TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI	SME.EDU - Mẫu 6.a
VIỆN CƠ KHÍ	Học kỳ: 1
Bộ môn Cơ điện tử	Năm học: 2023-2024

ĐỒ ÁN MÔN HỌC: THIẾT KẾ HỆ THỐNG CƠ ĐIỆN TỬ **Mã HP: ME5512**

Thời gian thực hiện: 15 tuần; Mã đề: VCK03-04

Ngày …/…/20…	Ngày …/…/20…	Ngày …/…/20…
ĐƠN VỊ CHUYÊN MÔN	**NGƯỜI RA ĐỀ**	**CB Hướng dẫn**
(ký, ghi rõ họ tên)	*(ký, ghi rõ họ tên)*	*(ký, ghi rõ họ tên)*

Ngày giao nhiệm vụ: …/…/20…; Ngày hoàn thành: …/…/20…

Họ và tên sv: Doãn Nhật Minh MSSV: 20195793 Mã lớp: ………… Chữ ký sv: ……

I. Nhiệm vụ thiết kế: Thiết kế hệ thống điều khiển robot SCARA 3 bậc tự do

II. Số liệu cho trước:

1. Tải trọng 20 kg.
2. Tầm với 0.7 m.
3. Độ chính xác lặp: (x, y) = 0.02 mm, (z) = 0.01 mm.
4. Vận tốc cực đại khâu tác động cuối 9.57 m/s
5. Gia tốc cực đại khâu tác động cuối 33 m/s2

III. Nội dung thực hiện:

1. Phân tích nguyên lý và thông số kỹ thuật hệ thống điều khiển

- Tổng quan về hệ thống điều khiển
- Nguyên lý hoạt động hệ thống điều khiển
- Xác định các thành phần của hệ thống điều khiển

2. Thiết kế hệ thống điều khiển

- Mô hình hóa và xác định hàm truyền
- Đánh giá tính ổn định của hệ thống
- Mô phỏng và phân tích, đánh giá các chỉ tiêu kỹ thuật của hệ thống
- Lựa chọn các thiết bị cho hệ thống điều khiển: cảm biến, thiết bị điều khiển, cơ cấu chấp hành
- Thiết kế sơ đồ mạch điện và mạch điều khiển (1 bản A0)

3. Lập trình điều khiển

- Lập trình điều khiển robot (1 chương trình điều khiển)
- Lập trình mô phỏng chuyển động (1 chương trình mô phỏng trên Simmechanics, …)

ACKNOWLEDGMENTS

Group: Automatic equipment, Faculty of Mechatronics, School of Mechanical Engineering, Hanoi University of Science and Technology.

Under the enthusiastic guidance of **Assoc. Prof. Pham Duc An**, I had the chance to apply some of the the knowledge that I have learned in the university training program to complete the project, so that I could perfect and synthesize knowledge before graduating from university.

However, with limited professional knowledge, the report cannot avoid shortcomings and cannot be completed perfectly as expected. I look forward to your help and suggestions so that I can improve.

For completing this project, I would like to thank all lecturers in Hanoi University of Science and Technology, especially lecturers in the School of Mechanical Engineering. I would like to express my deep gratitude to my instructor **Assoc. Prof. Pham Duc An**, who wholeheartedly guided and created favorable conditions for me to complete this project.

Thank you very much!

Hanoi, January 13, 2024

Doãn Nhật Minh

Table of Contents

LIST OF SYMBOLS AND ACRONYMS

Symbol	Meaning
s_1	: $\sin\theta_1$
s_2	: $\sin\theta_2$
c_1	: $\cos\theta_1$
c_2	: $\cos\theta_2$
s_{12}	: $\sin(\theta_1+\theta_2)$
c_{12}	: $\cos(\theta_1+\theta_2)$
θ_{12}	: $\theta_1+\theta_2$
${}^{i-1}\mathbf{C}_i$	: Coordinate transformation matrix from reference frame i to i-1
${}^{0}\underline{C}_i$	: Coordinate transformation matrix from reference frame i to base frame 0
${}^{0}\underline{R}_i$	: The direction cosine matrix of the link i with respect to the base frame 0
${}^{i-1}\underline{r}_i$	: Homogeneous vector of the centroid of link i with respect to reference frame i-1
$\underline{v}_{Ci}$	: Linear velocity at the centroid of the i-th link
$\underline{a}_{Ci}$	: Linear acceleration at the centroid of the i-th link
${}^{0}\underline{\varepsilon}_1$	: Angular acceleration at the centroid of the i-th link
${}^{i-1}\underline{\omega}_i$	: Angular velocity of the i-th link with respect to link i-1
${}^{0}\underline{\omega}_i$	: Angular velocity of the i-th link with respect to base frame 0
K	: End effector

LIST OF PICTURES AND DIAGRAMS

ABSTRACT

I. PROBLEM STATEMENT

The Scara robot was first introduced in 1979, this is a type of mechanical hand with a special structure used in many assembly jobs. It performs jobs that require high precision. Scara robots have been widely applied and developed around the world, they are built and developed on a large scale. However, in Vietnam, research and development is still limited both at research institutes and universities. Due to practical needs and passion for research under the dedicated direction of **Assoc. Prof. Pham Duc An,** Faculty of Mechatronics - School of Mechanical Engineering - Hanoi University of Science and Technology, we chose the topic "**Design control system for 3-degree of freedom SCARA ROBOT**".

II. PURPOSE OF THE TOPIC

The purpose of the topic is first to learn, get acquainted with scientific research and apply the knowledge learned in practice. In the process of practical research, the topic has helped us improve our knowledge, approach new problems of reality as well as serve as a solid foundation for mechatronics engineers after graduation.

III. SCOPE

- Plan movement trajectory in the workspace of the Scara SR8 plus
- Modeling of the Scara SR8 control system
- Stability assessment simulation
- Control system design for Scara SR8 robot

IV. BRIEF CONTENTS

The content of the project is divided into 4 chapters as follows:

Chapter 1: *Overview of the control system*

This chapter presents an overview of the Robot SCARA SR8 PLUS, principle of operation and specifications.

Chapter 2: *Kinematics, Dynamics, and Design of Motion Trajectories*

This chapter discusses kinematics, dynamics, and design motion planning strategies.

Chapter 3: *Control System Modeling*

This chapter presents the problem of modeling each link of the robot individually, defining the transfer function and building a control system model for the whole robot.

Chapter 4: *Simulation and Stability Assessment*

This chapter covers using MATLAB-Simulink to simulate and evaluate the stability of control systems.

Chapter 5: *Control System Design*

This chapter covers how to design PID controllers, select elements for control systems, electrical circuits, and how to generate codes for microcontrollers from MATLAB/Embedded Coder.

Chapter 1: OVERVIEW OF THE CONTROL SYSTEM

1.1 SPECIFICATIONS OF ROBOT TURBO SCARA SR4/6/8

a) Introduction

The Scara robot is one of the most commonly used industrial robots today. The movement of this robot is very simple but suitable for effective lines and applications in the task of picking and placing products. Scara robot (Selectively Compliant Articulated Robot Arm) means selectively assembled robot.

The kinetic structure of this type of hand belongs to the biomimetic system, has axes of rotation, the joints are all vertical. It has a construction of two joints in the arm, one joint in the wrist and one translational joint.

Some types of scara robots from manufacturers

a) Robot Scara SR8 plus [internet] | b) Robot Scara from EPSON [internet] | c) Robot Scara from MITSUBISHI [internet]

Figure 1.1 Some variations of Scara robots today

Within the scope of this project, the author used the Turbo Scara SR8 Plus robot to research, calculate and design a control system for a 3-dof Scara robot model with 2 degrees of rotational freedom with 1 degree of translational freedom.

Turbo Scara SR8 Plus robot is an industrial robot of Rexroth of Bosch Group, equipped with preeminent features including high flexibility in operation, wide operating range, wide applicability in teaching processes and production lines, large working space, high precision, connecting peripheral devices such as computers, PLCs ..., the mechanisms are scientifically designed in accordance with international standards and have high aesthetics.

b) Turbo Scara SR8 plus configuration

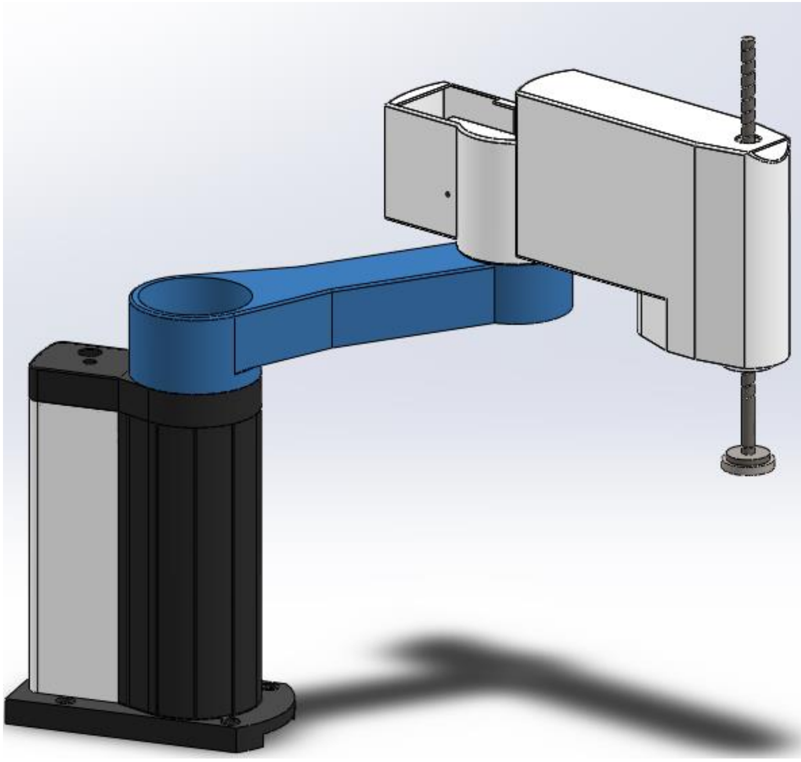

Figure 1.2 Solidworks model of Turbo Scara SR8

Turbo Scara SR8 robot consists of 2 rotational joint and 1 translational joint:

- Joint 1 rotates about the Z_0 axis at an angle θ_1
- Joint 2 rotates about the Z_1 axis at an angle θ_2
- Joint 2 translates along Z_2 axis at a displacement d_3

c) Technical details of Turbo Scara SR8 plus

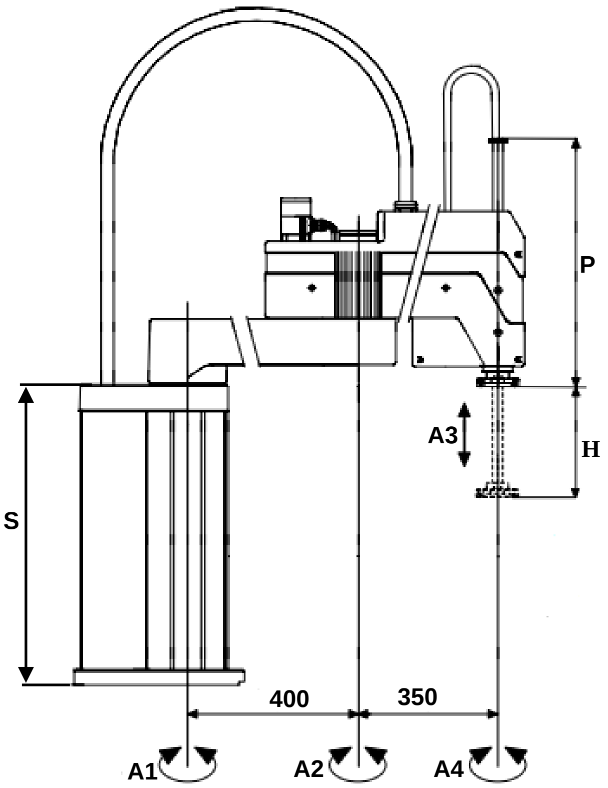

Figure 1.3 Configuration of Tubo Scara SR8

Table 1.1 Technical specification of SCARA

<table>
<tr><th>Option</th><th>Parameter</th><th>Designation</th><th colspan="3">Value</th></tr>
<tr><td rowspan="6">Standard</td><td>Base height</td><td>S</td><td colspan="3">450</td></tr>
<tr><td>Maximum elongation of joint 3</td><td>H</td><td>280</td><td>280</td><td>280</td></tr>
<tr><td>Length of lead screw</td><td>P</td><td>450</td><td>450</td><td>450</td></tr>
<tr><td>Thickness</td><td>F</td><td>60</td><td>60</td><td>60</td></tr>
<tr><td>Length of link 2</td><td>a_1</td><td>400</td><td>400</td><td>400</td></tr>
<tr><td>Length of link 3</td><td>a_2</td><td>350</td><td>350</td><td>350</td></tr>
</table>

Number of DOFs	3	
Number of joints	3	
Standard load	2	kg
Maximum load	20	kg
Maximum rotational angle of link 1	+/- 150	∘
Maximum rotational angle of link 2	+/- 160	∘
Maximum displacement of link 3	280	mm
Repeatability (x, y)	+/- 0.02	mm
Repeatability (z)	+/- 0.01	mm
Speed of link 1	450	∘ / s
Speed of link 2	667	∘ / s
Speed of link 3	2780	mm / s
Overall speed	9570	mm / s

d) Working principle

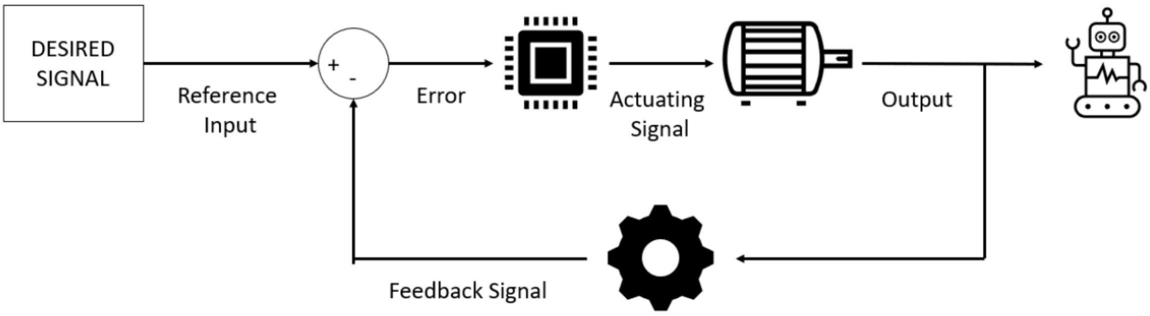

Figure 1.4 Genera block diagram of a control system

Depending on the task requirements and the pre-programmed program, the controller will output control signals to the actuators that perform movements and operations (rotating motor as a rotating mechanism, open gripper, or welding torch to start welding ...)

The sensor system will then return the signal to the controller to correct and handle the deviations (if any). The goal of using a closed loop control system is to suppress disturbance signals and control the operation of the robot, ensuring that the robot operates exactly as required.

e) Basic elements of a control system

- Servo motor

Figure 1.5 Servo motor from Yaskawa.

Servo motors serve as the main actuations of the robot. They turn electricity into mechanical energy for robots to perform movements.

- Limit switch

Figure 1.6 Limit switch

Limit limit the stroke of the moving part, receive a signal when there is a physical impact on the mechanical part of the sensor, convert it into an electrical signal that controls the actuator.

- Programmable logic controller (PLC)

Figure 1.7 PLC Siemens Simatic S7 - 1200

PLCs are programmable logic controllers. Users can completely change the control algorithm by programming PLC. In the robot system, PLC acts as a "brain" to help calculate and control all activities of the robot.

- Micro Controller Unit

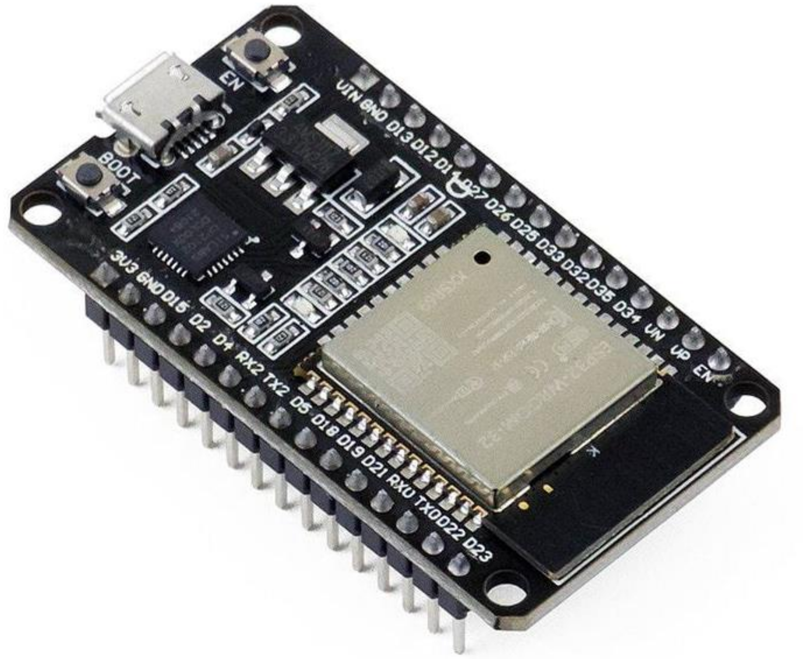

Figure 1.8 MCU ESP32 from Espressiff

In addition to PLCs, robotic systems can also use microcontrollers in the role of "brains". Compared to PLCs, microcontrollers require simpler, smaller connections, but are less durable and less resistant to interference when working in industrial environments.

- Electronic circuits

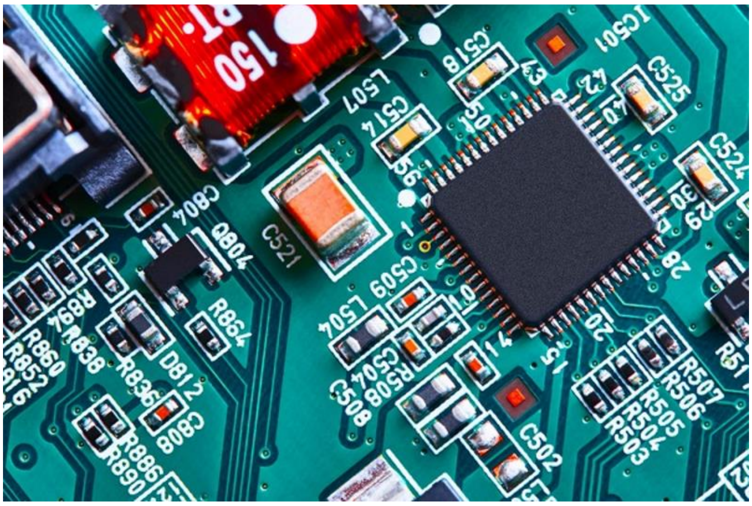

Figure 1.9 Electronic circuits

The electronic circuitry in the robotic system acts as the color circuit in the human body. It plays the role of connecting the parts of the robot into a unified whole. Electronic circuits may include:

– Microcontroller circuit: Acts as a robot's brain, controlling the entire operation of the robot according to a pre-programmed program.

– Sensor circuit: Acts as the robot's senses, helping the robot recognize its surroundings to send signals to the microcontroller.
– Power circuit: Has the function of controlling the power and speed of the motor, used to control the operation of the robot's mechanisms.
- Human-Machine Interface Device (HMI)

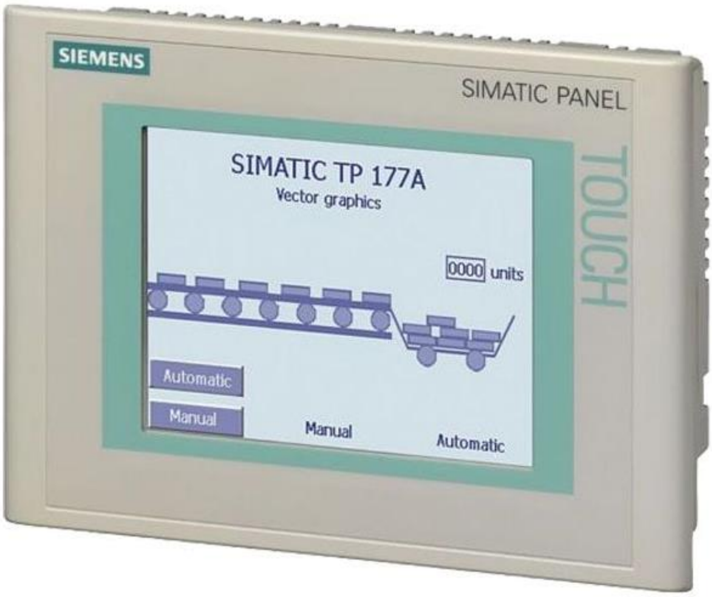

Figure 1.10 HMI Siemens TP177A

Typically, robots will be equipped with a human-robot communication device (HMI). The type of HMI varies from the operation method: button, touch or a combination of both and many sizes depending on the needs and actual characteristics. Thanks to HMI, the process of controlling & monitoring robots becomes more intuitive and convenient.

Chapter 2: KINEMATICS, DYNAMICS, AND TRAJECTORY DESIGN

2.1 SET UP THE ROBOT'S KINEMATIC EQUATION

a) Denavit-Hartenberg method

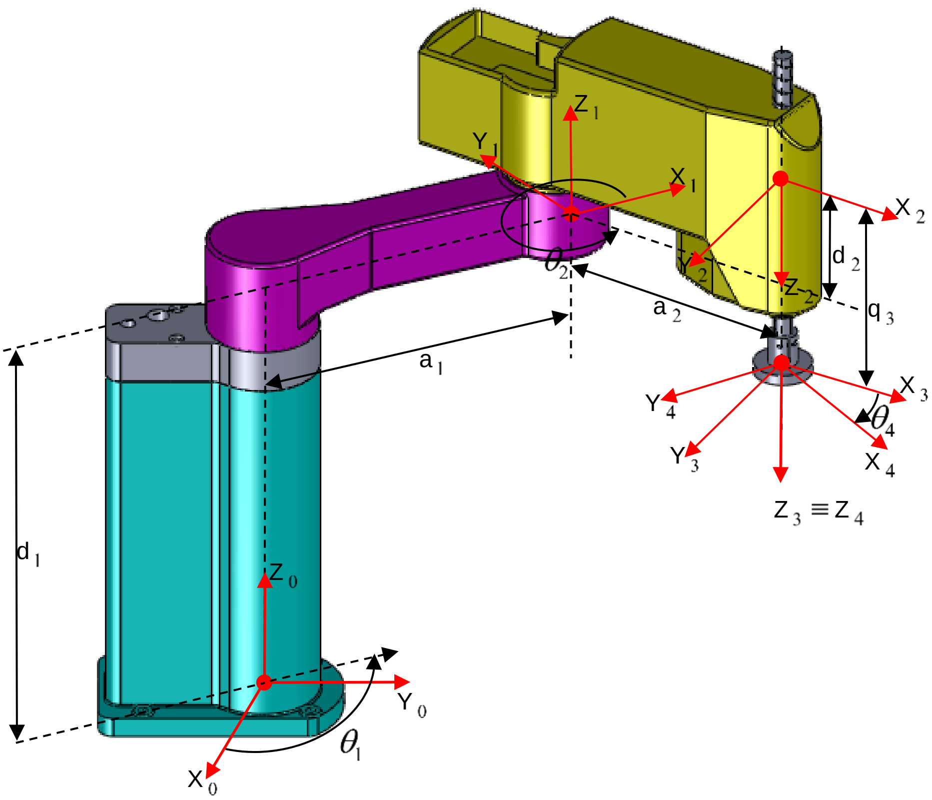

Figure 2.1 Denavit-Hartenbeg's coordinates

With the placement of the axis system as shown in Figure 2.1 and the rules for determining the kinetic parameters according to D-H, we can make D-H table.

Table 2.1 Denavit-Hartenberg table:

Joint	d_i	θ_i	a_i	α_i
1	d_1	θ_1	a_1	0
2	d_2	θ_2	a_2	π
3	q_3	0	0	0

where: q_3, θ_1, θ_2 are joint variables.

b) Homogeneous transformations matricies

General D-H matrix that transforms frame R_{i-1} to R_i, is ${}^{i-1}A_i$ having this form:

$$
{}^{i-1}\underline{C}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_1\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

From D-H table above, we have:

$$
{}^{0}\underline{C}_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{1}\underline{C}_2 = \begin{bmatrix} \cos\theta_2 & \sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & -\cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & -1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}\underline{C}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix};
$$

Homogeneous transformations:

$$
{}^{0}\underline{C}_2 = {}^{0}\underline{C}_1\,{}^{1}\underline{C}_2
$$

$$
= \begin{bmatrix} \cos(\theta_1+\theta_2) & \sin(\theta_1+\theta_2) & 0 & a_1\cos(\theta_1)+a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & -\cos(\theta_1+\theta_2) & 0 & a_1\sin(\theta_1)+a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & -1 & d_1+d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}\underline{C}_3 = {}^{0}\underline{C}_2\,{}^{2}\underline{C}_3
$$

$$
= \begin{bmatrix} \cos(\theta_1+\theta_2) & \sin(\theta_1+\theta_2) & 0 & a_1\cos(\theta_1)+a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & -\cos(\theta_1+\theta_2) & 0 & a_1\sin(\theta_1)+a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & -1 & d_1+d_2-d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

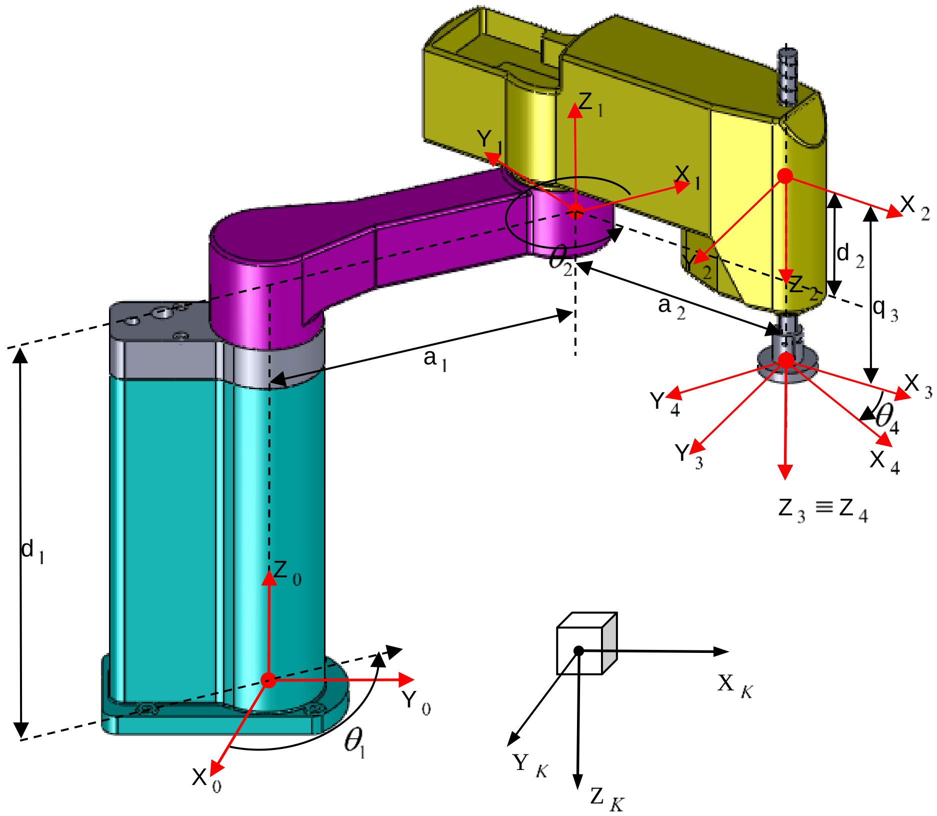

Figure 2.2 End effector location

c) Kinematics equations

Let K, φ are location and orientation of the end effector. We have the following end effector matrix:

$$\underline{T}_K = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & x_K \\ \sin\varphi & \cos\varphi & 0 & y_K \\ 0 & 0 & -1 & z_K \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.1}$$

Let $T_K = {}^0C_3$, we have:

$$\begin{bmatrix} cos\,\phi & -\,sin\,\phi & 0 & x_K \\ sin\,\phi & cos\,\phi & 0 & y_K \\ 0 & 0 & -1 & z_K \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} cos(\theta_1+\theta_2) & sin(\theta_1+\theta_2) & 0 & a_2\,cos(\theta_1+\theta_2)+a_1\,cos\,\theta_1 \\ sin(\theta_1+\theta_2) & -\,cos(\theta_1+\theta_2) & 0 & a_2\,sin(\theta_1+\theta_2)+a_1\,sin\,\theta_1 \\ 0 & 0 & -1 & d_1+d_2-q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

Matching coefficients, we have the forward kinematics equations:

$$\begin{cases} cos\,\phi = & cos(\theta_1+\theta_2) \\ x_K = & a_2\,cos(\theta_1+\theta_2)+a_1\,cos\,\theta_1 \\ y_K = & a_2\,sin(\theta_1+\theta_2)+a_1\,sin\,\theta_1 \\ z_K = & d_1+d_2-q_3 \end{cases} \quad (1.3)$$

- In forwar kinematics we need to find $x_K,\ y_K, z_K$ và φ

$$\begin{cases} x_K = & a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1 \\ y_K = & a_2\sin(\theta_1+\theta_2)+a_1\sin\theta_1 \\ z_K = & d_1+d_2-q_3 \\ \varphi = & \theta_1+\theta_2 \end{cases} \quad (1.4)$$

- In inverse kinematics we need to find θ_1, θ_2, q_3 given $x_K,\ y_K, z_K$ và φ

From (1.4) :

$$\begin{cases} x_K = a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1 \\ y_K = a_2\sin(\theta_1+\theta_2)+a_1\sin\theta_1 \end{cases} \quad (1.5)$$

$$\Rightarrow\ x_K{}^2+y_K{}^2 = a_1{}^2+a_2{}^2+2a_1a_2\left[(\cos(\theta_1+\theta_2)\cos\theta_1+\sin(\theta_1+\theta_2)\sin\theta_1\right]$$

$$x_K{}^2+y_K{}^2 = a_1{}^2+a_2{}^2+2a_1a_2\cos\theta_2$$

$$\Rightarrow \cos\theta_2 = \frac{x_K{}^2+y_K{}^2-a_1{}^2-a_2{}^2}{2a_1a_2}$$

$$\sin\theta_2 = \pm\sqrt{1-\cos^2\theta_2}$$

So, $\theta 2 = a\tan 2(\sin\theta_2;\cos\theta_2)$

Rewrite (1.4) :

$$\begin{cases} x_K = a_2(c_1c_2-s_1s_2)+a_1c_1 \\ y_K = a_2(s_1c_2+s_2c_1)+a_1s_1 \end{cases}$$

(1.6)

where $\begin{cases} s_1 = \sin\theta_1, s_2 = \sin\theta_2 \\ c_1 = \cos\theta_1, c2 = \cos\theta_2 \end{cases}; \begin{cases} (a_2c_2+a_1)c_1 - a_2s_2s_1 = x_K \\ a_2s_2c_1 + (a_2c_2+a_1)s_1 = y_K \end{cases}$

Solve (1.6:

$$\Delta = \begin{vmatrix} a_2c_2+a_1 & -a_2s_2 \\ a_2s_2 & a_2c_2+a_1 \end{vmatrix} = (a_2c_2+a_1)^2+(a_2s_2)^2$$

$$\Delta_1 = \begin{vmatrix} x_K & -a_2 s_2 \\ y_K & a_2 c_2 + a_1 \end{vmatrix} = x_K (a_2 c_2 + a_1) + y_K a_2 s_2$$

$$\Rightarrow \; c_1 = \frac{x_K (a_2 c_2 + a_1) + y_K a_2 s_2}{(a_2 c_2 + a_1)^2 + {a_2}^2 {s_2}^2}$$

$$\Delta_2 = \begin{vmatrix} a_2 c_2 + a_1 & x_K \\ a_2 s_2 & y_K \end{vmatrix} = (a_2 c_2 + a_1) y_K - a_2 s_2 x_K$$

$$\Rightarrow \quad s_1 = \frac{(a_2 c_2 + a_1) y_K - a_2 s_2 x_K}{(a_2 c_2 + a_1)^2 + {a_2}^2 {s_2}^2}$$

So, $\theta_1 = a\tan 2(s_1; c_1)$

And: $q_3 = d_1 + d_2 - z_K$

The inverse kinematics equations are:

$$\begin{cases} \theta_1 = & a\,tan\,2\,(sin\,\theta_1\,;cos\,\theta_1) \\ \theta_2 = & a\,tan\,2\,(sin\,\theta_2\,;cos\,\theta_2) \\ q_3 = & d_1 + d_2 - z_K \end{cases} \tag{1.7}$$

where $\cos\theta_2 = \dfrac{{x_K}^2 + {y_K}^2 - {a_1}^2 - {a_2}^2}{2a_1 a_2}$, $\sin\theta_2 = \pm\sqrt{1-\cos^2\theta_2}$

2.2 WORKSPACE OF THE SCARA

a) Theoretical basis

The workspace of the robot is the area of the robot's operating space in which the robot still operates normally with all degrees of freedom. There are many different workspace calculation methods such as geometric methods, calculus methods, numerical methods. However, in the project, I only use geometric methods.

b) Geometric method

From forward kinematics, we have :

$$\begin{cases} x_K = & a_2\, cos(\theta_1 + \theta_2) + a_1\, cos\,\theta_1 \\ y_K = & a_2\, sin(\theta_1 + \theta_2) + a_1\, sin\,\theta_1 \\ z_K = & d_1 + d_2 - q_3 \end{cases} \quad (1.8)$$

In Oxy plane, we find x_K, y_K such that :

$$\begin{cases} x_K = & a_2 \cos(\theta_1 + \theta_2) + a_1 \cos\theta_1 \\ y_K = & a_2 \sin(\theta_1 + \theta_2) + a_1 \sin\theta_1 \end{cases}$$

where $\theta_1 = \pm 150°$; $\theta_2 = \pm 160°$. We build the coordinate domain x, y then extends the coordinate domain x, y along the z axis we get the workspace of the robot. There are many methods to simulate the workspace, in this project the author simulates the workspace using Matlab software.

Step 1: Initiate constants, $d_1 = 450mm;\ d_2 = 100mm;\ a_1 = 400mm;\ a_2 = 350mm$

Step 2: Create vectors containing the distribution of points in the workspace based on the limits specified in the specifications section, $\theta_1 = \pm 150^o;\ \theta_2 = \pm 160^o; q_3 = 280mm$

Step 3: Use matrix operations to represent a system of end effector coordinate equations (1.8) in MATLAB.

Step 4: Use the plot3 function to draw the workspace.

The figure below is the simulated workspace.

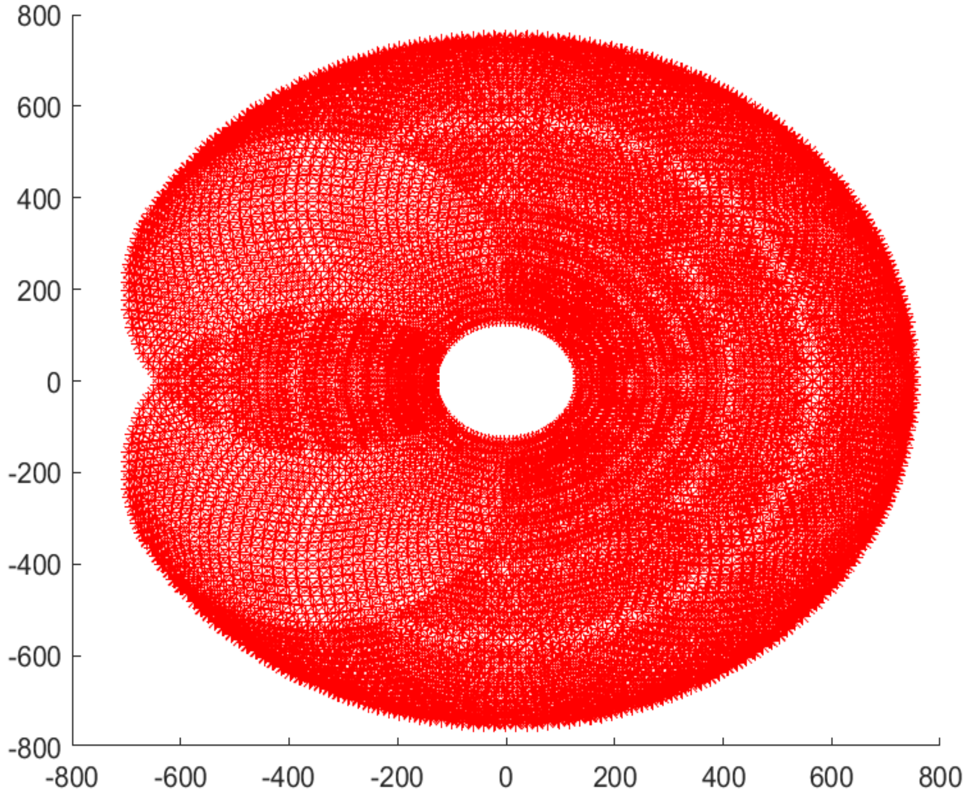

Figure 2.3 Top view of the workspace

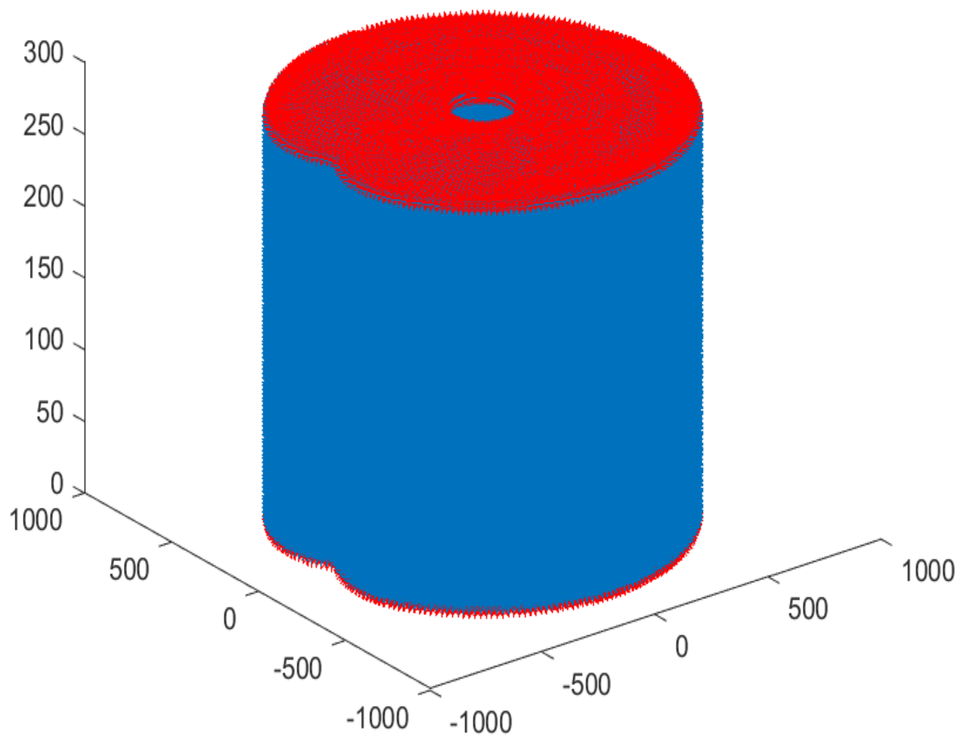

Figure 2.4 Isometric view of the workspace

2.3 SET UP FORMULAS FOR CALCULATING THE CENTROID SPEED OF THE LINKS

a) Linear velocity

- Centroid velocity of link 1.

We have:

$^{0}\underline{r}_{C1} = {}^{0}\underline{C}_{1} . {}^{1}\underline{r}_{C1}$

$$= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -(a_1 - l_1) \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} l_1\cos\theta_1 \\ l_1\sin\theta_1 \\ d_1 \\ 1 \end{bmatrix}$$

Differentiate with respect to time:

$$^{0}\underline{\dot{r}}_{C1} = \begin{bmatrix} -l_1\dot{\theta}\sin\theta_1 \\ l_1\dot{\theta}\cos\theta_1 \\ 0 \\ 0 \end{bmatrix}$$

So.

$$^{0}\underline{v}_{C1} = \begin{bmatrix} -l_1\dot{\theta}\sin\theta_1 \\ l_1\dot{\theta}\cos\theta_1 \\ 0 \end{bmatrix}$$

- Centroid velocity of link 2.

We have:

$^{0}\underline{r}_{C2} = {}^{0}\underline{C}_{2} . {}^{2}\underline{r}_{C2}$

$$= \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & a_1\cos(\theta_1)+a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & a_1\sin(\theta_1)+a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & -1 & d_1+d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -(a_2 - l_2) \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} a_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) \\ a_1\sin\theta_1 + l_2\sin(\theta_1+\theta_2) \\ d_1+d_2 \\ 1 \end{bmatrix}$$

Differentiate with respect to time:

$$^{0}\underline{\dot{r}}_{C2} = \begin{bmatrix} -a_1\dot{\theta}_1\sin\theta_1 - l_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \\ a_1\dot{\theta}_1\cos\theta_1 + l_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \\ 0 \\ 0 \end{bmatrix}$$

So.

$$ {}^{0}\underline{v}_{C2} = \begin{bmatrix} -a_1\dot{\theta}_1 \sin\theta_1 - l_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \\ a_1\dot{\theta}_1\cos\theta_1 + l_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \\ 0 \end{bmatrix} $$

- Centroid velocity of link 3.

We have:

$$ {}^{0}\underline{r}_{C3} = {}^{0}\underline{C}_{3}.{}^{3}\underline{r}_{C3} $$

$$ = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & a_1\cos(\theta_1)+a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & a_1\sin(\theta_1)+a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & -1 & d_1+d_2-q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -l_4 \\ 1 \end{bmatrix} $$

$$ = \begin{bmatrix} a_1\cos\theta_1 + a_2\cos(\theta_1+\theta_2) \\ a_1\sin\theta_1 + a_2\sin(\theta_1+\theta_2) \\ d_1+d_2+l_4-q_3 \\ 1 \end{bmatrix} $$

Differentiate with respect to time:

$$ {}^{0}\underline{\dot{r}}_{C3} = \begin{bmatrix} -a_1\dot{\theta}_1\sin\theta_1 - a_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \\ a_1\dot{\theta}_1\cos\theta_1 + a_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \\ \dot{q}_3 \\ 0 \end{bmatrix} $$

So:

$$ {}^{0}\underline{v}_{C3} = \begin{bmatrix} -a_1\dot{\theta}_1\sin\theta_1 - a_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \\ a_1\dot{\theta}_1\cos\theta_1 + a_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \\ \dot{q}_3 \end{bmatrix} $$

b) Angular velocity

- Link 1.

We have:

$$ \underline{R}_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

$$ \Rightarrow \underline{\tilde{\omega}}_1 = \underline{R}_1^T\,\underline{\dot{R}}_1 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\dot{\theta}_1\sin\theta_1 & -\dot{\theta}_1\cos\theta_1 & 0 \\ \dot{\theta}_1\cos\theta_1 & -\dot{\theta}_1\sin\theta_1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\theta}_1 & 0 \\ \dot{\theta}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} $$

So angular velocity of link 1 is:

$$ \underline{\omega}_1 = [0 \quad 0 \quad \dot{\theta}_1]^T $$

- Link 2 and 3:

Since joint 3 is a prismatic joint, the angular velocity of link 3 is equal to the angular velocity of link 2.

$$\underline{R}_2 = \underline{R}_3 = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\underline{\tilde{\omega}}_2 = \underline{R}_2^T \underline{\dot{R}}_2$$

$$= \begin{bmatrix} \cos(\theta_1+\theta_2) & \sin(\theta_1+\theta_2) & 0 \\ -\sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} -(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) & -(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) & 0 \\ (\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) & -(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -(\dot{\theta}_1+\dot{\theta}_2) & 0 \\ \dot{\theta}_1+\dot{\theta}_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

So angular velocity of link 2 and 3 is:

$$\underline{\omega}_2 = \underline{\omega}_3 = [0 \quad 0 \quad \dot{\theta}_1+\dot{\theta}_2]^T$$

2.4 SET UP FORMULAS FOR CALCULATING THE CENTROID ACCELERATION OF THE LINKS

a) Linear acceleration.

Taking derivatives of linear velocities of each link, we have formulas for linear acceleration of each link as follows:

- Link 1

$$^0\underline{v}_{C1} = \begin{bmatrix} -l_1\dot{\theta}\sin\theta_1 \\ l_1\dot{\theta}\cos\theta_1 \\ 0 \end{bmatrix} \Rightarrow \underline{a}_{C1} = \begin{bmatrix} -l_1s_1\ddot{\theta}_1 - l_1c_1\dot{\theta}_1^{\,2} \\ l_1c_1\ddot{\theta}_1 - l_1s_1\dot{\theta}_1^{\,2} \\ 0 \end{bmatrix}$$

- Link 2

$$^0\underline{v}_{C2} = \begin{bmatrix} -a_1\dot{\theta}_1\sin\theta_1 - l_2(\dot{\theta}_1+\dot{\theta}_2)\sin(\theta_1+\theta_2) \\ a_1\dot{\theta}_1\cos\theta_1 + l_2(\dot{\theta}_1+\dot{\theta}_2)\cos(\theta_1+\theta_2) \\ 0 \end{bmatrix}$$

$$\Rightarrow {}^0\underline{a}_{C2} = \begin{bmatrix} -l_2s_{12}\ddot{\theta}_{12} - a_1s_1\ddot{\theta}_1 - l_2c_{12}\dot{\theta}_{12}^{\,2} - a_1c_1\dot{\theta}_1^{\,2} \\ l_2c_{12}\ddot{\theta}_{12} + a_1c_1\ddot{\theta}_1 - l_2s_{12}\dot{\theta}_{12}^{\,2} - a_1s_1\dot{\theta}_1^{\,2} \\ 0 \end{bmatrix}$$

- Link 3

$$
{}^{0}\underline{v}_{C3} = \begin{bmatrix} -a_1\dot{\theta}_1 \sin\theta_1 - a_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1+\theta_2) \\ a_1\dot{\theta}_1 \cos\theta_1 + a_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1+\theta_2) \\ \dot{q}_3 \end{bmatrix}
$$

$$
\Rightarrow {}^{0}\underline{a}_{C3} = \begin{bmatrix} -a_2 s_{12}\ddot{\theta}_{12} - a_1 s_1 \ddot{\theta}_1 - a_2 c_{12}\dot{\theta}_{12}^{\ 2} - a_1 c_1 \dot{\theta}_1^{\ 2} \\ a_2 c_{12}\ddot{\theta}_{12} + a_1 c_1 \ddot{\theta}_1 - a_2 s_{12}\dot{\theta}_{12}^{\ 2} - a_1 s_1 \dot{\theta}_1^{\ 2} \\ \ddot{q}_3 \end{bmatrix}
$$

b) Angular acceleration

Taking derivatives of angular velocities of each link, we have formulas for angular acceleration of each link as follows:

- Link 1

$$
\underline{\omega}_1 = [0 \quad 0 \quad \dot{\theta}_1]^T \rightarrow \underline{\varepsilon}_1 = \underline{\dot{\omega}}_1 = [0 \quad 0 \quad \ddot{\theta}_1]^T
$$

- Link 2 and Link 3

$$
\underline{\omega}_2 = \underline{\omega}_3 = [0 \quad 0 \quad \dot{\theta}_1 + \dot{\theta}_2]^T \rightarrow \underline{\varepsilon}_2 = \underline{\varepsilon}_3 = \underline{\dot{\omega}}_2 = \underline{\dot{\omega}}_3 = [0 \quad 0 \quad \ddot{\theta}_1 + \ddot{\theta}_2]^T
$$

2.5 DYNAMICS

The general differential equation describing motion of the robot is:

$$
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Q
$$

We have:

Mass matrix M:

$$
M(q) = \begin{bmatrix} m_{11}(q) & m_{12}(q) & m_{13}(q) \\ m_{21}(q) & m_{22}(q) & m_{23}(q) \\ m_{31}(q) & m_{32}(q) & m_{33}(q) \end{bmatrix}
$$

where:

$$
\begin{cases} m_{11}(q) = I_{zz1} + I_{zz2} + I_{zz3} + \frac{a_1^2}{4} m_1 + (a_1^2 + \frac{a_2^2}{4} + a_1 a_2 c_2) m_2 + (a_1^2 + a_2^2 + 2a_1 a_2 c_2) m_3 \\ m_{12}(q) = m_{21}(q) = I_{zz2} + I_{zz3} + \left(\frac{a_2^2}{4} + \frac{1}{2} a_1 a_2 c_2\right) m_2 + (a_2^2 + a_1 a_2 c_2) m_3 \\ m_{22}(q) = I_{zz2} + I_{zz3} + \frac{a_2^2}{4} m_2 + a_2^2 m_3 \\ m_{33}(q) = m_3 \\ m_{13}(q) = m_{31}(q) = m_{23}(q) = m_{32}(q) = 0 \end{cases}
$$

Coriolis matrix C:

$$
C(q,\dot{q}) = \begin{bmatrix} c_{11}(q,\dot{q}) & c_{12}(q,\dot{q}) & c_{13}(q,\dot{q}) \\ c_{21}(q,\dot{q}) & c_{22}(q,\dot{q}) & c_{23}(q,\dot{q}) \\ c_{31}(q,\dot{q}) & c_{32}(q,\dot{q}) & c_{33}(q,\dot{q}) \end{bmatrix}
$$

where: $$\begin{cases} c_{11}(q,\dot{q}) = -\left(\frac{m_2}{2}+m_3\right)a_1a_2s_2\dot{q}_2 \\ c_{12}(q,\dot{q}) = -\left(\frac{m_2}{2}+m_3\right)a_1a_2s_2\dot{q}_1 - \left(\frac{m_2}{2}+m_3\right)a_1a_2s_2\dot{q}_2 \\ c_{21}(q,\dot{q}) = \left(\frac{m_2}{2}+m_3\right)a_1a_2s_2\dot{q}_1 \\ c_{13}(q,\dot{q}) = c_{22}(q,\dot{q}) = c_{23}(q,\dot{q}) = c_{31}(q,\dot{q}) = c_{32}(q,\dot{q}) = c_{33}(q,\dot{q}) = 0 \end{cases}$$

Gravitational effect matrix G:

$$G(q) = [0 \quad 0 \quad -m_3g]^T$$

Non-conservative forces Q:

$$Q = U + J_E^T F_E$$

Where:

- The force that acts on the end effector is: $F_E = [0 \quad 0 \quad -F_z]^T$
- $U = [\tau_1 \quad \tau_2 \quad \tau_3]^T$ is the vector of actuation forces at each joint.

So $Q = [0 \quad 0 \quad -F_z]^T + [\tau_1 \quad \tau_2 \quad \tau_3]^T$

Remark:

The solution to the inverse dynamics problem is:

$$\begin{bmatrix} m_{11}(q) & m_{12}(q) & 0 \\ m_{21}(q) & m_{22}(q) & 0 \\ 0 & 0 & m_{33}(q) \end{bmatrix}\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} + \begin{bmatrix} c_{11}(q,\dot{q}) & c_{12}(q,\dot{q}) & 0 \\ c_{21}(q,\dot{q}) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_3g \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ F_z \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}$$

We can derive the required actuation forces at each joint as follows:

$$U = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} m_{11}(q)\ddot{q}_1 + m_{12}(q)\ddot{q}_2 + c_{11}(q,\dot{q})\dot{q}_1 + c_{12}(q,\dot{q})\dot{q}_2 \\ m_{21}(q)\ddot{q}_1 + m_{22}(q)\ddot{q}_2 + c_{21}(q,\dot{q})\dot{q}_1 \\ m_{33}(q)\ddot{q}_3 - m_3g + F_z \end{bmatrix}$$

2.6 DESIGN OF THE TRAJECTORY PLANNING IN THE WORKSPACE

a) Linear interpolation

For the SCARA robot performing the task of picking and placing objects, we will choose a trajectory that is a straight line with a trapezoidal velocity profile. Choosing this type of trajectory has 2 advantages, it both simplifies calculations and design, and allows the robot to have fast, accurate and agile movements when picking up or placing objects.

Straight line equation passing through $M(x0;\, y0;\, zo)$ and $N(xe\,;\, ye\,;\, ze)$ is:

$$\frac{x - x_0}{x_e - x_0} = \frac{y - y_0}{y_e - y_0} = \frac{z - z_0}{z_e - z_0}$$

The equations of y and z in term of x are:

$$\begin{cases} y = \dfrac{y_e - y_0}{x_e - x_0}\,x + \dfrac{y_0x_e - y_ex_0}{x_e - x_0} \\ z = \dfrac{z_e - z_0}{x_e - x_0}\,x + \dfrac{z_0x_e - z_ex_0}{x_e - x_0} \end{cases}$$

We have these conditions:

$$\begin{cases} x_E(0) = x_0 \\ \dot{x}_E(0) = 0 \\ x_E(t_e) = x_e \\ \dot{x}_E(t_e) = 0 \end{cases}$$

Designing a trapezoidal velocity profile:

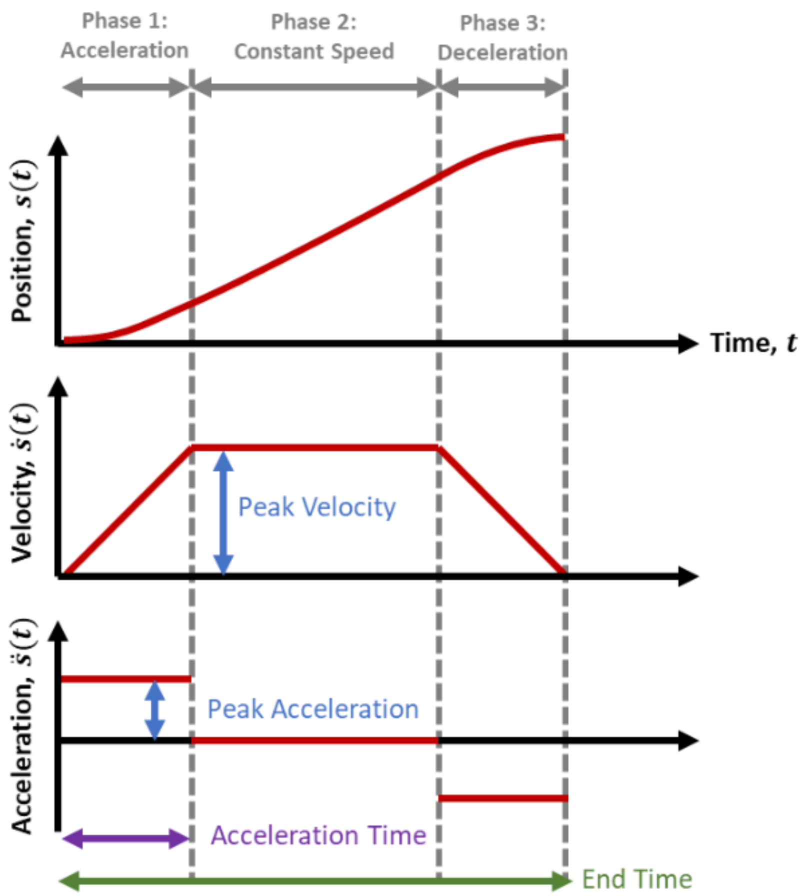

Figure 2.5 Trapezoidal velocity profile

A typical trapezoidal velocity profile includes the following three phases:

- Acceleration phase: The acceleration of this phase is a positive constant, velocity is a first-order function, and the position is parabolic.
- Constant velocity phase: The acceleration of this phase is zero, the velocity is a constant corresponding to the maximum velocity in the trajectory and the position has the form of a first order function.
- Deceleration phase: The acceleration of this phase is a negative constant, the velocity is a first-order function, and the position is parabolic.

We choose the parameters for the trajectory as follows:

$\begin{cases} v_{max} = \frac{1}{\sqrt{3}} \times 9570 = 5530\ mm/s \\ \qquad t_e\ \text{tùy chọn} \end{cases}$ where 9570 mm/s is the maximum overall speed of the Scara and t_e (s) is the travel time from M to N that the user can choose such that the acceleration (shown below) does not excess the maximum allowable value.

We choose the acceleration time: $t_a = \frac{1}{3} t_e$

Thus, acceleration in the acceleration/deceleration phase is $a = \frac{v_{max}}{t_a} = \frac{3\, v_{max}}{t_e}$

Finally, we can design the trapezoidal velocity profile in 3 phases as follows:

- Acceleration phase: $\begin{cases} \ddot{x}(t) = a \\ \dot{x}(t) = a\, t \\ x(t) = a \frac{t^2}{2} \end{cases}$ where $t = [0, t_a]$
- Constant velocity phase: $\begin{cases} \ddot{x}(t) = 0 \\ \dot{x}(t) = v_{max} \\ x(t) = v_{max}\, t - \frac{v_{max}^2}{2a} \end{cases}$ where $t = (t_a, t_e - t_a]$
- Deceleration phase: $\begin{cases} \ddot{x}(t) = -a \\ \dot{x}(t) = a\,(t_e - t) \\ x(t) = \frac{2\, a\, v_{max}\, t_e - 2v^2 - a^2 (t - t_e)^2}{2\, a} \end{cases}$ where $t = (t_e - t_a, t_e]$

b) Circular interpolation

For SCARA robots that perform tasks such as welding and cutting metal, we will design another common trajectory that is an arc trajectory. Combined with the above-mentioned straight-line trajectory, we can give robots more complex trajectories.

Since the method of designing a trapezoidal velocity profile is completely similar to the case of straight-line trajectories, in this section the author only shows the method of creating a trajectory that has the shape of a circular arc between two given points.

Because the structure of the SCARA SR8 plus robot has 2 revolute joints about the Oz axis and 1 prismatic joint along the Oz direction, to simplify the problem, we only designed a circular arc trajectory in the Oxy plane.

Suppose we want the Scara to move its end effector from $O(x_0, y_0, z_0)$ to $E(x_e, y_e, z_0)$ following a circular arc trajectory having radius of R(m) in the period $t_e(s)$.

We know that the center of the arc through 2 points O, E is on the perpendicular bisector line of the line segment O, E so the normal vector of the perpendicular bisector line of the line segment O, E is: $n = (x_n, y_n)$ where $\begin{cases} x_n = x_e - x_0 \\ y_n = y_e - y_0 \end{cases}$ and midpoint of line segment OE is: $m = (x_m, y_m)$ trong đó $\begin{cases} x_m = \frac{(x_e + x_0)}{2} \\ y_m = \frac{(y_e + y_0)}{2} \end{cases}$

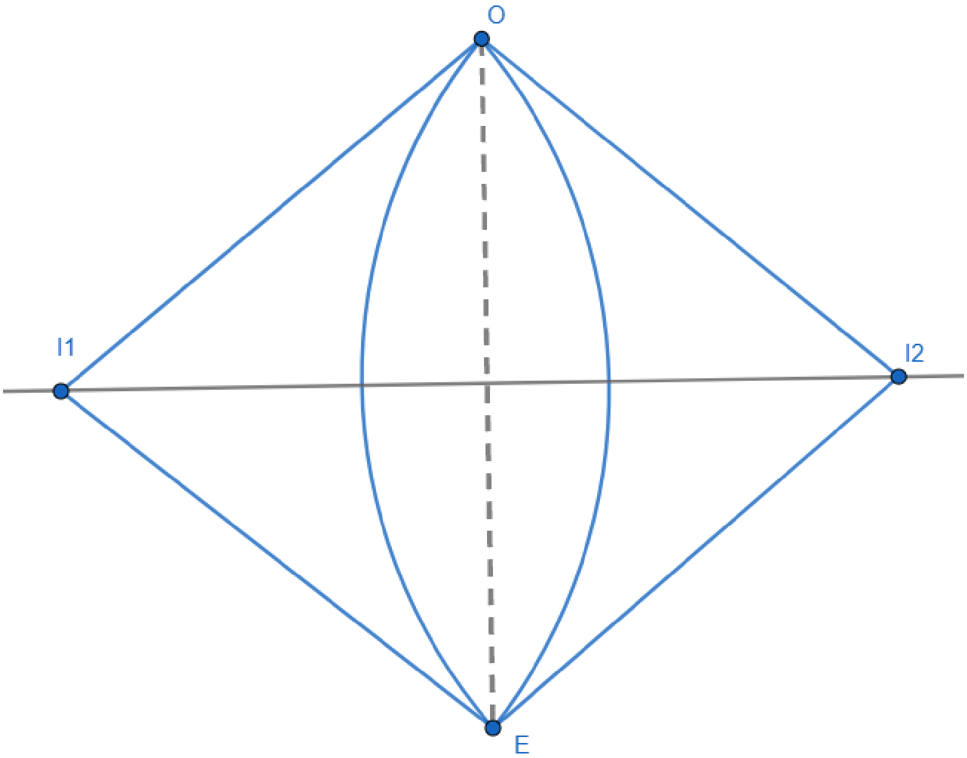

Figure 2.6 Illustration for circular interpolation

To find the center of the arc, we solve the following system of equations:

$$\begin{cases} x_n(x - x_m) + y_n(y - y_m) = 0 & (*) \\ (x - x_0)^2 + (y - y_0)^2 = R^2 & (**) \end{cases}$$

where (*) is the perpendicular bisector line equation of OE and (**) is the circle equation centered at $I(x, y)$ through 2 points O, E having radius R.

The above system of equations will give 2 real or no real solutions. The case of no real solution occurs when the radius R is less than the distance OE. In case there are 2 real solutions corresponding with 2 coordinates of the center $I_1(x_1, y_1)$ *and* $I_2(x_2, y_2)$, one of which yields the methods of negative circular interpolation and the other yields the method of positive circular interpolation. In particular, negative circular interpolation corresponds to the center I which satisfies the condition that the cross product of 2 vectors IO and IE results in a vector pointing in the opposite direction of the Oz axis. Positive circular interpolation corresponds to center I which satisfies the condition that the cross product of 2 vectors IO and IE results in a vector pointing in the same direction as the Oz axis.

So, the intermediate points along the trajectory can be calculated as follows:

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} = R \begin{pmatrix} \cos\left(\frac{t}{t_e}(\theta_2 - \theta_1) + \theta_1\right) \\ \sin(\frac{t}{t_e}(\theta_2 - \theta_1) + \theta_1) \end{pmatrix} + I_i$$

Where, $i \in \{1; 2\}$, θ_1 is the angle of the vector IO and θ_2 is the angle of the vector IE.

2.7 CONCLUSION OF CHAPTER 2

In chapter 2, I have shown how to find the end effector coordinates and the rotation angle of the joints by the problem of forward kinematics, inverse kinematics.

Calculate the velocity and acceleration of the links of the robot.

Calculation of control forces at joints through dynamics problems.

Demonstrate and illustrate the workspace of the robot.

Design of linear and circular motion trajectories in the robot's workspace.

Chapter 3:CONTROL SYSTEM MODELING

3.1 MODELING OF EACH LINK OF THE SCARA

3.1.1 Modeling DC motor

Mechanical part:

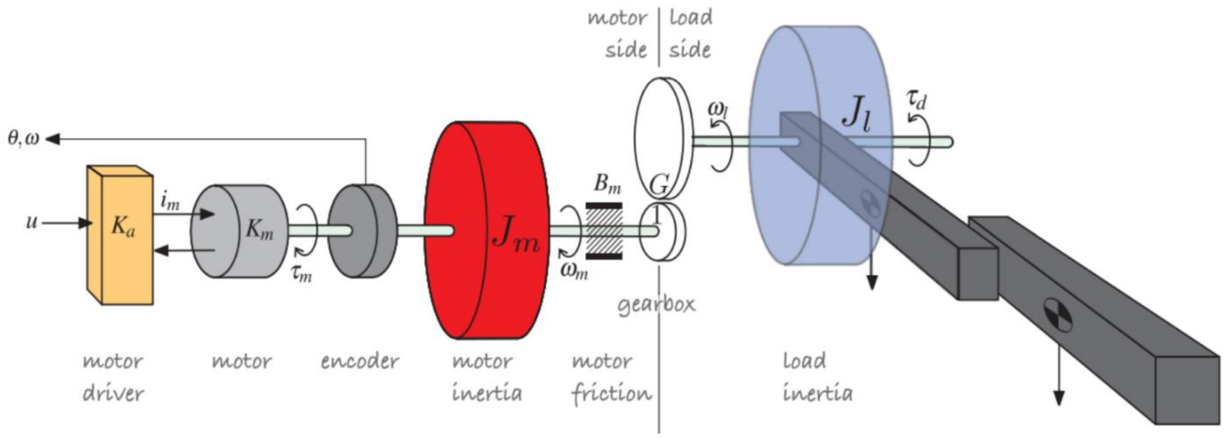

Figure 3.1 One joint of the Scara

In which:

- J_m is the moment of inertia of the motor's shaft.
- J_l is the moment of inertia of the link mounted on that joint.
- τ_m is the torque of the motor.
- τ_l is the torque incurred on the motor by the load.
- θ_m is the rotational angle of the motor.
- θ_s is the rotational angle of the load.
- B_m is the damping factor of the motor.
- B_l is the damping factor of the load.
- $n = \frac{\omega_s}{\omega_m}$ is the gear ratio between the motor and the load

According to the D'Alembert principle, we have:

$$\tau_l - B_l\dot{\theta}_s = J_l\ddot{\theta}_s$$

$$\Leftrightarrow \tau_l - B_l\omega_s = J_l\dot{\omega}_s$$

Use the above principle for the motor shaft we have:

$$\tau_m - n\tau_l - B_m\dot{\theta}_m = J_m\ddot{\theta}_m$$

Inferring:

$$\tau_m = (J_m + n^2J_l)\ddot{\theta}_m + (B_m + n^2B_l)\dot{\theta}_m$$

Collapse the equation into:

$$\boxed{\tau_m = J_{td}\dot{\omega}_m + B_{td}\omega_m}$$

Where: $\begin{cases} J_{td} = J_m + n^2J_l \\ B_{td} = B_m + n^2B_l \end{cases}$

Electrical part:

DC motors (DC electric motors) are independently excited, controlled by armature voltage. The schematic diagram of this type of motor is shown in the figure below.

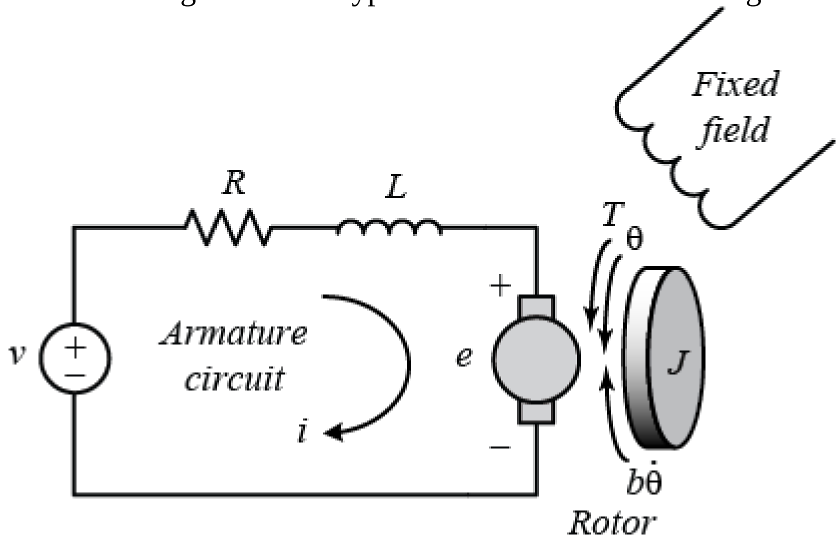

Figure 3.2 Armature circuit of the DC motor

With:

- Input is the voltage $v = u(Volt)$
- Output is the angular velocity of the motor $\omega(rad/s)$

<u>Armature circuit equation:</u>

$$U = L\frac{di}{dt} + Ri + K_e\omega$$

Where:

- R is armature resistance (Ω)
- L is armature inductance (H)
- i is armature current (A)
- K_e is the electromotive-force constant (Vs/rad)

Apply Laplace transform, we get:

$$U(s) = LsI(s) + RI(s) + K_e\omega(s)$$

Block diagram:

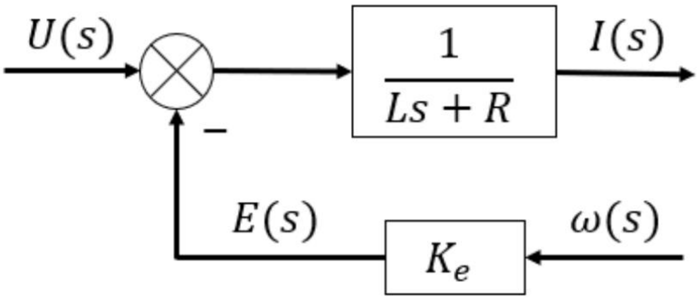

Figure 3.3 Block diagram of the armature circuit

<u>Electromagnetic torque equation of motor:</u>

$$M = K_m i$$

Where K_m is the torque constant of the motor ($N.m/A$)

Apply Laplace transform, we get:

$$M(s) = K_m I(s)$$

Block diagram:

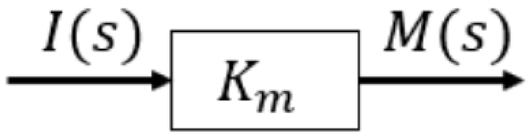

Figure 3.4 Block diagram of the electromagnetic torque equation

<u>Torque balance equation on the motor shaft:</u>

$$M = J\frac{d\omega}{dt} + B\omega + M_t$$

Where:

- J is the moment of inertia of the motor and the load with respect to the motor shaft $(kg.m^2)$
- B is the damping coefficient of the engine and the load with respect to the engine shaft $(kg.m^2)$
- M_t is the disturbance $(N.m)$

Apply Laplace transform, we get:

$$M(s) = Js\omega(s) + B\omega(s) + M_t(s)$$

Block diagram:

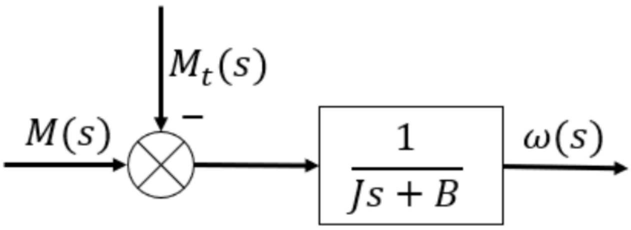

Figure 3.5 Block diagram of the torque balance equation on the motor shaft

<u>Summary:</u>

Combining the block diagrams, we obtain the overall block diagram of the motor:

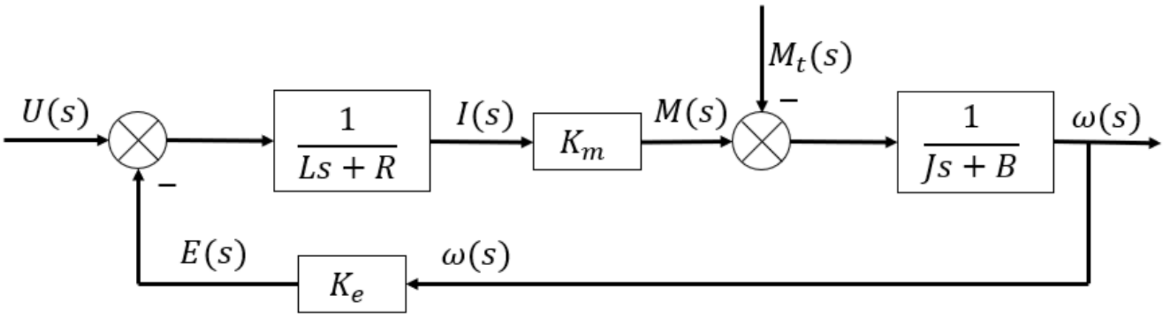

Figure 3.6 Block diagram of the motor with voltage as input and angular velocity as output

Transfer function of a DC motor with voltage as input and angular velocity as output:

$$G_{dc}(s) = \frac{\omega(s)}{U(s)} = \frac{\frac{K_m}{(Ls+R)(Js+B)}}{1+\frac{K_mK_e}{(Ls+R)(Js+B)}} = \frac{K_m}{LJs^2+(LB+RJ)s+(K_mK_e+RB)}$$

In this project, we are interested in controlling the position of the robot, specifically the angle of rotation at the joints, so when modeling the motor, the output signal must be the angular displacement.

Let θ_m be the rotation angle of the motor. Hence, the rotation angle is the integral of the angular velocity, in the Laplace domain we have:

$$\theta_m(s) = \frac{\omega(s)}{s}$$

Thus, Transfer function of a DC motor with voltage as input and angular displacement as output is:

$$G_{dc}(s) = \frac{\theta(s)}{U(s)} = \frac{K_m}{s(LJs^2+(LB+RJ)s+(K_mK_e+RB))}$$

In short, we have:

$$G_{dc}(s) = \frac{K_{td}}{T_1s^3+T_2s^2+s}$$

Where : $\begin{cases} T_1 = \frac{\tau_t\tau_c RB_{td}}{K_mK_e+RB_{td}} \\ T_2 = \frac{(\tau_t+\tau_c)RB_{td}}{K_mK_e+RB_{td}} \\ K_{td} = \frac{K_m}{K_mK_e+RB_{td}} \end{cases}$ with $\begin{cases} \tau_t = \frac{L}{R} \\ \tau_c = \frac{J_{td}}{B_{td}} \end{cases}$

Block diagram:

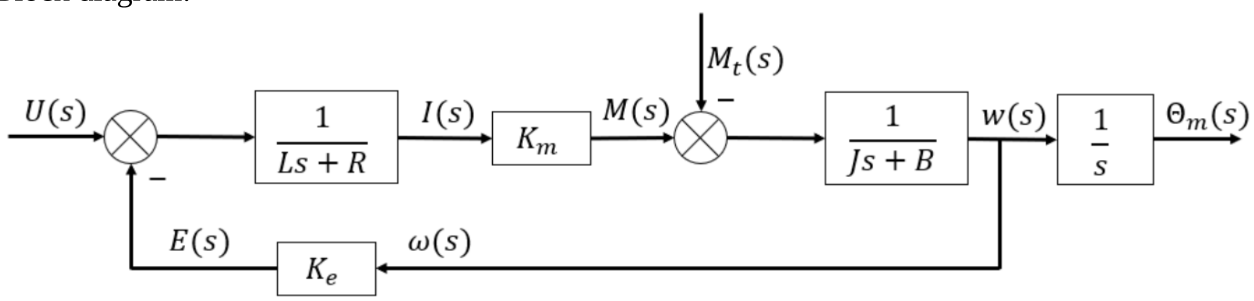

Figure 3.7 Block diagram of the motor with voltage as input and angular displacement as output

Table 3.1 DC servo motor specification used in this project:

T (Nm)	48
I (A)	54.7
Ke (Vs/rad)	0.957
Km (Nm/A)	0.957
Jm (kg.m2)	125×10^{-4}
R (Ω)	1.45
L (mH)	5.4
B (Nms/rad)	6.78×10^{-4}

3.1.2 Modeling of individual link

Ball screw as link 3:

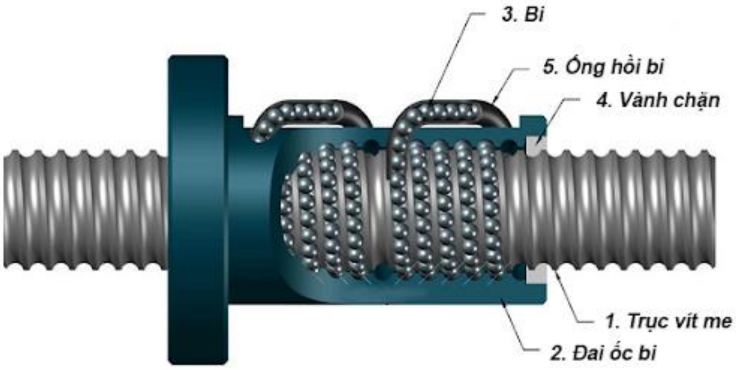

Figure 3.8 Ball screw

where:

- Input: angular velocity $\omega(t)$
- Output: linear displacement $y(t)$ of link 3
- Let p(m) be the pitch.

We have the equation on the Laplace domain as follows:

$$Y(s) = \frac{p}{2\pi} \times \frac{\omega(s)}{s}$$

So, the transfer function is:

$$\frac{Y(s)}{\omega(s)} = \frac{K}{s}$$

Where: $K = \frac{p}{2\pi}$

In combination with the block diagram of the motor, we have the block diagram of link 3 as:

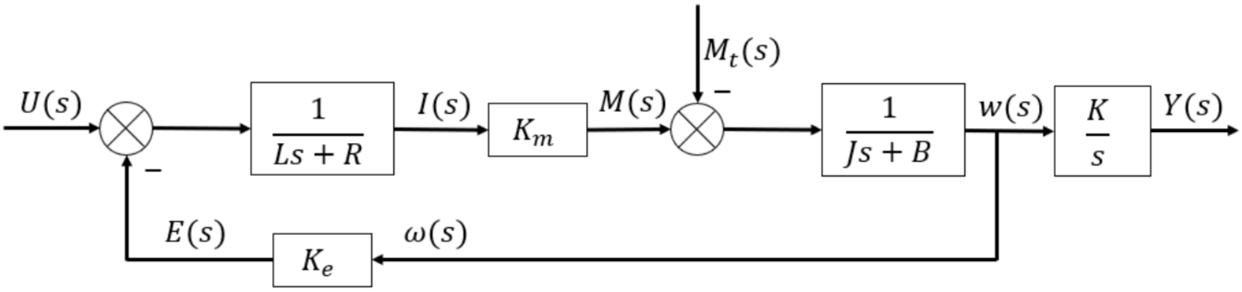

Figure 3.9 Block diagram of link 3

The transfer function of link 3 is:

$$G_3(s) = \frac{Y(s)}{U(s)} = \frac{K_{td}\frac{p}{2\pi}}{T_1 s^3 + T_2 s^2 + s}$$

Using Solidworks software to calculate the moment of inertia, we have:

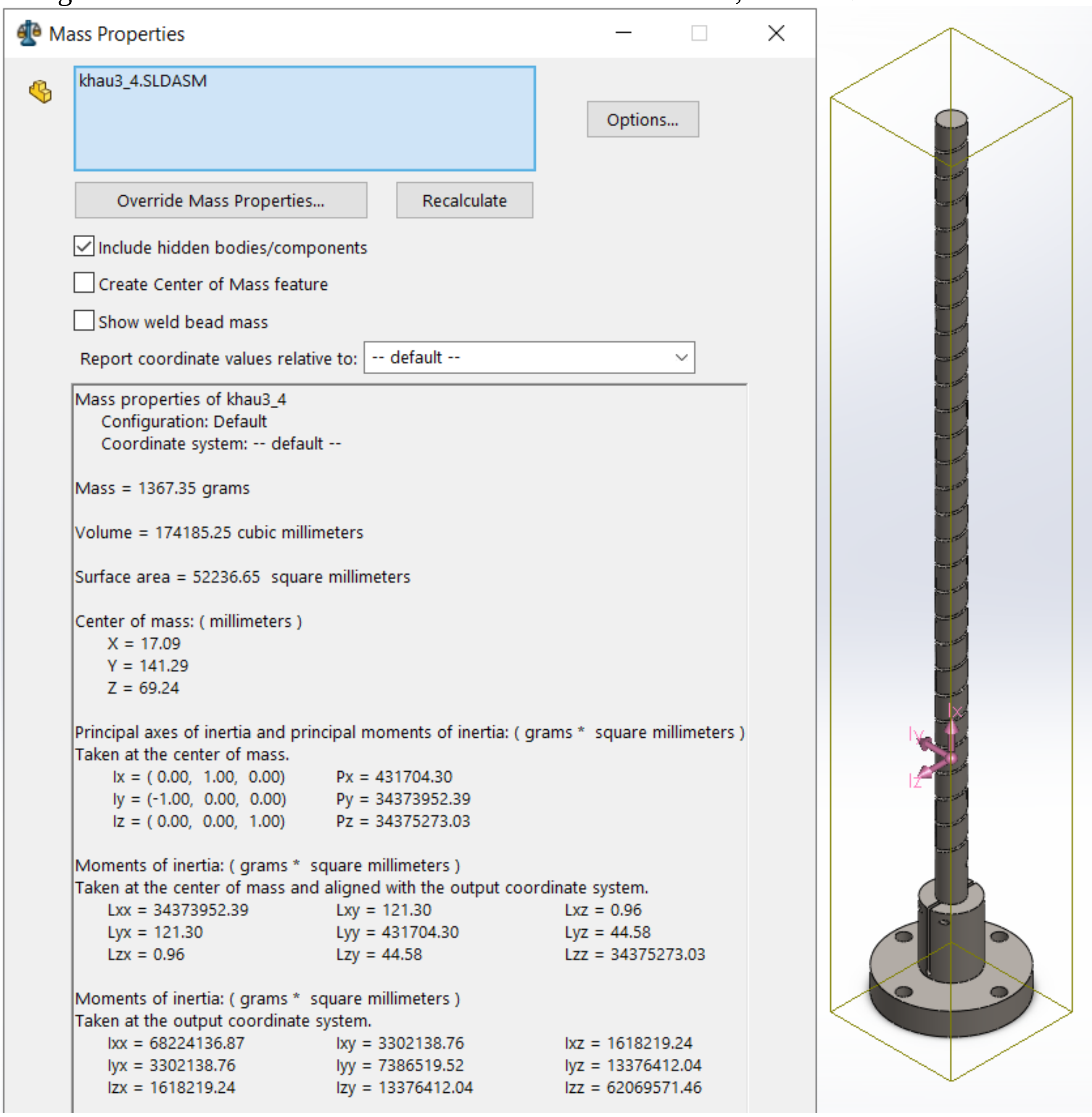

Figure 3.10 Mass properties of link 3

We have: $\begin{cases} J_{l3} = 431704 \times 10^{-9}\ kg.m^2 \\ n_3 = 10 \\ p_3 = 15 \times 10^{-3}\ m \\ J_m = 125 \times 10^{-4}\ kg.m^2 \\ B_m = 6.78 \times 10^{-4}\ Nms/rad \\ B_l = 0.001\ Nms/rad \\ K_m = K_e = 0.957 \\ R = 1.45\ \Omega \\ L = 5.4 \times 10^{-3}\ H \end{cases}$

Thus, we calculate the parameters: $\begin{cases} J_{td3} = J_{m3} + n_3^2 J_{l3} = 0.0557 \\ B_{td3} = B_{m3} + n_3^2 B_{l3} = 0.1007 \\ K_{td} = \frac{K_m}{K_m K_e + RB_{td}} = 0.9013 \\ T_1 = \frac{\tau_t \tau_c RB_{td}}{K_m K_e + RB_{td}} = 2.831 \times 10^{-4} \\ T_2 = \frac{(\tau_t + \tau_c) RB_{td}}{K_m K_e + RB_{td}} = 0.0765 \end{cases}$

$$\Rightarrow \boxed{G_3(s) = \frac{K_{td}\frac{p}{2\pi}}{T_1 s^3 + T_2 s^2 + s} = \frac{0.002152}{0.0002831\ s^3 + 0.07653\ s^2 + s}}$$

Stability test of link 3

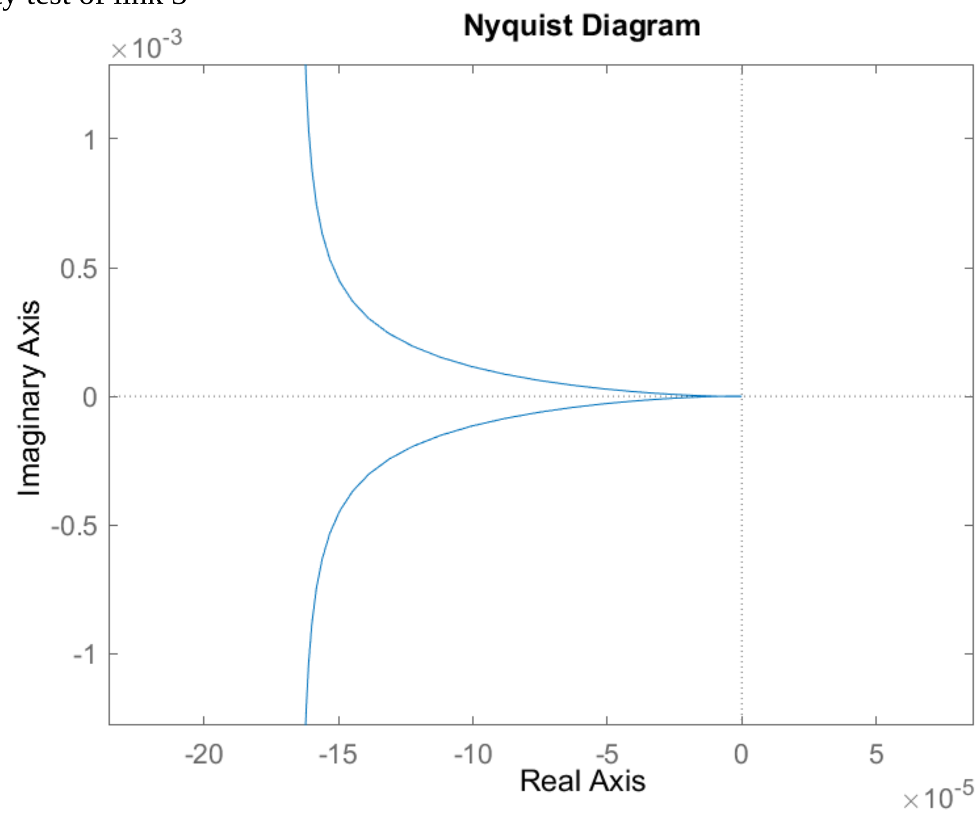

Figure 3.11 Nyquist diagram of link 3

We see that the Nyquist diagram does not cross the point (−1 + 0j) so the system will be stable.

Modeling of link 2:

Because link 2 only uses a combination of DC motor and gear reducer, without using additional drive mechanisms, the transfer function of link 2 is the transfer function of the motor built in section **3.1.1.**

We have:

$$G_2(s) = \frac{K_{td}}{T_1 s^3 + T_2 s^2 + s}$$

Where: $\begin{cases} T_1 = \dfrac{\tau_t \tau_c R B_{td}}{K_m K_e + R B_{td}} \\ T_2 = \dfrac{(\tau_t + \tau_c) R B_{td}}{K_m K_e + R B_{td}} \\ K_{td} = \dfrac{K_m}{K_m K_e + R B_{td}} \end{cases}$ with $\begin{cases} \tau_t = \dfrac{L}{R} \\ \tau_c = \dfrac{J_{td}}{B_{td}} \end{cases}$

Using Solidworks software to calculate the moment of inertia, we have:

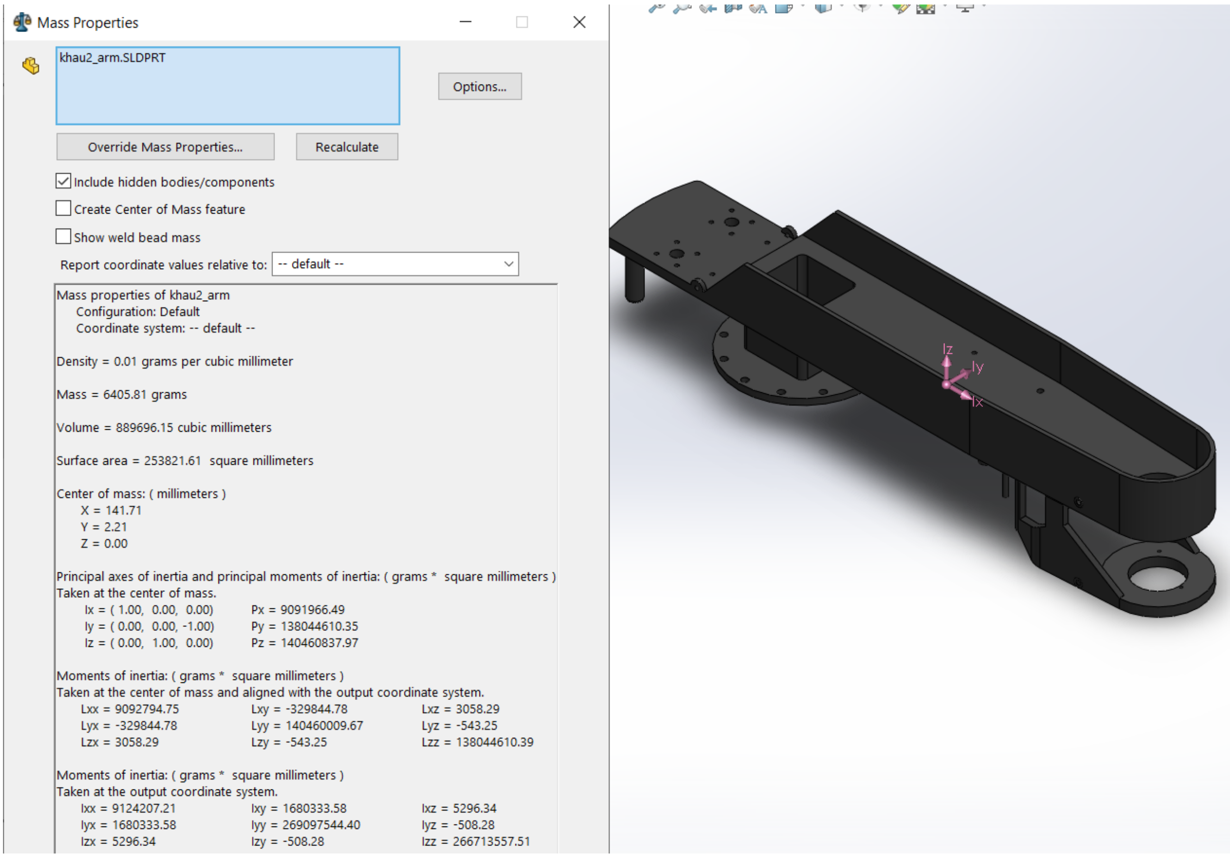

Figure 3.12 Mass properties of link 2

We have: $\begin{cases} J_{l2} = 140.46 \times 10^{-6}\ kg.m^2 \\ n_2 = \frac{1}{80} \\ J_m = 125 \times 10^{-4}\ kg.m^2 \\ B_m = 6.78 \times 10^{-4}\ Nms/rad \\ B_l = 0.001\ Nms/rad \\ K_m = K_e = 0.957 \\ R = 1.45\ \Omega \\ L = 5.4 \times 10^{-3}\ H \end{cases}$

Thus, we calculate the parameters: $\begin{cases} J_{td2} = J_{m2} + n_2^2 J_{l2} = 0.0557 \\ B_{td2} = B_{m2} + n_2^2 B_{l2} = 0.1007 \\ K_{td} = \frac{K_m}{K_m K_e + R B_{td}} = 0.9013 \\ T_1 = \frac{\tau_t \tau_c R B_{td}}{K_m K_e + R B_{td}} = 2.831 \times 10^{-4} \\ T_2 = \frac{(\tau_t + \tau_c) R B_{td}}{K_m K_e + R B_{td}} = 0.0765 \end{cases}$

$$\Rightarrow \boxed{G_2(s) = \frac{K_{td}}{T_1 s^3 + T_2 s^2 + s} = \frac{1.044}{7.36 \times 10^{-5}\ s^3 + 0.01977\ s^2 + s}}$$

Stability test of link 2

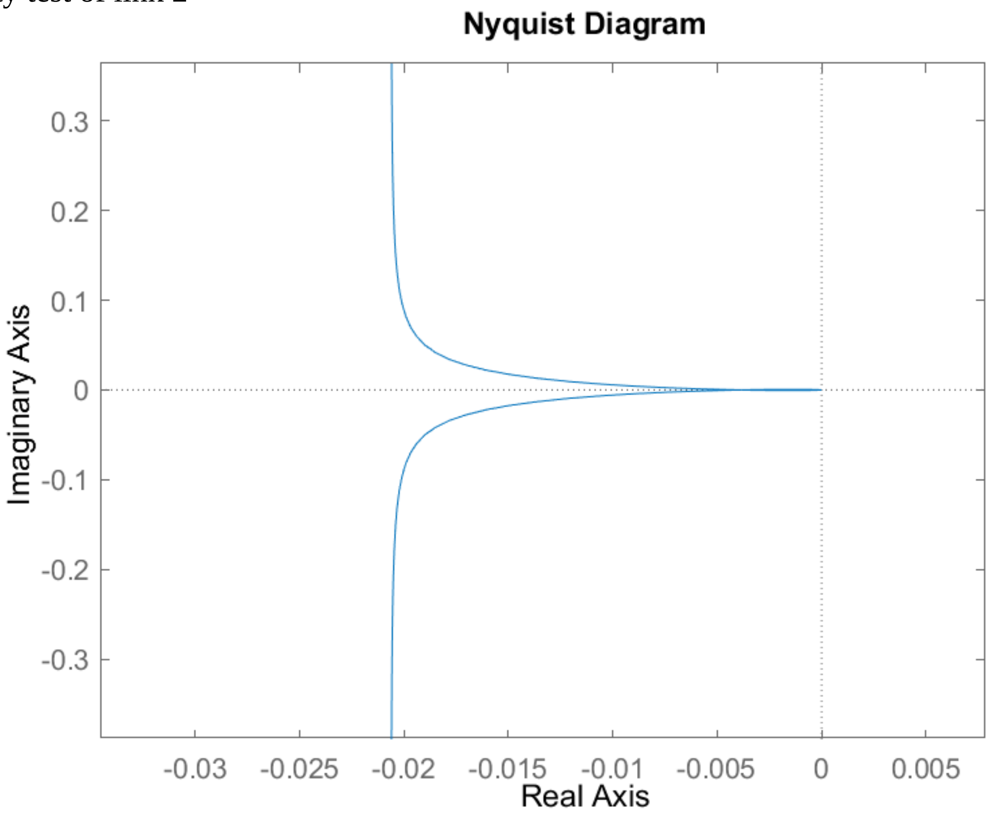

Figure 3.13 Nyquist diagram of link 2

We see that the Nyquist diagram does not cross the point (−1 + 0j) so the system will be stable.

Modeling of link 1

Because link 1 only uses a combination of DC motor and gear reducer, without using additional drive mechanisms, the transfer function of link 2 is the transfer function of the motor built in section **3.1.1.**

We have:

$$G_1(s) = \frac{K_{td}}{T_1 s^3 + T_2 s^2 + s}$$

Where: $\begin{cases} T_1 = \frac{\tau_t \tau_c R B_{td}}{K_m K_e + R B_{td}} \\ T_2 = \frac{(\tau_t + \tau_c) R B_{td}}{K_m K_e + R B_{td}} \\ K_{td} = \frac{K_m}{K_m K_e + R B_{td}} \end{cases}$ with $\begin{cases} \tau_t = \frac{L}{R} \\ \tau_c = \frac{J_{td}}{B_{td}} \end{cases}$

Using Solidworks software to calculate the moment of inertia, we have:

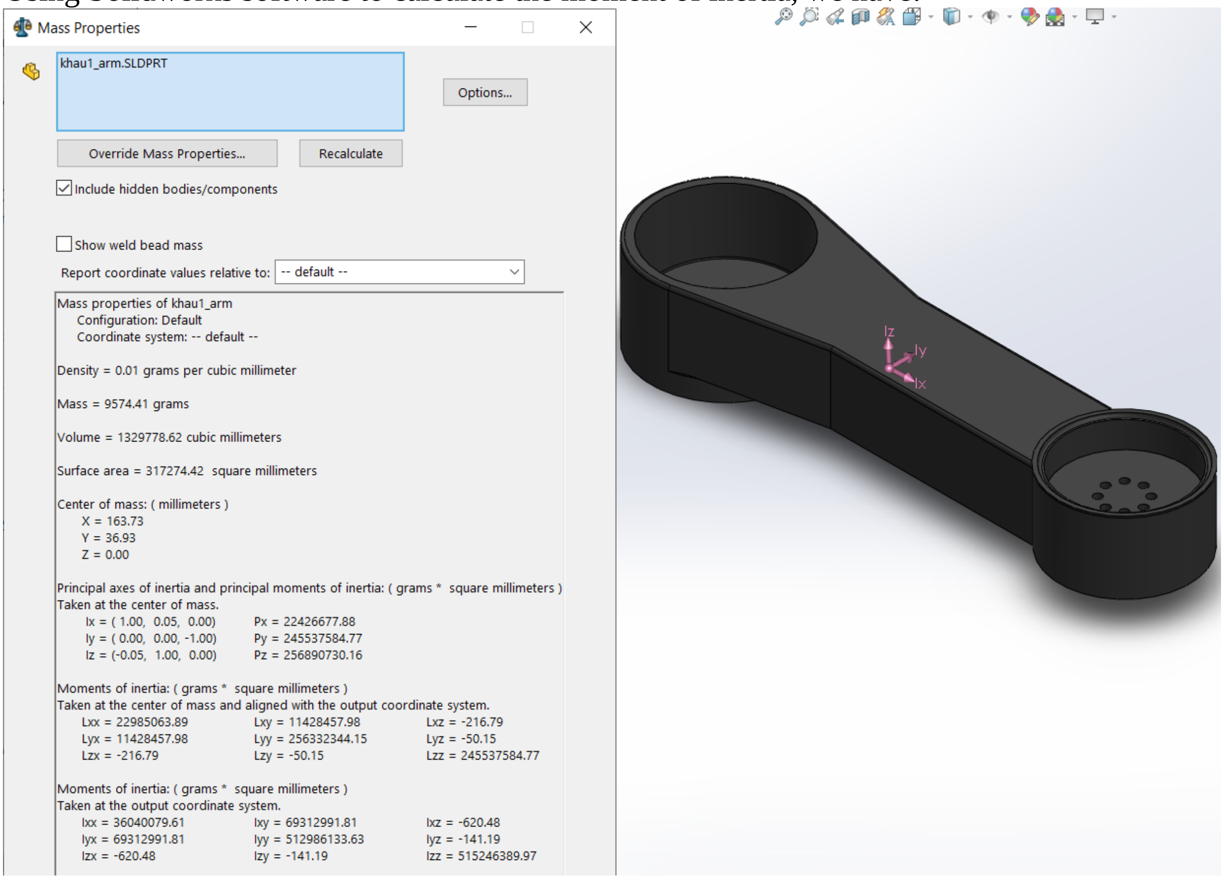

Figure 3.14 Mass properties of link 1

We have: $\begin{cases} J_{l1} = 256.33 \times 10^{-6}\ kg.m^2 \\ n_1 = \frac{1}{80} \\ J_m = 125 \times 10^{-4}\ kg.m^2 \\ B_m = 6.78 \times 10^{-4}\ Nms/rad \\ B_l = 0.001\ Nms/rad \\ K_m = K_e = 0.957 \\ R = 1.45\ \Omega \\ L = 5.4 \times 10^{-3}\ H \end{cases}$

Thus, we calculate the parameters: $\begin{cases} J_{td1} = J_{m1} + n_1^2 J_{l1} = 0.0557 \\ B_{td1} = B_{m1} + n_1^2 B_{l1} = 0.1007 \\ K_{td} = \frac{K_m}{K_m K_e + R B_{td}} = 0.9013 \\ T_1 = \frac{\tau_t \tau_c R B_{td}}{K_m K_e + R B_{td}} = 2.831 \times 10^{-4} \\ T_2 = \frac{(\tau_t + \tau_c) R B_{td}}{K_m K_e + R B_{td}} = 0.0765 \end{cases}$

$$\Rightarrow \boxed{G_1(s) = \frac{K_{td}}{T_1 s^3 + T_2 s^2 + s} = \frac{1.044}{7.36 \times 10^{-5}\ s^3 + 0.01977\ s^2 + s}}$$

Stability test of link 1

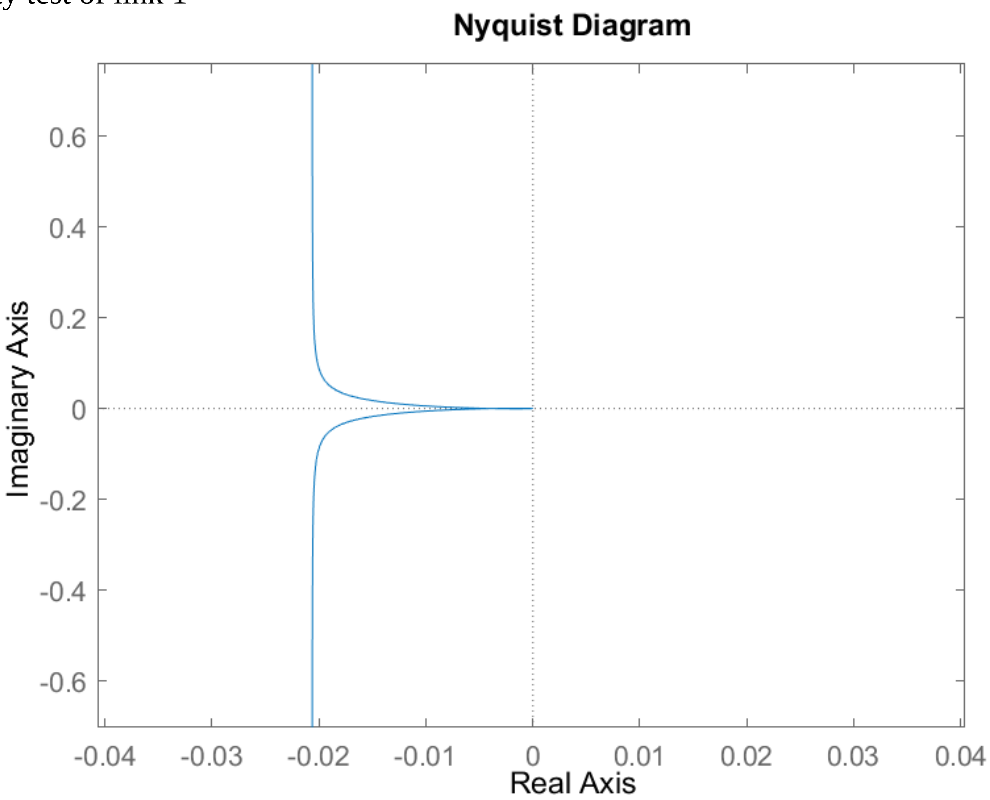

Figure 3.15 Nyquist diagram of link 1

We see that the Nyquist diagram does not cross the point $(-1 + 0j)$ so the system will be stable.

3.2 DESIGN A CONTROL SYSTEM FOR THE WHOLE ROBOT

The sequence of steps to perform the design process:

1) Determine of the physical parameters of the robot
2) Model with SolidWorks software
3) Export the corresponding CAD to MATLAB/Simscape
4) PID controller design and simulation

The SCARA SR8 plus model exported into MATLAB/SIMSCAPE:

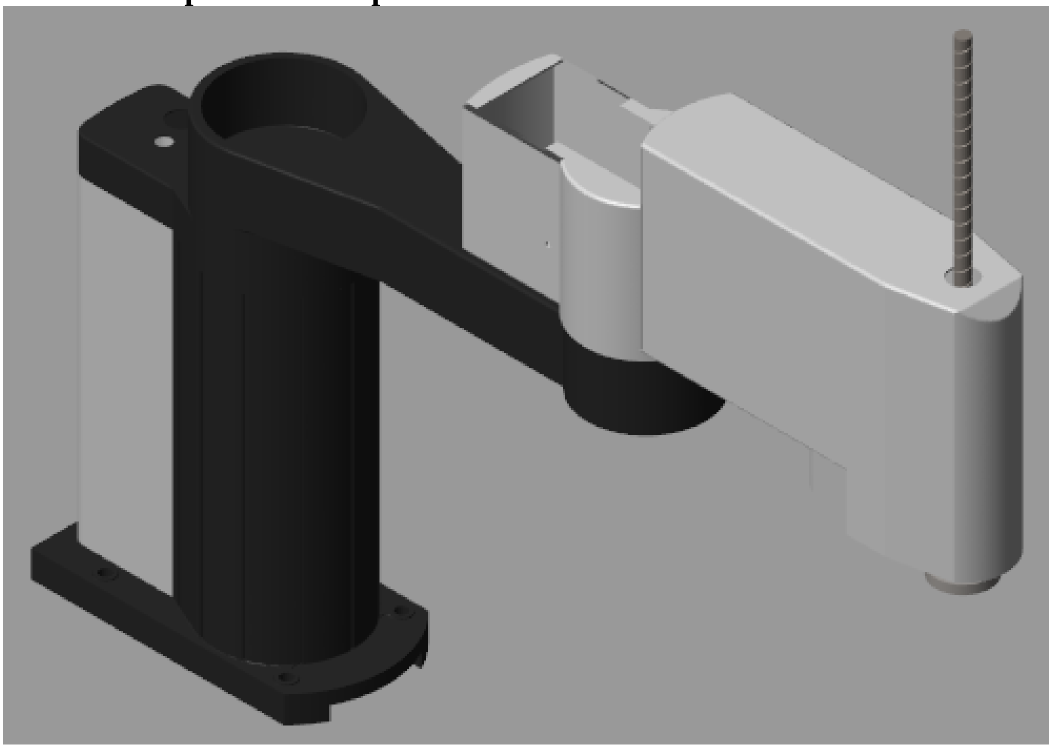

Figure 3.16 The robot model in MATLAB/SIMSCAPE

The SCARA SR8 plus model in MATLAB/SIMULINK:

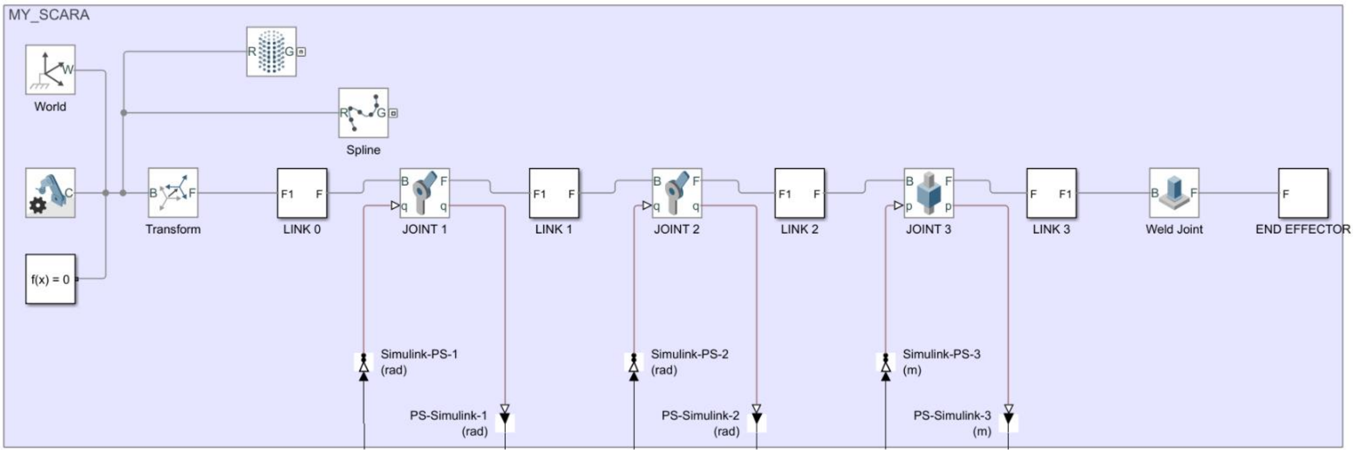

Figure 3.17 Robot blocks in Simulink

Modeling a control system for the whole robot:

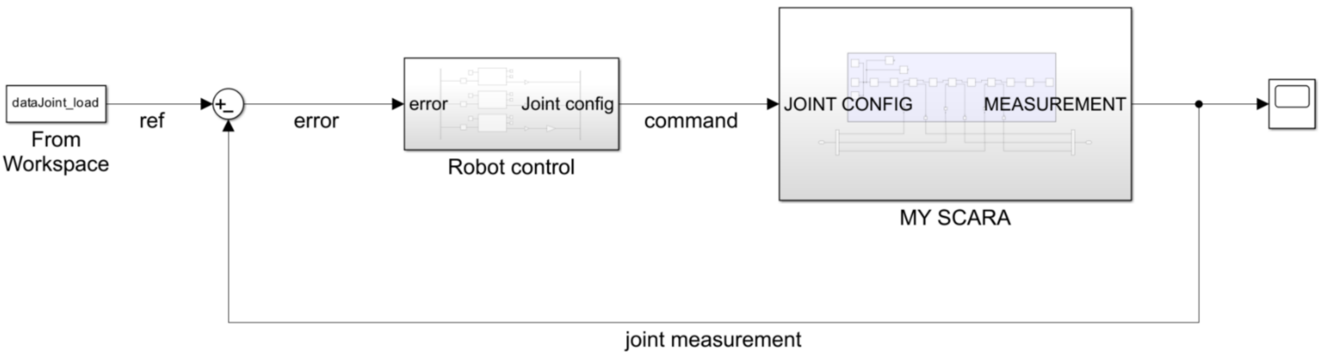

Figure 3.18 Block diagram of the control system of SCARA robot

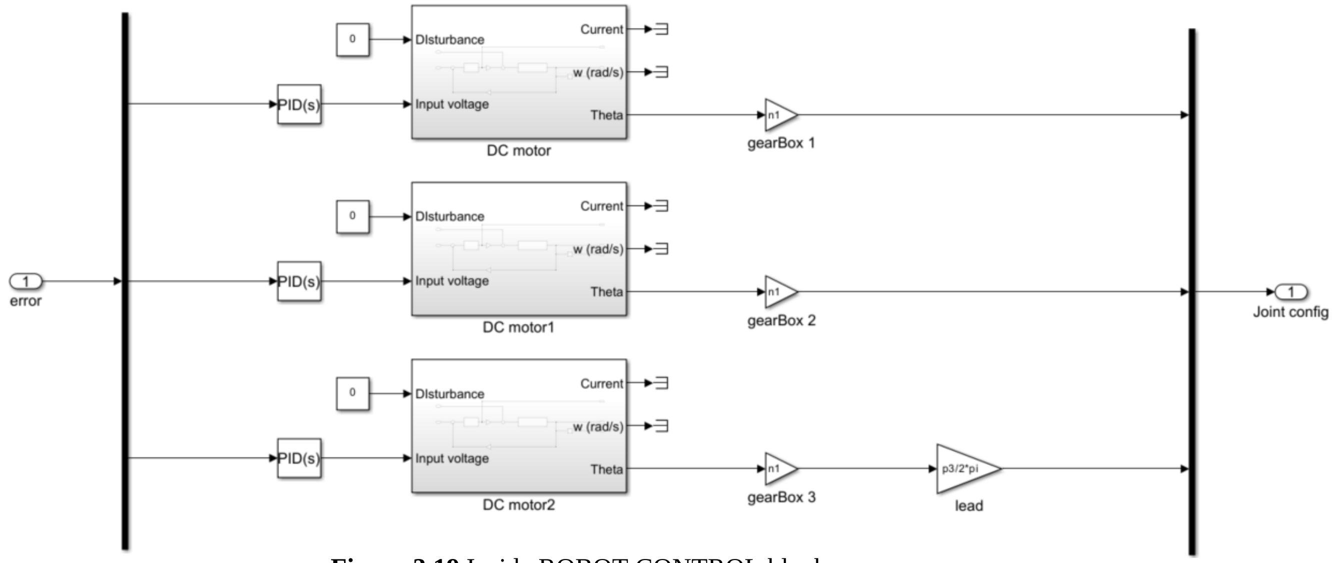

Figure 3.19 Inside ROBOT CONTROL block

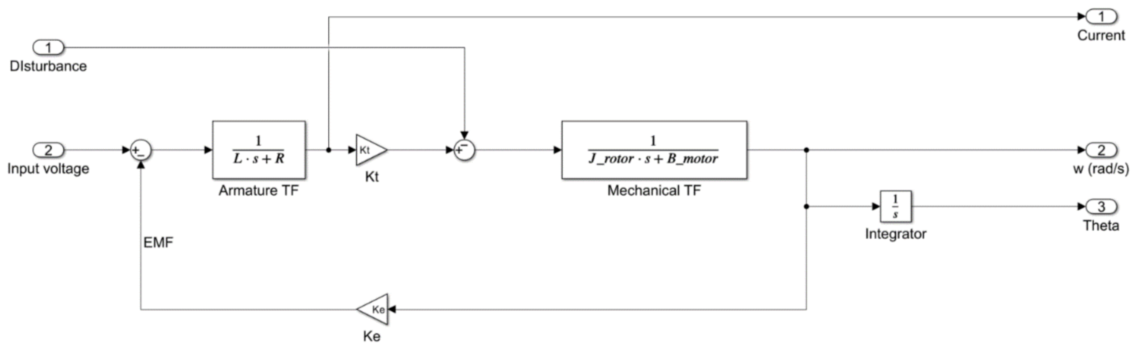

Figure 3.20 DC motor block

The input of the system is the joint variables ($q1$, $q2$, $q3$) depending on the position of the end effector in space. Here the joint variables satisfy the given geometric trajectories which are solved by the inverse kinematics problem as shown in sections 2.1 and 2.2.

The input will pass through the PID Control block, DC motor block and 3D model block of the robot.

The output of the system is the trajectory of the joint variables.

3.3 CONCLUSION OF CHAPTER 3

In chapter 3, we have completed the following tasks:

- Find transfer function of each link of the SCARA SR8 plus robot.
- Test for stability of the above open-loop transfer functions using Nyquist diagram.
- Design a control system for the entire robot.

Chapter 4: SIMULATION AND STABILITY EVALUATION

4.1 USE MATLAB/SIMULINK TO SIMULATE ONE LINK AND THE ENTIRE ROBOT

Simulink diagram of the 3 links:

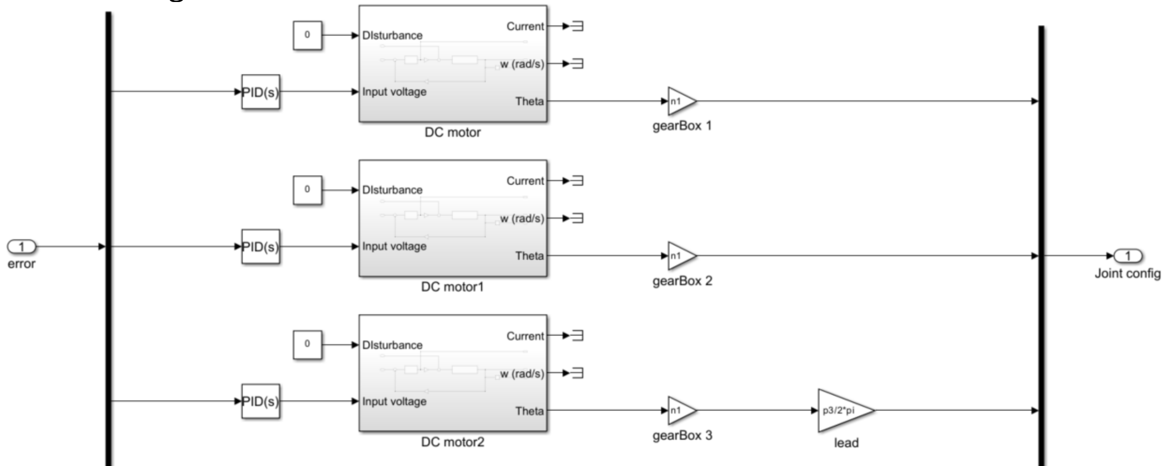

Figure 4.1 Simulink diagram of the 3 links

Determine steady-state error and test for stability of each link:
To determine steady-state error, we analyze the response of the closed-loop system with input being a step function $u(t) = \begin{cases} 1\,; if\ t \geq 0 \\ 0\,; otherwise \end{cases}$

<u>Link 1:</u>
Transfer function of link 1 is: $G_1(s) = \frac{K_{td}}{T_1 s^3 + T_2 s^2 + s} = \frac{1.044}{7.36\times10^{-5}\ s^3 + 0.01977\ s^2 + s}$
The transfer function of the closed-loop system of link 1 is: $H_{1(s)} = \frac{G_1(s)}{1+G_1(s)}$
We can find $H_1(s)$ using the built-in function $feedback(G_1(s), 1)$ in MATLAB. Here is the step response graph.

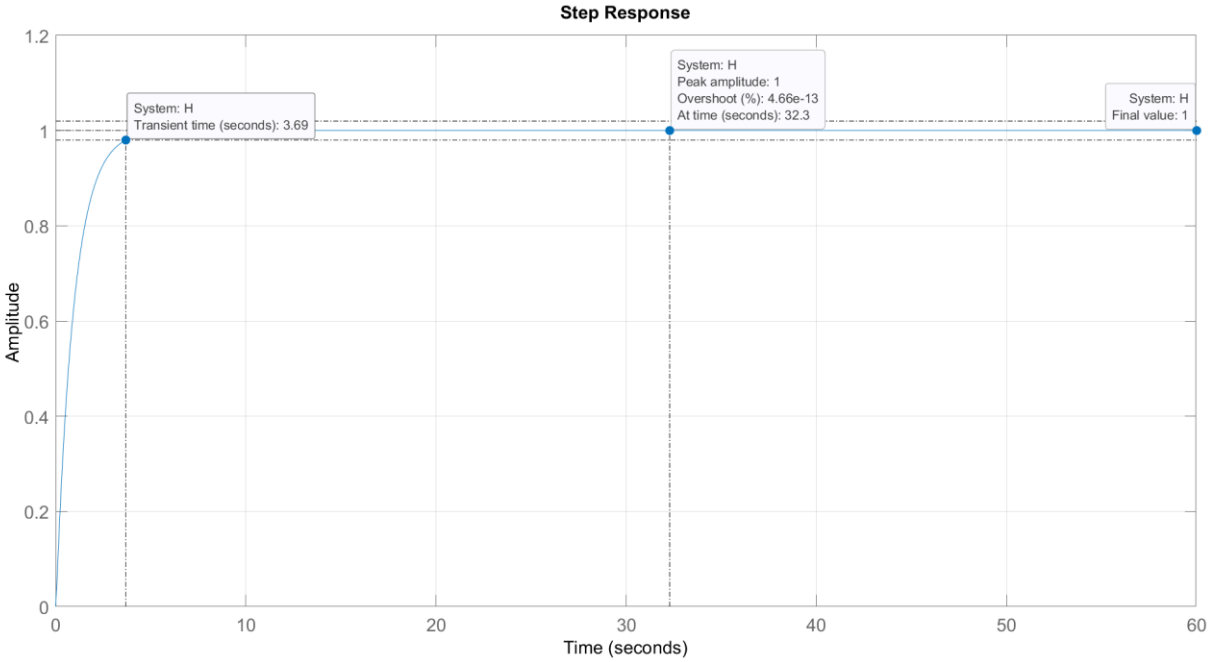

Figure 4.2 Step response of link 1's transfer function

Remark:

- Overshoot: 0%.
- Transient time: 3.69s (the time it takes for the system to reach 2% error in the input or 98% of the reference value).
- Final value: 1.
- Steady state error: 0.

⇒ Link 1's transfer function is stable.

<u>Link 2:</u>

Transfer function of link 2 is: $G_2(s) = \frac{K_{td}}{T_1 s^3 + T_2 s^2 + s} = \frac{1.044}{7.36\times10^{-5}\ s^3 + 0.01977\ s^2 + s}$

The transfer function of the closed-loop system of link 2 is $H_{2(s)} = \frac{G_2(s)}{1+G_2(s)}$

We can find $H_2(s)$ using the built-in function $feedback(G_2(s), 1)$ in MATLAB. Here is the step response graph.

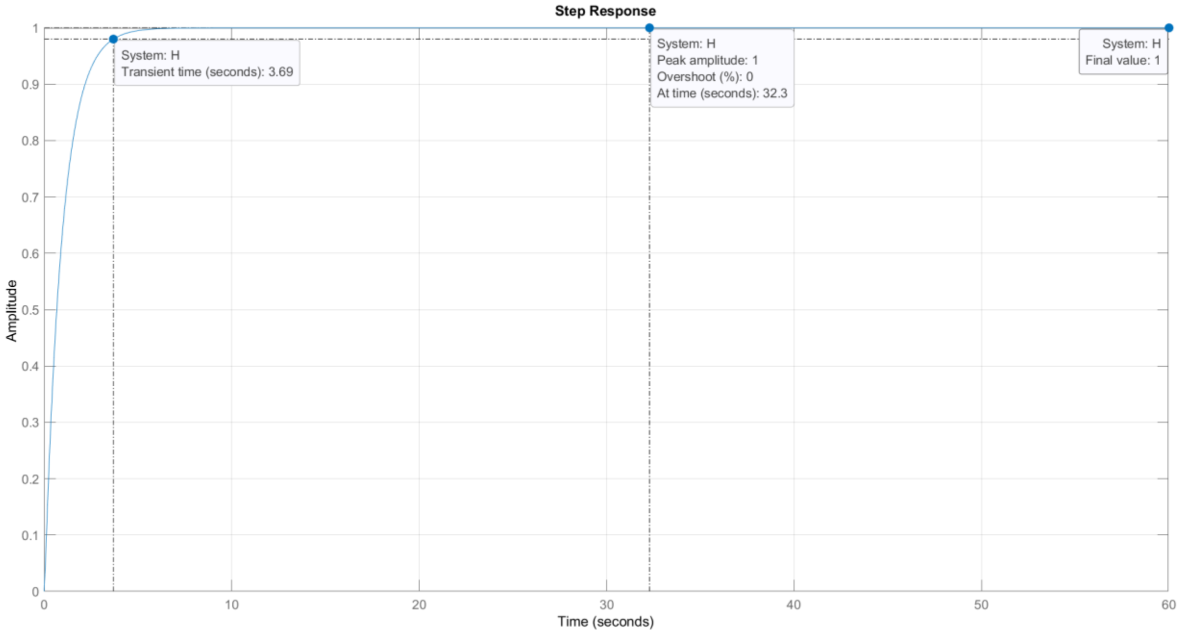

Figure 4.3 Step response of link 2's transfer function

Remark:

- Overshoot: 0%.
- Transient time: 3.69s (the time it takes for the system to reach 2% error in the input or 98% of the reference value).
- Final value: 1.
- Steady state error: 0.

⇒ Link 2's transfer function is stable.

Link 3:

Transfer function of link 3 is: $G_3(s) = \frac{K_{td}\frac{p}{2\pi}}{T_1 s^3 + T_2 s^2 + s} = \frac{0.002152}{0.0002831\ s^3 + 0.07653\ s^2 + s}$

The transfer function of the closed-loop system of link 3 is $H_{3(s)} = \frac{G_3(s)}{1 + G_3(s)}$

We can find $H_3(s)$ using the built-in function $feedback(G_3(s), 1)$ in MATLAB. Here is the step response graph.

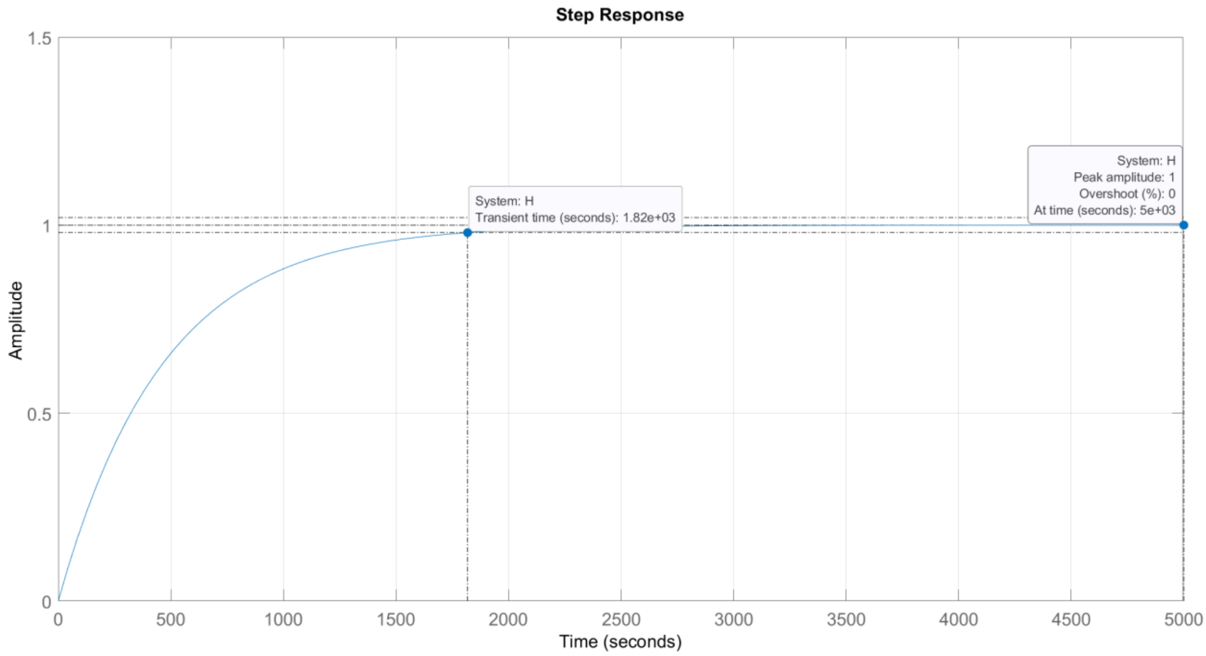

Figure 4.4 Step response of link 3's transfer function

Remark:

- Overshoot: 0%.
- Transient time: 1820s (the time it takes for the system to reach 2% error in the input or 98% of the reference value).
- Final value: 1.
- Steady state error: 0.

⇒ Link 3's transfer function is stable.

4.2 CONCLUSION:

Although the robot can reach a stable state, the transient time is too large, the response is not fast enough. Therefore, it is necessary to design controllers to speed up response, reducing the transient time of the system.

Chapter 5: CONTROL SYSTEM DESIGN

5.1 PID CONTROLLERS DESIGN:

PID Controller (Propotional – Integral – Derivative Controller) – is a control feedback loop mechanism widely used in industry. The PID controller calculates the error value (the difference between the actual measured output signal and the desired signal). The PID will then reduce the error by adjusting the input signal. The smaller the deviation, the closer the system to the desired working state.

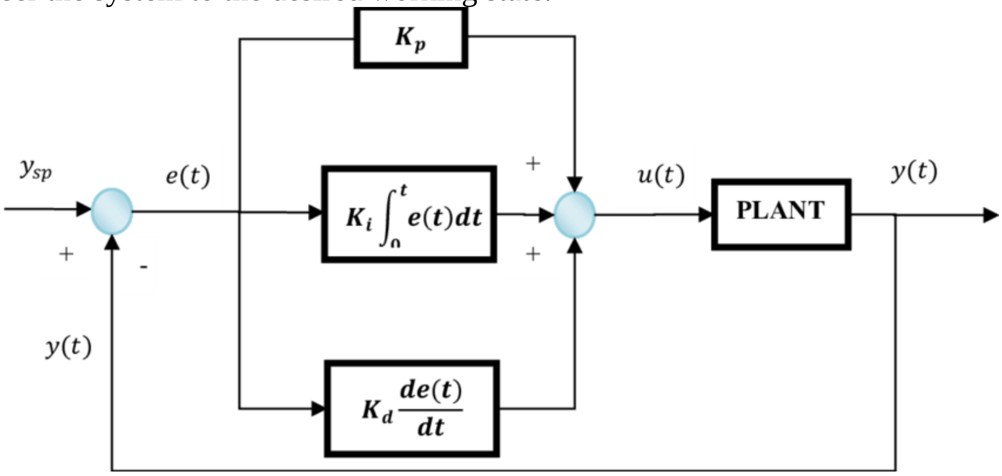

Figure 5.1 PID controller block diagram

In this project, with the goal of controlling the robot to reach any desired position in the workspace (the joint variables must reach the values according to the required trajectory), we will use the PID control law for the robot. The control signal of the PID Controller is the difference in position or in other words, the control signal is the error between the current position and the desired position.

$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt} + K_I \int_0^t e(t)dt$$

where:

$$\begin{cases} e(t) = q_{ref}(t) - q(t) \text{ is the error between measurement value and reference value} \\ q_{ref}(t) \text{ is the reference value} \\ q(t) \text{ is the measurement value} \end{cases}$$

Proceed to design the PID controller in MATLAB as follows:

Setting up the Simulink diagram for the controller

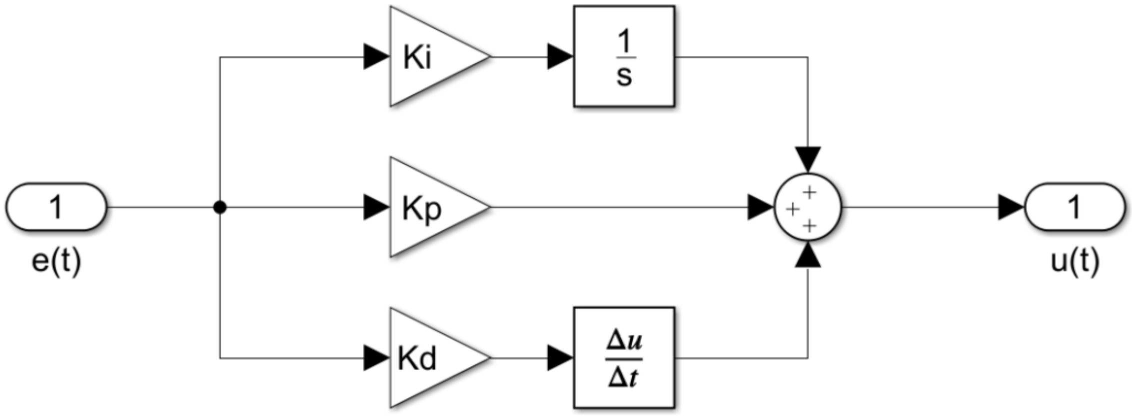

Figure 5.2 PID controller in Simulink

In MATLAB/Simulink there is already a PID Controller block, we will use this block directly without rebuilding.

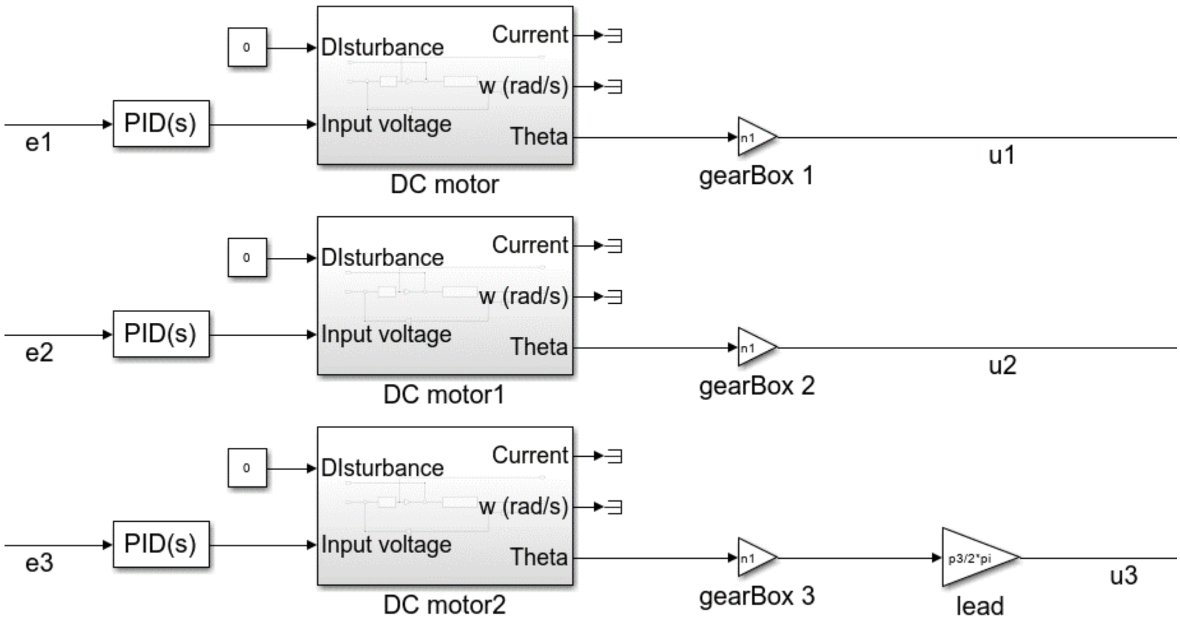

Figure 5.3 PID blocks controlling all three links.

The selection of parameters for the PID Controller can be carried out using several methods. The simplest is manual tuning, trying different sets of parameters until the desired signal is reached. The second method is that we can use mathematical methods such as Ziegler-Nichols, Chien-Hrones-Reswick, magnitude optimization ... And the 3rd method, which is also used in this project, is to use the integrated PID Tuner in MATLAB.

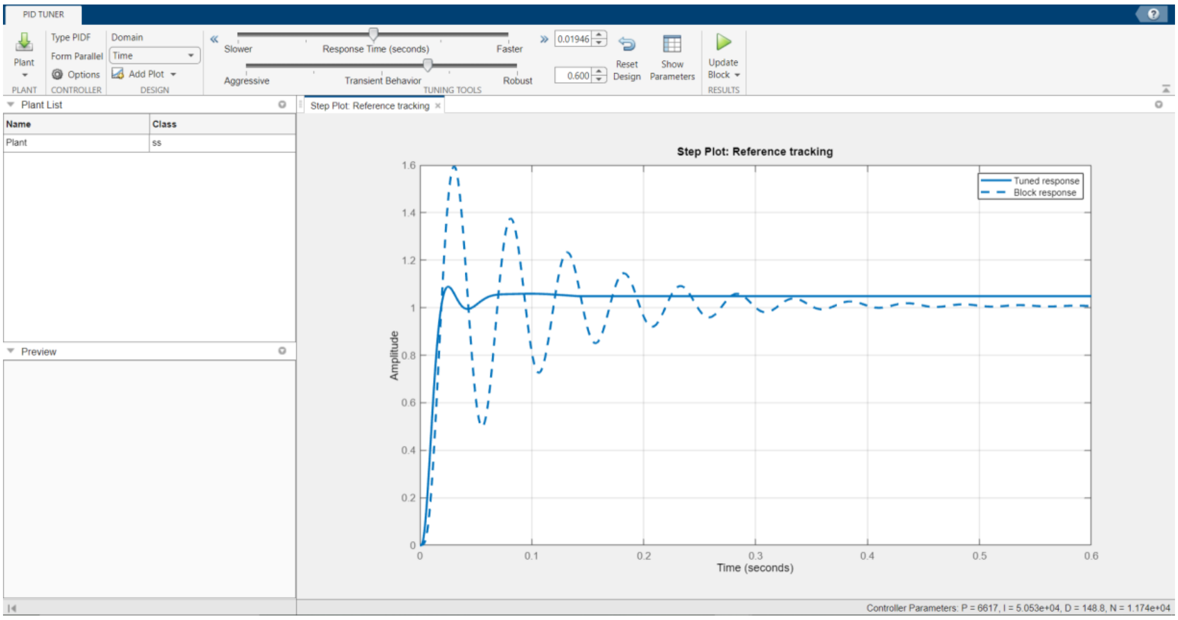

Figure 5.4 User interface of PID tuner

After tuning, we acquire the parameters of our 3 PID controllers as shown in the table below:

Table 5.1 Parameters of the 3 PID Controllers after tuning:

Khâu	K_p	K_I	K_D
1	4509	24314	93
2	6013	43737	143
3	176487	988052	3651

Evaluate the quality of the designed controller with step function input:

<u>Link 1:</u>

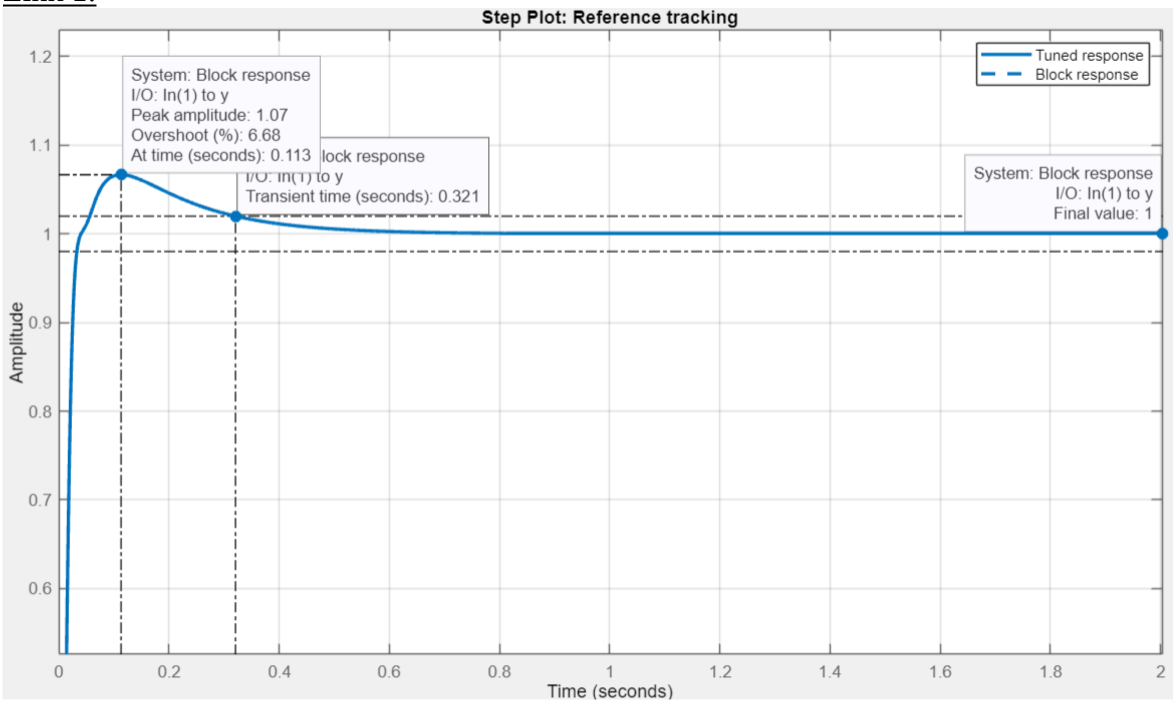

Figure 5.5 Step response of link 1

- Overshoot: 6.68%
- Transient time: 0.321s
- Final value: 1.
- Steady state error: 0.

⇒ Link 1 is stable.

Link 2:

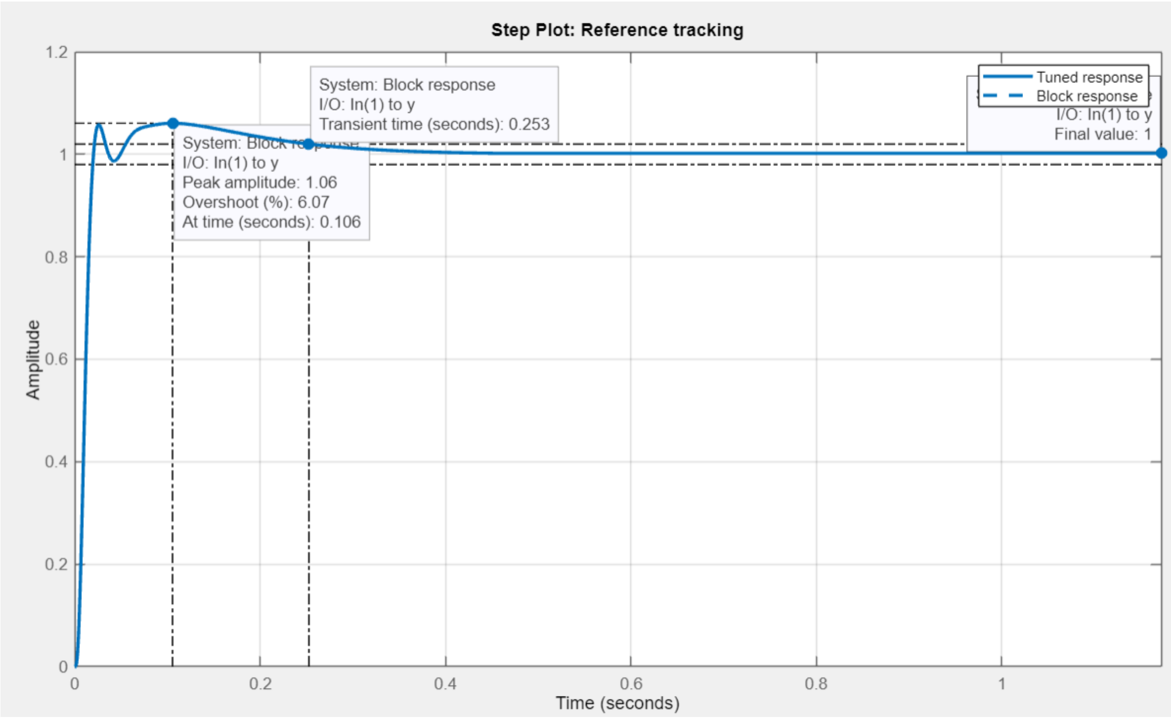

Figure 5.6 Step response of link 2

- Overshoot: 6.07%
- Transient time: 0.253s
- Final value: 1
- Steady state error: 0.

⇒ Link 2 is stable.

Link 3:

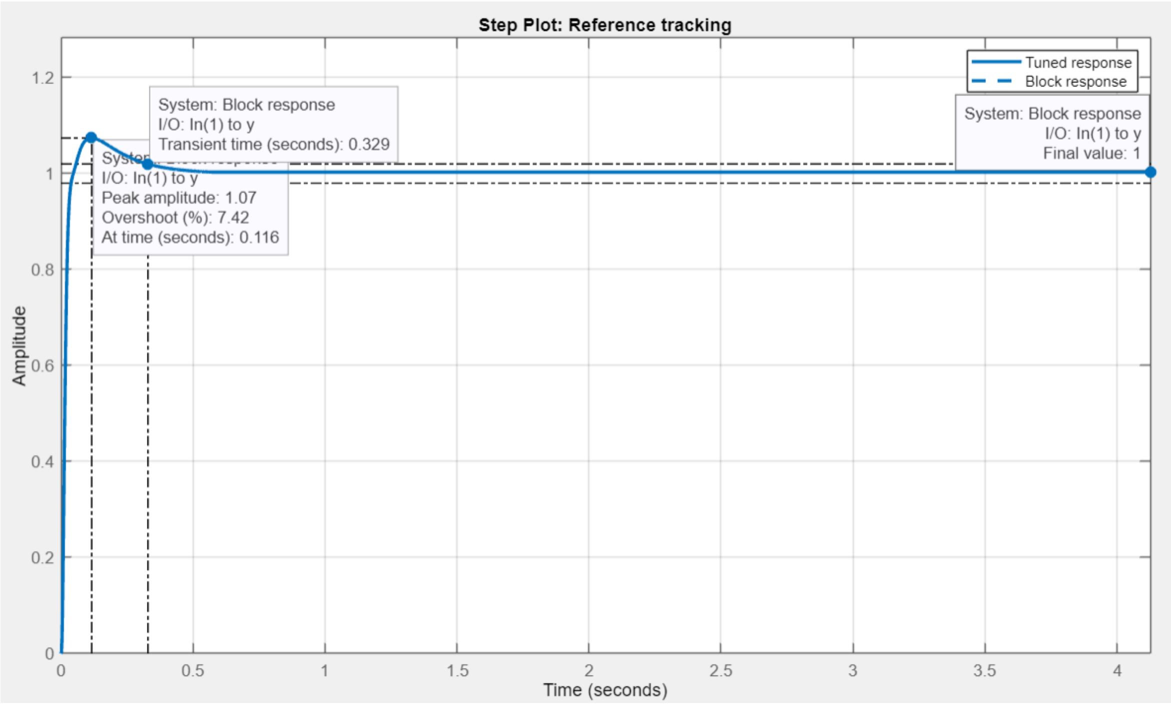

Figure 5.7 Step response of link 3

- Overshoot: 7.42%
- Transient time: 0.329s
- Final value: 1.
- Steady state error: 0.

⇒ Link 3 is stable.

Remark: After designing the PID controller, it was found that the system had a small overshoot, because this is almost inevitable, so it is completely acceptable, above all, the response speed is very fast and the transient time has been reduced extremely significantly, which will make the system more efficient.

After designing the PID Controller for the robot, we obtain a graph of the trajectory of each joint variable compared to the reference signal as follows:

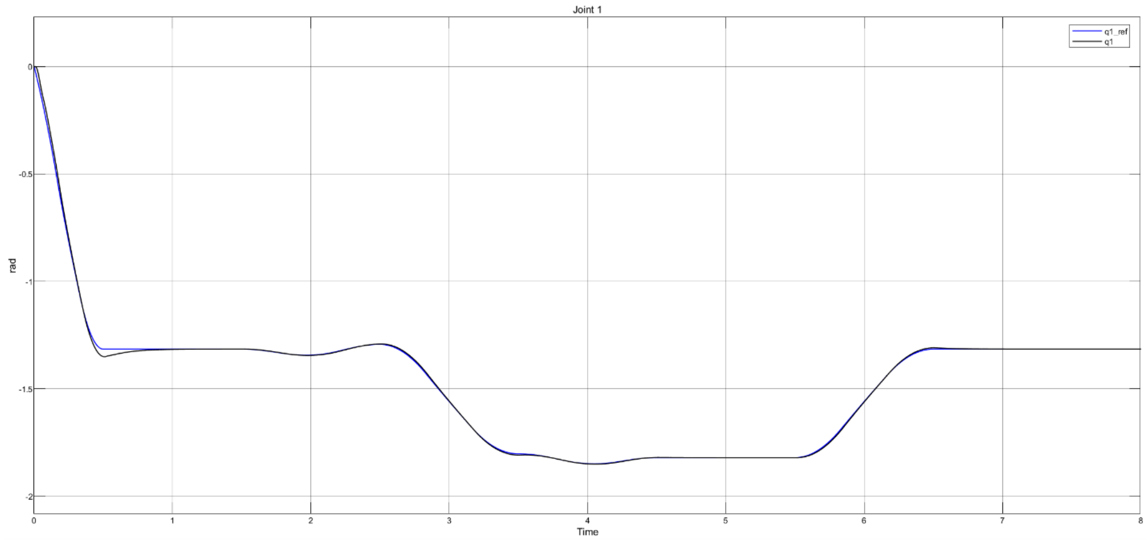

Figure 5.8 Joint 1 measurement vs Joint 1 reference

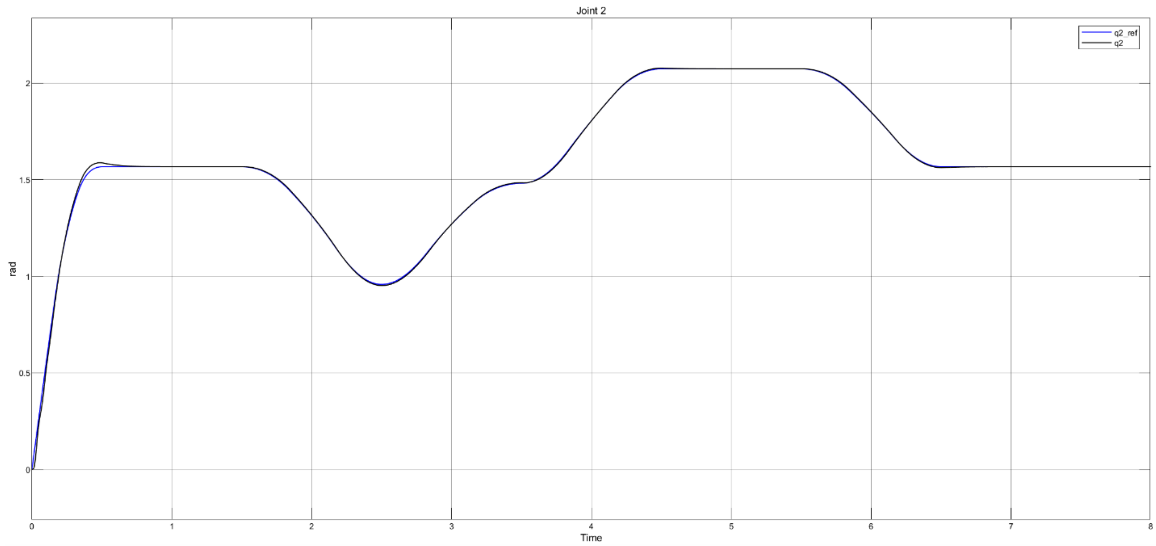

Figure 5.9 Joint 2 measurement vs Joint 2 reference

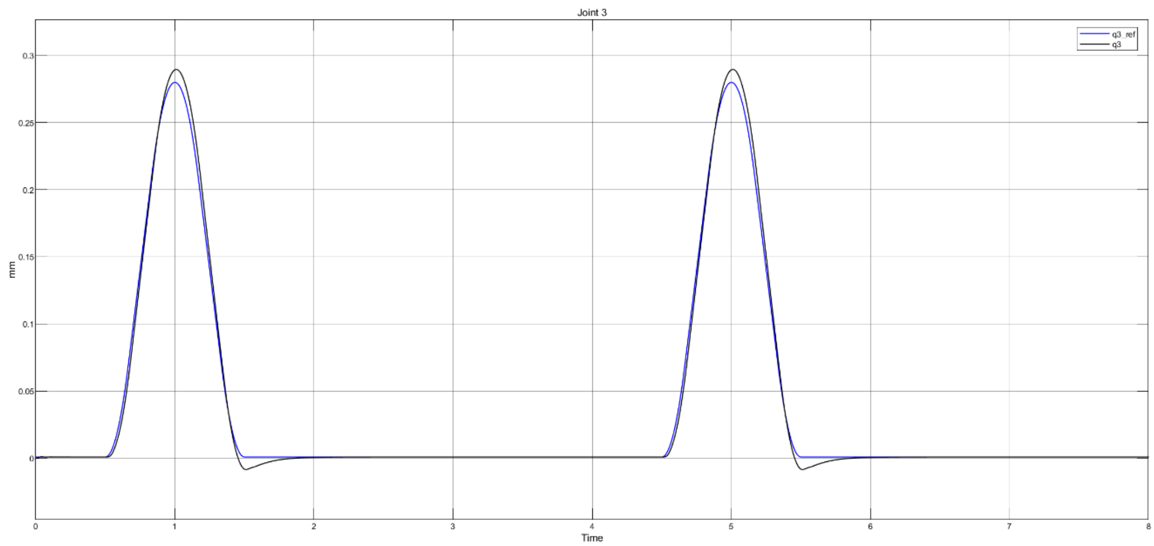

Figure 5.10 Joint 3 measurement vs Joint 3 reference

5.2 SELECTION OF COMPONENTS FOR THE CONTROL SYSTEM:

The robot control system consists of 2 basic and important elements: Servo motor control system and robot controller.

Servo motor control system:

The servo control device assembly consists of 2 components: driver and servo motor. With servo motor drivers, there are usually 2 options in industry. Firstly, we will use the inverter to control the motor. In the past, inverters were often used to control AC motors for high-power applications such as air compressors or heavy conveyors, but the development of inverter technology has allowed to expand the range of applications, so it is possible to use inverters to control servo. Second, we will use the Servo Driver. Servo drivers are modular, meaning that each driver will be compatible with certain types of servo motors, specified by manufacturers. This type is suitable for small power applications, requiring large torque and high-precision position control, applications in robots, CNC machines ...

	Inverter	Servo Driver
Control application	Use for applications that do not require high speed and accuracy, stability	Used for high-speed and high-precision control applications, fast and continuous state changes
Control mode	Basic speed control	Control position, speed, and torque
Ability to control multiple motors	One inverter can control multiple motors	A basic servo driver controls only a single servo motor
Responsiveness	Slow 100 rad/s or less	Fast About 200 - 15000 rad/s
Highly precise stop and position control	Up to about 100 μm	Up to about 1μm
Lock mode	no	yes
Start/stop frequency	20 rpm or less	20 – 600 rpm
Maximum torque	about 150%	about 300%
Power	100W – 300kW	10W – 60kW
Dimension	Big, heavy	More compact and lightweight
Investment cost	Relatively low	High

From the above characteristics, we will choose Servo Driver to control servo for SCARA robot system. In this project, I will choose SGDV-550A driver and SGMGV-75A servo motor from Yaskawa.

Figure 5.11 SGMGV servo motor series from Yaskawa

Figure 5.12 SGDV servo driver series from Yaskawa

Servomotor Model: SGMGV-□□□		03A	05A	09A	13A	20A	30A	44A	55A	75A	1AA	1EA
Rated Output[*1]	kW	0.3	0.45	0.85	1.3	1.8	2.9	4.4	5.5	7.5	11	15
Rated Torque[*1]	N·m	1.96	2.86	5.39	8.34	11.5	18.6	28.4	35.0	48.0	70.0	95.4
Instantaneous Peak Torque[*1]	N·m	5.88	8.92	13.8	23.3	28.7	45.1	71.1	87.6	119	175	224
Rated Current[*1]	A_{rms}	2.8	3.8	6.9	10.7	16.7	23.8	32.8	42.1	54.7	58.6	78
Instantaneous Max. Current[*1]	A_{rms}	8	11	17	28	42	56	84	110	130	140	170
Rated Speed[*1]	min^{-1}	1500										
Max. Speed[*1]	min^{-1}	3000									2000	
Torque Constant	$N·m/A_{rms}$	0.776	0.854	0.859	0.891	0.748	0.848	0.934	0.871	0.957	1.32	1.37
Rotor Moment of Inertia	$\times 10^{-4}$ kg·m^2	2.48 (2.73)	3.33 (3.58)	13.9 (16)	19.9 (22)	26 (28.1)	46 (54.5)	67.5 (76.0)	89.0 (97.5)	125 (134)	242 (261)	303 (341)
Rated Power Rate[*1]	kW/s	15.5 (14.1)	24.6 (22.8)	20.9 (18.2)	35.0 (31.6)	50.9 (47.1)	75.2 (63.5)	119 (106)	138 (126)	184 (172)	202 (188)	300 (283)
Rated Angular Acceleration[*1]	rad/s^2	7900 (7180)	8590 (7990)	3880 (3370)	4190 (3790)	4420 (4090)	4040 (3410)	4210 (3740)	3930 (3590)	3840 (3580)	2890 (2680)	3150 (2960)
Applicable SERVOPACK	SGDV-□□□□	3R8A	3R8A	7R6A	120A	180A	330A 200A[*2]	330A	470A	550A	590A	780A

Figure 5.13 Specification of SGMGV servo motor

SERVOPACK Model SGDV-□□□□		R70A	R90A	1R6A	2R8A	3R8A	5R5A	7R6A	120A	180A	200A	330A	470A	550A	590A	780A
Applicable Servomotor Max. Capacity	kW	0.05	0.1	0.2	0.4	0.5	0.75	1.0	1.5	2.0	3.0	5.0	6	7.5	11	15
Continuous Output Current	A_{rms}	0.66	0.91	1.6	2.8	3.8	5.5	7.6	11.6	18.5	19.6	32.9	46.9	54.7	58.6	78
Max. Output Current	A_{rms}	2.1	2.9	6.5	9.3	11	16.9	17	28	42	56	84	110	130	140	170
Main Circuit		Three-phase 200 to 230 VAC+10% to −15% 50/60 Hz														
Control Circuit		Single-phase 200 to 230 VAC+10% to −15% 50/60 Hz														

Figure 5.14 Specification of SGDV servo driver

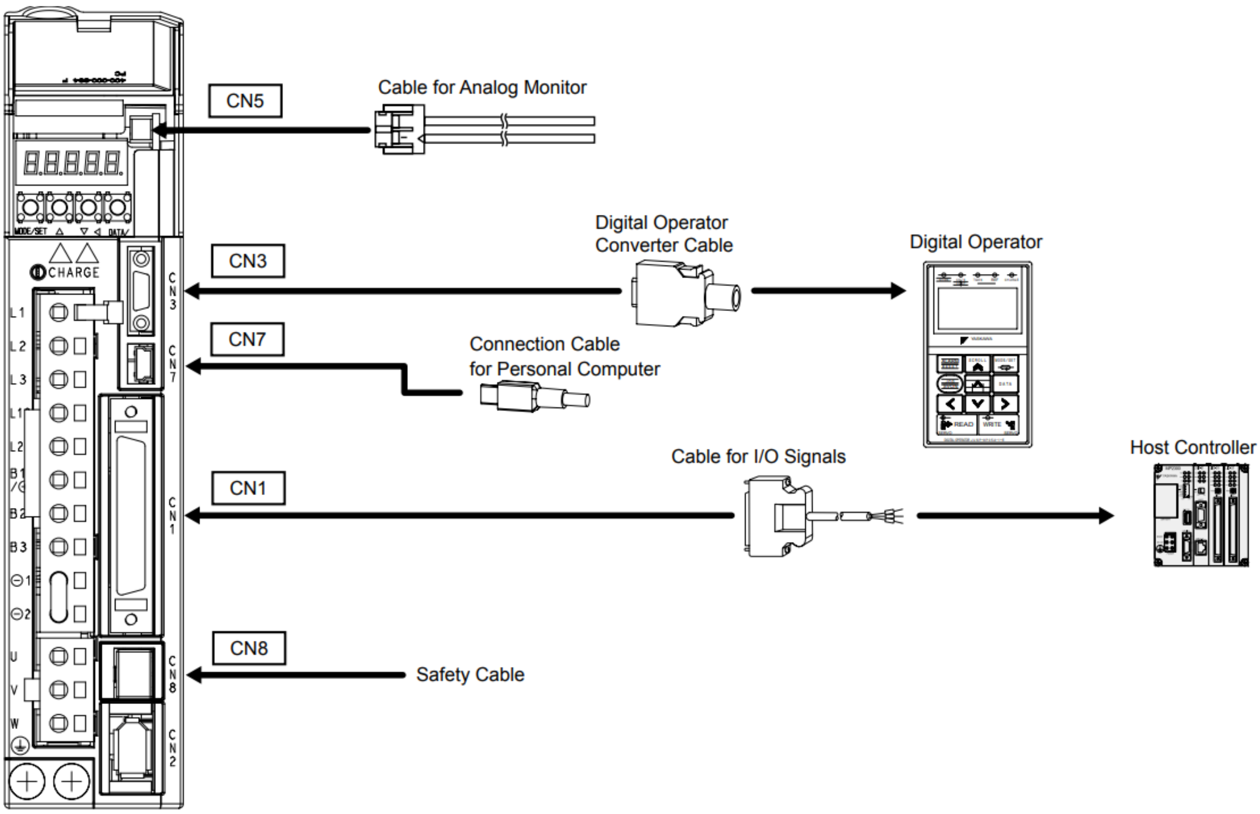

Figure 5.15 Basic connection of SGDV driver

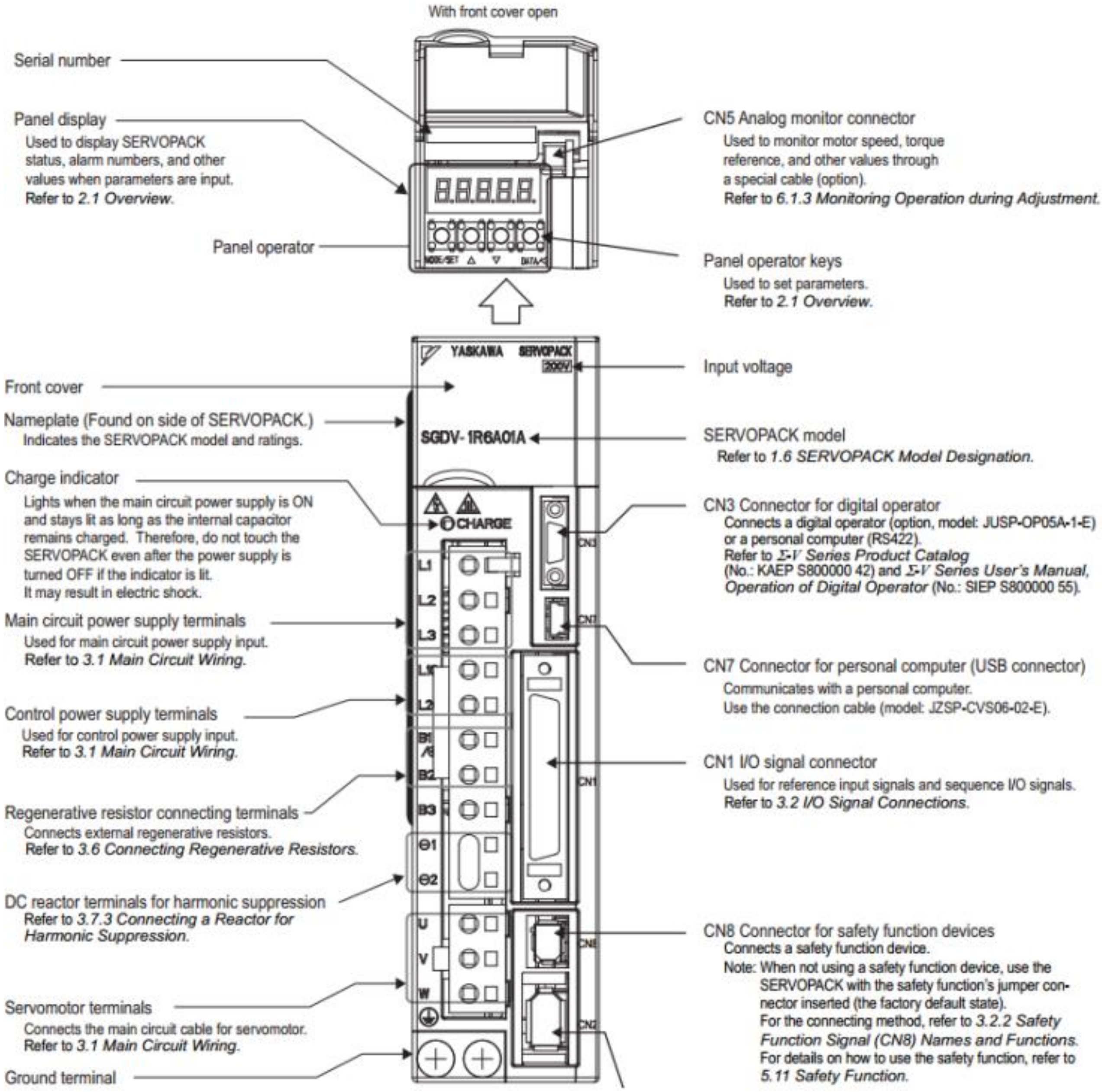

Figure 5.16 Interface of the SGDV driver

The driver is divided into 3 main areas:

- Panel Operator:
    - Includes 1 LED display and buttons, used to configure functions, set the mode for the driver to operate.
- The power supply section (located on the left flank), consists of:
    - L1, L2, L3 ports: Power supply to the motor.
    - L1C, L2C ports: Power supply for the driver to operate.
    - Port U, V, W: Port connected to motor.
- The input/output port (located on the right flank), includes:
    - CN5 port: Annalog display output port.
    - CN3 port: Gateway to control devices.
    - CN7 port: USB connector for computers.
    - CN1 port: I/O port in and out, receiving control signals from the central controller, outputting warning signals, …
    - CN8 port: Functional Gate for Safety Devices.
    - CN2 port: Connect to the motor encoder.

A driver is a closed-loop feedback controller in a modular form, such as a PID control unit in the design of the control system. The control signals will be received from control ports such as CN3, CN7 or CN1. Depending on the application design requirements, users can choose different central controllers to provide control signals to the driver.

If controlled via CN1 port, we can have an additional set of cables to output the pin for specific purposes. For example, connecting to microcontrollers, devices that need to use multiple I/O pins.

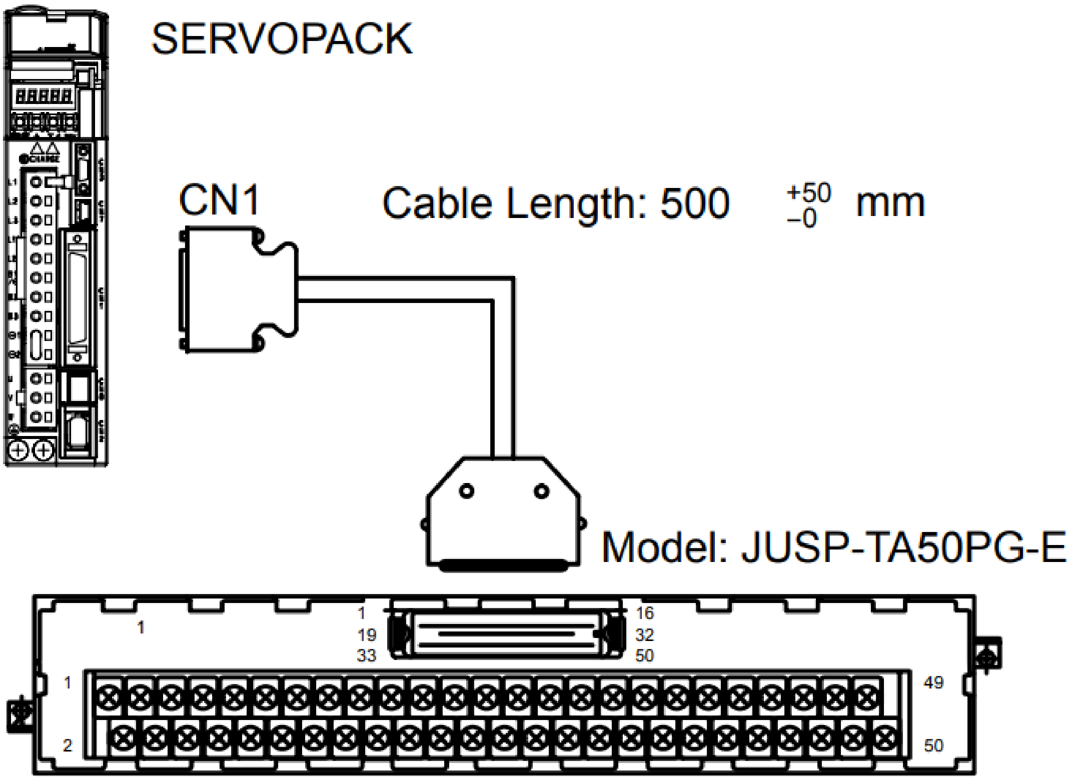

Figure 5.17 Connecting with CN1 port.

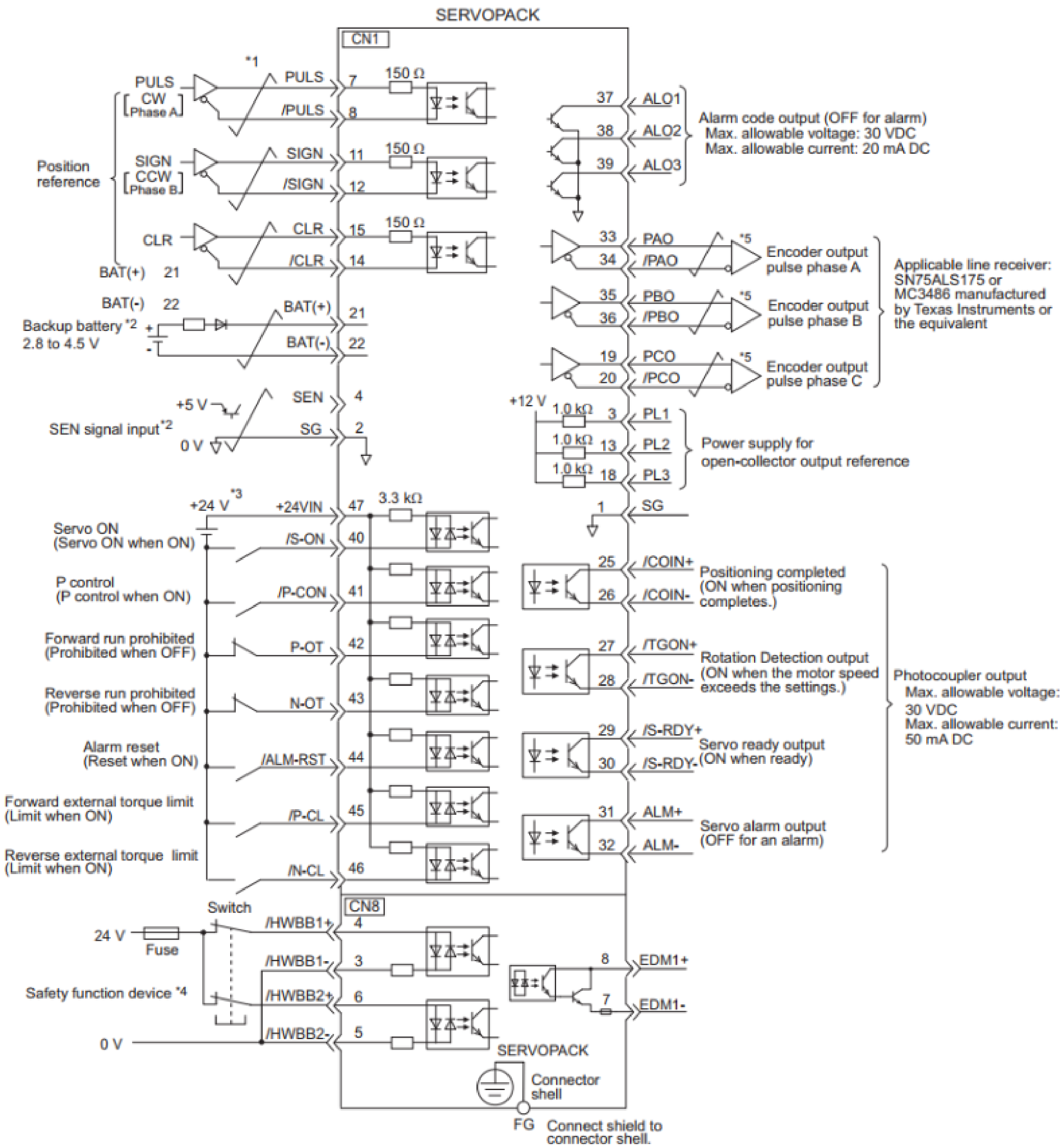

Figure 5.18 IO map of CN1 port.

The position of the motor is controlled by 2 input pins, PULS (pins 7 and 8) and SIGN (pins 11 and 12), where PULS is the pulse supply pin to the driver, each pulse supplied to the motor will move the motor corresponding to the pulse of the encoder mounted on the motor. The mode settings for Driver, here select Factory Settings mode:

Parameter		Reference Pulse Form	Input Pulse Multiplier	Forward Run Reference	Reverse Run Reference
Pn200	n.□□□0 [Factory setting]	Sign + pulse train (Positive logic)	–	PULS (CN1-7) SIGN (CN1-11) H level	PULS (CN1-7) SIGN (CN1-11) L level
	n.□□□1	CW + CCW pulse train (Positive logic)	–	CW (CN1-7) L level CCW (CN1-11)	CW (CN1-7) CCW (CN1-11) L level
	n.□□□2	Two-phase pulse train with 90° phase differential	×1	90° Phase A (CN1-7) Phase B (CN1-11)	90° Phase A (CN1-7) Phase B (CN1-11)
	n.□□□3		×2		
	n.□□□4		×4		
	n.□□□5	Sign + pulse train (Negative logic)	–	PULS (CN1-7) SIGN (CN1-11) L level	PULS (CN1-7) SIGN (CN1-11) H level
	n.□□□6	CW + CCW pulse train (Negative logic)	–	CW (CN1-7) H level CCW (CN1-11)	CW (CN1-7) CCW (CN1-11) H level

Figure 5.19 Instruction on how to provide pulses to PULS and SIGN in position control mode.

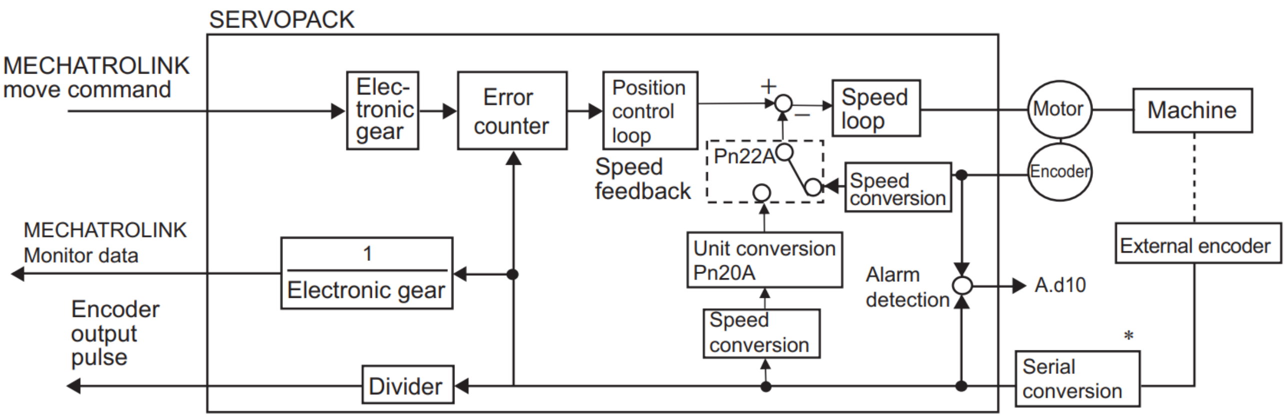

Figure 5.20 Postion control block diagram of SGDV servo driver

In summary, from the 20bit encoder → the number of pulses / revolutions of the motor, through the ratio of the gear reducer, we can calculate the number of pulses for the motor to rotate 1 revolution after the gear reducer is:

$$n = 2^{20} \times k$$

where k is the gear reducer ratio.

For example, when we want to control the motor to spin one revolution, we need to feed $n \times a$ pulses to the PULS pin. To reverse, change the logic level in the SIGN pin.

In this project, the objective of the control system design is that the robot needs to meet the position requirements, thus using the robot's position control mode. A minimum of 3 PULS pins and 3 SIGNS pins will be required to control the 3 sets of drivers. In addition, a few additional INPUT ports will be needed to allow the servo to operate, warning of errors...

Robot controller:

When choosing a controller for an automatic mechanism, as summarized in Chapter 1, we usually think of 2 options: microcontroller or PLC.

Based on my understanding of microcontrollers, along with accessibility, open resources, large support community, flexibility in programming, I would like to propose the choice of microcontroller ESP-32 with board ESP32 DEVKIT V1 from NodeMCU using the microcontroller ESP32-WROOM-32 from Espressif.

Figure 5.21 ESP32 DEVKIT V1

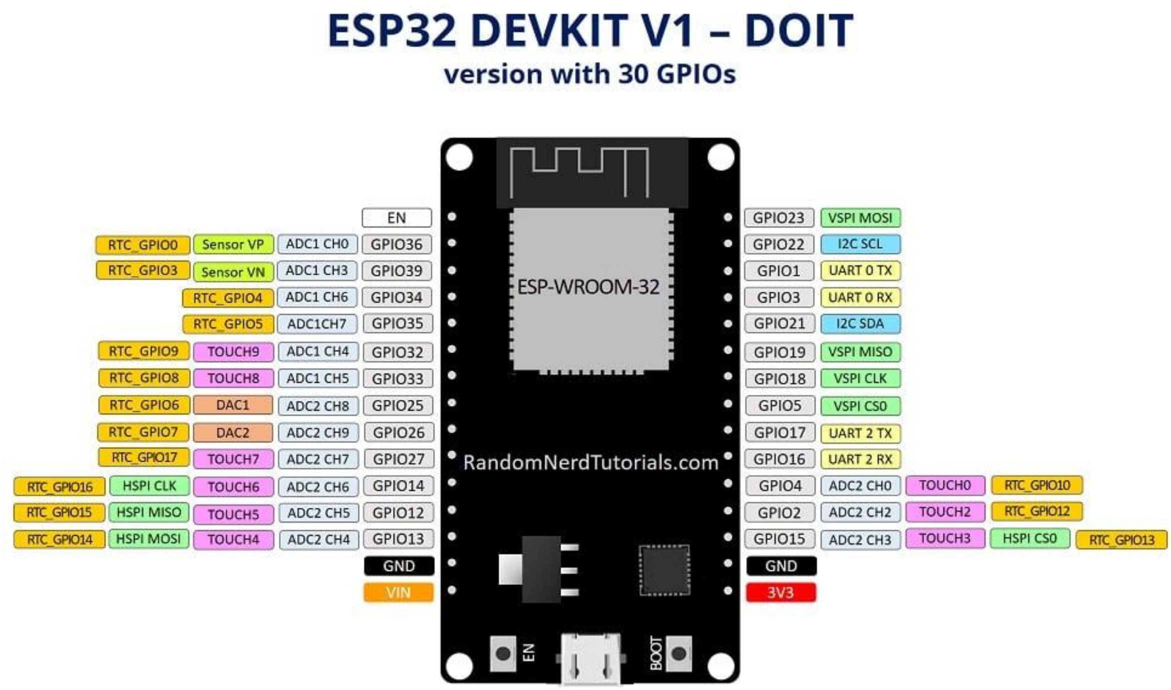

Figure 5.22 ESP32 DEVKIT V1 pin map

Technical specifications of ESP32-WROOM-32:

– Dual-core Xtensa LX7 CPU, up to 240 MHz, and supporting single-precision floating point processing unit.
– 512 KiB SRAM, 384 KiB ROM, and 16 KiB RTC SRAM
– Capable of connecting to external PSRAM and Flash via Quad SPI or Octal SPI and share the same 32 MiB address space.
– Ultra-low power RISC-V (RV32IMC) coprocessor clocked at 17.5 MHz approximately.
– Ultra-low power Finite state machine coprocessor.
– Wi-Fi 2.4 GHz (IEEE 802.11 b/g/n)
– Bluetooth 5 (LE)
– 45 programmable GPIOs
– 2 × 12-bit SAR ADCs, up to 20 channels
– 30 x GPIO can be attached to external hardware interrupts.
– 3 x UART, 2 x I2C, 2 x I2S, 4 x SPI

5.3 ELECTRICAL CIRCUITS AND COMPONENTS WIRING:

Power supply circuit:

The system uses 3 drivers to control 3 servo motors, the voltage supplied to the motor operates using 3-phase AC 220V power supply. The power source for the driver is a 1-phase AC 220V power supply. This power can be obtained from 2 wires of a 3-phase electrical system. The power supply for the controller is a 24VDC external power.

The power supply system includes such main equipment as:

- 1QF: Circuit breaker
- 1FLT: Noise filter
- 1KM: Magnetic contactor for power supply
- 2KM: Magnetic contactor for main circuit
- 1Ry: Relay
- 1PL: Indicator lamp
- 1SA, 2SA, 3SA: Surge absorber
- 1D: Flywheel diode
- Other contactors

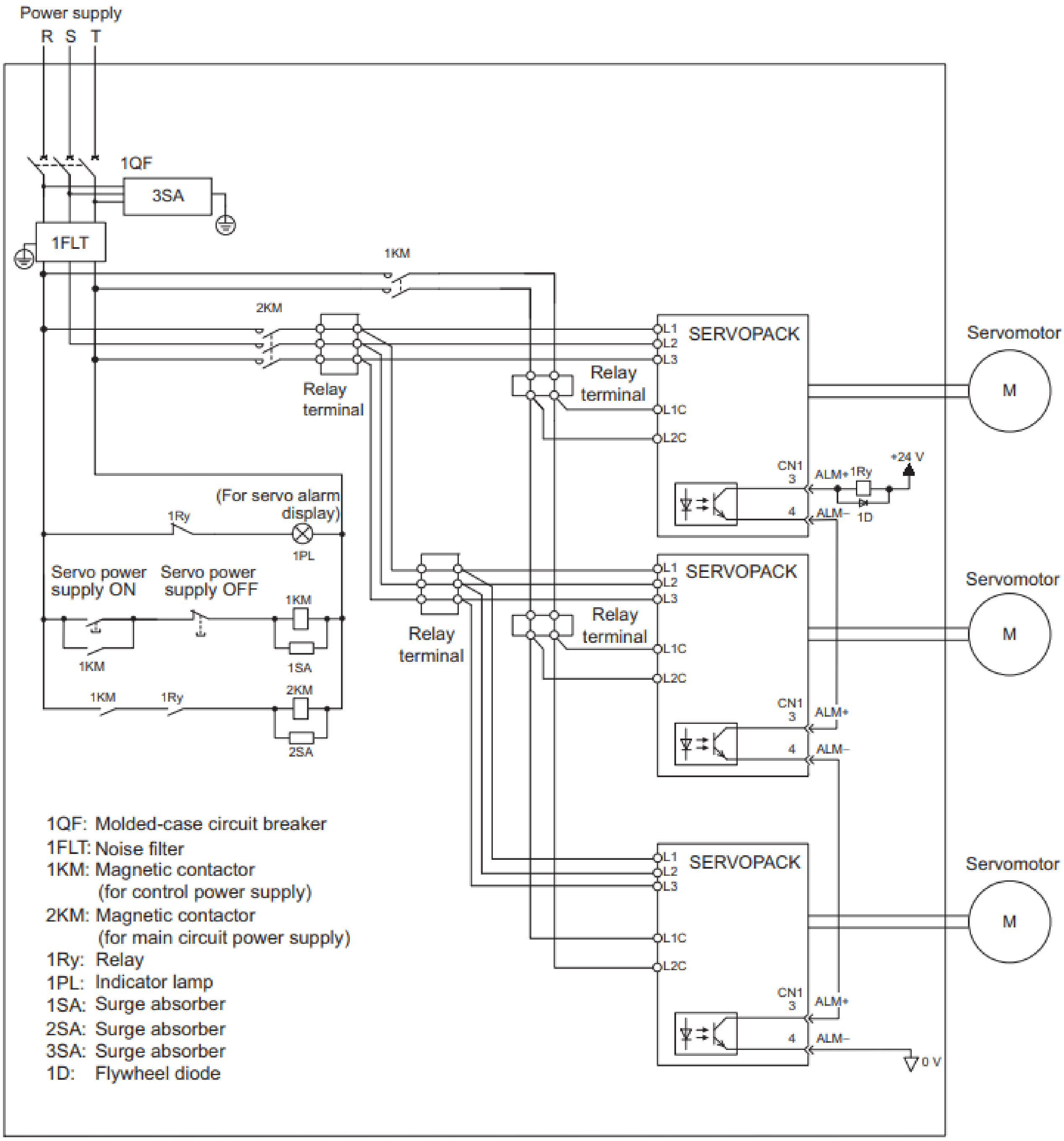

Figure 5.23 Diagram of connecting 3 motors and drivers to 3-phase VAC power supply.

The 3-phase power supply is taken from the system and fed into the motor power port consisting of pins L1, L2, L3 corresponding to the 3 phase wires of the power supply. On the line there are Relay Terminals to connect the line extension to the drivers. The motor is connected to the U, V, W ports on the driver and encoder signal wire. L1C, L2C ports take 1-phase power from 1 phase wire and 1 neutral wire to power the driver.

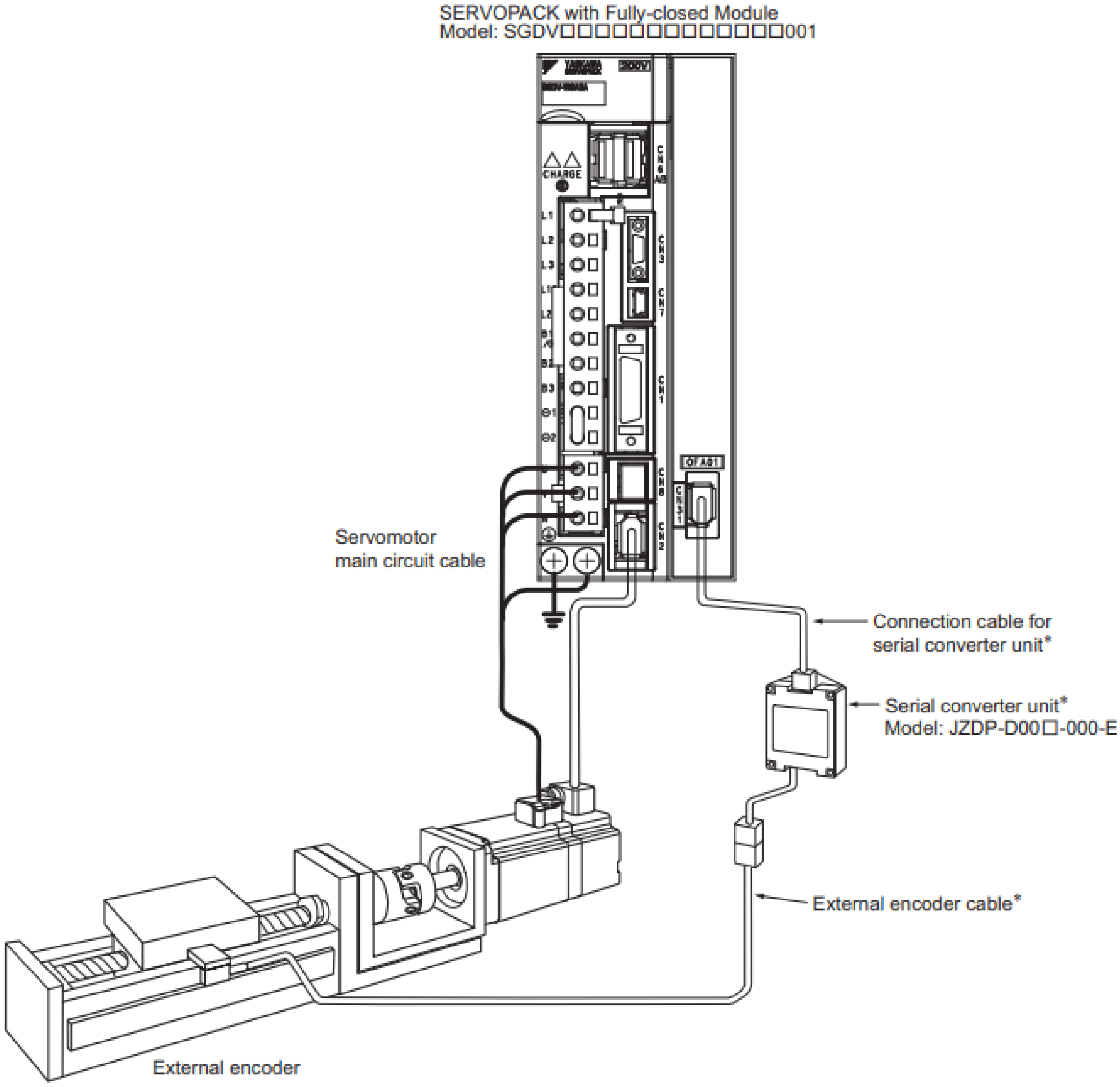

Figure 5.24 Diagram of connecting a motor with a driver.

Components wiring

Referring to Figure 5.18, to control position, we use PULS and SIGN pins to control the position and direction of rotation of the motor. There are also INPUT pins with other functions such as:

- S-ON: Driver enable.
- P-CON: Positon Control mode.
- P-OT: Limiting the rotation of the motor in the forward direction.
- N-OT: Limiting the direction of rotation of the motor in the opposite direction.

OUTPUT pins:

- ALO1, ALO2, ALO3: Driver error bits
- COIN: Arrived at the desired position.
- TGON: Exceed the set velocity.
- SRDY: Servo ready.

Thus, for a motor, a minimum of 10 I/O ports is required to control and monitor the driver's operation. 3 motors will need 30 ports. And the I/O pins on the selected control circuit are enough to control the system.

In addition, since the logic level of the servo driver is 24V and the logic level of the microcontroller selected above is 3.3V, the microcontroller needs to communicate with the driver via Opto couplers, which are intended to isolate the signal and to shift the voltage level. The Opto coupler featured in this project is AL-ZARD's DST-1R4P 4-Bit with a maximum signal switching frequency of 20KHz from 3.3V to 24V.

Figure 5.25 4-bit opto coupler from 3.3V to 24V.

5.3 GENERATE CODE FOR MICROCONTROLLERS FROM MATLAB SIMULINK:

After designing a controller for a system, we will have to program that controller in C/C++ or Micro Python for the microcontroller to control the system in real life. In this project, I will use the Embedded Coder app of MATLAB/SIMULINK to automatically generate C program for the ESP-32 microcontroller.

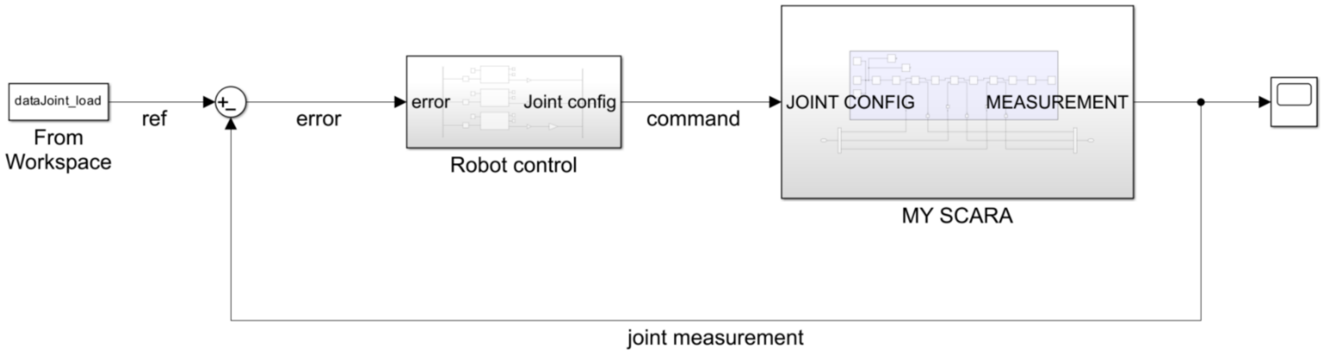

Figure 5.26 Control system for SCARA

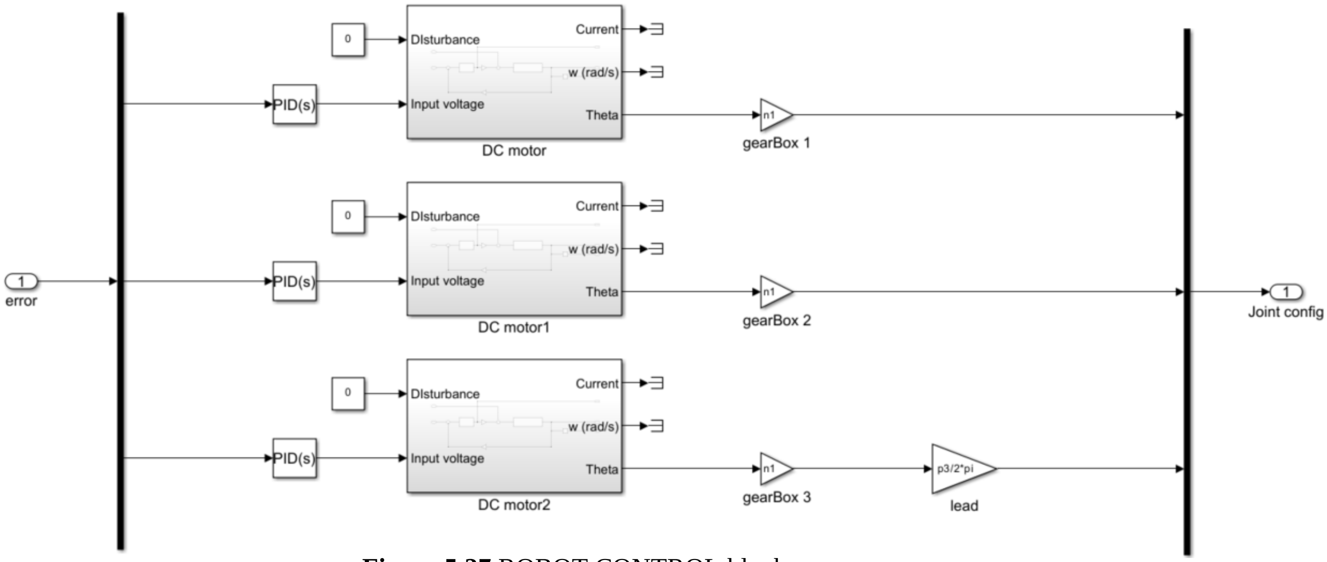

Figure 5.27 ROBOT CONTROL block

We observe that in this control model, there are only 3 PIDs in the ROBOT CONTROL block that need to be installed into the microcontroller, because the ROBOT and the motor are both real-life systems.

The steps are as follows:

– We create a Subsystem containing these 3 PIDs and copy to another Simulink file.

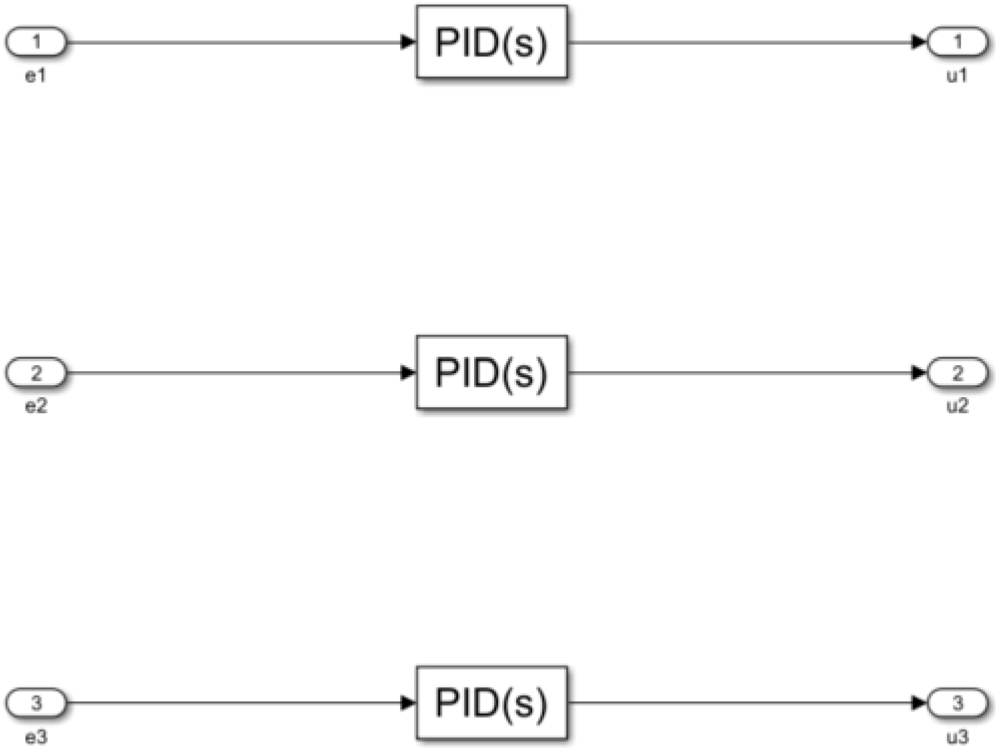

Figure 5.28 Subsytem contains 3 tuned PID blocks.

– Select the app Embedded coder in Simulink's app tab. Simulink will then create a new tab named "C code". Click on "C code" and go to Settings. Declare the following parameters to generate the C language program for the ESP32 microcontroller:

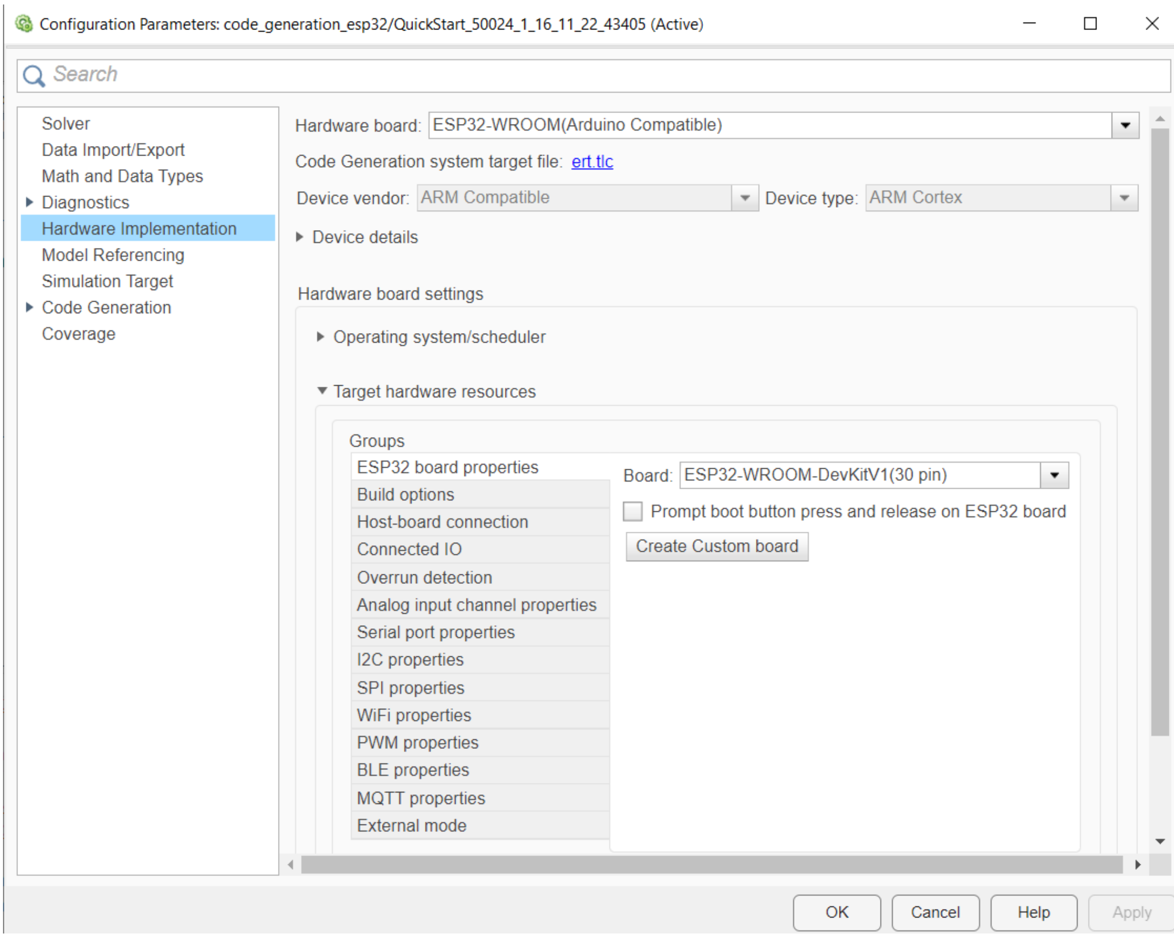

Figure 5.29 Configuration for code generation

– Click "Quick start" to enter the Embedded Coder interface:

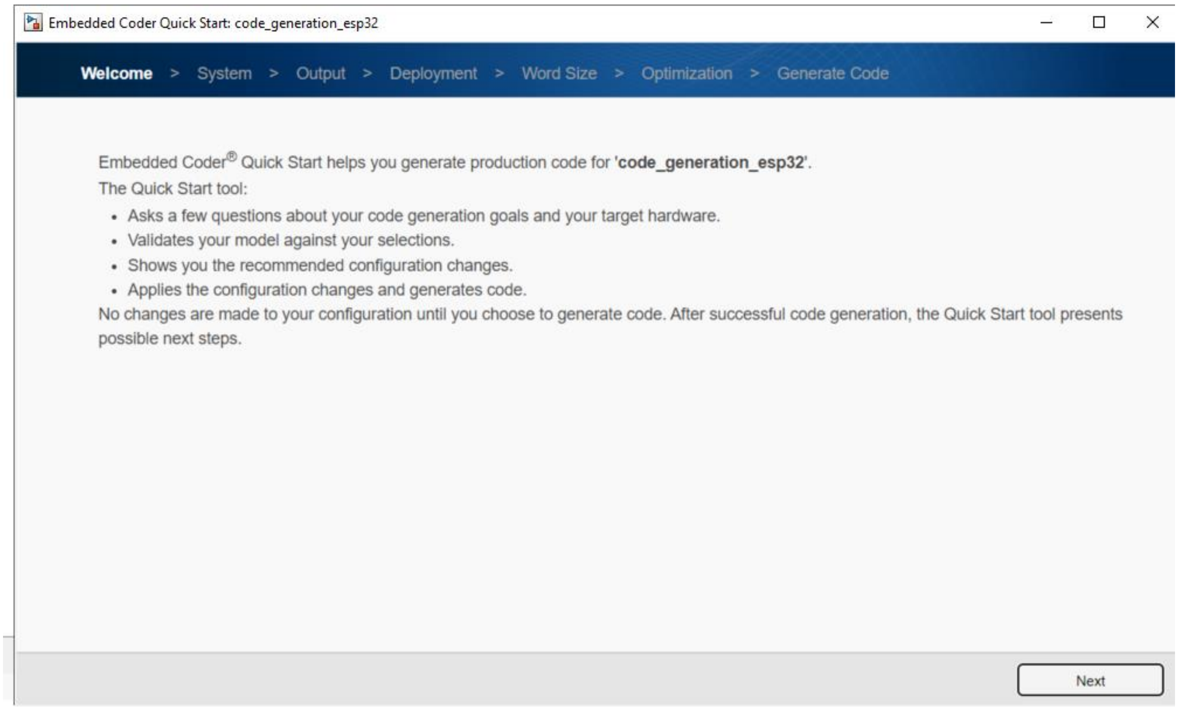

Figure 5.30 User interface of Embedded Coder

- Click Next and select the default parameters, SIMULINK will automatically generate program C for the ESP32 microcontroller into a folder located in the current path. You can use the Arduino IDE to load the program in this folder into the ESP32 microcontroller.

CONCLUSION

Above is my entire course project report on the topic "**Design control system for 3-degree of freedom SCARA ROBOT** ". I took it seriously to follow the assigned task. With the knowledge I have been taught about robotics, I have completed the following tasks:

- Modeling of electric motors and transfer functions of links.
- Build a control system for the entire robot.
- Using MATLAB-Simulink to simulate the control system.
- Stability assessment, identification of steady-state error from the transfer functions of each link.
- Building PID controllers for the robot.
- Selection of control components and components wiring.
- Building a GUI that simulates forward kinematics, inverse kinematics.

Through the process of research, calculation, and design, I have completed the content of the report according to the objectives of the project. However, in the process of implementation, despite the dedicated guidance of **Assoc. Prof. Pham Duc An**, due to limitations in terms of knowledge, experience as well as time, I will not be able to avoid shortcomings. I look forward to receiving your comments and assessments to make the report more complete. Through this, I will draw valuable knowledge and experience for other modules, graduation projects and future work problems.

Once again, I would like to sincerely thank **Assoc. Prof. Pham Duc An** for your helpful guidance and suggestions.

REFERENCES

1. Rexroth "*Turboscara SR4/6/8plus Mechanical manual*"
2. YASKAWA - AC Servo Drives $\sum -V\ Series$ USER'S MANUAL Design and Maintenance Rotational Motor Analog Voltage AC Servo Drives $\sum -V\ Series$and Pulse Train Reference
3. YASKAWA - AC Servo Drives $\sum -V\ Series$ Product Catalog
4. Matlab tutorial, website: *http://www.mathworks.com/help/matlab/code-to-run-the-gui.html*