



UNIVERSITY OF THE  
WITWATERSRAND,  
JOHANNESBURG



SCHOOL OF MECHANICAL,  
INDUSTRIAL & AERONAUTICAL  
ENGINEERING

# Solar Tracker

MECN4029A - Mechatronics II

## Group 14

|                    |         |
|--------------------|---------|
| Lindo Mtsweni      | 1794257 |
| Castello Chetty    | 2081719 |
| Santhiran Govender | 1242722 |

A project report submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Bachelor of Science in Engineering.

Johannesburg, May 2024



**Disclosure – Use of Artificial-Intelligence (AI) Generated Content**

2024 V2

***Students must acknowledge all use of AI.***

Select all applicable statements and complete the sections fully. Delete all statements that are not applicable.

***1. Disclosure: No AI use***

I acknowledge that no AI tools/technologies (Grammarly, ChatGPT, Bard, Quillbot, OpenAI etc) were used in the completion of this assessment.

***I declare that the disclosure is complete and truthful.***

Student number: 2081719; 1242722 1794257;

Course code: MECN4029A

Date: 20 May 2024

## **Executive Summary**

This report presents an in-depth analysis of control strategies for a solar tracking system, focusing on the comparison between PID and Root Locus controllers. The Root Locus controller was selected for its superior performance, achieving a rise time of 0.1641 seconds and a settling time of 4.2408 seconds, both surpassing the desired specifications. Additionally, it maintained an overshoot of 9.85% and eliminated steady-state error completely. With a phase margin of 123.92 degrees and a gain margin of 13.393 dB, the Root Locus controller ensured robust performance and stability, resulting in excellent disturbance rejection and energy generation capabilities. The findings demonstrate that the complex implementation of the Root Locus controller is justified by its significant performance benefits. These metrics underline the Root Locus controller's ability to provide precise and reliable solar tracking, which is critical for optimizing the efficiency of solar energy systems. The analysis also confirms that despite its inherent complexity, the Root Locus controller provides significant improvements in system performance, making it the preferred choice for the modelled solar tracker.

# Contents

|  |             |
|--|-------------|
| <b>Contents</b>                                  | <b>iii</b>  |
| <b>List of Figures</b>                           | <b>viii</b> |
| <b>List of Tables</b>                            | <b>xi</b>   |
| <b>1 Introduction</b>                            | <b>1</b>    |
| 1.1 Background . . . . .                         | 1           |
| 1.2 Existing Solutions . . . . .                 | 1           |
| 1.3 Problem Breakdown . . . . .                  | 3           |
| 1.4 Methodology . . . . .                        | 4           |
| 1.5 Objectives . . . . .                         | 5           |
| 1.6 Problem Description . . . . .                | 7           |
| <b>2 Physical Model</b>                          | <b>10</b>   |
| <b>3 Mathematical Model</b>                      | <b>14</b>   |
| 3.1 Modelling System . . . . .                   | 14          |
| 3.2 Modelling Wind Disturbance . . . . .         | 14          |
| 3.2.0.1 Solar Tracker Dynamics . . . . .         | 17          |
| 3.2.0.2 Motor Dynamics . . . . .                 | 17          |
| 3.2.0.3 Wind Disturbance . . . . .               | 17          |
| 3.3 Desired Performance Specifications . . . . . | 17          |
| 3.3.1 Time Domain Specifications . . . . .       | 18          |

|          |  |           |
|----------|--|-----------|
| 3.3.2    | Frequency Domain Specifications . . . . .                          | 19        |
| 3.3.3    | Reference Tracking . . . . .                                       | 19        |
| 3.3.3.1  | Ramp Tracking . . . . .  | 20        |
| 3.3.3.2  | Sine Wave Tracking . . . . .                                       | 20        |
| <b>4</b> | <b>MATLAB-Simulink® Model</b>                                      | <b>21</b> |
| 4.1      | Model step 1: Model Constants . . . . .                            | 21        |
| 4.2      | Non Linear Simulink® Model . . . . .                               | 21        |
| <b>5</b> | <b>Linearisation</b>   | <b>23</b> |
| 5.1      | Small Perturbation . . . . .                                       | 23        |
| 5.2      | Simulink® . . . . .  | 25        |
| <b>6</b> | <b>Linear vs Non-Linear Models Comparison</b>                      | <b>26</b> |
| 6.1      | Non-Linear vs Linear: Step Input Voltage Response . . . . .        | 27        |
| 6.2      | Non-Linear vs Linear: Ramp Input Voltage Response . . . . .        | 27        |
| 6.3      | Non-Linear vs Linear: Sine Input Voltage Response . . . . .        | 28        |
| 6.4      | Non-Linear vs Linear: White Noise Input Voltage Response . . . . . | 28        |
| 6.5      | Non-Linear vs Linear: Impulse Input Voltage Response . . . . .     | 29        |
| 6.6      | Linearisation Assessment . . . . .                                 | 29        |
| <b>7</b> | <b>Uncontrolled System: Stability Analysis</b>                     | <b>30</b> |
| 7.1      | Uncontrolled Analysis: Step Response . . . . .                     | 30        |
| 7.2      | Uncontrolled Analysis: Routh-Hurwitz Stability . . . . .           | 31        |
| 7.3      | Uncontrolled Analysis: Pole-Zero Map . . . . .                     | 31        |

|          |   |           |
|----------|---|-----------|
| 7.4      | Uncontrolled Analysis: Eigenvalues . . . . .                | 32        |
| 7.5      | Uncontrolled System: Nyquist Analysis . . . . .             | 33        |
| 7.6      | Uncontrolled System: Root Locus Analysis . . . . .          | 35        |
| 7.7      | Uncontrolled System: Frequency Analysis . . . . .           | 36        |
| 7.8      | Uncontrolled System Stability Analysis Assessment . . . . . | 36        |
| <b>8</b> | <b>PID Controller Design</b>                                | <b>38</b> |
| 8.1      | Tuning PID with Control Systems Designer . . . . .          | 39        |
| 8.2      | Load Control Systems Designer . . . . .                     | 39        |
| 8.3      | Set Performance Specifications . . . . .                    | 40        |
| 8.4      | Tuning PID . . . . .  | 40        |
| 8.5      | Transforming C(s) . . . . .                                 | 41        |
| 8.6      | P Controller . . . . .                                      | 43        |
| 8.7      | PD Controller . . . . .                                     | 44        |
| 8.8      | PI Controller . . . . .                                     | 46        |
| 8.9      | PID Controller . . . . .                                    | 48        |
| <b>9</b> | <b>Root Locus Design</b>                                    | <b>50</b> |
| 9.1      | Root Locus Design Methodology . . . . .                     | 50        |
| 9.2      | Effect of Poles and Zeros . . . . .                         | 51        |
| 9.3      | Root Locus Tuning . . . . .                                 | 52        |
| 9.4      | Controller Selection . . . . .                              | 54        |
| 9.5      | Candidate Controllers: Step Response Comparison . . . . .   | 56        |
| 9.6      | Candidate Controllers: Ramp Response Comparison . . . . .   | 57        |

|   |           |
|---|-----------|
| 9.7 Candidate Controllers: Sine Response Comparison . . . . .           | 58        |
| 9.8 Controller Selection . . . . .                                      | 59        |
| <b>10 Analysis of Controlled System</b>                                 | <b>61</b> |
| 10.1 Time Domain Response . . . . .                                     | 61        |
| 10.2 Roots and Pole Analysis . . . . .                                  | 62        |
| 10.3 Frequency Domain Analysis . . . . .                                | 63        |
| <b>11 Performance Specifications</b>                                    | <b>64</b> |
| 11.1 Instrumentation Needed for Implementation . . . . .                | 64        |
| 11.1.1 Position Sensor / Encoder . . . . .                              | 64        |
| 11.1.2 Light Sensor . . . . .   | 65        |
| 11.1.3 Microcontroller . . . . .  | 66        |
| 11.2 Strategies for Managing Unmeasurable States . . . . .              | 67        |
| 11.2.1 Kalman Filter . . . . .  | 67        |
| 11.2.2 Extended Kalman Filter . . . . .                                 | 67        |
| <b>12 Discussion</b>  | <b>68</b> |
| 12.1 Impact of Model Assumptions on Solar Tracker Performance . . . . . | 71        |
| <b>13 Conclusion</b>  | <b>73</b> |
| <b>References</b>   | <b>74</b> |
| <b>14 Appendix</b>  | <b>76</b> |
| 14.1 Model Constants . . . . .  | 76        |

|                                    |    |
|------------------------------------|----|
| 14.2 Compare Linear and Non-Linear | 77 |
| 14.3 Controller Analysis           | 78 |
| 14.4 Disturbance Plot              | 83 |
| 14.5 Linear Analysis               | 84 |
| 14.6 PID Tuning                    | 86 |
| 14.7 Running Animations            | 87 |
| 14.8 Linearisation Output          | 88 |
| 14.8.1 State Space                 | 88 |
| 14.8.2 Transfer Function           | 89 |
| 14.8.3 Zero-Pole Gain              | 89 |

## List of Figures

|    |   |    |
|----|---|----|
| 1  | Orientating Solar Tracker [3]                     | 1  |
| 2  | Orientating Solar Tracker [4]                     | 2  |
| 3  | Problem Breakdown                                 | 3  |
| 4  | Assumed Control Scheme.                           | 3  |
| 5  | Methodology Flow                                  | 5  |
| 6  | Solar Used to Power Borehole Pump [3] [6]         | 7  |
| 7  | 425 W Mono Panel [8]                              | 9  |
| 8  | Side View of the Solar Tracker Assembly           | 11 |
| 9  | Motor-Solar Tracker Free Body Diagram             | 11 |
| 10 | 3D Model of the Solar Tracker                     | 12 |
| 11 | Solar Tracker 3-View                              | 13 |
| 12 | Electro-Mechanical System [9]                     | 14 |
| 13 | Disturbance Force Modelling                       | 15 |
| 14 | Beaufort Scale[11]                                | 16 |
| 15 | Disturbance Moment as a Function of Wind Speed    | 16 |
| 16 | Solar Tracker Path [12]                           | 19 |
| 17 | R(s) Modelled as Sine and Ramp Signals            | 20 |
| 18 | Non-Linear Model                                  | 22 |
| 19 | Linearisation Process on Simulink <sup>®</sup>    | 25 |
| 20 | Non-linear and Linearised Models                  | 26 |
| 21 | Non-Linear vs Linear: Step Input Voltage Response | 27 |

|    |  |    |
|----|--|----|
| 22 | Non-Linear vs Linear: Ramp Input Voltage Response . . . . .    | 27 |
| 23 | Non-Linear vs Linear: Ramp Input Voltage Response . . . . .    | 28 |
| 24 | Non-Linear vs Linear: Ramp Input Voltage Response . . . . .    | 28 |
| 25 | Non-Linear vs Linear: Impulse Input Voltage Response . . . . . | 29 |
| 26 | Uncontrolled Analysis: Step Response . . . . .                 | 30 |
| 27 | Uncontrolled Analysis: Pole-Zero Map . . . . .                 | 32 |
| 28 | Uncontrolled System: Nyquist Analysis . . . . .                | 33 |
| 29 | Uncontrolled System: Root Locus Analysis . . . . .             | 35 |
| 30 | Uncontrolled System: Frequency Analysis . . . . .              | 36 |
| 31 | PID Controller on a System Block Diagram [13] . . . . .        | 38 |
| 32 | Control System Designer Interface . . . . .                    | 39 |
| 33 | Setting Desired Performance Specifications . . . . .           | 40 |
| 34 | Successfully Tuned Example . . . . .                           | 41 |
| 35 | Step Response Plot: P Controller . . . . .                     | 43 |
| 36 | Step Response Results: P Controller . . . . .                  | 43 |
| 37 | Step Response Plot: PD Controller . . . . .                    | 44 |
| 38 | Step Response Results: PD Controller . . . . .                 | 44 |
| 39 | Step Response Plot: PI Controller . . . . .                    | 46 |
| 40 | Step Response Results: PI Controller . . . . .                 | 46 |
| 41 | Step Response Plot: PID Controller . . . . .                   | 48 |
| 42 | Step Response Results: PID Controller . . . . .                | 48 |
| 43 | Root Locus Editor . . . . .                                    | 50 |

|    |  |    |
|----|--|----|
| 44 | Output Response for the Root Locus . . . . .                       | 52 |
| 45 | Step Response Plot: Root Locus . . . . .                           | 52 |
| 46 | Step Response Results: Root Locus . . . . .                        | 53 |
| 47 | Root Locus Plot of Root Locus Controller System . . . . .          | 53 |
| 48 | Simulink® Controller Comparison Setup . . . . .                    | 54 |
| 49 | Candidates: Step without Disturbance . . . . .                     | 56 |
| 50 | Candidates: Step with Disturbance . . . . .                        | 56 |
| 51 | Candidates: Ramp without Disturbance . . . . .                     | 57 |
| 52 | Candidates: Ramp with Disturbance . . . . .                        | 57 |
| 53 | Candidates: Sine without Disturbance . . . . .                     | 58 |
| 54 | Candidates: Sine without Disturbance at Higher Frequency . . . . . | 58 |
| 55 | Candidates: Sine with Disturbance at Higher Frequency . . . . .    | 59 |
| 56 | Controller Selection . . . . .                                     | 60 |
| 57 | Controlled System Analysis: Step Response . . . . .                | 61 |
| 58 | Controlled System Analysis: Pole-Zero Map . . . . .                | 62 |
| 59 | Controlled System Analysis: Bode Plot . . . . .                    | 63 |
| 60 | AS5048A High-Resolution Position Sensor [14] . . . . .             | 65 |
| 61 | BPW21R Silicon Photodiode [15] . . . . .                           | 65 |
| 62 | Arduino Mega 2560 Rev3 Microcontroller [16] . . . . .              | 66 |
| 63 | Animations Setup . . . . .   | 87 |

## List of Tables

|    |  |    |
|----|--|----|
| 1  | Parameter Values at Den Hagen Guest Farm [7] . . . . .                       | 8  |
| 2  | Solar Panel Specifications [8] . . . . .                                     | 8  |
| 3  | Inverter Specifications . . . . .  | 9  |
| 4  | Battery Specifications . . . . .   | 9  |
| 5  | Time Domain Specifications for a Solar Tracker . . . . .                     | 18 |
| 6  | Frequency Domain Specifications for a Solar Tracker . . . . .                | 19 |
| 7  | Comparing Uncontrolled Plant with Desired Performance . . . . .              | 30 |
| 8  | Routh-Hurwitz Stability Analysis . . . . .                                   | 31 |
| 9  | Effect of adding a Zero or a Pole . . . . .                                  | 51 |
| 10 | Practical Considerations in Controller Design and System Response Tuning . . | 51 |
| 11 | Step Response of Controlled System . . . . .                                 | 61 |
| 12 | Frequency Response . . . . .   | 63 |
| 13 | Summary of Performance Specification . . . . .                               | 64 |
| 14 | Position sensor specifications [14] . . . . .                                | 65 |
| 15 | Light Sensor Specifications [15] . . . . .                                   | 66 |
| 16 | Light Sensor Specifications [16] . . . . .                                   | 66 |
| 17 | Comparing candidate controller performance to desired performance . . . . .  | 69 |
| 18 | Performance Summary using Root Locus Controller . . . . .                    | 71 |
| 19 | Impact of Model Assumptions on System Performance . . . . .                  | 72 |

## Nomenclature

|                 |   |
|-----------------|---|
| $\beta$         | Motor angular velocity ( $rad/s$ )  |
| $\ddot{\theta}$ | Angular acceleration ( $rad/s^2$ )  |
| $\Delta\theta$  | Small perturbation of angular velocity about equilibrium point ( $rad/s$ )  |
| $\Delta I_a$    | Small perturbation of motor electric current about equilibrium point ( $A$ )  |
| $\Delta V_a$    | Small perturbation of input voltage about equilibrium point ( $V$ )   |
| $\dot{\theta}$  | Angular velocity ( $rad/s$ )  |
| $\epsilon_a$    | Back Electro-Motive Force ( $V$ )   |
| $\eta$          | Solar module efficiency (%)   |
| $\rho$          | Density of air ( $kg/m^3$ )   |
| $\theta$        | Elevation angle ( $rad$ )   |
| $\theta_0$      | Elevation angle at equilibrium point ( $rad$ )  |
| $A_s$           | Area of solar panels ( $m^2$ )  |
| $c$             | Damping constant ( $kg/s$ )   |
| $d$             | Distance between the centre of mass and the shaft centre ( $m$ )  |
| $EMF$           | Electro-Motive Force ( $V$ )  |
| $F_B$           | Wind force calculated using dynamic pressure on the Beaufort scale ( $N$ )  |
| $F_W$           | Component of wind force acting on the solar tracker calculated using dynamic pressure on the Beaufort scale ( $N$ ) |
| $g$             | Acceleration due to gravity ( $m/s^2$ )   |
| $G(s)$          | Transfer function   |
| $I_a$           | Motor electric current ( $A$ )  |
| $I_{a0}$        | Motor electric current at equilibrium point ( $A$ )   |
| $J$             | Moment of inertia ( $kg \cdot m^2$ )  |
| $K_d$           | Derivative constant   |
| $k_e$           | Back Electro-Motive Force constant ( $N \cdot m/\sqrt{W}$ )   |

|               |   |
|---------------|---|
| $K_i$         | Integral constant   |
| $K_p$         | Proportional constant   |
| $k_t$         | Motor torque constant ( $N \cdot m/A$ )                               |
| $k_v$         | Motor velocity constant ( $RPM$ )                                     |
| $L_a$         | Motor inductance ( $H$ )  |
| $m$           | Mass ( $kg$ )   |
| $M_d$         | Moment due to the disturbance wind force ( $N.m$ )                    |
| $P$           | Power ( $kWh$ )   |
| $q$           | Distance between centre of shaft and point of action of $F_W$ ( $m$ ) |
| $q_b$         | Dynamic pressure ( $Pa$ )   |
| $R(s)_{ramp}$ | Ramp reference path   |
| $R(s)_{sine}$ | Sine wave reference path  |
| $R_a$         | Motor resistance ( $\Omega$ )   |
| $t$           | Time ( $s$ )  |
| $T_M$         | Torque of motor ( $N.m$ )   |
| $T_r$         | Mechanical transmission ratio   |
| $u$           | Wind speed ( $m/s$ )  |
| $V_a$         | Input voltage ( $V$ )   |
| $V_{a_0}$     | Input voltage at equilibrium point ( $V$ )                            |

# 1 Introduction

## 1.1 Background

The increase in local and international tourists in South Africa has given hotels and inns a thriving chance in business [1]. However, some of the businesses are in regions that are heavily stricken with power shortages, in addition to the current national load-shedding. For example, the Kam'Bati River Resort in Swellendam, which has self-catering cabins, has faced electricity challenges and has resorted to using solar power, even for pumping water. [2]

## 1.2 Existing Solutions

The use of dynamically orientating solar panels, shown in Figure 1, is a common practice that can improve solar power production by up to 40%. [3]



Figure 1: Orientating Solar Tracker [3]

Solar trackers can have multiple degrees of freedom; however, a single-axis orienting solar panel is a sufficient way of tracking the sun and improving the power generation of the system. A study by Lassio et al. [4] highlighted the significance of single-axis trackers, shown in Figure 2.

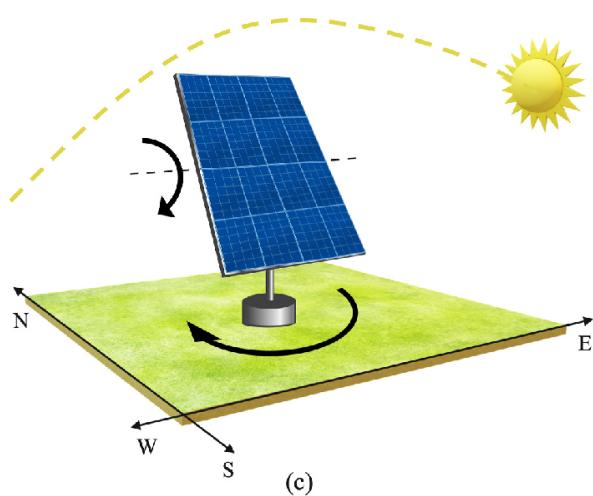
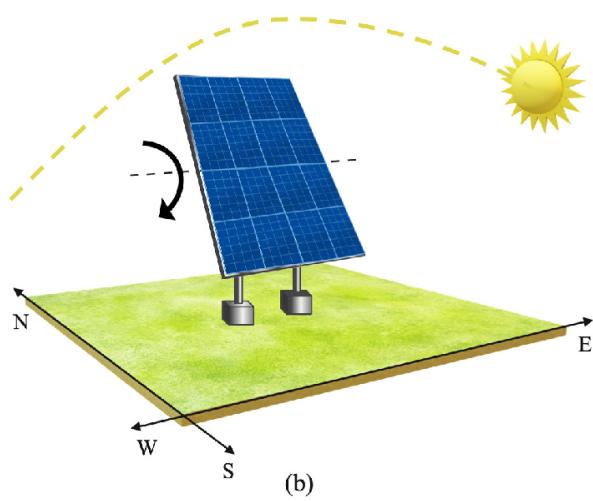
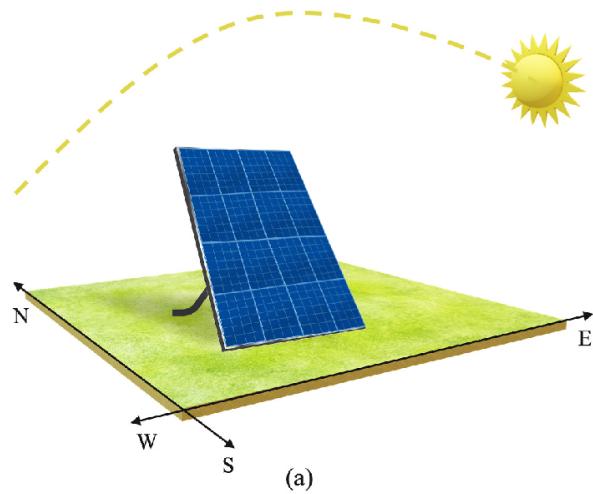


Figure 2: Orientating Solar Tracker [4]

### 1.3 Problem Breakdown

Generally, the solar tracker is controlled by a motor that provides torque and orients it at a desired angle for maximum exposure to the Sun. From the breakdown of the problem on Figure 3, the problem being focused on is a control system problem.

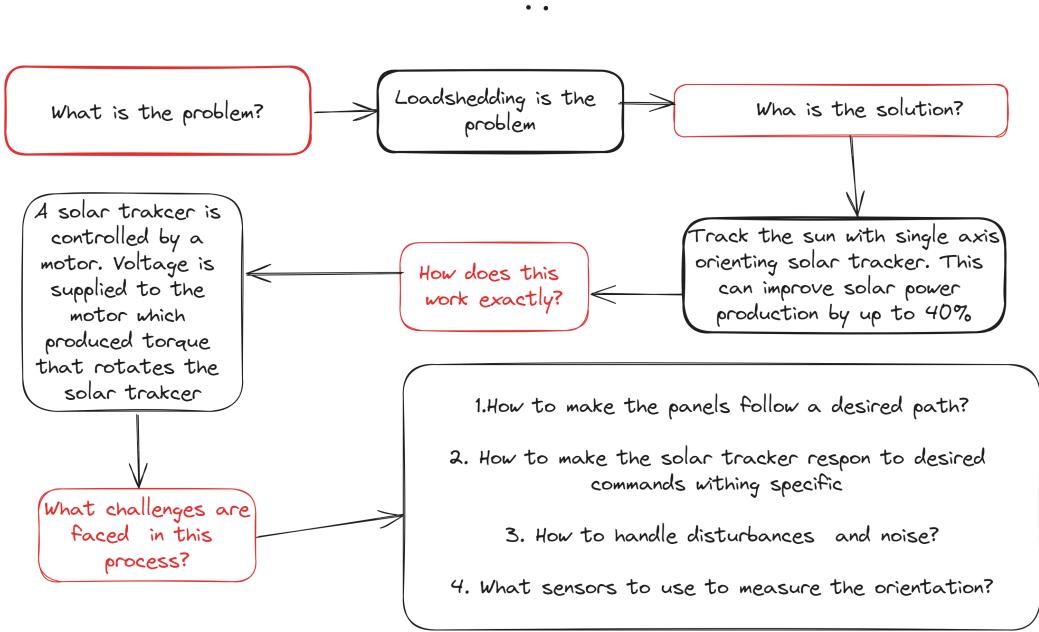


Figure 3: Problem Breakdown

The program then controls the solar tracker with the use of a controller, sensors and a motor. A typical operation of the solar tracker is that a user gives the desired orientation to a computer program as on Figure 4.

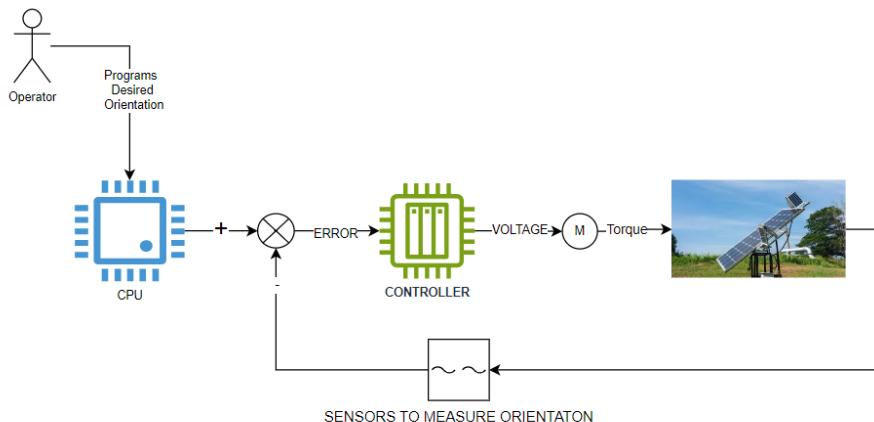


Figure 4: Assumed Control Scheme.

The general flow of the system is as follows:

1. User sets the desired angle.
2. The CPU takes the desired and actual angles measured by the sensor/s and computes the error.
3. The controller takes the error and computes the voltage supply to the motor.
4. The voltage based on the error and controller operations turns the motor, producing a torque.
5. Usually, some gears amplify or reduce the torque.
6. The torque from the motor then rotates the solar tracker.
7. The process is repeated until the desired angle is reached within some set tolerance.
8. The motor can then either be turned off automatically, and the solar trackers are locked in place automatically by some "smart" mechanism, or the motor can keep running to hold the panels in place, which is however, inefficient due to power usage.

The problem to solve is the design of the control system focusing on the controller. Some assumptions that can be made are:

- The mechanisms that keep the solar panels in the desired position are "black-boxed", i.e. the internal complexity is not examined or taken into account.
- The operations of the computer are "black boxed".
- The sensors can be assumed to have "perfect" feedback. The sensor measurements are treated as completely accurate and error-free for the sake of simplicity. However, in reality, sensors are subject to imperfections such as noise and other inaccuracies that can affect their readings.
- The control schemes are a PID (Proportional Integral Derivative) and a Root Locus compensator.

## 1.4 Methodology

- The controller is to be designed using MATLAB and Simulink<sup>®</sup>.
- To achieve this, a mathematical model is required.

- To obtain a mathematical model, a physical model is required.
- To obtain a physical model, a realistic system is required.

The controller design is done on a linearized system. The reason is linearised systems are simpler and more predictable, and a lot of control theory exists for linearised systems [5]. Hence, the linearised system ought to behave in the same manner as its non-linear equivalent such that the controller can be transferred to the non-linear system. The methodology flow is given on Figure 5. A real problem is essentially reduced to a physical model with some simplifications to represent the real-world problem whilst minimising complexity. In doing so, the choice of assumptions is important in achieving the goal. In this case, the real-world problem is a solar tracker for a guest house or resort that uses solar energy to pump water, i.e. Kam'Bati River Resort.

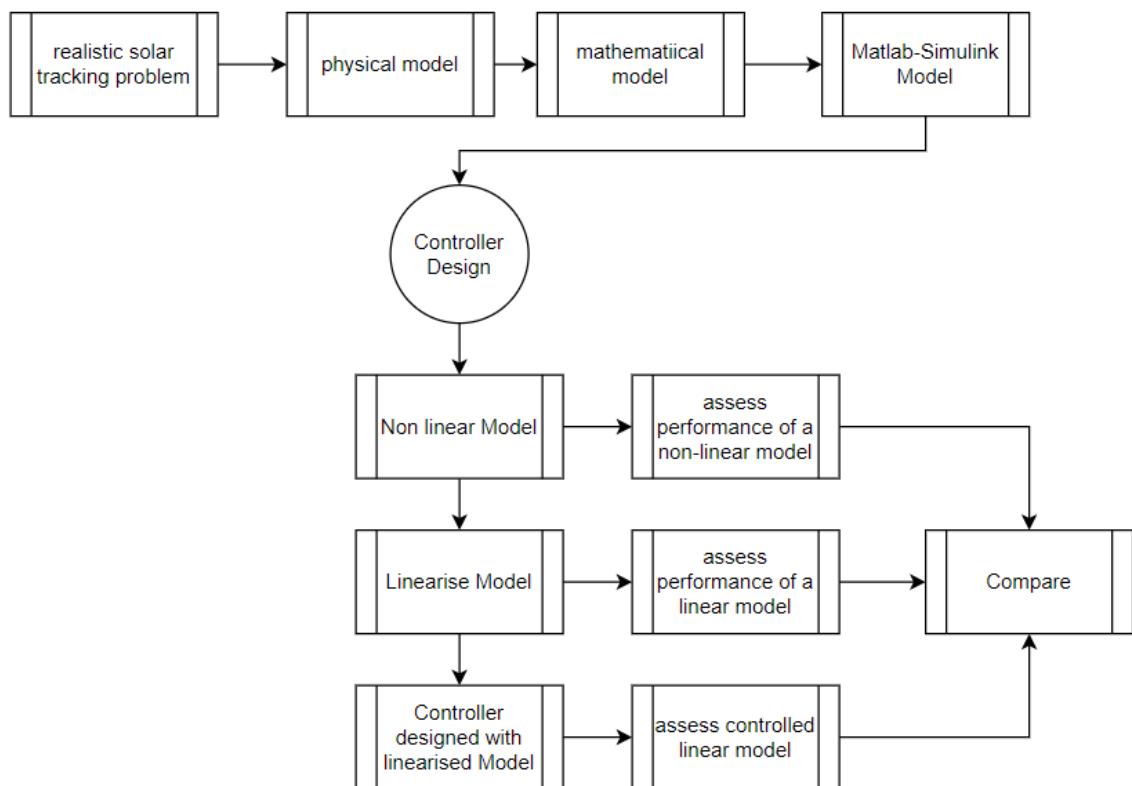


Figure 5: Methodology Flow

## 1.5 Objectives

1. Develop a physical model of a solar tracker for a solar water pumping system.
2. From the physical model, develop a mathematical model of the system.

3. Develop a MATLAB-Simulink<sup>®</sup> model of the mathematical model.
4. Linearise the system.
5. Compare the linear and non-linear systems for a range of realistic input signals.
6. Conduct stability analysis of the uncontrolled linear and non-linear systems.
7. Develop a PID controller for the system.
8. Develop a compensator using the root locus method.
9. Conduct a full stability analysis of the controlled system.
10. Assess the performance of the controller, paying attention to disturbances and reference tracking.
11. Create an animation of the controlled system.
12. Discuss the results paying attention to expected results from published data/literature and assumptions made.

## 1.6 Problem Description

The solar application is for either the Den Hagen Guest Farm guest house or the Kam'Bati River Resort, refer to Table 1. A 250W LARS borehole pump, as shown on Figure 6, will be powered by solar energy. Assuming a 24-hour non-stop operation, the daily power consumption is given by equation 1:

$$P = \text{Watts} \times \text{time} \quad (1)$$

$$P = 6 \text{ kWh} \quad (2)$$

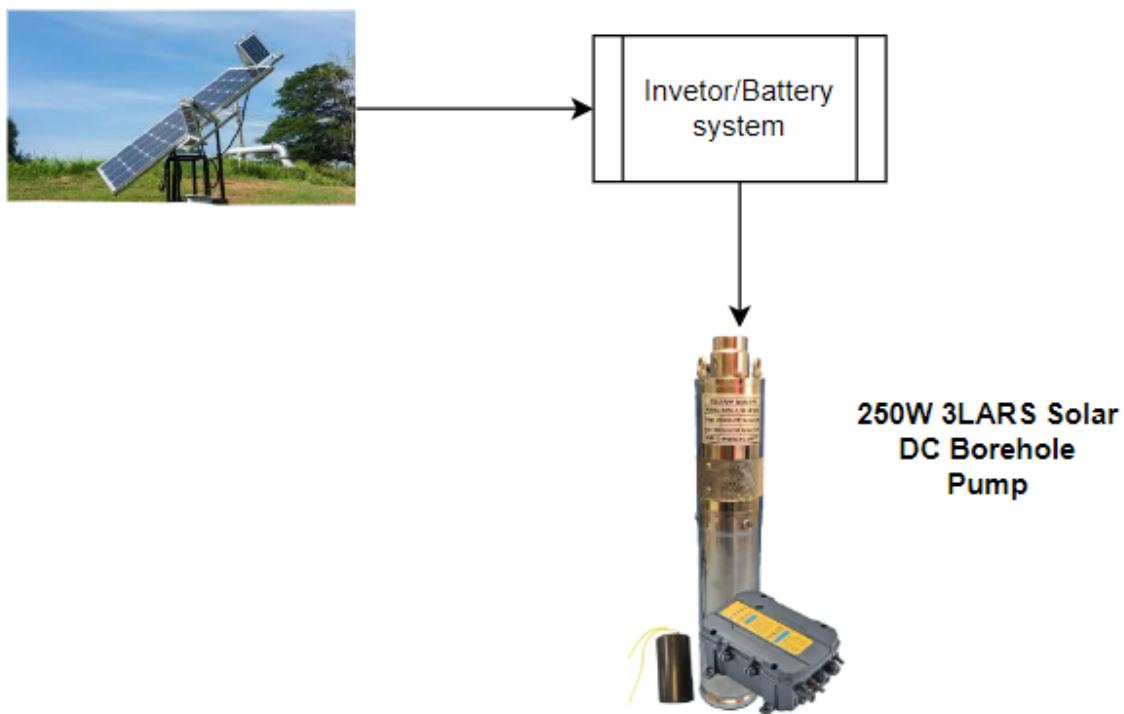


Figure 6: Solar Used to Power Borehole Pump [3] [6]

With a 3kWh inverter, two 425 W mono panels can supply the required daily energy, refer to Figure 7. The solar panel specifications are shown on Table 2.

Table 1: Parameter Values at Den Hagen Guest Farm [7]

| Parameter                      | Value               |
|--------------------------------|---------------------|
| Latitude                       | -30.743507774746487 |
| Longitude                      | 27.956952851603862  |
| Elevation above sea level      | 1935 m              |
| Average hours of sun in Summer | 1281.576            |
| Average temperature in Summer  | 14.13 °C            |
| Average hours of sun in Winter | 1703.472            |
| Average temperature in Winter  | 5.07 °C             |

Table 2: Solar Panel Specifications [8]

| Parameter   | Value                     |
|---|---------------------------|
| Number of mono solar panel modules                | 2                         |
| Peak power  | 425 W                     |
| Weight of modules                                 | 43.6 kg                   |
| Dimensions (L × W × H)                            | 1762 mm × 1134 mm × 30 mm |
| Operating temperature range                       | -40 °C to 85 °C           |
| Nominal operating cell temperature                | 43 ± 2 °C                 |
| Module efficiency ( $\eta$ )                      | 21.3%                     |
| Maximum power voltage                             | 39.5 V                    |
| Maximum power current                             | 8.13 A                    |
| Temperature coefficient for peak power            | -0.34% / °C               |
| Temperature coefficient for open circuit voltage  | -0.25% / °C               |
| Temperature coefficient for short circuit current | 0.04% / °C                |
| Number of inverters                               | 1                         |
| Inverter energy rating                            | 3 kW                      |
| Number of cells per module                        | 144                       |

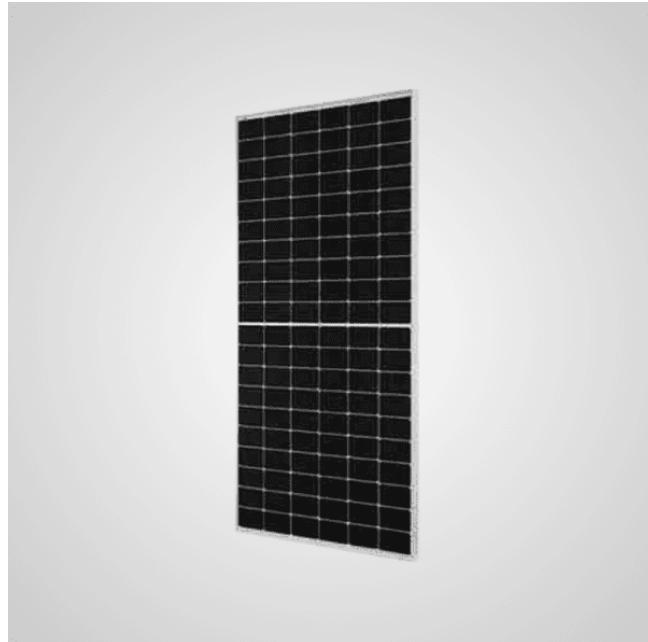


Figure 7: 425 W Mono Panel [8]

The inventor specifications are shown on Table 3. The battery specifications are on Table 4.

Table 3: Inverter Specifications

| Parameter                | Value   |
|--------------------------|---|
| Inverter type            | High frequency hybrid solar inverter with pure sine wave output |
| Grid connection          | Off-grid  |
| Peak efficiency          | 93%   |
| Nominal AC voltage Input | 230 V   |
| Rated power              | 3000 W  |
| Surge power              | 6000 W  |
| Nominal frequency Range  | 50 Hz / 60 Hz (auto sensing)                                    |
| Mass                     | 7.8 kg  |
| Dimensions (L × W × H)   | 290 mm × 342 mm × 125 mm  |

Table 4: Battery Specifications

| Parameter                  | Value   |
|----------------------------|---------|
| Nominal voltage            | 25.6 V  |
| Nominal capacity           | 100 Ah  |
| Nominal energy             | 2560 Wh |
| Maximum continuous current | 100 A   |

## 2 Physical Model

To simplify the mathematical model of the systems, several assumptions were made about the system. These include, but are not limited to:

- Motion tracking occurs along a single axis, simplifying both control algorithm and mechanical design complexities.
- Energy storage and discharge in batteries or capacitors are assumed to occur instantaneously, disregarding dynamics like charge/discharge rates and energy losses.
- The electrical network connected to the solar panels is assumed to behave as an ideal voltage or current source, neglecting impedance effects and voltage fluctuations.
- The panel motion range within the solar tracking system is limited to the period of the day, excluding astronomical twilight (when the sun is at a position between  $0^\circ$  and  $18^\circ$  elevation). This assumption is based on the premise that sufficient solar energy capture is feasible during the remaining daylight hours to meet the household's energy requirements.
- The mechanical components can be treated as rigid bodies.
- The moment of inertia of the panels is assumed to remain constant as the panel follows the position of the sun
- There is a damping coefficient of value  $c$  between the motor and solar tracker.
- Rotational stiffness is negligible.
- The disturbance force acts at a point located at a distance  $q$  from the centre of the motor shaft.

Figure 8 provides a side-view of the sun-tracking panel configuration, offering a detailed illustration of its constituent components and their integration. The solar panels are securely positioned atop the mounting bracket, which serves as a secure foundation for their support. Affixed to the mounting bracket via a flange, the DC motor serves as the mechanism responsible for panel orientation adjustments. The torque applied by the DC motor is transmitted to the panel through the intermediary of the flange, ensuring effective control over its positioning.

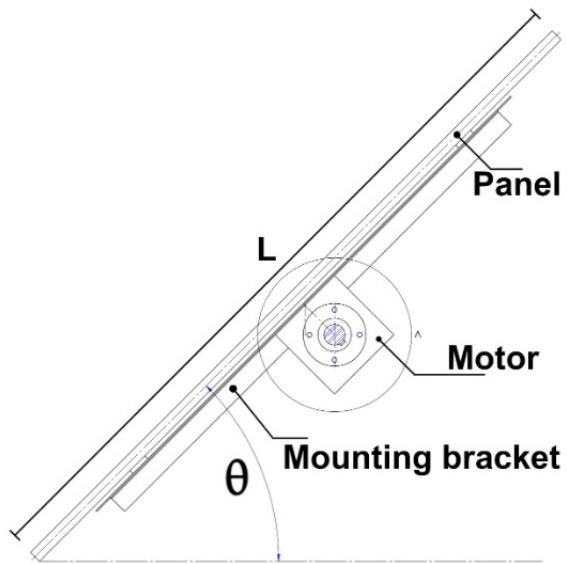


Figure 8: Side View of the Solar Tracker Assembly

Figure 9 shows the force and torque diagram of the assembly. The torque  $T_M$  originates from the motor, which opposes the moment induced by the weight of the assembly and the rotational mechanical damping. The rotation angle  $\theta$  is physically constrained from  $0^\circ$  to  $135^\circ$ , refer to the physical model on Figure 10.

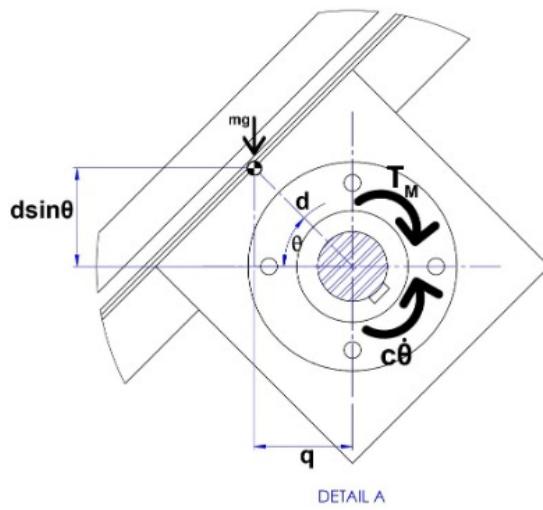


Figure 9: Motor-Solar Tracker Free Body Diagram

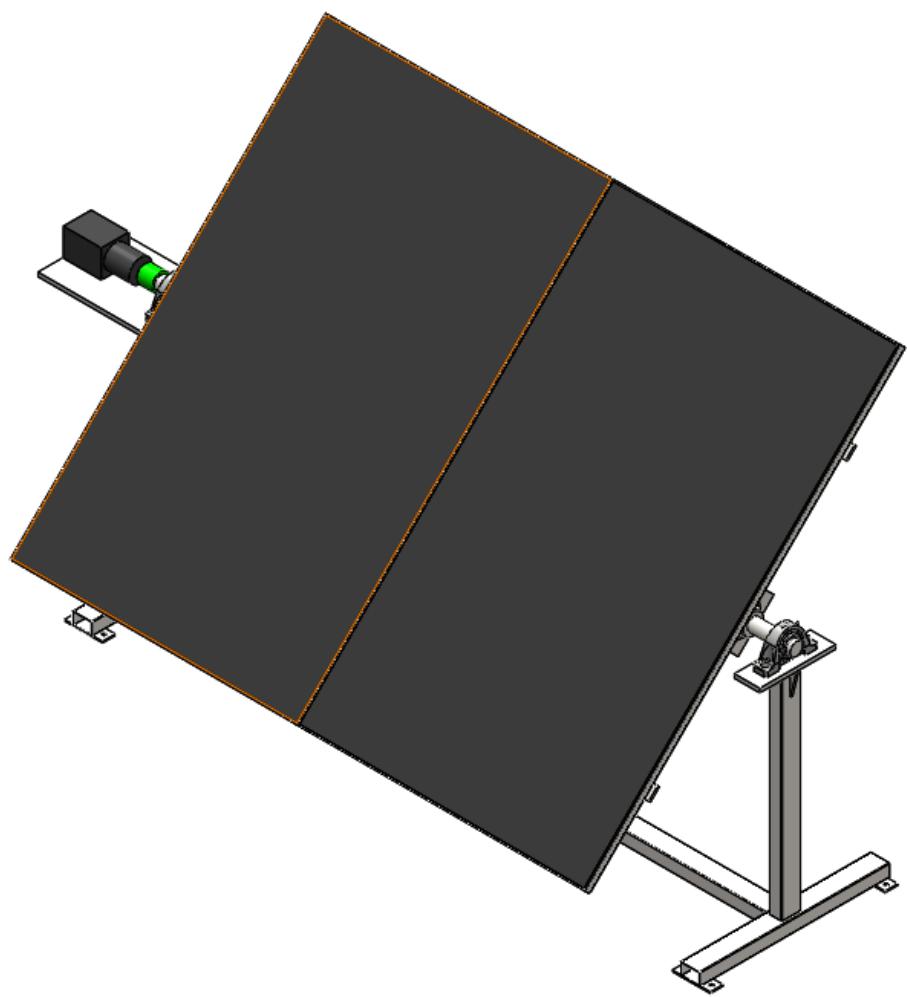


Figure 10: 3D Model of the Solar Tracker

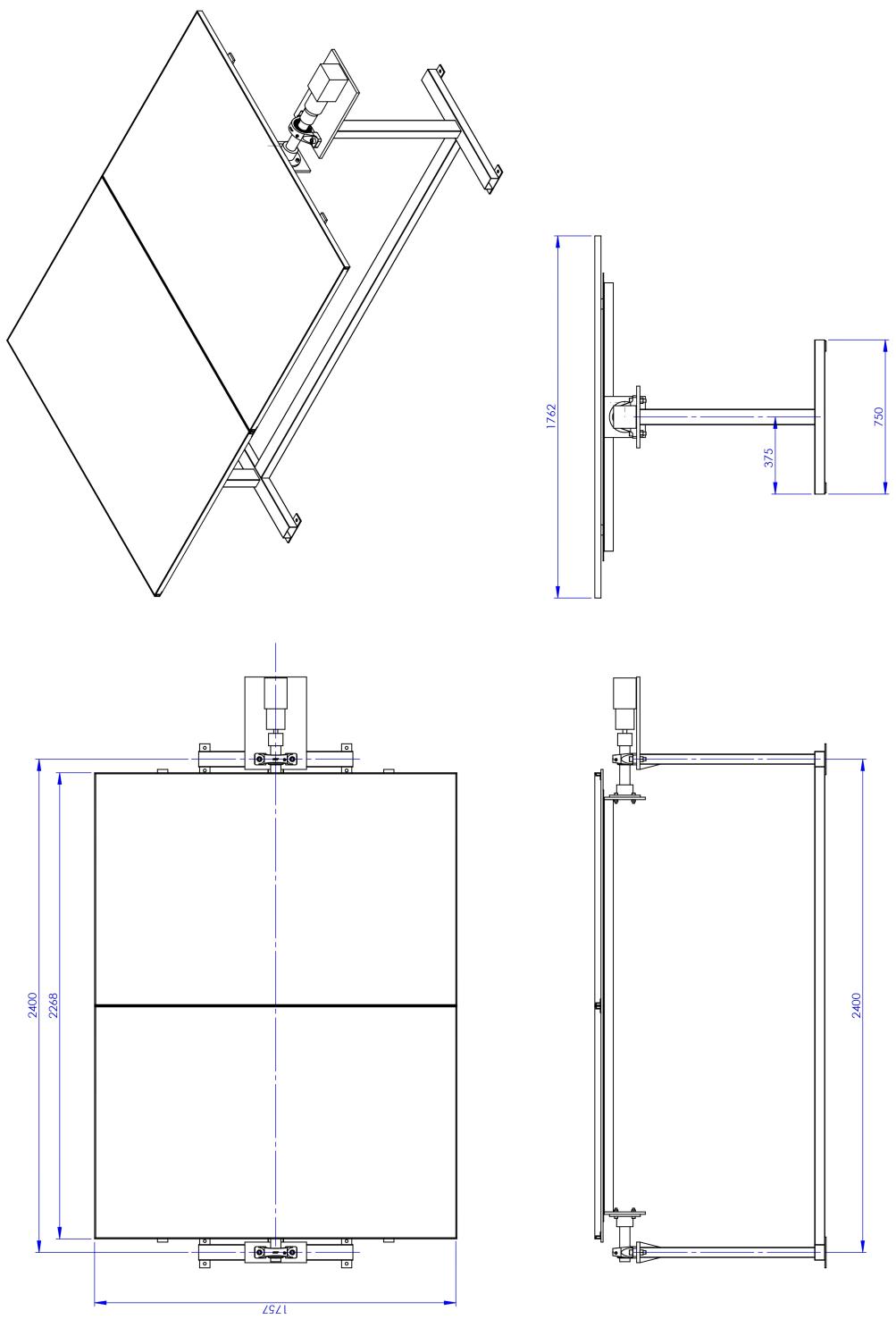


Figure 11: Solar Tracker 3-View

### 3 Mathematical Model

#### 3.1 Modelling System

Applying Newton's second law for rotation about the centre of the motor's shaft, from Figure 9, results in equation 3.

$$J\ddot{\theta} = T_M - mgd\cos\theta - c\dot{\theta} \quad (3)$$

Considering Kirchhoff's Voltage Law for the electro-mechanical system, shown on Figure 12, results in equation 4. The back electromotive force is given by equation 5 and the motor torque by equation 6.

$$V_a = RI_a + L\frac{dI_a}{dt} + \epsilon_a \quad (4)$$

$$\epsilon_a = k_e\dot{\beta} \quad (5)$$

$$T_M = k_t I_a \quad (6)$$

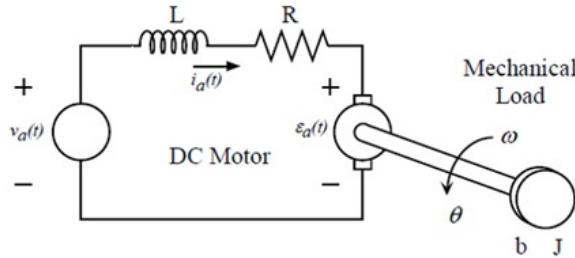


Figure 12: Electro-Mechanical System [9]

#### 3.2 Modelling Wind Disturbance

The wind is modelled using the concept of dynamic pressure [10] given by equation 7.

$$q_p = \frac{1}{2}\rho u^2 \quad (7)$$

The wind force is assumed to act at a distance q from the pivot point, as shown on Figure 13. The force is calculated using equation 8.

$$F_W = q_p A_s \sin\theta \quad (8)$$

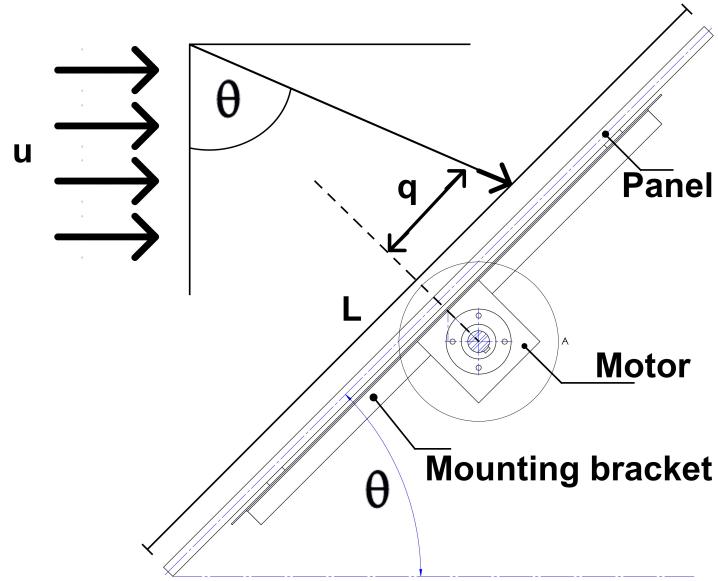


Figure 13: Disturbance Force Modelling

The moment due to the disturbance force is given by equation 9.

$$M_d = F_W q \quad (9)$$

$$M_d = \frac{1}{2} \rho u^2 A_s \sin \theta \quad (10)$$

The disturbance in the time domain is :

$$M_d = \frac{1}{2} \rho u^2 A_s \sin(\theta(t))$$

Laplace transform of the original equation:

$$\mathcal{L}\{M_d\} = \frac{1}{2} \rho u^2 A_s \mathcal{L}\{\sin \theta\}$$

Laplace transform of  $\sin \theta$ :

$$\begin{aligned} \mathcal{L}\{\sin \theta\} &= \int_0^\infty \sin(\theta) e^{-st} dt \\ \mathcal{L}\{\sin(\theta)\} &= \frac{1}{s^2 + 1} \end{aligned}$$

$\therefore$  The disturbance transfer function is given by

$$T_d(s) = \frac{\rho u^2 A_s}{(s^2 + 1)} \quad (11)$$

The wind speed,  $u$  in  $m/s$ , varies according to the Beaufort scale shown on Figure 14. The strength varies between 1-12 with wind speeds from  $1.6 \text{ km/hr}$  to more than  $117 \text{ km/hr}$ .

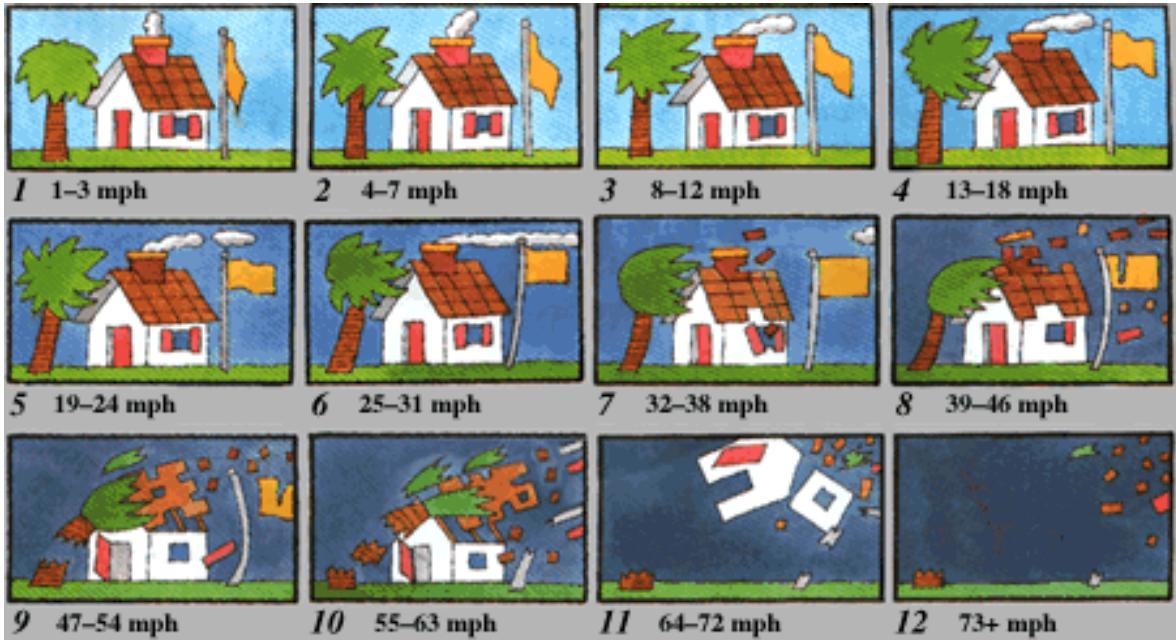


Figure 14: Beaufort Scale[11]

The disturbance moment is highest when the panels are at  $90^\circ$ . The conditions assumed are ISA sea level with air density of  $1.225 \text{ kg/m}^3$ . The disturbance moment has a parabolic profile as shown on Figure 15.

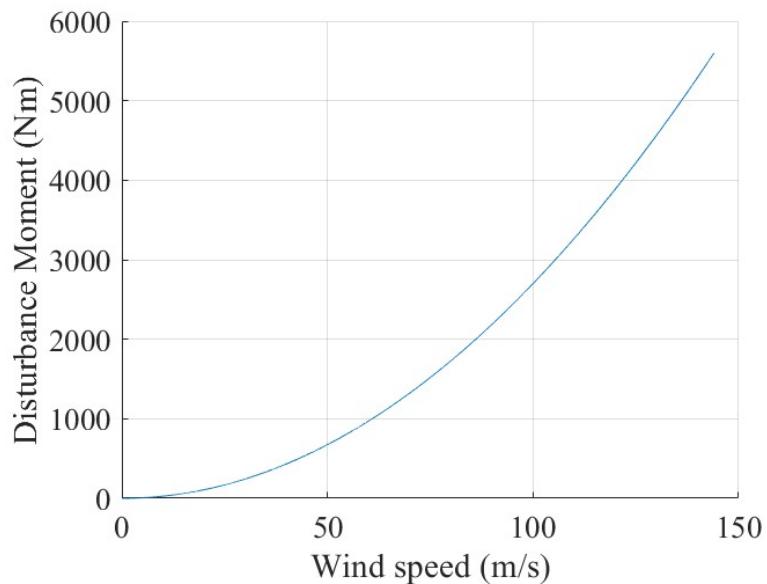


Figure 15: Disturbance Moment as a Function of Wind Speed

The set of non-linear coupled differential equations describing the system is:

#### 3.2.0.1 Solar Tracker Dynamics

$$\ddot{\theta} = \frac{T_M - mgd\cos\theta - c\dot{\theta}}{J} \quad (12)$$

Where  $\theta = [0^\circ, 135^\circ]$

#### 3.2.0.2 Motor Dynamics

$$\frac{dI_a}{dt} = \frac{V_a - RI_a - \epsilon_a}{L} \quad (13)$$

$$\epsilon_a = k_e \dot{\beta} \quad (14)$$

$$T_M = k_t I_a \quad (15)$$

#### 3.2.0.3 Wind Disturbance

$$M_d = \frac{1}{2} \rho u^2 A_s \sin \theta(t)$$

### 3.3 Desired Performance Specifications

The specifications are categorized as:

- Step response, i.e. the solar tracker should be able to respond to a step command within a specified set of time domain performance specifications, refer to Table 5.
- Frequency response requirements as outlined on Table 6.
- Reference tracking, i.e. follow a specified set of angles with time. For example, it must follow the position of the sun at a rate of change of  $x^\circ/\text{hr}$  from sunrise to sunset.
- The specifications are based on design prerogatives, standard control systems principles and typical operations of a solar tracker.
- The system must have good disturbance rejection, i.e. this is quantified by the magnitude of the wind disturbances it can reject, as outlined on Figure 15 .

### 3.3.1 Time Domain Specifications

Table 5: Time Domain Specifications for a Solar Tracker

| Specification                                | Value        | Comment/Justification   |
|--|--------------|---|
| Rise time (10% to 90%)                       | 2 seconds    | Ensures the tracker quickly responds to changes in sunlight position.             |
| Settling time (within 2% of the final value) | 5 seconds    | Ensures stability and quick stabilization after a disturbance or position change. |
| Steady-state error                           | $\leq 0.5\%$ | Ensures high sun tracking accuracy to maximize energy generation.                 |
| Percent overshoot                            | 10%          | Balances between responsiveness and avoiding excessive oscillations.              |
| Delay time                                   | 1 second     | Measures the time taken to respond to sunlight changes initially.                 |
| Peak time                                    | 3 seconds    | Ensures the system quickly reaches its first peak after a step input.             |
| Tracking accuracy                            | $1^\circ$    | Ensuring the solar panels are aligned closely with the position of the sun.       |
| Response time                                | 3 seconds    | Measures overall responsiveness from input change to output action.               |

### 3.3.2 Frequency Domain Specifications

Table 6: Frequency Domain Specifications for a Solar Tracker

| Specification    | Value           | Comment/Justification   |
|------------------|-----------------|---|
| Bandwidth        | 0.1 Hz          | Ensures the system can respond to changes in sunlight position without excessive delay. |
| Phase margin     | $\geq 45^\circ$ | Ensures adequate stability and damping to prevent oscillations.                         |
| Gain margin      | $\geq 10$ dB    | Provides a safety margin to maintain system stability under varying conditions.         |
| Resonant peak    | $\leq 1.5$      | Limits the amplification of disturbances at the resonant frequency.                     |
| Cutoff frequency | 0.05 Hz         | Defines the frequency beyond which the response of the system significantly attenuates. |
| Damping ratio    | 0.7             | Balances between fast response and minimal overshoot.                                   |

### 3.3.3 Reference Tracking

The solar tracker must follow some path as the position of the sun changes with time, i.e. follow some reference as illustrated on Figure 16.

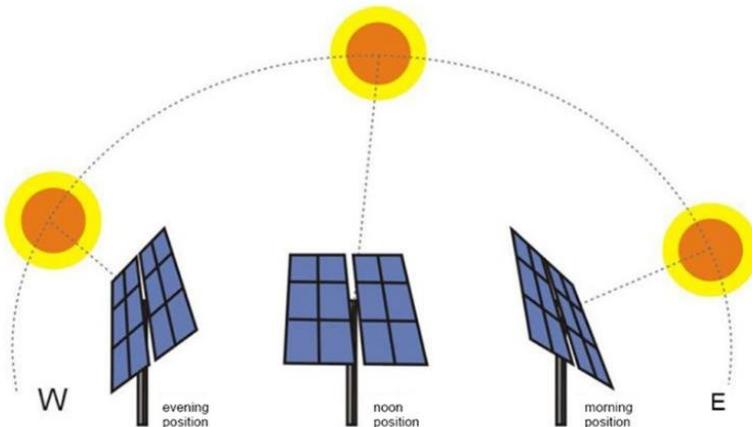


Figure 16: Solar Tracker Path [12]

This can be modelled as either a ramp or a sine wave. These two reference blocks in Simulink<sup>®</sup> can represent a typical path the tracker can take during the day to maximise solar energy generation.

### 3.3.3.1 Ramp Tracking

Assuming a 12-hour day operation and an azimuth angle of  $-90^\circ$  to  $90^\circ$  at sunset, the reference can be modelled as a linear function  $R(s)$  by equation 16.

$$R(s)_{ramp} = \frac{\pi}{12 \times 3600} t - \frac{\pi}{2} \quad (16)$$

This translates to moving at a rate of  $\frac{\pi}{(12 \times 3600)}$  radians per second with an offset of  $-90^\circ$ .

### 3.3.3.2 Sine Wave Tracking

The tracker can also follow a frequency input, i.e. a sinusoidal path with the general equation 17.

$$R(s)_{sine} = A_m \sin(\alpha t) - \frac{\pi}{2} \quad (17)$$

When  $t = 6$  hrs (43 200 seconds), i.e. mid-day, the angle is  $0^\circ$ .  $A_m = \frac{\pi}{2}$  and  $\alpha = \frac{\pi}{(12 \times 3600)}$ , introducing a negative sign at midday as shown on Figure 17.

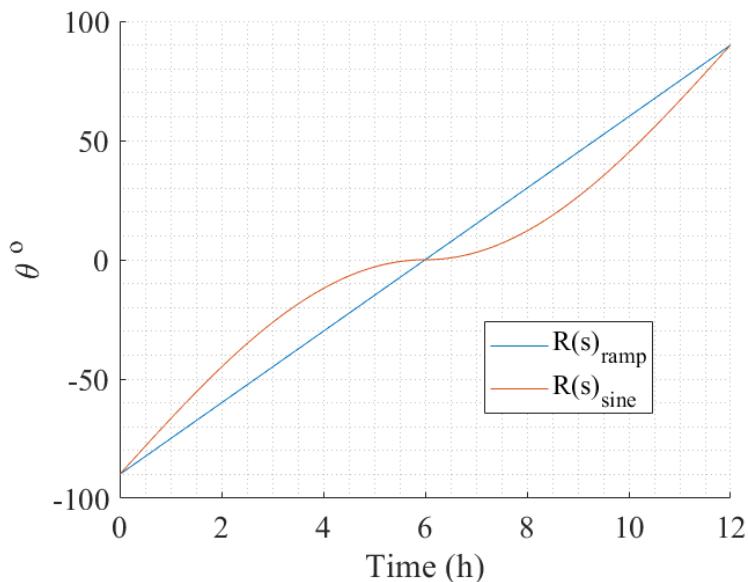


Figure 17:  $R(s)$  Modelled as Sine and Ramp Signals

## 4 MATLAB-Simulink<sup>®</sup> Model

### 4.1 Model step 1: Model Constants

Set the constants in the MATLAB file and run the file.

```
1 J=25.5759;%Moment of Inertia of the system
2 c=8; %damping constant of the system
3 k=0; % stiffness constant of the system
4 R=0.5; %Resistance of the motor
5 L=5e-3;%Motor inductance
6 kt=4; %Motor constant
7 ke=0.00083; %back emf constant
8 I=19e-10; %Motor rotor inertia
9 bm=0.00123;%motor damping constant
10 m=98.87; %m [kg]
11 g=9.81; %gravitational constant m/s^2
12 d=0.11055; % [m] distance to pivot from cg.
13 q=0.11055; % [m] moment arm for disturbance (pseudo random)
14 Vmax=230; % Volts
15 length_tracker=2.268;%Solar trakcer length [m]
16 width_tracker=1.758; %Solar tracker width [m]
17 As=length_tracker*width_tracker; %Solar tracker area [m^2]
```

### 4.2 Non Linear Simulink<sup>®</sup> Model

- Simulink<sup>®</sup> 2023b was used.
- Saturation was added to limit voltage input (-230 V to 230 V).
- Saturation was added for physical limitation of the solar tracker (-90° to 90°).

Figure 18 shows the non-linear model in Simulink<sup>®</sup>.

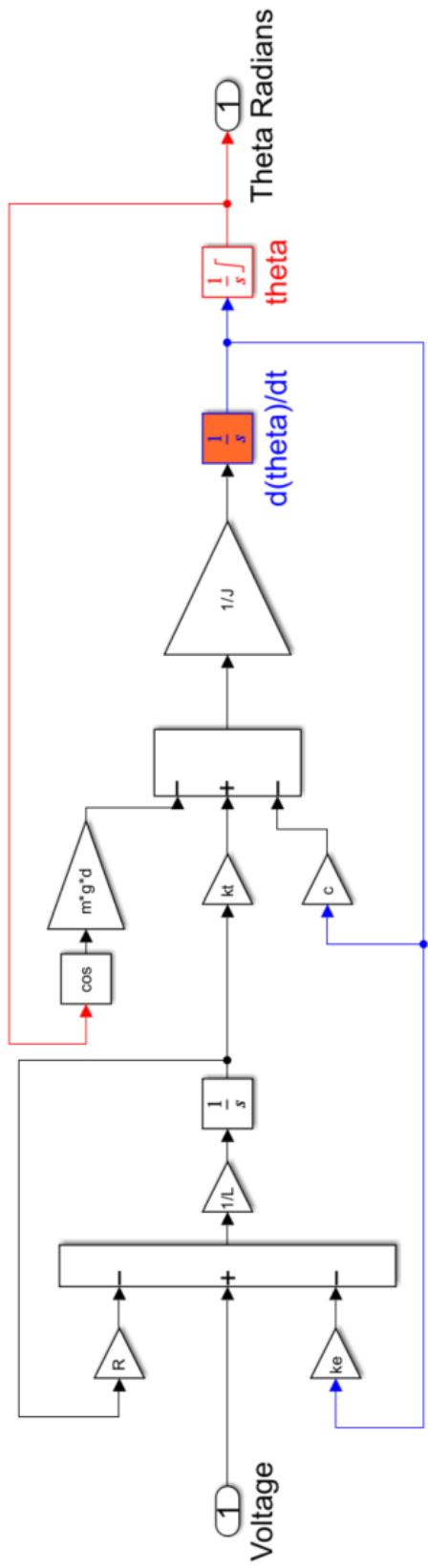


Figure 18: Non-Linear Model

## 5 Linearisation

Two methods are used to linearise the non-linear governing equation, the first method used applies a small perturbation to the non-constant terms in the governing non-linear equation and then linearises around the equilibrium point.

The second method uses the Simulink<sup>®</sup> linear analysis tool to get a linear equivalent of the non-linear governing equation.

### 5.1 Small Perturbation

Small perturbation theory is used to linearise the non-linear governing equation. The equation of motion for the system is given by:

$$J\ddot{\theta} = k_t I_a - mgd \cos \theta - c\dot{\theta} \quad (18)$$

Linearising about the equilibrium point  $(\theta_0, I_{a0})$ :

$$\theta = \theta_0 + \Delta\theta \quad (19)$$

$$I_a = I_{a0} + \Delta I_a \quad (20)$$

Substituting these into the equation of motion:

$$J \frac{d^2(\theta_0 + \Delta\theta)}{dt^2} = k_t(I_{a0} + \Delta I_a) - mgd \cos \theta - c \frac{d(\theta_0 + \Delta\theta)}{dt} \quad (21)$$

Since  $\theta_0$  is a constant:

$$\frac{d^2(\theta_0 + \Delta\theta)}{dt^2} = \frac{d^2(\Delta\theta)}{dt^2} = \Delta\ddot{\theta} \quad (22)$$

For the damping term:

$$\frac{d(\theta_0 + \Delta\theta)}{dt} = \frac{d(\Delta\theta)}{dt} = \Delta\dot{\theta} \quad (23)$$

For the mass term, the following *cosine* trigonometric identity is used.

$$\cos(\theta_0 + \Delta\theta) = \cos \theta_0 \cos \Delta\theta - \sin \theta_0 \sin \Delta\theta \quad (24)$$

For the current term:

$$k_t(I_{a_0} + \Delta I_a) = k_t I_{a_0} + k_t \Delta I_a \quad (25)$$

Combining the equations gives:

$$J\Delta\ddot{\theta} = k_t \Delta I_a + k_t \Delta I_a - c\Delta\dot{\theta} - mgd(\cos \theta_0 \cos \Delta\theta - \sin \theta_0 \sin \Delta\theta) \quad (26)$$

Assuming  $I_{a_0} = 0$  and  $\theta_0 = 0$ :

$$J\Delta\ddot{\theta} = k_t \Delta I_a - c\Delta\dot{\theta} - mgd \cos \Delta\theta \quad (27)$$

Assuming zero initial conditions, the Laplace transform of equation 27 is taken, then the transfer function is given by equation 28 is determined:

$$G(s) = \frac{\Theta(s)}{V_a(s)} \quad (28)$$

Where the voltage is written in terms of current.

## 5.2 Simulink®

The non-linear model on Figure 18 was linearised using the Simulink® linear analysis tool, refer to Figure 19.

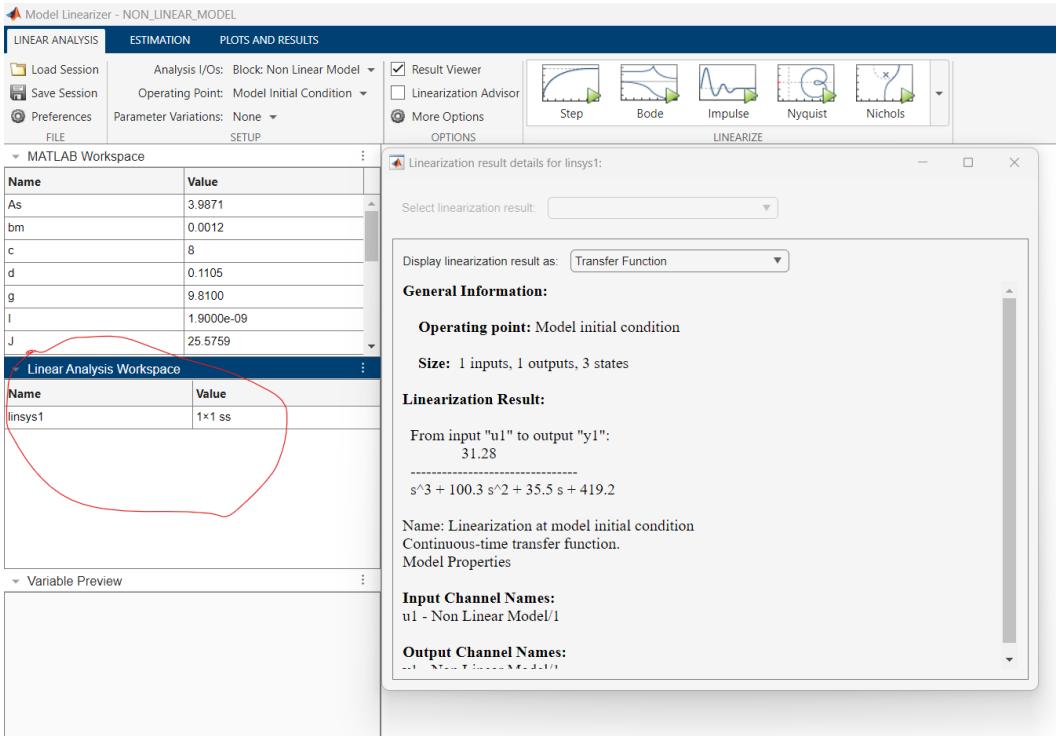


Figure 19: Linearisation Process on Simulink®

The output - *linsys1* (circled on Figure 19) was exported to the Matlab workspace and the resulting transfer function for the open loop system was given by the equation 29.

$$G(s) = \frac{31.28}{s^3 + 100.3s^2 + 35.5s + 419.24} \quad (29)$$

- The system is 3<sup>rd</sup> order.
- The system does not have numerator dynamics as the numerator is a constant value (there is no  $s$  term).
- The open loop dynamics are governed by the characteristic equation, which is the denominator of the transfer function.
- The various input signals represent the possible input voltages the system can take in the real non linear world.

## 6 Linear vs Non-Linear Models Comparison

In control theory and controller design, linearised equations are commonly employed. This approach involves developing a controller based on a simplified linearised system, which offers ease and convenience. However, it is crucial for the controller to function effectively when applied to the original non-linear system. To ensure equivalence between the two systems, their outputs are compared across varying input signals, as depicted on Figure 20. If the outputs match for corresponding inputs, the system is deemed linearised successfully.

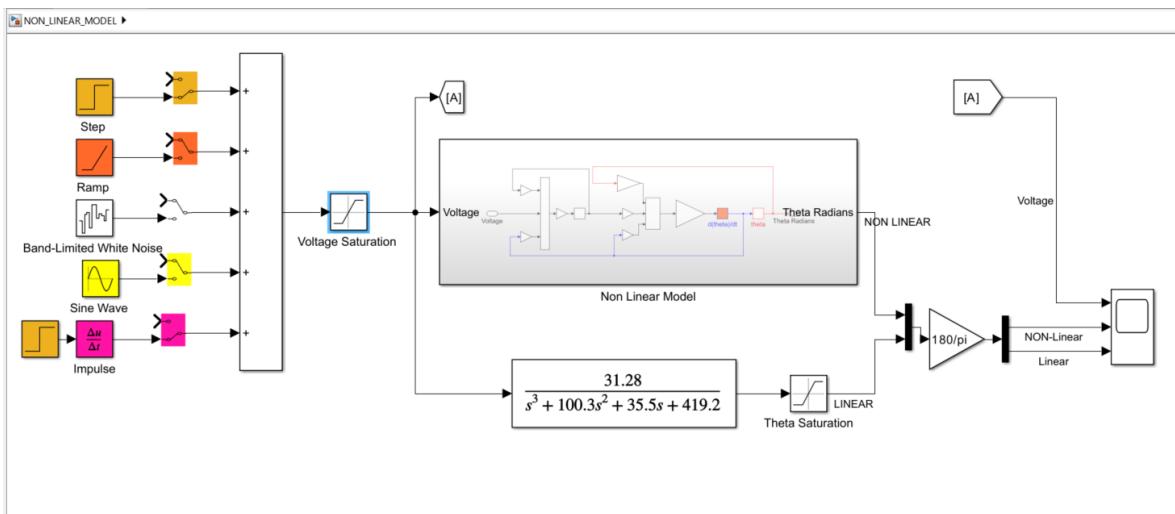


Figure 20: Non-linear and Linearised Models

The following is undertaken to compare the linear and non-linear systems on Figure 20.

1. Select a single voltage signal and set all other signals to zero.
2. Run simulation.
3. Compare results of linear and non-linear models.
4. Systems are the same if the difference between them is low, i.e less than 1%.
5. System considered linearised if the outputs are the same for all input voltages.

## 6.1 Non-Linear vs Linear: Step Input Voltage Response

The linear and non-linear responses for a step input voltage are the same with a percentage difference within 1%, as shown on Figure 21.

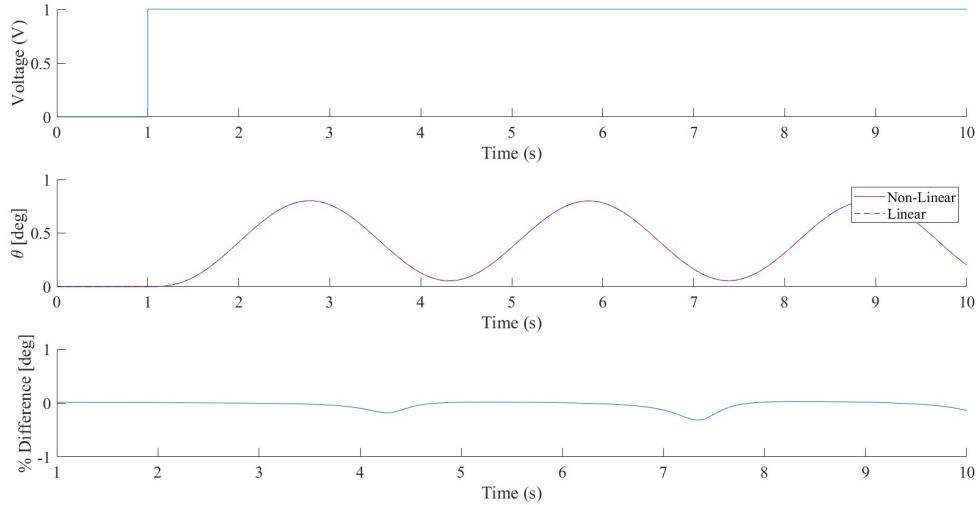


Figure 21: Non-Linear vs Linear: Step Input Voltage Response

## 6.2 Non-Linear vs Linear: Ramp Input Voltage Response

The linear and non-linear responses for ramp input voltage are the same with a percentage difference within 1%, as shown on Figure 22.

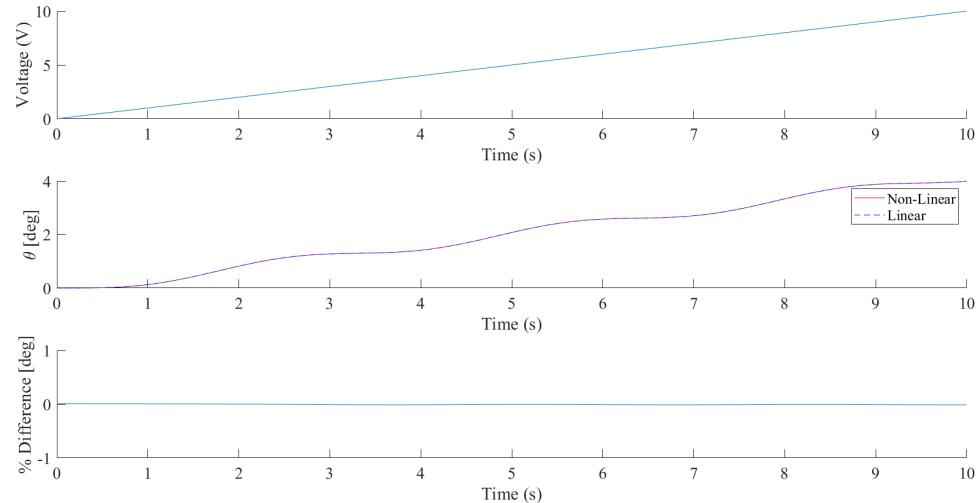


Figure 22: Non-Linear vs Linear: Ramp Input Voltage Response

### 6.3 Non-Linear vs Linear: Sine Input Voltage Response

The linear and non-linear frequency responses are the same with a percentage difference within 1%, as shown in Figure 23. Qualitatively, one can visualise the phase difference between the input voltage and the output angle.

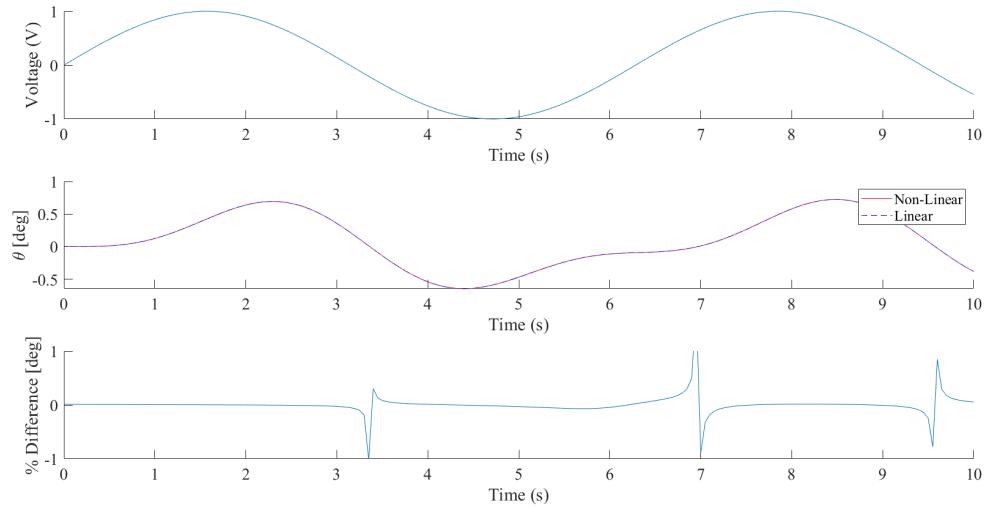


Figure 23: Non-Linear vs Linear: Ramp Input Voltage Response

### 6.4 Non-Linear vs Linear: White Noise Input Voltage Response

The white noise voltage signal has an output difference within 1%, as shown on Figure 24.

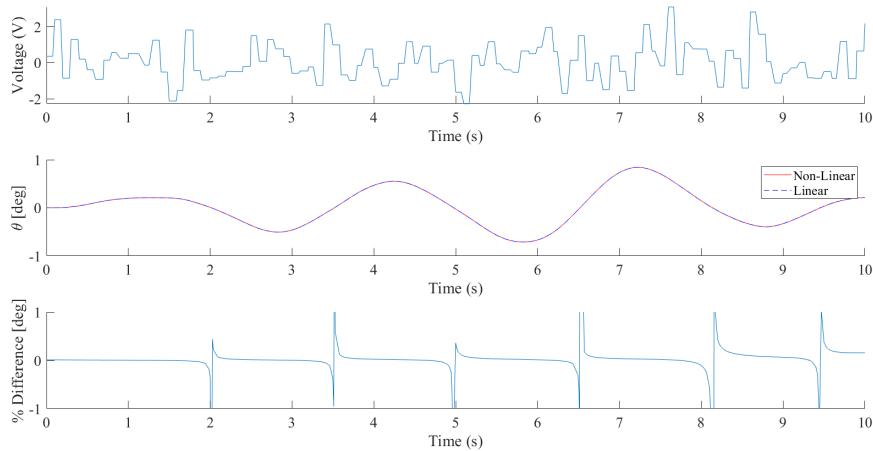


Figure 24: Non-Linear vs Linear: Ramp Input Voltage Response

## 6.5 Non-Linear vs Linear: Impulse Input Voltage Response

The Impulse voltage signal has an output difference within 1%, as shown on Figure 25.

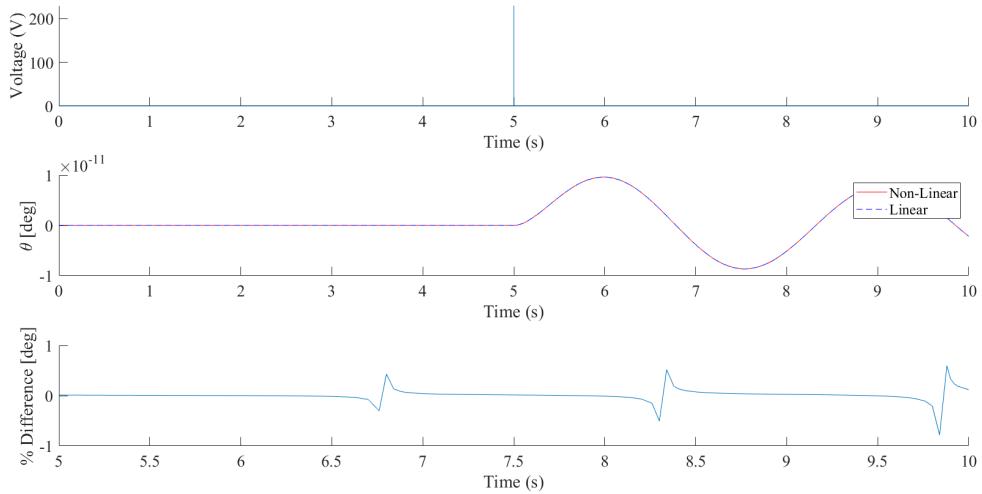


Figure 25: Non-Linear vs Linear: Impulse Input Voltage Response

## 6.6 Linearisation Assessment

- The system is successfully linearised for various input signals.
- The signals are realistic and have a saturation limit to reflect the realistic voltage input limit.

## 7 Uncontrolled System: Stability Analysis

The uncontrolled system has the transfer function given by equation 29. The stability in the time and frequency domains is to be assessed. Assuming negative unit feedback, the closed loop uncontrolled response is given by equation 30.

$$T(s) = \frac{31.28}{s^3 + 100.3s^2 + 35.5s + 419.24} \quad (30)$$

### 7.1 Uncontrolled Analysis: Step Response

The step response to a desired angle without a controller is given on Figure 26.

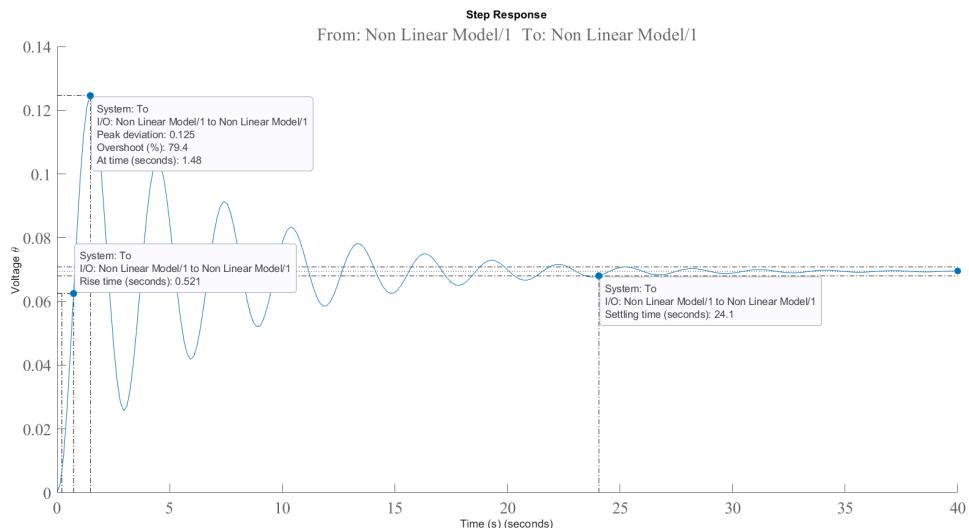


Figure 26: Uncontrolled Analysis: Step Response

Step response of the uncontrolled plant is compared to the desired performance on Table 7.

Table 7: Comparing Uncontrolled Plant with Desired Performance

| Parameter         | Uncontrolled | Desired |
|-------------------|--------------|---------|
| Rise time (s)     | 0.521        | 2       |
| Settling time (s) | 24.1         | 5       |
| Overshoot (%)     | 79.4         | 10      |

Table 7 shows that the uncontrolled plant has rise time that is within the desired 2 seconds. However, both the settling time and overshoot of the uncontrolled are much greater than what is desired 5 seconds and 10% , respectively.

## 7.2 Uncontrolled Analysis: Routh-Hurwitz Stability

Using the characteristic equation from the closed loop transfer function shown in equation 30, the first column on the Routh-Hurwitz criterion does not have any sign changes, hence making the system stable. This is shown on Table 8

Table 8: Routh-Hurwitz Stability Analysis

| $S^n$ | Column 1 | Column 2 | Column 3 |
|-------|----------|----------|----------|
| $S^3$ | 1        | 35.5     | 0        |
| $S^2$ | 100.3    | 419.24   | 0        |
| $S^1$ | 31.32    | 0        | 0        |
| $S^0$ | 419.24   | 0        | 0        |

While based on Table 8, the system is stable, the system does not satisfy the desired performance specifications outlined on Table 5, refer to Table 7. The system therefore requires a controller that will enable it to satisfy the desired performance.

## 7.3 Uncontrolled Analysis: Pole-Zero Map

The system has 3 roots as it is a  $3^{rd}$  order type. The system has two imaginary roots which lie just to the left and the last root lies further on the real negative axis as shown on Figure 27. This explains the oscillatory behaviour on Figure 26. From the dominant root concept, the oscillatory roots are the dominant roots as they are closest to the imaginary axis.

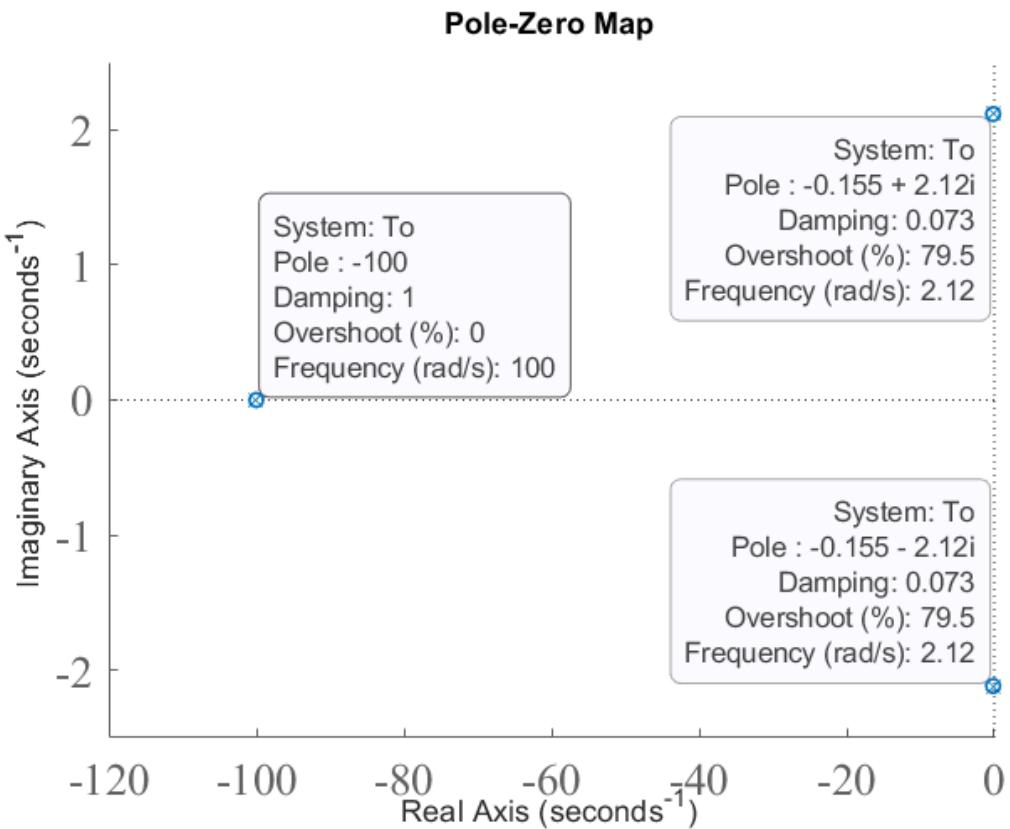


Figure 27: Uncontrolled Analysis: Pole-Zero Map

## 7.4 Uncontrolled Analysis: Eigenvalues

The generic response of a system is:

$$\Theta(t) = Ae^{nt} \cos(\omega t) + Be^{nt} \sin(\omega t) \quad (31)$$

Where the eigenvalues are:

$$\lambda_{1,2} = n \pm i\omega \quad (32)$$

The eigenvalues are the roots of the denominator system in equation 30. If the system has all roots with a negative real part, then it is stable as the response has an exponential decay, refer to equation 31. The system has negative roots with a root separation factor of more than 10, (100). The dominant roots are the pair of complex roots which cause the oscillations. The system can be approximated as second order for analysis due to the large root separation factor.

The eigenvalues of the uncontrolled system are:

$$\begin{aligned} & -100 + 0.0000i \\ & -0.0015 + 2.12i \\ & -0.0015 - 2.12i \end{aligned}$$

## 7.5 Uncontrolled System: Nyquist Analysis

The Nyquist plot on Figure 28 implies stability as the plot does not circle -1. The open-loop system is stable.

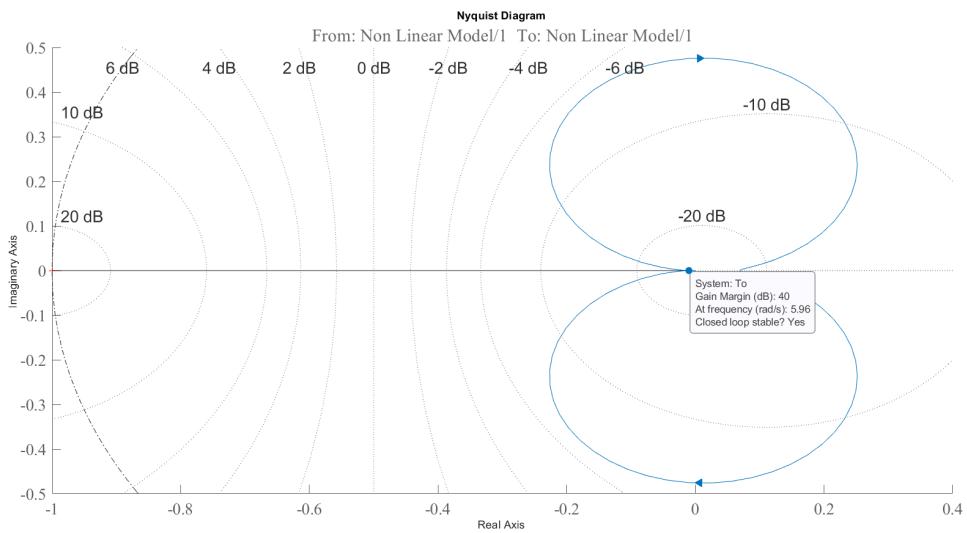


Figure 28: Uncontrolled System: Nyquist Analysis

The Nyquist plot maps the frequency response of the system as a locus in the complex plane, plotting the real and imaginary parts of the open loop transfer function of the system as the frequency varies from negative to positive infinity.

From Figure 28, the gain margin is measured at 40 dB. This signifies that the system can handle significant variations in gain before crossing into instability. This margin is evaluated at a specific frequency, here noted as 5.96 rad/s. The gain margin indicates the distance to the point where the gain would become infinite, and a positive gain margin, especially as large as 40 dB, is a strong indicator of system stability.

According to the Nyquist Stability Criterion, for a system to be stable, the Nyquist plot must not encircle the  $(-1 + j0)$  point. On Figure 28, the trajectory does not encircle the  $(-1 + j0)$  point, confirming that the closed-loop system remains stable under the given conditions.

## 7.6 Uncontrolled System: Root Locus Analysis

For the Root Locus analysis, the system is only stable for the initial value of  $K = 0$  in equation 33. This is shown on Figure 29.

$$1 + KT_o(s)H(s) = 0 \quad (33)$$

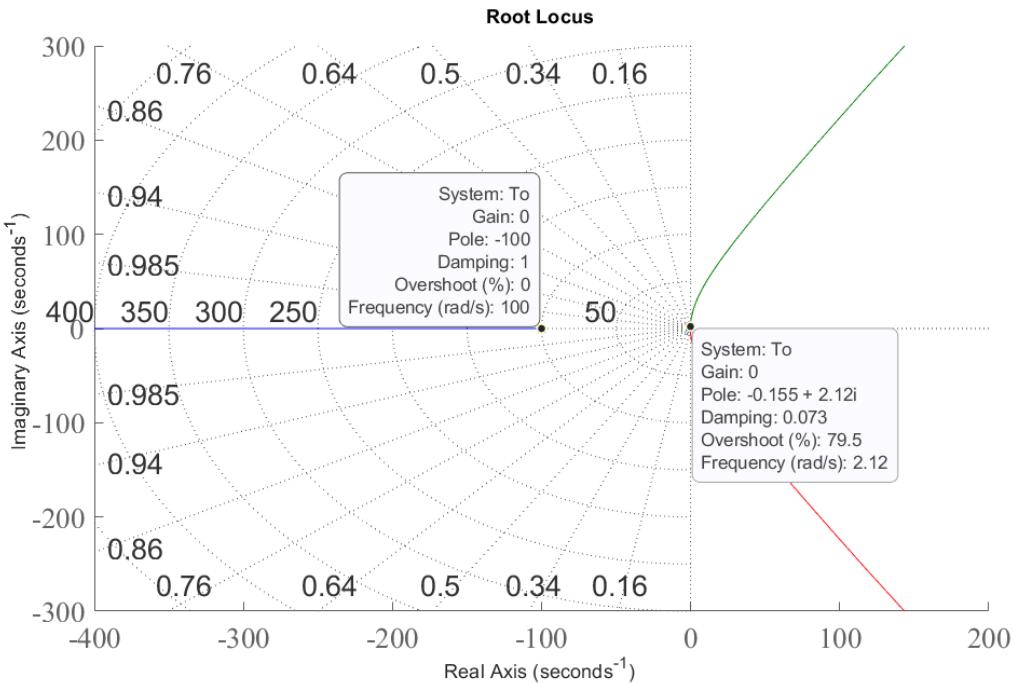


Figure 29: Uncontrolled System: Root Locus Analysis

The Root Locus plot represents the possible locations of the closed-loop poles of a control system as a single system parameter is varied, typically the gain -  $K$ . The Root Locus plot is essential for understanding how the poles of a system move in the complex plane as the gain changes, providing insights into system stability.

On Figure 29, location of the poles are marked as they move through the complex plane for different values of gain,  $K$ . The trajectories of these poles are depicted as green and red lines starting from the initial pole locations (marked at gain = 0) and moving outward as the gain increases. The plot indicates that for the initial configuration ( $K = 0$ ), the system is stable, with poles located in the left half-plane and on the imaginary axis. However, as the gain is increased the system becomes unstable since the poles that were initially on the imaginary axis follow the path into the right half-plane.

## 7.7 Uncontrolled System: Frequency Analysis

In a stable system, either both margins should be positive, or the phase margin should be greater than the gain margin, which is the case on Figure 30. The system is closed-loop stable.

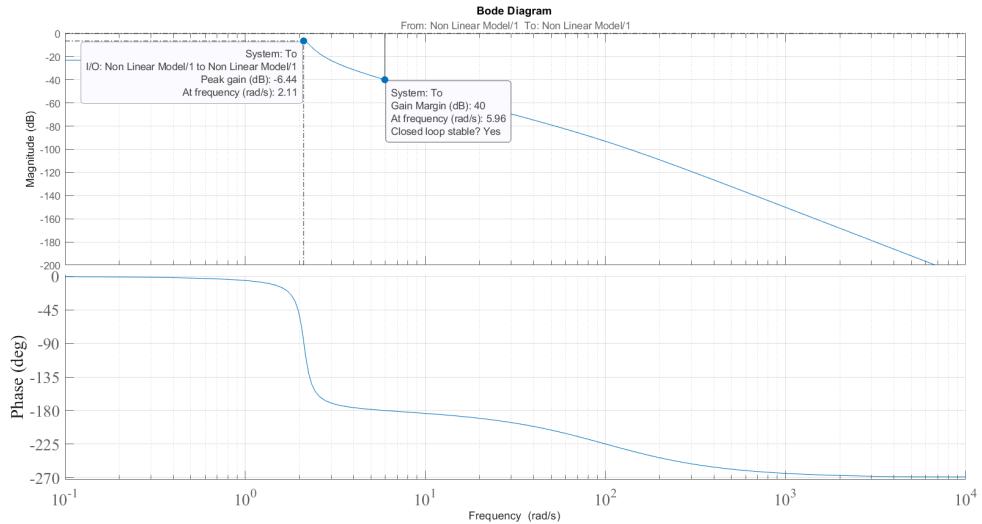


Figure 30: Uncontrolled System: Frequency Analysis

The magnitude plot, on Figure 30, shows that the system has a gain margin of 40 dB at a frequency of 5.96 rad/s, indicating that the system can increase its gain by 40 dB before becoming unstable. This significant positive gain margin is indicative of a stable system. Additionally, the phase plot on Figure 30, while not explicitly showing the phase margin, implies stability as there is no phase crossing the critical -180 degrees near the gain crossover frequency.

## 7.8 Uncontrolled System Stability Analysis Assessment

1. The system is open loop and closed loop stable.
2. A controller is required to meet the desired performance specifications outlined on Section 3.3 even though the system is stable.
3. The alteration required would move the roots into the left half of the plane, refer to Figure 28.
4. This is to be achieved by adding poles and zeros to the systems in a fitting manner (i.e. PID controller, Root Locus compensator).

5. The goal is to alter the system without adding too much complexity (i.e. minimise system complexity), which makes implementation easier.

## 8 PID Controller Design

A standard PID without a filter is shown on Figure 31.

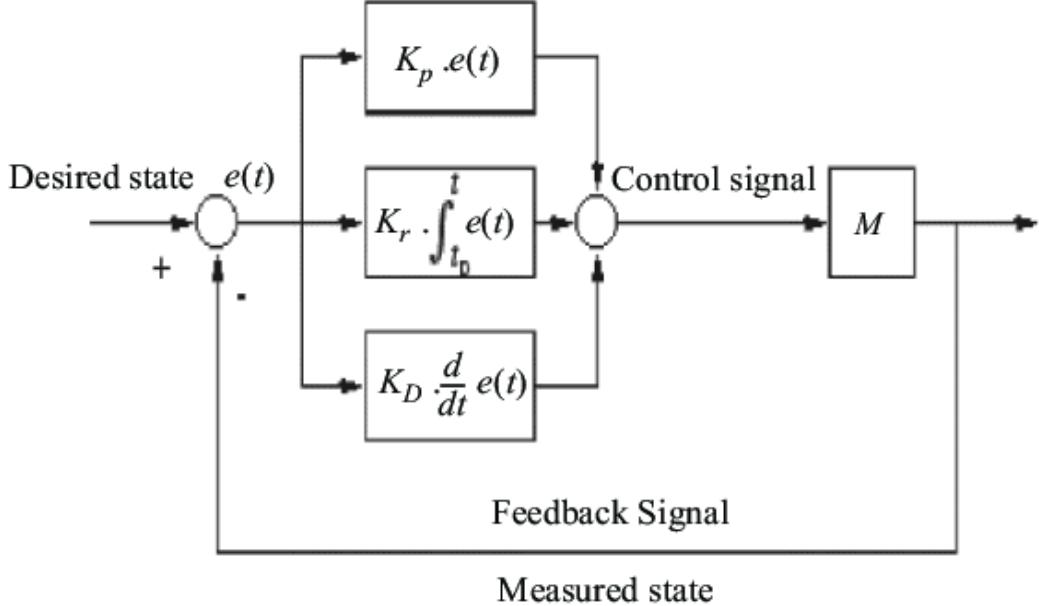


Figure 31: PID Controller on a System Block Diagram [13]

The general equation in the Laplace domain is given by

$$C_{pid}(s) = K_p + \frac{K_i}{s} + K_d s \quad (34)$$

$$C_{pid}(s) = \frac{K_d^2 + K_p s + K_i}{s} \quad (35)$$

With M in Figure 31 representing the transfer function of the system, the closed-loop system with a PID assuming negative unit feedback is given by:

$$Tc(s) = \frac{C_{pid}(s)G(s)}{1 + H(s)C_{pid}(s)T_o(s)} \quad (36)$$

$$G(s) = \frac{31.28}{s^3 + 100.3s^2 + 35.5s + 419.24} \quad (37)$$

## 8.1 Tuning PID with Control Systems Designer

The linearisation output from the linearisation process in Section 5 is loaded as follows:

## 8.2 Load Control Systems Designer

```
1 clc; close all;
2 load('simulink_linearisation_output')
%Generate transfer function
4 Gs=tf(linsys1); disp(Gs);
5 To=Gs
6 controlSystemDesigner(Gs)
```

This loads the control system designer shown on Figure 32 with the uncontrolled transfer function from equation 29.

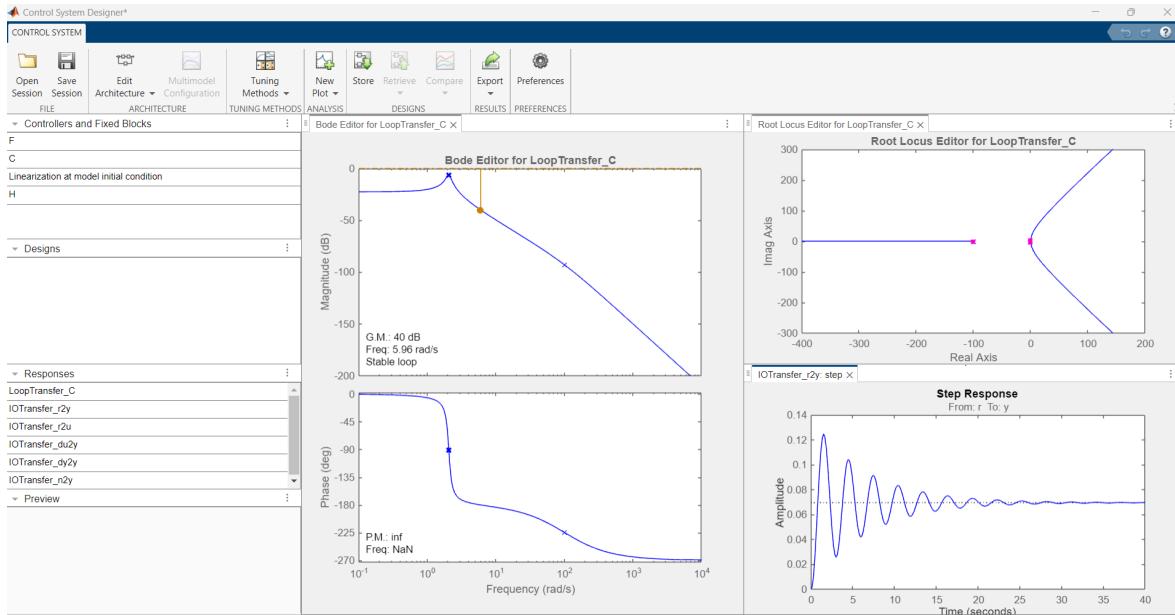


Figure 32: Control System Designer Interface

### 8.3 Set Performance Specifications

The next step is setting the desired performance specifications, as shown on Figure 33. The time domain step response specifications from Table 5 are set.

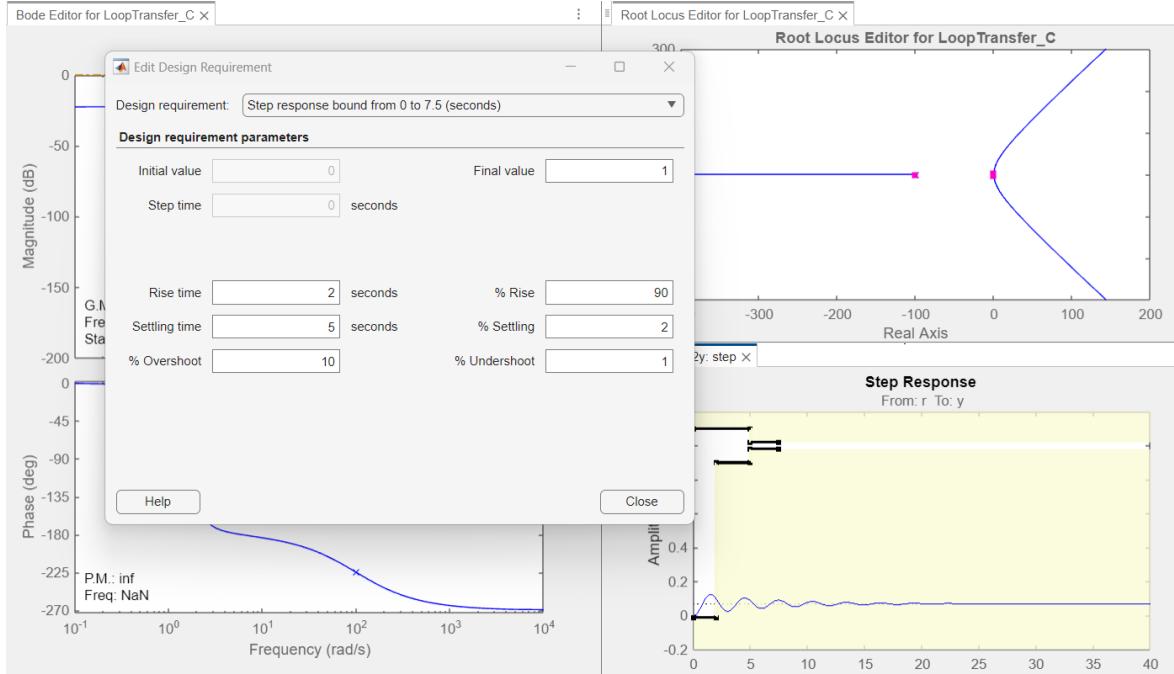


Figure 33: Setting Desired Performance Specifications

### 8.4 Tuning PID

- A PID tuner tool is used for obtaining the PID gains.
- A simple P controller is tried first.
- A PI controller is then tuned.
- A PD controller is tuned, provided that the previous PI controller does not meet the requirements.
- A PID controller is then tuned if the previous controllers do not meet the requirements.
- The effects of each gain are assessed in the process.
- Unlike the analytical approach, the use of the tuner tool makes for a rapid solution.  $K_d$  is introduced, etc.
- PID control theory is still used in the process, i.e. to eliminate steady-state error, integral component is introduced, to improve the rate of the system, the gain  $K_d$  is introduced etc.

- The tuner adjusts the gains based on the controller type, response time and transient behaviour.
- The three variables are adjusted until the output for the step response is within the white region shown on Figure 34.

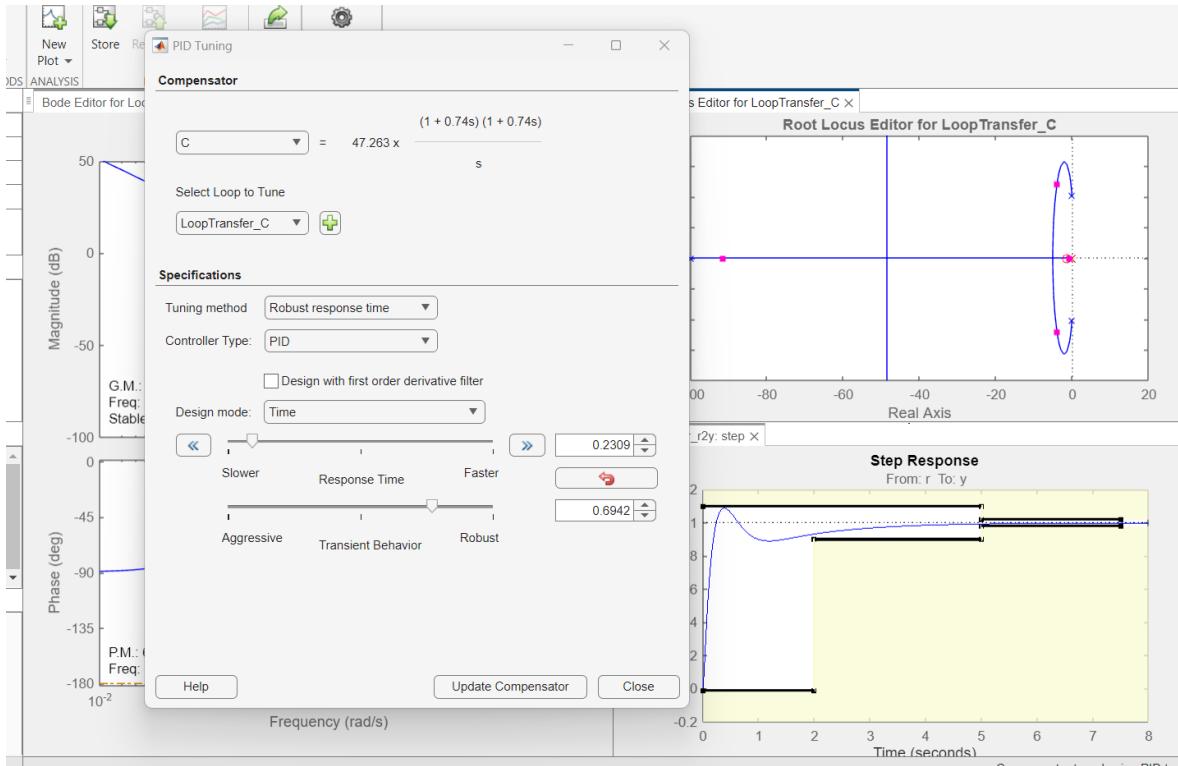


Figure 34: Successfully Tuned Example

## 8.5 Transforming $C(s)$

The desired controller  $C(s)$  in Figure 34 is transformed into the standard PID form shown in equation 35.

$$C(s) = \frac{25.626(s + 1.358)^2}{s} \quad (38)$$

$$C(s) = \frac{25.626s^2 + 69.600216s + 47.258547}{s} \quad (39)$$

Equating terms :

$$K_d = 25.626$$

$$K_p = 69.600216$$

$$K_i = 47.258547$$

The same process applies to the P, PI, and PD controllers.

## 8.6 P Controller

The P controller is defined by equation 40. It can be tuned by the root locus shown on Figure 29 by simply adjusting the value  $K_p$  to find stability.

$$C_p(s) = K_p \quad (40)$$

A gain value of  $K_p = 50$  has a stable system, but it does not meet the other performance specifications, such as settling time, and there is a big steady-state error. This is shown on Figure 35 and Figure 36.

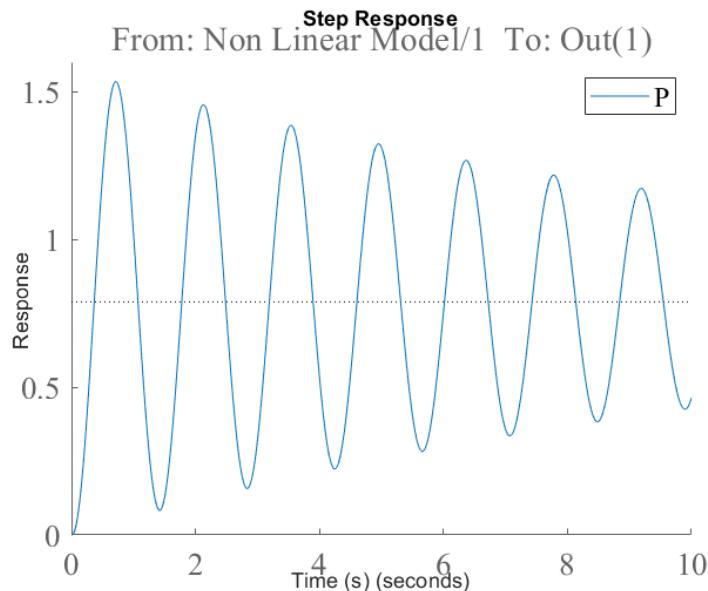


Figure 35: Step Response Plot: P Controller

| Parameter          | Desired | P controller | % Difference | Met |
|--------------------|---------|--------------|--------------|-----|
| Rise Time (%90)    | 2       | 0.2384       | 88.1         | Yes |
| Settling Time (%2) | 5       | 49.4962      | -889.9       | No  |
| Overshoot          | 10      | 94.4284      | -844.3       | No  |

Figure 36: Step Response Results: P Controller

The following deductions were made for the P controller:

- System is stable.
- Desired performance not met.

- The rate of error correction is not ideal, hence adding a derivative component can improve response.

## 8.7 PD Controller

The addition of the  $K_d$  and tuning results in equation 41, where  $K_d = 57.5271$  and  $K_p = 140.31$ , with response outputs on Figure 37 and Figure 38.

$$C_{pd}(s) = 57.5271s + 140.31 \quad (41)$$

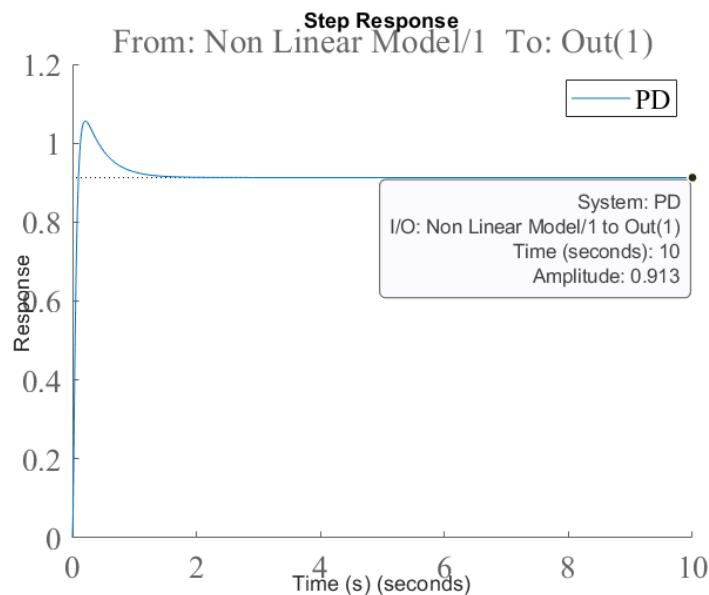


Figure 37: Step Response Plot: PD Controller

| Parameter          | Desired | PD controller | % Difference | Met   |
|--------------------|---------|---------------|--------------|-------|
| Rise Time (%90)    | 2       | 0.06          | 97           | Yes ▾ |
| Settling Time (%2) | 5       | 0.9178        | 81.6         | Yes ▾ |
| Overshoot          | 10      | 15.6611       | -56.6        | No ▾  |

Figure 38: Step Response Results: PD Controller

The following deductions were made for the PD controller:

- The system is stable, but the overshoot is above the desired 10%.
- The introduction of the  $K_d$  gain improves the rate of response.

- There is a steady state error of approximately 9%. This is seen on Figure 37.
- Trying out a PI controller can reduce the steady state error and perhaps give a better response.

## 8.8 PI Controller

The PI controller is defined by equation 42 with  $K_p = 10$  and  $K_i = 3$ . The outputs are shown on Figure 40 and Figure 39.

$$C_{pi}(s) = \frac{10s + 3}{s} \quad (42)$$

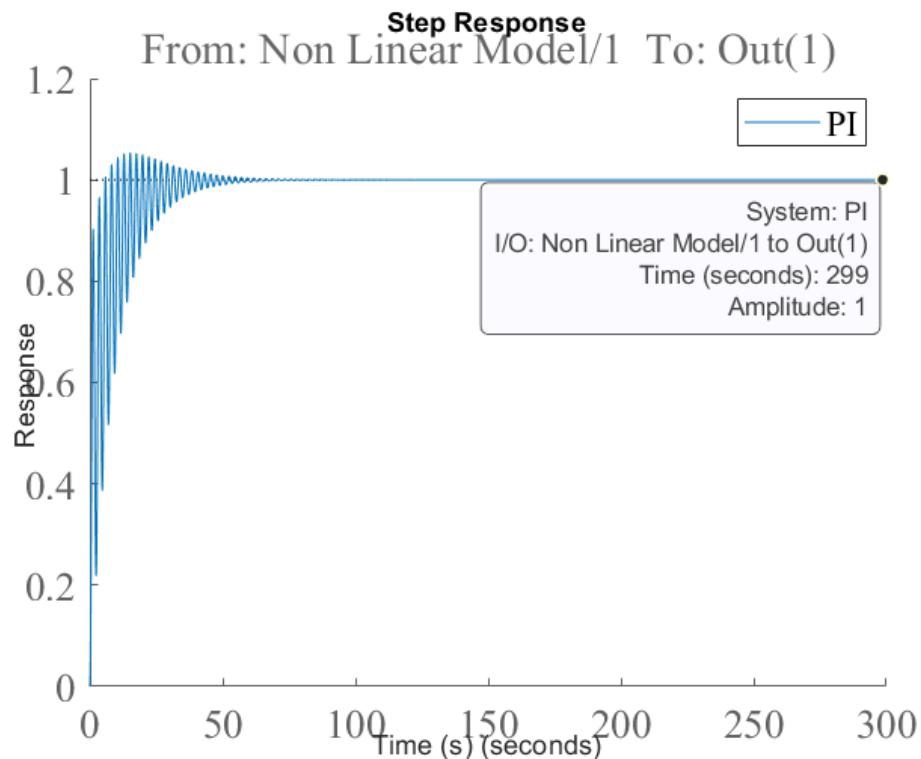


Figure 39: Step Response Plot: PI Controller

| Parameter          | Desired | PI controller | % Difference | Met   |
|--------------------|---------|---------------|--------------|-------|
| Rise Time (%90)    | 2       | 3.0525        | -52.6        | No ▾  |
| Settling Time (%2) | 5       | 39.8467       | -696.9       | No ▾  |
| Overshoot          | 10      | 5.2432        | 47.6         | Yes ▾ |

Figure 40: Step Response Results: PI Controller

The following deductions were made for the PI controller:

- Adding the Integral component eliminates the steady state error. This is shown on Figure 39.

- Removing the Derivative component has a negative impact on the rate of error correction as the settling time significantly increases.
- Combining the Integral (for steady-state error correction), Derivative (rate of error response) and Proportional (kick-starting response) is expected to result in a working solution.

## 8.9 PID Controller

The tuned PID controller is described by equation 45, where  $K_p=70$ ,  $K_i = 47$  and  $K_d = 26$ . The output results are on Figure 42 and Figure 41.

$$C_{pid}(s) = \frac{26s^2 + 47s + 70}{s} \quad (43)$$

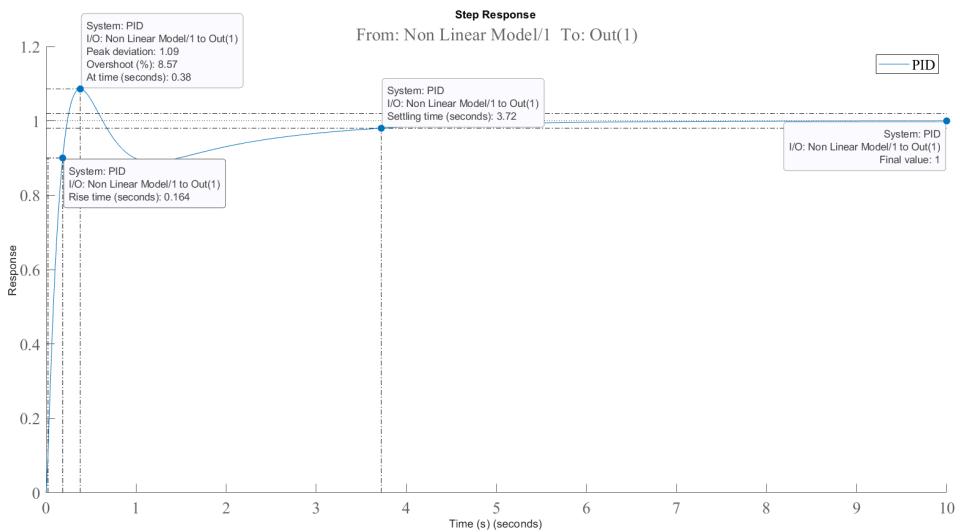


Figure 41: Step Response Plot: PID Controller

| Parameter          | Desired | PID controller | % Difference | Met   |
|--------------------|---------|----------------|--------------|-------|
| Rise Time (%90)    | 2       | 0.1645         | 91.8         | Yes ▾ |
| Settling Time (%2) | 5       | 3.7219         | 25.6         | Yes ▾ |
| Overshoot          | 10      | 8.5661         | 14.3         | Yes ▾ |

Figure 42: Step Response Results: PID Controller

The following deductions were made for the PID controller:

- The system is stable with all the time domain performance specifications met.
- The steady state error is 0 due to the Integral component.
- The PID, however, has to be assessed with saturation as the controller has no notion of the limited voltage, i.e. there is no constraint imposed on the amount of voltage that can be output. This is to be done in Simulink®.

- Before an in-depth analysis of the PID controller, a Root Locus-based design is to be first implemented and compared with the PID. Section 9 explores the Root Locus controller design.

## 9 Root Locus Design

The Root Locus editor shown on Figure 43 is used for the Root Locus-based design.

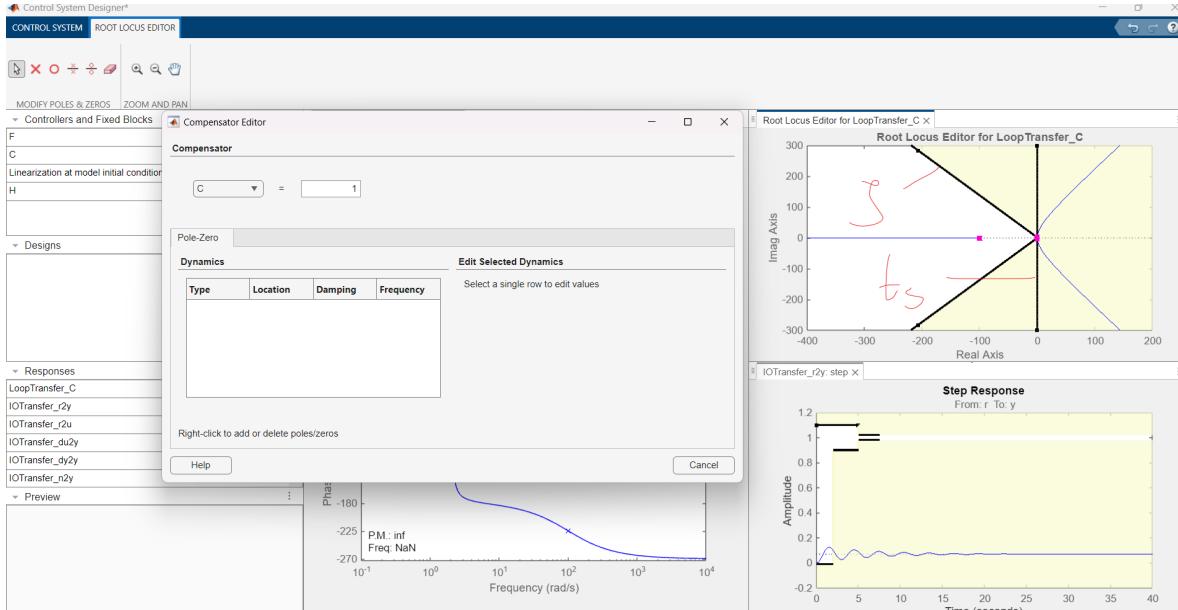


Figure 43: Root Locus Editor

### 9.1 Root Locus Design Methodology

- The desired performance specifications are set creating the white and yellow regions on Figure 43.
- The vertical line is related to the settling time (from control theory). To meet the settling time the roots have to lie to the left of the line.
- The damping ratio, hence overshoot, is related to the lines at an angle, shown on Figure 43.
- The roots that meet both settling time and overshoot must lie at some intersection of these lines.
- The Root Locus plot is altered until the desired outputs are met by adding poles and zeros to the system as well as shifting the root location. Qualitatively, this is checked by the step response plot on Figure 43.

## 9.2 Effect of Poles and Zeros

Table 9 discusses some of the effects that come into play as a result of adding a zero or pole.

Table 9: Effect of adding a Zero or a Pole

| Adding a Zero   | Adding a Pole   |
|---|---|
| Enhances response speed, damping, and has the potential to minimize overshoot | Delays the response of the system, increases settling time, and may induce instability  |
| Alters the trajectory of the Root Locus, drawing it closer to the zero        | Pushes the branches of Root Locus away from the pole                                    |
| Steady-state error can either improve or worsen depending on placement        | Usually, it leads to an increase in steady-state error and adds complexity to stability |

Table 10 discusses some of the practical considerations in controller design together with the system response tuning.

Table 10: Practical Considerations in Controller Design and System Response Tuning

| Consideration          | Description   |
|------------------------|---|
| PID controllers        | Adding a zero through the derivative action improves the transient response and reduces the overshoot, then adding a pole through the integral action can eliminate the steady-state error. |
| Lead compensates       | Introduce a zero before a pole to enhance the phase margin and stability while at the same time improving the transient response.   |
| Lag compensates        | Introduce a pole before a zero to improve the steady-state error without significantly compromising the transient response.   |
| System response tuning | Adding a zero near an existing pole can neutralize the effect of the pole, simplifying the system and potentially improving the performance of the system.                                  |
| Trade-offs             | Balancing of speed, stability, and accuracy based on application requirements.  |

### 9.3 Root Locus Tuning

The resultant Root Locus compensator is described by equation 44, where  $K$  is a constant gain that determines the location of the roots. For example,  $K = 37$  gives a stable response, as shown on Figure 44. The output response for the Root Locus are on Figure 46 and Figure 45.

$$C_{rlocus}(s) = K \frac{(1 + 0.78s)(1 + 0.78s)}{s(1 + 0.0059s)(1 + 0.01s)} \quad (44)$$

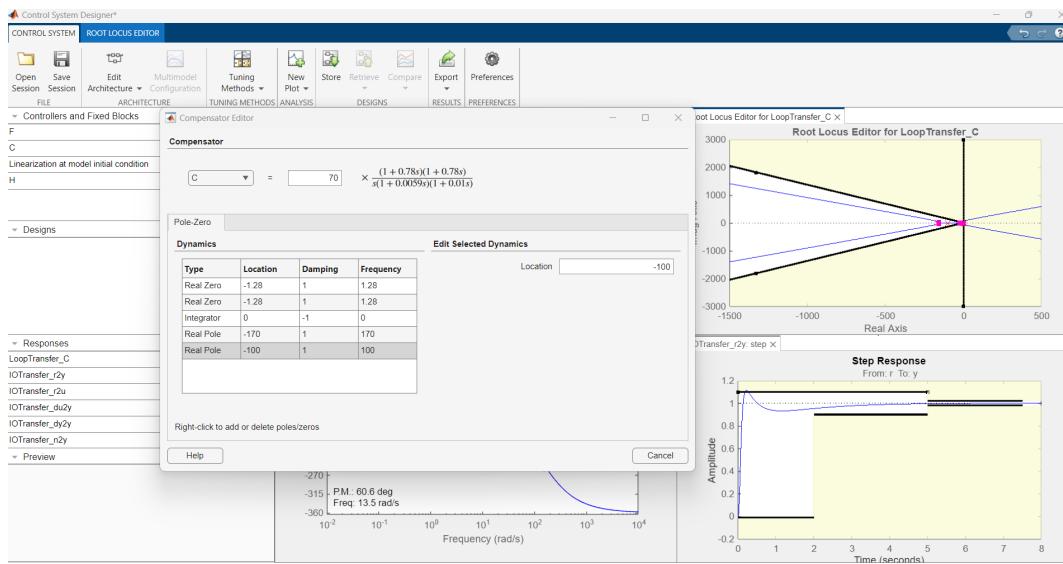


Figure 44: Output Response for the Root Locus

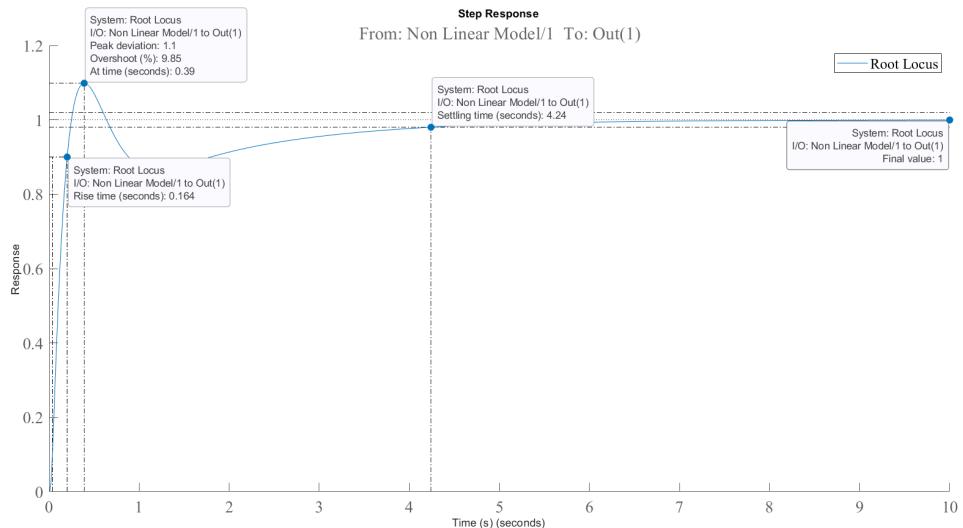


Figure 45: Step Response Plot: Root Locus

| Parameter          | Desired | RL controller | % Difference | Met |
|--------------------|---------|---------------|--------------|-----|
| Rise Time (%90)    | 2       | 0.1641        | 91.8         | Yes |
| Settling Time (%2) | 5       | 4.2408        | 15.2         | Yes |
| Overshoot          | 10      | 9.8529        | 1.5          | Yes |

Figure 46: Step Response Results: Root Locus

The following deductions were made for the Root Locus compensator.

- The system is stable and meets all the time domain performance specifications for a range of  $K$  values. This is shown on Figure 47.
- The compensator comprises two real poles, two real zeros and an integrator (i.e. pole at 0).
- The impact of voltage saturation has to be investigated.
- The best controller is to be selected between the PID and the Root Locus compensator.

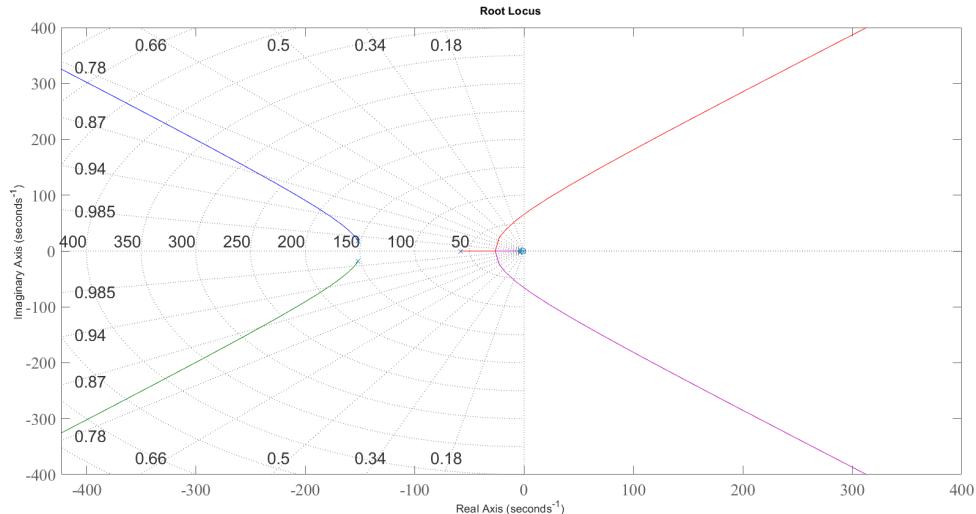


Figure 47: Root Locus Plot of Root Locus Controller System

## 9.4 Controller Selection

The candidate controllers have equations 45 and 46. The Simulink<sup>®</sup> setup on Figure 48 is used to assess the candidate controllers.

$$C_{pid}(s) = \frac{26s^2 + 47s + 70}{s} \quad (45)$$

$$C_{rlocus}(s) = 37 \frac{(1 + 0.78s)(1 + 0.78s)}{s(1 + 0.0059s)(1 + 0.01s)} \quad (46)$$

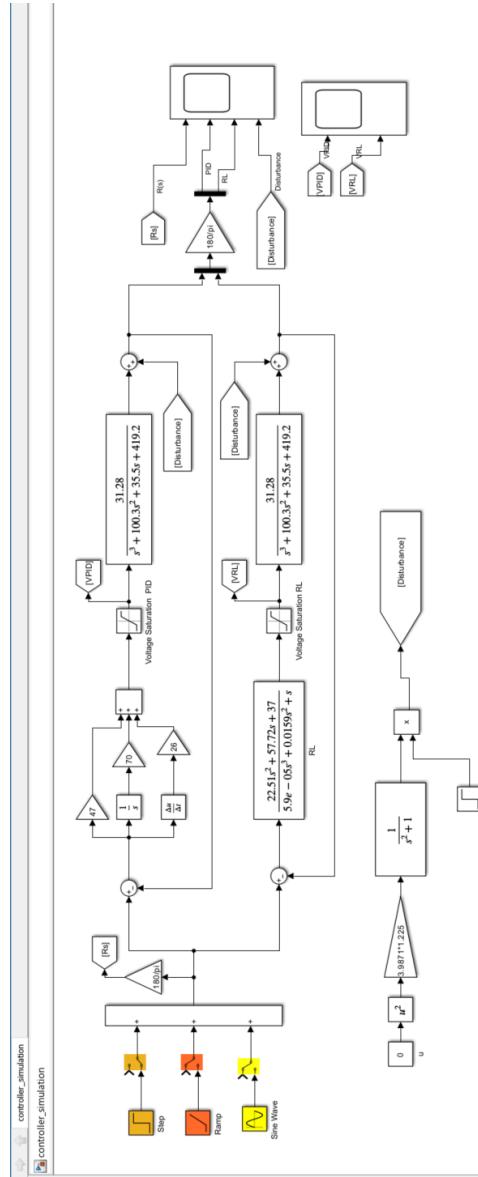


Figure 48: Simulink<sup>®</sup> Controller Comparison Setup

- Saturation voltage is added to assess the feasibility of the controllers with a realistically limited voltage.

- The controllers are compared for the same step, ramp and sine input signals.
- A disturbance, as modelled in Section 3, is added to assess the disturbance rejection qualities of the candidate controllers.

## 9.5 Candidate Controllers: Step Response Comparison

For a  $10^\circ$  step command, the Root Locus controller has a better response. It outperforms the PID controller, this is shown on Figure 49. Both controllers are feasible as they can control the solar tracker within the realistic voltage range.

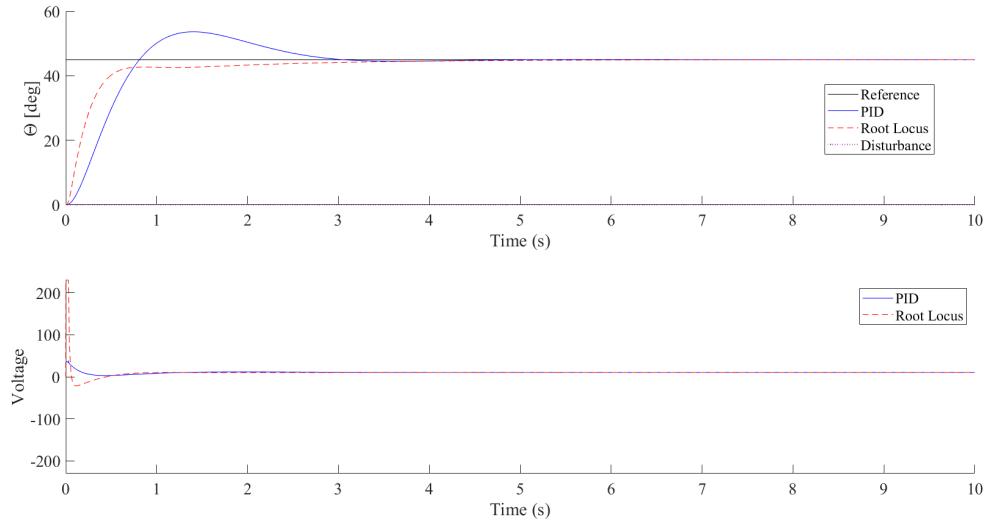


Figure 49: Candidates: Step without Disturbance

Qualitative assessment shows that the PID controller has better disturbance rejection when trying to maintain a step reference. This is shown on Figure 50.

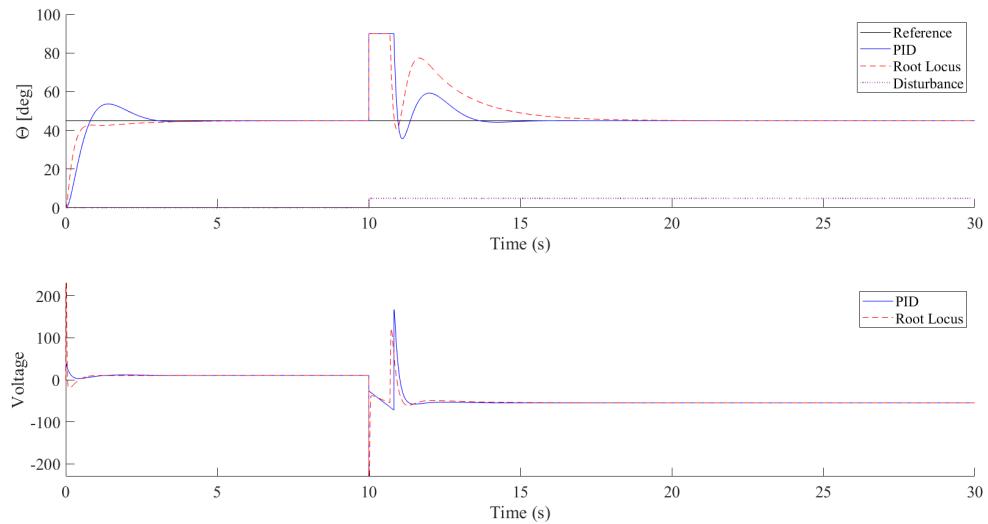


Figure 50: Candidates: Step with Disturbance

## 9.6 Candidate Controllers: Ramp Response Comparison

Both controllers have good ramp-following characteristics. Qualitative assessment shows that the PID controller has better disturbance rejection when trying to maintain a ramp reference, as seen on Figure 52. The PID controller returns to the desired path faster compared to the Root Locus.

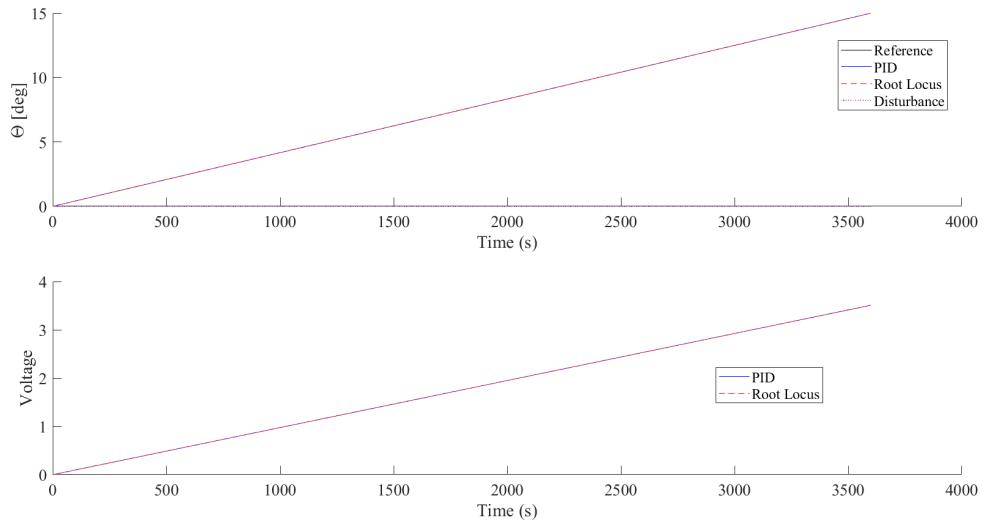


Figure 51: Candidates: Ramp without Disturbance

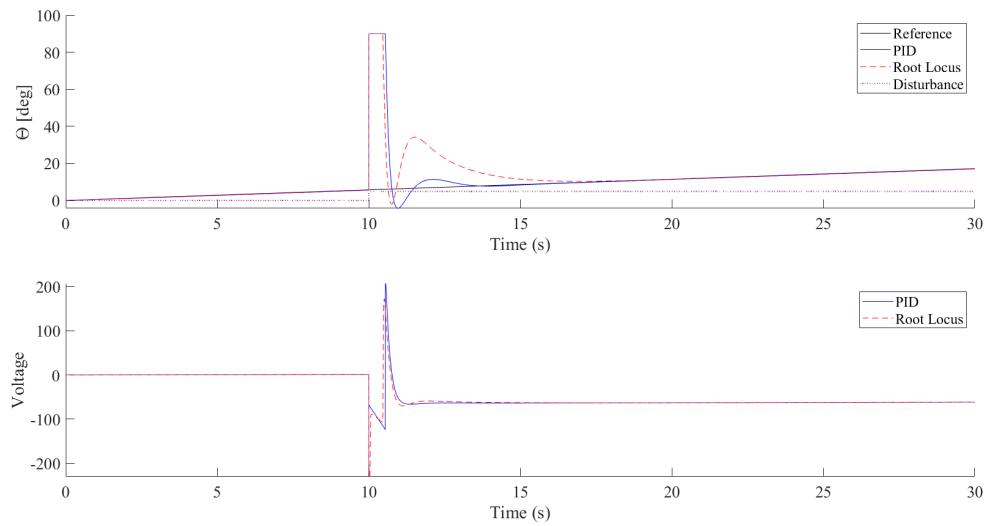


Figure 52: Candidates: Ramp with Disturbance

## 9.7 Candidate Controllers: Sine Response Comparison

Both controllers can follow a sinusoidal path, i.e. follow the "sun" in a sinusoidal manner as shown on Figure 53.

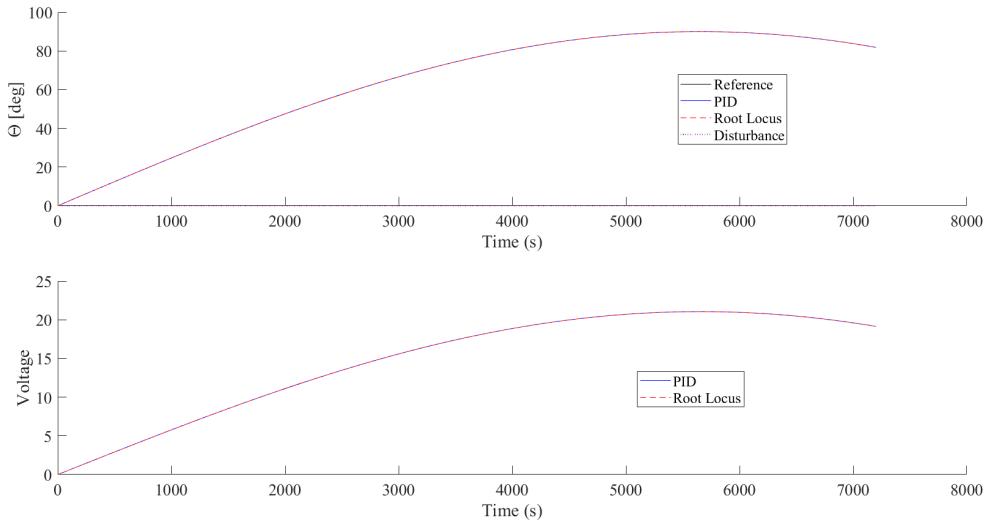


Figure 53: Candidates: Sine without Disturbance

With increasing frequency, both controllers have good frequency response domain characteristics. However, the Root Locus does better with following a sinusoidal reference, this is shown on Figure 54. The Root Locus response is more in sync with the desired response regarding phase and gain margin than the PID controller response.

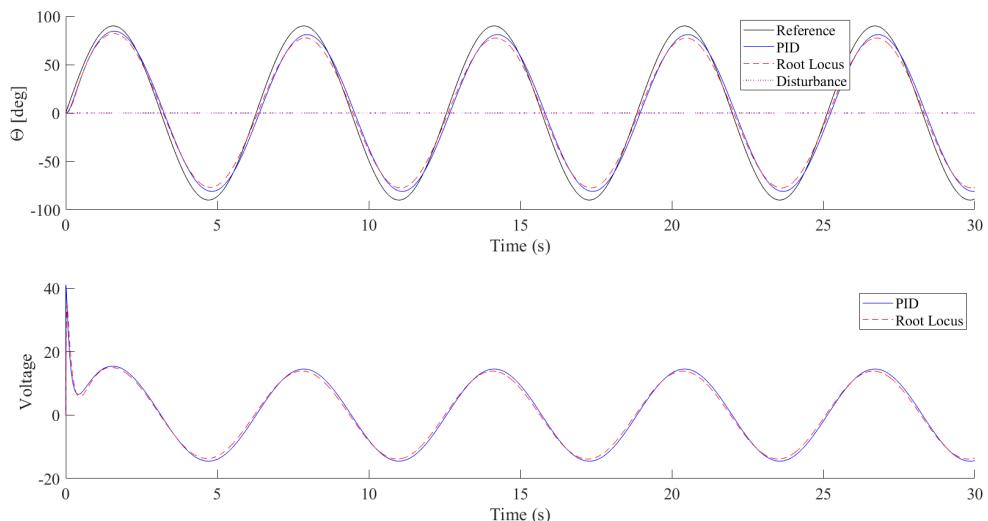


Figure 54: Candidates: Sine without Disturbance at Higher Frequency

The Root Locus controller also has better disturbance rejection when following a sinusoidal reference than the PID controller, refer to Figure 55.

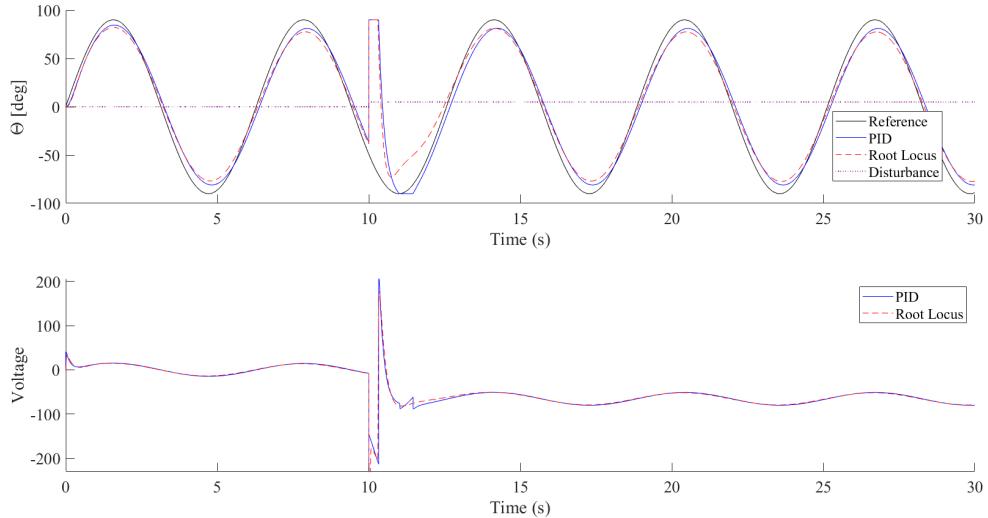


Figure 55: Candidates: Sine with Disturbance at Higher Frequency

## 9.8 Controller Selection

The controller selection on Figure 56 is based on a weighting system as follows:

- Each of the important parameters is given a weight for a cumulative value of 100.
- The controllers are compared for each parameter, with the best getting a score of 2, and if there is a tie, they both get 2.
- The total controller score is a sum of the score multiplied by the weight of each parameter.
- The winning controller is the Root Locus.
- Despite it having more complexity, which can have implementation issues, the benefits of good disturbance rejection and better response with minimal or no overshoot, depending on the case, outweighs the complexity issue.
- The Root Locus compensator can be digital or analogue, preferably digital.
- The digital implementation would be relatively easier.
- The implementation is also dependent on the instrumentation used.

- The sensors have been black-boxed, which can also have an impact on the response when factored in, which has the potential to sway the best controller towards the PID controller.

| Parameter                | Weight | PID controller | RL controller | Winner |
|--------------------------|--------|----------------|---------------|--------|
| Rise Time (%90)          | 20     | 1              | 2             | draw   |
| Settling Time (%2)       | 20     | 1              | 2             | RL     |
| Overshoot                | 10     | 1              | 2             | RL     |
| Complexity /2            | 10     | 2              | 1             | PID    |
| Disturbance Rejection /2 | 10     | 1              | 2             | RL     |
| Steady State Error /2    | 10     | 2              | 2             | draw   |
| Reference Tracking /2    | 10     | 1              | 2             | RL     |
| Frequency Response /2    | 10     | 1              | 2             | RL     |
| Score                    | 100    | 120            | 190           |        |

Figure 56: Controller Selection

- The closed-loop controlled system is hence given by equation 47.
- The system is 6<sup>th</sup> order.

$$T_{controlled} = \frac{704.1s^2 + 1805s + 1157}{5.9e-05s^6 + 0.02182s^5 + 2.597s^4 + 100.9s^3 + 746.3s^2 + 2225s + 1157}. \quad (47)$$

## 10 Analysis of Controlled System

The controlled closed-loop system has the transfer function :

$$T_{controlled} = \frac{704.1s^2 + 1805s + 1157}{5.9e - 05s^6 + 0.02182s^5 + 2.597s^4 + 100.9s^3 + 746.3s^2 + 2225s + 1157}. \quad (48)$$

### 10.1 Time Domain Response

Table 11 and Figure 57 give the time domain response of the closed-loop system.

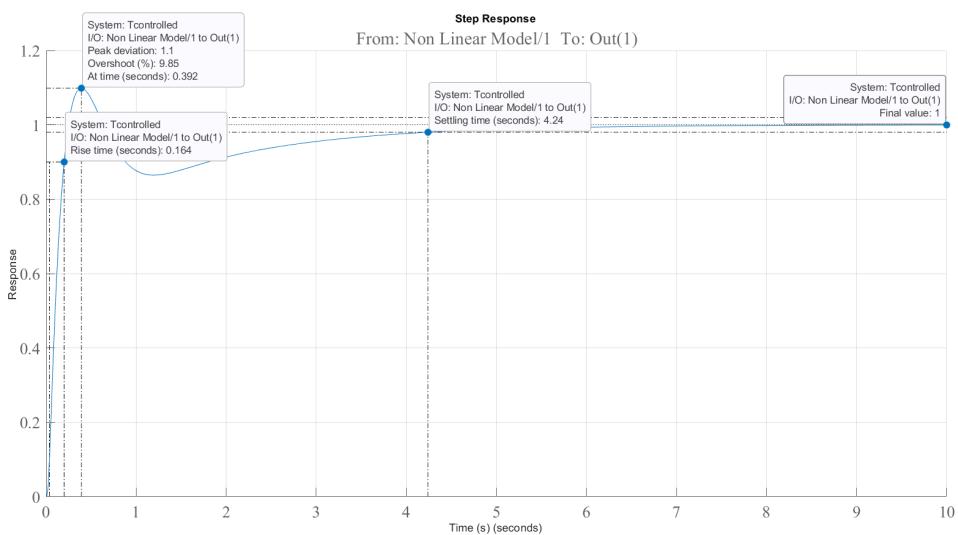


Figure 57: Controlled System Analysis: Step Response

Table 11: Step Response of Controlled System

| Parameter          | Value  |
|--------------------|--------|
| Rise time (s)      | 0.1641 |
| Transient time (s) | 4.2408 |
| Settling time (s)  | 4.2408 |
| Settling min       | 0.8647 |
| Settling max       | 1.0985 |
| Overshoot (%)      | 9.8529 |
| Undershoot         | 0      |
| Peak               | 1.0985 |
| Peak time (s)      | 0.3922 |

## 10.2 Roots and Pole Analysis

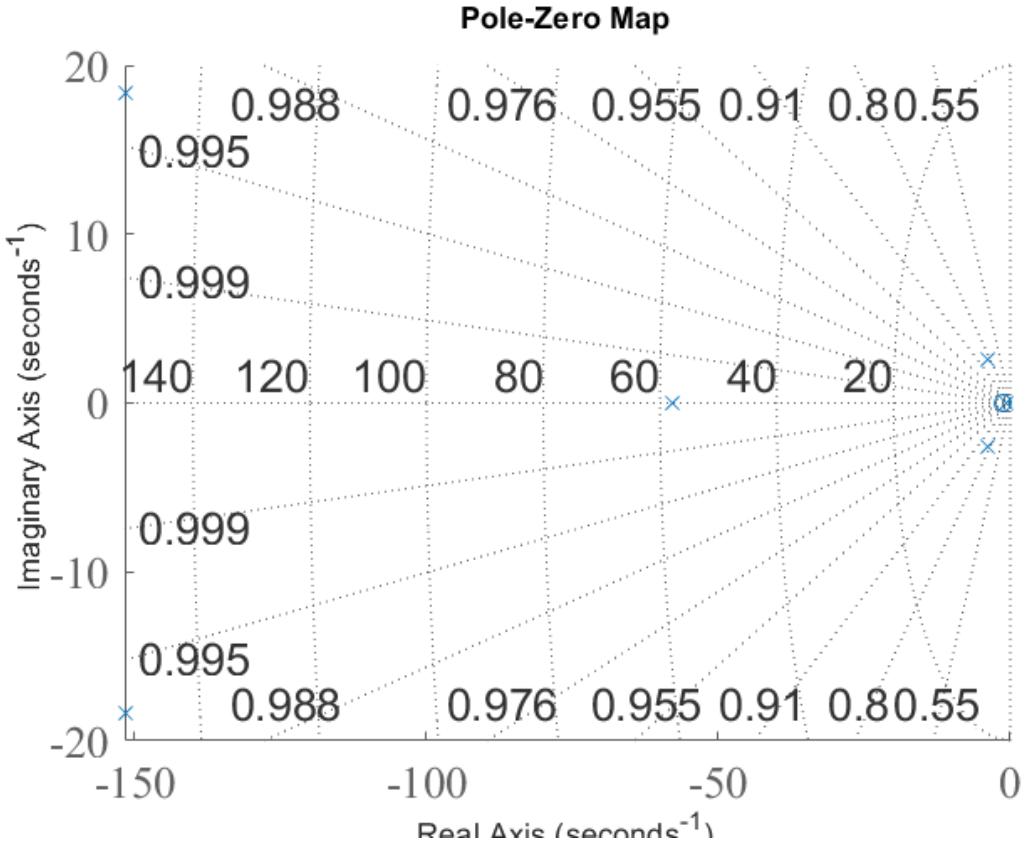


Figure 58: Controlled System Analysis: Pole-Zero Map

The poles and zeros of the closed-loop system are given by:

$$\text{Poles} = 10^2 \times \begin{cases} -1.5170 + 0.1836i, \\ -1.5170 - 0.1836i, \\ -0.5788 + 0.0000i, \\ -0.0394 + 0.0262i, \\ -0.0394 - 0.0262i, \\ -0.0065 + 0.0000i \end{cases} \quad (49)$$

$$\text{Zeros} = \begin{cases} -1.2821 + 0.0000i, \\ -1.2821 - 0.0000i \end{cases} \quad (50)$$

### 10.3 Frequency Domain Analysis

The Bode plot of the controlled system shows stability, as seen on Figure 59. The frequency domain specifications from the analysis are shown on Table 12.

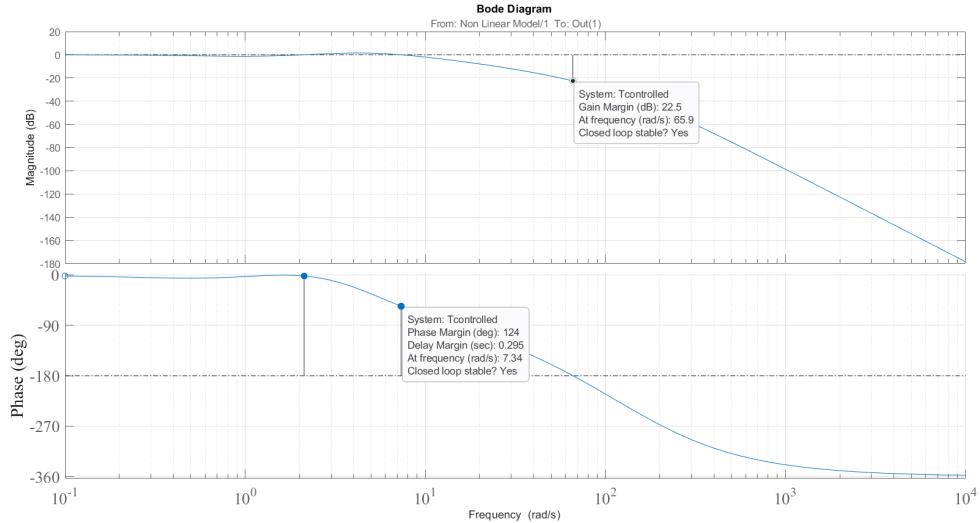


Figure 59: Controlled System Analysis: Bode Plot

Table 12: Frequency Response

| Specification                    | Value    |
|----------------------------------|----------|
| Gain margin (dB)                 | 13.393   |
| Phase margin (degrees)           | 123.9181 |
| Gain crossover frequency (rad/s) | 65.9254  |

## 11 Performance Specifications

Table 13 shows the summary of the solar tracker performance specification achieved by the use of the Root Locus controller

Table 13: Summary of Performance Specification

| Specification                    | Value    |
|----------------------------------|----------|
| Rise time (s)                    | 0.1641   |
| Transient time (s)               | 4.2408   |
| Settling time (s)                | 4.2408   |
| Settling min                     | 0.8647   |
| Settling max                     | 1.0985   |
| Overshoot (%)                    | 9.8529   |
| Undershoot                       | 0        |
| Peak                             | 1.0985   |
| Peak time (s)                    | 0.3922   |
| Gain margin (dB)                 | 13.393   |
| Phase margin (degrees)           | 123.9181 |
| Steady-state error               | 0        |
| Gain crossover frequency (rad/s) | 65.9254  |

To help achieve the performance specification shown on Table 13, instrumentation for implementation is required. The required instrumentation is discussed on following subsection

### 11.1 Instrumentation Needed for Implementation

In order to implement the controller fully, the following instrumentation is required:

#### 11.1.1 Position Sensor / Encoder

Position sensors are needed to measure the angular position of the solar tracker and provide feedback to the controller. The position sensor needs to have a high resolution to ensure precise tracking and an adequate range to cover the full range of motion of the solar tracker. A high resolution position sensor that is applicable in this case is shown on Figure 60.

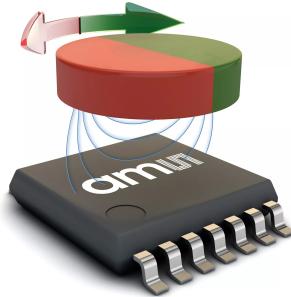


Figure 60: AS5048A High-Resolution Position Sensor [14]

Specifications for the position sensor are shown on Table 14.

Table 14: Position sensor specifications [14]

| Specification                              | Value |
|--|-------|
| Output resolution (Bit)                    | 14    |
| PWM frequency (kHz)                        | 1     |
| Output sampling rate (kHz)                 | 11.25 |
| System propagation delay ( $\mu\text{s}$ ) | 100   |

### 11.1.2 Light Sensor

Light sensors assist in dynamically adjusting the tracker based on sunlight intensity and direction, optimizing performance under varying conditions. The output of the light sensor will essentially inform the user as to whether the panel is obtaining maximum solar energy generation at the current position. The sensor should match the solar spectrum and have high sensitivity to detect low light levels, in the case of a day with overcast conditions. A light sensor which can be used is shown on Figure 61.

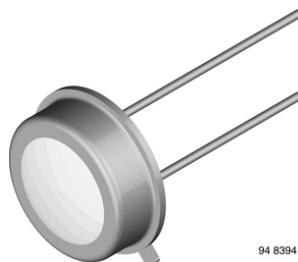


Figure 61: BPW21R Silicon Photodiode [15]

Specifications for the light sensor are shown on Table 15.

Table 15: Light Sensor Specifications [15]

| Specification                       | Value      |
|-------------------------------------|------------|
| Power dissipation (mW)              | 300        |
| Operating temperature range (°C)    | -40 to 125 |
| Range of spectral bandwidth (nm)    | 420 to 675 |
| Wavelength of peak sensitivity (nm) | 565        |
| Forward voltage (V)                 | 1          |

### 11.1.3 Microcontroller

A microcontroller is needed to process the input from sensors and control the actuators. They are essential for the real-time operation of the solar tracker, ensuring timely and accurate adjustments of the solar panels. The microcontroller should also have an adequate number of I/O ports for connecting sensors and actuators. The Arduino Mega 2560 Rev3 is a microcontroller that can be used. This is shown on Figure 62.



Figure 62: Arduino Mega 2560 Rev3 Microcontroller [16]

Specifications for the microcontroller are shown on Table 16.

Table 16: Light Sensor Specifications [16]

| Specification               | Value |
|-----------------------------|-------|
| Operating voltage (V)       | 5     |
| Clock speed (MHz)           | 16    |
| DC current per I/O pin (mA) | 20    |
| Number of digital I/O pins  | 54    |
| Number of analog input pins | 16    |

## 11.2 Strategies for Managing Unmeasurable States

State observers are algorithms that estimate the internal states of a dynamic system, in this case the solar tracker, using measurable outputs and a mathematical model of the system [17]. They are highly useful because they provide accurate estimates of states that cannot be directly measured, enabling effective control of the system.

### 11.2.1 Kalman Filter

Since the modelled solar tracker is exposed to noise and uncertainties and is linear, the Kalman filter is an ideal solution. The Kalman filter assumes that both process and measurement noise are Gaussian and updates state estimates iteratively using a prediction-correction mechanism [18]. During the prediction step, the filter uses the system model to forecast the next state and its uncertainty. In the correction step, it updates these predictions based on new measurements, weighted by the Kalman gain, which minimizes the mean squared error of the estimates [18]. The Kalman filter is highly effective in various applications, including navigation, tracking, robotics, and signal processing, due to its robustness in handling noisy and incomplete data. [19]

### 11.2.2 Extended Kalman Filter

When implementing the solar tracker in the real world, the system is likely to be non-linear due to the varying environmental conditions and complex dynamics of the mechanical and electrical components involved. The Extended Kalman Filter (EKF) is essentially the non-linear counterpart of the Kalman Filter. The EKF linearises the non-linear model around the current state estimate, enabling it to apply the principles of the linear Kalman filter to a broader range of systems [20]. This approach involves computing Jacobians to approximate the system's behavior locally, ensuring the state estimates remain accurate even when system dynamics are not strictly linear [20]. This method is directly applicable when implementing the modelled solar tracker in real life. However, if the modelled system only remains theoretical then the linear Kalman Filter will suffice.

## 12 Discussion

The design and evaluation of the solar tracker on this project is centered on optimizing the controller performance to meet the specified time and frequency domain requirements. The primary objective of the system was to ensure high accuracy in tracking the position of the sun, thereby maximizing energy generation.

**Time Domain Analysis:** The non-linear and linear uncontrolled plants were tested in order to ensure that the linear equivalent produces the relatively the same results as the non-linear. This was done to ensure that the controller designed based on a linear equivalent system would still work properly on a non-linear system as the real-life systems are non-linear. The response from non-linear and linear was obtained using voltage inputs in the form of step, ramp, sinusoidal, white noise and impulse inputs. All the responses from these voltage inputs were similar, with less than 1% difference. This proved that the non-linear governing equation was successfully linearised and could be used going forth for the analysis of the system and controller design of the system.

Using the step input voltage, the performance of the uncontrolled system was obtained and compared with the desired performance. The exaggerated overshoot and the settling time from the uncontrolled plant demonstrated that without control, the system failed to meet the required performance specification, thereby necessitating the implementation of control.

**Stability Analysis:** The pole-zero plot analysis showed that the system is stable as the poles of the system are all on the left-hand side of the imaginary axis. The stability of the uncontrolled system was also confirmed using Routh-Hurwitz criterion and Nyquist plot. However, the stability of the system did not translate into acceptable performance. The zeros did not contribute into improving the transient response of the system as there were no zeros. This analysis proved that having an inherently stable systems does not translate to having a system that meets the desired performance, hence necessitating control.

**Frequency Domain Analysis:** In the frequency domain, the performance analysis of the system was carried out using the Bode plot. The plot provided insights about the gain and phase margins of the system. The Bode diagram showed that while the system is stable, it did not meet the desired performance specifications, specifically, the gain crossover frequency and phase margin indicated that the system could not adequately reject disturbances or achieve bandwidth for desired performance. These deficiencies highlighted the limitation of the uncontrolled system in the frequency domain and indicated the need for control to enhance the responsiveness and stability of the system.

**Controller Design Performance:** Given that the system was stable but did not meet the desired performance requirements, several controllers were designed, i.e. P, PD and PI controllers. The P controller improved the stability but had much higher settling time and overshoot that were much higher than the desired values, hence the PD controller was designed. The introduction of the derivative gain reduced the rise time further and improved the settling time from 49.49 seconds to 0.92 seconds which was well within the desired 5 seconds. While the rise and settling time improved, the overshoot also improved from 94.42% to 15%. However, this was still higher than the required 10%. As a result, the PI controller was designed. The introduction of the integral gain improved the overshoot to 5.24% which was within the desired 10%. However, the rise time and settling time were compromised as they did not meet the desired performance. The discussed controllers proved to improve certain performance while compromising other performance which led to an overall unsatisfactory performance. This led to the design of the PID controller which satisfied all the requirements, with a rise time of 0.16 seconds, settling time of 3.72 seconds, and an overshoot of 8.57%. While the PID controller satisfied the desired performance, it had to be compared to an alternative control technique so that an ideal controller can be picked for the solar tracker, in this case, the Root Locus compensator was the alternative controller. The Root Locus controller was tuned until it met the desired requirements, i.e rise time of 0.16 seconds, settling time of 4.24 seconds and an overshoot of 9.85 %. Table 17 summarises the comparison of the controllers performance.

Table 17: Comparing candidate controller performance to desired performance

| Parameter         | Desired | PID    | Root Locus |
|-------------------|---------|--------|------------|
| Rise time (s)     | 2       | 0.1645 | 0.1641     |
| Settling time (s) | 5       | 3.7219 | 4.2408     |
| Overshoot(%)      | 10      | 8.5661 | 9.8529     |

While both the PID and Root Locus controllers both satisfied the desired requirements. These were further compared as candidate controllers to look beyond rise time, settling time and overshoot to select the most suitable control technique for the solar tracker.

The evaluation was only done after it was proven that both these candidate controllers meet the specified desired performance requirements. Both the controllers were assessed based on their step, ramp and sinusoidal response characteristics, alongside with their capabilities to reject disturbance, in this, case the wind force.

- **Step Response:** The Root Locus controller showed a superior step response compared to the PID controller, this was demonstrated by faster response times and better overall

performance without disturbances. However, the PID controller showed better disturbance rejection capabilities compared to the Root Locus controller when subjected to the step input.

- **Ramp Response:** Both the PID and the Root Locus controllers performed well in following the reference signal. However, the PID controller demonstrated a faster recovery and better disturbance rejection, returning to the desired path more effectively than the Root Locus controller.
- **Sinusoidal Response:** Both the PID and the Root Locus controllers effectively followed the sinusoidal path which was crucial for accurately tracking the movement of the sun. The Root Locus controller excelled at higher frequencies, maintaining better phase and gain margins. It also outperformed the PID controller in terms of disturbance rejection under sinusoidal inputs.

### Selection Criteria and Practical Considerations

The selection of the optimal controller was guided by a weighted scoring system, evaluating each controller based on:

- **Stability:** Stability was an important factor for evaluating the controllers. The Root Locus controller provided better stability margins, this ensured that the system provided adequate rejection against disturbances while operating. This behavior against the disturbances was a critical requirement for a solar tracker exposed to varying environmental factors.
- **Accuracy:** High accuracy in tracking was a fundamental requirement in ensuring maximising solar energy generation. The Root Locus controller maintained tracking accuracy within  $1^\circ$  of the position of the sun. This was essential for optimizing the incident sunlight on the solar panels.
- **Responsiveness:** In terms of rise time and settling time, responsiveness was an important selection criterion. The Root Locus controller exhibited a great ability to achieve a rise time within 2 seconds and a settling time within 5 seconds. This ensured that the solar tracker swiftly adjusted to changes which, in turn, maximised the solar energy capture.
- **Complexity:** While the PID controller offered a simpler and straight-forward implementation, the complexity of the Root Locus was justified by its enhanced performance.

This complex effort in tuning the response of the system was deemed worthwhile for the gains achieved in stability and accuracy.

Despite the simpler implementation of the PID controller, the Root Locus controller was ultimately selected as the optimal controller due to its superior performance in disturbance rejection and maintenance of minimal overshoot.

The practical aspects of the controller design were also considered. The implementation of PID involved tuning the proportional ( $K_p$ ), Integral ( $K_i$ ), and derivative ( $K_d$ ) gains to achieve the desired balance between steady-state error correction, transient response, and overall stability. The Root Locus controller required careful placement of the poles and zeros to optimize the performance of the system across different operating conditions.

The Root Locus controller enabled the system to meet and exceed the specified performance criteria. This highlighted the effectiveness of the selected controller and the overall mechatronic approach. The key performance requirements achieved by the use of the Root Locus controller are summarised on Table 18.

Table 18: Performance Summary using Root Locus Controller

| Parameter              | Desired    | Root Locus | Satisfactory/Unsatisfactory |
|------------------------|------------|------------|-----------------------------|
| Rise time (s)          | 2          | 0.1641     | Satisfactory                |
| Settling time (s)      | 5          | 4.2408     | Satisfactory                |
| Overshoot(%)           | 10         | 9.8529     | Satisfactory                |
| Steady-state error (%) | $\leq 0.5$ | 0          | Satisfactory                |
| Phase margin (degrees) | $\geq 45$  | 123.9181   | Satisfactory                |
| Gain margin (dB)       | $\geq 10$  | 13.393     | Satisfactory                |

## 12.1 Impact of Model Assumptions on Solar Tracker Performance

Although the system exhibited excellent performance with the Root Locus compensator, there were several assumptions made in Section 2 to simplify the model. These assumptions introduce limitations that could affect the system's real-world performance. Table 19 summarises the impact of each assumption on the system's real-world performance.

Table 19: Impact of Model Assumptions on System Performance

| <b>Assumption</b>  | <b>Limitation</b>   |
|--|---|
| Energy storage and discharge in batteries or capacitors are assumed to occur instantaneously.                        | This disregards real-world dynamics such as charge/discharge rates and energy losses, which can impact the efficiency and longevity of the energy storage system. In reality, delays and inefficiencies in energy transfer can reduce overall system performance. |
| The electrical network connected to the solar panels is assumed to behave as an ideal voltage or current source.     | Real electrical networks have impedance and experience voltage fluctuations. These factors can cause energy losses and affect the system's ability to efficiently convert and store energy.   |
| Mechanical components are treated as rigid bodies.   | In reality, mechanical components experience deformation and wear over time. Ignoring these factors can lead to inaccuracies in the model and unexpected maintenance issues.  |
| The moment of inertia of the panels remains constant.  | The moment of inertia may change with the panel's orientation and accumulated dirt or debris, affecting the tracking accuracy of the solar tracker.   |
| Rotational stiffness is negligible, and there is a specific damping coefficient between the motor and solar tracker. | Real systems have rotational stiffness and varying damping properties, which can affect the stability and responsiveness of the tracker.  |
| The disturbance wind force acts at a specific point from the motor shaft.  | In practice, forces may be distributed across the solar panels, leading to more complex stress and strain distributions that can impact system durability and performance.  |

## 13 Conclusion

The focus of this report was on the design, implementation, and evaluation of control strategies for a solar tracking system. After careful evaluation of possible controllers, the performance of PID (Proportional-Integral-Derivative) and Root Locus controllers were compared in guiding the solar panels to track the path of the Sun for maximum energy generation. The evaluation was based on several criteria, including step response, ramp response, sinusoidal response, and disturbance rejection capabilities.

The PID controller was found to be particularly effective in disturbance rejection and recovery. It maintained the desired path alignment under a variety of conditions, demonstrating robustness. This made it a suitable choice for applications where simplicity and quick adjustment were essential.

On the other hand, the Root Locus compensator outperformed the PID controller in terms of stability and tracking accuracy. This controller exhibited superior phase and gain margins, making it highly effective at handling higher frequency inputs and providing robust performance against disturbances. These characteristics were crucial for maximizing solar energy generation by ensuring precise orientation with the sun.

The viability of implementation complexities and accuracy of performance were investigated for both controllers. After thorough analysis and evaluation, the Root Locus compensator was selected as the preferred solution due to its enhanced stability, accuracy, and responsiveness. Although its implementation was more complex, the benefits significantly outweighed these complexities. Hence, the performance justified its selection. This controller ensured that the solar tracker could quickly and accurately adjust to changes, even in the presence of wind disturbances in the position of the sun, optimizing energy generation throughout the day.

The detailed evaluation of PID and Root Locus controllers demonstrated the importance of choosing a control strategy that offers a balance of stability, accuracy, and responsiveness. This was testament of the fact that no perfect solution exists. A compromise always exists - in this case, it was the implementation complexities of the Root Locus controller.

## References

- [1] Government of South Africa. *Tourism on rise of tourist visits in South Africa*. Accessed on 27 July 2023. 2023. URL: <https://www.gov.za/news/media-statements/tourism-rise-tourist-visits-south-africa-27-jul-2023>.
- [2] Oena Lodges. *Loadshedding in South Africa and how does it affect our guest experience*. Accessed on [Insert Access Date]. 2023. URL: <https://www.oena.co.za/loadshedding-in-south-africa-and-how-does-it-affect-our-guest-experience/>.
- [3] K. Zipp. *How does a solar tracker work?* Accessed on [Insert Access Date]. 2013. URL: <https://www.solarpowerworldonline.com/2013/04/how-does-a-solar-tracker-work/>.
- [4] J. Lassio et al. ‘Environmental life cycle-based analysis of fixed and single-axis tracking systems for photovoltaic power plants: A case study in Brazil’. In: *Cleaner Engineering and Technology* 11 (2022), p. 100586. DOI: [10.1016/j.clet.2022.100586](https://doi.org/10.1016/j.clet.2022.100586).
- [5] J. Ackermann et al. ‘Linear and Nonlinear Controller Design for Robust Automatic Steering’. In: *IEEE Transactions on Control Systems Technology* 3.1 (1995), pp. 90–98. DOI: [10.1109/87.370621](https://doi.org/10.1109/87.370621).
- [6] Unknown. *250W 3LARS Solar DC Borehole Pump*. Water Pumps Online. 2024. URL: <https://waterpumpsonline.co.za/water-pump-online-store/solar-dc-bh-3lars1-0-50-24-250w/>.
- [7] *GPS Elevation*. Accessed on 27 March 2024.
- [8] Solar Way Suppliers. *Trina Solar 425W Solar Panel Mono*. Accessed on 2024-05-11. Year of Access. URL: <https://www.solarwaysuppliers.co.za/product/trina-solar-425w-solar-panel-mono/>.
- [9] K. A. M. Waleed I. Hameed. ‘Speed Control of Separately Excited DC Motor Using Fuzzy Neural Model Reference Controller’. In: *International Journal of Instrumentation and Control Systems (IJICS)* 2.4 (2012), pp. 27–39. DOI: [10.5121/ijics.2012.2403](https://doi.org/10.5121/ijics.2012.2403).
- [10] National Aeronautics and Space Administration. *Dynamic Pressure*. Page Last Updated: April 4, 2024. 2024. URL: <https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/dynamic-pressure-2/#:~:text=Although%20pressure%20itself%20is%20a,the%20same%20in%20all%20directions>.
- [11] U. Today. *Beaufort Scale*. Space and astronomy news. Unknown. URL: <https://www.universetoday.com/89174/beaufort-scale/>.
- [12] E. Mohamed et al. ‘Design and Implementation of Sun Tracking System’. In: 2016.

- [13] H. Bansal. ‘Tuning of PID Controllers using Simulink’. In: *International Journal of Mathematical Modeling, Simulation and Applications* (2009), pp. 337–344.
- [14] ams-OSRAM AG. *ams AS5048A High-Resolution Position Sensor*. [Online; accessed 2024-05-19]. 2018.
- [15] Vishay. *BPW21R Photo Detectors*. <https://www.vishay.com/en/product/81519/>. [Online; accessed 2024-05-19]. 2021.
- [16] . *Mega 2560 Rev3*. <https://docs.arduino.cc/hardware/mega-2560/>. [Online; accessed 2024-05-19].
- [17] L. A. Liu, D. Wei and Y. Li. *Handbook of Research on Computational and Systems Biology: Interdisciplinary Applications*. [Online; accessed 2024-05-19]. IGI Global, 2011.
- [18] W. Franklin. ‘Kalman Filter Explained Simply’. In: *The Kalman Filter* (2020). [Online; accessed 2024-05-19].
- [19] L. Shi, K. H. Johansson and R. M. Murray. ‘Kalman Filtering with Uncertain Process and Measurement Noise Covariances with Application to State Estimation in Sensor Networks’. In: *2007 IEEE International Conference on Control Applications*. [Online; accessed 2024-05-19]. IEEE. 2007.
- [20] K. Kitani. *Extended Kalman Filter - 16-385 Computer Vision*. Tech. rep. [Online; accessed 2024-05-19]. Carnegie Mellon University, pp. 1–20.

## 14 Appendix

### 14.1 Model Constants

---

```
1 clear;clc;close all;
2 J=25.5759;%Moment of Inertia of the system
3 c=8; %damping constant of the system
4 k=0; % stiffness constant of the system
5 R=0.5; %Resistance of the motor
6 L=5e-3;%Motor inductance
7 kt=4; %Motor constant
8 ke=0.00083; %back emf constant
9 I=19e-10; %Motor rotor inertia
10 bm=0.00123;%motor damping constant
11 m=98.87; %m [kg]
12 g=9.81; %gravitational constant m/s^2
13 d=0.11055; % [m] distance to pivot from cg.
14 q=0.11055; % [m] moment arm for disturbance (pseudo random)
15 Vmax=230; % Volts
16 length_tracker=2.268;%Solar trakcer length [m]
17 width_tracker=1.758; %Solar tracker width [m]
18 As=length_tracker*width_tracker; %Solar tracker area [m^2]
19
20
21
22 h_tracker; %Solar tracker area [m^2]
```

---

## 14.2 Compare Linear and Non-Linear

---

```
1 clc;close all;
2
3 compare=out.compare_linear;
4
5 time=compare.time;
6
7 Voltage=compare.signals(1).values;
8 Non_Linear=compare.signals(2).values;
9 Linear=compare.signals(3).values;
10 Diff=100*(Non_Linear-Linear)./Non_Linear;
11
12
13 figure;
14 subplot(3,1,1);
15 plot(time,Voltage);
16 xlabel('Time (s)');
17 ylabel('Voltage (V)');
18 set(gca,'FontSize',16,'FontName','Times New Roman');
19 box off;
20
21 subplot(3,1,2);
22 plot(time,Non_Linear,'r',time,Linear,'--b');
23 xlabel('Time (s)');
24 ylabel('\theta');
25 legend('Non-Linear','Linear');
26 set(gca,'FontSize',16,'FontName','Times New Roman');
27 box off;
28
29 subplot(3,1,3);
30 plot(time,Diff);
31 xlabel('Time (s)');
32 ylabel('% Difference [deg]');
33 set(gca,'FontSize',16,'FontName','Times New Roman');
34 box off;
35 ylim([-1 1]);
```

### 14.3 Controller Analysis

```
1 clc;close all;
2 load('simulink_linearisation_output')
3 %Generate transfer function
4 Gs=tf(linsys1);disp(Gs);
5 K_d= 25.626;
6 K_p= 69.600216;
7 K_i= 47.258547;
8 time=0:0.01:10;time=time';
9 %Uncontrolled System, Closed loop
10 To=feedback(Gs,1);
11 %P controller Only
12 G_p=pid(50,0,0);
13 T_p=feedback(G_p*Gs,1);
14 figure;
15 step(T_p,time);
16 xlabel('Time (s)');
17 ylabel('Response');
18 legend('P');
19 box off;
20 set(gca,'FontSize',16,'FontName','Times New Roman');
21 T_p_out=stepinfo(T_p)
22 %extract struct to table
23 T_p_out_table=struct2table(T_p_out);
24
25 %% PD Controller 57.5271s+140.31
26 G_pd=pid(140.31,0,57.5271);
27 T_pd=feedback(G_pd*Gs,1);
28 figure;
29 step(T_pd,time);
30 xlabel('Time (s)');
31 ylabel('Response');
32 legend('PD');
33 box off;
34 set(gca,'FontSize',16,'FontName','Times New Roman');
35 T_pd_out=stepinfo(T_pd)
36 %extract struct to table
37 T_pd_out_table=struct2table(T_pd_out);
38
39
40 %%PI Controller
41 G_pi=pid(10,3,0);
42 T_pi=feedback(G_pi*Gs,1);
43 figure;
44 time_pi=0:0.01:300;time_pi=time_pi';
```

```

45 step(T_pi,time_pi);
46 xlabel('Time (s)');
47 ylabel('Response');
48 legend('PI');
49 box off;
50 set(gca,'FontSize',16,'FontName','Times New Roman');
51 T_pi_out=stepinfo(T_pi)
%extract struct to table
53 T_pi_out_table=struct2table(T_pi_out);
54
55 %%PID Controller
56 G_pid=pid(70,47,26);
57 T_pid=feedback(G_pid*Gs,1);
58 figure;
59 time_pid=0:0.01:10;time_pid=time_pid';
60 step(T_pid,time_pid);
61 xlabel('Time (s)');
62 ylabel('Response');
63 legend('PID');
64 box off;
65 set(gca,'FontSize',16,'FontName','Times New Roman');
66 T_pid_out=stepinfo(T_pid)
%extract struct to table
68 T_pid_out_table=struct2table(T_pid_out);
69
70 %%Root locus compensator
71 K=37;
72 N=[0.6084 1.56 1];
73 D=[0.000059 0.0159 1 0];
74 GRL=K*tf(N,D)
75 C=GRL
76
77
78 T_rl=feedback(GRL*Gs,1);
79 figure;
80 step(T_rl,time);
81 xlabel('Time (s)');
82 ylabel('Response');
83 legend('Root Locus');
84 box off;
85 set(gca,'FontSize',16,'FontName','Times New Roman');
86 T_rl_out=stepinfo(T_rl)
%extract struct to table
88
89 rlocus(T_rl)
set(gca,'FontSize',16,'FontName','Times New Roman');
91 grid on;
92 grid minor;

```

```

93
94
95 %Comapring root locus and PID
96 figure;
97 step(T_pid,time_pid,'-b');
98 hold on;
99 step(T_rl,time,'--r');
100 xlabel('Time (s)');
101 ylabel('Response');
102 legend('PID','Root Locus');
103 box off;
104 set(gca,'FontSize',16,'FontName','Times New Roman');

105
106
107 Tcontrolled=T_rl

108
109
110 %% Analysis of controlled system
111 figure
112 subplot(2,2,1);
113 %step bode plot
114 bode(Tcontrolled);
115 set(gca,'FontSize',12,'FontName','Times New Roman');
116 grid minor;

117
118 subplot(2,2,2);
119 %p-z map
120 pzmap(Tcontrolled);
121 set(gca,'FontSize',12,'FontName','Times New Roman');
122 box off;
123 grid minor;

124
125 subplot(2,2,3);
126 %Root locus
127 rlocus(Tcontrolled);
128 set(gca,'FontSize',12,'FontName','Times New Roman');
129 box off;
130 grid on;
131 grid minor;

132
133 subplot(2,2,4);
134 %Nyquist
135 nyquist(Tcontrolled);
136 set(gca,'FontSize',12,'FontName','Times New Roman');
137 box off;
138 grid on;
139 grid minor;
140

```

```

141
142 figure ;
143 %step response
144 step(Tcontrolled);
145 xlabel('Time (s)');
146 ylabel('Response');
147 set(gca,'FontSize',16,'FontName','Times New Roman');
148 box off;
149 grid minor;
150
151
152 figure;
153 %pzmap
154 pzmap(Tcontrolled);
155 set(gca,'FontSize',16,'FontName','Times New Roman');
156 box off;
157 grid minor;
158
159
160 figure;
161 %bode
162 bode(Tcontrolled);
163 set(gca,'FontSize',16,'FontName','Times New Roman');
164 box off;
165 grid minor;
166
167
168 figure;
169 %nyquist
170 nyquist(Tcontrolled);
171 set(gca,'FontSize',16,'FontName','Times New Roman');
172 box off;
173 grid minor;
174
175 % Roots
176 clc;
177 Denominator=Tcontrolled.denominator{1};
178 Roots=roots(Denominator)
179
180 Numerators=Tcontrolled.numerator{1};
181 Poles=roots(Numerators)
182
183
184 stepinfo(Tcontrolled)
185 figure
186 h=bodeplot(Tcontrolled);
187 info=getoptions(h);
188 sys=Tcontrolled;

```

```
189 % Compute and display frequency response specifications
190 [mag, phase, wout] = bode(sys); % Compute magnitude, phase, and frequency
191     values
192 [Gm, Pm, Wcg, Wcp] = margin(sys); % Compute gain margin, phase margin,
193     crossover frequencies
194 disp(['Gain Margin (dB): ', num2str(Gm)]);
195 disp(['Phase Margin (degrees): ', num2str(Pm)]);
196 disp(['Gain Crossover Frequency (rad/s): ', num2str(Wcg)]);
```

---

## 14.4 Disturbance Plot

---

```
1
2 clear;clc;close all;
3
4 solar_tracker_constants;
5
6 theta_deg=90;
7 u=0:0.01:(40*3.6);u=u';
8 Md=(0.5)*1.225*u.^2.*As*x;
9
10 figure;
11 plot(u,Md);
12 xlabel('Wind speed (m/s)');
13 ylabel('Disturbance Moment (Nm)');
14 box off;
15 grid on;
16 set(gca,'FontSize',16,'FontName','Times New Roman');
```

## 14.5 Linear Analysis

```
1 clc;close all;
2 load('simulink_linearisation_output')
3
4
5 %Generate transfer function
6 Gs=tf(linsys1);disp(Gs);
7 To=feedback(Gs,1);disp(To);
8
9
10 %Open Loop Analysis
11
12 %1. Step Info
13 stepinfo(To)
14 %1. Step Response
15 figure;
16 step(To);
17 xlabel('Time (s)');
18 ylabel('Voltage \theta');
19 set(gca,'FontSize',16,'FontName','Times New Roman');
20 box off;
21
22
23
24 %2. Pzmpap
25 figure;
26 pzmap(To);
27 set(gca,'FontSize',16,'FontName','Times New Roman');
28 box off;
29
30
31 %3. Roots
32 Denominator=To.denominator{1};
33 Roots=roots(Denominator)
34
35
36 %4. Bode Plot
37 figure;
38 bode(To);
39 set(gca,'FontSize',16,'FontName','Times New Roman');
40 box off;
41 grid minor;
42
43 %5. Nyquist Plot
44 figure;
```

```

45 nyquist(To);
46 set(gca , 'FontSize' ,16 , 'FontName' , 'Times New Roman');
47 box off;
48 grid minor;
49
50
51 % Rlocus
52 figure;
53 rlocus(To);
54 set(gca , 'FontSize' ,16 , 'FontName' , 'Times New Roman');
55 box off;
56 grid minor;
57
58
59
60
61 figure
62
63 t=0:0.001:(12*3600); t=t';
64 Rs=((pi)/(12*3600))*t-(pi/2);
65 Am=pi/2;
66 alpha= pi/(12*3600);
67
68
69 Rs_sine=Am*sin(alpha*t)-(pi/2);
70 %fin index of t=6*3600
71 fin_index=find(t==6*3600);
72 Rs_sine(fin_index:end)=-1*Rs_sine(fin_index:end);
73 figure;
74 plot(t/3600,Rs*180/pi);hold on;
75 plot(t/3600,Rs_sine*180/pi);hold on;
76 xlabel('Time (h)');
77 ylabel('\theta ^o');
78 set(gca , 'FontSize' ,16 , 'FontName' , 'Times New Roman');
79 box off;
80 grid minor;
81 legend('R(s)_ramp','R(s)_sine');

```

## 14.6 PID Tuning

---

```
1 clc;close all;
2 load('simulink_linearisation_output')
3 %Generate transfer function
4 Gs=tf(linsys1);disp(Gs);
5 To=Gs
6
7 controlSystemDesigner(Gs)
```

## 14.7 Running Animations

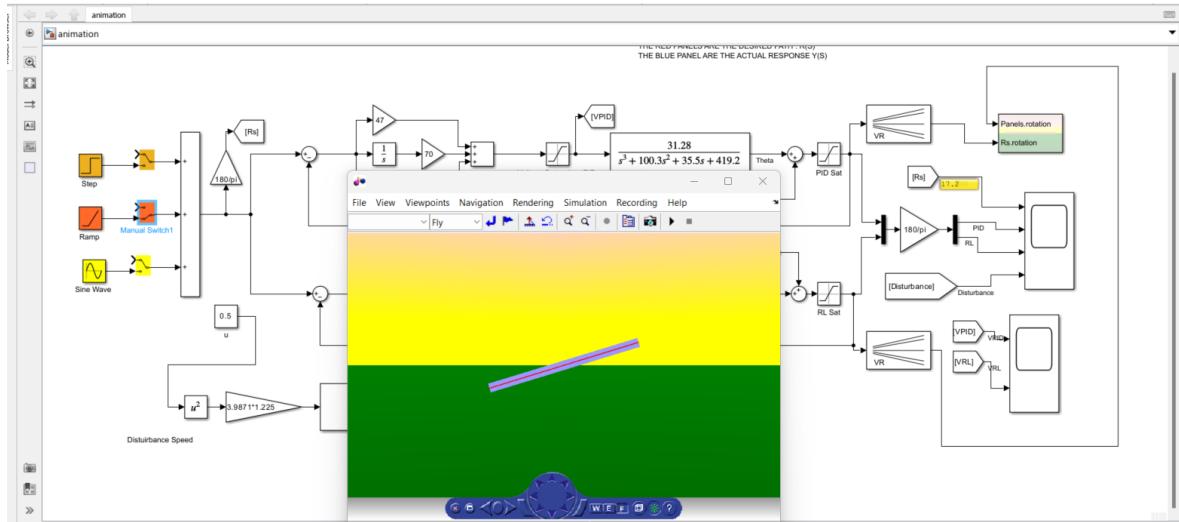


Figure 63: Animations Setup

- Red panels represent the desired position or reference  $R(s)$ .
- Blue panels represent the actual path  $Y(s)$ .
- Set the desired signal and disturbance speed.

## 14.8 Linearisation Output

### 14.8.1 State Space

General Information:

Operating point: Model initial condition

Size: 1 inputs, 1 outputs, 3 states

Linearization Result:

A =

|    | x1     | x2      | x3     |
|----|--------|---------|--------|
| x1 | -100   | -0.166  | 0      |
| x2 | 0.1564 | -0.3128 | -4.192 |
| x3 | 0      | 1       | 0      |

B =

|    | u1  |
|----|-----|
| x1 | 200 |
| x2 | 0   |
| x3 | 0   |

C =

|    | x1 | x2 | x3 |
|----|----|----|----|
| y1 | 0  | 0  | 1  |

D =

|    | u1 |
|----|----|
| y1 | 0  |

Name: Linearization at model initial condition

Continuous-time state-space model.

Model Properties

State Names:

x1 - Integrator2

x2 - d(theta)//dt

x3 - theta

Input Channel Names:

u1 - Non Linear Model/1

Output Channel Names:

y1 - Non Linear Model/1

#### 14.8.2 Transfer Function

General Information:

Operating point: Model initial condition

Size: 1 inputs, 1 outputs, 3 states

Linearization Result:

From input "u1" to output "y1":

31.28

-----  
 $s^3 + 100.3 s^2 + 35.5 s + 419.2$

Name: Linearization at model initial condition

Continuous-time transfer function.

Model Properties

Input Channel Names:

u1 - Non Linear Model/1

Output Channel Names:

y1 - Non Linear Model/1

#### 14.8.3 Zero-Pole Gain

General Information:

Operating point: Model initial condition

Size: 1 inputs, 1 outputs, 3 states

Linearization Result:

From input "u1" to output "y1":

31.279

-----  
 $(s+100) (s^2 + 0.3131s + 4.192)$

Name: Linearization at model initial condition

Continuous-time zero/pole/gain model.

Model Properties

Input Channel Names:

u1 - Non Linear Model/1

Output Channel Names:

y1 - Non Linear Model/1