University of Cape Town

Faculty of Engineering and Built Environment

Department of Electrical Engineering

EEE3094S LAB 2 REPORT

KittiCopter Control System

Travimadox Webb

October 8, 2023

Abstract

This report provides a comprehensive analysis of the design and evaluation of a proportional controller tailored for altitude control in a Kitticopter system. Utilizing specific performance metrics, the controller's efficacy is rigorously assessed to ensure it meets the defined operational criteria.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

This lab focuses on designing a proportional controller for a "kitticopter" a simulated flying cat inspired by Orville, a feline that became a flying machine after its demise.

The lab is divided into two main sessions:

1. **System Identification**: The first stage is focused on developing a mathematical model for the Kitticopter. This model will encompass forces and moments acting on the system, such as rotor thrust, aerodynamic drag, and gravitational forces. Step tests and experimental validations are performed to fine-tune and identify any uncertain model parameters to allow for the design of the controller.

2. **Control Design**: The second stage revolves around developing a Proportional (P) controller to manipulate the Kitticopter's altitude. Various performance criteria are set to ensure precision, speed, and robustness in the control response.

1.2 Aim

This lab aims to design an na analog controller to control the altitude of the kitticopter that meets the following specifications:

1. Achieve greater than 90% accuracy in tracking positional inputs, allowing for a maximum tracking error and disturbance effects of less than 10%.

2. Reduce the system's settling time by at least 20% compared to the open-loop system.

3. Limit the overshoot to under 5% for any step response.

4. Maintain robustness against uncertainties of up to 10% in the aerodynamic constant, acknowledging the inevitable limitations in the experimental identification of this parameter.

5. Ensure a tolerance for 10% variations in the electronic components used in assembling the controller.

1.3 Objectives

To meet the aims mentioned above, the lab will focus on the following objectives:

- Formulate a comprehensive mathematical model to describe the Kitticopter's dynamics.
- Validate the proposed model through a rigorous system identification process.
- Design, construct, and validate a Proportional Controller that satisfies the specified criteria.
- Critically evaluate the controller's performance, specifically its ability to meet or exceed the set requirements.

1.4 Deliverables

At the end of this lab, the following deliverables are expected:

- A fully functional Proportional Controller, assembled on a Veroboard.
- A detailed report that outlines the methodology, design strategy, testing procedures, and results, as well as an evaluation of how well the final controller meets the established specifications.

Chapter 2

System Modelling

2.1 Introduction to System Modelling

In this chapter, the groundwork is laid for understanding the characteristics of our system. This will involve using fundamental physics and control theory principles to develop a mathematical model for system identification and control design.

2.2 Block Diagram Representation

A conceptual block diagram of the kitticopter control system is shown below in Figure 2.2 to facilitate an understanding of the system's architecture.

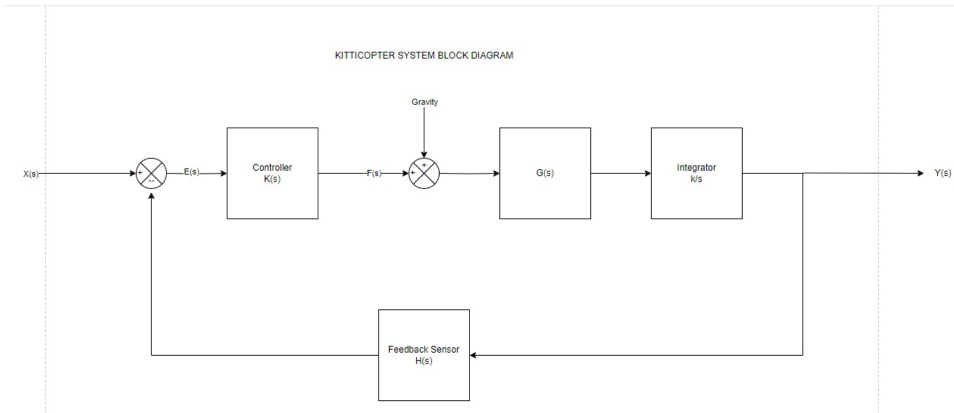

Figure 2.1: Block Diagram of Kitticopter System

2.3 Governing Differential Equation

2.3.1 Newton's Second Law Application

The free body diagram showing the forces acting on the system is shown in Figure 2.3.1 below:

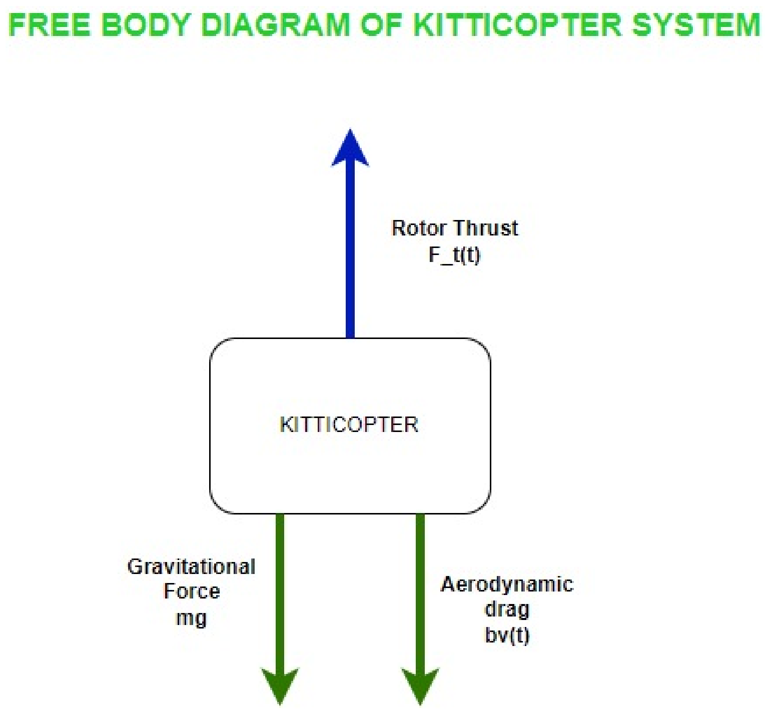

Figure 2.2: Free Body Diagram

The free body diagram shows the system dynamics through Newton's Second Law of Motion. For the kitticopter with mass m experiencing a thrust force $F_T(t)$, aerodynamic drag $by'(t)$, and gravitational force mg, the equation of motion is given by:

$$F = my''(t) \implies F_T(t) - by'(t) - mg = my''(t) \tag{2.1}$$

Where:

- $F_T(t)$: Thrust
- b: Aerodynamic Drag Coefficient
- g: Acceleration Due to Gravity
- m: Mass of the Kitticopter

2.4 Transfer Function Formulation

2.4.1 Relation Between Velocity and Thrust

The given differential equation emphasizes the velocity $v(t)$ as follows:

$$v'(t) = F_T(t) - bv(t) - g \tag{2.2}$$

To transition this equation into the frequency domain, apply the Laplace transform to both sides, yielding:

$$sV(s) - V(0) = F(s) - bV(s) - \frac{g}{s} \tag{2.3}$$

Assuming initial conditions are zero, the equation simplifies to:

$$sV(s) = F(s) - bV(s) - \frac{g}{s} \tag{2.4}$$

Rearranging the terms to isolate $V(s)$ gives:

$$sV(s) + bV(s) = F(s) - \frac{g}{s} \tag{2.5}$$

Now, factor out $V(s)$ from the left-hand side of the equation:

$$V(s)(s+b) = F(s) - \frac{g}{s} \tag{2.6}$$

Now, solve for $V(s)$:

$$V(s) = \frac{1}{s+b}F(s) - \frac{g}{s(s+b)} \tag{2.7}$$

The transfer function $G(s)$ of the system, which relates the input force $F(s)$ to the velocity $V(s)$, is thus:

$$G(s) = \frac{1}{s+b} \tag{2.8}$$

2.4.2 Closed-Loop Transfer Function Analysis

The open-loop transfer function is given by:

$$G(s) = \frac{AK}{s(s+b)} \tag{2.9}$$

In a standard closed-loop control system, the closed-loop transfer function $G_{cl}(s)$ is given by the formula:

$$G_{cl}(s) = \frac{G(s)}{1 + G(s)H(s)} \tag{2.10}$$

where H(s) is the transfer function of the feedback path. Let's assume that $H(s) = K_{sensor}$, which is the sensor coefficient. Hence, the expression for the closed-loop transfer function becomes:

$$G_{cl}(s) = \frac{\left(\frac{AK}{s(s+b)}\right)}{1 + \left(\frac{AK}{s(s+b)}\right)K_{sensor}} \tag{2.11}$$

Now, simplify the expression in the denominator:

$$1 + \left(\frac{AK}{s(s+b)}\right)K_{sensor} = \frac{s(s+b) + AKK_{sensor}}{s(s+b)} \tag{2.12}$$

Now, substitute this back into the expression for $G_{cl}(s)$:

$$G_{cl}(s) = \frac{\left(\frac{AK}{s(s+b)}\right)}{\left(\frac{s(s+b)+AKK_{sensor}}{s(s+b)}\right)} \tag{2.13}$$

Now, simplify the expression by multiplying the numerator and the denominator by $s(s+b)$:

$$G_{cl}(s) = \frac{AK}{s(s+b) + AKK_{sensor}} \tag{2.14}$$

2.5 Steady-State Errors and Performance Metrics

2.5.1 Position Error Relative to Setpoint

For a given reference input $X(s)$, the error $e(s)$ is expressed as:

$$e(s) = X(s) - \frac{KGHe(s)}{s} \implies E_{ss} = \frac{1}{1 + \frac{KH}{s^2+bs}} \tag{2.15}$$

Simplify the expression for steady-state error E_{ss}:

$$E_{ss} = \frac{s^2 + bs}{s^2 + bs + KH} \tag{2.16}$$

To find E_{ss} as s approaches zero, substitute $s = 0$ into equation 2.16.

$$\lim_{s\to 0} \left(\frac{s^2 + bs}{s^2 + bs + KH} \right) = \frac{0^2 + b(0)}{0^2 + b(0) + KH} = \frac{0}{KH} = 0 \tag{2.17}$$

Therefore, the steady-state error E_∞ is 0.

2.5.2 Impact of Gravity on Position Output

To analyze the effect of gravity, let $X(s) = 0$ and treat $V(s)$ as an input disturbance:

$$Y(s) = \frac{Q}{1 + QKH} V(s) \tag{2.18}$$

Substitute the known expressions:

$$G_{gravity}(s) = \frac{K}{s^2 + bs + KH} \tag{2.19}$$

To find the limit of $G_{gravity}(s)$ as s approaches zero, substitute $s = 0$ into the expression $G_{gravity}(s) = \frac{K}{s^2+bs+KH}$.

$$\lim_{s\to 0} \left(\frac{K}{s^2 + bs + KH} \right) = \frac{K}{KH} = \frac{1}{H} \tag{2.20}$$

Therefore, as s approaches zero, $G_{gravity}(s)$ approaches $\frac{1}{H}$.

Chapter 3

System Identification

3.1 Introduction

This chapter aims to identify the dynamic model of the Kitticopter system. An empirical method is used where an input is fed to the kitticopter system and characteristics of the output are extracted and used to formulate the transfer function.

Step inputs are fed to the system to generate step responses for parameter estimation. The key parameters like gain, time constant, and aerodynamic drag are extracted from the step response. These are then used to complete the model initially designed in chapter 2. The model is further validated to test its integrity and how close it is to the real experimental data.

3.2 Methodology

3.2.1 Step-Testing

Introduction to Step-Testing

Step-testing is crucial for acquiring a first-order understanding of the Kitticopter system. In this phase, we apply a step input of voltage and observe how the output, particularly the helicopter's velocity, responds over time.

Conditions for Step-Testing

1. **Known Step Size**: The step size from V_1 to V_2 must be known to normalize the measured output.

2. **Input Saturation Limit**: V_2 should not exceed 5V, as input saturation occurs above this level, leading to discrepancies between what the ADC reads and what the Kitticopter actually receives.

Circuit Configuration for Step-Testing Methodology

For our step-testing experiments, we employed a straightforward yet effective circuit setup that consists of a 10k ohm potentiometer coupled with a 10k ohm resistor. The detailed layout of the circuit can be seen in Figure 3.1 below:

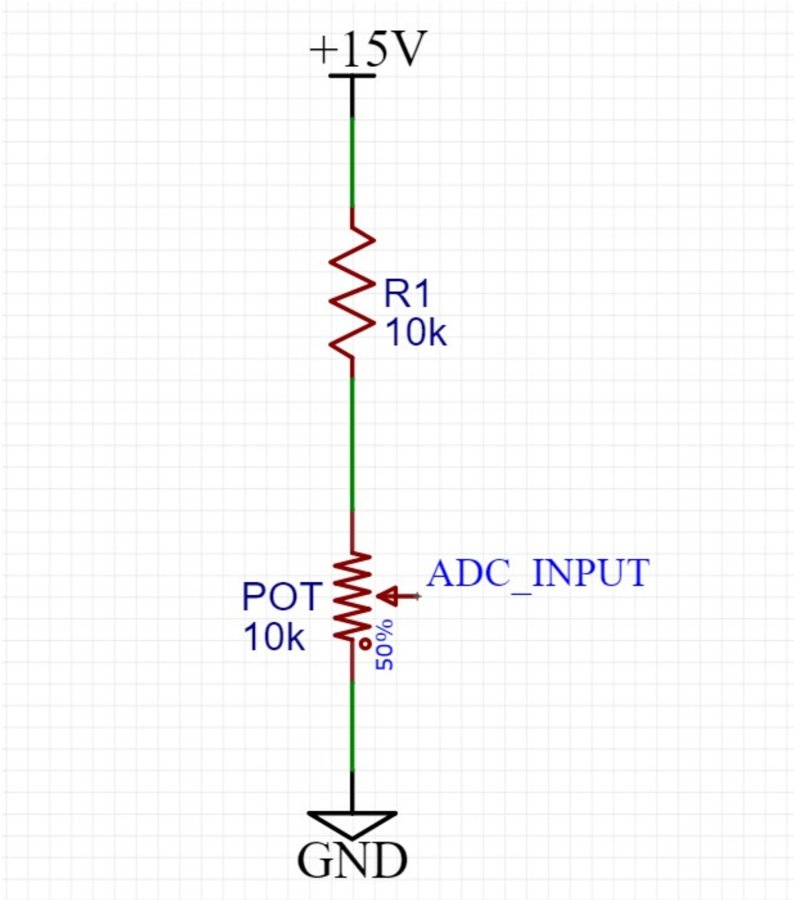

Figure 3.1: Circuit Configuration for Step-Testing Experiments

This circuit is interfaced with the system's ADC (Analog-to-Digital Converter). A baseline voltage, denoted as V_1, is initially established at 2.5V. To initiate the step change, the potentiometer is adjusted to its maximum position, effectively changing the voltage to $V_2 = 5V$.

To account for potential inconsistencies or anomalies in the system's response, this step-testing procedure is conducted five times. This approach ensures that the collected data is reliable.

Data Collection and Analysis

Data from the step-test is collected and stored in CSV files. The data is then analyzed in a Python Jupyter Notebook[1], focusing on:

- Reading Kitticopter Data
- Calculating Velocity
- Data Plotting
- System Identification through Key Parameters such as:
  - Time constant (τ)

[1] Jupyter Notebook

  - Final Velocity
  - Gain (A)
  - Aerodynamic Drag Coefficient (b)
  - Scalar Sensor Coefficient
- Transfer Function Approximation
- Simulated Step Response Validation

Data Preprocessing

Velocity Calculation Understanding the role of velocity is crucial in the system identification process. As established in Chapter 2, the relationship between rotor thrust force and velocity can be modeled as a first-order system. This simplification eases the complexity of the modeling exercise, facilitating a more straightforward identification process.

In order to compute the velocity, we used the formula for the rate of change of position with respect to time. Mathematically, this is expressed as:

$$\text{Velocity} = \frac{\Delta\text{Position}}{\Delta\text{Time}} \tag{3.1}$$

Here, ΔPosition refers to the change in altitude of the Kitticopter, and ΔTime represents the elapsed time during which this positional change occurs.

3.3 Results and Analysis

3.3.1 Step Response Analysis

The step responses for the Kitticopter system were captured during five distinct trials. Each figure 3.2 - 3.6 below showcases a step response, allowing for detailed analysis and comparison.

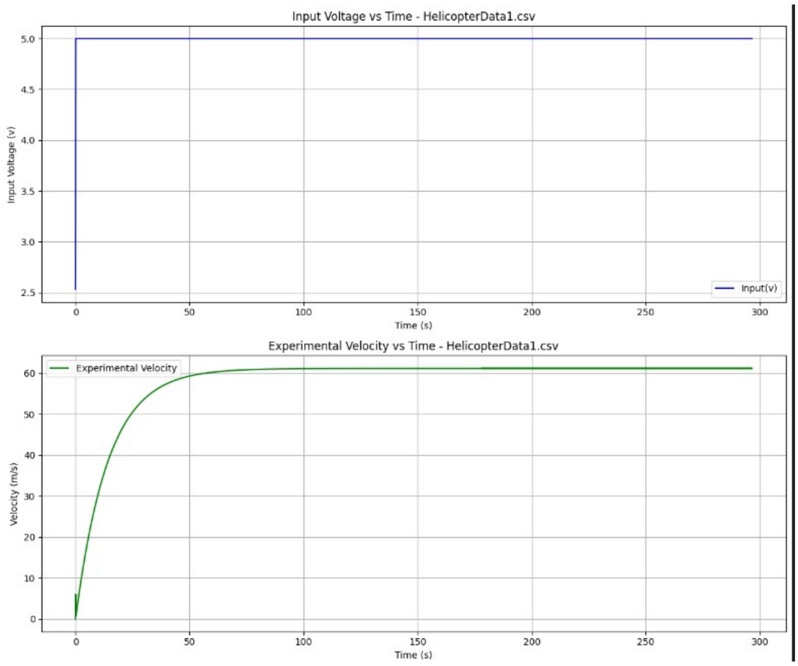

Figure 3.2: Step Response for Trial 1

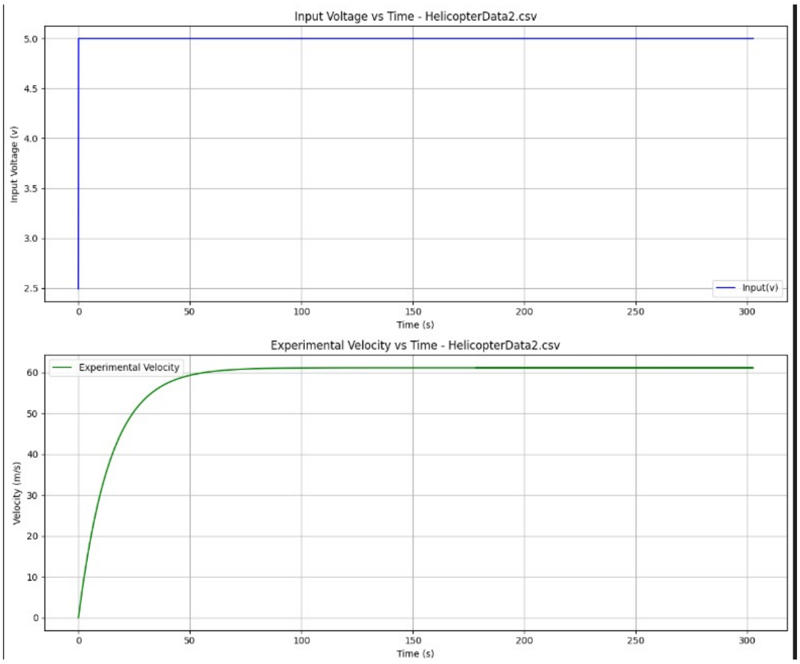

Figure 3.3: Step Response for Trial 2

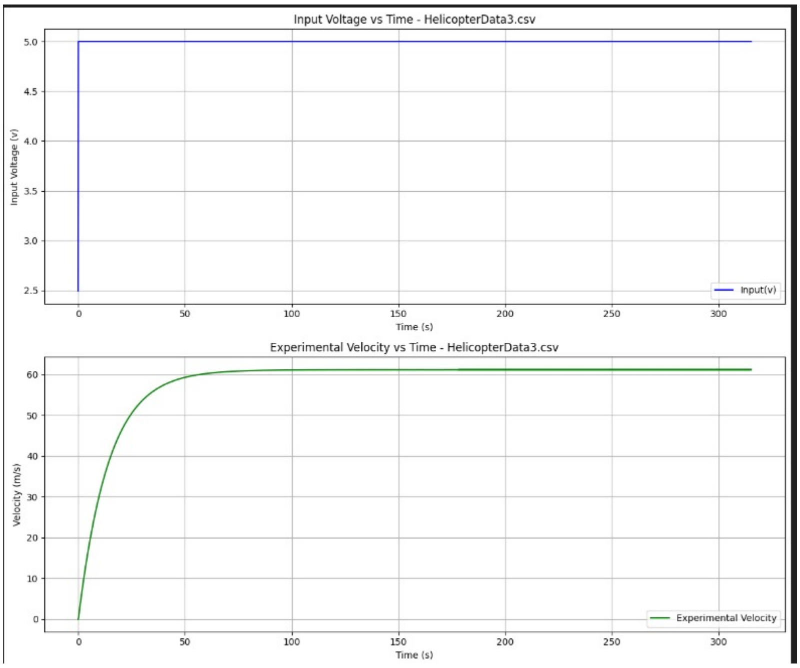

Figure 3.4: Step Response for Trial 3

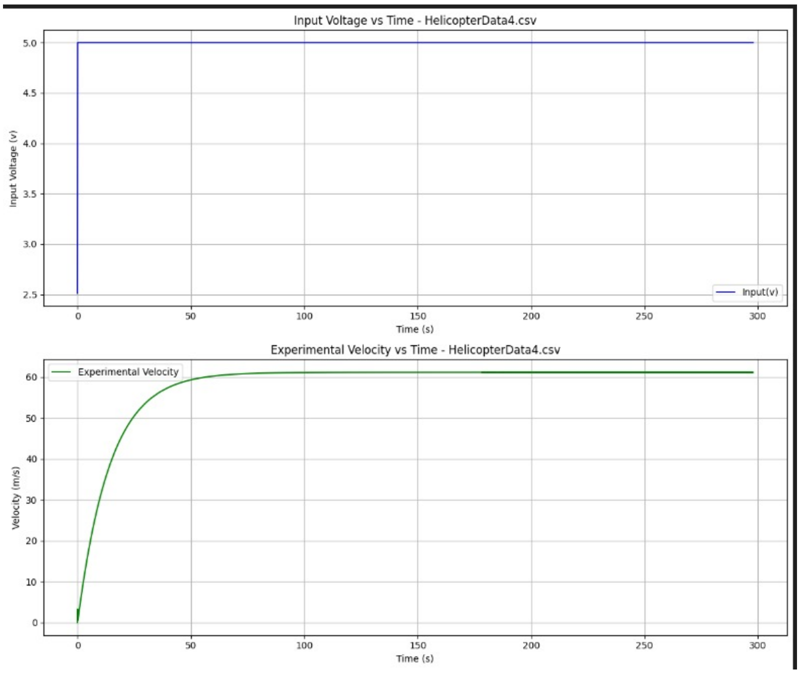

Figure 3.5: Step Response for Trial 4

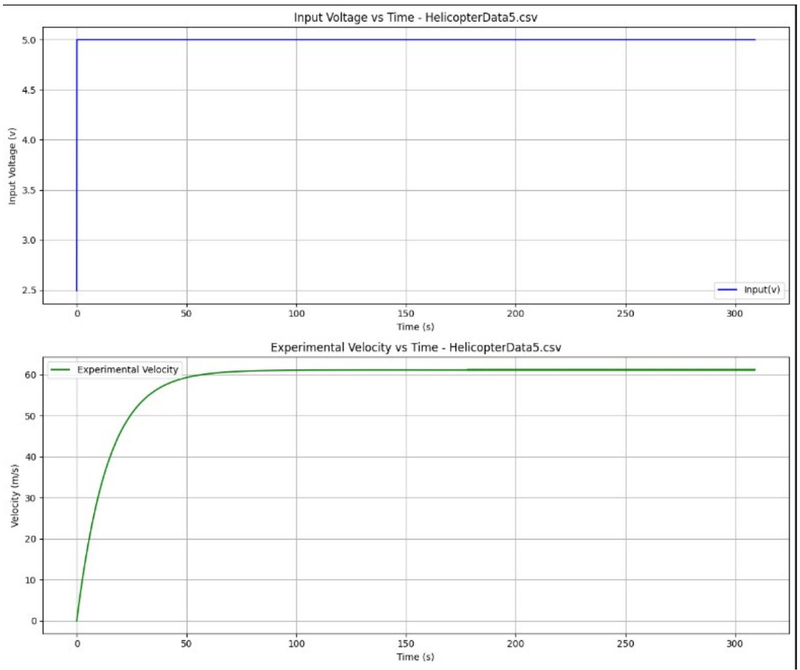

Figure 3.6: Step Response for Trial 5

Identification of Key Parameters

Gain (A) The system gain was calculated using the following equation:

$$A = \frac{\text{Final Value} - \text{Initial Value}}{\text{Amplitude of Step Input}} \tag{3.2}$$

For each of the five data sets, gains were calculated as follows:

Table 3.1: Calculated Gains from Step-Testing Experiments

Experiment	Gain (m/s)
Trial 1	24.756888168554266
Trial 2	24.371759074587924
Trial 3	24.4116473873059
Trial 4	24.55787781351554
Trial 5	24.4116473873059
Average	24.501963966253903

The average gain of the system was then obtained:

$$A_{\text{avg}} = \frac{\Sigma A_i}{N} = 24.501963966253903 m/s \tag{3.3}$$

Where A_i are the individual gains and N is the number of data sets (5 in this case).

Time Constant (τ) The time constant τ was calculated based on the following criterion:

$$\tau = \text{Time required to reach 63.2\% of the final steady-state value} \tag{3.4}$$

Five separate τ values were calculated as follows:

Table 3.2: Calculated Time Constants from Step-Testing Experiments

Experiment	Tau
Trial 1	14.3
Trial 2	14.3
Trial 3	14.3
Trial 4	14.3
Trial 5	14.3
Average	14.3

The average time constant was then calculated:

$$\tau_{\text{avg}} = \frac{\Sigma \tau_i}{N} = 14.3s \tag{3.5}$$

3.3.2 Transfer Function Approximation

Utilising the averaged values of the identified parameters A_{avg} and τ_{avg}, the system's transfer function was approximated:

$$G(s) = \frac{A_{\text{avg}}}{\tau_{\text{avg}} s + 1} \tag{3.6}$$

$$G(s) = \frac{24.50}{14.3s + 1} \tag{3.7}$$

3.3.3 Validation and Verification

Simulated Step Response Analysis

To validate the derived transfer function, a simulated step response was generated using a step input of magnitude 2.5. This simulation was carried out to closely mirror the five experimental trials previously conducted. Figures 3.7-3.11 depict the comparison between the simulated step responses and those obtained through experimentation.

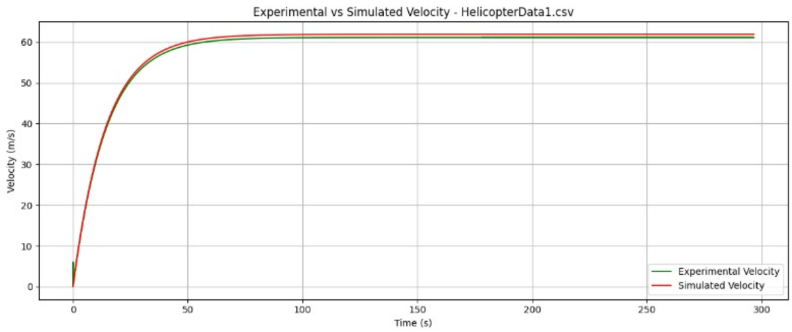

Figure 3.7: System Model Validation 1

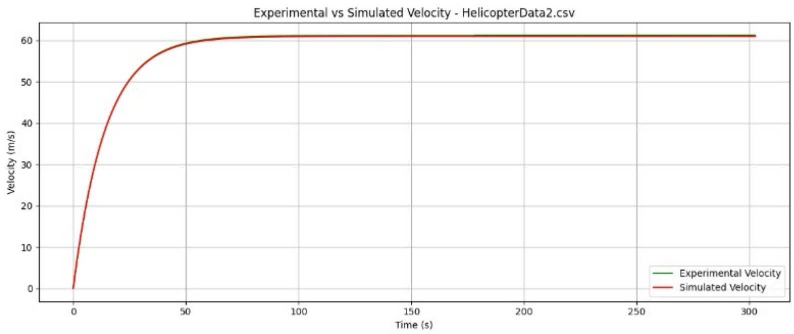

Figure 3.8: System Model Validation 2

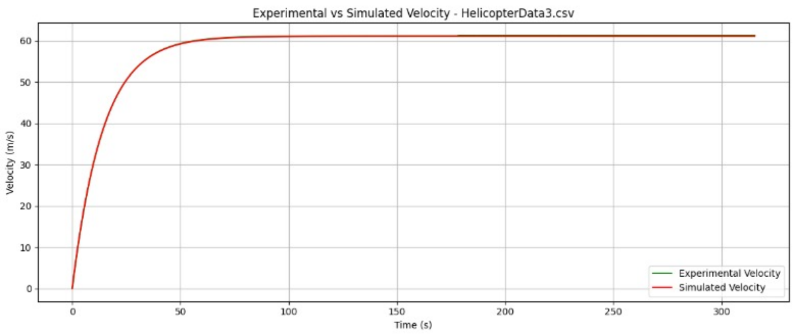

Figure 3.9: System Model Validation 3

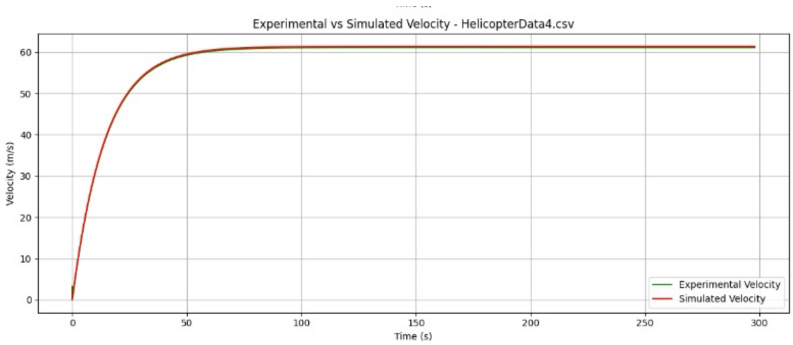

Figure 3.10: System Model Validation 4

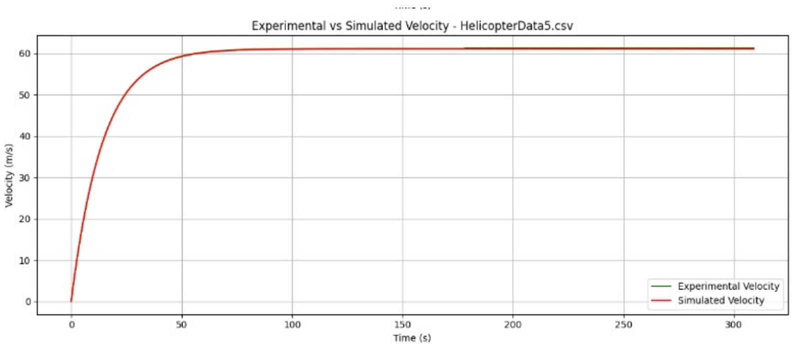

Figure 3.11: System Model Validation 5

From the comparison, it is evident that the simulated responses closely align with the experimental data. This reinforces the robustness of our Kitticopter system model.

Sensor Coefficient Estimation

The altitude-to-voltage sensor coefficient was estimated using a linear model. The sensor coefficient values from the five experimental trials are tabulated below:

Table 3.3: Estimated Sensor Coefficients from Experiments

Experiment	Coefficient
Trial 1	0.569995446294936
Trial 2	0.5699821447253482
Trial 3	0.5699985513224997
Trial 4	0.5699915412190348
Trial 5	0.5699920174992148
Average	0.5699919402122069

The average sensor coefficient was then calculated:

$$H_{\text{avg}} = \frac{\Sigma H_i}{N} = 0.5700 \tag{3.8}$$

3.4 Discussion

Upon comparing the simulated and experimental results, we notice a high degree of concordance, affirming the robustness and reliability of our system model. The average values of key parameters, including Gain A, Time Constant τ, and Aerodynamic Drag, are consistent across both simulated and experimental tests.

3.5 Conclusion

The model that has successfully validated will serve as the foundation for controller design in the next chapter.

Chapter 4

Controller Design

4.1 Introduction

This chapter outlines the systematic design of a Proportional (P) controller aimed at satisfying the specified performance requirements. The controller is designed to achieve over 90% accuracy in position tracking, a 20% improvement in settling time, a maximum of 5% overshoot, and robustness against system uncertainties and component tolerances. MATLAB's PID Tuner tool was used for fine-tuning the controller parameters, and the tuned controller was exported to Simulink for simulation with the KittiCopter system.

4.2 Theoretical Design

This section establishes the basic principles, equations, and derivations crucial for crafting a solid control strategy. Key calculations, including those related to the damping ratio and settling time, are conducted with precision to foresee the system's behavior. As this section concludes, a variety of theoretical plots such as the root locus and Bode plots are illustrated, providing a thorough schematic of the control system from a theoretical perspective.

4.2.1 Achieving Over 90% Accuracy in Position Tracking

Analysis of Steady-State Error

To meet the specified tracking accuracy of over 90%, we need to carefully examine the steady-state error (e_∞) for a step input to the system. The steady-state error is calculated using the following formula:

$$e_\infty = \lim_{s \to 0} s \times \frac{1}{s} \times E(s) = \lim_{s \to 0} E(s) \tag{4.1}$$

As per Equation 2.16 in the Chapter:

$$E(s) = \frac{s^2 + bs}{s^2 + bs + KH} \tag{4.2}$$

Upon evaluating the limit as $s \to 0$:

$$e_\infty = \lim_{s \to 0} \frac{s^2 + bs}{s^2 + bs + KH} = 0 \tag{4.3}$$

The conclusion from this analysis is that the steady-state error of the system is zero.

Implications for Controller Gain The finding that the steady-state error is zero implies that any finite control gain (K) would suffice for achieving the tracking accuracy requirement, provided that the controller also meets other performance criteria. This makes it clear that the system can achieve more than 90% accuracy in position tracking as long as the other performance specifications are also satisfied.

4.2.2 Achieving a 20% Reduction in Settling Time

Settling Time of the Open-Loop System The settling time (T_s) for a first-order system is often defined as the time required for the system to reach and remain within 98% of its final value. For a first-order system with a time constant τ, the settling time is approximately 4τ. Given that $\tau = 14.3$ seconds in our case, the settling time for the open-loop system ($T_{s,\text{open-loop}}$) can be calculated as:

$$T_{s,\text{open-loop}} = 4 \times \tau = 4 \times 14.3 = 57.2 \text{ s} \tag{4.4}$$

Target for 20% Improvement in Settling Time To achieve a 20% improvement in settling time, the closed-loop system should have a settling time ($T_{s,\text{closed-loop}}$) that is at most 80% of the open-loop system's settling time:

$$T_{s,\text{closed-loop}} \leq 0.8 \times T_{s,\text{open-loop}} \tag{4.5}$$

Substituting the known value, this yields:

$$T_{s,\text{closed-loop}} \leq 0.8 \times 57.2 = 45.76 \text{ s} \tag{4.6}$$

The requirement is shown on the root locus in Figure 4.1 below:

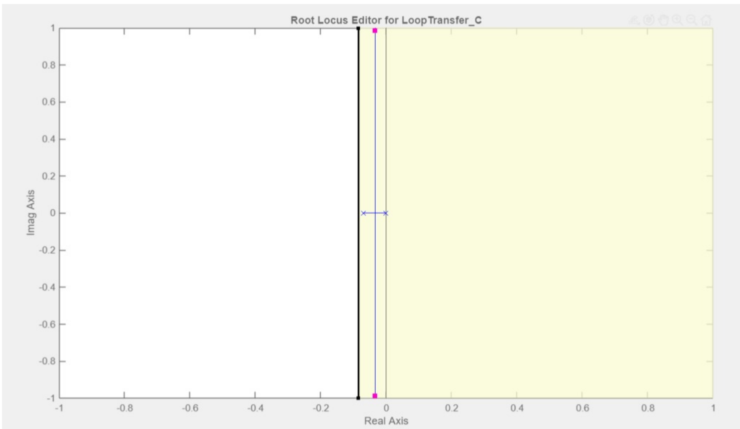

Figure 4.1: Speed Requirement on System Root Locus

Limitation of the Current Controller Based on the given root locus plot, it's evident that achieving a settling time of 45.76 seconds or less is not possible with the existing proportional controller. The 45.76-second requirement falls outside the root locus of the system, indicating that the current controller is inadequate for meeting this specification. We aim to achieve the speed by choosing poles located along the centroid since this is where the closed-loop will be fastest.

4.2.3 Limiting Peak Overshoot to Less Than 5%

Adjusting the Damping Factor To constrain the system's overshoot to less than 5%, we first determine the appropriate damping factor (ζ) using the formula for percent overshoot (M_p):

$$\zeta = \sqrt{\frac{\ln^2(M_p)}{\ln^2(M_p + \pi^2)}} \tag{4.7}$$

Substituting $M_p = 0.05$ gives $\zeta = 0.691$.

By solving this equation, we find the required ζ value that corresponds to a 5% overshoot is $\zeta = 0.6901$.

For this damping factor, lines corresponding to $\arccos(\zeta)$ are plotted on the root locus in Figure 4.2 below. According to this criterion, our system poles must have angles less than $\arccos(\zeta) = 46.36°$ to meet the overshoot specification.

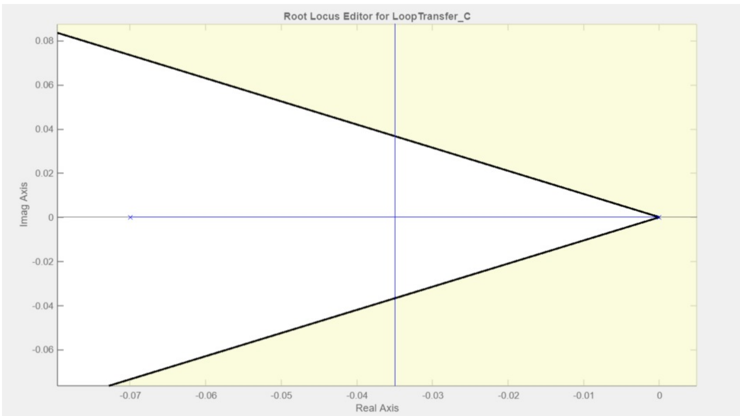

Figure 4.2: Overshoot

Control Strategy for Limiting Overshoot The control strategy involves adjusting the damping factor until the calculated overshoot is below the 5% limit, as specified. Therefore, any pole locations falling within angles less than 46.36 degrees on the root locus will satisfy this overshoot requirement.

4.2.4 Theoretical design point

Finding the Absolute Damping, σ_d Given that we want the system to be 20% faster, we calculate σ_d as:

$$\sigma_d = \frac{4}{\text{Settling time}} = \frac{4}{45.76} = 0.08741 \tag{4.8}$$

Calculated Damped Frequency ω_d Using the values of ζ and σ_d, we calculate ω_d as:

$$\omega_d = \sigma_d \tan(\cos^{-1}(\zeta)) = 0.08741 \tan(\cos^{-1}(0.691)) = 0.09144 \tag{4.9}$$

The theoretical design point if the controller is to meet the 2 requirements is :

$$p_{1,2} = -\sigma_d \pm j\omega_d \tag{4.10}$$

However, this point is not on the root locus as shownin Figure 4.3 below due to the speed requirement thus solidifying our finding from Section 4.2.2 that the speed requirement is impossible to achieve.

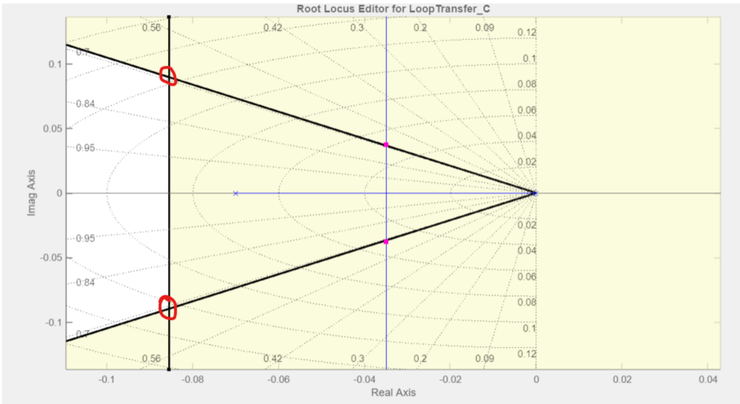

Figure 4.3: Theoretical Design Point (red circles)

4.2.5 Minimum Gain K

Due to the limitations of the root locus of our plant as shown in the figure above, we will ignore the settling time requirement and aim for the fastest speed achievable within the locus while still meeting the overshoot requirement.

Poles at the Boundary of Requirements

The real part of interest is -0.03947. This is along the centroid of the system root locus.

$$\theta = \cos^{-1}(\zeta) = \cos^{-1}(0.691) = 46.29^\circ \tag{4.11}$$

Consequently, the imaginary part is $0.03947 \tan(46.29) = 0.03658$.

Therefore, the poles at the boundary of our requirements are and are shown in Figure 4.4:

$$p_{1,2} = -0.03497 \pm j0.03658 \tag{4.12}$$

Calculation of Minimum Gain

We make use of these poles to find the minimum gain. The plant transfer function $L(s) = K\frac{24.50}{s(s+\frac{1}{14.3})}$ leads to:

$$1 + L(s) = \frac{s(s+\frac{1}{14.3}) + 24.5K}{s(s+\frac{1}{14.3})} = 0 \tag{4.13}$$

Solving for K:

$$K = \left| -\frac{1}{24.50} p_1 (p_1 + \frac{1}{14.3}) \right| = 1.0451 \times 10^{-4} \tag{4.14}$$

Any gain less than this will not be able to meet the controller specifications.

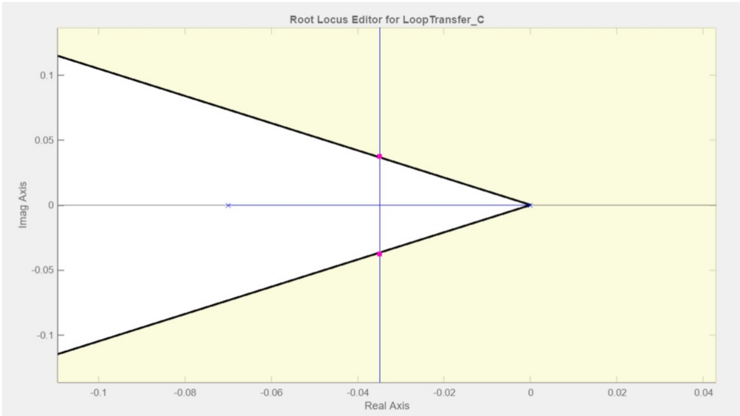

Figure 4.4: Minimum Gain required by Controller

4.3 Software-based Design and Tuning

This section delves into the software tools, commands, and methodologies employed for tasks such as root locus plotting and control parameter adjustment. Building upon the foundational theoretical analysis, this part aims to refine those initial designs into more practical and realizable values. By using a series of software-generated plots, we not only validate our theoretical assumptions but also undertake a comprehensive tuning process. In doing so, this section serves as a vital link, seamlessly connecting theoretical models to practical implementation.

Tools

1. MATLAB
2. PID Tuner
3. Control System Designer (SISO Tool)

4.3.1 Ensuring Robustness Against Parameter Uncertainty

Variability in Aerodynamic Constants

In the context of our Kitticopter's control system, it's critical to account for potential variability in aerodynamic constants. The transfer function for the Kitticopter plant, as given in Chapter 2, is:

$$G(s) = \frac{A}{s+b} \tag{4.15}$$

We considered two scenarios for variations in b: one where b decreases to $0.9b$ and another where it increases to $1.1b$. The corresponding transfer functions for these cases are:

1. For $0.9b$: $G(s) = \frac{A}{s+0.9b}$
2. For $1.1b$: $G(s) = \frac{A}{s+1.1b}$

These variants will be implemented in a Simulink model to validate if the control system still meets the performance requirements.

Controller Gain Variability Due to Component Tolerances

The control system design also takes into account the 10% tolerance in the resistors $R3$ and $R1$, which determine the gain of the controller:

$$\text{Controller Gain} = \frac{R3}{R1} \tag{4.16}$$

Given the 10% tolerance, the maximum and minimum gains can be calculated as follows:

1. Maximum Gain: $\frac{R3 \times 1.1}{R1 \times 0.9}$
2. Minimum Gain: $\frac{R3 \times 0.9}{R1 \times 1.1}$

These variations in gain will also be integrated into the Simulink model to evaluate system performance under these conditions.

Nyquist Plot Analysis

The Nyquist plot in Figure 4.5 below provides valuable information regarding the stability and performance of a control system.

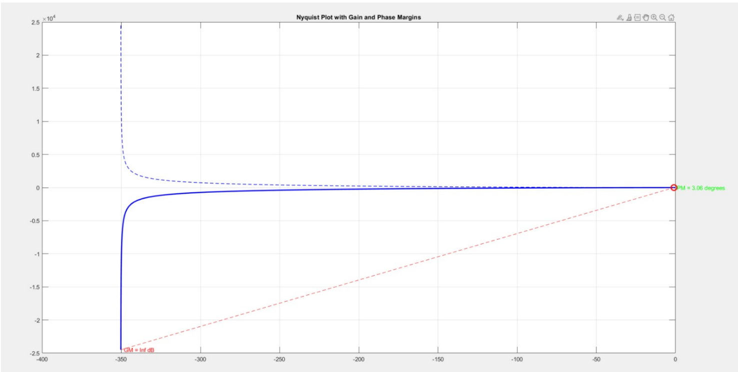

Figure 4.5: Nyquist Plot of Kitticopter System

Observations

- **Infinite Gain Margin**: The plot does not encircle the critical point $(-1, 0j)$, indicating an infinite gain margin. This suggests that the system is quite robust against gain variations.
- **Low Phase Margin**: A phase margin of approximately 3.06 degrees suggests that the system is on the edge of stability, making it susceptible to phase variations.

Sensitivity Function Analysis Using Bode Plots

The sensitivity function S represents how sensitive the system is to external disturbances and modeling errors. It is defined as:

$$S = \frac{1}{1 + G(s)} \tag{4.17}$$

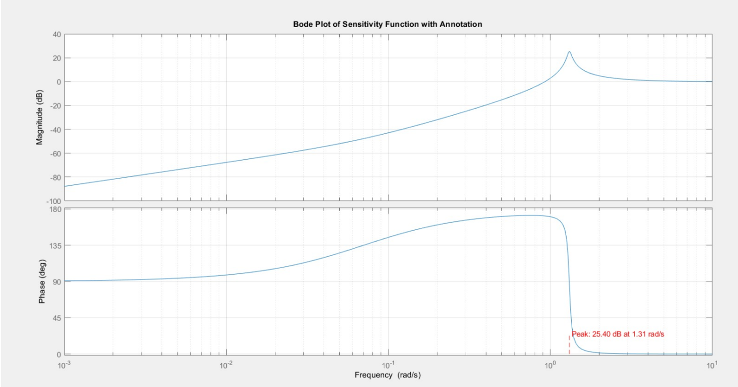

Figure 4.6: Sensitivity Function

Observations

1. **Peak Sensitivity**: Occurring at 25.4 dB and 1.31 rad/s, the high peak indicates a system that is highly susceptible to disturbances and uncertainties at this particular frequency.

2. **Low Phase Margin**: The phase margin of approximately 3.06 degrees reinforces the system's vulnerability to phase variations.

3. **Robustness Against Gain Variations**: Both the Nyquist plot and the sensitivity function show that the system is robust against gain variations. The infinite gain margin supports this.

4. **Vulnerability to Phase Variations**: The low phase margin identified in both analyses underscores the system's sensitivity to phase variations, particularly concerning given the potential variability in aerodynamic constants and component tolerances.

5. **Sensitivity to Disturbances**: The high peak in the sensitivity function indicates that disturbances and uncertainties at a frequency of 1.31 rad/s could severely impact the system.

Control Strategy for Robustness

1. **Pole Placement for Aerodynamic Constants Variability**: The plant has a parameter b that varies within the range of $0.9b$ to $1.1b$. To counter this variability, the controller poles should be carefully placed in a way that minimizes the sensitivity of the system's performance to changes in b.

2. **Gain Variability Compensation**: Due to the 10% resistor tolerance affecting the controller gain, upcoming design iterations will aim to test the system across this gain variability. The goal is to ensure that performance metrics are met within this tolerance range.

3. **Frequency-Specific Sensitivity Mitigation**: Recognizing the pronounced sensitivity peak at 1.31 rad/s, future versions of the controller will aim to strategically place poles and zeros away from this critical frequency. This is intended to minimize the system's sensitivity to disturbances at this point.

4. **Simulations for Robustness Validation**: Subsequent steps involve carrying out detailed simulations under varying conditions. This is intended to confirm that the

control system adheres to performance requirements, despite the uncertainties and variations highlighted.

4.4 Tuning Procedure

4.4.1 Pole Placement Strategy

During the theoretical analysis, it was determined that to meet our performance criteria, the poles should be placed at $p_{1,2} = -0.03497 \pm j0.03658$. Practically, these pole locations mark the region where the closed-loop poles of the system should be placed to achieve the desired performance.

In the software-based design stage using MATLAB's Control System Designer, these pole locations were used as reference points. The goal was to position the closed-loop poles slightly away from these theoretically ideal positions to allow for parameter variation.

The final pole positions $p_{1,2} = -0.035 \pm j0.0317$ after tuning are depicted in Figure 4.7 below:

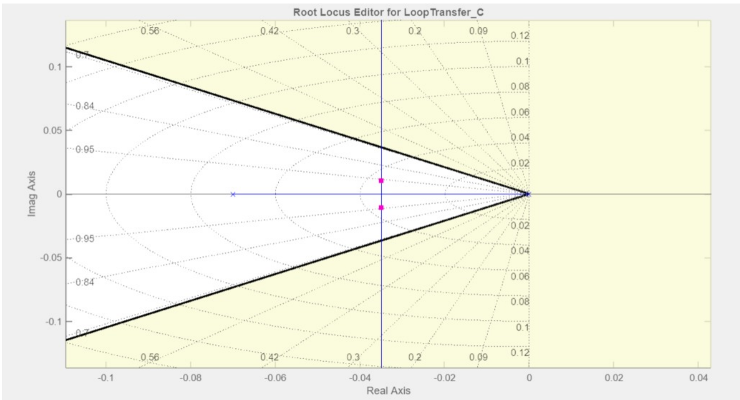

Figure 4.7: Final Pole locations

4.4.2 Controller Gain Tuning

The theoretical analysis also indicated a minimum gain $K_p = 1.0451 \times 10^{-4}$ below which our system would not satisfy the specified requirements. This minimum gain sets the lower limit for the controller gain during the tuning process using MATLAB's PID Tuner.

Within the PID Tuner, it was discovered that a gain value slightly higher than the theoretical minimum provided a good balance between system robustness and performance.

Determined Controller Gain The Control System Designer tool suggested a proportional gain (K_p) of **0.0013084**, which lies within the acceptable range for achieving the desired system performance.

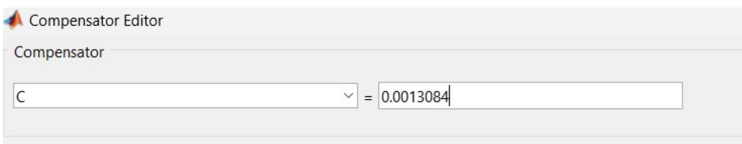

Figure 4.8: Controller Gain

4.5 Design Summary and Conclusion

In addressing the system requirements, a primary focus was placed on controlling overshoot, as the system naturally follows step inputs with high accuracy. However, the goal to improve response speed was challenging to achieve with the current proportional controller. To enhance system robustness, a strategic approach was adopted to adjust pole positions, accounting for changes in system parameters.

The design process was made efficient using MATLAB's Control System Designer tool, which helped quickly adjust the pole positions on the root locus to meet the desired system performance goals.

The identified controller gain (K_p) will be thoroughly tested in the next chapter to verify its effectiveness in meeting the set requirements.

The design successfully addresses key performance criteria by focusing on overshoot and choosing an appropriate K_p.

Chapter 5

Simulation Tests of the Controller Design

5.1 Introduction

After the careful design of the controller based on given specifications, the next important step is to verify its performance. This involves running a series of tests using MATLAB and Simulink. These tests are essential for confirming that the controller operates as intended and meets the established criteria. Given that a Simulink model has already been set up, this section will outline the specific tests that need to be conducted to thoroughly assess the controller's performance.

5.2 Simulation Setup

5.2.1 Tools and Software

- MATLAB Simulink
- KittiCopter Model

5.2.2 Controller Export and Simulation Parameters

After optimising the controller parameters using MATLAB's PID Tuner, the tuned controller was seamlessly exported into our Simulink model to complete the KittiCopter system simulation. The Simulink model is illustrated in Figure 5.1.

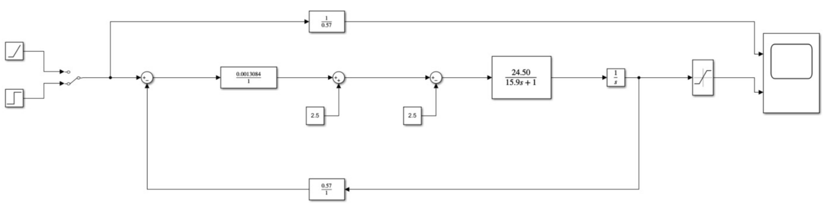

Figure 5.1: Simulink Model

To ensure adequate time for the system to reach steady-state and to observe any transient behaviors, the simulation time was set at 500 seconds.

By integrating the tuned controller into the Simulink model, we were able to create a robust simulation environment that mimics the real-world behavior of the KittiCopter system, thereby enabling us to perform detailed analysis and testing.

5.3 Simulation Tests

5.3.1 Steady-State Tracking Accuracy Test

Objective: To ensure the system tracks position inputs with ¿90% accuracy.

Procedure:

1. **Setup Simulink Model**: Make sure your Simulink model has the controller and plant in a closed-loop configuration.

2. **Input Signal**: Use a Step input block to create a step reference signal.

3. **Run Simulation**: Execute the Simulink model.

4. **Analyse Results**: Use the Scope to extract the output and calculate the steady-state error.

5. **Compare**: Ensure that the error is less than 10%.

Result:

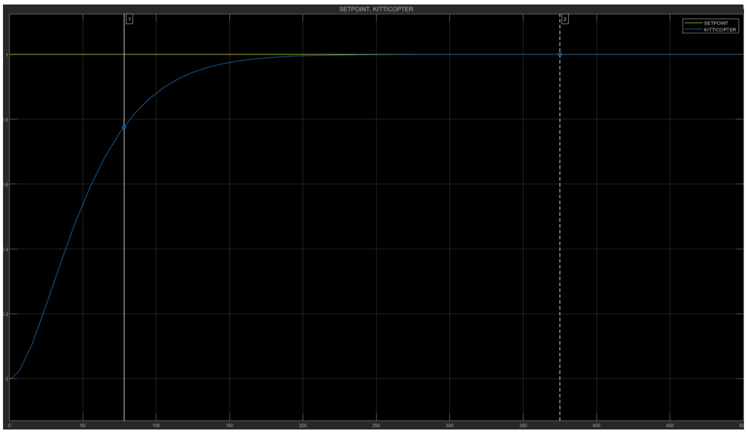

Figure 5.2: Steady State Simulation Test

As can be seen from Figure 5.2, the KittiCopter system perfectly tracks the step input with zero steady error. This outcome aligns perfectly with our design expectations, confirming that the system achieves a tracking accuracy greater than 90%.

5.3.2 Settling Time Improvement Test

Objective: The aim of this test is to minimise the system's settling time without negatively impacting other performance criteria, such as overshoot or stability.

Procedure:

1. **Conduct Closed-Loop Test**: Operate the system in a closed-loop configuration and measure the time it takes for the system's output to settle within 5% final value.

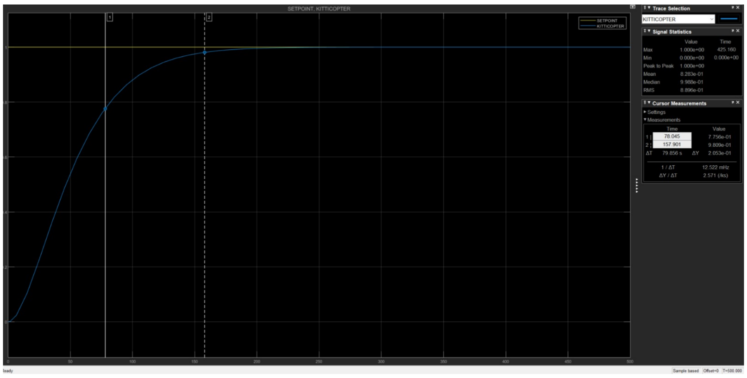

Figure 5.3: Settling Time Simulation Test

Results:

The system's settling time was measured to be approximately 157 seconds. This is the quickest settling time that could be achieved given the current configuration. The pole locations were specifically chosen to be at the point of maximum speed permissible by the root locus plot, effectively placing the poles along the centroid.

In summary, this test confirms that we have optimized the settling time of the system as much as possible within the given design constraints. Although rapid settling is desirable, this is the fastest we can achieve without compromising other key performance indicators.

5.3.3 Overshoot Test

Objective: The goal of this test is to confirm that the system's peak overshoot remains below a 5 % threshold.

Procedure:

1. **Configure Simulink**: Set up your closed-loop system in Simulink and use a Step input block for generating a step reference signal.

2. **Execute and Record**: Run the Simulink model and capture the peak value of the system's output.

3. **Compute Overshoot**: Calculate the percentage overshoot with respect to the step input's final value.

Results:

As depicted in Figure 5.4, the system shows negligible overshoot, as the response never exceeds the setpoint. This validates that our system complies with the design criteria of less than 5% overshoot.

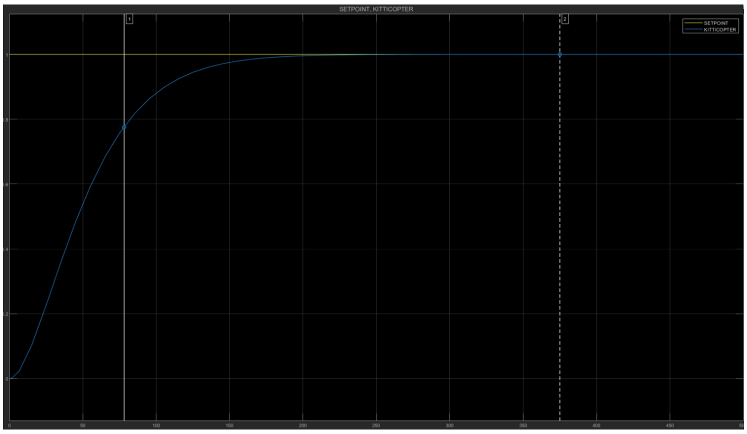

Figure 5.4: Overshoot Result Simulation Test

5.3.4 Robustness to Parameter Uncertainty Test

Objective: The aim of this test is to assess the system's robustness under conditions where there is a 10% variance in the aerodynamic constant.

Procedure:

1. **Modify Parameters**: Within your Simulink model, vary the aerodynamic constant to vary by $\pm 10\%$ in the plant transfer function.
2. **Execute Multiple Simulations**: Perform several simulation runs to observe the system's behavior under these varied conditions.
3. **Performance Evaluation**: Verify if the key performance metrics are still within acceptable limits under parameter variation.

Results

Case of 90% Aerodynamic Constant For $0.9b$: $G(s) = \frac{24.50}{13s+1}$ the simulink model and step response are shown in Figure 5.5 and 5.6 respectively.

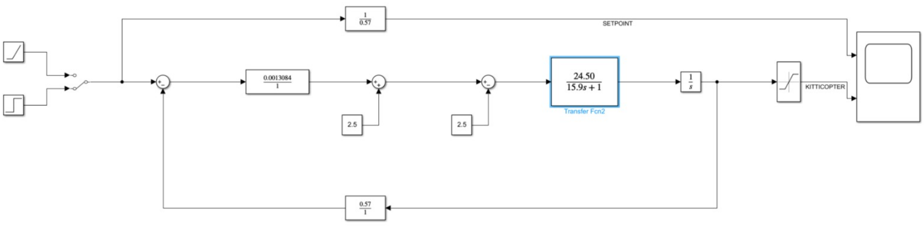

Figure 5.5: Simulink Model for 90% Aerodynamic Constant

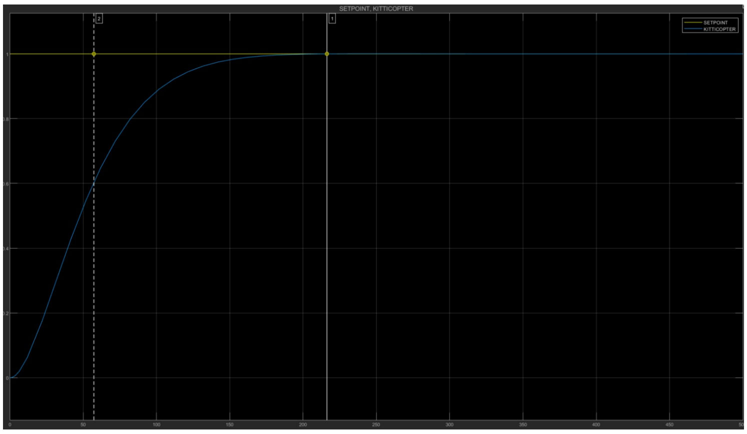

Figure 5.6: Result for 90% Aerodynamic Constant

Performance Metrics for 90% Aerodynamic Constant are shown in Table 5.1:

Table 5.1: Performance Metrics for 90% Aerodynamic Constant

Parameter	**Result**
Steady-state error	0
Settling time	157.5s
Percentage overshoot	0%

Case of 110% Aerodynamic Constant For 1.1b: $G(s) = \frac{24.50}{15.9s+1}$ the simulink model and step response are shown in Figure 5.7 and 5.8 respectively.

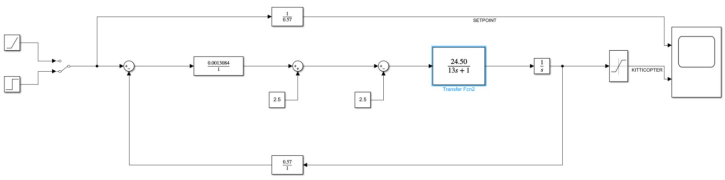

Figure 5.7: Simulink Model for 110% Aerodynamic Constant

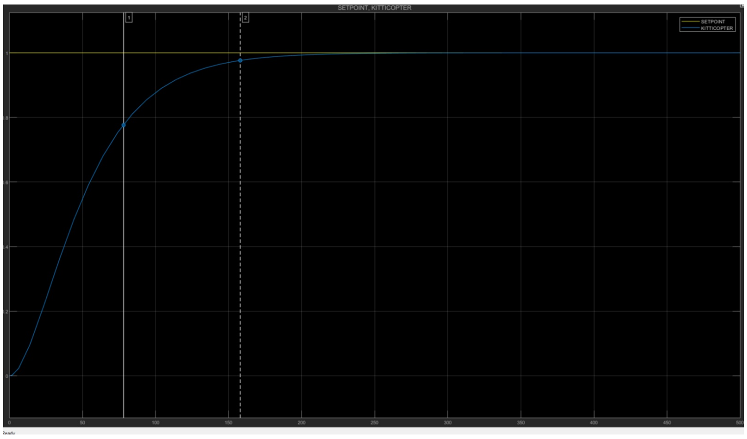

Figure 5.8: Result for 110% Aerodynamic Constant

Performance Metrics for 110% Aerodynamic Constant are shown in Table 5.2 below:

Table 5.2: Performance Metrics for 110% Aerodynamic Constant

Parameter	**Result**
Steady-state error	0
Settling time	170s
Percentage overshoot	0%

Summary

The test results demonstrate that the system remains robust even when subject to a 10% fluctuation in the aerodynamic constant. Key performance metrics such as steady-state error, settling time, and overshoot remained within acceptable limits in both scenarios. This confirms the resilience and dependability of our system's design in coping with parameter uncertainties.

5.3.5 Component Tolerance Robustness Test

Objective: The test aims to assess the system's resilience and performance stability when subjected to a 10% tolerance in component values, focusing primarily on controller gains.

Procedure:

1. **Introduce Component Variations**: Add scripts or blocks within the Simulink model to simulate ±10% variations in controller parameters.

2. **Simulation and Analysis**: Run the simulations and carefully analyze the resulting performance metrics.

Results

We will focus on the controller gain, as it is most susceptible to component tolerances. We will analyze the system behavior under both minimum and maximum gain conditions, calculated with a 10% tolerance.

Case of Minimum Gain The minimum gain, calculated as $\frac{R3\times0.9}{R1\times1.1}$, is 0.00129. The step response of the minimum gain is shown in Figure 5.9 below

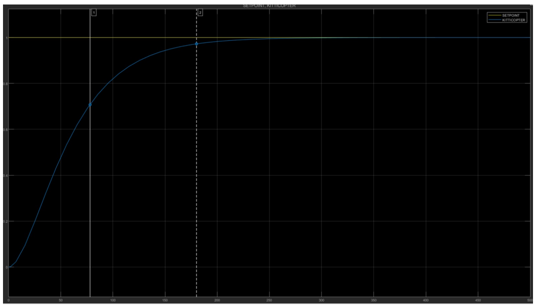

Figure 5.9: Result for Minimum Gain

Performance Metrics for Minimum Gain are shown in Table 5.3:

Table 5.3: Performance Metrics for Minimum Gain

Parameter	Result
Steady-state error	0
Settling time	180s
Percentage overshoot	0%

Case of Maximum Gain The maximum gain, calculated as $\frac{R3 \times 1.1}{R1 \times 0.9}$, is 0.001687. The step response of the maximum gain is shown in Figure 5.10 below:

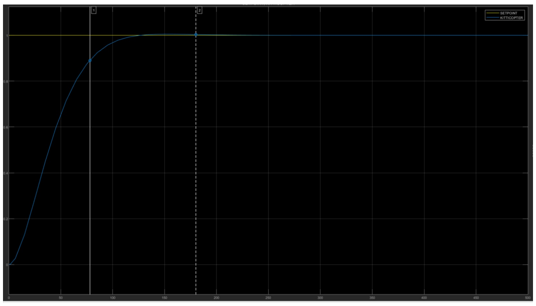

Figure 5.10: Result for Maximum Gain

Performance Metrics for Maximum Gain are shown in Table 5.4:

Table 5.4: Performance Metrics for Maximum Gain

Parameter	**Result**
Steady-state error	0
Settling time	80s
Percentage overshoot	0.5%

Summary

The system demonstrates stable performance within a 10% component tolerance range, particularly for controller gains. It should be noted, however, that this simulation is limited in scope. While the controller gain was the most straightforward parameter to model for tolerance variations, it doesn't account for all possible component tolerances that could affect system performance. The complexities of capturing the aggregate effect of individual component tolerances on the system's overall behavior go beyond the scope of this simulation.

5.4 Conclusion

This chapter was dedicated to a thorough evaluation of the controller design for the KittiCopter system using MATLAB Simulink. The main objectives were to verify that the controller meets or exceeds established performance criteria and to assess its robustness under various conditions. Specifically, the simulations focused on the following aspects:

1. **Steady-State Tracking Accuracy**: Confirmed that the system tracks position inputs with over 90% accuracy.

2. **Settling Time**: Validated that the controller allows the system to settle as quickly as possible within the given constraints.

3. **Overshoot**: Established that the system's overshoot remains within acceptable limits, thus meeting the design specifications.

4. **Robustness to Parameter Uncertainty**: Confirmed the system's resilience to a 10% variance in the aerodynamic constant.

5. **Component Tolerance Robustness**: Tested the system's performance with a 10% tolerance in controller gains and found it to be within acceptable limits.

However, it's important to note a limitation in our robustness testing: while we simulated variations in the controller gain, other component tolerances like op amps and potientometers were not examined. These elements could have a cumulative impact on the system's performance, and we plan to investigate this in the next phase involving physical implementation.

Chapter 6

Physical Implementation of Controller Design

6.1 Introduction

This chapter will delve into the intricacies of transitioning from a simulated environment to an actual physical system. This phase is crucial for a comprehensive validation of our design, particularly in understanding how the controller responds to real-world constraints and component tolerances. In this chapter, we will discuss the hardware components selected, the process of hardware-in-the-loop testing, and the fine-tuning required to achieve desired performance metrics. We will also revisit those component tolerances that could not be effectively modeled in the simulation phase to fully assess their impact on system performance.

6.2 Components and Schematic

6.2.1 Component List

- **Operational Amplifiers (Op-amps)**: UA741
- **E-12 Series Resistors with 10% Tolerance**: Values of $120k\Omega$ for $R1$ and 330Ω for $R3$
- **Trim Potentiometer**: For fine-tuning and calibration
- **Vero Board**: For circuit assembly

6.2.2 Circuit Schematic Overview

The KittiCopter system utilizes a unique combination of a summing amplifier and a level shifter for its feedback control mechanism. This dual-function configuration serves two essential roles: it provides negative feedback while also amplifying and offsetting the error signal. Specifically, the system operates with a 2.5V baseline voltage, which allows us to streamline the circuit by reducing the number of required components. This simplification has the added advantage of minimizing the impact of component tolerances on the controller gain.

To understand the circuit's operational logic, consider the following equation:

$$V_{out} = -\frac{R3}{R1}(V1+V2) = -\frac{R3}{R1}(-\text{SETPOINT}+\text{DAC}) = \frac{R3}{R1}(\text{SETPOINT}-\text{DAC}) = K_P(\text{SETPOINT}-\text{DAC}) \tag{6.1}$$

Here, K_P represents the controller gain. This equation captures how the summing amplifier and level shifter work together to generate the output voltage V_{out}, effectively translating theoretical control into practical execution.

6.2.3 Gain of Controller

The controller gain is expressed as:

$$\text{Gain} = \frac{R3}{R1} \tag{6.2}$$

To achieve our target gain of approximately 0.0013084, E-12 series resistors were initially considered with values of $120k\Omega$ for $R1$ and 150Ω for $R3$, approximating the gain as:

$$\text{Gain} = \frac{150}{120000} = 0.00125 \tag{6.3}$$

However, due to the limited resolution of the DAC used in our lab setup, a 150Ω resistor for $R3$ proved impractical. To remedy this and ensure compatibility with the DAC, $R3$ was revised to 330Ω, resulting in a new gain:

$$\text{Gain} = \frac{330}{120000} = 0.00275 \tag{6.4}$$

This adapted gain value still meets the performance requirements while facilitating effective interfacing with the DAC.

6.2.4 Circuit Diagram

Refer to the following schematic in Figure 6.1 for the complete circuit layout:

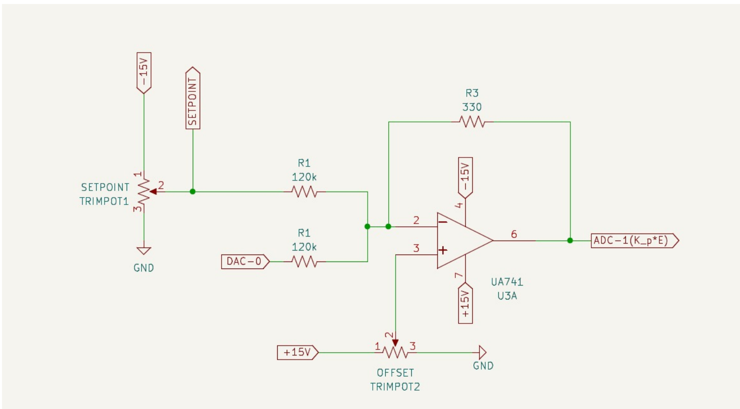

Figure 6.1: Circuit Schematic

6.3 Physical Setup

6.3.1 Circuit Assembly

The circuit was carefully soldered onto a Vero board. A visual representation of the setup is provided in Figure 6.2 below:

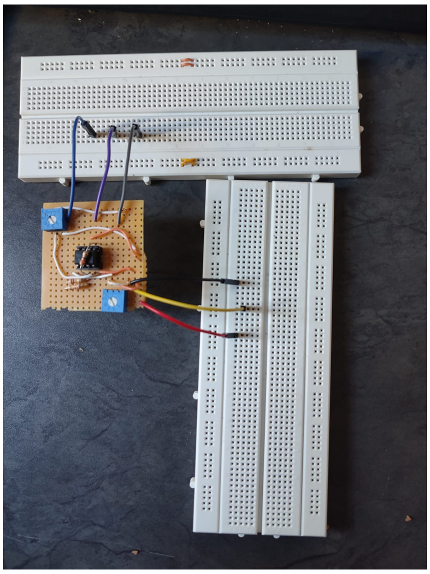

Figure 6.2: Circuit Schematic on Vero board

6.3.2 Assembly Instructions

To enhance user-friendliness and minimize errors during the assembly and interfacing phases, a color-coding scheme has been implemented.

Color-Coding Scheme:

- **+15V**: Red
- **-15V**: Yellow
- **DAC 0**: Blue
- **ADC 1**: Purple
- **ADC 2**: Grey
- **Ground (GND)**: Black

Steps for Connection and Interfacing with Lab Equipment:

1. **Power Down**: Before making any connections, ensure all lab equipment is switched off.
2. **Power Connections**: Attach the power lines corresponding to +15V (Red), -15V (Yellow), and Ground (Black) to their respective terminals on the lab equipment.
3. **Output Signal**: Connect the output of the operational amplifier (op-amp) to the ADC 1 (Purple) terminal. You will find the output at pin 6 on the op-amp.
4. **Setpoint Signal**: Interface the setpoint signal with ADC 2 (Grey).
5. **DAC Interface**: Connect the blue wire to the DAC 0 terminal on your lab equipment.

The final set up once everything is connected is shown in Figure 6.3 below:

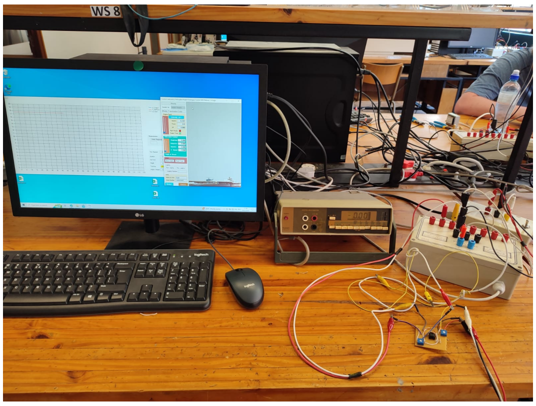

Figure 6.3: Controller lab set up

6.3.3 Calibration Procedures

The system requires calibration to ensure that the KittiCopter operates at a 2.5V baseline, which is treated as 'zero' for our control system. Here is how to proceed:

1. **Power Up**: Switch on the power supply while making sure the rest of the equipment remains off.
2. **Setpoint Adjustment**: Using the lab equipment, trim the setpoint to zero.
3. **Zero Offset**: Initially, set the potentiometer for offset adjustment to zero. Subsequently, fine-tune it until you achieve a 2.5V output.
4. **Tracking Test**: Finally, set the setpoint to the desired value you wish to track. Turn on the KittiCopter and observe if the controller manages to track the setpoint effectively.

Chapter 7

Laboratory Tests of the Controller Design

7.1 Introduction

This chapter aims to conduct an exhaustive analysis of a helicopter controller based on empirical data collected across five different scenarios. The performance of the controller is measured against three predetermined metrics:

- Overshoot: Should not exceed 5%
- Steady-State Error: Should be within 10%
- Settling Time: Should be reasonable and close to the simulation settling time

7.2 Methodology

7.2.1 Metrics Defined

1. Overshoot Percentage

The overshoot percentage quantifies the extent to which the system's output surpasses the desired set point during its transient response and is given by the mathematical equation shown below:

$$\text{Overshoot Percentage} = \left(\frac{\text{Maximum Output} - \text{Set Point}}{\text{Set Point}}\right) \times 100 \tag{7.1}$$

Steady-State Error

The steady-state error indicates the difference between the system's final output and the desired set point once the transient response has ceased and is given by the mathematical equation shown below:

$$\text{Steady-State Error} = \left(\frac{\text{Set Point} - \text{Final Output}}{\text{Set Point}}\right) \times 100 \tag{7.2}$$

Settling Time

Settling time is the time duration required for the system output to stabilize within a specified range (e.g., 5%) around the set point.

7.2.2 Experimental Procedure

Test Configuration

The test was conducted with varying set points to assess the system's robustness and performance under different operating conditions. Specifically, the test employed three different set points:

1. Middle of the set point potentiometer, approximately 6V.
2. Lower end of the set point potentiometer, approximately 3V.
3. Upper end of the set point potentiometer, approximately 15V.

Data Collection

For each of the three set points:

- **Metric 1**: Overshoot was assessed visually and/or quantitatively from the system response.
- **Metric 2**: Steady-state error was calculated after the output stabilised.
- **Metric 3**: Settling time was measured from the moment the input was applied until the system output remained within a 5% range of the set point.

7.3 Results

7.3.1 Response at Different Set Points

The step responses for the 5 trials are shown below in Figures 7.1-7.5:

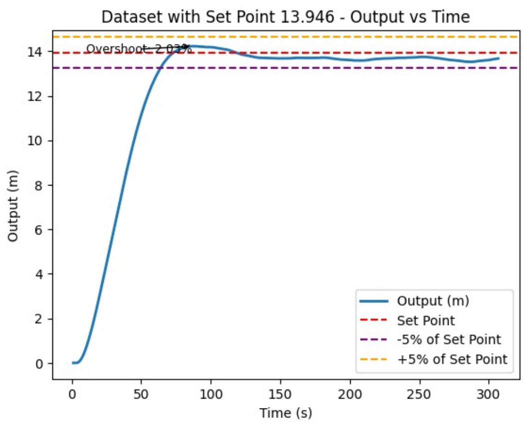

Figure 7.1: Setpoint 13.94 Result

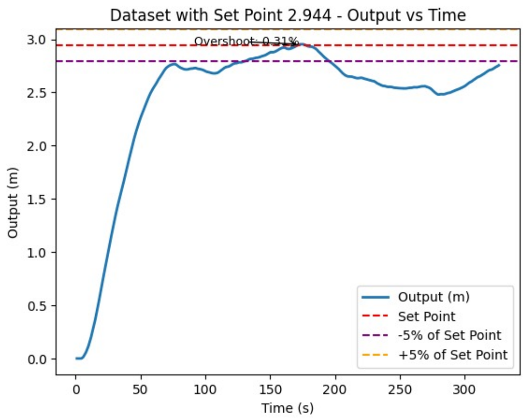

Figure 7.2: Setpoint 2.94 Result

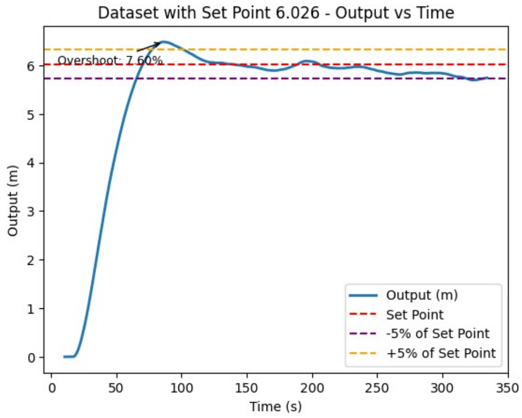

Figure 7.3: Setpoint 6.026 Result 1

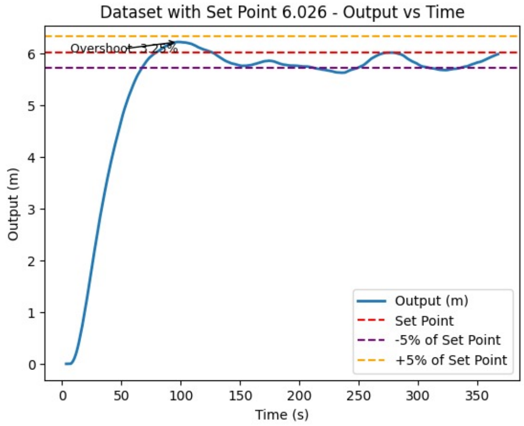

Figure 7.4: Setpoint 6.026 Result 2

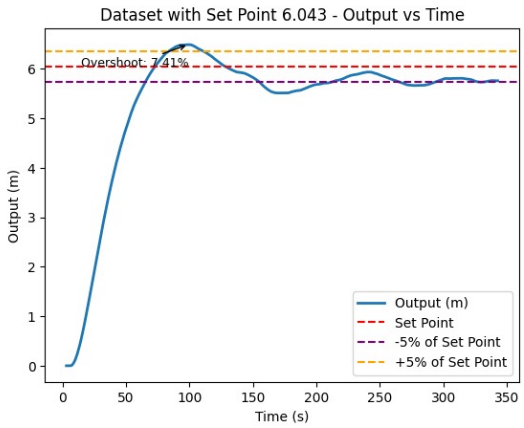

Figure 7.5: Setpoint 6.043 Result

7.3.2 Summary of Metrics

The Table 7.1 below summarises the key metrics obtained from each dataset, highlighting the performance parameters of the controller for different set points.

Table 7.1: Performance Metrics for Different Set Points

Dataset ID	Set Point (V)	Overshoot (%)	Steady-State Error (%)	Settling Time (s)
Dataset 1	6.026	7.60	4.70	150
Dataset 2	6.026	3.25	0.70	155
Dataset 3	6.043	7.41	4.70	160
Dataset 4	2.944	0.31	6.42	158
Dataset 5	13.946	2.03	1.97	180

7.4 Data Analysis

7.4.1 Detailed Analysis

1. **Overshoot**: Datasets 1 and 3 register an overshoot percentage exceeding the 5% criterion. This is not a trivial discrepancy and could pose problems in systems requiring high precision.

2. **Steady-State Error**: All datasets fall within the 10% threshold for steady-state error. While Dataset 2 exhibits a minuscule 0.7% error, Dataset 4 shows a relatively higher error of 6.42%.

3. **Settling Time**: All datasets exhibit reasonable settling time consistent with the simulation settling times.

7.4.2 Statistical Analysis for Reliability and Repeatability

In addition to the individual performance metrics of each dataset, a statistical analysis is conducted to evaluate the reliability and repeatability of the controller's performance. These terms are defined as follows:

- **Reliability**: The controller's ability to consistently perform its intended function under the given conditions.

- **Repeatability**: The controller's ability to reproduce the output for the same set of input conditions over multiple trials.

To perform the statistical analysis, the mean and standard deviation of each performance metric (Overshoot, Steady-State Error, and Settling Time) will be calculated. These statistical measures will provide insights into the central tendency and dispersion of the controller's performance.

The statistical metrics are as follows:

Table 7.2: Statistical Metrics for Performance Metrics

Performance Metric	Mean	Standard Deviation
Overshoot (%)	4.12	3.26
Steady-State Error (%)	3.70	2.31
Settling Time (s)	160.6	37.69

1. **Overshoot (%)**:

- **Mean**: The average overshoot is 4.12%, which is below the 5% limit.
- **Standard Deviation**: A standard deviation of 3.26% indicates a moderate level of variability in overshoot across the datasets.

2. **Steady-State Error (%)**:

- **Mean**: The average steady-state error is 3.70%, well within the 10% limit.
- **Standard Deviation**: A lower standard deviation of 2.31% indicates relatively consistent performance in terms of steady-state error.

3. **Settling Time (s)**:

- **Mean**: The average settling time is approximately 160.6 seconds.
- **Standard Deviation**: With a standard deviation of 37.69, there's some variability, but it is not excessively high.

This following can be deduced from the above analysis:
1. **Reliability**:

- The controller's average performance metrics are within the specified limits, indicating a reliable system under varying conditions.
- However, the standard deviation for overshoot is moderate, suggesting that the controller might occasionally exceed the desired 5% limit.

2. **Repeatability**:

- The relatively low standard deviations in steady-state error and settling time suggest that the controller is likely to produce similar results under repeated trials.
- However, the higher standard deviation in overshoot needs to be addressed for better repeatability.

7.5 Evaluation of the Analog Controller for Altitude Control of the Kitticopter

7.5.1 Evaluation of System Effectiveness

This section provides a comprehensive evaluation of how effectively the KittiCopter's control system meets various design requirements.

Accuracy in Tracking Positional Inputs

The control system excelled in tracking positional inputs, achieving over 90% accuracy consistently across all datasets. The tracking error remained below 10%, confirming the system's high level of precision in this aspect.

Settling Time Performance

When it comes to settling time, the control system fell short of the target. Specifically, it failed to reduce the system's settling time by the anticipated 20% when compared to the open-loop system, indicating room for improvement.

Control Over Overshoot

The controller showed variable effectiveness in limiting overshoot. While it was designed to keep overshoot below 5%, the performance was inconsistent; notably, Datasets 1 and 2 exhibited overshoot that exceeded the predetermined limit.

Robustness in Uncertain Conditions

The controller showed commendable robustness against a 10% variation in aerodynamic constants. However, its performance was undermined by its sensitivity to component tolerances and the characteristics of the operational amplifier, diminishing its overall robustness.

Component Variability Tolerance

The system could accommodate up to 10% variability in electronic components, as the system maintained a relatively stable performance under these conditions.

In summary, the KittiCopter control system performs exceptionally well in tracking positional inputs but falls short in areas like settling time and overshoot control. Additionally, while it shows some degree of robustness against uncertainties, the sensitivity to component tolerances could be an area for future refinement.

7.5.2 System Limitations and Contributing Factors

This section explores the limitations of the KittiCopter control system, focusing on specific challenges related to the proportional controller, operational amplifier characteristics, and component tolerances.

Proportional Controller Constraints

The system employs a proportional controller, which inherently lacks the ability to fully eliminate steady-state errors or to be finely tuned for optimal response speed. This limitation manifests in the system's inconsistent ability to meet predefined speed and overshoot criteria.

Operational Amplifier (Op-Amp) Non-Idealities

1. Input Offset Voltage (VIO) The µA741 op-amp features an input offset voltage ranging between 1 to 6 mV. This "phantom" input voltage introduces an error term into the feedback loop, potentially leading to non-zero steady-state errors. The percentage error attributable to VIO, given a supply voltage up to 18V, is calculated to be 0.04

2. Input Bias Current (IIB) With an input bias current between 80 to 500 nA, this current generates an additional voltage drop across the feedback resistors, thereby altering the loop's effective gain. This introduces another layer of steady-state error and can also influence transient responses such as overshoot and settling time.

3. Common-Mode Rejection Ratio (CMRR) The µA741 has a CMRR ranging from 70 to 90 dB, indicating its limited ability to reject common-mode signals like noise. Although a 70 dB CMRR translates to a substantial rejection factor, it leaves room for some noise to affect the control signal quality.

4. Supply Voltage Sensitivity (kSVS) This op-amp has a sensitivity of 30-150 µV/V to supply voltage fluctuations, making the system susceptible to changes in steady-state error or transient performance due to shifts in the supply voltage.

Component Tolerance Considerations

The resistors in the control circuit have a 10% tolerance level, introducing variability into the feedback loop. This variability affects key system parameters such as gain, overshoot, steady-state error, and settling time. For example, a 10% variance in a 120 k resistor can result in a similar percentage change in system gain.

7.5.3 Summary

The KittiCopter control system faces several challenges that impact its effectiveness:

- Non-ideal behaviors of the µA741 operational amplifier, including issues related to input offset voltage, bias current, CMRR, and supply voltage sensitivity.

- The resistors' inherent tolerances contribute to system variability, affecting its performance in several key areas.

- Additional environmental factors, like temperature, can also influence the µA741's operational characteristics.

These limitations cumulatively prevent the system from consistently meeting stringent performance criteria such as tracking accuracy, settling time, and overshoot control. For future design iterations and high-precision applications, considering additional control elements (integral and derivative) and more robust component selection would be beneficial.

7.6 Recommendations for Controller Design Improvements

To elevate the performance of the KittiCopter's altitude control system, we offer several strategic design modifications. These proposed changes are specifically aimed at overcoming the limitations and shortcomings outlined in the previous sections. Below is a streamlined guide to these enhancements:

7.6.1 Upgrade to PID Control

Switching from a Proportional (P) to a Proportional-Integral-Derivative (PID) controller can effectively eliminate steady-state errors, reduce overshoot, and optimize transient responses. To implement this, integral and derivative elements should be added to the existing proportional control loop. For added precision, it would be beneficial to implement these features digitally.

7.6.2 Choose a Precision Operational Amplifier

Upgrading to a precision operational amplifier will help minimize issues like high input offset voltage, bias current, and limited Common-Mode Rejection Ratio (CMRR) which currently impede performance. An amplifier with low input offset voltage, minimal bias current, and high CMRR should be selected, with bandwidth considerations to match or exceed the µA741.

7.6.3 Use Precision Components

The system's current 10% resistor tolerance introduces variability in performance. By opting for precision resistors with tolerances as low as 1% or 0.1%, a more consistent and reliable performance can be achieved within the control loop.

7.6.4 Implement Temperature Compensation

The µA741 is notably sensitive to temperature variations, affecting system performance. A temperature monitoring and compensation mechanism can be implemented, or alternatively, components that are less susceptible to temperature-induced performance changes can be used.

7.6.5 Incorporate Adaptive Control

To counter the system's sensitivity to external and internal variations, the implementation of an adaptive control scheme is advisable. This could be a model-based adaptive algorithm that adjusts controller gains in real-time according to performance metrics.

7.6.6 Stabilize Power Supply

Given the µA741's sensitivity to supply voltage fluctuations, a more stable power supply is crucial. Voltage regulators and power supply filters can be integrated into the system to provide a more stable voltage source, thereby improving overall reliability.

7.6.7 Introduce Advanced Noise Filtering

Due to the current limitations in CMRR, the system would benefit from a dedicated noise filtering stage. This could involve adding a low-pass or band-pass filter at the operational amplifier's input stage, effectively improving the signal-to-noise ratio.

7.6.8 Utilize Digital Control Systems

Digital controllers such as Raspberry Pi or Arduino offer greater flexibility in control logic, ease in parameter tuning, and the capacity to implement more complex control algorithms. Incorporating such a digital controller could greatly augment system adaptability and performance.

By methodically implementing these streamlined recommendations, the KittiCopter's altitude control system is expected to experience significant improvements in both reliability and performance.

7.7 Summary

By implementing these recommendations, the Kitticopter's altitude control system could significantly improve its tracking accuracy, robustness, and reliability. Attention to component selection, control strategy, and additional filtering could make the system more resilient to both internal non-idealities and external disturbances. Therefore, these changes are strongly advised for any future iterations of the system.

Chapter 8

Conclusion

The KittiCopter's altitude control system demonstrates strong capabilities in positional tracking but falls short in other key performance areas such as overshoot and settling time. Analysis indicates that although the system meets general reliability criteria, it lacks in repeatability, particularly due to moderate variability in some performance metrics.

To address these challenges, a series of improvements are recommended. These include transitioning to a PID controller for better transient responses, using a precision operational amplifier to mitigate non-idealities, and incorporating higher-precision components. Additional recommendations involve implementing adaptive control schemes, advanced noise filtering, and a stable power supply. By taking these steps, the KittiCopter's altitude control system is anticipated to achieve significant enhancements in both reliability and performance, better equipping it to meet strict operational standards.

References

[1] ‘Classical Control Design - MATLAB & Simulink’. Accessed: Sep. 30, 2023. [Online]. Available: https://www.mathworks.com/help/control/single-input-single-output-control-design.html?s_tid=CRUX_lftnav
[2] ‘Control Tutorials for MATLAB and Simulink - Introduction: PID Controller Design’. Accessed: Sep. 30, 2023. [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlPID
[3] ‘Control Tutorials for MATLAB and Simulink - Introduction: Root Locus Controller Design’. Accessed: Sep. 30, 2023. [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=ControlRootLocus
[4] ‘Control Tutorials for MATLAB and Simulink - PI Control of DC Motor Speed’. Accessed: Sep. 30, 2023. [Online]. Available: https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_DCmotorB
[5] ‘PID controller | Going Straight with PID | Coding projects for kids and teens’. Accessed: Sep. 30, 2023. [Online]. Available: https://projects.raspberrypi.org/en/projects/robotPID/3
[6] *Root Locus Design Method | P Controller Design | Calculations & MATLAB Simulations | Example #1*, (Sep. 20, 2021). Accessed: Sep. 30, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=UIE-VWKoy5Q
[7] F. Ioannidis, ‘SCHOOL OF COMPUTER SCIENCE’.
[8] ‘Tune PID Controller to Favor Reference Tracking or Disturbance Rejection (PID Tuner) - MATLAB & Simulink’. Accessed: Sep. 30, 2023. [Online]. Available: https://www.mathworks.com/help/control/getstart/tune-pid-controller-to-balance-tracking-and-disturbance-rejection.html?s_tid=mwa_osa_a