary design stage, all the configurations were evaluated and based on the requirements, the main rotor-tail rotor configuration was chosen by Sky Eye Systems, Pisa. Other design considerations include, rotor-wing and propeller-rotor interactions. The configuration that will be studied in this thesis is as illustrated below.

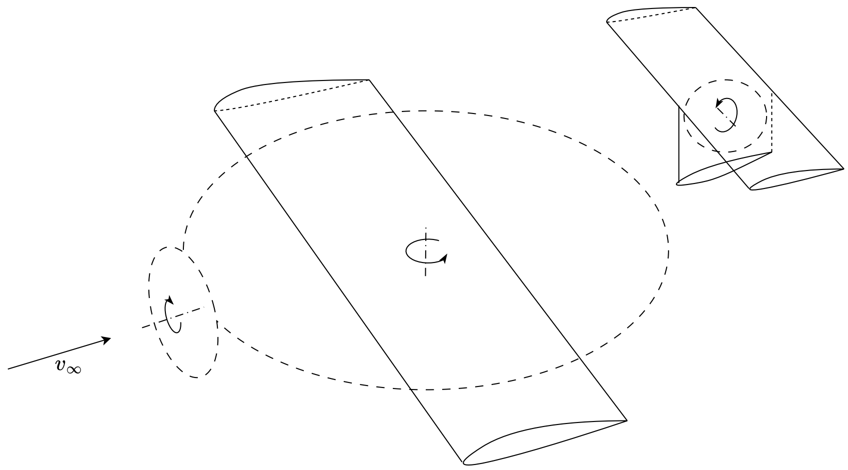

Figure 1.2: Sky Eye Systems gyrodyne configuration

As discussed earlier, since gyrodyne can operate as a helicopter during take-off and transition to forward flight, the flow phenomena is extremely complex. This requires high fidelity simulations such as Computational Fluid Dynamics (CFD) to understand the aerodynamic behaviour such as Rotor-Wing, Propeller-Rotor interactions. These interactions affect the dynamics and stability of the aircraft, hence it is valuable to design a mid-fidelity model which can remain computationally inexpensive but capture some of these dynamics.

1.2 Research Objectives

The following objectives were set for the thesis work:

- Develop fundamental understanding of aerodynamic principles behind mathematical models e.g., rotor dynamics, etc.
- Study the different interactions that can affect the accuracy of the model.
- Implementing the mathematical model on Simulink environment.
- Validate the model and analyse the inherent stability of the aircraft and propose a control system.
- Design and implement a control system for the aircraft.

1.3 Thesis Outline

The thesis is divided into six chapters, the work aims to provide a theoretical and practical explanation. The chapters are structured as follows:

Chapter 2: Discusses the modelling principles, derivations, assumptions, validations and implementation of subsystems.

Chapter 3: Introduces a numerical trim procedure, autorotation envelope and linearisation operation using MATLAB.

Chapter 4: Presents the design of a classical control system based on linear control theory.

Chapter 5: Provides an overview of control allocation methods and presents the implementation of the proposed neural network-based control allocation framework.

Chapter 6: Summarises the main findings of the thesis and discusses the limitations of the proposed approach. Recommendations for future work and potential improvements are also presented.

Chapter 2

Modelling

This chapter documents the mathematical model used in the simulation. The mathematical model is non-linear, and has nine degrees of freedom: The six rigid body, two rotor-flapping, and the rotor rotational degree of freedom. The rotor, propeller and tail rotor is modelled using Blade Element Momentum Theory (BEMT). The fuselage uses a curve fit representation using data from CFD over different angle of attacks and sideslip angles. The wing is modelled using lift curve slope between the stall limits. In the following sections, the mathematical models along with the assumptions and limitations are studied in detail. Any notations used is listed above in Nomenclature and Conventions.

To solve the ordinary differential equations (ODEs), Runge-Kutta 4th order scheme with fixed time step of 0.01s was used.

2.1 Main Rotor

The main rotor is modelled using fundamental rotor dynamics. The combination from the earliest work of Rankine and Froude resulted in BEMT. In this thesis, the forces and moments generated by the rotor is derived by omitting the harmonic terms [1]. Some of the assumptions when modelling the rotor are:

1. The rotor blade is rigid and blade twist is linear.
2. The flapping angle and inflow angles are assumed to be small and simple strip theory was used in the process.
3. Tip loss factor was assumed to be 1.
4. The compressibility and stall effects are ignored.
5. The lift curve slope of the airfoil is assumed to be linear.
6. The rotor blade does not undergo any lead-lag, coning or flapping.[1]

[1] Lateral stability in hover is provided by titling the entire rotor disk, replacing explicit flapping hinges.

Because of these assumptions and simplifications, the results of the analysis are valid for limited flight conditions. The authors of [2] has concluded that this type of analysis is valid for stability and control analysis up to advance ratio of 0.3.

The forces and moment expression written in hub-wind frame are given below:

$$\begin{aligned}
T_{MR} &= \frac{n_b}{2}\rho acR(\Omega R)^2\Big\{\frac{\lambda}{2}(1-\epsilon^2)+\theta_0\left[\frac{1}{3}+\frac{\mu^2}{2}(1-\epsilon)\right]+\theta_t\left[\frac{1}{4}+\frac{\mu^2}{4}(1-\epsilon^2)\right] \\
&\quad+\frac{\mu}{4}(1-\epsilon^2)\left(\frac{p_H}{\Omega}\cos\beta_w+\frac{q_H}{\Omega}\sin\beta_w\right)\Big\} \qquad (2.1) \\
H_W &= \frac{n_b}{2}\rho acR(\Omega R)^2\Big\{\frac{\delta\mu}{2a}(1-\epsilon^2)-\frac{\theta_0}{4}\left[2(1-\epsilon)\lambda\mu+\frac{2}{3}\left(\frac{p_H}{\Omega}\cos\beta_w+\frac{q_H}{\Omega}\sin\beta_w\right)\right] \\
&\quad-\frac{\theta_t}{4}\left[\mu(1-\epsilon^2)\lambda+\frac{1}{2}\left(\frac{p_H}{\Omega}\cos\beta_w+\frac{q_H}{\Omega}\sin\beta_w\right)\right] \\
&\quad-\frac{1}{4}\left[\frac{2a_0}{3}\left(-\frac{p_H}{\Omega}\sin\beta_w+\frac{q_H}{\Omega}\cos\beta_w\right)\right] \\
&\quad-2(1-\epsilon^2)\lambda\left(\frac{p_H}{\Omega}\cos\beta_w+\frac{q_H}{\Omega}\sin\beta_w\right)+\frac{\mu}{4}(1-\epsilon^2)a_0^2\Big\} \qquad (2.2) \\
Y_W &= \frac{n_b}{2}\rho acR(\Omega R)^2\Big\{-\frac{\theta_0}{4}\left[3a_0(1-\epsilon^2)\mu-\frac{2}{3}\left(-\frac{p_H}{\Omega}\sin\beta_w+\frac{q_H}{\Omega}\cos\beta_w\right)\right] \\
&\quad-\frac{\theta_t}{4}\left[2a_0\mu-\frac{1}{2}\left(-\frac{p_H}{\Omega}\sin\beta_w+\frac{q_H}{\Omega}\cos\beta_w\right)\right] \\
&\quad-\frac{1}{4}\left[\frac{2a_0}{3}\left(\frac{p_H}{\Omega}\cos\beta_w+\frac{q_H}{\Omega}\sin\beta_w\right)-2(1-\epsilon^2)\lambda\left(-\frac{p_H}{\Omega}\sin\beta_w+\frac{q_H}{\Omega}\cos\beta_w\right)\right] \\
&\quad-\frac{3}{2}a_0\lambda\mu(1-\epsilon)\Big\} \qquad (2.3) \\
Q &= \frac{n_b}{2}\rho acR^2(\Omega R)^2\Big\{\frac{\delta}{4a}(1+(1-\epsilon^2)\mu^2) \\
&\quad-\theta_0\left[\frac{\lambda}{3}+\frac{\mu}{6}\left(\frac{p_H}{\Omega}\cos\beta_w+\frac{q_H}{\Omega}\sin\beta_w\right)\right]-\frac{\theta_t\lambda}{4}-\frac{1}{2}(1-\epsilon^2)\left(\lambda^2+\frac{\mu^2a_0^2}{2}\right) \\
&\quad+\frac{\mu}{3}a_0\left(-\frac{p_H}{\Omega}\sin\beta_w+\frac{q_H}{\Omega}\cos\beta_w\right)-\frac{1}{8}\left[\left(\frac{p_H}{\Omega}\right)^2+\left(\frac{q_H}{\Omega}\right)^2\right]\Big\} \qquad (2.4) \\
L &= 0 \qquad (2.5) \\
M &= 0 \qquad (2.6)
\end{aligned}$$

It can be observed from equation 2.1 that rotor inflow ratio, λ is required to compute the thrust.
The rotor inflow is defined as:

$$\lambda = \frac{w_H}{\Omega R}-\frac{C_T}{2\left(\mu^2+\lambda^2\right)^{1/2}} \qquad (2.7)$$

Since this is an implicit relation, this can be solved using an appropriate root-finding iterative method such as Newton-Ralphson, Halley's method, etc. In this work,

MATLAB's pre-existing *fzero* routine which uses Brent's method.
The advance ratio and the sideslip angles for the main rotor is given by:

$$\mu = \frac{\sqrt{u_H^2 + v_H^2}}{\Omega R}, \quad \beta = \tan^{-1}\left(\frac{v_H}{u_H}\right) \tag{2.8}$$

The rotor profile drag co-efficient, δ which is used to compute H_W and Q is computed using:

$$\delta = 0.142Re^{-0.2} + \frac{0.3}{a^2}\left[\frac{6C_T}{\sigma\left(1 + \frac{3}{2}\mu^2\right)}\right]^2 \tag{2.9}$$

The equations 2.5 and 2.6 are reduced to 0 because there is no flapping hinge, instead the rotor disk tilts to provide the necessary lateral forces.

The hub-wind frame forces and moments are transformed into hub frame as follows:

$$\begin{aligned} H_H &= H_W \cos(\beta_W) + Y_W \sin(\beta_W) \\ Y_H &= -H_W \sin(\beta_W) + Y_W \cos(\beta_W) \\ M_H &= M_W \cos(\beta_W) + L_W \sin(\beta_W) \\ L_H &= -M_W \sin(\beta_W) + L_W \cos(\beta_W) \end{aligned} \tag{2.10}$$

The transformation of forces and moments to body frame is done ignoring the effect of shaft-tilt and the moment about C.G caused by rotor forces.

$$\begin{aligned} X_{MR} &= -H_H \\ Y_{MR} &= Y_H \\ Z_{MR} &= -T \\ L_{MR} &= L_H + Y_{MR}\left(WL_H - WL_{c.g}\right) - Z_{MR}\left(BL_{c.g}\right) \\ M_{MR} &= M_H - Z_{MR}\left(STA_{c.g} - STA_H\right) - X_{MR}\left(WL_H - WL_{c.g}\right) \\ N_{MR} &= Q + X_{MR}\left(BL_{c.g}\right) + Y_{MR}\left(STA_{c.g} - STA_H\right) \end{aligned} \tag{2.11}$$

which can be written as,

$$\begin{aligned} F_{rotor} &= [X_{MR} \quad Y_{MR} \quad Z_{MR}]' \\ M_{rotor} &= [L_{MR} \quad M_{MR} \quad N_{MR}]' \end{aligned}$$

2.1.1 Dynamic Inflow

The authors of [1] used uniform inflow which means no inflow dynamics were used, entailing that the induced velocity builds up instantaneously. Neglecting the wake dynamics leads to unrealistic thrust responses during transient manoeuvres. For flight dynamics and control applications, a simple harmonic, finite-state, non-uniform inflow model for induced velocity similar to that originally proposed by Glauert is still being used extensively.The major advantage of the proposed method is the low computational effort required during operation.

In this thesis, the goal is to make sure one has reasonable response to thrust build-up. A study comparing the inflow models [3] concluded that Pitt-Peters provided better overall performance over others. Hence, for this reason, Pitt-Peters dynamic inflow model is used. Pitt-Peters is a semi-empirical method of computing inflow, where it uses a combination of actuator disk, momentum theory and an assumed wake shape.

The Pitt-Peters model is given by:

$$\lambda(\bar{r}, \psi) = \lambda_0 + \lambda_s \bar{r} \sin\psi + \lambda_c \bar{r} \cos\psi \tag{2.12}$$

where, $\lambda_0, \lambda_s, \lambda_c$ are the the inflow states and $\bar{r}$ is the azimuthal positions. Now since the BEMT used in [1] is averaged over the entire rotation of blade, the equation 2.12 reduces to $\lambda(\bar{r}, \psi) = \lambda_0$ so in the process, the non-uniform inflow distribution is lost. But the implementation can be continued in order obtain the information regarding inflow build up.

$$M \begin{Bmatrix} v_0 \\ v_s \\ v_c \end{Bmatrix}' + L^{-1} \begin{Bmatrix} v_0 \\ v_s \\ v_c \end{Bmatrix} = \begin{Bmatrix} C_T \\ C_l \\ C_m \end{Bmatrix}$$

where, L and M is the Pitt-Peters co-efficient matrix and apparent mass matrix.

$$L = \begin{bmatrix} \frac{1}{2} & 0 & -\frac{15}{64}\pi\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} \\ 0 & \frac{4}{1+\sin\alpha} & 0 \\ \frac{15}{64}\pi\sqrt{\frac{1-\sin\alpha}{1+\sin\alpha}} & 0 & \frac{4}{1+\sin\alpha} \end{bmatrix} \quad , \quad M = \frac{1}{\pi}\begin{bmatrix} \frac{8}{3} & 0 & 0 \\ 0 & -\frac{16}{45} & 0 \\ 0 & 0 & -\frac{16}{45} \end{bmatrix}$$

The ODE can be solved using any ODE solvers such as forward Euler or Runge-Kutta but the stability of the ODE must be considered when choosing time step. For stability reasons, the inflow ratio was bounded ± 0.1 within the λ obtained from BEMT.

2.1.2 Cyclic Control

The cyclic control for the gyrodyne is obtained by tilting the entire rotor disk, similar to most autogyro's. The configuration is as illustrated below:

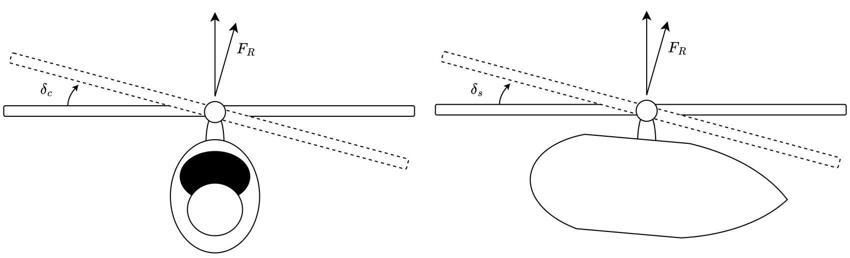

Figure 2.1: Cyclic control mechanism

The rotor forces are tilted to orient the forces in order to provide the necessary lateral forces. The modelling principle also remains the same where the thrust is decomposed based on the cyclic input.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_s & \sin\theta_s \\ 0 & -\sin\theta_s & \cos\theta_s \end{bmatrix} \quad , \quad R_y = \begin{bmatrix} \cos\theta_c & 0 & -\sin\theta_c \\ 0 & 1 & 0 \\ \sin\theta_s & 0 & \cos\theta_c \end{bmatrix}$$

$$\begin{aligned} R &= R_y \times R_x \\ F &= R \times F_{rotor} \end{aligned} \tag{2.13}$$

2.1.3 Main Rotor Simulation

The following sections discusses some of the results obtained from the simulation of isolated main rotor. The simulations are run for hover and forward flight conditions (powered). This analysis provides an opportunity to validate the behaviour. Here, the rotor forces and moments are validated against CFD results.

Hover Condition

During the analysis, the angular rate is set constant. The density of air, $\rho = 1.225$, the rotor is assumed to perfectly above the C.G location and no external influence of wind is present.

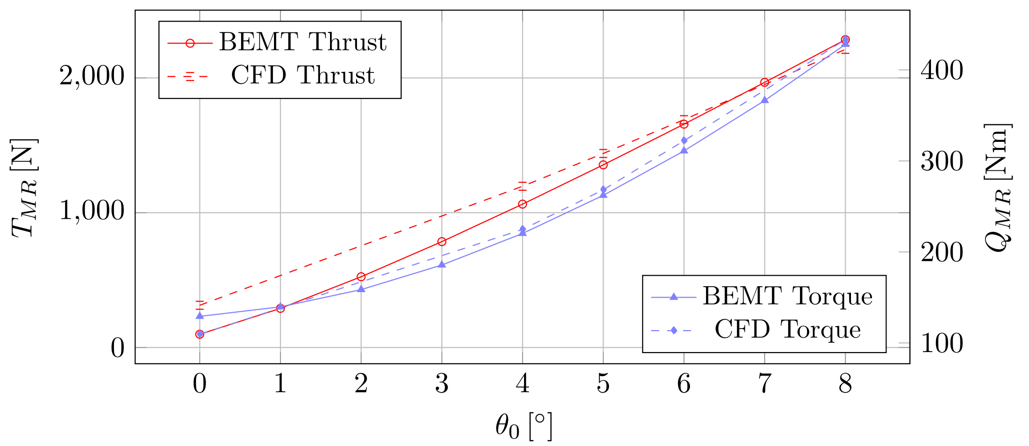

Figure 2.2: θ_0 vs Thrust and Torque

The figure 2.2 shows the plot of thrust and moment as a function of collective pitch in degrees. The BEMT is in acceptable agreement with the CFD results. The relative thrust error on average is 7% and decreases as collective increases. It must be noted that the CFD thrust is slightly higher than BEMT which is in contrast to classical rotor theory. Regardless of the reasons, the modelling approach can be validated for designing a controller. The moments have a relative error of 3%, the error is relatively low since some of the co-efficients of equation 2.9 and the induced losses correction parameter K_{ind} was tuned.

Forward Flight Condition

Similar to the hover condition, the forward flight condition is tested. Since no CFD data was available to validate, only qualitative behaviour can be analysed. The plots of Thrust against velocity are plotted and compared against theory from classical literature such as [4].

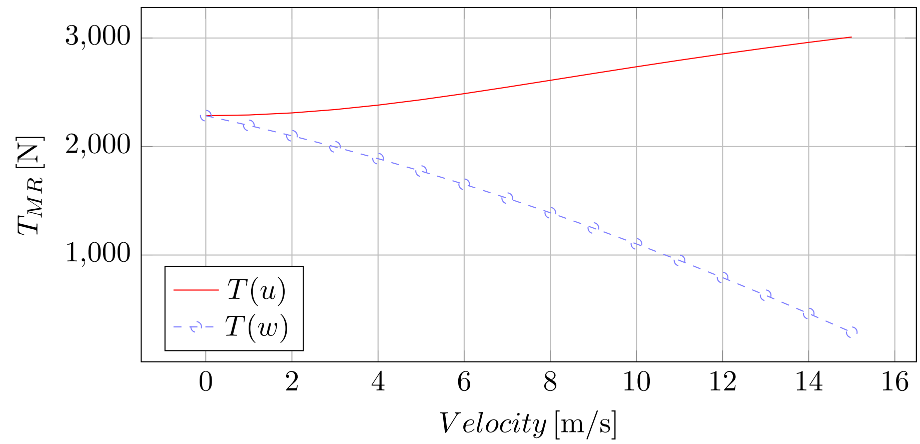

Figure 2.3: Velocity vs Thrust

The figure shows the variation of thrust with velocity, and we can observe that thrust increases slightly with increase in u-velocity and decreases with increase in w-velocity because as the rotor climb, the angle of attack reduces.

2.2 Tail Rotor

The tail rotor in this case is used to provide the necessary anti-torque during hover. In this gyrodyne configuration, fenestron rotor is employed. The tail rotor has no cyclic control and only the collective pitch is controlled. It is again assumed to be tethered, with a constant coning angle a_0. The local flow at the tail rotor must be computed which includes the downwash from the main rotor.

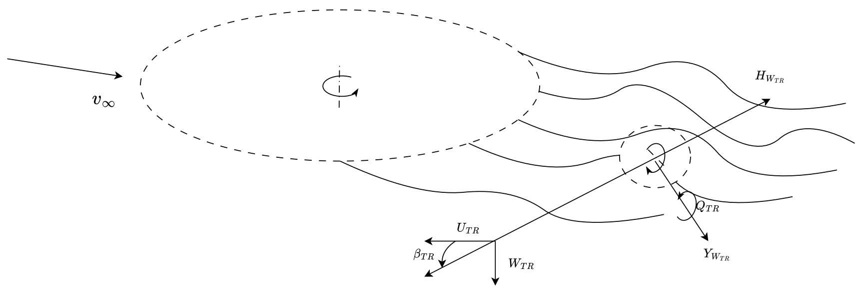

Figure 2.4: Tail Rotor in the wake on main rotor

The authors of ref [1] have neglected ϵ and n_b, due to the fact that number of tail rotor used was two, and ϵ was negligible. In fenestron rotors, the number of blades are much higher with large hinge offsets. Hence, necessary changes to equations have been adopted. The rotor forces and moments are calculated as follows:

$$\begin{aligned}T_{TR} &= \frac{n_b}{2}\rho acR(\Omega R)^2\Bigg\{\frac{\lambda}{2}(1-\epsilon^2)+\theta_0\left[\frac{1}{3}+\frac{\mu^2}{2}(1-\epsilon)\right]+\theta_t\left[\frac{1}{4}+\frac{\mu^2}{4}(1-\epsilon^2)\right]\\ &\quad+\frac{\mu}{4}(1-\epsilon^2)\left(\frac{p_{TR}}{\Omega}\right)\Bigg\} \qquad (2.14)\\ H_{W_{TR}} &= \frac{n_b}{2}\rho acR(\Omega R)^2\Bigg\{\frac{\delta\mu}{2a}-\frac{1}{2}\theta_0\lambda\mu-\frac{1}{4}\theta_t\mu\lambda+\frac{1}{4}\mu a_0^2\\ &\quad-\left(\frac{\theta_0}{6}+\frac{\theta_t}{8}+\frac{\lambda}{2}\right)\frac{p_{TR}}{\Omega}-\frac{a_0}{6}\frac{q_{TR}}{\Omega}\Bigg\} \qquad (2.15)\end{aligned}$$

$$\begin{aligned}Y_{W_{TR}} &= \frac{n_b}{2}\rho acR(\Omega R)^2\Bigg\{-\frac{3}{4}\theta_0 a_0\mu-\frac{1}{2}\theta_t a_0\mu-\frac{6}{4}\mu a_0\lambda-\frac{a_0}{6}\frac{p_{TR}}{\Omega}\\ &\quad+\left(\frac{\theta_0}{6}+\frac{\theta_t}{8}+\frac{\lambda}{2}\right)\frac{q_{TR}}{\Omega}\Bigg\} \qquad (2.16)\\ Q_{TR} &= \frac{n_b}{2}\rho acR^2(\Omega R)^2\Bigg\{\frac{\delta}{4a}(1+\mu^2)-\theta_0\left[\frac{\lambda}{3}+\frac{\mu}{6}\frac{p_{TR}}{\Omega}\right]-\frac{\theta_t\lambda}{4}-\frac{1}{2}\left(\lambda^2+\frac{\mu^2a_0^2}{2}\right)\\ &\quad+\frac{\mu}{3}a_0\frac{q_{TR}}{\Omega}-\frac{1}{8}\left[\left(\frac{p_{TR}}{\Omega}\right)^2+\left(\frac{q_{TR}}{\Omega}\right)^2\right]\Bigg\} \qquad (2.17)\end{aligned}$$

where, inflow ratio λ is given below and this implicit relation is again solved similar to equation 2.7.

$$\lambda = \frac{-v_{TR}}{\Omega_{TR}R_{TR}}-\frac{C_{T_{TR}}}{2\left(\mu_{TR}^2+\lambda_{TR}^2\right)^{1/2}} \qquad (2.18)$$

The local velocities as the tail rotor hub is obtained by:

$$\begin{aligned}u_{TR} &= u_B\\ v_{TR} &= v_B+p_B\left(WL_{TR}-WL_{c.g}\right)-r_B\left(STA_{TR}-STA_{c.g}\right)\\ w_{TR} &= w_B+q_B\left(STA_{TR}-STA_{c.g}\right)\end{aligned} \qquad (2.19)$$

The effects of rotor wake, $w_{i_{TR}}$ are ignored because the tail rotor is operational only during hover and transition. Hence, it is assumed that the tail rotor is not affected by the main rotor's wake.
The advance ratio and side-slip angles are given by:

$$\mu = \frac{\sqrt{u_{TR}^2+w_{TR}^2}}{\Omega_{TR}R_{TR}}, \quad \beta_{TR} = \tan^{-1}\left(\frac{w_{TR}}{u_{TR}}\right) \qquad (2.20)$$

The tail rotor forces and moments are transformed to body axis similarly:

$$\begin{aligned}X_{TR} &= -\left(Y_{TR}\right)_W\sin\beta_{TR}-\left(H_{TR}\right)_W\cos\beta_{TR}\\ Y_{TR} &= T_{TR}\\ Z_{TR} &= \left(Y_{TR}\right)_W\cos\beta_{TR}-\left(H_{TR}\right)_W\sin\beta_{TR}\\ L_{TR} &= Y_{TR}\left(WL_{TR}-WL_{c.g}\right)\\ M_{TR} &= -Q_{TR}+Z_{TR}\left(STA_{TR}-STA_{c.g}\right)-X_{TR}\left(WL_{TR}-WL_{c.g}\right)\\ N_{TR} &= -Y_{TR}\left(STA_{TR}-STA_{c.g}\right)\end{aligned} \qquad (2.21)$$

which can be written as,

$$F_{tail\ rotor} = [X_{TR} \quad Y_{TR} \quad Z_{TR}]'$$
$$M_{tail\ rotor} = [L_{TR} \quad M_{TR} \quad N_{TR}]'$$

2.2.1 Tail Rotor Simulation

This section discusses the results from the tail rotor. An isolated tail rotor is simulated. Since no CFD was available, the rotor forces were validated against experimental results of [5]. The paper used a fenestron rotor and the parameters are listed in the table below:

Parameter	Value
Tail rotor blade count	10
Chord length [m]	0.058
Inner wall radius [m]	0.3667
Root pitch [°] at 0.46 (r/R)	11
Tip pitch [°] at 1.0 (r/R)	0

Table 2.1: Tail Rotor Parameters

The rotor used NACA63A312 airfoil, linearly twisted blade. Since the fenestron rotor is ducted, the rotor used duct diffusion half-angle of 5°. The rotor was isolated and no influence of the main rotors wake is taken into account. The plots of thrust as function of collective is given below:

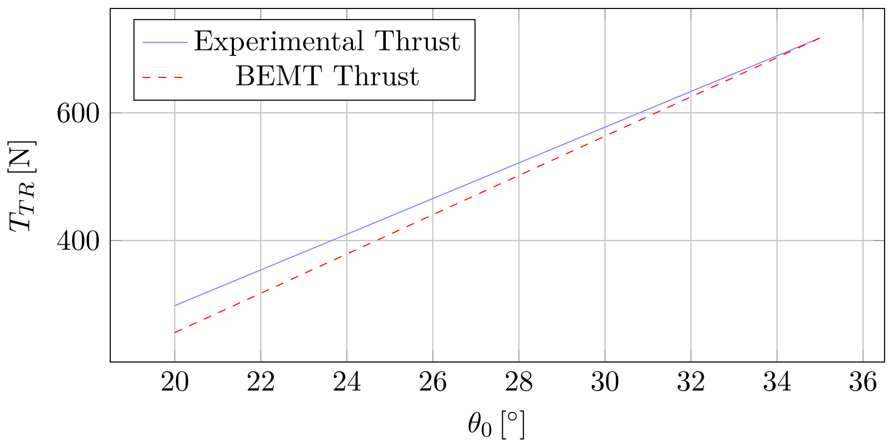

Figure 2.5: Tail thrust vs θ_0

The BEMT is again in acceptable agreement with experimental thrust. The BEMT relative to experiment has a relatively small error on average of 12% and decreases with increase in collective. It must be noted that this is validated against the fenestron used by the authors of [5] and hence it is necessary to validate with the rotor used in this gyrodyne configuration.

2.3 Propeller

The propeller is modelled similar to main rotor and tail rotor. In this case, the propeller used is a variable pitch, three bladed, constant RPM driven by an engine through a 3 : 1 reduction and is illustrated below. The propeller is assumed to be tethered, and all of the assumptions used during rotor modelling in section 2.1 is still valid.

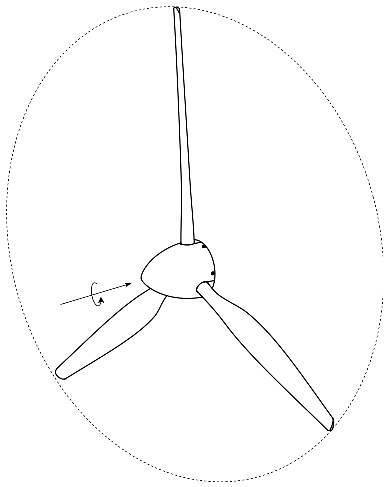

Figure 2.6: Propeller Configuration

The propeller forces and moments are computed as follows:

$$T_{PR} = \frac{n_b}{2}\rho acR(\Omega R)^2\Bigg\{\frac{\lambda}{2}(1-\epsilon^2) + \theta_0\left[\frac{1}{3}+\frac{\mu^2}{2}(1-\epsilon)\right] + \theta_t\left[\frac{1}{4}+\frac{\mu^2}{4}(1-\epsilon^2)\right] + \frac{\mu}{4}(1-\epsilon^2)\frac{p_{PR}}{\Omega}\Bigg\} \tag{2.22}$$

$$H_{wPR} = \frac{n_b}{2}\rho acR(\Omega R)^2\Bigg\{\frac{\delta\mu}{2a} - \frac{\theta_0}{2}\lambda\mu - \frac{\theta_t}{4}\mu\lambda + \frac{1}{4}\mu a_0^2 - \left(\frac{\theta_0}{6}+\frac{\theta_t}{8}+\frac{\lambda}{2}\right)\frac{p_{PR}}{\Omega} - \frac{a_0}{6}\frac{q_{PR}}{\Omega}\Bigg\} \tag{2.23}$$

$$Y_{wPR} = \frac{n_b}{2}\rho acR(\Omega R)^2\Bigg\{-\frac{3}{4}\theta_0 a_0\mu - \frac{1}{2}\theta_t a_0\mu - \frac{3}{2}\mu a_0\lambda - \frac{a_0}{6}\frac{p_{PR}}{\Omega} + \left(\frac{\theta_0}{6}+\frac{\theta_t}{8}+\frac{\lambda}{2}\right)\frac{q_{PR}}{\Omega}\Bigg\} \tag{2.24}$$

$$Q_{PR} = \frac{n_b}{2}\rho acR^2(\Omega R)^2\Bigg\{\frac{\delta(1+\mu^2)}{4a} - \frac{\lambda\theta_0}{3} - \frac{\lambda\theta_t}{4} - \frac{1}{2}\left(\lambda^2+\frac{\mu^2 a_0^2}{2}\right) - \frac{\mu\theta_0}{6}\frac{p_{PR}}{\Omega} + \frac{\mu a_0}{3}\frac{q_{PR}}{\Omega} - \frac{1}{8}\left(\frac{p_{PR}^2+q_{PR}^2}{\Omega^2}\right)\Bigg\} \tag{2.25}$$

the inflow ratio, λ is given below.

$$\lambda = \frac{-u_{PR}}{\Omega_{PR} R_{PR}} - \frac{C_{T_{PR}}}{2\left(\mu_{PR}^2 + \lambda_{PR}^2\right)^{1/2}} \tag{2.26}$$

The advance ratio and side-slip angles are given by:

$$\mu = \frac{\sqrt{v_H^2 + w_H^2}}{\Omega R}, \quad \beta = \tan^{-1}\left(\frac{w_H}{v_H}\right) \tag{2.27}$$

The propeller forces and moments are transformed to body axis using:

$$\begin{aligned} X_{PR} &= T_{PR} \\ Y_{PR} &= (Y_{PR})_W \cos\beta_{PR} - (H_{PR})_W \sin\beta_{PR} \\ Z_{PR} &= (Y_{PR})_W \sin\beta_{PR} + (H_{PR})_W \cos\beta_{PR} \\ L_{PR} &= Q_{PR} + (BL_{PR} - BL_{c.g.})Z_{PR} - (WL_{PR} - WL_{c.g.})Y_{PR} \\ M_{PR} &= (WL_{PR} - WL_{c.g.})X_{PR} - (STA_{PR} - STA_{c.g.})Z_{PR} \\ N_{PR} &= (STA_{PR} - STA_{c.g.})Y_{PR} - (BL_{PR} - BL_{c.g.})X_{PR} \end{aligned} \tag{2.28}$$

which can be written as,

$$\begin{aligned} F_{propeller} &= [X_{PR} \quad Y_{PR} \quad Z_{PR}]' \\ M_{propeller} &= [L_{PR} \quad M_{PR} \quad N_{PR}]' \end{aligned}$$

2.3.1 Propeller Simulation

This section discusses the results obtained from simulation of an isolated propeller. The static thrust and moments are plotted below.

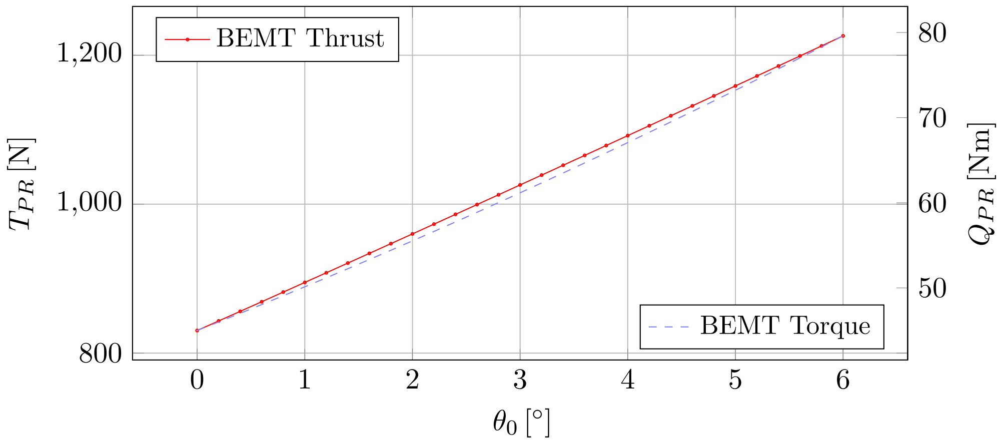

Figure 2.7: θ_0 vs Thrust and Torque

The above figure shows the plot of thrust, moment against collective pitch. Similar to tail rotor simulation, since no CFD data was available, the BEMT model was validated against paper.

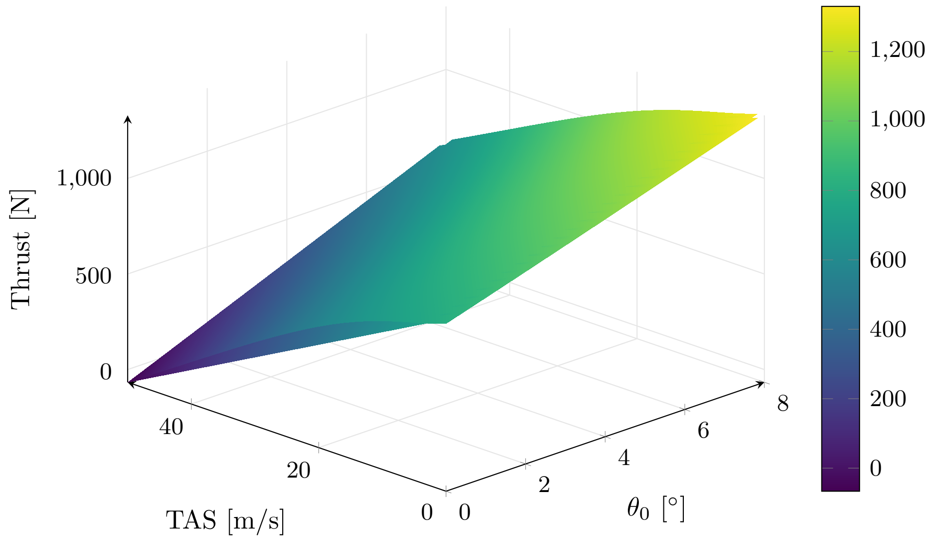

Figure 2.8: Thrust surface plot

A surface plot of Thrust vs Collective vs TAS is plotted above. Here, it can be observed that thrust increases as collective increases. Also, thrust decreases as airspeed increases since angle of attack decreases. The initial guess value for the trimming the aircraft is obtained from the surface plotted above.

2.4 Wing

The wing is modelled using classical aerodynamics of lifting surfaces. The wing uses a NACA 4-series airfoil, linearly tapered, no twist or sweep, with plain flaps as high lift device, ailerons for roll control. Some of the assumptions when modelling the wing are:

1. The lift and drag forces are applied at the quarter chord at span-wise location at centre of area.
2. The lift curve slope prior to stall is linear and no influence of rotor wake is taken into account.
3. The maximum lift co-efficient $C_{L_{max}}$ is specified from CFD and post stall C_L is small and linear.
4. The induced drag co-efficient varies as the square of the calculated lift co-efficient.
5. The effect of flap deployment is again calculated using CFD and look up table (LUT) based calculations is employed to obtain $C_{L_{\delta f}}$

The wing does not have any dihedral angle and is low-wing configuration. The configuration for the wing is illustrated below in figure 2.9.

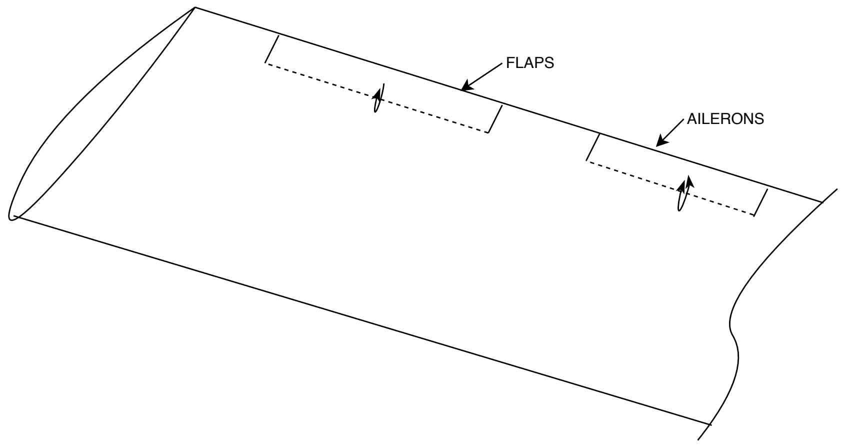

Figure 2.9: Wing Configuration

Before modelling, it is important to understand some of the interactions between Rotor-Wing, Wing-Rotor, Propeller-Wing and finally Rotor-tail rotor. These interactions can affect the performance of the subsystem which can have a large change in trim values.
The author of [6] studied the effect of these interactions on a compound helicopter. In this scenario, the wings are influenced by the main rotors wake which affects the angle of attack, however it is also noticed that there is an asymmetrical influence due to the different inflow velocity at the advancing and retreating side. On average the effective angle of attack can be obtained using simple trigonometry.

$$\alpha_{eff} = \alpha - tan^{-1}\left(\frac{v_i}{V}\right) \tag{2.29}$$

Similarly, the main rotors performance due to the wing is also affected. The author concluded that the main rotors efficiency decreased by around 4%. The propeller has a major influence,[7] and some of the effects are:

1. It increases the dynamic pressure.
2. It alters the wing angle of attack
3. Interference with the flow over the wing causes changes in the lift slope and the induced drag factor.

In this gyrodyne configuration, the main rotors influence is highest during transition and negligible in forward flight. However, since the rotor is not powered, propeller has the most influence during cruise. These effects although can be compensated, will affect the final trim conditions.

The aerodynamic forces expressed given below:

$$\begin{aligned} L &= \frac{1}{2}\rho V_\infty^2 S C_L \\ D &= \frac{1}{2}\rho V_\infty^2 S C_D \\ Y &= \frac{1}{2}\rho V_\infty^2 S C_Y \end{aligned} \tag{2.30}$$

where V_∞ is free stream velocity, S is equivalent surface area and C_L, C_D, C_Y are the lift, drag and sideforce co-efficients given by:

$$\begin{aligned} C_L &= C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\dot\alpha}}\dot\alpha + C_{L_q}q + C_{L_{\delta e}}\delta_e + C_{L_{\delta f}}\delta_f \\ C_D &= C_{D_0} + kC_L^2 \\ C_Y &= C_{Y_0} + C_{Y_\beta}\beta + C_{Y_{\dot\beta}}\dot\beta + C_{Y_p}p + C_{Y_r}r + C_{Y_{\delta a}}\delta_a + C_{Y_{\delta r}}\delta_r \end{aligned} \tag{2.31}$$

The three moments in wind frame is given below:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}\rho V_\infty^2 S C_l b \\ M &= \frac{1}{2}\rho V_\infty^2 S C_m \bar{c} \\ N &= \frac{1}{2}\rho V_\infty^2 S C_n b \end{aligned} \tag{2.32}$$

The co-efficients used for moments is obtained similarly from CFD and is given by:

$$\begin{aligned} C_l &= C_{l_0} + C_{l_\beta}\beta + C_{l_{\dot\beta}}\dot\beta + C_{l_p}p + C_{l_r}r + C_{l_{\delta a}}\delta_a + C_{l_{\delta r}}\delta_r \\ C_m &= C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\dot\alpha}}\dot\alpha + C_{m_q}q + C_{m_{\delta e}}\delta_e + C_{m_{\delta f}}\delta_f \\ C_n &= C_{n_0} + C_{n_\beta}\beta + C_{n_{\dot\beta}}\dot\beta + C_{n_p}p + C_{n_r}r + C_{n_{\delta a}}\delta_a + C_{n_{\delta r}}\delta_r \end{aligned} \tag{2.33}$$

here, co-efficients used in equation 2.31 are obtained from CFD. The angle of attack computed using UVW velocities, pqr from 6-DOF block. The aerodynamic forces and moments can now be expressed in wind frame using:

$$\overrightarrow{F}^{(w)}_{aero} = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix}, \quad \overrightarrow{M}^{(w)}_{aero} = \begin{bmatrix} L \\ M \\ N \end{bmatrix}$$

which can be expressed in body frame using a rotation matrix:

$$T_{BW} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -sin\alpha & 0 & cos\alpha \end{bmatrix}^T$$

$$\overrightarrow{F}^{(b)}_{aero} = T_{BW}\begin{bmatrix} -D \\ Y \\ -L \end{bmatrix}, \quad \overrightarrow{M}^{(b)}_{aero} = T_{BW}\begin{bmatrix} L \\ M \\ N \end{bmatrix}$$

2.5 Fuselage

The aerodynamic model of the fuselage is modelled similar to the wing. Although analogous, the aerodynamics of the fuselage differs fundamentally due to it being a bluff body. Hence, the model must be able to provide an estimate of the forces and moments at small angles and side-slip angles that will be encountered in forward flights. The model employed is a combination of linear interpolation using CFD data and equivalent flat plate area representation which is illustrated below.

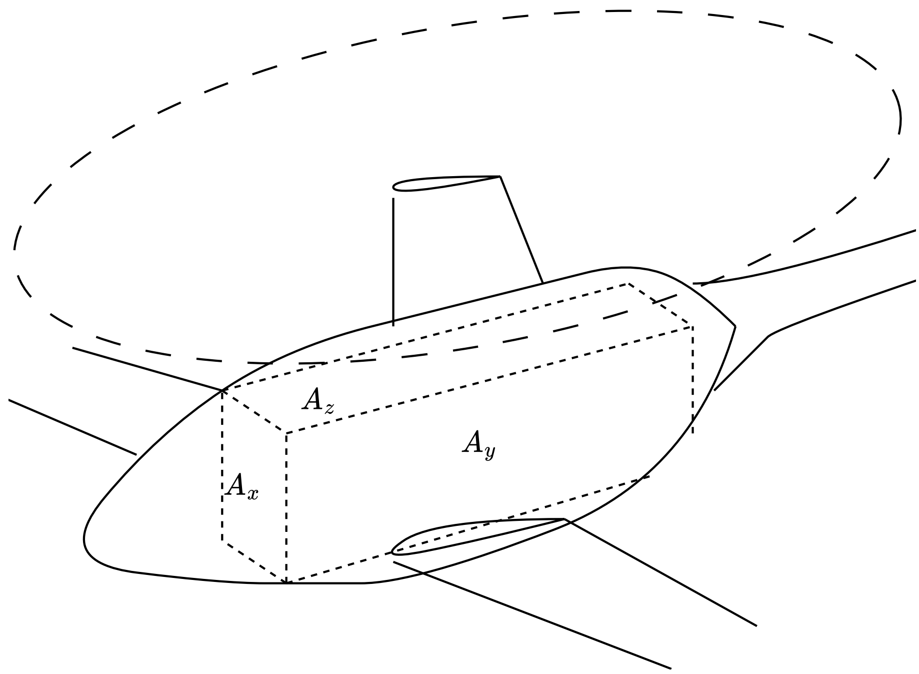

Figure 2.10: Flat Plate Area Representation

Some of the modelling assumptions are:

1. The aerodynamic forces are applied at the centre of pressure of the fuselage.
2. The location of the C.P does not move with change in angle of attack.
3. The fuselage does not contribute to any lift force.
4. It is assumed that the longitudinal forces and moments are dependent on the angle of attack and lateral forces on side-slip angle.

The fuselage forces and moments expressed in wind frame is given below:

$$\overrightarrow{F}^{(w)}_{fuse} = -\frac{1}{2}\rho \begin{bmatrix} A_x \times TAS(1) \\ A_y \times TAS(2) \\ A_z \times TAS(3) \end{bmatrix} \|TAS\|$$

$$\overrightarrow{M}^{(w)}_{fuse} = \overrightarrow{F}^{(w)}_{fuse} \times \begin{bmatrix} STA_{c.g} - STA_{c.p} \\ BL_{c.p} - STA_{c.g} \\ WL_{c.g} - WL_{c.p} \end{bmatrix}$$

The fuselage forces and moments can again be multiplied by T_{BW} to obtain the body frame forces and moments.

2.5.1 Wing and Fuselage Simulation

This section describes the results obtained from simulation of wing and fuselage. The simulations are run for same free stream velocities at different angle of attack, the flaps fully retracted, ailerons at zero deflection and finally, forces expressed below are in wind frame. This analysis can be used to validate the behaviour of the wing and fuselage aerodynamics.

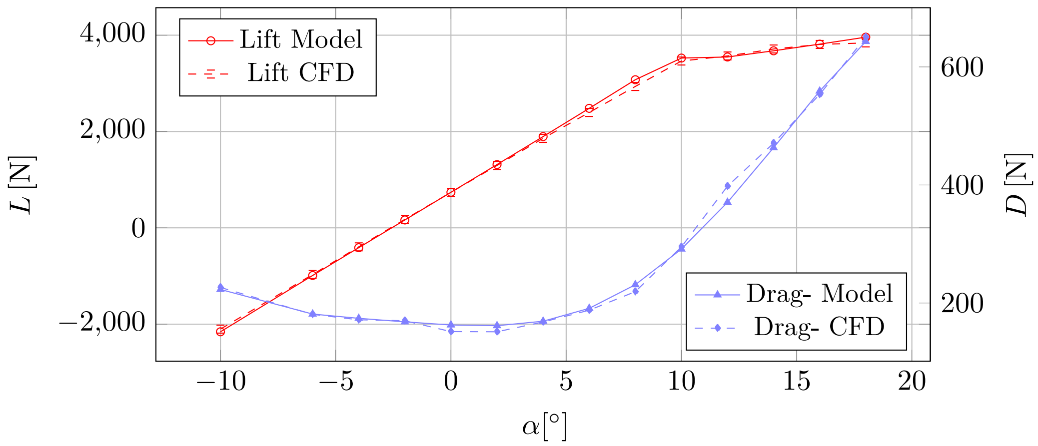

Figure 2.11: α vs Lift and Drag

The aerodynamic model of the wing and fuselage conforms very closely with the CFD results, the error on average is 2%. This is because the co-efficients are obtained from CFD and hence correlated very closely.

2.6 Environment

The aerodynamic environment is defined by the speed of sound, density, and kinematic viscosity. These quantities can be obtained using ISA Atmosphere Model which uses U.S. Standard Atmosphere.[8] The gravity is computed using, WGS84 Taylor Series gravity model. The assumptions when using this model are:

1. Gravity potential is assumed to be the same everywhere on the ellipsoid.
2. The Taylor's series model is accurate at low altitudes and decreases at medium and high geodetic heights.

The wind model used is "Dryden Wind Turbulence Model" which is used in later chapters to evaluate the control system against disturbance. However, during trim, no wind condition is used.

2.7 Complete Non-linear State-Space Representation

In the above sections, all the forces and moments are computed in body frame. These forces and moments are used to compute the time derivative of state vectors according to Newton-Euler rigid body equations. The non-linear rigid body dynamics are represented as:

$$\dot{x} = f(x, u) \tag{2.34}$$

The output of the 6 DOF is given by:

$$x = \begin{bmatrix} u & v & w & p & q & r & \phi & \theta & \psi & x & y & z \end{bmatrix}$$

while, the control input vector is given by:

$$u = [\delta_e \quad \delta_f \quad \delta_a \quad \delta_r \quad \delta_{th} \quad \delta_{\theta_0} \quad \delta_s \quad \delta_c \quad \delta_{ped}]$$

This model uses *MATLAB*'s pre-existing "Simple Variable Mass 6DOF" with Euler angle representation. While using this block, the main assumption of flat earth reference frame must be noted. The initial conditions for the simulations are obtained using trim procedure which is described in detail in the following chapter.

Chapter 3

Autorotation, Trim and Linearisation Analysis

In order for the aircraft to work as a gyrodyne, the rotor must be in autorotation while trim solution is required for the aircraft to remain in equilibrium in specified flight state. Finally linearisation procedure is required to understand the stability of the aircraft, design control system, etc. This chapter details each of the procedures that have been been incorporated along with some assumptions and shortcomings.

3.1 Autorotation

In the gyrodyne configuration, during forward flight, the rotor is allowed to freely rotate and the rotor spins naturally and is lifted by the aerodynamic forces acting on the blades. As the aircraft moves forward, which is powered by the propeller driven by the engine. The airflow interacts with the rotor which is titled back slightly relative to the airflow. This interaction generates lift and allows the aircraft to stay aloft. As long as the aircraft is in forward flight, the rotor will continue to autorotate. This process is illustrated below:

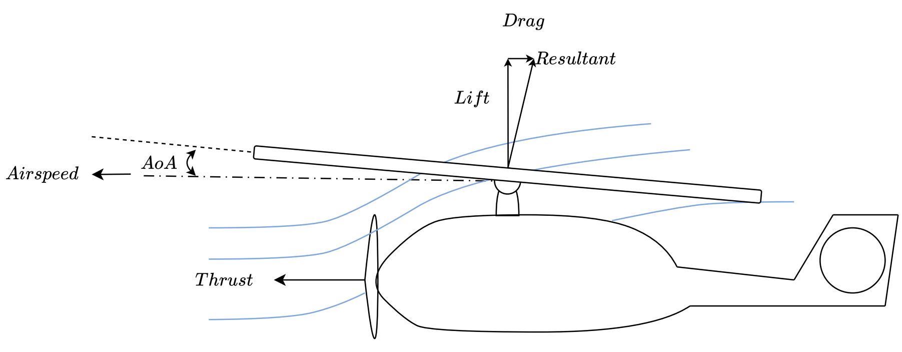

Figure 3.1: Autorotation in Forward Flight

To determine the autorotative state during forward flight, literature such as [9] used autorotation diagrams that employ airfoil's characteristics C_{d_0}/C_l plotted against angle of attack, α. While [4] used classical momentum and blade element theories using either the energy principle (net power analysis) and; force and torque balance on the rotor.

In this work, to find the autorotative states, net power is analysed. Since the rotor is already modelled using blade element and momentum theory. An iterative approach to find conditions where total torque, $Q = 0$ is used. Variables such as rotors angular rate, Ω vertical velocity, V_z and collective pitch, θ_0 is varied and the total torque is computed.

The rotors angular rate is discretized in the order of 10's of RPM considering the fact that rotor will be linked to gears. The range of angular rate considered is between 100-200 RPM, relatively low angular rate is to achieve low rotor drag while lift performance is not of paramount importance. The collective θ_0 in order of 1/10 of [deg]°, between range [-5,5]° and the V_z is used to compute the rotors required angle of attack using:

$$\alpha = tan^{-1}\left(\frac{V_z}{V_x}\right) \tag{3.1}$$

In the words of Jaun de la Cierva, [10] "It makes no difference at what angle the autogiro is climbing or flying. The blades are always gliding towards a point a little below the focus of forward flight. It is impossible, therefore, for autorotation to stop while the machine is going anywhere." So in other words, the gyrodyne's rotor in this case will remain in autorotation as long as the aircraft is flying forward. This however can be tested and the results are presented below although it must be noted that Cierva used rotor that was angled back as shown in figure 3.1. The *MATLAB*'s script used to to find the autorotative state is given below in Appendix A. The tolerance for auto-rotative state is said to be when $Q \leq 2[Nm]$. The results are given below.

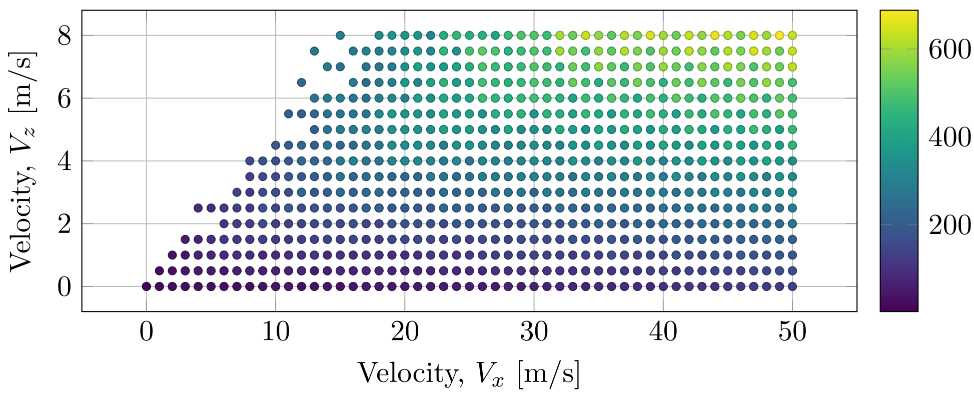

Figure 3.2: Autorotation envelope: vertical vs forward velocity

The above plot of forward velocity vs vertical velocity shows points where autorotation is possible, the contour represents the magnitude of thrust. It is seen that the thrust is smaller at small vertical velocity, it can also be seen that the vertical velocity is necessary for the rotor to remain in autorotation. This plot however is discrete and a continuous trend is expected in between these values.

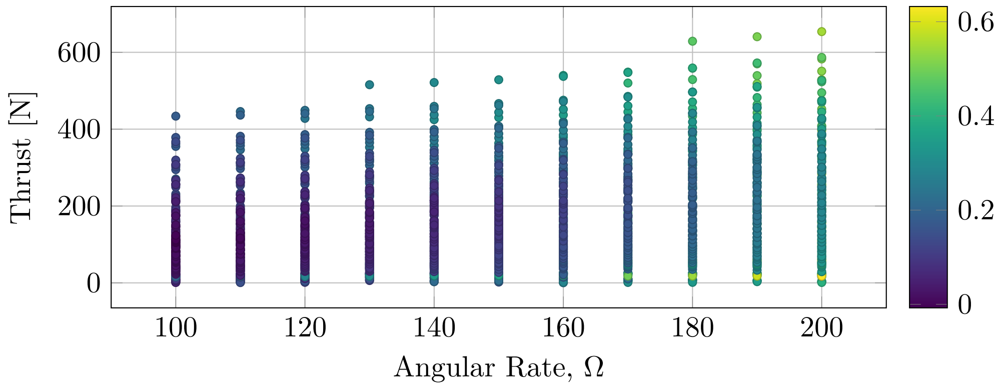

Figure 3.3: Thrust vs Rotational Velocity in Autorotation

It is now established that for reasonable thrust, V_z must be positive. Now for any positive V_z, it can see that from figure 3.3 the thrust increases as the angular rate increases where the contour shows the magnitude of power.

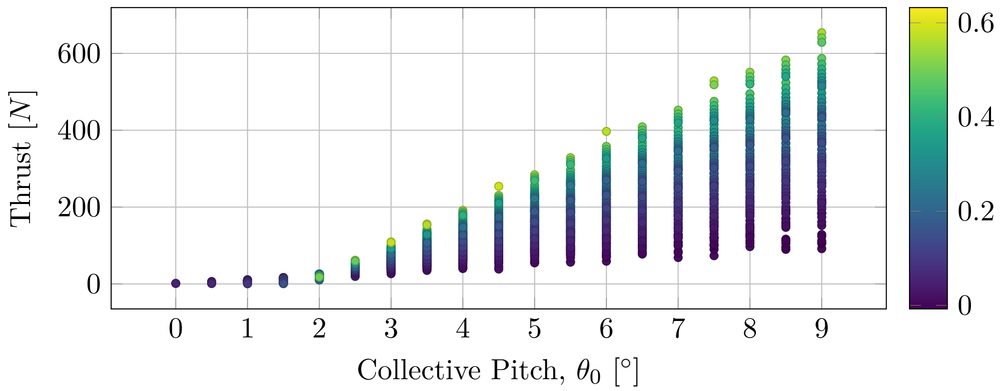

Figure 3.4: Collective pitch vs Thrust during autorotation

The above plot, it can be seen that the variation of thrust with change in collective, the contour again represents power. It is visible that changing the collective effectively increases the angle of attack which allows the blade to absorb more energy from the airstream which in turn can increase thrust, but also increases drag due to the increase in profile drag at large angle of attack.

Unlike Cierva's autogyro, since the gyrodyne also has wings, the lift force from the rotor and wings must be shared. The authors of [11] studied the optimum lift sharing and concluded that the rotor caries $8 - 9\%$ of the gross weight while the wing caries $91 - 92\%$ of gross weight. While this might be configuration specific, there is always a compromise between the aircraft's pitch required to have positive V_z going through the rotor and drag experienced by the whole aircraft due the pitch up manoeuvre. More detailed study with higher fidelity simulations such as CFD must be performed in order to evaluate lift sharing. In this thesis, during cruise, the lift is shared between rotor and wing by 10% and 90%; therefore the configuration where this is possible is picked.

3.2 Trim

The aircraft trim operation solves for the controls and motion that produce equilibrium in the specified flight state. The trim procedure in the simulation is necessary in order start the simulation. Different trim solutions are required for various flight states, therefore two main trim states are defined for analysis.

- Hover
- Forward Flight

For any trim procedure, the target for the total forces and moments are always zero. The trim state and control variables are:

$$x_0 = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi]$$

$$u_0 = [\delta_e \quad \delta_f \quad \delta_r \quad \delta_a \quad \Omega_{MR} \quad \theta_{0_{MR}} \quad \theta_s \quad \theta_c \quad \Omega_{TR} \quad \theta_{0_{TR}} \quad \Omega_{PR} \quad \theta_{0_{PR}}]$$

Now, trim procedure is defined by a pair x_0, u_0 such that $f(x_0, u_0) = 0$. That is

$$\dot{x}\big|_{x=x_0,\ u=u_0} = 0 \tag{3.2}$$

It can observed that the u_0, x_u has a total of 12 and 8 variables, but there are only have 6 equations i.e.,

$$\begin{aligned} \sum F_i &= 0, \\ \sum M_i &= 0 \quad \text{where,} \quad (i = x, y, z) \end{aligned} \tag{3.3}$$

hence, the system is under-determined. Although some of these variables such as Ω_{MR}, Ω_{TR}, Ω_{PR} are design configuration and hence can be omitted based on the flight state. However, there are still 6 equations and 9 variables and below, the procedure used is explained in detail.

3.2.1 Hover Trim

In hover, the gyrodyne configuration is more closely operated as a helicopter i.e., main rotor and tail rotor are the main sources of forces and moments. Trimming algorithms such as [12] and [13] are used to trim a helicopter. However, these algorithms based on "Powell's Dogleg" used in *fsolve* are used to trim for the complete dynamics of helicopter which include harmonic terms for the rotor which are neglected here. Also, since the forces on propeller and aerodynamic forces are be neglected, it can be reduced to:

$$x_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad \phi \quad \theta \quad 0]$$

$$u_0 = [0 \quad 0 \quad 0 \quad 0 \quad 600 \quad \theta_{0_{MR}} \quad \theta_s \quad \theta_c \quad 3000 \quad \theta_{0_{TR}} \quad 0 \quad 0]$$

In the above case, since the aircraft is in hover, the velocities and the angular rates are set to zero. Hence, with equal number of equations and variables, the non-linear equation can be solved using *lsqnonlin* which solves non-linear least-square problems. Since non-linear equations can have infinite number of solutions, it is necessary to pick a feasible solution that the aircraft is capable of remaining in stable equilibrium. To do so, bounds are set between physical actuation possible to limit the number of solutions and the tolerance for termination is set to 1×10^{-12}. Below are the trim results in hover for the aircraft in different altitudes.

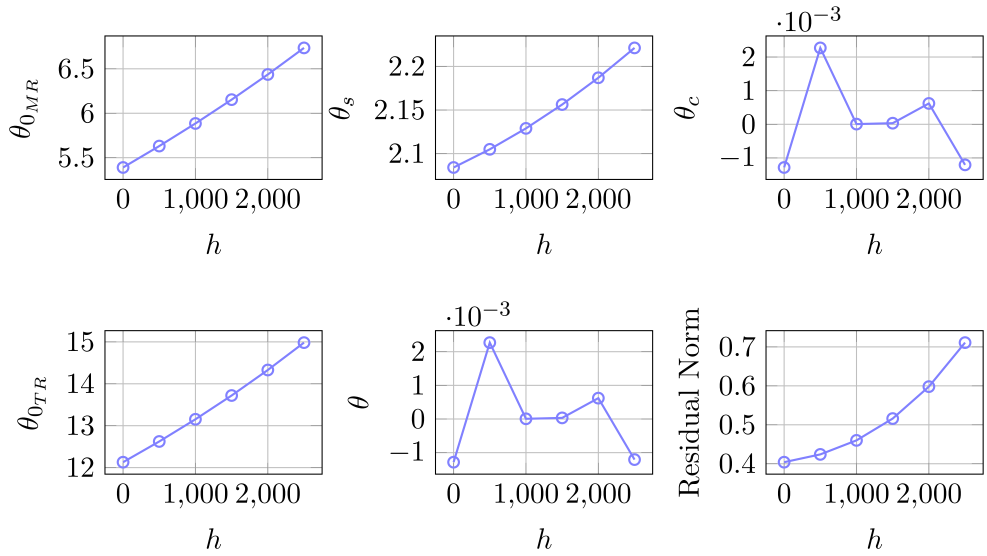

Figure 3.5: Trim variables as function of height

The hover simulations for the gyrodyne are performed, during the simulation, the height of hover was set to $10m$ and no wind disturbance was included. Below is a time history of the gyrodyne in hover trim:

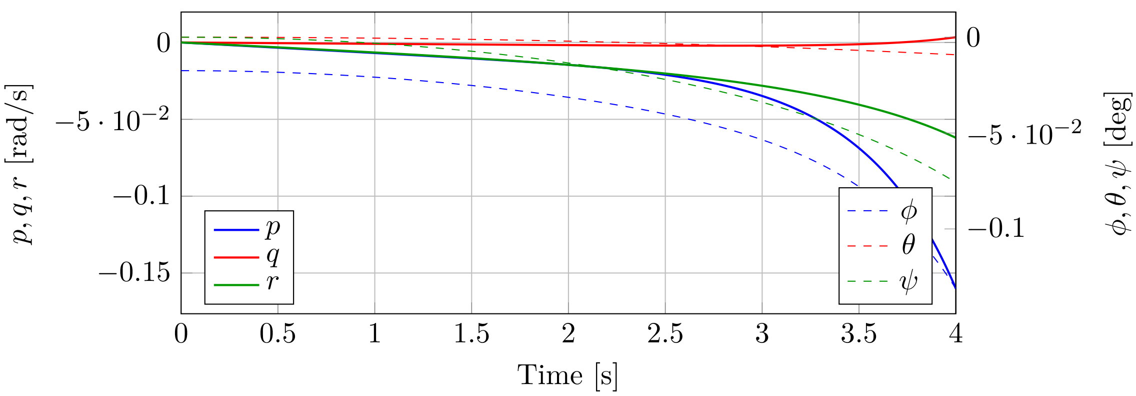

Figure 3.6: Angular rates and Euler angles vs time in hover trim

It can be seen from the graphs that the aircraft is inherently unstable. The aircraft begins from a good but not perfect trim condition and but with time, quickly becomes unstable. This is due to the fact that the flapping of the main rotor blade is not present. Usually, with a conventional helicopter, the flapping hinge causes the rotor disk to tilt opposite to roll rate which passively stabilizes ensuring that they do not diverge violently. Evidently, without passive stability, an active control system with fast response rate will be required to stabilize the aircraft in hover.

3.2.2 Forward Flight Trim

In forward flight state, the trim procedure is slightly different because now there are additional control variables to solve for. During forward flight, the the tail rotor forces can be neglected. Variables, $\delta_f, \theta_s, \theta_c$ are also set to zero during trim.

$$x_0 = [u \quad v \quad w \quad p \quad q \quad r \quad \phi \quad \theta \quad \psi]$$

$$u_0 = [\delta_e \quad 0 \quad \delta_r \quad \delta_a \quad \Omega_{MR} \quad \theta_{0_{MR}} \quad 0 \quad 0 \quad 0 \quad 0 \quad \Omega_{PR} \quad \theta_{0_{PR}}]$$

First, the autorotation procedure described in section 3.1 is used to determine the main rotors angular rate and collective pitch. Variables such as $\delta_e, \delta_r, \delta_a, \theta_{PR}, \phi_0$ and θ_0 using *lsqnonlin*. The cost function is set to 1×10^{-12} and control actuation physical bounds are set to each control variables. The results of the trim procedure is shown below:

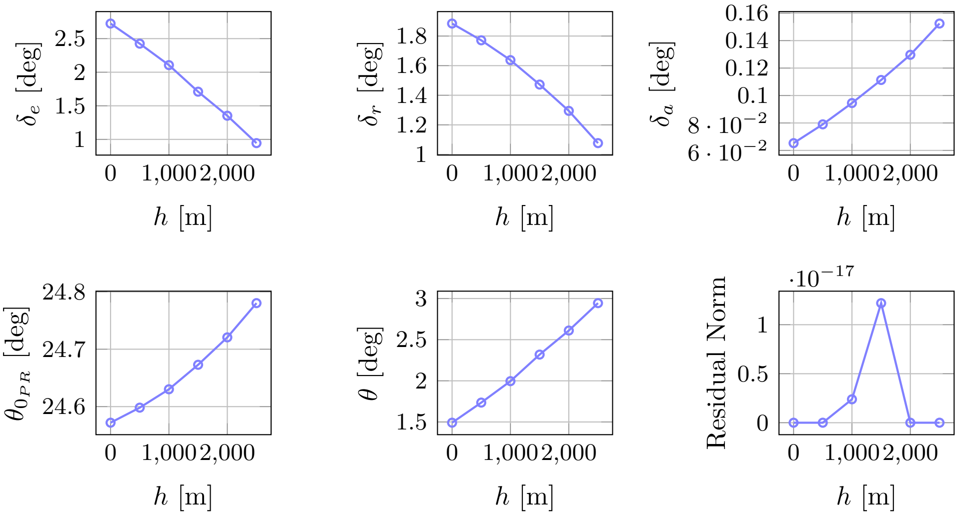

Figure 3.7: Trim variables as a function of altitude (h)

The above trim procedure was performed at cruise velocity of 40 m/s with flaps completely retracted and γ equal to zero. It can be seen from the velocity and Euler angles plot that the aircraft starts from a sustainable trim state.

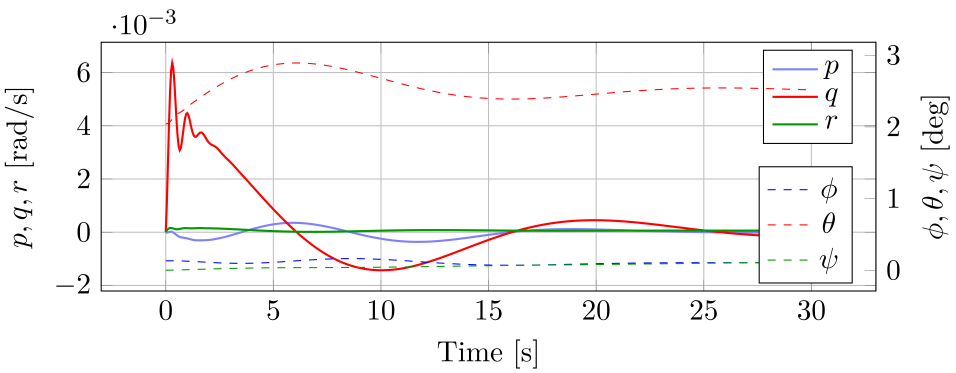

Figure 3.8: Angular rates and Euler angles in forward flight trim

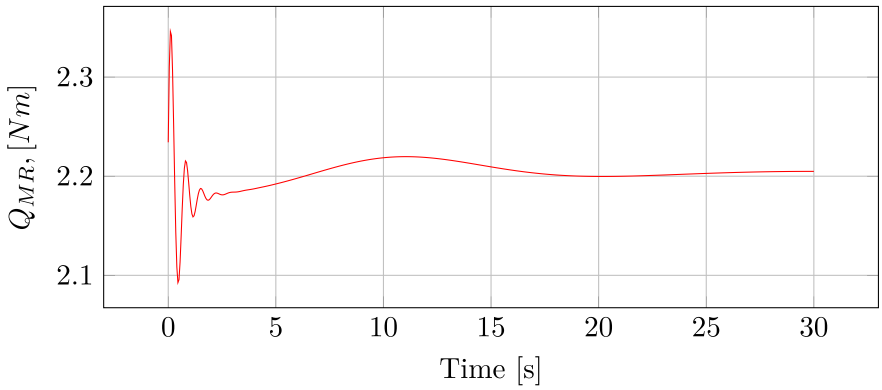

Figure 3.9: Time history of main rotor torque [Nm]

The transient in the beginning comes from the fast dynamics and the subsequent oscillatory behaviour subsides. Plot of main rotor torque's time history also shows that the main rotor is in stable autorotation. Nonetheless, the aircraft drift slowly to the right which can be corrected using any closed loop control system. The result of the simulation for 30 seconds initialised in trim condition is shown in the above figure.

3.3 Linearisation

The dynamics and stability of the gyrodyne is the least studied within the rotorcraft category, as such aircraft's are typically homebuilt or experimental. To understand the behaviour of the aircraft it is essential to study the linear algebraic small-perturbation equations. First developed by G.H Bryan [14], the linear equations is derived algebraically from non-linear equations. The standard state equation for a linear differential system has the form:

$$\dot{x}\left(t\right) = Ax\left(t\right) + Bu\left(t\right) \tag{3.4}$$

where for a time-invariant system, A is a constant $n \times n$ matrix and B is a constant $n \times k$ matrix. The standard output equation has the form

$$y\left(t\right) = Cx\left(t\right) + Du\left(t\right) \tag{3.5}$$

where C is constant $l \times n$ matrix and D is constant $l \times k$ matrix. A generalised derivation of linear system equations are given in [15] with assumption of constant mass, flying over a flat and non-rotating earth. In this work, linearisation about nominal trajectory is done using the same standard perturbation method, which uses a central difference scheme

$$y = \frac{f(x+h,u) - f(x-h,u)}{2h} \tag{3.6}$$

By linearising the non-linear aircraft equations of motion about the trim condition, the state-space matrices A and B can be obtained for the aircraft. The decoupled, longitudinal and lateral-directional matrices along with their dynamical behaviour are explain in detail below.

3.3.1 Longitudinal Dynamics

The state co-efficient matrix, A and driving matrix, B are obtained algebraically using method from equation 3.6. The equation 3.4 can be represented as a transfer function by taking the Laplace transform and time response can be plotted to unit step input.

$$A_{lon} = \begin{bmatrix} X_u & X_w & X_q - W_0 & -g\cos\theta_0 & X_\Omega \\ Z_u & Z_w & Z_q + U_0 & -g\sin\theta_0 & Z_\Omega \\ M_u & M_w & M_q & 0 & M_\Omega \\ 0 & 0 & 1 & 0 & 0 \\ Q_u & Q_w & Q_q & 0 & Q_\Omega \end{bmatrix} \quad B_{lon} = \begin{bmatrix} X_{\delta_e} & X_{\delta_{th}} & X_{\delta_s} \\ Z_{\delta_e} & Z_{\delta_{th}} & Z_{\delta_s} \\ M_{\delta_e} & M_{\delta_{th}} & M_{\delta_s} \\ 0 & 0 & 0 \\ Q_{\delta_e} & Q_{\delta_{th}} & Q_{\delta_s} \end{bmatrix}$$

Observing the state-space matrices, there are additional co-efficients, this is because the longitudinal dynamics of the gyrodyne differs quite differently to typical fixed wing or the helicopter.

- In a fixed wing, the longitudinal dynamics exhibit two dominant modes (short period and phugoid) however, for the gyrodyne the rotors effect cannot be ignored.
- The stability derivatives are not present in a helicopter because it is assumed that the rotor speed remains constant, also the torque changes are quite small under normal flight conditions [16].

Knowing the importance of the rotor speed mode, the stability of the gyrodyne can now be studied using the eigenvalue analysis.

$$|\lambda I - A| = 0 \tag{3.7}$$

where, I is a 5×5 identity matrix. By expanding the determinant, can be written as a fifth degree polynomial.

$$\lambda^5 + a_1\lambda^4 + a_2\lambda^3 + a_3\lambda^2 + a_4\lambda + a_5 = 0 \tag{3.8}$$

The aircraft maybe said to be dynamically stable if all its eigenvalues have negative real parts. The root locus of the gyrodyne is plotted below:

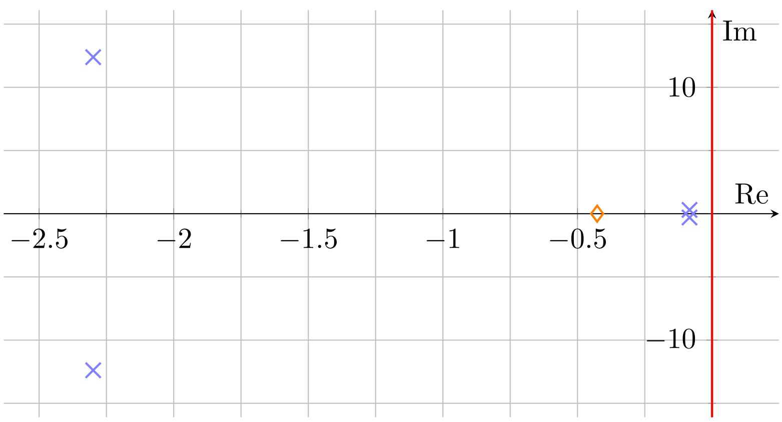

Figure 3.10: Root locus showing the longitudinal poles

The table below shows the longitudinal eigenvalues along with rotor-speed mode.

Mode	Eigenvalue	Damping (ζ)	ω_n [rad/s]
Short Period	$-0.0838 \pm 0.3i$	0.269	0.312
Phugoid	$-2.3 \pm 12.4i$	0.183	12.6
Rotor-speed	-0.427	1	0.427

Table 3.1: Longitudinal Eigenvalues and Stability Characteristics

The root locus shows distinct short period and phugoid mode with additional rotor-speed mode represented by ◊. The aircraft is moderately damped in short period while phugoid is weakly damped. The rotor-speed mode is stable and has no oscillatory behaviour, this can be further understood in detail by investigating the stability derivatives.

$$\dot{\Omega} = Q_u u + Q_w w + Q_q q + Q_\Omega \Omega + Q_\theta \theta \tag{3.9}$$

Among these derivatives, the major contributor to rotor-speed mode is Q_Ω which describes change of aerodynamic torque with rotor RPM. This can indeed be either stabilizing or destabilizing term depending on which region of "torque bucket", the rotor is in autorotation.

In section 3.1, it was evident that there are n-number of autorotation points; however, not all of them are a viable option in terms of stability of gyrodyne. Below, a single autorotation point is picked (although a dramatic point is picked to illustrate the idea).

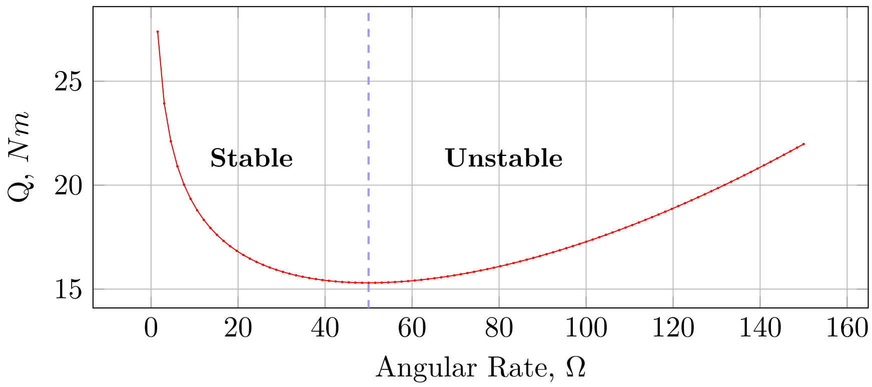

Figure 3.11: Torque Variation with angular rate in autorotation

The torque bucket illustrated above helps understanding the 'sign' of the Q_Ω. If the angular rate of the rotor is on the left side of the curve, when disturbed, the drag causes the rotor to slow down hence providing a stabilizing effect. On the right side of the curve, when the rotor is disturbed, the angular rate continues to increase which which makes it dynamically unstable. Hence, it is important for Q_Ω to be strictly negative for longitudinal stability. Now, to visualise the effect of rotor coupling, the step response is plotted below.

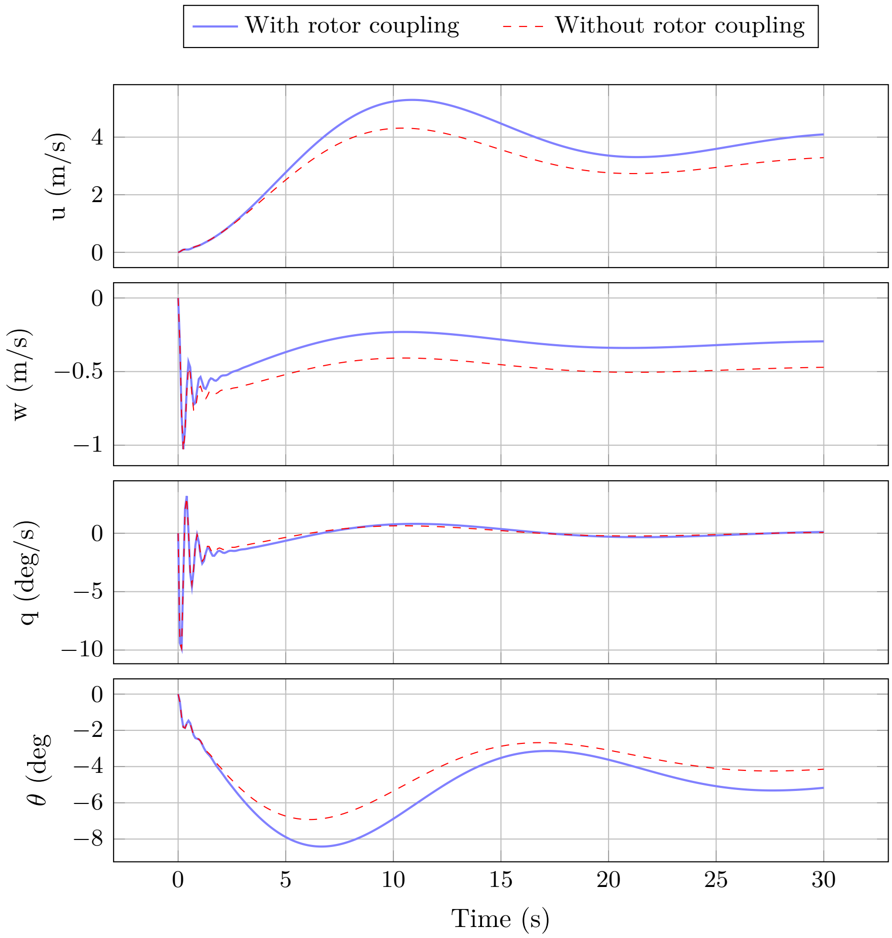

Figure 3.12: Comparison of longitudinal state step responses.

The rotor mode is stable however affects the longitudinal dynamics of the aircraft and is important for achieving effective automatic flight control, including stability augmentation, is paramount nonetheless.

3.3.2 Lateral-Directional Dynamics

The characteristic polynomial of the lateral-directional can be derived similarly from the A_{lat} and B_{lat} matrices. The space-state matrices for the lateral-directional dynamics can be written as:

$$\mathbf{A}_{lat} = \begin{bmatrix} Y_v & Y_p + W_0 & Y_r - U_0 & g\cos\theta_0 \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & \tan\theta_0 & 1 \end{bmatrix} \qquad \mathbf{B}_{lat} = \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} & Y_{\delta_c} \\ L_{\delta_a} & L_{\delta_r} & L_{\delta_c} \\ N_{\delta_a} & N_{\delta_r} & N_{\delta_c} \\ 0 & 0 & 0 \end{bmatrix}$$

In lateral-directional dynamics, the state-space matrices are the same as the one for fixed wing, in essence, the effect of rotor is minimal. This is because the rotor's torque variation is negligible with perturbation of v velocity hence keeping the rotors angular rate relatively constant.

In the same manner as for the longitudinal dynamics, the characteristic polynomial for the lateral stability can be studied using $|\lambda I - A| = 0$, this yields a fifth degree polynomial:

$$\lambda^5 + b_1\lambda^4 + b_2\lambda^3 + b_3\lambda^2 + b_4\lambda + b_5 = 0 \tag{3.10}$$

Which when factorized gives directional, roll and dutch-roll modes. The root locus of the lateral-directional modes are plotted below:

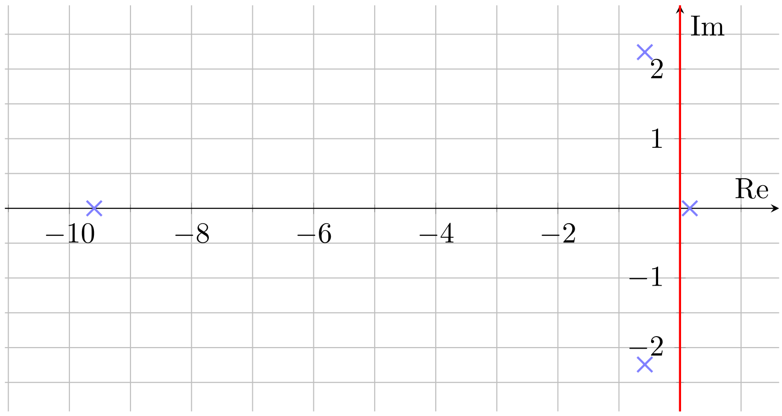

Figure 3.13: Root locus showing the lateral poles

The lateral modes exhibit strong damping in roll, moderate-low damping in dutch-roll with adequate oscillation frequency. The spiral mode is unstable although very small which is indeed a desirable state because it is advantageous to have higher dutch-roll frequency over spiral stability. The table below show the lateral eigenvalues and stability characteristics.

Mode	Eigenvalue	Damping (ζ)	ω_n [rad/s]
Roll	-9.59	1	9.59
Dutch-roll	$-0.576 \pm 2.24i$	0.249	2.32
Spiral	0.16	-1	0.16

Table 3.2: Lateral Eigenvalues and Stability Characteristics

These results suggests that gyrodyne exhibits conventional lateral-directional stability characteristics. However, as discussed previously, some of the interactions such as rotor-wing, rotor-propeller and more have not been taken into considerations while computing the aerodynamic derivatives which can indeed change the stability of the gyrodyne. Also, all the analysis has been done assuming C.G location does not change, while this can change depending on the location of fuel tank, this must also be taken into consideration.

For the aforementioned reasons, any control system designed for the aircraft should have a sufficient phase and gain margin in order for the aircraft to have stable behaviour to account for uncertainties. The lateral-directional state step responses are plotted below.

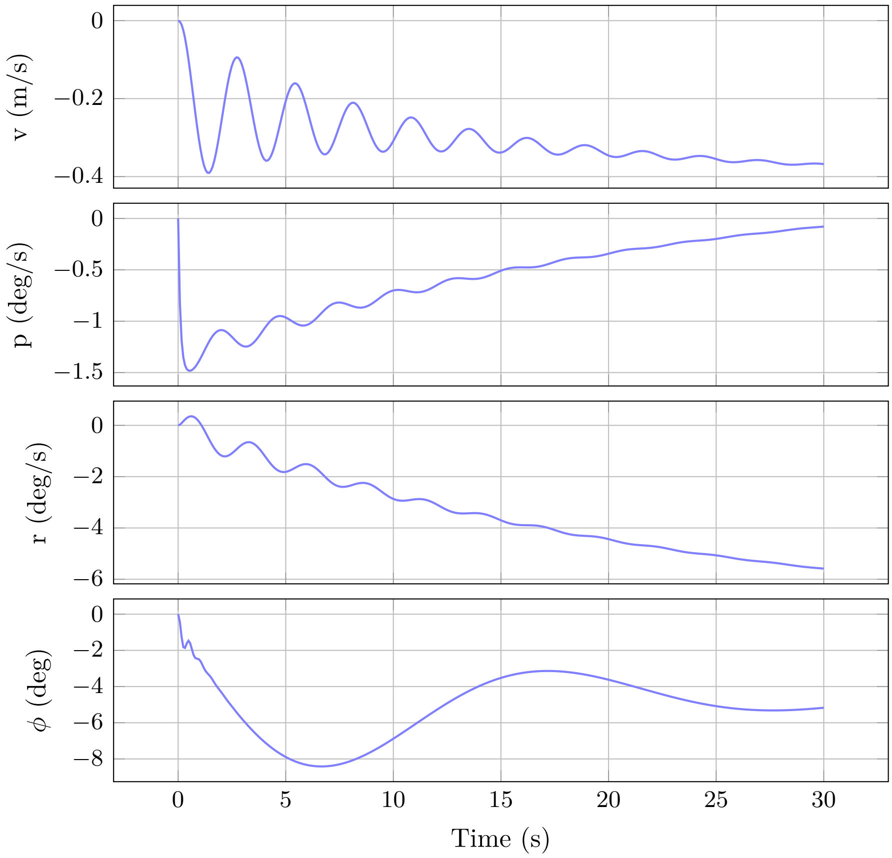

Figure 3.14: Lateral state step response

3.3.3 Limitations of Linear Model

Linearisation gives a good insight into the aircraft dynamics and stability characteristics; however, it is only an accurate representation for small perturbations about a specific trimmed flight condition. When the aircraft operates at different flight conditions, the model must be re-linearised to capture the corresponding changes in aerodynamic behaviour and control effectiveness. For example, the inherently non-linear aerodynamic force and moment variations are approximated by local stability derivatives such as Z_w, M_q, and Y_v, which are valid only in the vicinity of the trim point. As the angle of attack, sideslip angle, dynamic pressure, or actuator deflections change, these derivatives can vary significantly, reducing the accuracy of the linear model. Consequently, linearised models are most suitable for local analysis and controller design, whereas the full non-linear model is required to accurately represent aircraft behaviour over a wider operating envelope.

Chapter 4

Control System

In order to control the aircraft, certain state variables are controlled in order to behave in desired way. A control system is used to enhance the flying qualities of the aircraft or to place the aircraft in desired state. In this chapter, classical control techniques are used to design a Stability Augmentation System (SAS) and attitude control system. This will serve as a baseline for control allocation in the following chapters.

4.1 Control Architecture

Traditionally, an aircraft is controlled with two loops, an inner stability augmentation loop and an outer attitude control loop. Usually, SAS inputs are considered secondary, the flying quantities are enhanced by the control action of the feedback control system in way that that suppress disturbances on the aircraft. For this gyrodyne configuration, there are multiple actuators that can achieve the same final output, i.e., elevator and longitudinal cyclic can be used to control the pitching moment on the aircraft, and lateral-directional can use ailerons, rudders and lateral cyclic. In this chapter traditional controls such as elevator, rudder and ailerons are used for control. The feedback gains for the Proportional Integral Derivative (PID) control is tuned using *sisotool* from *MATLAB*'s. The control architecture block diagram is given below:

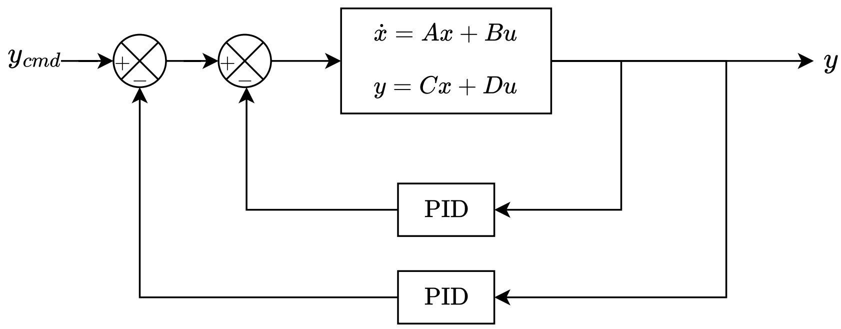

Figure 4.1: Control Architecture

Although a Stability Augmentation System (SAS) influences the aircraft stability derivatives, only certain derivatives can be effectively modified to enhance flying qualities. Stability derivatives such as C_{m_α},C_{m_q}, C_{n_β}, C_{n_r}, C_{n_p}, and C_{l_β} can be obtained more effectively using automatic control than by physically sizing the aerodynamic surfaces. In this work, conventional SAS found on modern aircraft such as pitch rate SAS, yaw damper, roll rate damper. While attitude control system form the inner loop of the path control system. This work is adopted from work of Donald McLean [17].

The actuators used in the aircraft are electric and the dynamics of the actuators in this design is considered to be fast enough to be regarded instantaneous which is represented by this second order transfer function:

$$G(s) = \frac{1421}{s^2 + 45.24s + 1421} \tag{4.1}$$

Another factor that can affect the control system is the sensor dynamics, in this work, the sensor models are not included and the controller uses the state variables directly from the 6DOF simulation. While noise from the sensor is not present, this can indeed affect the performance of the control system. However sensors have a in-built filters to improve the signal to noise ratio (SNR).

4.1.1 Requirements

For the control dynamics some of the requirements are listed below, these requirements are drawn from SAE AS94900 [18], ADS-33E-PRF [19] and [20].

Parameter	Requirement Value	Units
Damping Ratio (Short Period)	$\geqslant 0.35$	-
Frequency (Short Period)	$\geqslant 6$	$[rad/s]$
Damping Ratio (Phugoid)	$\geqslant 0.04$	-
Frequency (Phugoid)	$0.1 - 0.5$	$[rad/s]$
Gain Margin	$\geqslant 6$	dB
Phase Margin	$\geqslant 45$	$[deg]^\circ$
Phase Delay	$\leqslant 0.1$	s

Table 4.1: Desired performance for control system

Other qualitative desired performance include:

- $\geq 30s$ of stabilized hover in desired region.
- ± 0.5m deviation (both longitudinal and lateral) from hover centre.
- No undesirable motions (bobble, overshoots/undershoots) that impact task performance during the transition to hover or stabilized hover.

Because of the unique nature of the gyrodyne, no specific requirements are drawn up and so a combination of fixed wing and rotary aircraft in sub-150 kg class is chosen.

4.2 Longitudinal Control Design

In this section, the longitudinal control system is developed for cruise flight with elevator as the main control surface. A pitch rate SAS serves as the inner loop with pitch attitude control system on outer loop. The control system design, tuning and performance are given below.

4.2.1 Pitch Rate SAS

The pitch rate SAS is used to augment M_q which increases the damping ratio of the short period. The block diagram of conventional pitch rate SAS is given below.

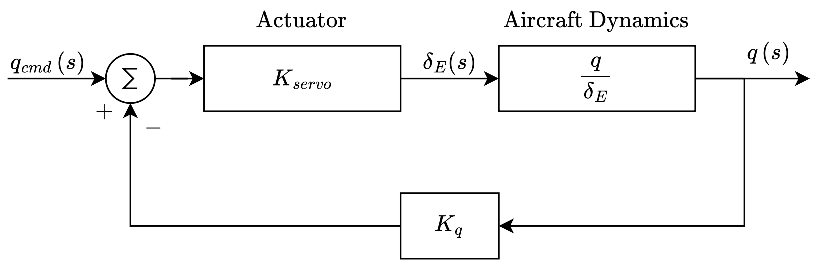

Figure 4.2: Pitch rate stability augmentation system

The control system used above only affects the short period motion of the aircraft and the phugoid motion is unaffected. The pitch rate response to 1° step input is plotted below.

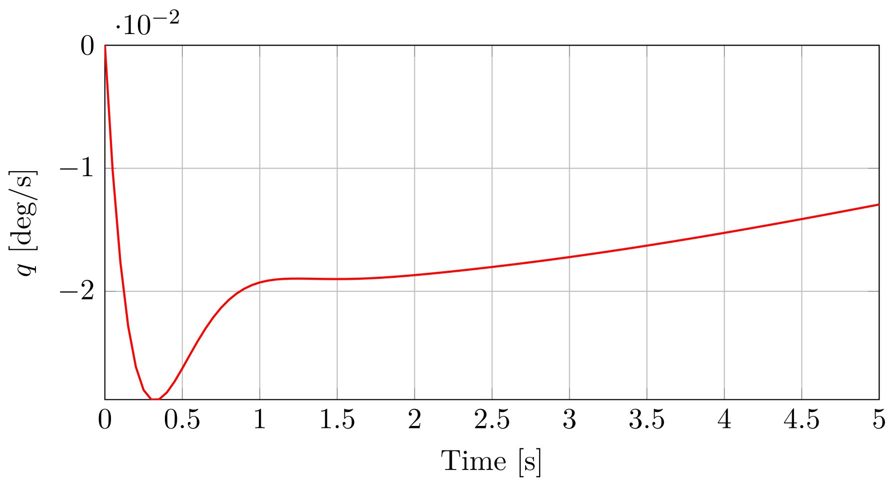

Figure 4.3: q vs Time

With the gain, K_q, the damping ratio increases from 0.269 to 0.501 and its frequency is 13.1 [rad/s]. The controller significantly decreases oscillations in short period mode effectively increasing damping. A damping of 0.5 satisfies the requirements set in table 4.1. Also, it be seen that there is a clear transition to phugoid due to the inherent small damping, this can be addressed using a slower outer loop with Proportional Integral (PI) control to remove any steady state error.

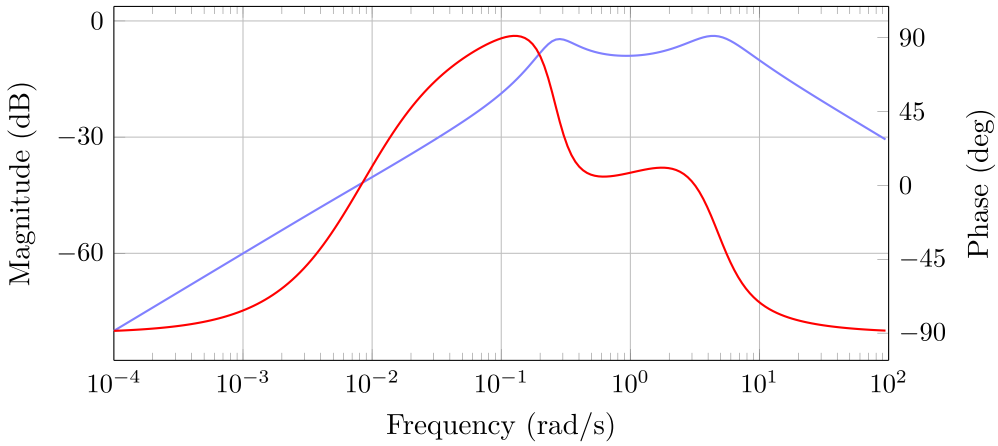

Figure 4.4: Bode diagram and Nichols diagram

The Bode and Nichols diagram for the system is shown in figure 4.4. The large gain margin of 328 *dB* and phase margin of 116 *deg* makes the controller tolerant of modelling uncertainties. Additionally, feedback terms such as normal acceleration, $a_{z_{c.g}}$ can be used to achieve required handling qualities or using phase advance compensation (lead-compensator) to counteract the lag of the system, actuator, etc. While for now, no such techniques have been used in order to maintain simplicity.

4.2.2 Pitch Attitude Control System

Pitch attitude control system is used to place and maintain an aircraft at a desired orientation in space and again, elevator is used as the primary control surface. With large K_θ in the pitch attitude control system can increase the short period frequency (ω_{sp}) but also decrease the damping ratio (ζ_{sp}). The phugoid mode is also affected by the pitch feedback loop, this affects the phugoid period increases until they become non-oscillatory. The block diagram of pitch attitude control system is shown below:

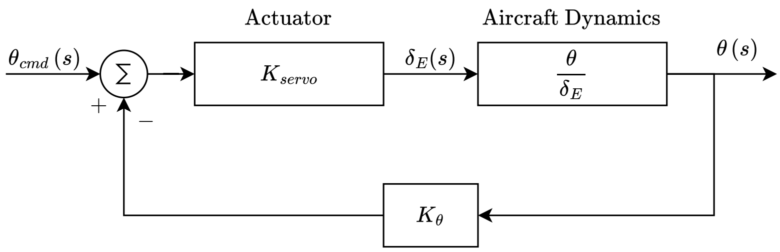

Figure 4.5: Pitch attitude control system

The block diagram shows the feedback structure, the measured pitch attitude from the 6DOF is fed back through K_θ which generates elevator command. The control law uses Proportional Integral controller tuned using *sisotool*. The step response of pitch attitude control system is shown in the figure below.

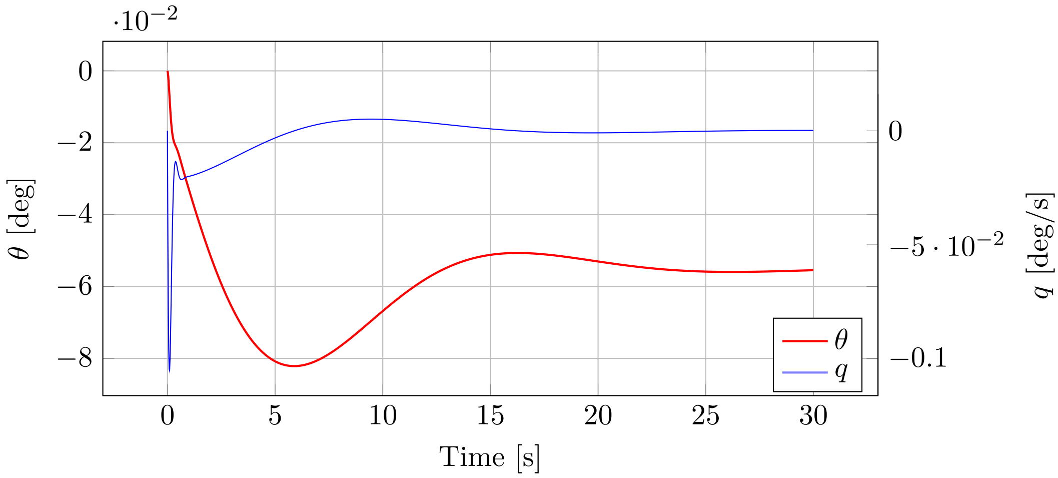

Figure 4.6: Pitch angle and pitch rate vs Time

The figure shows the step response to a step input of 1 $[deg]$ using the control law. The steady state error is very small because the control law uses the integral term. The transient response shows that the aircraft reaches commanded pitch attitude with small overshoot. Also, the response is well damped with damping ratio of 0.65.

4.3 Lateral-Directional Control Design

For lateral-directional control, there are usually three modes: roll, spiral and dutch roll. The lateral-directional motion is controlled with two control surfaces, rudder and ailerons, and from table 3.2, the damping on dutch roll is very small which makes execution of co-ordinated turns very difficult. The lateral-directional motion will be controlled similar to longitudinal with one faster-inner loop and a slower-outer loop.

4.3.1 Yaw Damper

As discussed in the previous section, the aircraft possesses sufficient inherent Dutch-roll damping to satisfy the specified handling-quality requirements. Nevertheless, the mode remains oscillatory in nature and may still lead to undesirable yaw-rate oscillations. To further improve lateral-directional response characteristics, a yaw damper can be implemented by augmenting the yaw damping derivative, N_r, through conventional rate-feedback control. It must also be noted that the actuator for the rudder is usually quite slow and an appropriate transfer function for the rudder actuator must be used. The block diagram for the yaw damper is shown below:

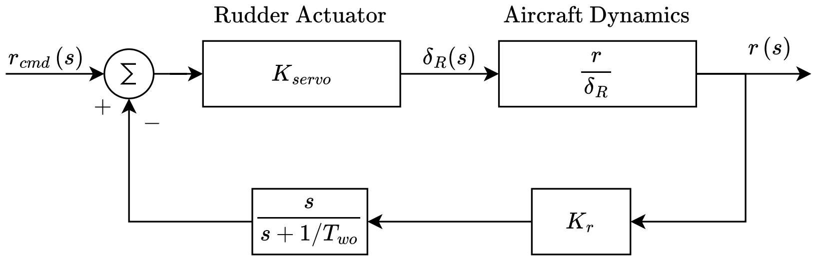

Figure 4.7: Yaw damper block diagram

To increase the damping ratio to be greater than 0.4, the transfer function r/δ_R is used to obtain the appropriate gain and tuned further using *sisotool*. The step response to a unit step input of $1[deg/s]$ is plotted from the complete lateral-directional motion.

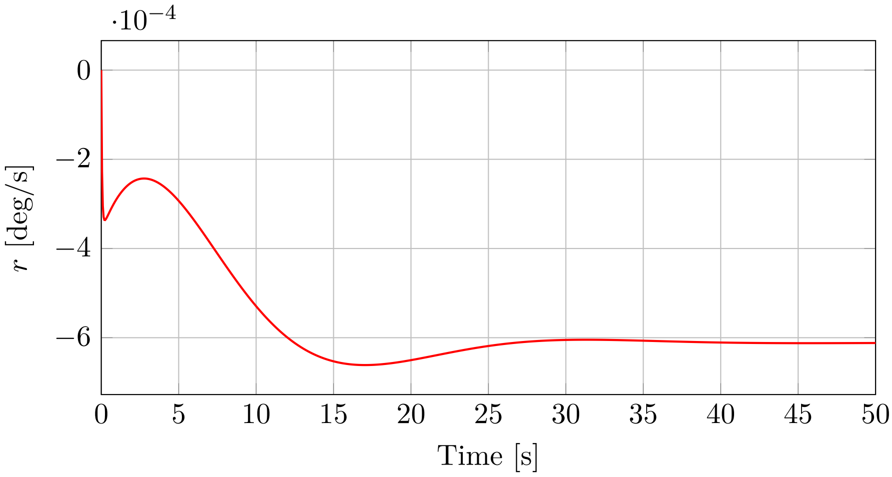

Figure 4.8: Step response to yaw rate

It can be seen that from the simulation that with an initial disturbance, there is an absence of oscillatory motion. The new damping ratio is 0.535 with natural frequency of 0.261 [rad/s]. With the stability derivative, L_r being large, there is significant coupling of dutch-roll into the rolling subsidence; hence the transfer function r/δ_R cannot be simplified with approximations. The gain and phase margins are 8.14 [dB] and 52.7 [deg]. A wash-out filter with T_{wo} of $1s$ is used in order to distinguish between a disturbance and commanded input.

4.3.2 Roll Rate Damper

Roll rate damper is used to improve the roll performance of the aircraft, which augments the stability derivative L_p. This reduces the response time of the aircraft. The gyrodyne has strong inherent roll damping with roll subsidence pole of $9.62s^{-1}$. In other words, the roll response is quite adequate which also benefits dutch-roll damping. Hence, the roll rate damper is not used for this aircraft. The step response to the approximated and complete transfer function, p/δ_A is given below:

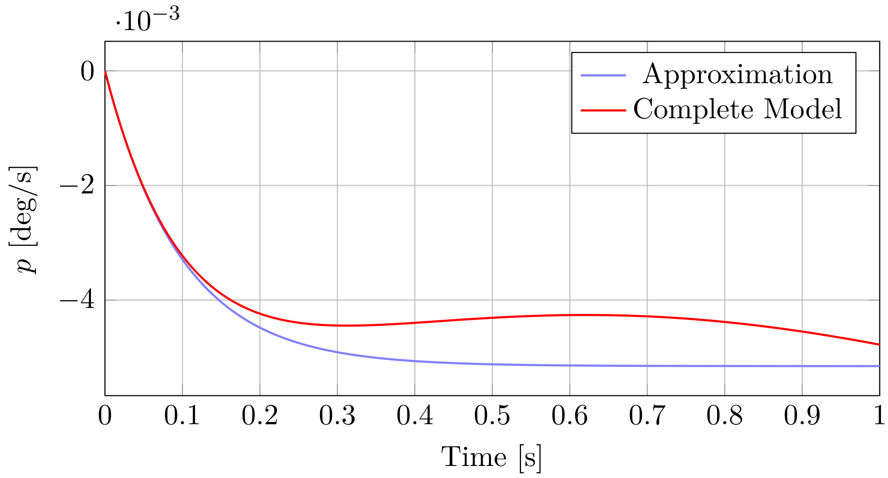

Figure 4.9: Roll rate response

4.3.3 Roll Angle Control System

To control roll angle, ailerons can be used to create the rolling moments. In order to design a control system, the aircraft dynamics along roll is usually approximated by assuming large spiral time, $T_R \gg T_s$ which gives:

$$\frac{p}{\delta_A} = \frac{L_{\delta_A}}{s - L_p} \tag{4.2}$$

Although in this case, since the spiral time is relatively small, the spiral pole cannot be ignored and the controller is tuned using the complete model. The block diagram of the roll angle control system is given below:

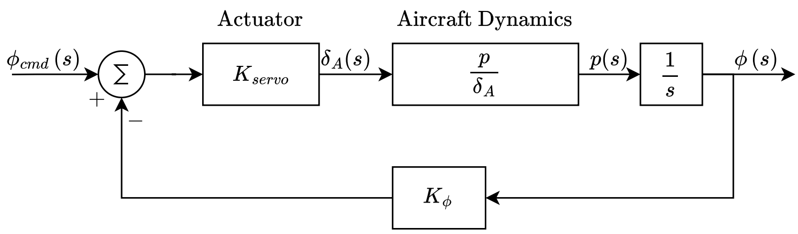

Figure 4.10: Bank angle control system

The step response of the bank angle control system is given below. Proportional and integral control law is required to stabilize the roll mode in case the model is tuned using complete model and proportional is sufficient if tuning approximate function, although this induces significant dutch-roll when used on complete dynamics.

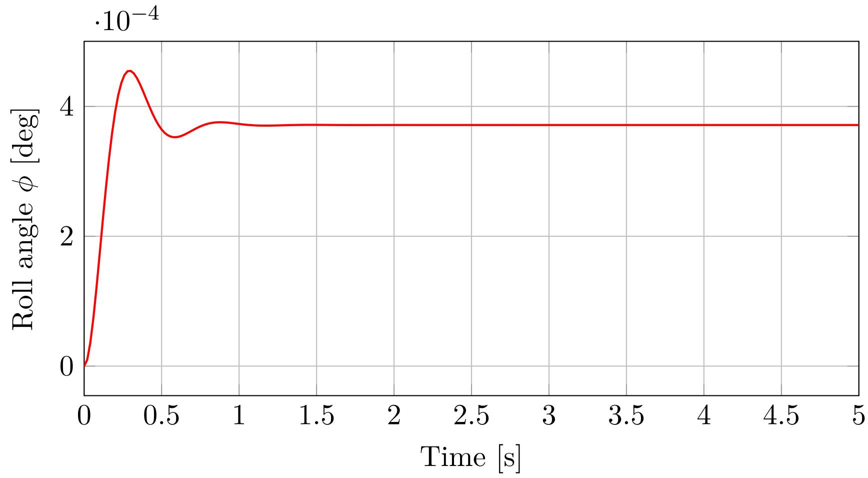

Figure 4.11: Step response to bank angle control system

4.3.4 Spiral Mode Stabilization

The aircraft is inherently unstable in spiral mode as it can be seen from the figure 3.13 with spiral time period of $6.23s$ which is very small. Usually, the spiral mode time is a magnitude larger. Due to the coupling motion, the aircraft cannot to be perfectly stable in spiral and have good dutch-roll damping. Since, this gyrodyne configuration has the stability derivative, N_{δ_A} positive, the spiral mode can be stabilized without causing dutch-roll to become unstable (for small gains). The transfer function r/δ_A is used to stabilize the spiral mode and the block diagram is given below:

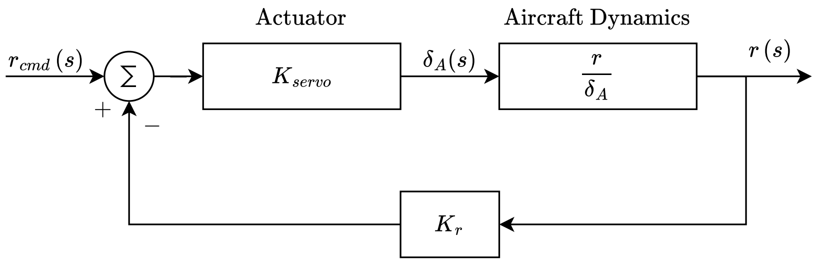

Figure 4.12: Spiral mode stabilization system

4.4 Flight Path Control System

With the help of flight stability augmentation system, the flying qualities have been brought closer to the desired values from table 4.1 and later, attitude control system was employed to maintain desired orientation of the aircraft in both longitudinal and lateral axis. Now, in the following section, flight path controls are used to maintain the aircraft in desired flight conditions such as height, heading and airspeed. Attitude control system will form the inner loop of the flight path control system and is used to navigate, along a desired way-point or follow trajectory.

4.4.1 Height Control System

The aircraft's altitude is either controlled or maintained by means of elevator deflection. An altimeter is used to measure the height (usually barometric altimeter is used for higher altitudes and radio altimeter for precise low altitude), regardless of the sensor, it can be assumed that the altimeter provides error free readings when designing the controller. The block diagram of the height hold system is illustrated below:

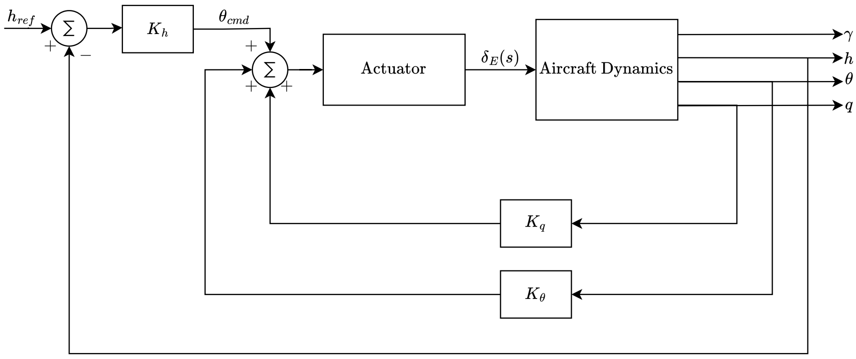

Figure 4.13: Height control system

The transfer function, h/δ_E can be written as:

$$\frac{h}{\delta_E} = \frac{1}{s^2}\left(\frac{-a_z}{\delta_E}\right) \tag{4.3}$$

which can be further tuned using *sisotool*, with Proportional Derivative (PD) control law. The step response to elevator input is given below, it can be observed the response time is relatively large, this is done so the gain margin does not drop below the threshold set during the requirements. A small steady state error exists but adding a Integrator in the control law introduces another pole which leads to -270° phase lag and leads to instability.

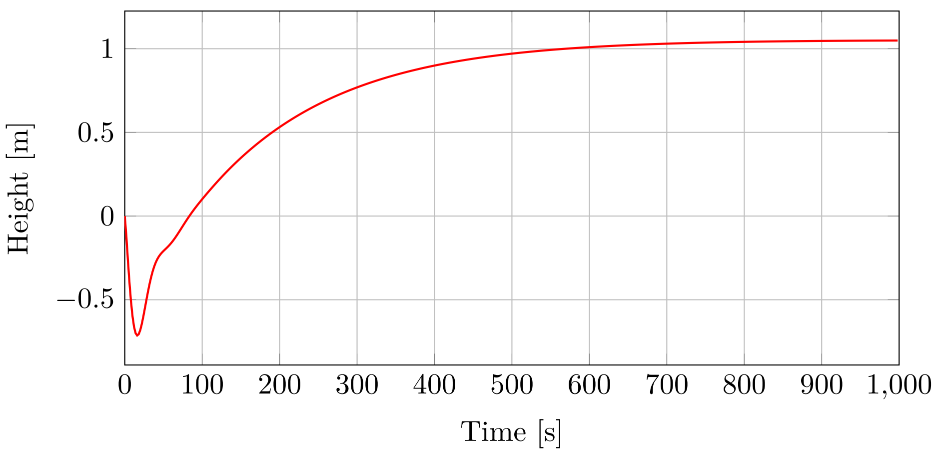

Figure 4.14: Response of height control system

From the bode plot given below, it can be seen that higher gains pushes the gain margin less than 6 dB which is not desirable. Very small gain margins can make the aircraft unstable when operating in different conditions or when unmodelled physics are experienced by the aircraft.

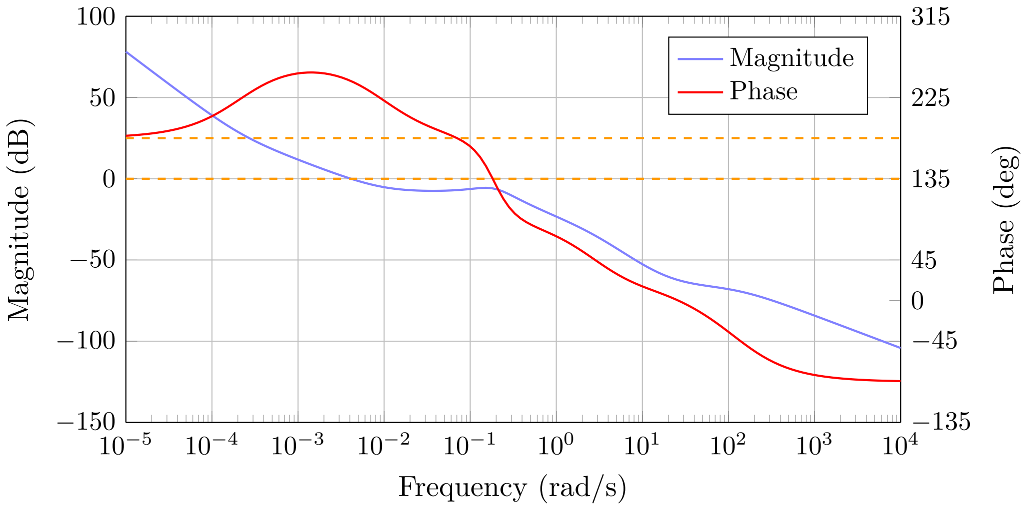

Figure 4.15: Bode plot for height hold system.

4.4.2 Airspeed Control System

The speed of the aircraft is essential to control the aircraft's flight path and to ensure the aircraft always remains within the operational limits, for example, the aircraft's velocity has to always be above the stall velocity and below the maximum velocity. Speed can be controlled by changing the throttle position, δ_{th} which changes the fuel flow for the engine. A block diagram representing typical airspeed control system is shown below:

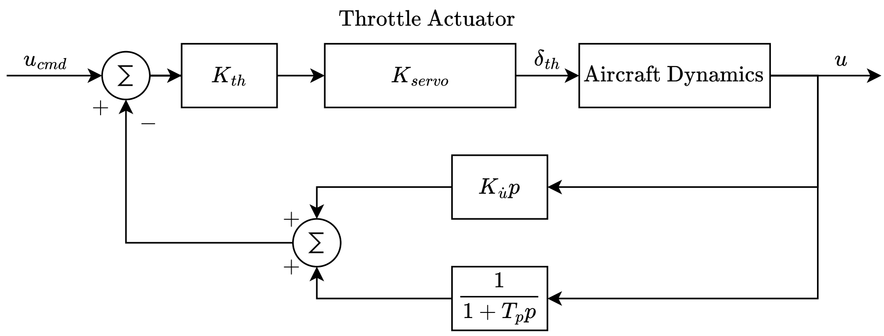

Figure 4.16: Airspeed control system

A proportional-integral type controller is used, the integrator is used to remove any steady state error. Also since the airspeed is measured using Pitot tubes (barometric device), it is represented by a first order transfer function to model the delay of the sensor and the time delay, T_p used is $0.1s$. Assuming 10% authority, the proportional gain can either be computed using $T = W(D/L)$ or by tuning the transfer function ${}^{u}/_{\delta_{th}}$. The step response is plotted below:

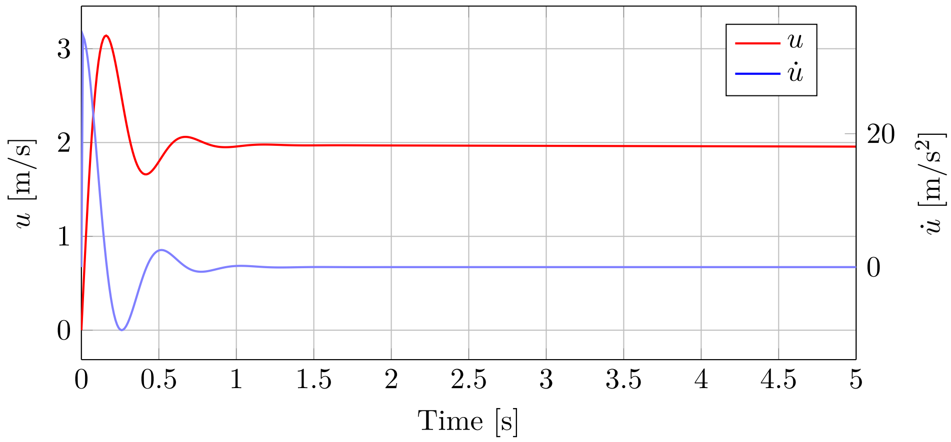

Figure 4.17: Response of airspeed control system

The response of the airspeed control system depends significantly on the airspeed sensor time delay and they actuator transfer function. Slower actuation leads to oscillations before transitioning to steady state. The error on steady state reaches a small value with the proportional controller alone, hence integrator is not required.

4.4.3 Direction Control System

To steer an aircraft along a particular direction, ailerons are used to manoeuvre and the heading of the aircraft is taken as its yaw angle, ψ. In this case, it is assumed that no sideslip angle, β is present when making a turn i.e., all turns are co-ordinated. The block diagram for a direction control system is illustrated below:

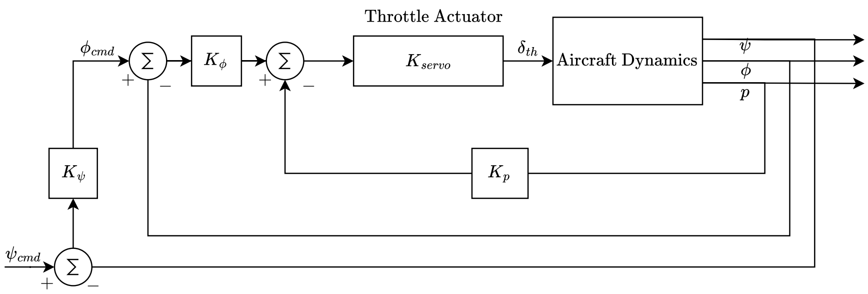

Figure 4.18: Direction control system

The direction control system serves as the outer loop of the roll angle control system with control law: $\phi_{cmd} = K_\psi \left(\psi_{ref} - \psi\right)$ where the controller gain is obtained by tuning the transfer function:

$$\frac{\psi}{\delta_A} = \frac{1}{s^2} \frac{g}{U_0} \left(\frac{p}{\delta_A}\right) \tag{4.4}$$

With the assumption that $r = (g/U_0)\,\phi = \dot{\psi}$. The approximate transfer function p/δ_A is used to tune and the unit step response of the complete system is shown below, there is negligible difference in terms of behaviour.

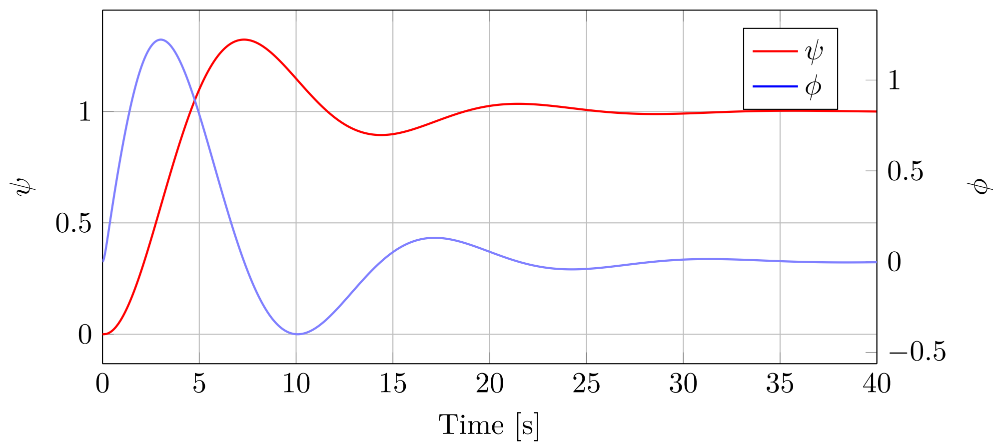

Figure 4.19: Step response of direction control system

The yaw angle and roll angle is plotted and it can be observed that the settling time is relatively longer. The maximum sideslip angle experienced by the aircraft is 0.192°.

4.5 Performance and Stability Evaluation

Evaluation of controllers performance is essential to understand how effective and controllable the system is to inputs. It can serve as a basis to verify if the control requirements are satisfied during the operation. The most common way to do is by defining performance and stability indices. First and foremost, the closed loop eigenvalues can be analysed to understand if the flying quality requirements such as the one described in table 4.1 are satisfied. These directly impact the handling quality of the aircraft and can give first look into how the aircraft behaves [21].

4.5.1 Stability and Performance Metrics

Traditionally, control systems are certified by ensuring they meet specification for stability margins among other things. Typically addressed using gain and phase margin as a safeguard against unmodelled effects and uncertainity. The stability metrics using the phase and gain margins, defined here as M_1 and M_2 metrics where:

$$\begin{aligned} M_1 &= GM \\ M_2 &= PM \end{aligned} \tag{4.5}$$

Another index that can be used is the Time-delay Margin and is defined as the time delay, t_d in the control signal that the closed loop system can tolerate without instability. The time delay margin can be obtained from simulations i.e., time delay t_d is introduced into control design and then adjusted until it is on the verge of instability. Usually the threshold of instability is defined as twice the nominal system response.

$$M_3 = TDM \triangleq t_d \tag{4.6}$$

Similarly, performance metrics include transient performance metrics that look at overshoot or undershoots. One transient performance metric is the $\mathcal{L}_\infty$ norm of the tracking error relative to reference input and is defined as

$$M_4 = \frac{\|x(t) - x_m(t)\|_{\mathcal{L}_\infty}}{\|x_m(t)\|_{\mathcal{L}_\infty}} \tag{4.7}$$

where, $x_m(t)$ is the state of the reference model, $\|.\|$ is the Euclidean norm.
To evaluate the Steady-State performance, and is defined as the time it takes to quickly settle to a steady state operating condition and is determined from the simulation. More formally, the time required for the response curve to reach and stay within a range of certain percentage (usually 2% or 1%) of the final value [22].

Control effort must be evaluated in order to ensure that the actuator constrains are satisfied. Control limits can be measured either using $\mathcal{L}_2$ or $\mathcal{L}_\infty$ while verifying that the actuator position and rate limits are not exceeded.

4.5.2 Closed Loop Stability Analysis

The inherent stability of the aircraft was studied in section 3.3 using eigenvalue analysis, now the same process can be repeated using the designed controller. The root locus plot of longitudinal and lateral-directional dynamics are presented below:

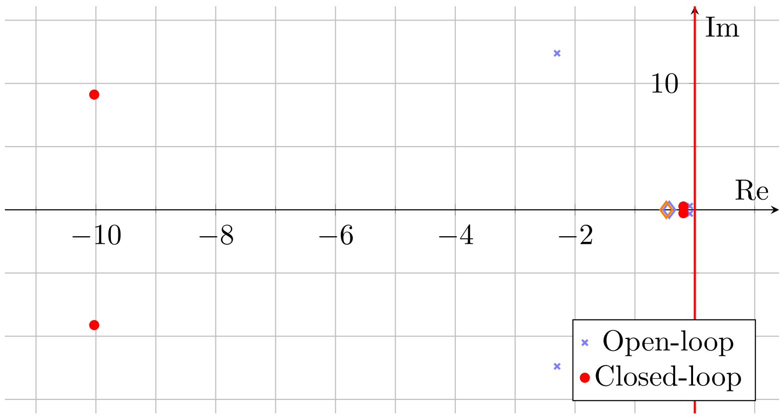

Figure 4.20: Longitudinal root locus showing pole migration

The table below shows the closed loop longitudinal eigenvalues:

Mode	Eigenvalue	Damping (ζ)	ω_n [rad/s]
Short Period	$-0.191 \pm 0.267i$	0.582	0.328
Phugoid	$-10 \pm 9.12i$	0.74	13.6
Rotor-speed	-0.476	1	0.476

Table 4.2: Closed Loop Longitudinal Eigenvalues and Stability Characteristics

The eigenvalues of the longitudinal dynamics have good short period and phugoid damping. The rotor-speed mode damping has not changed significantly and continues to be stable.

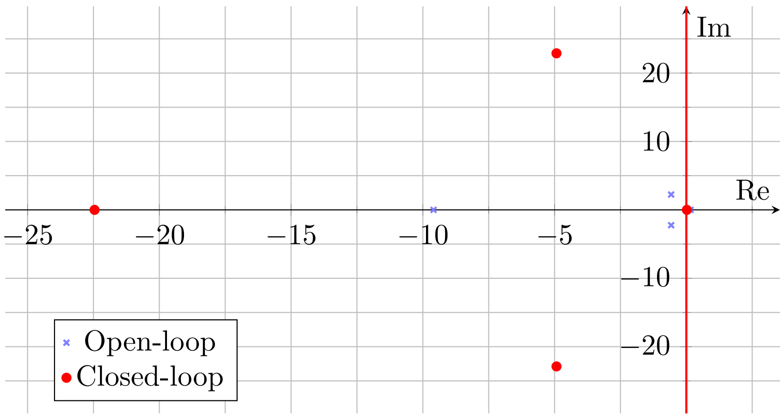

Figure 4.21: Lateral-Directional root locus showing pole migration due to control.

The table shows the closed loop lateral-directional eigenvalues:

Mode	Eigenvalue	Damping (ζ)	ω_n [rad/s]
Roll	-21.9	1	21.9
Dutch-roll	$-1.07 \pm 2.06i$	0.461	2.32
Spiral	-0.062	1	0.062

Table 4.3: Closed loop Lateral-Directional Eigenvalues and Stability Characteristics

The roll mode shows relatively good dutch-roll damping while also maintaining stable spiral.

4.6 Limitations of Classical Control

Classical control design is based on linearised models which is specific to one operating point. This means that the performance is only guaranteed locally. Any variation to the conditions means that either the controller is operating outside its optimality or the controller must be tuned a priori at different conditions and gain scheduling must be performed. For example, conventional aircraft's make use of the later option where the control engineers tune gains at multiple operating conditions discretized in n operating points and stored as n-dimensional matrices. With this, not only does the task of tuning increase but also the appropriate gains must be sorted and picked by the flight computer in real time.

A fundamental limitation that arises from the face that controller is tuned using linear time invariant (LTI) model but the actual aircraft's exhibit non-linear dynamics. This results in controller which may not accurately capture the system behaviour for large perturbation.

Also, when the number of loops increases, especially in Multi-Input Multi Output system with coupled dynamics, the design process becomes difficult and the success in terms of performance and stability is not guaranteed across the full operating envelope. These limitations can motivate the use of advanced non-linear control system which explicitly accounts for non-linearities and different operating conditions.

Chapter 5

Control Allocation

The gyrodyne configuration considered in this work combines conventional aerodynamic control surfaces with rotor-based control effectors, resulting in a redundant control system. Consequently, multiple combinations of actuator commands can generate similar forces and moments, making the determination of control inputs a non trivial task.

The flight control system developed in the previous chapters generates the force and moment commands required to track a reference trajectory. These commands were previously distributed directly among the primary aerodynamic control surfaces, namely the aileron, elevator, and rudder. However, additional control authority is available through the rotor tilt actuators, which can be exploited to realize a larger portion of the attainable moment set. A control allocation scheme is therefore required to distribute the commanded moments among the available control effectors while respecting actuator limits.

Conventional linear allocation methods are attractive due to their simplicity and low computational cost. However, their performance may be limited when the control effectiveness exhibits significant non linearities. Optimization based approaches, such as Quadratic Programming (QP) allocators [23], can account for these non linear effects and actuator constraints more effectively. Nevertheless, such methods require the solution of an optimization problem at every control update, resulting in an increased computational burden that may be undesirable for real-time implementation on resource-constrained unmanned aerial vehicles.

To address this issue, a neural-network-based non linear control allocation framework is proposed adopted from [24]. The network is trained offline to learn the inverse control effectiveness map and is subsequently employed as a real-time allocator. By replacing the online optimization process with a computationally efficient neural-network inference, the proposed approach aims to retain the benefits of non-linear allocation while significantly reducing computational complexity. The performance of the proposed allocator is evaluated against a baseline allocation strategy through non-linear flight simulations.

5.1 Control Effectors and Authority Analysis

The gyrodyne considered in this work combines conventional aerodynamic control surfaces with a rotor-based thrust vectoring mechanism. In addition to the aileron, elevator, and rudder, the vehicle employs rotor disc tilting to redirect the main rotor thrust vector. This combination results in a redundant control configuration in which multiple effectors contribute to the same body forces and moments.

This redundancy leads to overlapping control authority, particularly during forward flight, where both aerodynamic surfaces and rotor thrust vectoring simultaneously influence the aircraft dynamics. As a result, the mapping between control inputs and the resulting force – moment vector is not unique, motivating the need for a structured control allocation strategy.

The control effectors considered in this work are the elevator, aileron, rudder, and rotor disc tilt (longitudinal and lateral tilt components). Their qualitative influence on the body force and moment vector:

$$\begin{bmatrix} F_x & F_y & F_z & L & M & N \end{bmatrix}^T$$

is summarized in Table 5.1.

Actuator	F_x	F_y	F_z	L	M	N
Elevator	Low	Low	Low	Low	High	Low
Aileron	Low	Low	Low	High	Low	Low
Rudder	Low	Low	Low	Low	Low	High
Rotor Disc Tilt (Longitudinal)	High	Low	High	Low	High	Low
Rotor Disc Tilt (Lateral)	Low	High	High	High	Low	Low

Table 5.1: Qualitative control authority of gyrodyne effectors.

5.2 Baseline Control Allocation Method

As a benchmark for evaluating the proposed neural-network-based control allocation scheme, a linear pseudo-inverse allocation strategy is implemented. The control allocation problem is formulated using a linear control effectiveness model of the form:

$$\tau_d = B\mathbf{u} \tag{5.1}$$

where $\tau_d \in \mathbb{R}^3$ represents the commanded moment vector, $\mathbf{u}$ denotes the vector of control effector inputs, and B is the control effectiveness matrix.

In the baseline implementation, the available control inputs are defined as:

$$\mathbf{u} = \begin{bmatrix} \delta_a & \delta_e & \delta_r & \delta_c & \delta_s \end{bmatrix}^T \tag{5.2}$$

corresponding to aileron, elevator, rudder, and rotor disc tilt (lateral and longitudinal components). To resolve redundancy and obtain a unique solution, a pseudo-inverse based allocation is used:

$$\mathbf{u} = B^{\dagger}\tau_d \tag{5.3}$$

where $B^{\dagger}$ denotes the Moore–Penrose pseudo-inverse of the control effectiveness matrix.

In practice, the control allocation is implemented with a fixed weighting structure that prioritizes aerodynamic control surfaces over rotor-based actuation. This is achieved by defining a weighting vector:

$$\mathbf{w} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}^T \tag{5.4}$$

which effectively biases the solution such that aileron, elevator, and rudder are used as primary control effectors, while rotor disc tilt inputs are minimized under nominal conditions. This baseline strategy is used as a reference case for comparison with the proposed neural-network based control allocation method.

5.3 Neural-Network-Based Control Allocation

As discussed in the previous sections, the gyrodyne possesses multiple control effectors capable of generating overlapping force and moment contributions. The resulting control allocation problem is inherently redundant, since several combinations of actuator commands can produce similar aircraft responses. Furthermore, the interaction between aerodynamic control surfaces and the rotor thrust-vectoring mechanism introduces non linearities and coupling effects that vary throughout the flight envelope.

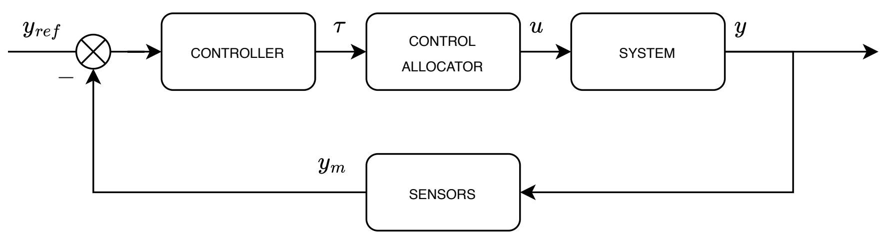

Figure 5.1: Feedback-loop structure with Control Allocation

Artificial Neural Networks (ANNs) provide an alternative framework for addressing non linear control allocation problems. Due to their universal function approximation capability, ANNs can learn complex non linear mappings directly from data. Instead of solving an optimization problem online, the desired mapping between commanded moments and actuator deflections can be learned offline and subsequently evaluated through a simple forward pass of the network.

In this work, the non linear control allocation problem is formulated as the inverse of the control effectiveness map. Rather than determining the actuator commands through iterative optimization, a neural network is trained to approximate the function that maps desired moments and flight condition information to the corresponding control effector commands. Once trained, the network acts as a real-time allocator capable of exploiting the available control redundancy while accounting for the non linear characteristics of the gyrodyne dynamics.

The following sections describe the network architecture, the generation of the training dataset, and the training procedure used to develop the proposed neural-network-based control allocator.

5.3.1 Network Architecture

The control allocator is implemented as a fully connected feed forward artificial neural network and the architecture is described in the below figure 5.2. The input vector consists of the commanded moment vector and key aerodynamic states, selected to capture variations in control effectiveness across the flight envelope. Specifically, the input is defined as:

$$\mathbf{x} = \begin{bmatrix} L_d & M_d & N_d & \alpha & \beta & q \end{bmatrix}^T \tag{5.5}$$

where L_d, M_d, and N_d are the desired roll, pitch, and yaw moments, α is the angle of attack, β is the sideslip angle, and q is the dynamic pressure.

The inclusion of dynamic pressure as an input is motivated by the strong dependence of aerodynamic control effectiveness on the flight condition. Since the moments generated by a given control deflection are proportional to dynamic pressure, incorporating q allows the neural network to adapt the allocation strategy across different combinations of airspeed and altitude without requiring separate allocators for each operating condition.

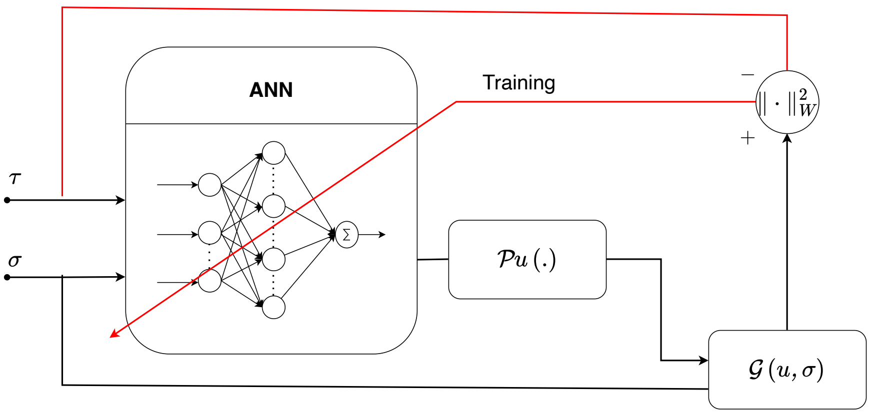

Figure 5.2: ANN Training Architecture

The output layer generates the control effector commands:

$$\mathbf{u} = \begin{bmatrix} \delta_a & \delta_e & \delta_r & \delta_c & \delta_s \end{bmatrix}^T \tag{5.6}$$

corresponding to aileron, elevator, rudder, and rotor disc tilt in longitudinal and lateral directions.

The network consists of multiple hidden layers with nonlinear activation functions (ReLU). The performance of the neural network depends on the number of hidden layers, number of nodes and comparison is given in the following sections. The final layer uses a linear output layer followed by a clipping operation that constrains the normalized actuator commands to the interval $[-1, 1]$. This ensures that the predicted actuator commands remain within the normalized limits used during training.

5.3.2 Training Data Generation

The training dataset is generated using a mid-fidelity non linear simulation of the gyrodyne model. Random combinations of control inputs and flight conditions are sampled within the allowable actuator and flight envelope limits using Latin Hypercube Sampling (LHS). The block diagram used to generate the training data is shown in the below figure.

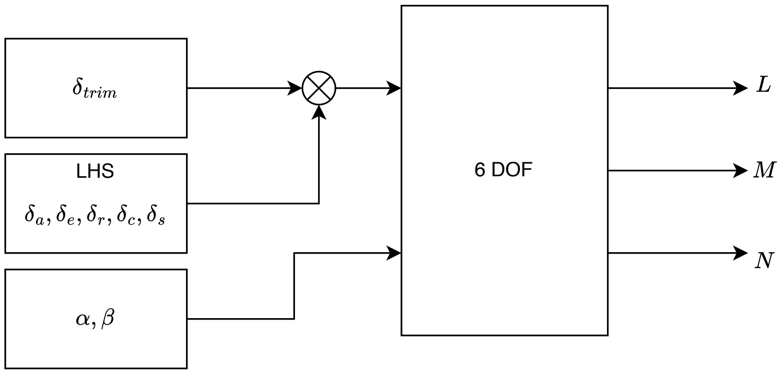

Figure 5.3: ANN Training Data Generation

For each sampled input, the non linear control effectiveness model is evaluated to obtain the resulting moment vector. This results in a dataset of the form:

$$(L, M, N, \alpha, \beta, q) \to \mathbf{u} \tag{5.7}$$

where the inclusion of dynamic pressure enables the network to capture variations in aerodynamic control authority resulting from changes in altitude and airspeed which is used to train the inverse mapping. It must be noted that, the data is obtained for different dynamic conditions i.e., at velocities of 28, 40, 50 m/s and altitudes of 0, 500, 1000, 1500, 2000 m. This gives a good range of dynamic pressure values where the neural network can learn the physics of the problem.

It must also be noted from the Figure 5.3 that LHS sampling is performed around a trimmed condition which gives a good starting point for neural network at different conditions rather than random effector position.

5.4 Assumptions and Scope of the Neural-Network Allocator

The proposed neural-network-based control allocation framework is developed under several assumptions intended to simplify the learning problem while remaining representative of the expected operating conditions of the gyrodyne.

1. The actuator dynamics are assumed to be sufficiently fast relative to the flight-control update rate. Consequently, actuator commands generated by the allocator are assumed to be achieved instantaneously, and actuator rate limits are neglected during network training.

2. The control effectiveness characteristics represented in the training dataset are assumed to accurately capture the nonlinear relationship between actuator deflections and the resulting moments. The allocator therefore relies on the fidelity of the underlying nonlinear aircraft model used for data generation.

3. The allocator is trained using data obtained within a predefined flight envelope consisting of representative combinations of airspeed, altitude, angle of attack, and sideslip angle. The network is therefore intended to operate within this envelope, and its performance outside the training domain is not explicitly guaranteed.

4. Dynamic pressure is assumed to adequately capture the primary variation in aerodynamic control effectiveness associated with changes in airspeed and altitude. Consequently, dynamic pressure is included as an input to the neural network rather than treating airspeed and altitude as independent variables.

5. The allocation problem is formulated using commanded body moments only. Force allocation is not considered in the present implementation, and thrust magnitude is assumed to be handled separately by the propulsion-control system.

6. Sensor measurements used by the allocator are assumed to be available without significant delay. Although robustness to measurement noise is evaluated through simulation, communication delays and estimator dynamics are not explicitly incorporated into the training process.

7. The allocator is developed as a static mapping between desired moments, flight-condition variables, and actuator commands. Temporal dependencies are therefore not modeled directly, and recurrent network architectures are not considered.

Under these assumptions, the neural network learns a nonlinear approximation of the inverse control-effectiveness map and can be implemented as a computationally efficient real-time control allocator.

5.5 Training Procedure

The proposed control allocation framework is trained in two stages. First, a forward neural-network model of the control effectiveness map is developed. Subsequently, this model is used to train the control allocator through a moment-tracking objective.

5.5.1 Forward Model Training

A forward neural network, referred to as the *plant model*, is first trained to approximate the nonlinear control effectiveness mapping of the gyrodyne. The network receives the actuator commands and flight condition variables as inputs and predicts the resulting body moments:

$$(\delta_a, \delta_e, \delta_r, \delta_c, \delta_s, \alpha, \beta, q) \to (L, M, N) \tag{5.8}$$

The training data generated in the previous section was used for this purpose. Prior to training, all input and output variables were normalized to the range $[-1, 1]$ to improve numerical stability and convergence. The network was trained using the Adam optimizer with a learning rate of 1×10^{-3} and a Mean Squared Error (MSE) loss function. The plant model was trained using two hidden layers with 128 neurons each and ReLU activation functions. The output layer contains three neurons corresponding to the roll, pitch, and yaw moments.

Once trained, the plant model provides a differentiable approximation of the nonlinear control effectiveness map and is subsequently frozen during allocator training.

5.5.2 Allocator Training

The control allocator network is trained to learn the inverse mapping between desired moments and actuator commands. Unlike conventional supervised learning approaches, the allocator is not trained directly using target actuator commands. Instead, the allocator is trained through the previously developed plant model.

The allocator receives the desired moment vector and flight condition variables as inputs:

$$(L_d, M_d, N_d, \alpha, \beta, q) \to (\delta_a, \delta_e, \delta_r, \delta_c, \delta_s) \tag{5.9}$$

The predicted actuator commands are passed through the frozen plant model to compute the moments generated by the corresponding control inputs. The training objective is then defined in terms of the difference between the desired and achieved moments:

$$J_{\text{track}} = \frac{1}{N} \sum_{i=1}^{N} \left[(L_i - L_{d,i})^2 + (M_i - M_{d,i})^2 + (N_i - N_{d,i})^2 \right] \tag{5.10}$$

To discourage excessive actuator activity, an additional control-effort penalty is incorporated into the loss function:

$$J_{\text{effort}} = \sum_{j=1}^{5} w_j \delta_j^2 \tag{5.11}$$

where w_j denotes the weighting associated with each actuator.

The total loss function is therefore expressed as:

$$J = J_{\text{track}} + \lambda J_{\text{effort}} \tag{5.12}$$

where λ is a tuning parameter controlling the trade-off between moment-tracking accuracy and actuator usage.

Training was performed using the Adam optimizer with a batch size of 128 samples. During each epoch, the allocator network generated actuator commands, the frozen plant model predicted the resulting moments, and the resulting tracking error was backpropagated through both networks to update only the allocator parameters. The final model was selected based on the lowest validation loss. The "Python" script used to train the model is given in Appendix B

Several network architectures and hyperparameter configurations were investigated, including different hidden-layer sizes, learning rates, and training-set sizes. Model performance was monitored using both training and validation losses throughout the learning process. The final model was selected based on the lowest validation loss, ensuring good generalization while avoiding overfitting.

5.5.3 Convergence Analysis

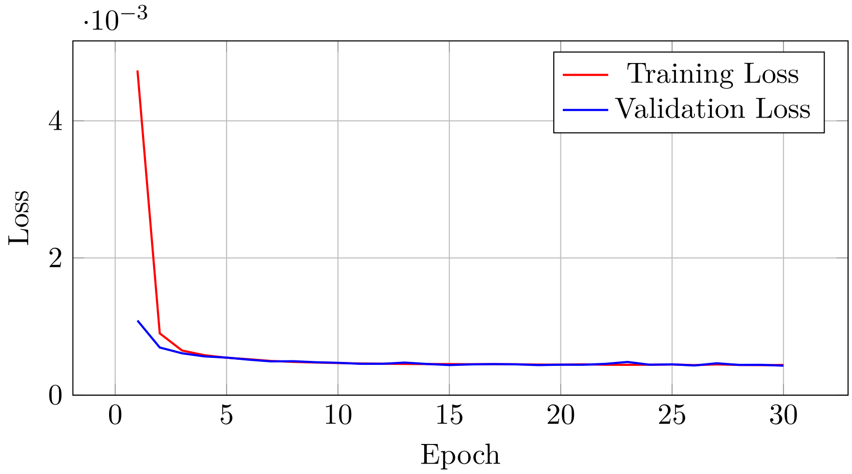

Figure 5.4: Learning Curve (Loss vs Epoch)

Figure 5.4 illustrates the learning curves of the final selected model. Both the training and validation losses decrease rapidly during the first few epochs and gradually converge to approximately 4.4×10^{-4}. The close agreement between the two curves indicates that the model generalizes well and does not exhibit significant over fitting. Furthermore, the stabilization of both losses after approximately 15 to 20 epochs suggests that the optimization process successfully reached a near optimal solution.

Additionally, the training history shows only minor fluctuations after convergence, with both losses remaining tightly bounded within a narrow interval around the final value. The absence of divergence between training and validation curves throughout the entire optimization process further confirms that the model is not over-parameterized relative to the dataset complexity. Small oscillations observed in later epochs are consistent with stochastic gradient effects rather than any systematic degradation in performance.

5.5.4 Effect of Training Dataset Size

To evaluate the influence of data availability on model performance, experiments were conducted using training subsets containing 10000, 20000, 40000, 60000 and 120000 samples. The Figures 5.5, 5.6 show both training and validation losses decrease consistently as the dataset size increases. This trend indicates that the model benefits from additional training samples and that further performance gains could potentially be achieved with larger datasets.

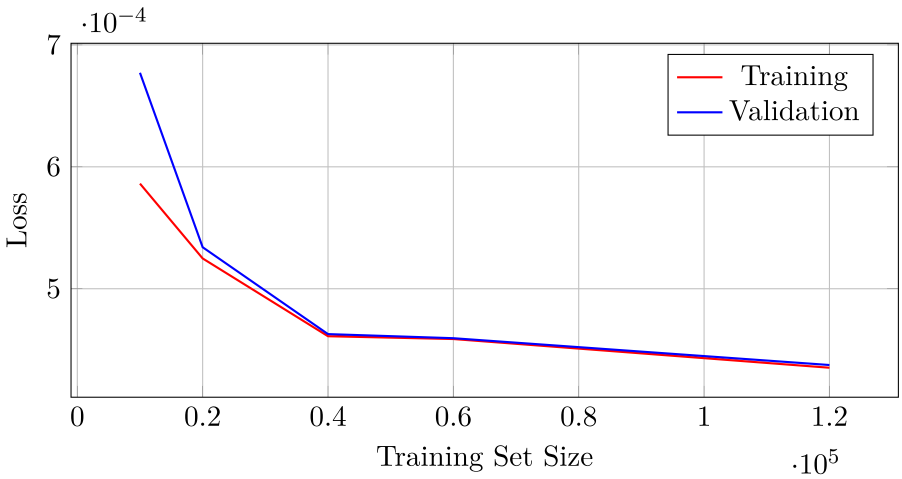

Figure 5.5: Effect of training dataset size on model performance

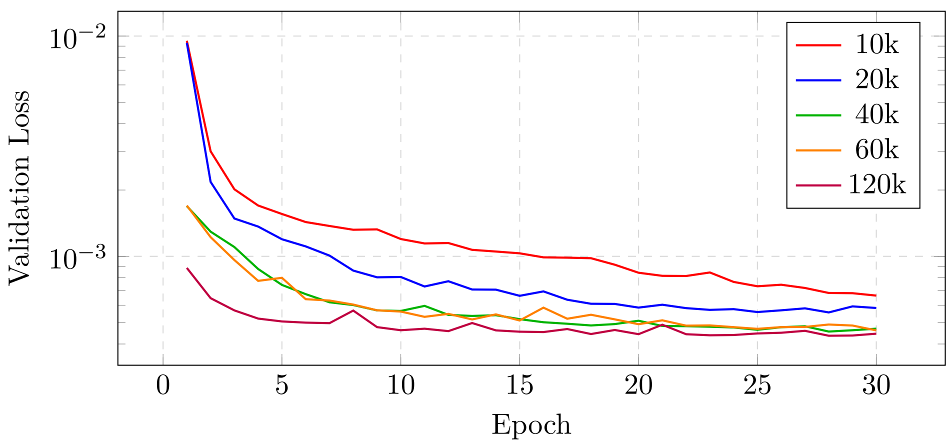

Figure 5.6: Validation loss vs epoch for different dataset sizes

The validation-loss curves for different dataset sizes further demonstrate that larger datasets produce faster convergence and lower final error values. In contrast, smaller datasets exhibit higher losses and greater fluctuations, suggesting increased sensitivity to sampling variability.

Furthermore, the relationship between dataset scale and convergence efficiency introduces a critical trade-off in the training pipeline. While the 10,000 sample configuration achieves its minimum error plateau within the first few epochs, its performance remains constrained by its higher error floor. Conversely, the 120,000 sample dataset requires a more sustained optimization path but continuously refines its weights to reach a significantly lower validation loss. This behaviour demonstrates that smaller data regimes suffer from an early premature convergence, where the model exhausts the information content of a limited sample pool well before optimizing its full structural capacity. Consequently, maximizing the data volume not only guarantees a superior generalization threshold but also ensures that the model successfully captures nuanced, long-tail patterns that are otherwise obscured by the sampling variability inherent in smaller subsets.

5.5.5 Effect of Learning Rate

To establish an optimal configuration for the optimization process, the hyper parameter space governing the model's update velocity was systematically investigated. Selecting a proper step size is critical, as it directly determines the stability of the gradient descent trajectory and dictates whether the optimizer can successfully locate the global minimum of the loss surface without overshooting or stagnating.

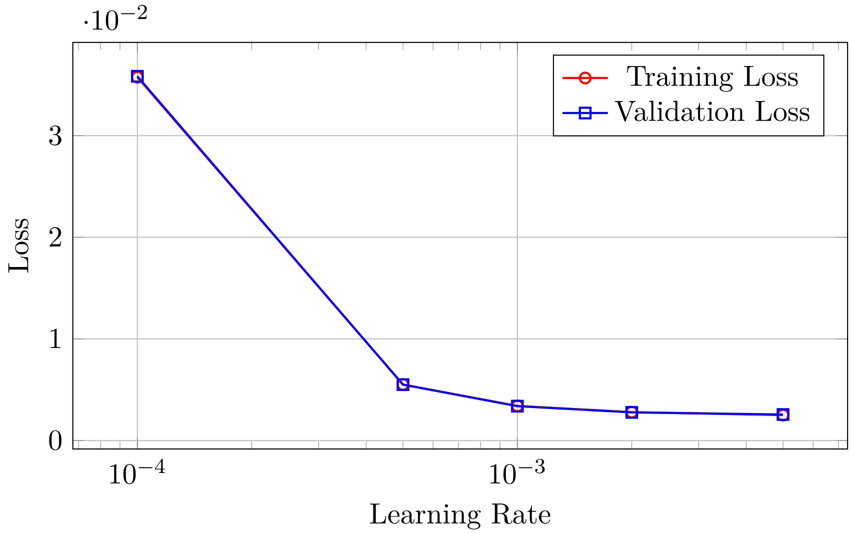

Figure 5.7: Effect of learning rate on model performance

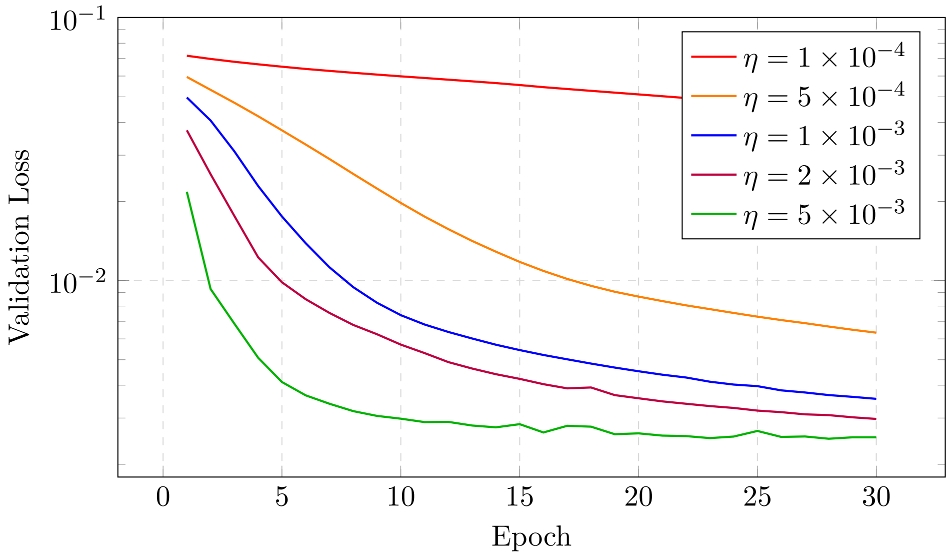

Figure 5.8: Validation loss vs epoch across different learning rates (η).

A learning rate sweep was performed to identify an appropriate optimization step size. Values ranging from 1×10^{-4} to 5×10^{-3} were evaluated. Figure 5.7 and 5.8 shows that very small learning rates resulted in slow convergence and relatively high final losses, while larger learning rates accelerated convergence substantially.

These experiments reveal that while aggressive step sizes risk destabilizing the optimization process, an properly tuned, moderately high learning rate strikes the ideal balance between convergence velocity and final error minimization. Based on these findings, a learning rate of 1×10^{-3} was selected as the baseline configuration for all subsequent experiments, ensuring both robust generalization and training efficiency.

5.5.6 Integration with Control System

The trained neural network is integrated into the flight control architecture as a direct replacement for the conventional allocation module. The desired force and moment commands generated by the attitude controller are directly mapped to actuator commands through a single forward pass of the network. This eliminates the need for solving an optimization problem at each control cycle while preserving the non linear coupling characteristics of the actuator system.

5.6 Integration with Flight Control System

The control allocation module is integrated within a hierarchical flight control architecture. The overall control system is structured such that high level trajectory commands are progressively converted into actuator commands through a sequence of nested control loops.

At the highest level, the trajectory controller generates reference commands for position and velocity tracking. These references are then processed by the attitude controller, which computes the required force and moment vector to achieve the desired rotational and translational motion of the gyrodyne. This results in a virtual control input in the form of desired moments and thrust-related forces.

The control allocation module operates at the lowest level of this hierarchy. Its function is to map the commanded force–moment vector into individual actuator commands while respecting actuator constraints and accounting for the coupled nature of the gyrodyne control effectors. In this work, the allocator is implemented either as a baseline pseudo-inverse method or as the proposed neural network based mapping.

Finally, the computed actuator commands are applied to the physical system through the aerodynamic control surfaces and rotor disc tilt mechanism, which generate the corresponding forces and moments acting on the vehicle. This hierarchical structure allows the control design to be decoupled into stabilization and allocation layers, ensuring modularity while enabling the use of redundant actuators for improved control authority and efficiency.

5.7 Performance Evaluation Metrics

The performance of the proposed neural-network-based control allocation scheme is evaluated by comparing it against the baseline control allocation method described previously. The comparison is performed using several quantitative metrics that assess both trajectory tracking performance and aircraft dynamic response.

5.7.1 Trajectory Tracking Error

The primary objective of the flight control system is accurate trajectory tracking. The tracking performance is evaluated using the Root Mean Square Error (RMSE) between the commanded and actual trajectory variables. For a generic state variable x, the RMSE is defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x_{ref,i})^2} \tag{5.13}$$

where x_i is the measured state, $x_{ref,i}$ is the corresponding reference value, and N is the total number of samples. The RMSE is computed for position and attitude variables to quantify the overall tracking accuracy of the flight control system.

5.7.2 Aerodynamic Response Metrics

Since the proposed control allocator modifies the distribution of control effort among the available actuators, additional performance indicators are required to evaluate the resulting aircraft response.

Particular attention is given to the angle of attack (α) and sideslip angle (β), since excessive values of these variables may increase aerodynamic drag, reduce efficiency, and potentially degrade flight handling qualities. The mean and peak values of α and β are therefore monitored throughout each simulation.

5.7.3 Robustness Assessment

To evaluate robustness, the control allocation methods are tested under multiple disturbance environments including atmospheric turbulence, sensor noise, and wind shear conditions. The resulting tracking errors and aerodynamic response metrics are compared to determine the sensitivity of each allocation strategy to external disturbances.

5.8 Results and Discussion

5.8.1 Case 1: Trajectory

The proposed neural-network based control allocator was first evaluated using a demanding trajectory consisting of climb, descent, and figure-eight maneuver segments as shown in the Figure 5.9. The objective of this test was to assess the allocator's ability to generate appropriate actuator commands during continuously varying flight conditions while maintaining trajectory tracking performance.

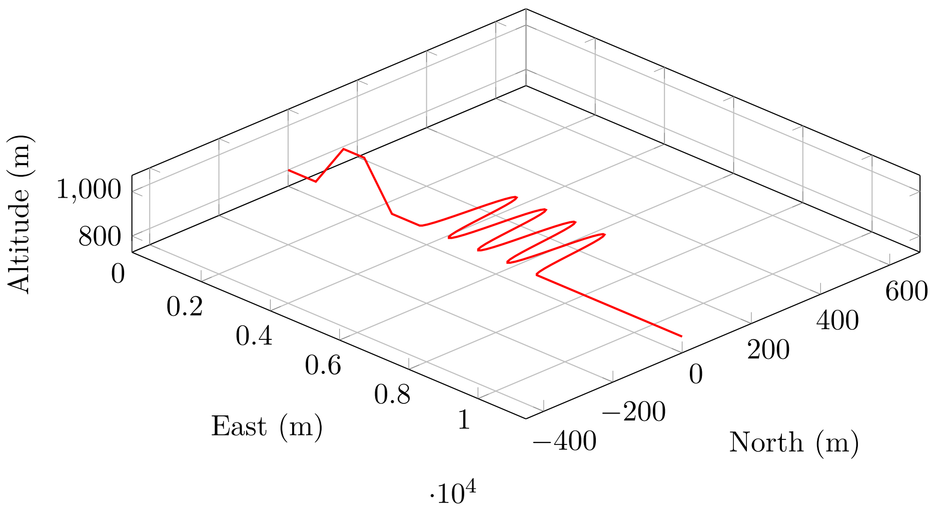

Figure 5.9: Test Trajectory

Attitude Tracking Performance

Figures 5.10(a) and 5.10(b) compare the Euler-angle and angular rate responses obtained using the baseline allocator and the proposed neural network based allocator. It can be observed that both approaches produce nearly identical attitude responses throughout the manoeuvre while trajectory tracking improves by approximately 5%. The roll, pitch, and yaw angles closely follow the commanded trajectories, with no noticeable degradation in tracking performance when the neural network allocator is employed.

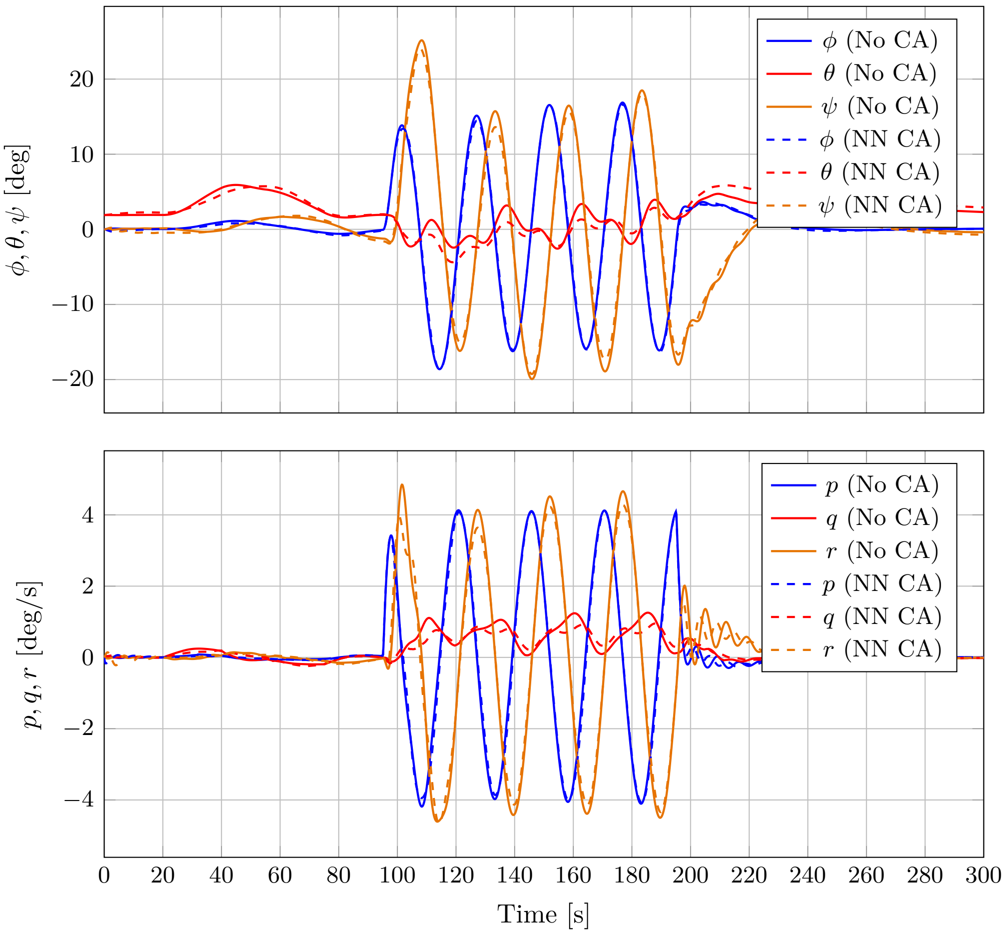

Figure 5.10: Comparison of Euler angles and angular rates with and without NN Control Allocation

A similar observation can be made from the angular-rate responses shown in Figure 5.10(b). The roll, pitch, and yaw rates generated by the two allocation methods exhibit very similar amplitudes and phase characteristics throughout the manoeuvre. This indicates that the neural network allocator is capable of generating the required control moments with a level of accuracy comparable to the baseline allocation strategy.

The corresponding RMSE values presented in Table 5.2 further confirm that the overall trajectory tracking performance of the two methods is nearly identical. Nevertheless, the neural network based allocator consistently exhibits a small improvement, achieving an average reduction in tracking error of approximately 6%. These results indicate that the proposed allocator preserves the stability and tracking performance of the flight control system while providing a modest enhancement in overall accuracy.

Aerodynamic Response

Although both allocation methods achieve comparable tracking performance, notable differences can be observed in the aerodynamic states of the aircraft.

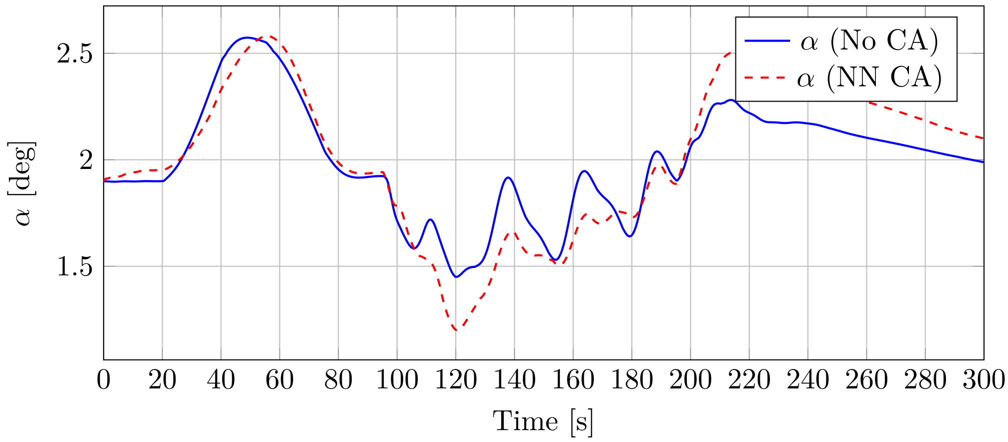

Figure 5.11: Comparison of angle of attack with and without NN Control Allocation with Test Trajectory

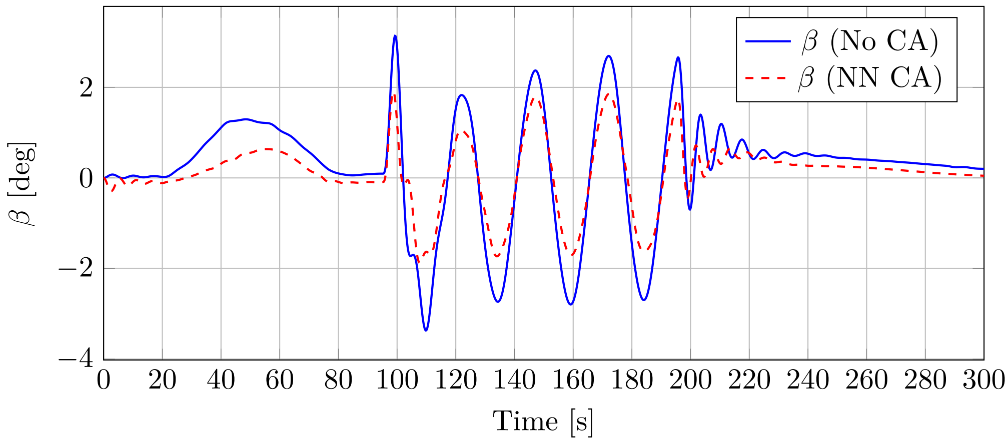

Figure 5.12: Comparison of Sideslip angle with and without NN Control Allocation with Test Trajectory

Figure 5.11 presents the angle-of-attack response during the manoeuvre. Both allocation methods maintain similar overall trends. However, the neural network allocator generally produces slightly lower values of angle of attack during the most demanding portions of the trajectory. The reduction is particularly visible during the oscillatory segment between approximately $100s$ and $200s$ simulation steps.

More significant differences are observed in the sideslip response shown in Figure 5.12. The neural-network allocator consistently reduces the magnitude of sideslip oscillations throughout the manoeuvre. Peak sideslip excursions are noticeably lower than those obtained using the baseline allocator, particularly during the aggressive turning segments where strong lateral-directional control inputs are required.

The reduction in sideslip angle suggests that the neural network allocator exploits the redundant control authority of the gyrodyne more effectively. By appropriately distributing control effort between the conventional aerodynamic surfaces and the rotor tilt actuators, the allocator is able to generate the required moments while maintaining improved lateral aerodynamic coordination.

Actuator Allocation and Control Effort

To reveal the control allocation mechanisms driving these aerodynamic state differences, the actuator deflection profiles during the climb, descent, and figure eight trajectory were analysed. Figure 5.13 compares the conventional control surface commands (aileron, elevator, and rudder) and the rotor cyclic inputs (lateral and longitudinal cyclic) generated by both strategies.

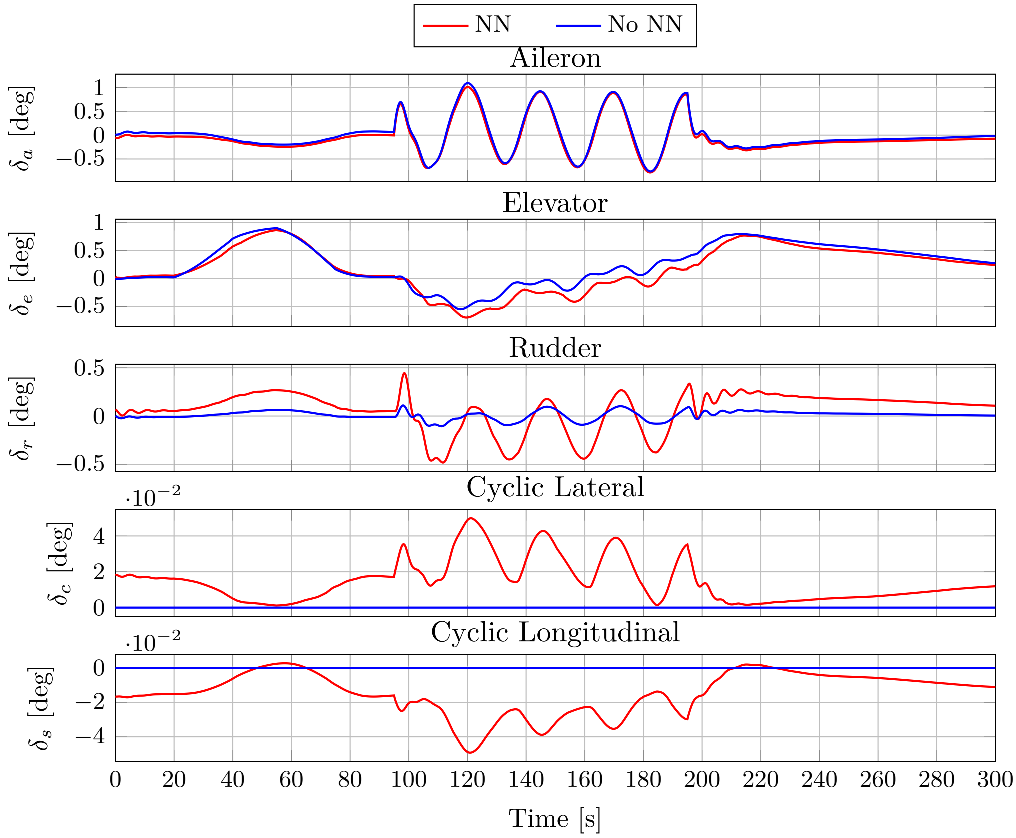

Figure 5.13: Control surface deflections comparison during test trajectory

As seen in figure 5.13, the baseline allocator indicated by the blue curves allocates control moments exclusively to the traditional aerodynamic surfaces, leaving both cyclic control channels unutilised. Conversely, the proposed neural network allocator dynamically coordinates all five control channels by utilizing the gyrodyne's inherent control redundancy.

In the longitudinal channel, both strategies demand highly similar aileron and elevator trends during the initial climb and descent phases 0 to 100s. However, during the highly dynamic figure eight segment, the neural network allocator commands a slightly deeper elevator deflection. This variation is directly complemented by the active utilization of the cyclic longitudinal control, which oscillates in phase between approximately 0 and 2 deg. This combined longitudinal authority explains why the neural network is able to track the primary attitude commands while establishing a marginally lower and more favourable angle of attack profile during the aggressive portions of the manoeuvre.

The most substantial differences manifest in the lateral-directional channel, matching the significant reductions in sideslip angle (β) previously highlighted. During the highly oscillatory figure-eight manoeuvre, the baseline allocator relies solely on the ailerons and small rudder deflections. The neural-network allocator, by contrast, radically alters the lateral distribution strategy. It drives the rudder with significantly larger, highly dynamic oscillations while concurrently modulating the cyclic lateral input.

By aggressively pairing the rudder with cyclic lateral movements during severe turns, the neural network optimizes lateral aerodynamic coordination. This multi channel distribution effectively suppresses the undesirable peak sideslip excursions and oscillations experienced by the baseline aircraft, demonstrating a highly effective exploitation of the redundant control authority.

5.8.2 Case 2: Climbing Turn

To evaluate the control allocator's performance under simultaneous longitudinal and lateral-directional demands, a climbing turn maneuver was simulated. The aircraft was commanded to perform a 90° heading change while concurrently climbing from an altitude of 1,000 m to 3,000 m. This profile tests the allocator's ability to handle severe cross-coupling effects and aerodynamic transitions.

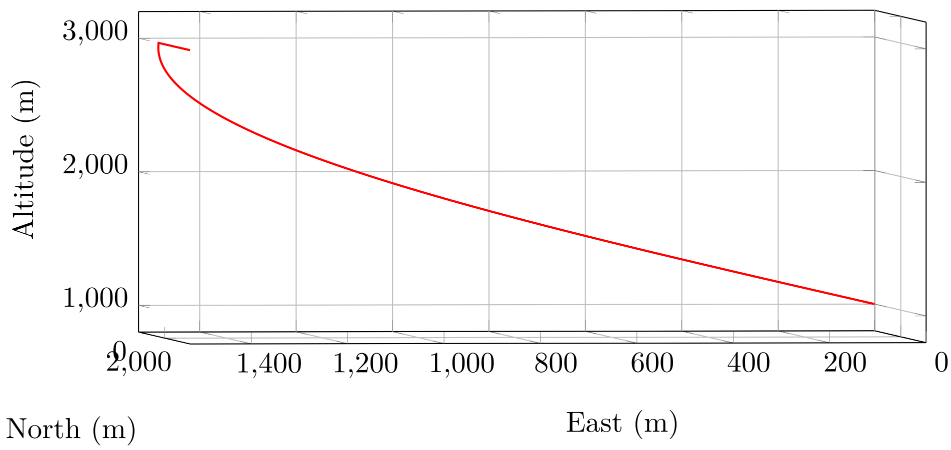

Figure 5.14: Climbing Turn Trajectory

Attitude Tracking Performance

Figures 5.15(a) and 5.15(b) display the attitude responses and angular rates during the climbing turn. As seen in Figure 5.15(a), the heading angle successfully executes the commanded 90° turn, exhibiting a transient overshoot up to approximately 120° before settling smoothly at the target heading.

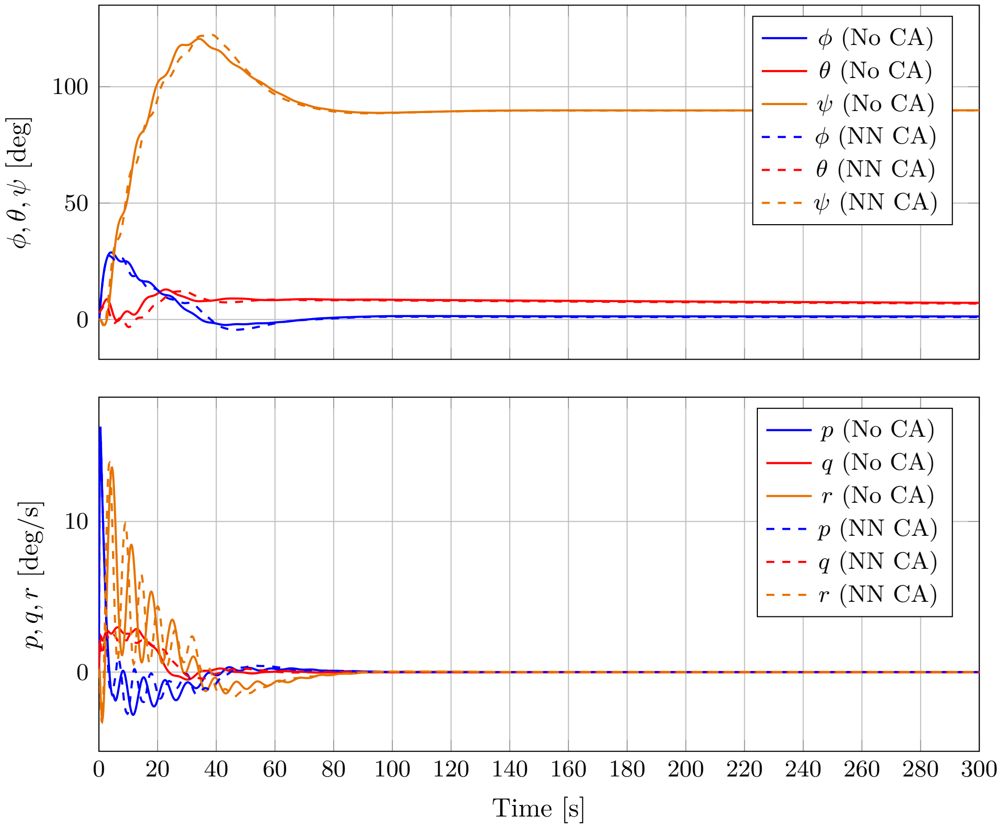

Figure 5.15: Comparison of Euler angles and angular rates with and without NN Control Allocation during Climbing Turn

Mirroring the results from the first trajectory, the baseline allocator and the proposed neural network based allocator yield virtually indistinguishable tracking profiles. The angular rates Figure 5.15(b) exhibit highly dynamic, high frequency oscillations during the initial aggressive phase of the manoeuvre 100s as the vehicle initiates the climbing turn. Both allocation strategies track these rapid rate variations with identical amplitude and phase matching, dampening completely to zero as the vehicle stabilizes in its steady state climb. This reinforces that the neural network introduces no lag or degradation in primary trajectory tracking loop authority.

Aerodynamic Response

While primary state tracking remains identical, the underlying aerodynamic state allocations reveal how the neural network allocator re optimizes control authority during a steady climb.

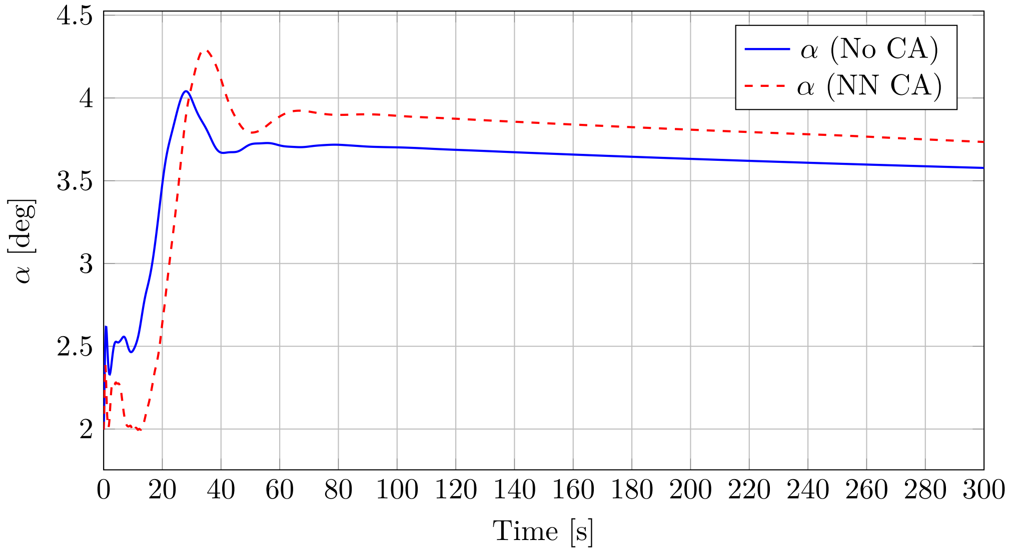

Figure 5.16: Comparison of angle of attack with and without NN Control Allocation with Climbing Turn

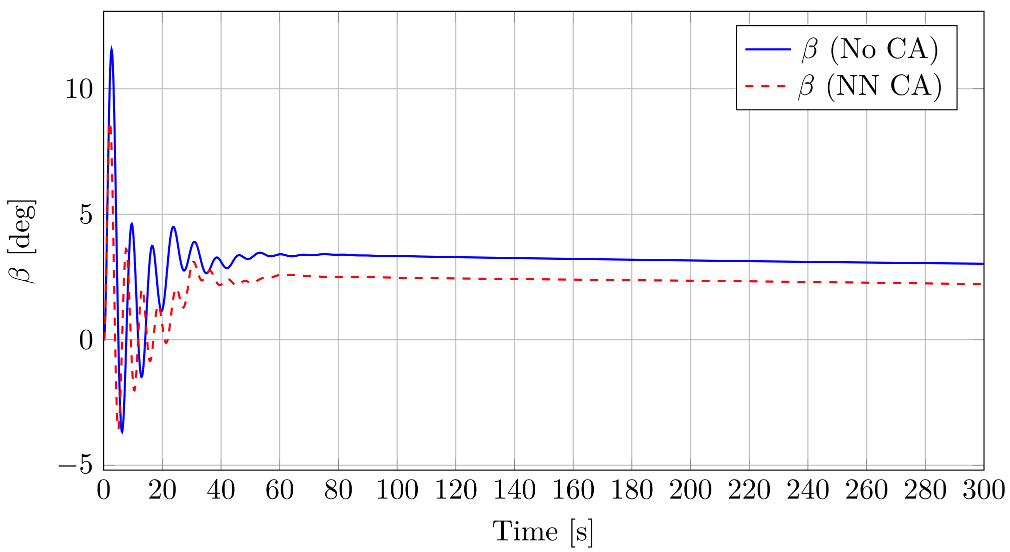

Figure 5.17: Comparison of Sideslip angle with and without NN Control Allocation with Climbing Turn

Figure 5.16 shows the angle of attack (α) behaviour. After an initial transient peak near 4.2° corresponding to the initial pull-up, both allocators settle into a steady climb trim configuration. Notably, the neural-network allocator manages to maintain a consistently higher angle of attack (approximately 0.2° higher) than the baseline allocator throughout the steady-state portion of the climb.

More pronounced differences show up in the side slip angle (β) response highlighted in Figure 5.17. The initial entry into the turn triggers severe lateral-directional oscillations ranging from $-4°$ to $+11°$. Both controllers damp these oscillations effectively by $50s$. However, they settle into fundamentally different trim states. The neural network allocator holds a distinct steady state sideslip angle offset relative to the baseline allocation method.

This variance demonstrates that the neural network has successfully discovered an alternative distribution of redundant control authority balancing the gyrodyne's rotor tilt actuators and standard aerodynamic surfaces differently than the baseline algorithm to satisfy the same moment demands during a steady climbing turn.

Actuator Use

To understand the underlying mechanism behind the variations observed in the aerodynamic states, the actuator deflection profiles during the climbing turn manoeuvre are evaluated. Figure 5.18 presents a comparative breakdown of the conventional aerodynamic surface deflections (aileron, elevator, and rudder) alongside the rotor cyclic controls (lateral and longitudinal cyclic).

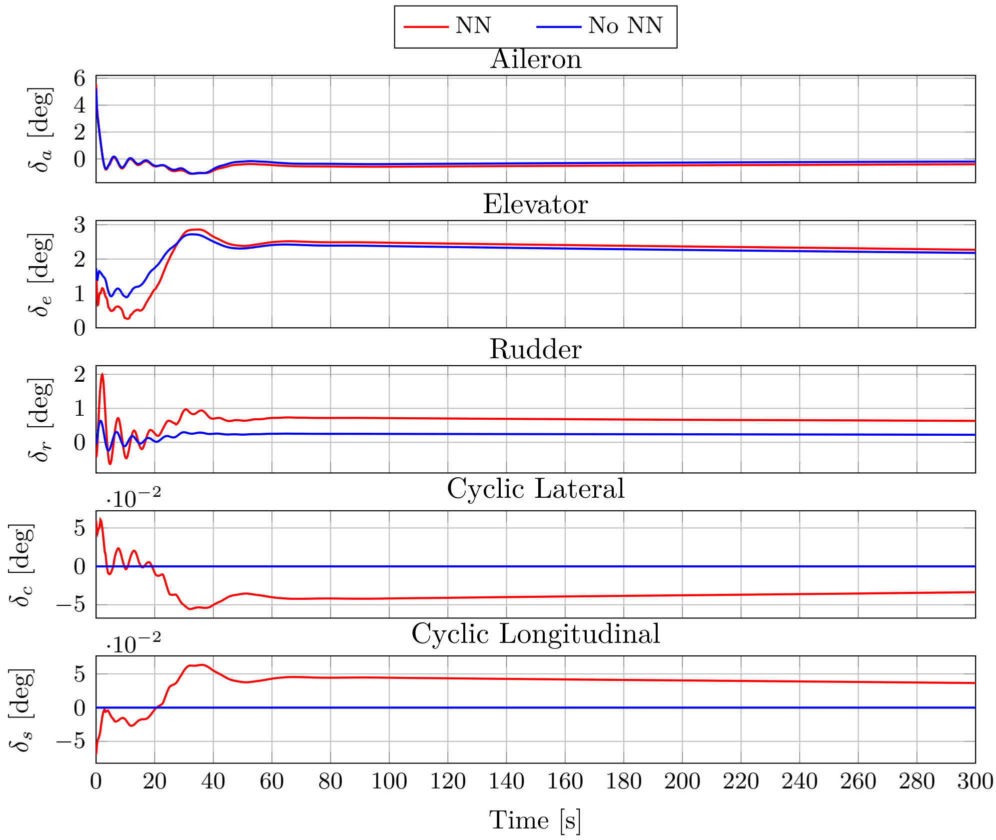

Figure 5.18: Control surface deflections comparison during Climbing Turn

The most prominent distinction between the two allocation strategies lies in the utilization of control redundancy. The baseline allocator (represented by the "No NN" trend) relies strictly on the conventional aerodynamic control surfaces, leaving the cyclic lateral and cyclic longitudinal inputs entirely unactuated. In contrast, the proposed neural network based allocator actively engages all five available control channels to distribute the moment generation effort.

In the longitudinal channel, both allocators command a similar transient profile for the elevator to initiate the climb, peaking near 3° before settling into a steady-state trim command. However, the neural network allocator maintains a marginally lower elevator deflection throughout the steady climb phase. This deficit is compensated for by the active engagement of the longitudinal cyclic input, which transitions to a steady state forward command of approximately 0.04 rad after 0.5×10^4 simulation steps.

More significant discrepancies are visible in the lateral-directional actuators, aligning with the pronounced differences previously observed in the sideslip angle response. During the highly dynamic entry phase (0 to $50s$), the baseline allocator induces localized, high-frequency oscillations in the rudder command. The neural-network allocator, conversely, distributes the lateral-directional requirements between the aerodynamic surfaces and the rotor. It commands a distinct steady state rudder offset (approximately 0.7°) while concurrently modulating the cyclic lateral input.

By offloading a portion of the control authority to the rotor cyclic dynamics, the neural network allocator re-balances the trim forces acting on the gyrodyne. This alternative distribution of control effort proves highly effective; it satisfies the identical primary attitude commands while establishing a more coordinated aerodynamic trim state during the steady climbing turn.

5.8.3 Robustness Evaluation: Turbulence, Noise, and Wind Shear

To rigorously assess the tracking capabilities and disturbance rejection of the proposed neural-network-based control allocator, the climbing turn manoeuvre was repeated under two severe environmental conditions:

1. **Measurement Noise and Atmospheric Turbulence:** Injected stochastic sensor noise alongside continuous background turbulence to evaluate high-frequency disturbance rejection.

2. **Wind Shear Profile:** Introduced a localized, sharp wind gradient during the steady climb phase to test the allocator's ability to handle sudden aerodynamic state transitions.

Trajectory Tracking Performance

Rather than presenting identical trajectory tracking plots for each environmental condition, the comparative performance is summarized using the Root Mean Square Error (RMSE) of the position tracking error, as reported in Table 5.2.

Scenario	No Neural Network	Neural Network	Improvement (%)
No Noise	94.80	88.70	6.5
Turbulence	102.80	95.50	7.1
Turbulence + Sensor Noise	113.50	118.50	5.1

Table 5.2: Trajectory tracking RMSE comparison

The results demonstrate that the proposed neural-network-based control allocator consistently improves trajectory-tracking performance across all tested environmental conditions. Under nominal conditions, the trajectory RMSE is reduced from 94.8 m to 88.7 m, corresponding to an improvement of approximately 6.5%. Similar improvements are observed in the presence of atmospheric disturbances, with the largest reduction occurring in the turbulence case, where the RMSE decreases from 102.8 m to 95.5 m (7.1%).

When both turbulence and sensor noise are present, the overall tracking error increases for both allocation strategies, as expected. Nevertheless, the neural network allocator continues to provide a measurable improvement, reducing the RMSE from 118.5 m to 113.5 m. This indicates that the allocator retains its effectiveness even when operating under degraded measurement quality and disturbed flight conditions.

The observed improvement can be attributed to the allocator's ability to exploit the additional control authority provided by the rotor tilt mechanism. By distributing the commanded moments more effectively among the available actuators, the neural network allocator reduces the need for large aerodynamic excursions and allows the vehicle to maintain closer adherence to the reference trajectory.

Overall, the results indicate that the proposed control allocation framework not only preserves the robustness of the closed loop flight control system but also provides a consistent reduction in trajectory-tracking error across a range of operating conditions.

Aerodynamic Stability and Actuator Adaptability

The primary divergence under these harsh environmental conditions again manifests in the lateral-directional aerodynamic coordination and underlying actuator allocation effort.

When encountering turbulence, the baseline allocator experiences sustained, high-frequency oscillations throughout the entire steady-state climb phase as it struggles to damp lateral disturbances using aerodynamic surfaces alone. Conversely, the neural-network allocator leverages the rotor's cyclic lateral dynamics to absorb a significant portion of the lateral energy.

When encountering the sudden wind shear gradient as highlighted in Figure 5.19, the neural network dynamically reallocates moment commands between the rudder and cyclic channels nearly instantaneously. While this results in a localized spike in actuator rate usage, it successfully bounds the peak sideslip excursions (β) and prevents the prolonged directional ringing observed in the baseline aircraft configuration.

This behaviour is particularly important during take-off and landing operations, where aircraft typically encounter elevated angles of attack (α) and sideslip angles (β). Simultaneously operating at high α and β can significantly degrade aerodynamic stability and, in extreme cases, lead to departure from controlled flight or the onset of spin. Consequently, maintaining low sideslip angles while achieving the required control objectives is desirable not only from a tracking performance perspective but also from a flight-safety standpoint.

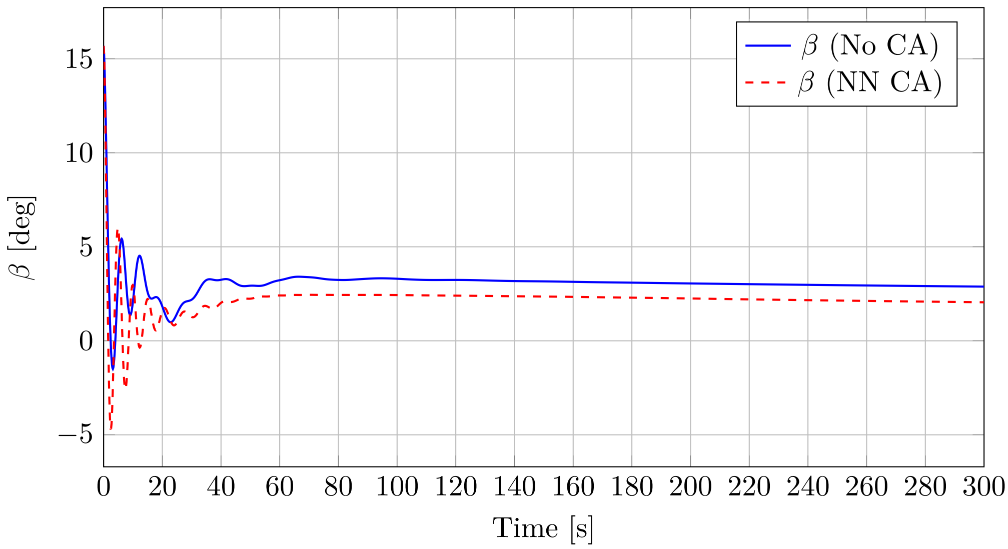

Figure 5.19: Comparison of Sideslip angle with and without NN Control Allocation with Climbing Turn with Wind Shear

Ultimately, these robustness tests demonstrate that the alternative control distribution strategy learned by the neural network is not merely an artifact of nominal trim, but a robust framework capable of maintaining superior aerodynamic coordination under realistic environmental uncertainties.

5.8.4 Control Authority Utilization

To further investigate the effect of the proposed control allocation strategy, the operating points corresponding to the different simulation scenarios were projected onto the control authority envelope shown in the below Figure 5.20.

The black boundary represents the attainable combinations of angle of attack and sideslip angle within the controllable flight region of the gyrodyne. The coloured markers indicate the average operating conditions obtained during the trajectory following simulations under nominal, turbulence, and wind-shear environments.

A consistent trend can be observed across all test cases. For each disturbance condition, the neural network based allocator shifts the operating point toward lower sideslip angles compared with the baseline allocation method. The reduction in sideslip is observed under nominal conditions, atmospheric turbulence, and wind shear disturbances, indicating that the behaviour is not limited to a particular flight scenario.

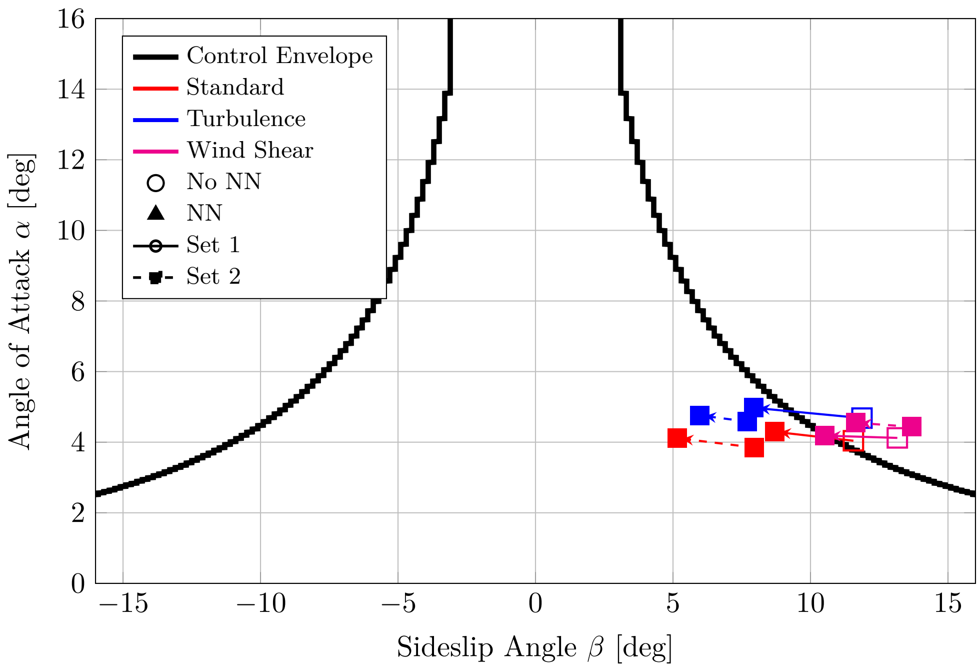

Figure 5.20: Control Authority Envelope with NN performance comparisons

In contrast, only minor variations in angle of attack are observed between the two allocation methods. The operating points remain clustered around $\alpha \approx 4°–5°$, suggesting that both allocators generate similar levels of aerodynamic loading while producing the required control moments.

The reduction in sideslip angle indicates that the neural-network allocator utilizes the available actuator redundancy more effectively. By distributing the required control effort between the aerodynamic control surfaces and the rotor tilt actuators, the allocator is able to generate the commanded moments while maintaining a more coordinated flight condition.

Furthermore, all operating points remain well within the attainable control-authority envelope, demonstrating that neither allocation method approaches the controllability limits of the vehicle during the tested manoeuvres. The proposed allocator therefore improves the aerodynamic operating condition without reducing the available control margin.

5.9 Limitations of Neural Network Allocation

Despite the promising performance demonstrated by the proposed neural network based allocator, several limitations must be acknowledged.

1. First, the allocator relies entirely on the quality and coverage of the training dataset. Since the network does not perform online optimization, its ability to generate appropriate actuator commands is limited to the flight conditions represented in the training data. Consequently, satisfactory performance requires the dataset to adequately cover the expected operating envelope of the gyrodyne. Performance outside the training region cannot be guaranteed.

2. Second, the control allocation process itself is open-loop. The neural network directly predicts actuator commands from the desired moments without explicitly evaluating the moments that are ultimately generated by the aircraft. Therefore, the allocator has no direct mechanism for correcting allocation errors arising from modelling inaccuracies, actuator non linearities, or unrepresented flight conditions. Although the overall flight control system remains closed-loop through the attitude and trajectory controllers, the allocation stage itself does not incorporate feedback of the achieved force and moment vector.

3. Finally, unlike optimization-based allocation methods, the proposed approach does not provide guarantees of optimality. The neural network approximates the inverse control effectiveness mapping learned from the training data and therefore generates feasible actuator commands rather than provably optimal ones. As a result, objectives such as minimum control effort, minimum actuator usage, or avoidance of actuator saturation cannot be explicitly enforced unless they are incorporated during the training process.

Future work may address these limitations through adaptive online learning techniques, closed-loop allocation architectures, or hybrid approaches that combine neural-network inference with optimization-based refinement.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

6.1.1 Summary of Modelling and Trim Analysis

This thesis presented the development of a nonlinear mathematical model of a gyrodyne aircraft and its implementation in a simulation based environment. The model incorporated the primary aerodynamic subsystems, including the main rotor, tail rotor, propeller, wing, and fuselage, together with environmental effects and rigid-body flight dynamics. A trim procedure was developed to obtain equilibrium conditions for hover and forward flight, while an autorotation analysis was performed to investigate the operating envelope of the unpowered rotor. Linearisation of the nonlinear model enabled the evaluation of the aircraft stability characteristics and provided the basis for subsequent flight-control design. Overall, the developed model demonstrated satisfactory agreement with available reference data and proved suitable for flight-dynamics and control-system investigations.

6.1.2 Evaluation of Flight Control

A hierarchical flight-control architecture was designed to provide stabilization and trajectory-tracking capability across the considered operating conditions. Classical control techniques were used to develop longitudinal, lateral-directional, and flight-path control loops. The resulting control system successfully stabilized the gyrodyne and achieved the desired handling-quality requirements. Closed-loop simulations demonstrated satisfactory tracking of commanded altitude, airspeed, attitude, and heading references. Furthermore, the control architecture provided a stable platform for evaluating different control-allocation strategies and assessing the benefits of actuator redundancy within the gyrodyne configuration.

6.1.3 Assessment of Neural Network Based Control Allocation

The primary contribution of this work was the development and evaluation of a neural network based control allocation framework for an over actuated gyrodyne aircraft. The neural network was trained offline to learn the inverse control-effectiveness mapping and subsequently integrated into the flight control system as a real time allocator. Simulation results showed that the proposed allocator achieved trajectory tracking performance comparable to the baseline allocation method, with no degradation in attitude response or closed loop stability.

Although both allocation methods produced nearly identical tracking accuracy, the neural network allocator demonstrated a more effective utilization of the available control authority. By coordinating conventional aerodynamic control surfaces with rotor tilt actuators, the allocator consistently reduced sideslip angle and improved aerodynamic coordination across multiple manoeuvres. These benefits remained evident under atmospheric turbulence, sensor noise, and wind shear disturbances, indicating good robustness to environmental uncertainties.

The results therefore demonstrate that neural-network-based control allocation represents a viable approach for exploiting actuator redundancy in advanced VTOL configurations. The proposed method preserves real-time computational efficiency while capturing the nonlinear characteristics of the control-effectiveness map, making it attractive for future unmanned gyrodyne applications.

6.2 Future Work

6.2.1 Aerodynamic Interaction and High-Fidelity Modelling

The aerodynamic model developed in this work relies on several simplifying assumptions intended to maintain computational efficiency. Future research should focus on improving model fidelity through the inclusion of rotor-wing, propeller-wing, and rotor-fuselage aerodynamic interactions. These effects become increasingly important during transition flight and aggressive manoeuvres. In addition, higher-fidelity approaches based on CFD or coupled aerodynamic models could be used to validate and refine the current simulation framework. Experimental validation through wind-tunnel or flight-test data would further increase confidence in the model predictions.

6.2.2 Advanced Non-linear Control Architectures

While the classical flight-control system provided satisfactory performance, future work may investigate advanced nonlinear control methodologies such as Model Predictive Control (MPC), Dynamic Inversion, Adaptive Control, or Backstepping Control. Such approaches may provide improved performance across a wider flight envelope and better accommodate the strongly nonlinear behaviour of gyrodyne dynamics. The authors of [24] showed that any robust controller may be used as long as they satify the Maximum Allocation Error (MAE). Furthermore, integration of the neural-network allocator with adaptive or learning-based controllers could enable more efficient exploitation of actuator redundancy under varying operating conditions.

6.2.3 Enhanced Control Allocation Strategies

The neural-network allocator developed in this thesis operates as an open-loop inverse mapping and depends strongly on the quality of the training dataset. Future research should investigate closed-loop allocation architectures that incorporate feedback of the achieved moments to compensate for modelling errors and actuator uncertainties. Additionally, online learning or adaptive neural-network techniques could be employed to extend performance beyond the training envelope. Hybrid approaches combining neural-network inference with optimization-based refinement may also provide guarantees on actuator usage, energy efficiency, and saturation avoidance while retaining real-time computational feasibility.

6.2.4 Real-Time Verification and Hardware Integration

A natural extension of this work is the implementation of the proposed flight-control and allocation framework in a real-time environment. Hardware-in-the-loop (HIL) and Software-in-the-loop (SIL) testing would enable assessment of computational requirements, actuator dynamics, communication delays, and sensor imperfections. Ultimately, experimental flight testing on a prototype gyrodyne platform would provide the final validation of both the aircraft model and the proposed neural-network-based control allocation methodology.

Bibliography

[1] Peter D. Talbot, Bruce E. Tinling, William A. Decker, and Robert T. N. Chen. A Mathematical Model of a Single Main Rotor Helicopter for Piloted Simulation. Technical report, NASA, 1982.

[2] Jr H. C. Curtiss. Rotor Contribution to Helicopter Stability Parameters. Technical report, Princeton University, 1963.

[3] Robert T. Chen. A Survey of Nonuniform Inflow Models for Rotorcraft Flight Dynamics and Control Applications. Technical report, NASA, 1989.

[4] J. Gordon. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2000.

[5] Bo Wang, Wei Zhang, Fei Wang, Qijun Zhao, Chenkai Cao, and Yuan Gao. Experimental investigation on aerodynamic and noise characteristics of Fenestron. *Chinese Journal of Aeronautics*, 36(12):88–101, 2023.

[6] Felix Frey. Aerodynamic Interactions on Airbus Helicopters' Compound Helicopter RACER. Technical report, University of Stuttgart, 2023.

[7] Antony Jameson. Analysis of Wing Slipstream Flow Interaction. Technical report, NASA, 08 1970.

[8] NOAA, NASA, and USAF. *U.S. Standard Atmosphere, 1976*. U.S. Government Printing Office, Washington, D.C., 1976.

[9] Alfred Gessow and Garry C Myers Jr. *Aerodynamics of the Helicopter*. Cambridge University Press, 1985.

[10] De La Cierva and Juan Don Rose. *Wings of Tomorrow: The Story of the Autogiro*. Brewer, Warren and Putnam, 1931.

[11] Hyeonsoo Yeo and Wayne Johnson. Optimum Design of a Compound Helicopter. *Journal of Aircraft - J Aircraft*, 46:19, 11 2006.

[12] Frederick D. Kim, Roberto Celi, and Mark B. Tischler. High-Order State Space Simulation Models of Helicopter Flight Mechanics. *American Helicopter Society*, 38(04), 09 1993.

[13] M. La Civita, G. Papageorgiou, W.C. Messner, and T Kanade. Integrated modeling and robust control for full-envelope flight of robotic helicopters. In

2003 IEEE International Conference on Robotics and Automation, volume 1, pages 552–557 vol.1, 2003.

[14] G. H. Bryan. *Stability in Aviation.* MACMILLAN and Co., Limited, 1911.

[15] Eugene L. Duke, Robert F. Antoniewicz, and Keith D. Krambeer. Derivation and Definition of a Linear Aircraft Model. Technical Report 1207, Ames Research Center, NASA, Edwards, California, 08 1988.

[16] A. R. S. Bramwell, George Done, and David Balmford. *Bramwell's Helicopter Dynamics.* Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford OX2 8DP, second edition, 2001.

[17] Donald McLean. *Automatic Flight Control Systems.* Prentice Hall International (UK) Ltd, 1990.

[18] SAE International. Design, Installation and Test of Piloted Military Aircraft, General Specification For, SAE Standard AS94900. Technical report, SAE International Technical Standard, Aerospace, 2007.

[19] United States Army. Aeronautical Design Standard Performance Specification Handling Qualities Requirements for Military Rotorcraft. Technical report, U.S. Army Aviation and Missile Command, 2011.

[20] David H. Klyde, Chase P. Schulze, Justin P. Miller, Jose A. Manriquez, Aditya Kotikalpudi, and David G. Mitchell. Defining Handling Qualities of Unmanned Aerial Systems: Phase II Final Report. Technical report, NASA, NASA Langley Research Center Hampton, Virginia, 02 2021.

[21] Vahram Stepanyan, Kalmanje Krishnakumar, Nhan Nguyen, and Luarens Van Eykeren. Stability and Performance Metrics for Adaptive Flight Control. Technical report, NASA Ames Research Center and Delft University of Technology, Moffett Field, CA 94035 and Delft, Netherlands, 2009.

[22] Teng-Tiow Tay, Iven Mareels, and John B. Moore. *High Performance Control.* Systems and Control: Foundations and Applications. Birkhäuser Boston, MA, first edition, 09 1998.

[23] Luke J. Miller and Jing Pei. A Comparison of Control Allocation Methods in the Presenceof Parametric Model Uncertainty. Technical report, NASA Langley Research Center, Hampton, VA, 2025.

[24] Hafiz Zeeshan Iqbal Khan, Surrayya Mobeen, Jahanzeb Rajput, and Jamshed Riaz. Nonlinear Control Allocation: A Learning Based Approach. arXiv, 2024.

Appendix

Appendix A

```
function [Omega, THETA0, Vz] = Autorotation( ...
    BLADES, ASLOPE, CHORD, CHORD75, ROTOR, a0, THETT, ...
    dyn_viscosity, k_ind, epsi, lever_arm_hub, chi, ...
    rho, Vcruise)
omega_TAS = [0 0 0]';
omega = [0 0 0]';
Weight = 9.81*150;
T_target = 0.10*Weight;
THETA0_range = deg2rad(0:0.1:5);
Vz_range = 0:0.1:5;
Omega_range = (100:10:200)*pi/30;
autorotation_data = [];
data = [];
dOmega = 0.5*pi/30;
for i = 1:length(THETA0_range)
    THETA0 = deg2rad(4) + THETA0_range(i);
    for j = 1:length(Vz_range)
        Vz = Vz_range(j);
        TAS = [Vcruise 0 Vz]';
        for k = 1:length(Omega_range)
            Omega = Omega_range(k);
            try
                [Frotor, Power, vi, Q, lambda] = ...
                    fcnRotor_new2( ...
                    BLADES, ASLOPE, CHORD, CHORD75, ...
                    ROTOR, a0, THETT, dyn_viscosity, ...
                    k_ind, epsi, lever_arm_hub, chi, ...
                    Omega, rho, TAS, omega_TAS, ...
                    omega, THETA0);
                [~,~,~,Qp,~] = ...
                    fcnRotor_new2( ...
                    BLADES, ASLOPE, CHORD, CHORD75, ...
                    ROTOR, a0, THETT, dyn_viscosity, ...
                    k_ind, epsi, lever_arm_hub, chi, ...
                    Omega + dOmega, rho, TAS, ...
                    omega_TAS, omega, THETA0);
                [~,~,~,Qm,~] = ...
                    fcnRotor_new2( ...
                    BLADES, ASLOPE, CHORD, CHORD75, ...
```

```
                    ROTOR, a0, THETT, dyn_viscosity, ...
                    k_ind, epsi, lever_arm_hub, chi, ...
                    Omega - dOmega, rho, TAS, ...
                    omega_TAS, omega, THETA0);
                Q_Omega = (Qp - Qm)/(2*dOmega);
                data(i,j,k).theta0 = THETA0;
                data(i,j,k).Vz = Vz;
                data(i,j,k).Omega = Omega;
                data(i,j,k).Power = Power;
                data(i,j,k).Thrust = Frotor(3);
                data(i,j,k).vi = vi;
                data(i,j,k).lambda = lambda;
                data(i,j,k).Q = Q;
                P_tol = 0.02 * Weight * ...
                    sqrt(abs(Weight)/(2*rho*pi*ROTOR^2));
                if abs(Power) < P_tol && ...
                   Frotor(3) < 0 && ...
                   vi < 0.2 && ...
                   Q_Omega < 0
                    autorotation_data = ...
                        [autorotation_data;
                         THETA0, Vz, Omega, Power, ...
                         Frotor(3), vi, lambda, Q];
                end
            catch
                data(i,j,k).Power = NaN;
                data(i,j,k).Thrust = NaN;
                data(i,j,k).vi = NaN;
                data(i,j,k).lambda = NaN;
                data(i,j,k).Q = NaN;
            end
        end
    end
end
cost = abs(autorotation_data(:,4)) + ...
       abs(autorotation_data(:,5) + T_target);
[~, opt_idx] = min(cost);
optimal = autorotation_data(opt_idx,:);
Omega = optimal(3);
THETA0 = optimal(1);
Vz = optimal(2);
end
```

Appendix B

```
import numpy as np
import pandas as pd
import tensorflow as tf
import scipy.io

from tensorflow.keras import layers, Model, Input # type:
    ignore
from tensorflow.keras.optimizers import Adam # type: ignore
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
# 1. LOAD DATA
X = pd.read_csv("X.csv").values
Y = pd.read_csv("Y.csv").values
print("Data shape:", X.shape, Y.shape)
# 2. NORMALIZATION
x_scaler = MinMaxScaler(feature_range=(-1, 1))
y_scaler = MinMaxScaler(feature_range=(-1, 1))
Xn = x_scaler.fit_transform(X)
Yn = y_scaler.fit_transform(Y)

# Split
X_train, X_val, Y_train, Y_val = train_test_split(
    Xn, Yn, test_size=0.2, random_state=1
)
# 3. TRAIN PLANT MODEL (Forward Model)
plant_in = Input(shape=(8,)) # Changed from 7 to 8
p = layers.Dense(128, activation='relu')(plant_in)
p = layers.Dense(128, activation='relu')(p)
plant_out = layers.Dense(3)(p)
plant_model = Model(plant_in, plant_out, name="Plant")
plant_model.compile(
    optimizer=Adam(learning_rate=1e-3),
    loss='mse'
)
plant_X = np.hstack([Yn, Xn[:, 3:]])
plant_Y = Xn[:, :3]
plant_model.fit(
    plant_X, plant_Y,
    epochs=40,
    batch_size=128,
    validation_split=0.2,
    verbose=1
)

# Freeze plant
plant_model.trainable = False
# 4. BUILD ALLOCATOR NETWORK
# Input: [L, M, N, alpha, beta, q]
# Output: normalized $\delta$ in [-1,1]

```

```
alloc_in = Input(shape=(6,))
a = layers.Dense(128, activation='relu')(alloc_in)
a = layers.Dense(32, activation='relu')(a)
a = layers.Dense(16, activation='relu')(a)
a = layers.Dense(8, activation='relu')(a)
alloc_out = layers.Dense(5)(a)
alloc_out = layers.Lambda(lambda x: tf.clip_by_value(x, -1.0, 1.0))(alloc_out)

allocator_model = Model(alloc_in, alloc_out, name="Allocator")
# 5. CONNECT (Allocator  to Plant)
class EffortPenalty(layers.Layer):
    def __init__(self, penalty_weight=0.01, **kwargs):
        super().__init__(**kwargs)
        self.penalty_weight = penalty_weight

    def call(self, delta):
        weights = tf.constant([0.2, 0.2, 0.2, 0.1, 0.1])  # [a, e, r, c, s]
        effort = tf.reduce_mean(weights * tf.square(delta))
        self.add_loss(self.penalty_weight * effort)
        return delta

# Get allocator prediction
raw_predicted_delta = allocator_model(alloc_in)

# Pass it through our custom layer to register the loss
predicted_delta = EffortPenalty(penalty_weight= 0.01)(raw_predicted_delta)

# Extract flight states: alpha, beta, and q from column index 3 to end
flight_states = alloc_in[:, 3:] # Grabs alpha, beta, q

# Combine for plant input matrix [predicted_delta (5) + flight_states (3)] = 8 columns
plant_input = layers.Concatenate(axis=1)([predicted_delta, flight_states])

# Predicted moments
achieved_moments = plant_model(plant_input)

# Full model
full_model = Model(alloc_in, achieved_moments)
# 6. LOSS FUNCTION
def weighted_mse(y_true, y_pred):
    weights = tf.constant([1.0, 1.0, 1.0])  # L, M, N
    return tf.reduce_mean(weights * tf.square(y_true - y_pred))

full_model.compile(
    optimizer=Adam(
        learning_rate=1.00e-3,
```

```
        beta_1=0.9,
        beta_2=0.999
    ),
    loss=weighted_mse
)
# 7. TRAIN ALLOCATOR
print("Training Allocator")
history = full_model.fit(
    X_train,
    X_train[:, :3], # target [L M N]
    epochs=30,
    batch_size=128,
    validation_data=(X_val, X_val[:, :3]),
    verbose=1
)
# 8. SAVE MODELS + WEIGHTS
allocator_model.save("allocator.keras")
weights = allocator_model.get_weights()
scipy.io.savemat("allocator_weights.mat", {"weights": weights})

# save scalers
 scipy.io.savemat(
     "scalers.mat",
     {
         "X_min": X.min(axis=0),
         "X_max": X.max(axis=0),
         "Y_min": Y.min(axis=0),
         "Y_max": Y.max(axis=0)
     }
 )

# 9. VALIDATION
# Allocator prediction across full dataset
delta_pred = allocator_model.predict(Xn, verbose=0)

# Pass through plant
plant_input = np.hstack([delta_pred, Xn[:, 3:]])
LMN_pred = plant_model.predict(plant_input, verbose=0)
LMN_true = Xn[:, :3]
error = LMN_pred - LMN_true
rmse = np.sqrt(np.mean(error**2, axis=0))
```