Systems Thinking Project

Team PeopleInDistress (PID)

M24 Systems Thinking

IIIT Hyderabad

M.P.Samartha
2023102038

Varun Ram Murty Shastry
2023112005

Srihari Padmanabhan
2023102021

Vedant Pahariya
2023112012

Mohd. Haris
2023102003

Sarvesh Takbhate
2023102039

Kiruba Pandi Selvakumar
2023112010

Siddarth Gottumukkula
2023102040

Contents

Chapter 1

Overview

Robotics has limitless applications in our constantly evolving world. Industry is one such sector where robotics has marked its niche and made a lasting difference. Throughout our discussions, it became clear that we all shared a common goal: to create a robot with meaningful real-world applications. After thorough brainstorming and careful consideration, we unanimously decided to move forward with a SCARA. In our project, we have made an attempt to develop a simplified SCARA.

The simplified SCARA has been designed to streamline an automated pick-and-place operation. The primary objective of the robot is to retrieve objects from a supply conveyor belt, classify them based on their mass, and efficiently sort them onto designated conveyor belts. Specifically, objects exceeding a predetermined mass threshold are directed to the first conveyor belt, while those meeting or falling below the threshold are placed onto the second. This system demonstrates both the precision and utility of SCARA units in handling mass-based sorting tasks in an industrial setting.

An advanced industrial SCARA has multiple degrees of freedom, both rotational and translation, allowing it a larger **work envelope**, minimized dead zone and higher precision. The simplified SCARA implemented in our project has two degrees of freedom, one translation and one rotation. Its work envelope is an annular cylinder (3 dimensional) . The control applied to this system is on the rotational section and the translation section of the robot.

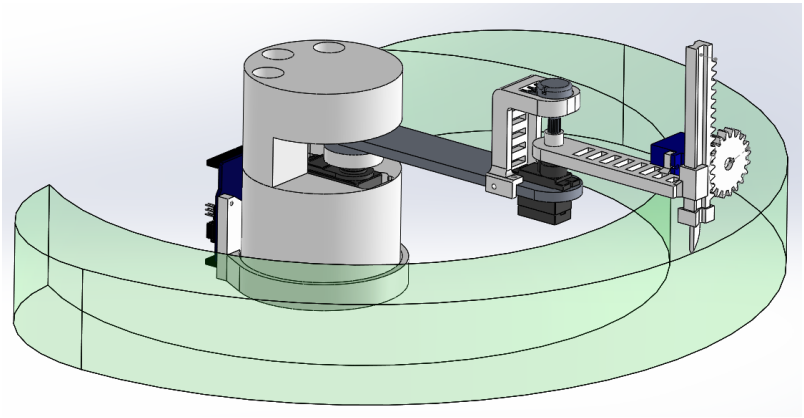

Figure 1.1: Operating envelope and dead zone of an industrial SCARA

This document covers essential aspects of designing a robot from a control theorist's perspective. The basics of a SCARA, our robot model, the equations from the first principles, state space representations, different controller actions, simulation results and further improvements.

Chapter 2

SCARA

2.1 What is a SCARA ?

SCARAs are one of the most popular and easy-to-use industrial robotic arms. They are commonly used in a variety of industries with different end-effectors, often for manufacturing and assembly applications (as need arises). Industrial SCARAs usually feature 4 degrees of freedom.

Figure 2.1: Range of motion of an industrial SCARA

2.2 Motion and Control of an industrial SCARA

- **Horizontal (X-Y) Plane Movement:** The two rotary joints allow the arm to move across a two-dimensional plane, similar to sweeping motions. The arm's compliance in this plane ensures precision during tasks like assembly, pick-and-place, or packaging.
- **Z-axis Movement:** The vertical movement is used to either lift or lower objects. The SCARA's stiffness in the vertical direction ensures accuracy, especially for tasks that require downward force like inserting a part into a hole.
- **End-Effector:** This is typically a gripper, suction, or tool used for manipulation. It moves according to the position and orientation determined by the rotational joints.

2.3 Applications of SCARA

- **Pick-and-Place:** SCARA are widely used for moving objects quickly and precisely from one location to another.
- **Assembly:** They are suitable for tasks such as screwing or inserting parts where high precision is required.
- **Packaging:** SCARA are often used in industries for high-speed sorting and packaging tasks.

- **Testing and Inspection:** In some cases, SCARA can be equipped with sensors or cameras to automate testing and inspection processes.
- **Household applications** : With an appropriate end-effector and control law, the bot can be used for daily tasks like sewing cloths or performing basic household tasks.

2.4 Limitations

- **Limited Range in Z-axis:** Since the SCARA is optimized for horizontal movement, its vertical reach is typically limited compared to articulated robots.
- **Work Envelope:** The working area is confined to a cylindrical space, which may not be suitable for tasks that require 3D movement in all directions.

In our project, we have restricted the rotary gears in the XY plane simplifying the degrees of freedom to be translation and rotational along the z-axis.

Chapter 3

Mathematical Foundation

We deal with the two degrees of freedom independently in our analysis. The variables that influence the system are listed below.

- Gravity : Gravitational force acts on the entire system and the block picked up throughout the process.
- Viscous Drag : A dissipative force acts on the bodies proportional to their velocities. **A major assumption considered throughout our analysis is that the drag force experienced by the system is independent of the geometry of the system**. This reductionist approach to our problem statement idealizes our simulations to an extent. Nevertheless, the essence of the system is not lost, enabling us to make this engineering approximation.
- Input force : Applied to the gears at the degrees of freedom.
- Masses and lengths of relevant components.

We will now define the structure and nomenclature used hereon to refer to it. A cylindrical 'base' is present which supports a horizontal 'rod'. This rod rotates in its plane about the point where it is attached to the base. At the end of the rod, there is a vertical 'arm' which moves along the z-axis. The end of this arm has an end-effector which is the point of contact between the robot and the physical work environment.

In case of Pick-and-Place SCARA, the end effector used in the latest robots is *suction-based*. A suction pump is the end effector which picks up the packages and sorts them as desired. In our project, the robot receives the package through a supply conveyor belt and places it on one of two different conveyor belts based on the weight of the package. Our implementation assumes that there are several systems that are in place to calculate the following quantities and provide them to the robot. This is not a unrealistic assumption as a few sensors can be used to calculate these quantities quickly, which have not been modelled in our simulations.

- Mass of the package
- Height between the end-effector and the package when placed on different conveyor belts.

Let us start by modelling the rotational degree of freedom mathematically. Rotating the base along with the rod will only increase the power required to rotate the system and serves no additional purpose. Thus, we rotate only the rod and not the base.

3.1 Deriving θ

Using first principles, we can write the governing law of the system. This is effectively constraining the rotatory motion of the robot. We know that the **vector sum of all torques acting on the robot must be proportional to its angular acceleration**.

- L is the length of the rod. In our Simulink simulations, we have called it L_{rod}
- $\theta, \dot{\theta}, \ddot{\theta}$ are the angular position, angular velocity and angular acceleration of the block respectively.
- M_{arm} is the mass of the robot's arm
- m is the mass of the block
- τ is the torque by virtue of the force applied by the controller.

- $\tau_{visc} = \tau_{\text{rod}_{\text{visc}}} + \tau_{\text{block}_{\text{visc}}}$ is the net viscous force acting on the rotating robot-block system.
- B is the damping coefficient. It is essentially the proportional constant introduced in the relation $F_{drag} \propto -v$, i.e, between the drag force acting on the block and its velocity. Instead of considering rotational viscous drag, we have modelled using the coefficient of translation viscous drag itself by a few manipulations.
- $M_{total} := M_{arm} + m$

A rough sketch is provided below for a simple visualisation

$$\sum_{\forall i} \tau_i = I\ddot{\theta}$$

$$\begin{aligned} I\ddot{\theta} &= \tau - (\tau_{\text{rod}_{\text{visc}}} + \tau_{\text{block}_{\text{visc}}}) \\ &= \tau - \left(\frac{1}{4}B\dot{\theta}L^2 + B\dot{\theta}L^2\right) \\ \left(M_{total} + \frac{1}{3}M_{rod}\right)L^2\ddot{\theta} &= \tau - \left(\frac{5}{4}BL^2\right)\dot{\theta} \end{aligned} \tag{3.1.1}$$

Taking the Laplace Transform of this equation, with the assumption of zero initial conditions, we get:

$$\left(M_{total} + \frac{1}{3}M_{rod}\right)L^2s^2 + \frac{5}{4}BL^2s = \frac{\tau(s)}{\theta(s)}$$

We thus obtain the transfer function $G_1(\mathbf{s})$:

$$\boxed{G_1(s) = \frac{\theta(s)}{\tau(s)} = \frac{1}{L^2} \cdot \frac{1}{s^2\left(M_{total} + \frac{M_{rod}}{3}\right) + \frac{5B}{4}s}} \tag{3.1.2}$$

Selecting the state variables as ϕ_1 and ϕ_2, such that:

$$\begin{aligned} \phi_1 &= \theta \\ \phi_2 &= \dot{\theta} \end{aligned} \tag{3.1.3}$$

From the equations 3.1.1 and 3.1.3, we get:

$$\begin{aligned} \dot{\phi_1} &= \dot{\theta} = \phi_2 \\ \dot{\phi_2} &= \ddot{\theta} = \frac{\tau}{\left(M_{total} + \frac{M_{rod}}{3}\right)L^2} - \frac{\frac{5}{4}B}{M_{total} + \frac{M_{rod}}{3}}\dot{\theta} \end{aligned}$$

So, we have the state space representation as below:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{5B}{4(M_{\text{total}} + \frac{1}{3}M_{\text{rod}})} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L^2(M_{\text{total}} + \frac{1}{3}M_{\text{rod}})} \end{bmatrix} \tau \tag{3.1.4}$$

The output equation is given below:

$$\theta = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \tag{3.1.5}$$

3.2 Deriving z_d

Using first principles, we can write the governing law of the system. This is effectively constraining the translation motion of the robot's arm. We know that the vector sum of all forces acting on the block must be proportional to its rate of change of momentum, i.e,

$$\sum_{\forall i} \mathbf{F_i} = M_{block}\ddot{z}$$

Here

$$-(M_{arm} + m)\mathbf{g} + \mathbf{F_{ext}} - B\dot{\mathbf{z}} = (M_{arm} + m)\ddot{\mathbf{z}}$$

- $\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}$ are the position, velocity and acceleration of the block respectively.
- M_{arm} is the mass of the robot's arm

- m is the block's mass
- $\mathbf{F_{ext}}$ is the force applied by the robot's arm on the block. This is equal to the input force applied by the controller.
- B is the damping coefficient. It is essentially the proportional constant introduced in the relation $\mathbf{F_{drag}} \propto -\mathbf{v}$, i.e, between the drag force acting on the block and its velocity.
- $M_{total} := M_{arm} + m$

Initial conditions of rest are established. Hence, we can conveniently use the Laplace Transform to find the transfer function of this system.

$$\left(s^2 M_{total} + Bs\right) Z(s) = \mathbf{F_{ext}} - \left(M_{total}\right) \mathbf{g}$$

Since the force $\left(M_{total}\right) g$ is always present on the system, let's define the RHS as offset force for ease of calculations, i.e, $\mathbf{F}_{\text{off}} := \mathbf{F_{ext}} - (M_{total})\mathbf{g}$

We thus obtain the transfer function $\mathbf{G_2(s)}$:

$$\boxed{G_2(s) = \frac{Z(s)}{F_{\text{off}}(s)} = \frac{1}{s^2 M_{total} + sB}} \tag{3.2.1}$$

Let's choose $\mathbf{x_1} = z$ and $\mathbf{x_2} = \dot{z}$ as state variables.

$$\dot{\mathbf{x_1}} = \mathbf{x_2} \tag{3.2.2}$$

$$\dot{\mathbf{x_2}} = \frac{-B}{M_{\text{total}}}\mathbf{x_2} + \frac{1}{M_{\text{total}}}\mathbf{F_{off}}$$

The state equation and the output equation are

$$\begin{bmatrix} \dot{\mathbf{x_1}} \\ \dot{\mathbf{x_2}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-B}{M_{\text{total}}} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M_{\text{total}}} \end{bmatrix} \mathbf{F_{off}} \tag{3.2.3}$$

The output equation is given below:

$$\mathbf{z} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} \tag{3.2.4}$$

3.3 Parameters

These are the fundamental governing set of equations in our system. We have chosen the values of the above defined parameters as follows

Parameter	Value
B	0.82 $\frac{Ns}{m}$
M_{rod}	10 Kg
M_{arm}	5 Kg
g	9.8 $\frac{m}{s^2}$
L	1 m

Table 3.1: Parameters

Substituting the values in the equations derived above, the transfer function $G_1(s)$ and the state space representation for θ (3.1.4) evaluate to:

$$G_1(s) = \frac{1}{(1)^2} \cdot \frac{1}{s^2\left((M_{arm} + m) + \frac{10}{3}\right) + \frac{5(0.82)}{4}s}$$

$$G_1(s) = \frac{1}{s^2\left(m + 8.33\right) + 1.025s}$$

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1.025}{(m+8.33)} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m+8.33} \end{bmatrix} \tau$$

And the transfer function $G_2(s)$, and state space representation for z_d (3.2.3) evaluate to expressions as below:

$$G_2(s) = \frac{1}{s^2(m+5) + (0.82)s}$$

$$\begin{bmatrix} \dot{\mathbf{x_1}} \\ \dot{\mathbf{x_2}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-0.82}{(m+5)} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{(m+5)} \end{bmatrix} \mathbf{F_{off}}$$

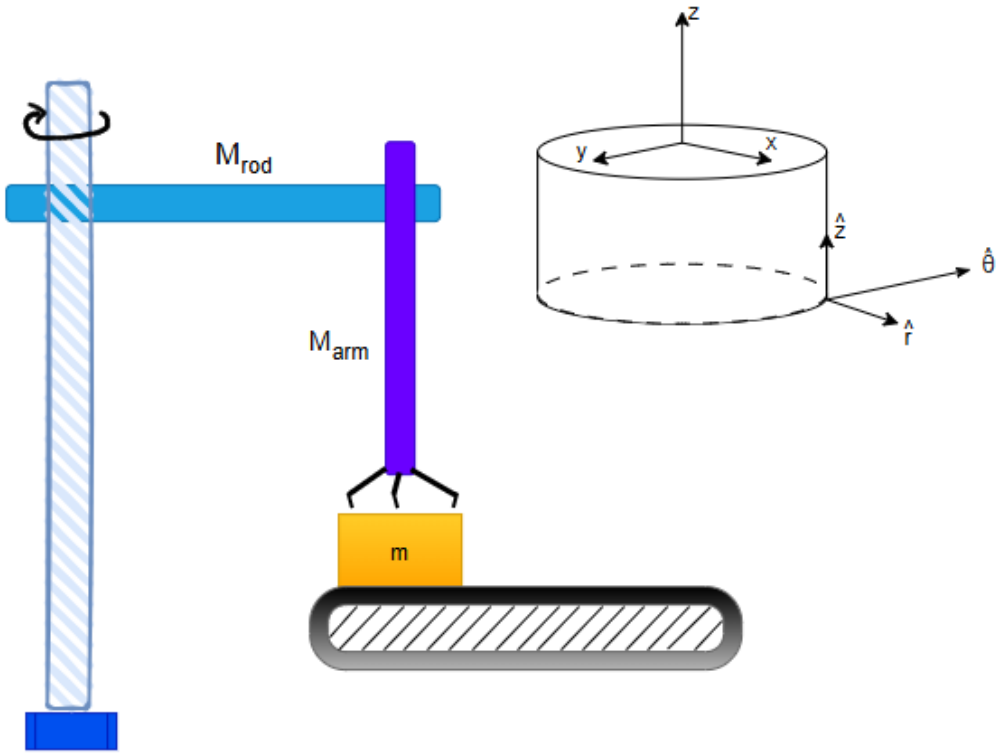

Figure 3.1: System Diagram

Chapter 4

Control Implementation in Simulink

Let us now analyze the control system used in the implementation of our project now. At both the degrees of freedom, control systems have been implemented, which operate independently. The system transfer function is given by the derivations done in Chapter 3. We add a controller block, which models the control law used. The block diagram is shown below.

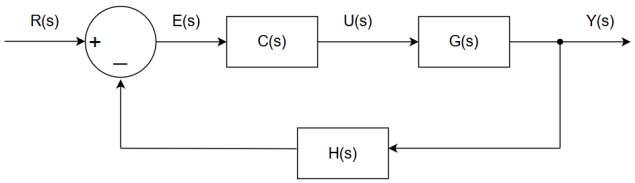

Figure 4.1: Generic Block Diagram

We have chosen to operate the system under unity feedback and now the block diagram changes as shown below.

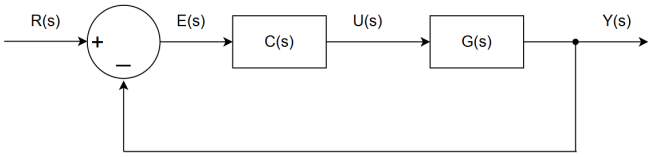

Figure 4.2: Block Diagram with Unity Feedback

This control flow is followed for rotational and translation motion. In case of the rotational DOF(degree of freedom), the input is the desired angular position and in case of the translation DOF, the input is the z-position.

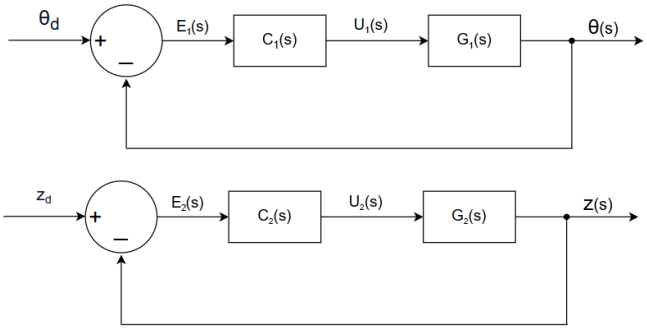

Figure 4.3: Block Diagrams for θ and z

Depending on the mass of the package, the robot will either turn by an angle θ_1 or by an angle θ_2 to place them on different conveyor belts. Thus, the value of θ_d changes according to the stage of operation. Similarly, z_d changes according to the mode of operation. The different stages of operation are specified below.

1. Initially, the bot must come to a *default* position which is when the rod is aligned with the supply belt and the arm is fully retracted. This is our reference position where $\theta = 0$ and $z = 0$.

2. In step 2, the arm will extend and attempt to pick up the box. As specified earlier, we will assume that the box will attach to the arm (possibly by means of suction or a claw) as soon as the contact is made between the box and the arm.

3. Once the object is picked up, the rod starts to rotate to the desired position and the arm will converge to the desired height as well.

4. The object is dropped and the robot returns to the default state. This process repeats to keep sorting all the boxes coming on the supply belt.

We have made 3D simulations as well to visualize the working of the robot. The design has been abstracted at various levels to compartmentalize and keep things simple. Various levels of abstraction have been explained below along with the Simulink images to aid in understanding.

- The highest level of abstraction is made at the level of control block and the simulation block. There are certain data lines that are exchanged between both of the blocks. The parameters assumed as inputs, explained above, are provided at this level. The simulation block features the environment and provides the current θ, z, velocity of the arm and the rotational velocity of the rod.

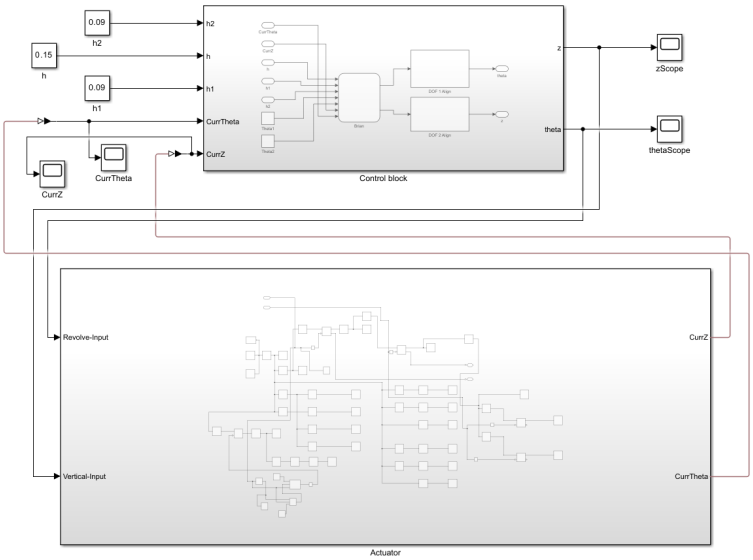

Figure 4.4: Complete Control System

- In the control section, the subsections are the brain of the system, which decides which direction to rotate, when to rotate, when to move the arm... Effectively, this Stateflow chart decides the value of θ_d and z_d to be provided to the control sections throughout the operation.

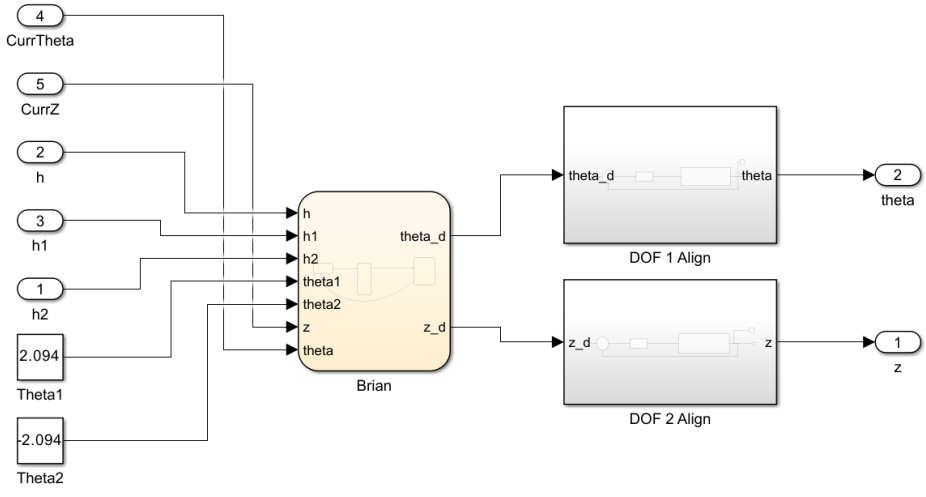

Figure 4.5: Control Block

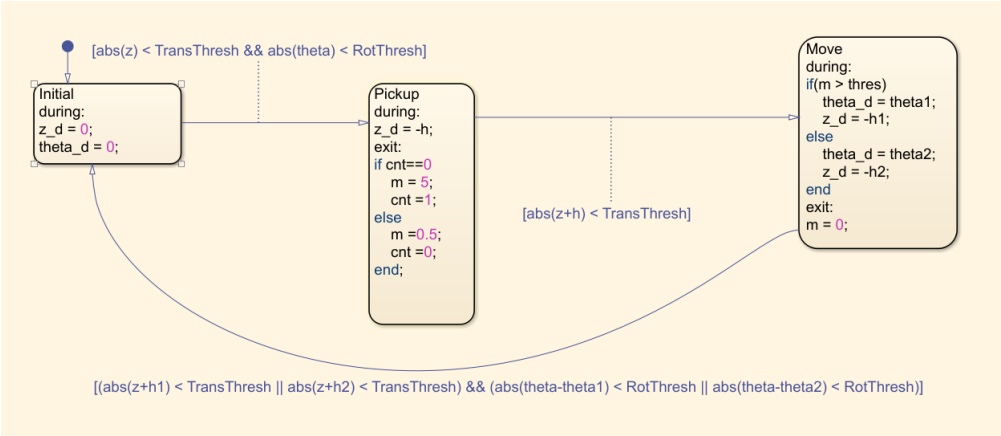

Figure 4.6: The 'Brain' of the system

- The values of θ_d and z_d are fed into the control blocks present in the control block, which perform the actual movement. They have the structure of the block diagram shown earlier implemented in them. The control blocks are where the feedback loop is present and the PID action takes place.

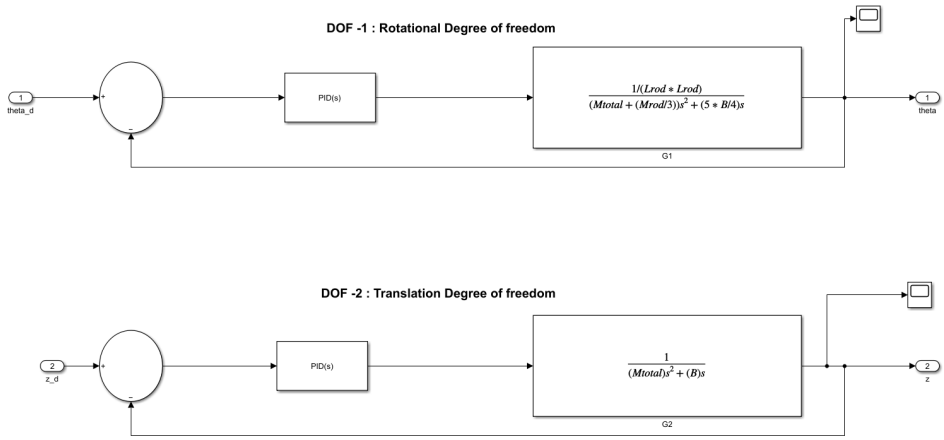

Figure 4.7: Block Diagram implementation for each DOF

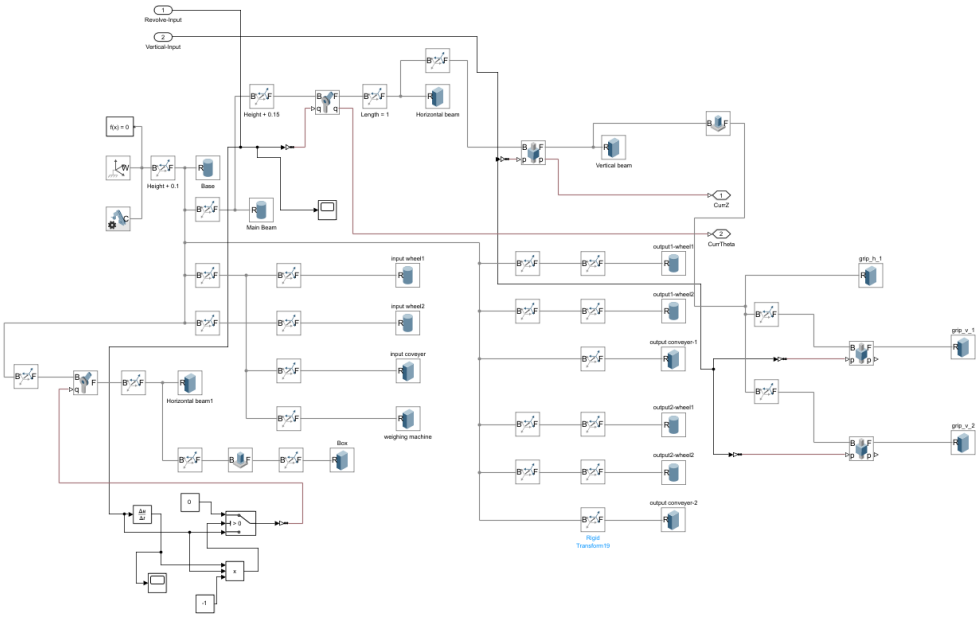

Figure 4.8: Actuator built using Simscape Multibody

The values that are chosen for our simulations are given below.

Parameter	Value
θ_1	120°
θ_2	-120°
h	0.15 m
h_1	0.09 m
h_2	0.09 m

Table 4.1: Values used for simulations

where θ_1 and θ_2 are the angles between the supply belts and the two different conveyor belts. h, h_1, and h_2 are the heights between the rod and the conveyor belts (supply, belt 1 and belt 2 respectively). The sections of Simulink are attached below, along with the simulation results in the following chapters.

Chapter 5

Controllers and their Impact on System Performance

Observe that in 3.1.2 and 3.2.1, the transfer functions are of the *second-order* form. Therefore, we perform the analysis for both systems in a general way by assuming the coefficient of the s^2 term to be α and the coefficient of the s term to be β.
That is,

$$G(s) = \frac{1}{\alpha s^2 + \beta s}$$

And thus the closed loop transfer function $T(s)$ can be expressed as:

$$T(s) = \frac{1}{\alpha s^2 + \beta s + 1} \tag{5.0.1}$$

We also have the general form of a second order system:

$$T_{\text{general}}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.0.2}$$

We thus analyze this T(s) in 5.0.1 with the closed-loop transfer functions obtained with the application of different controllers (especially the denominators, since the numerator in a second-order system is a scaled natural frequency ω_n term).

5.1 Proportional Control

The generic block diagram with the proportional controller, with proportional gain K_p is given below:

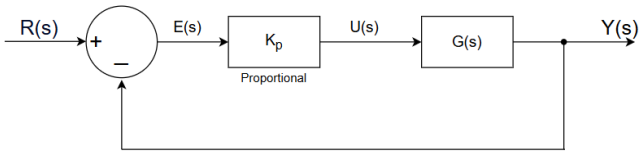

Figure 5.1: Block Diagram with a Proportional Controller

The **Closed loop transfer function** $T_P(s)$ of the system with a proportional controller is thus:

$$\frac{Y(s)}{R(s)} = \frac{k_p G(s)}{1 + k_p G(s)}$$

$$\therefore T_P(s) = \frac{K_p}{\alpha s^2 + \beta s + K_p} \tag{5.1.1}$$

Comparing the denominators in 5.0.1 and 5.1.1
For the proportional controller, the natural frequency ω_{n_P} can be given as:

$$\omega_{n_P} = \sqrt{K_p} \quad > 1$$

And thus the expression for the damping ratio, ζ_P is:

$$\zeta_P = \frac{\beta}{2\omega_{n_p}}$$
$$\zeta_P = \frac{\beta}{2\sqrt{K_p}} < \frac{\beta}{2\omega_n} \qquad \because \omega_n = 1 < \sqrt{K_p}$$

Clearly the natural frequency has changed, and if $K_p > 1$, ω_n has increased, and thus to maintain the same coefficient of the s-term, the damping ratio, ζ has to decrease. This results in **larger and faster oscillations**, with greater overshoots as a consequence.

5.2 Proportional Derivative Control

Say we now include a derivative control in addition, with a derivative gain K_d. The block diagram with the PD controller is given below:

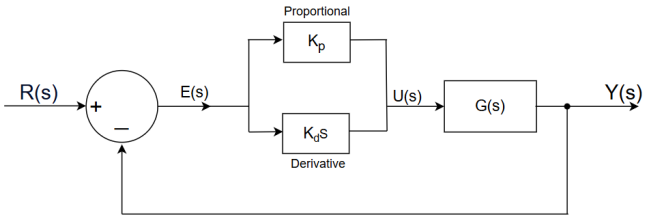

Figure 5.2: Block Diagram with a PD Controller

We thus get,

$$\frac{Y(s)}{R(s)} = \frac{(K_p + K_d s)}{\alpha s^2 + \beta s + (K_p + K_d s)}$$

The closed-loop transfer function of this block, $T_{PD}(s)$ is thus given as:

$$T_{PD}(s) = \frac{(K_p + K_d s)}{\alpha s^2 + (\beta + K_d)s + K_p} \tag{5.2.1}$$

To analyze the addition of a derivative control does to the system response we use 5.1.1 and 5.2.1. Clearly, the damping ratio, ζ_{PD} has increased, while the natural frequency has remained the same. Thus, the addition of the derivative control only increases the damping ratio, as the derivative control is an anticipative controller and reduces the error if it tends to increase in the transient response.

5.2.1 Practical Usage of Derivative Control

In a PID (Proportional-Integral-Derivative) controller, the **filter coefficient** is usually associated with the derivative control term. The general PID control equation is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

Where:

- K_p is the proportional gain.
- K_i is the integral gain.
- K_d is the derivative gain.
- $e(t)$ is the error signal (difference between set-point and the actual value).

The derivative term is susceptible to high-frequency noise because differentiation amplifies noise. To address this, the derivative is often filtered. We have the following transfer function expression for a Low Pass Filter:

$$H(s) = \frac{1}{1 + \frac{s}{\omega_0}}$$

where, ω_0 is the cutoff frequency for the filter.

The filter coefficient (sometimes denoted by N) smoothens the derivative term by introducing a low-pass filter. This modified derivative term is given by:

$$D(s) = \frac{K_d s}{1 + \frac{s}{N}}$$

Here:

- s is the Laplace operator.
- N is the filter coefficient, which controls how much filtering is applied. Increasing the value of N reduces filtering and increases responsiveness to changes, while a lower value smoothens the signal more but makes the system slower.

In summary, the filter coefficient N reduces the sensitivity of the derivative term to noise by adding a low-pass filter. With these insights about the derivative control, we integrate it with the proportional controller, and analyze the thus obtained PD Controller.

5.3 Proportional Integral Control

We now have the controller block to have the integral controller with integral gain as K_i and the proportional controller, with the same proportional gain K_p as before.

The block diagram with the PI controller is given below:

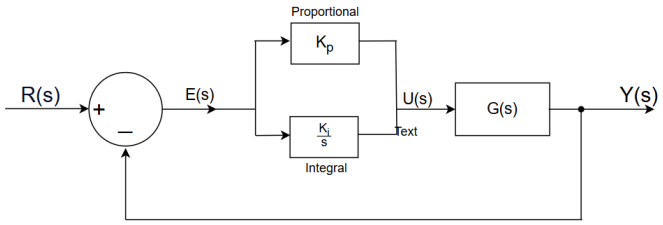

Figure 5.3: Block Diagram with a PI Controller

$$\frac{Y(s)}{R(s)} = \frac{G'(s)}{1 + G'(s)} \qquad \text{where, G'(s) is open loop transfer function}$$

$$\because G'(s) = \frac{(K_i + K_p s)}{s(\alpha s^2 + \beta s)}$$

$$\therefore \frac{Y(s)}{R(s)} = \frac{(K_i + K_p s)}{s(\alpha s^2 + \beta s) + (K_i + K_p s)}$$

The closed-loop transfer function of this block, T_{PI}(s) is thus given as:

$$T_{PI}(s) = \frac{(K_i + K_p s)}{\alpha s^3 + \beta s^2 + K_p s + K_i}$$

Clearly, the equation is of the order 3. Therefore the analysis cannot be done directly, to the second order equation parameterized by ω_n and ζ. But we can say that the effects of the individual components would still be evident, and would play a role in the system response.

5.4 Proportional Integral Derivative Control

We now have the PID controller with the gains for the proportional, derivative and integral control being K_p, K_d and K_i respectively.

The block diagram with the PID controller, is given below:

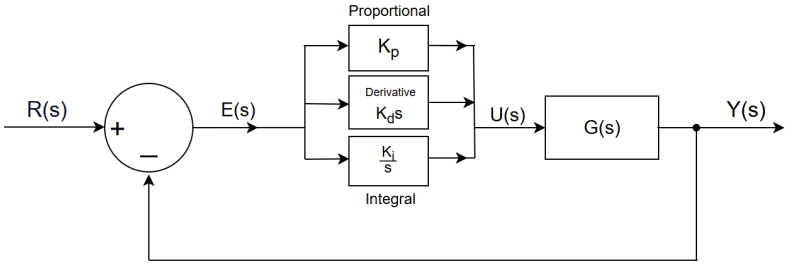

Figure 5.4: Block Diagram with PID

Here,

$$\because G'(s) = \frac{(K_i + K_p s + K_d s^2)}{s(\alpha s^2 + \beta s)}$$

$$\therefore \frac{Y(s)}{R(s)} = \frac{(K_i + K_p s + K_d s^2)}{s(\alpha s^2 + \beta s) + (K_i + K_p s + K_d s^2)}$$

The closed-loop transfer function of this block, T_{PID}(s) is thus given as:

$$T_{PID}(s) = \frac{(K_d s^2 + K_p s + K_i)}{\alpha s^3 + (\beta + K_d)s^2 + K_p s + K_i}$$

Similar to Proportional Integral, here also, the order of function is 3. So the analysis of this is not easy. But by the plots obtained from the simulation, we can say that damping is more in PID in comparison of PI control.

5.5 Steady State analysis

We now analyze the system with the above controllers for their steady state error. Since the **Final Value Theorem** can be applied only if the subject of the limit (sE(s) in this case) has poles with negative real part. we take the assumption that the denominator polynomial with a degree 1,2 or 3 (since we are only dealing with third-order systems in the worst case among the given controllers) has singularity points (poles) with negative real parts, to ensure stability. For the derivation of the steady-state error, *only in this section*, we'll have a slight change in notation, which is listed below.

- G(s) is the transfer function of the plant, with
- C(s) is the transfer function of the controller.

$$G(s) = \frac{1}{\alpha s^2 + \beta s}$$

  As a generic transfer function, which applies to both the systems addressing θ and z.

- T(s) is the feed-forward transfer function, which is the product of the transfer functions of the plant, and the controller. $(T(s) = G(s)C(s))$

Also, we have unity feedback, so H(s) = 1 The block diagram with this notation, is provided below:

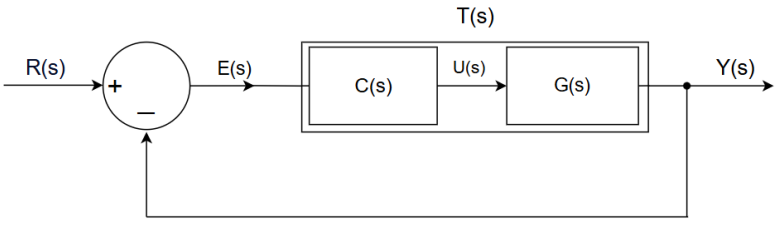

Figure 5.5: Block Diagram

We thus have, the closed loop transfer function,

$$\frac{Y(s)}{R(s)} = \frac{T(s)}{1 + T(s)H(s)}$$

$$\frac{Y(s)}{R(s)} = \frac{T(s)}{1 + T(s)}$$

But, Y(s) = E(s)T(s)

$$E(s) = \frac{R(s)}{1 + T(s)}$$

Now, with the final value theorem, we thus have:

$$\begin{aligned}\lim_{t\to\infty} e(t) &= \lim_{s\to 0} sE(s) \\ &= \lim_{s\to 0} \frac{sR(s)}{1 + T(s)}\end{aligned}$$

Since the reference input is just a scaled (say scaling factor is k) step signal, whose Laplace Transform is $\frac{1}{s}$, we can say sR(s) = k

We thus obtain the expression for the limit, as:

$$\lim_{s\to 0} \frac{k}{1 + T(s)} \tag{5.5.1}$$

We now use 5.5.1 to find the steady-state for different controllers.

5.5.1 Proportional (K_p) Controller

We thus have:

$$T(s) = \frac{K_p}{\alpha s^2 + \beta s}$$

And the limit of sE(s) evaluates to (from 5.5.1):

$$\begin{aligned}&\lim_{s\to 0} \frac{k}{1 + \frac{K_p}{\alpha s^2 + \beta s}} \\ &= \lim_{s\to 0} \frac{k\left(\alpha s^2 + \beta s\right)}{\alpha s^2 + \beta s + K_p} = 0\end{aligned}$$

The limit evaluates to 0, as the limiting expression takes a determinate form as s tends to 0. $\therefore \boxed{e_{SS} = 0}$ for the case of a proportional controller.

This might seem contradictory to our understanding that a proportional control has steady state error. However, the error is indeed zero in our system as there is no force acting which is proportional to the displacement/angular displacement. In other words, there is no constant term in the denominator which is contributing to the error in the system!

5.5.2 Proportional-Derivative (PD) (K_p, K_d) Controller

T(s) evaluates to:

$$T(s) = \frac{K_p + K_d s}{\alpha s^2 + \beta s}$$

Thus, from 5.5.1 the limit of sE(s) evaluates to:

$$\begin{aligned}&\lim_{s\to 0} \frac{k}{1 + \frac{K_p + K_d s}{\alpha s^2 + \beta s}} \\ &= \lim_{s\to 0} \frac{k\left(\alpha s^2 + \beta s\right)}{\alpha s^2 + (\beta + K_d)s + K_p} = 0\end{aligned}$$

The limit thus evaluates to 0, as the expression above takes a determinate form as s tends to 0. $\therefore \boxed{e_{SS} = 0}$ for the case of a PD controller as well!

5.5.3 Proportional-Integral (PI) (K_p, K_i) Controller

T(s) evaluates to:

$$T(s) = \frac{K_p + \frac{K_i}{s}}{\alpha s^2 + \beta s} = \frac{sK_p + K_i}{\alpha s^3 + \beta s^2}$$

Thus, from 5.5.1 the limit of sE(s) evaluates to:

$$\lim_{s\to 0} \frac{k}{1+\frac{sK_p+K_i}{\alpha s^3+\beta s^2}}$$
$$= \lim_{s\to 0} \frac{k\left(\alpha s^3+\beta s^2\right)}{\alpha s^3+\beta s^2+K_p s+K_i} = 0$$

The limit thus evaluates to 0, as the expression above takes a determinate form as s tends to 0. $\therefore \boxed{e_{SS}=0}$ for the case of a PI controller as well!

5.5.4 PID (K_p, K_i, K_d) controller

T(s) evaluates to:

$$T(s) = \frac{K_p+\frac{K_i}{s}+K_d s}{\alpha s^2+\beta s} = \frac{K_d s^2+sK_p+K_i}{\alpha s^3+\beta s^2}$$

Thus, from 5.5.1 the limit of sE(s) evaluates to:

$$\lim_{s\to 0} \frac{k}{1+\frac{K_d s^2+sK_p+K_i}{\alpha s^3+\beta s^2}}$$
$$= \lim_{s\to 0} \frac{k\left(\alpha s^3+\beta s^2\right)}{\alpha s^3+(\beta+K_d)s^2+K_p s+K_i} = 0$$

Hence, the steady state error of the PID controller zeroes out, and $\therefore \boxed{e_{SS}=0}$ for the PID Controller.

Chapter 6

Simulations with different control actions

In this section, we will display the trends of the system on applying four different control actions: P, PI, PD and PID.

1. **Proportional control**

   On applying only P control, we observe the below time response:

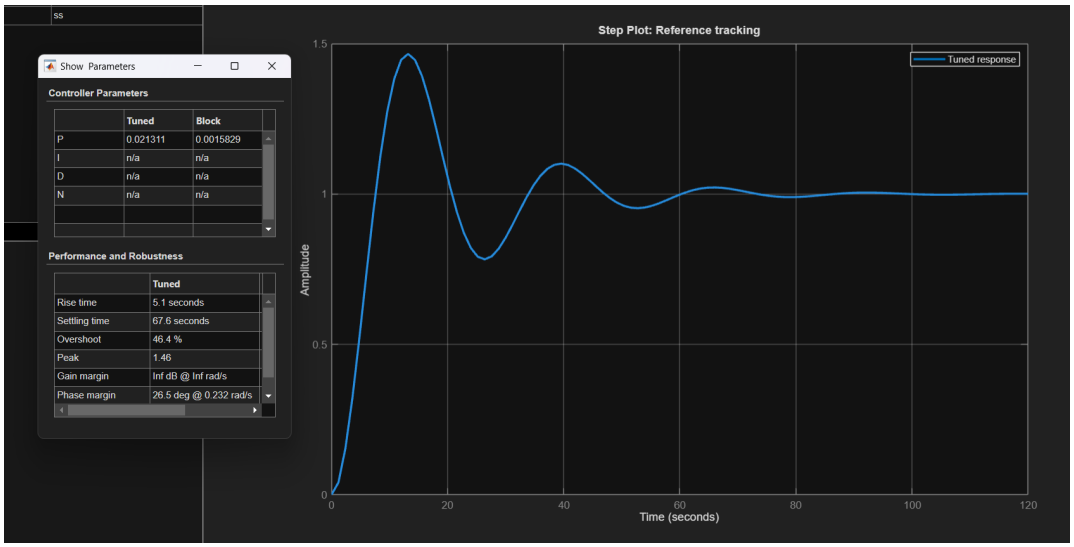

Figure 6.1: Time Response of the system under Proportional Control

Explanation of the plot: A high value of K_p means that the control signal is strong if the error is significant, which means that the control can be "overdone" causing some **overshoot** across the desired position. This is the cause for the oscillations observed in the plot about the steady state position.

2. **PI Control**

The time response under both Proportional and Integral control is given below:

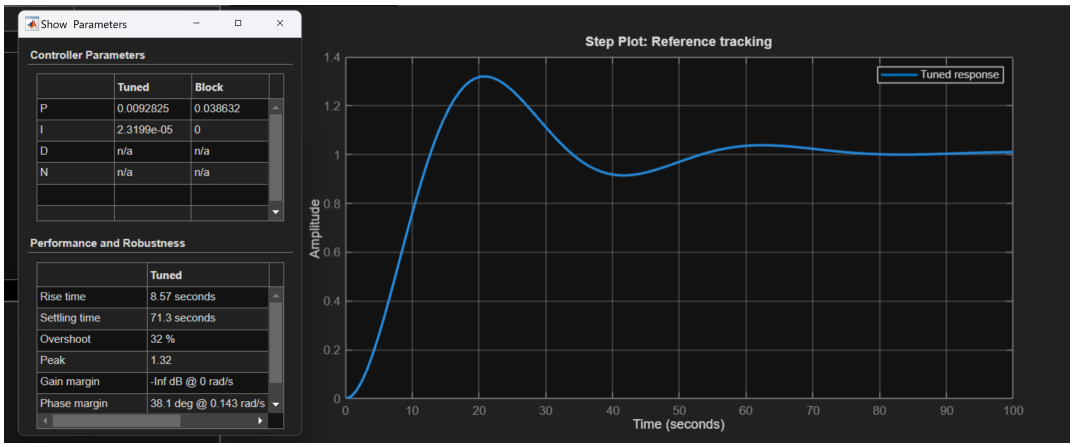

Figure 6.2: Time Response of the system under Proportional-Integral Control

In general, the benefit of adding an integral control is that the transfer function is modified such that steady state error is driven to 0, however, the nature of our transfer function in this control system is such that zero steady state error is attained even with just P control. Still, the nature of an integrator is reflected in the plot- **oscillations** are introduced by the fact that the error from time zero onwards has been summed up, which means that the integral controller may exert a control even if the system is currently in the steady state position, which causes small oscillations about the final position until the error curve falls to zero. In our case, the area under the curve also falls to 0 and the integral and proportional controls are both 0 at steady state.

3. **PD control**

After adding on derivative control to a solely proportional controller, we obtain the below response:

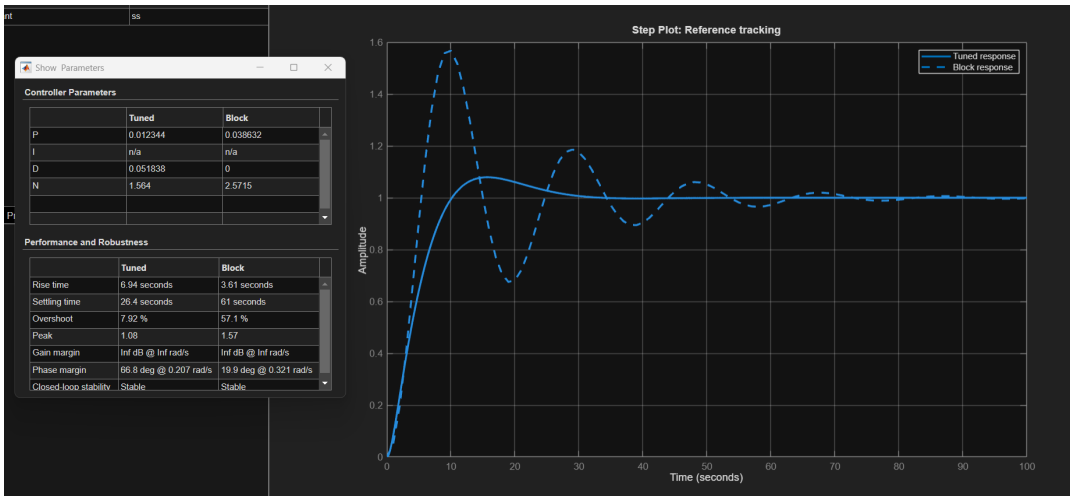

Figure 6.3: Time Response of the system under Proportional-Derivative Control

In this plot, the dotted lines represent the previously obtained P response. In comparison, we observe a significant amount of **damping** under the influence of the derivative controller. Higher the rate of change of the error signal, higher the damping effect of the derivative controller, hence the overshoot is counteracted by the Proportional-Derivative controller.

4. **PID Control**

A combination of Proportional, Integral and Derivative controllers produces the following response:

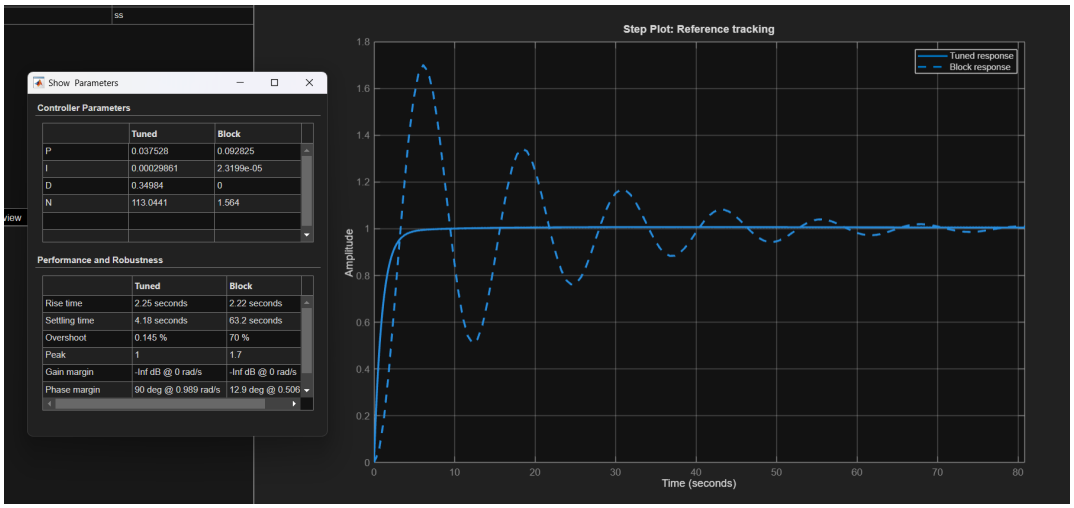

Figure 6.4: Time Response of the system under Proportional-Integral-Derivative Control

Under PID control, we observe "optimal" driving of the system towards the steady state - we are able to achieve quick settling time in addition to little to no overshoot - PID control shares all the positive characteristics of the previous sections, hence the practical benefits of applying all controllers simultaneously has been verified.

Chapter 7

Conclusion and Improvements

To visualize the controller action, we have attached a video of the simulation. Certain parameters were tweaked for the 3D simulation. This is to just capture the essence of the robot and its control law. We make a few assumptions, without any loss of generality, like the input boxes fall in alternate weight categories, the heights are the same, the belts are symmetric, etc. With changes in the program files, the robot can work in varying conditions as well.

Through this project, we have learnt how to model mechanical systems, simulate them, work with their dynamics, different control actions and applying control theory concepts to create real-world applications. Like any other human endeavour, our model is not impeccable and has huge potential for further development. One can add more DOFs to expand on the work envelope and cater to a wider range of problems. The control law can be used along with other advanced control algorithms.

We sincerely thank Professor Spandan and the Teaching Assistants for giving us this opportunity to make a simplified version of the SCARA robot. The learning curve was excellent and very enjoyable.