**HCM UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FALCULTY OF HIGH-QUALITY TRAINING**

**-----------------⸙∆⸙-----------------**

****

**PROJECT ON ROBOTICS**

**TOPIC: DESIGN AND CONTROL A ROBOT 3DOF PICKS UP OBJECTS**

|  |  |  |
| --- | --- | --- |
| **Instructor** | **:** | **Nguyen Tran Minh Nguyet** |
| **Student’s name** | **:** | **Nguyen Dinh Hoan – 20151277** |
|  |  | **Le Thanh Nam - 17142122** |

**Ho Chi Minh City, 6/2024**

**COMMENTS OF TEACHER INSTRUCTION**

Project name: DESIGN AND CONTROL A ROBOT 3DOF PICKS UP OBJECTS

Student’s name:

Nguyen Dinh Hoan 20151277

Le Thanh Nam 17142122

I. CONTENTS THAT NEED ADJUSTMENT AND SUPPLEMENTATION:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

II. COMMENT:  
1. Regarding the presentation form & the rationality of the essay structure:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………  
2. About content (evaluate essay quality, pro/cons and practical value):

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....

III. RECOMMENDATION AND REVIEW:

Rating (on a 10-point scale): ……………………………………………….

Ho Chi Minh City, June 22, 2024

**INSTRUCTOR**

*(Sign & write full name)*

Contents

[CHAPTER 1: OVERVIEW 5](#_Toc169963071)

[1.1 Introduction to the project 5](#_Toc169963072)

[1.2 Contents will be conducted in this project 6](#_Toc169963073)

[1.3 Limitations of this research 6](#_Toc169963074)

[1.4 The applicable ability of this project 6](#_Toc169963075)

[CHAPTER 2: BACKGROUND 8](#_Toc169963076)

[2.1 Paper Idea 8](#_Toc169963077)

[2.2 The block diagram 9](#_Toc169963078)

[2.3 Basic components of the system 10](#_Toc169963079)

[2.3.1 Adapter power 10](#_Toc169963080)

[2.3.2 Programmable Logic Controller 1214C DC/DC/DC 10](#_Toc169963081)

[2.3.3 Conveyors 11](#_Toc169963082)

[2.3.4 DC motor for belt conveyor. 13](#_Toc169963083)

[2.3.5 Sensors: 13](#_Toc169963084)

[2.3.6 Step Motor 42 15](#_Toc169963085)

[2.3.7 Dirver TB6600 15](#_Toc169963086)

[2.3.8 Relay 16](#_Toc169963087)

[2.3.9 Robot frame 17](#_Toc169963088)

[2.4 Working principle 19](#_Toc169963089)

[2.4.1. Forward Kinematics 19](#_Toc169963090)

[2.4.2. Inverse Kinematics 20](#_Toc169963091)

[Chapter 3: Hardware Connection 23](#_Toc169963092)

[3.1. Flow chart 23](#_Toc169963093)

[3.2 Hardware Design 24](#_Toc169963094)

[3.3. Connection Diagram 27](#_Toc169963095)

[Chapter 4: Matlab Code and Simulation 30](#_Toc169963096)

[4.1. Matlab Simulink 30](#_Toc169963097)

[CHAPTER 5: CONCLUSION AND FUTURE ORIENTATION 32](#_Toc169963098)

[5.1 Conclusion 32](#_Toc169963099)

[5.2 Future orientation 32](#_Toc169963100)

[APPENDIX 33](#_Toc169963101)

[REFERENCES 37](#_Toc169963102)

# CHAPTER 1: OVERVIEW

## 1.1 Introduction to the project

In the context of modern industry, the automation of production and logistics processes has become a key factor, playing an important role in improving efficiency, accuracy, and cost savings. Among various automation systems, robotic arms are highly regarded for their versatility and reliability. This study focuses on the design and control of a 3 Degrees of Freedom (3DOF) robotic arm tasked with picking up objects from a conveyor belt—a common requirement in assembly, packaging, and transportation lines.

A 3DOF robotic arm has the ability to mimic the flexibility and dexterity of a human arm, allowing it to perform various tasks in production. The design of this robot must ensure a balance between complexity and operational efficiency, suitable for applications with space and budget constraints. The design process includes planning the kinematic structure, selecting appropriate actuators, and integrating sensors to ensure precise and repeatable movements.

Additionally, the control strategy is a crucial factor in ensuring the robotic arm operates effectively. This involves developing motion control algorithms, enabling the robot to identify, grasp, and move objects from the conveyor belt to specified positions. Advanced control techniques such as inverse kinematics, trajectory planning, and feedback systems are applied to achieve smooth and accurate movements, accommodating the constantly changing nature of the conveyor belt.

This project aims to provide a comprehensive overview of the design and control of a 3DOF robotic arm, including mechanical design, material and component selection, kinematic analysis to determine the range of motion, and the implementation of control systems to manage the robot's operations. Additionally, the project addresses challenges such as real-time object recognition and adaptive control strategies to handle objects of varying sizes and positions on the conveyor belt.

The successful implementation of this robotic system promises to significantly enhance productivity in manufacturing environments, reduce manual labor, increase operational speed, and improve the overall quality of the handling process. Through this study, we hope to contribute valuable insights and practical solutions to the field of industrial automation. ****

## 1.2 Contents will be conducted in this project

**Design and control a robot 3DOF picks up object on conveyor belt**

* Chapter 1: Overview.
* Chapter 2: Background.
* Chapter 3: System Design
* Chapter 4: Simulation, experiments and results
* Chapter 5: Conclusion

## 1.3 Limitations of this research

While this research on the design and control of a 3 Degrees of Freedom (3DOF) robotic arm provides valuable insights into automating production processes, it also faces several limitations. Recognizing these constraints is essential for a realistic evaluation of the system's practical applications. The primary limitations of this research are as follows:

* Limited Practical Application.
* The current system can only lift objects like screws and is unable to detect product defects.
* There is no screen to display notifications or system errors to the users.

## 1.4 The applicable ability of this project

The project contributes significantly to the factory's production line by addressing the need for item sorting. It encompasses the following objectives:

* Automating the production process. Specifically: Designing and controlling a 3 Degrees of Freedom (3DOF) robot to pick up items on the conveyor belt.
* Minimizing the number of workers on the production line.
* Enhancing error detection capabilities on the production line.
* Effectively managing the production capacity of the production line.
* Reducing labor costs.

# CHAPTER 2: BACKGROUND

## 2.1 Paper Idea

In recent years, the field of robotics has witnessed remarkable advancements driven by innovations in mechanical design, sensor technology, and control algorithms. Robotic arms, in particular, have emerged as indispensable tools in various industries, revolutionizing processes in manufacturing, logistics, healthcare, and research. The ability of robotic arms to perform complex tasks with precision and efficiency has led to their widespread adoption and continuous development.

Figure 1: Model ABB IRB 660

The design and control of robotic arms represent crucial aspects of their functionality and performance. A robotic arm's mechanical structure must strike a balance between flexibility, strength, and payload capacity to accommodate a diverse range of applications. Simultaneously, advancements in actuation systems, such as motors and gears, have enabled robotic arms to achieve greater dexterity and speed while maintaining reliability and energy efficiency.

Sensor integration plays a pivotal role in enhancing the capabilities of robotic arms by enabling them to perceive and interact with their environment. Technologies such as vision systems, force sensors, and proximity sensors provide valuable feedback to the control system, allowing for adaptive and responsive behavior. These sensors facilitate tasks such as object detection, localization, and manipulation, expanding the range of applications for robotic arms.

Control strategies are fundamental to orchestrating the movements of robotic arms with precision and efficiency. Kinematic and dynamic control algorithms govern the motion of each joint, ensuring smooth and coordinated movements to achieve desired tasks. Additionally, trajectory planning algorithms optimize the path of the robotic arm to minimize time and energy consumption while avoiding obstacles and collisions. Feedback control loops continuously adjust the arm's movements based on sensor feedback, enabling real-time adaptation to changing conditions.

The aim of this paper is to explore the design and control of a 4-degree-of-freedom (4DOF) robotic arm, focusing on its application in object manipulation tasks. By examining recent advancements in mechanical design, sensor integration, and control strategies, this paper aims to contribute to the ongoing efforts in advancing robotic technology. Through experimental validation and analysis, insights into the performance and capabilities of the robotic arm will be gained, paving the way for future research and applications in automation and robotics.

## 2.2 The block diagram

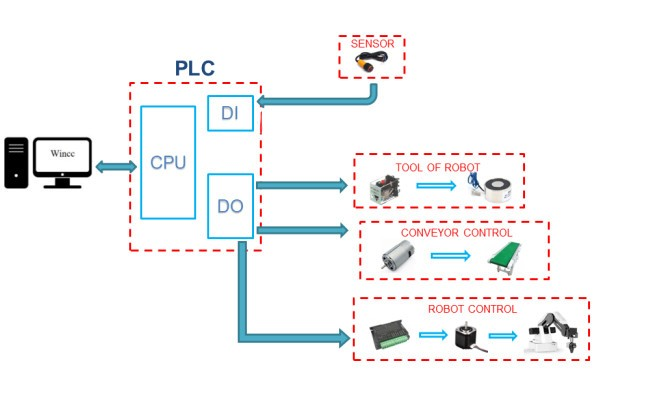
****

Figure 2: Block Diagram

## 2.3 Basic components of the system

**2.3.1 Adapter power**



Figure 3: Power supply EPS-5 24VDC 5A

Specifications:

* Power supply LITEON EPS-5: 24VDC 5A
* Input: AC 220V/ 50Hz.
* Output: DC 24V-2500mA

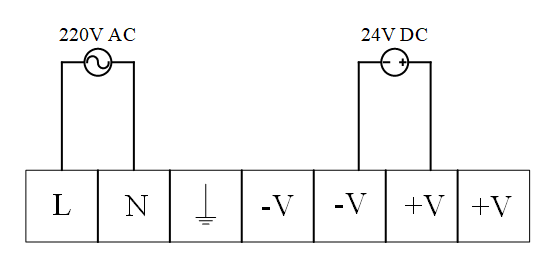


Figure 4: Diagram of power.

**2.3.2 Programmable Logic Controller 1214C DC/DC/DC**

* Programmable Logic Controller is a programmable control device that allows flexible implementation of logic control algorithms through a programming language. Depending on the requirements of each system, the user can program the corresponding events. These events are executed one after another in the program and iteratively. The CPU in the programmable controller will read program from memory, check the memory areas containing input values, and execute the instructions in the program in turn affecting the memory areas that store output values ​​to control actuators, motors, and lights …
* PLC overcomes the disadvantages of wired controllers such as: reducing the number of wires; easily add other devices to the system; flexible change of program; simple maintenance and repair; high reliability; Ability to communicate with other devices...



Figure 4. PLC Siemens S7-1200 CPU 1214C DC/DC/DC.

* Specifications: PLC Siemens S7-1200 CPU 1214C DC/DC/DC

*Product code*: 6ES7214-1AG40-0XB0

*Describe*:

* DC power supply: 20.4-28.8VDC
* 14 DI input: 24VDC- 4mA
* 10 DO output: 24VDC - 0.5A
* Analogue 2 AI: 0-10VDC
* Memory 100 KB
* Pulse frequency 100KHz (4 outputs)
* Connector 1 PROFINET
  + 1. **Conveyors**
* A conveyor belt is a mechanism designed to transport items or materials swiftly and accurately from one place to another. There is a wide variety of conveyor belts made from various materials, offering different functionalities and capabilities.



Figure 5: General structure of the conveyor belt.

(1) - A traction unit equipped with operational components directly responsible for carrying the object.

(2) - A tensioning mechanism responsible for generating and maintaining the required tension for the traction unit.

(3) - A drive station responsible for transmitting motion to the traction unit.

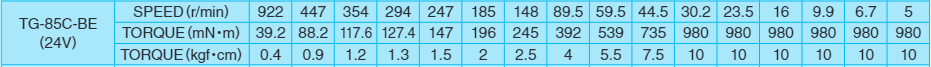
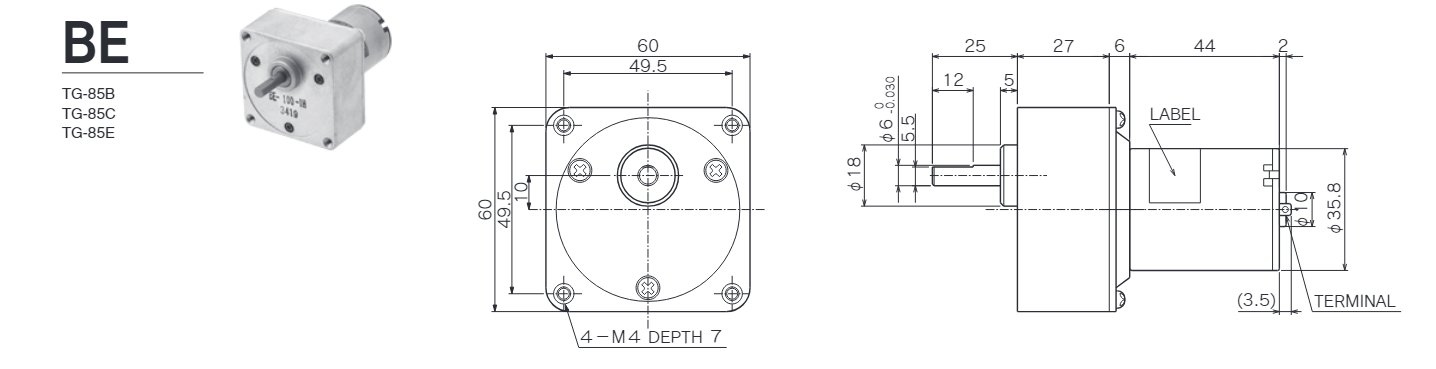
Additionally, there exists a support system (comprising rollers, supports, etc.) positioned beneath the traction unit, serving as a surface for both the traction unit and its operational components.

In practice, depending on the complexity of product classification needs, automatic classification systems vary in scale from large to small. Consequently, conveyor belts need to be available in various sizes to accommodate these classification systems. Recognizing this requirement, the project team aims to design and build a scaled-down model with equivalent functionality to real-world applications: facilitating product transportation and classification based on predetermined size criteria.

### 2.3.4 DC motor for belt conveyor.

In the model, since a belt conveyor is utilized and there's no demand for heavy loads, a high-capacity motor isn't necessary.

The requirements for the conveyor belt are relatively straightforward:

* It needs to operate continuously but should be able to stop when required.
* Precision isn't crucial, and it will mainly handle light loads.
* It should be easy to control and cost-effective.
* Specifications:
* Rated voltage: DC24V
* Motor Type: Permanent magnet
* Synchronous Speed: 5/922rpm
* Voltage Rating: 24 VDC
* Total length of motor: 104mm
* Motor shaft diameter: 6mm
* D axis diameter: 5.5\*
* Output shaft length: 20mm
* Gear: 1/12.5
* Bully: 20/40\*

**2.3.5 Sensors:**

For this application, a sensor is necessary to detect the position of products:

The conveyor belt has a width of 10 centimeters, and objects placed in the center of the belt are approximately 5 centimeters away from the sensor.

Any product material can be used. Therefore, an electro-optical sensor is selected for scanning and detecting products.

Based on these requirements, our group opted for the **E3F-DS10C4** optical sensor.

For product classification based on height, we employed the **E3F2-DS30B4-P1** optical sensor to identify and classify products.



Figure 7: Optical sensor E3F-DS10C4.

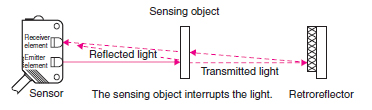


Figure 8: The operating structure of the diffuse reflection sensor

This is a diffuse reflective photoelectric sensor: output is NPN.

Function: detect objects to count and classify products.

***Sensor Features:***

- Good anti-interference.

- Compact and save space.

- Protection against short circuit and power pole connection.

- Operation mode: ON - light on, OFF - light off.

***Rated parameters and technical characteristics:***

* Specifications: E3F DS10C4.
* Dimensions (Diameter x Length): 22 x 70 mm.
* Detection distance: 100mm.
* Rated current: 200mA.
* Shell made of ABS material.
* Standard detection object: 100 x 100 mm.
* Delay characteristic: Up to 20% of detection distance.
* Light source (wavelength): infrared LED (860nm).
* Power supply voltage: 12VDC-24VDC.
* Power consumption: 25mA max.
* Response time: Up to 2.5ms.
* Ambient Temperature: Operating -25°C to 55°C (no freezing or condensation). Store -30°C to 70°C (no freezing or condensation)
* Ambient humidity: 35% to 85% operation, -30% to 95% storage
* Weight (including case): 85g.

**2.3.6 Step Motor 42**

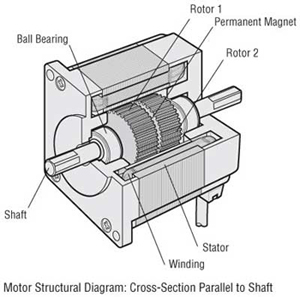
* Purpose: Utilized as a pivot point for the robot to facilitate movement to the locations for picking up and releasing objects.
* Given the requirement to operate under substantial loads, selecting a motor with a gearbox is imperative to enhance the motor's rotational torque.
* Structure of gearbox stepper motor:
* 

Figure 9: Structure of stepper motor

* + 1. **Dirver TB6600**

The TB6600 Stepper Motor Driver Circuit utilizes the TB6600HQ/HG integrated circuit and is designed for use with 2-phase or 4-wire stepper motors, including those with sizes of 42, 57, and 86, capable of handling a load current of up to 4A at 42VDC. This circuit is commonly applied in the construction of machinery such as CNC machines, laser equipment, and various other types of automated machinery.

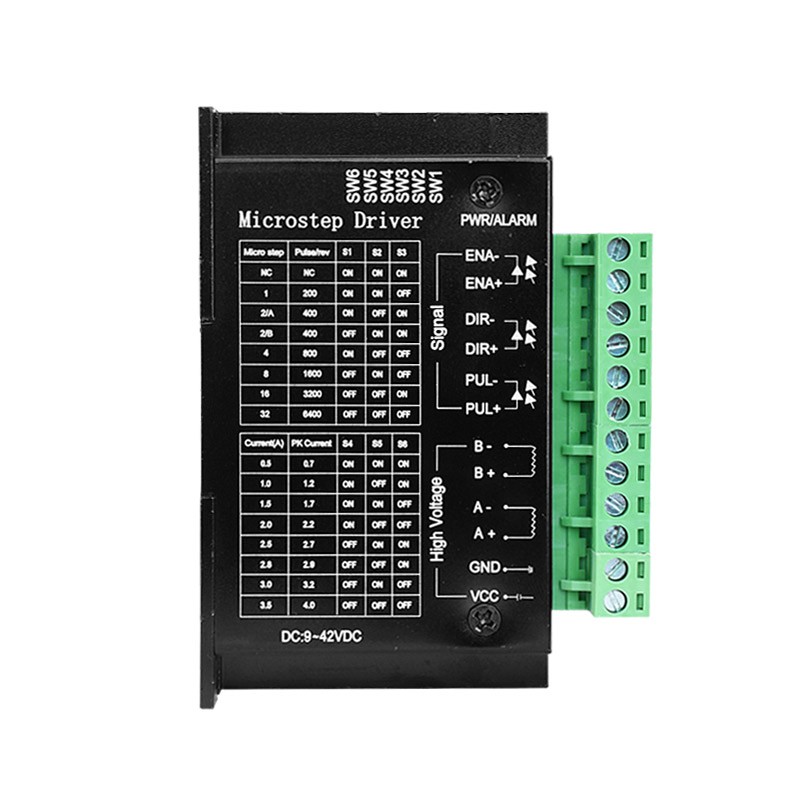


Figure 10: Driver TB6600

Specifications:

+ Input power is 9V – 42V.

+ Maximum supply current is 4A.

+ Input has optical isolation, high speed.

+ Has built-in overcurrent and overvoltage measurement.

+ Weight: 200G.

+ Size: 96 \* 71 \* 37mm.

* + 1. **Relay**

****

Figure 11: Solid State Relay

- Specifications:

- Max operating temperature: 7500C

24

- Current rating: 25A

- Insulation resistance: 1000MΩ 500VDC

- Input type: 5 ~ 60 VDC

- Voltage – Load: 5 ~ 60V

- Size: 62\*45\*23.5mm

- Weight: 98g

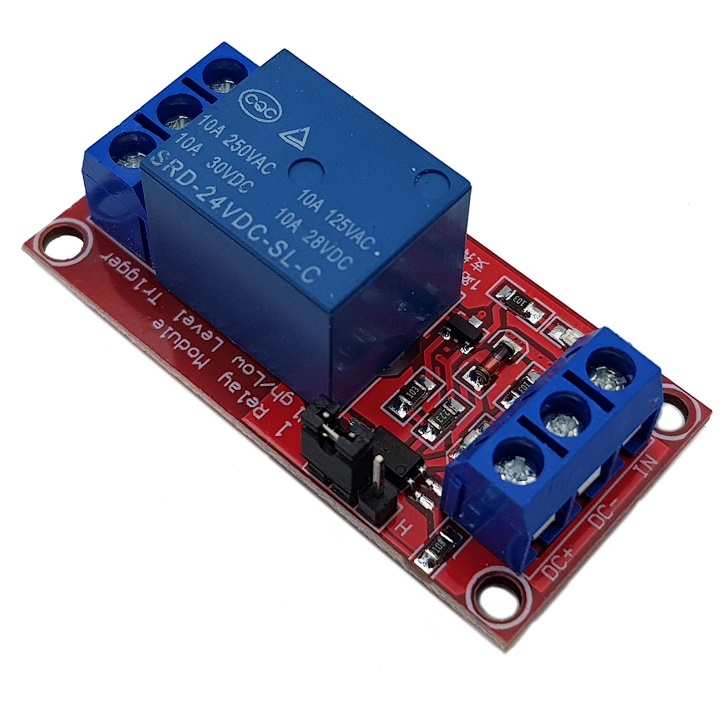


Figure 12: Opto Relay 24V High-Trigger

* Specifications:

+ Control Voltage (Input): 24V DC, with an input current typically around 7-15mA.

+ Output Voltage/Current: Can handle 0-240V AC or 0-60V DC, with a load current from 0.1A to 40A.

+ Isolation Voltage: Typically 2500V AC or higher.

+ Switching Method: Zero-crossing or random, with a response time of 1-10 ms.

+ On-State Voltage Drop/Off-State Leakage: Around 1-1.5V drop at max load, leakage current in microamps to milliamps.

+ Mounting/Environmental: Options for PCB, DIN rail, or panel mount; operating temperature range -30°C to +80°C.

**2.3.9 Robot frame**

The robot frame was conceived and sketched by the team in the robotics project course, with numerous industrial applications, particularly in product classification and assembly.

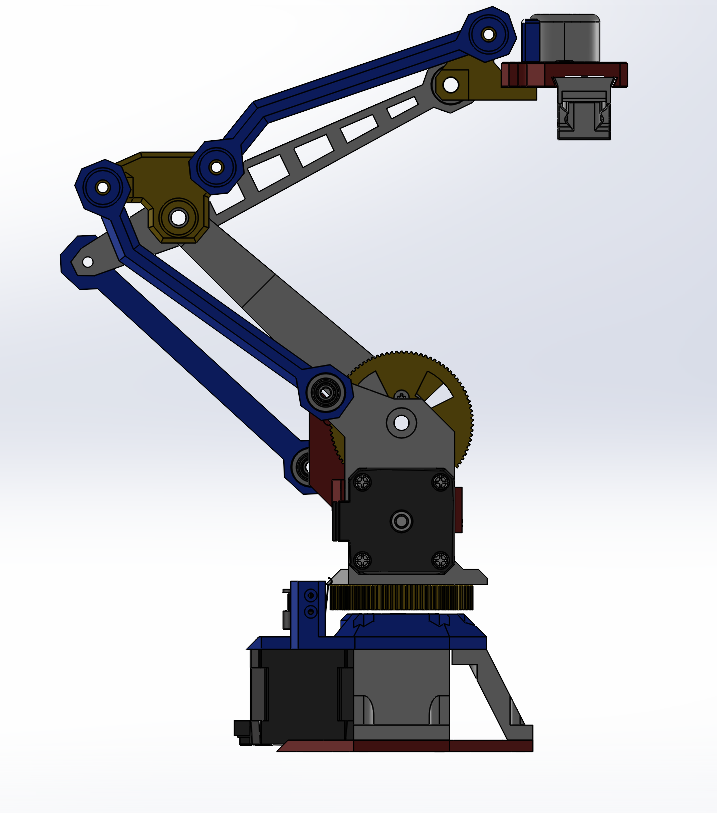


Figure 13: Solid-work model of Robotic arm

* The robot is a meticulously crafted multi-purpose robotic arm capable of gripping objects, performing advanced tasks such as 3D printing and laser engraving. Serving as a comprehensive educational tool, the system is tailored for student programming.
* The arm is engineered for swift transitions between functions such as gripping, utilizing a vacuum cup, holding a pen, performing 3D printing, and laser engraving.
* The frame is created using SolidWorks software and 3D printed using PLA plastic.

## 2.4 Working principle

### 2.4.1. Forward Kinematics

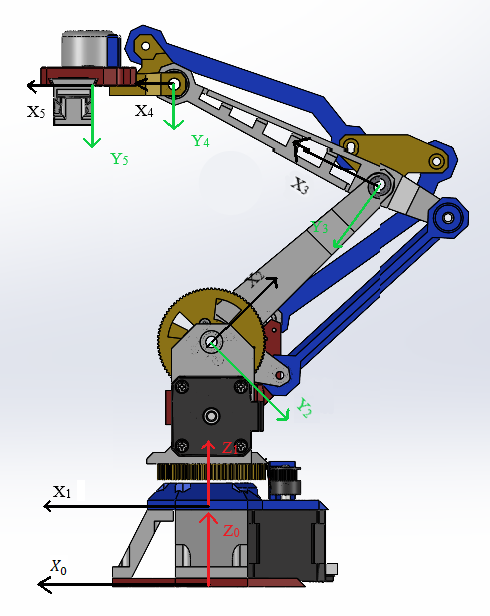
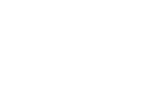


Figure 14: Robot Frame

DH table and forward kinematic:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i |  |  |  |  |
| 1 | 0 | 0 | d1 |  |
| 2 | 90 | 0 | d2 |  |
| 3 | 0 | L2 | d3 |  |
| 4 | 0 | L3 | d4 | 0 |
| 5 | 0 | L4 | 0 | 0 |

Where: d1 = 6.1214cm; d2 = 3cm; L2 = 8.872cm; d3 = 12.015cm; L3 = 11.5cm; D4 = 5.26cm; L4 = 4cm

From the DH-Table, we will get:

=

=

=

=

=

### 2.4.2. Inverse Kinematics

We have:

From (2,1) of matrix and , we have

y\*cos() - x\*sin() =

(where = cos(); sin())

Set , c =

= =>

Where

From (1,1) and (3,1) of matrix and , we have:

( = cos(); sin() ;

= cos(); sin();

= cos(); sin();

= cos(); sin() )

=

=

We set,

=(1)

= (2)

= +

=

=

=

= +

= + + + + +

= + + 2 ( + )

– – = 2 (( + ) + ( – ) )

– – = 2 ((+ ) + ( – ))

=

=

= atan2(,)

*Calculating , from equations (1) and (2)*

Using this formula to calculating , then we have the following formulas to calculating .

Where:

# Chapter 3: Hardware Connection

## 3.1. Flow chart

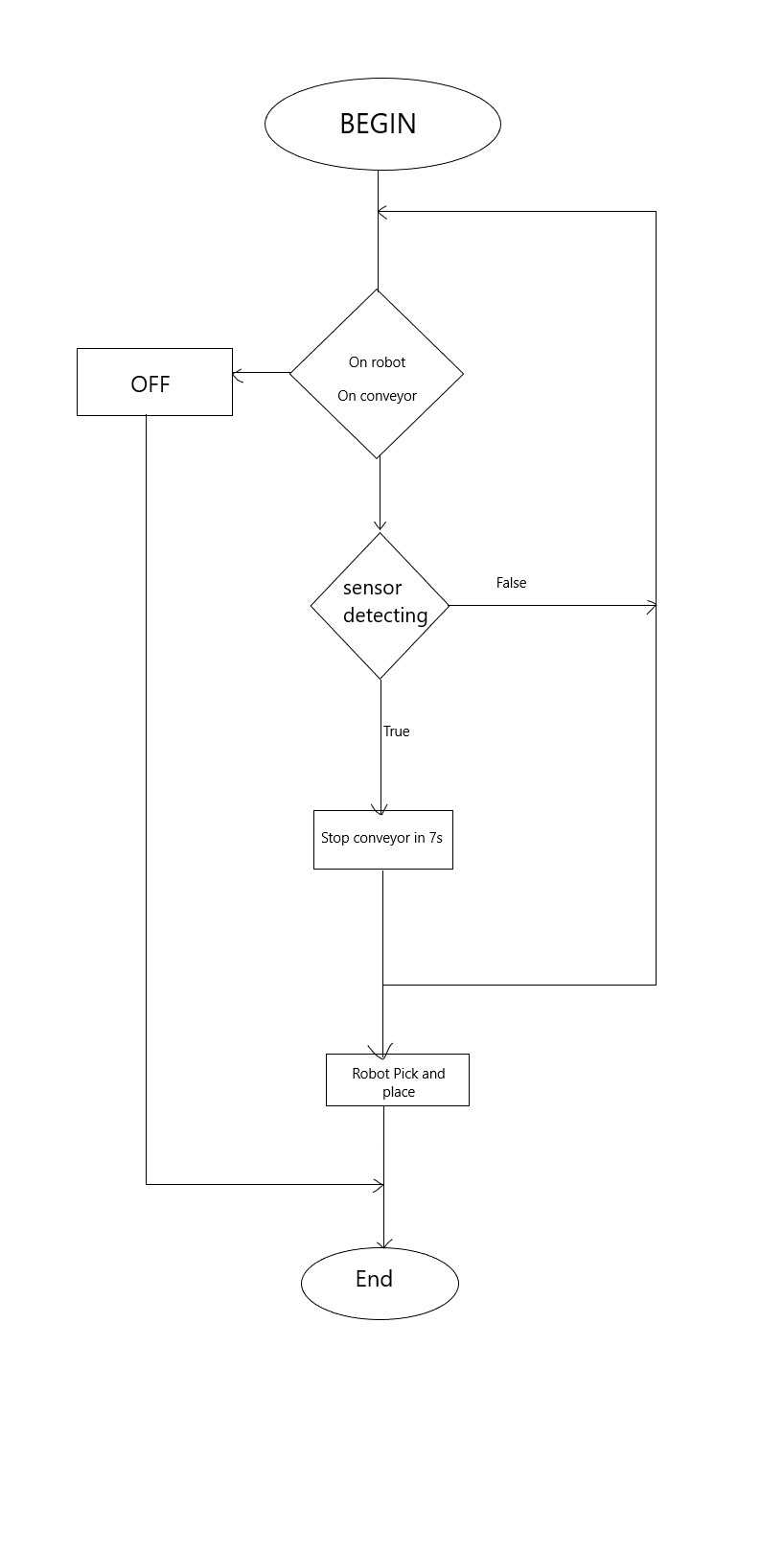
****

Figure 15: Flow chart

## 3.2 Hardware Design

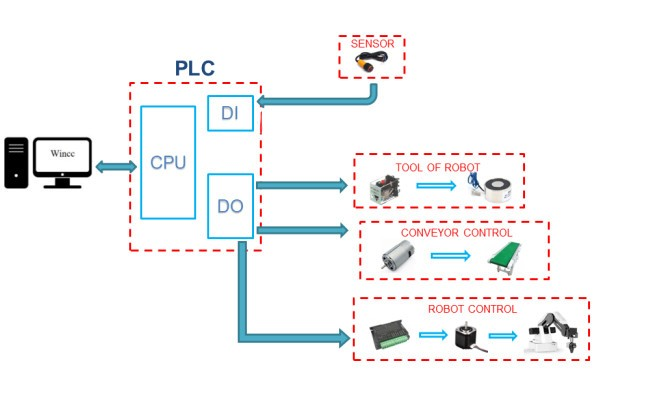
****

Figure 16: Hardware Design

In this section, we will provide a detailed explanation of the hardware design of the 3DOF robotic arm system. This includes a description of the main components and how they are connected to form a complete system.

1. **Power Supply (LITEON EPS-5)**:
   * **Figure 3: Power Supply LITEON EPS-5**
   * **Description**: The LITEON EPS-5 power supply provides stable power to the entire system. The specifications are:
     + Input: AC 220V/50Hz
     + Output: DC 24V, 5A
   * **Connection**: The power supply is connected to the main electrical grid and then distributes power to the PLC, motors, and other components through appropriate wiring and connectors. This ensures a consistent and reliable power source for all connected devices.
2. **Programmable Logic Controller (PLC) Siemens S7-1200 CPU 1214C DC/DC/DC**:
   * **Figure 4: PLC Siemens S7-1200**
   * **Description**: The PLC is a programmable control device that allows the implementation of flexible control algorithms. It can execute logic operations based on programmed instructions. The specifications include:
     + Power Supply: DC 20.4-28.8V
     + Digital Inputs: 14 DI (24VDC-4mA)
     + Digital Outputs: 10 DO (24VDC-0.5A)
     + Analog Inputs: 2 AI (0-10VDC)
     + Memory: 100 KB
     + Pulse Frequency: 100KHz (4 outputs)
     + Connector: 1 PROFINET
   * **Connection**: The PLC is connected to the power supply, sensors, and actuators through its input and output ports. The sensors provide feedback to the PLC, which processes the information and controls the actuators accordingly.
3. **Conveyors**:
   * **Figure 5: General Structure of the Conveyor Belt**
   * **Description**: The conveyor belt is used to transport items or materials efficiently from one place to another. It consists of:
     + Traction Unit: Moves the items along the conveyor.
     + Tensioning Mechanism: Maintains the required tension in the traction unit.
     + Drive Station: Provides the motion to the traction unit.
   * **Connection**: The conveyor system is powered by a DC motor and controlled by the PLC. The sensors on the conveyor belt detect the position of items and send signals to the PLC.
4. **DC Motor for Belt Conveyor**:
   * **Description**: A DC motor with a permanent magnet is used to drive the conveyor belt. The specifications include:
     + Rated Voltage: 24VDC
     + Motor Type: Permanent magnet
     + Synchronous Speed: 5/922rpm
     + Motor Shaft Diameter: 6mm
   * **Connection**: The DC motor is connected to the conveyor belt drive system and controlled via the PLC. Power is supplied from the main power supply.
5. **Sensors**:
   * **Figure 7: Optical Sensor E3F-DS10C4**
   * **Description**: Optical sensors are used to detect the position of products on the conveyor belt. The specifications include:
     + Detection Distance: 100mm
     + Power Supply Voltage: 12VDC-24VDC
     + Power Consumption: 25mA max
     + Response Time: Up to 2.5ms
   * **Connection**: Sensors are connected to the PLC’s digital input ports. They send real-time data to the PLC, which then processes this information to control the actuators.
6. **Step Motor 42**:
   * **Figure 9: Structure of Stepper Motor**
   * **Description**: Stepper motors are used to rotate the robot joints. These motors include a gearbox to increase the torque. The structure comprises the stepper motor and a driver (TB6600).
   * **Connection**: The stepper motor is connected to the robot joints, and the driver (TB6600) interfaces with the PLC for control signals. Power is supplied through the main power supply.
7. **Driver TB6600 (Stepper Motor Driver)**:
   * **Figure 10: Driver TB6600**
   * **Description**: The TB6600 driver controls the stepper motor using the TB6600HQ/HG IC. It is capable of handling up to 4A at 42VDC. Specifications include:
     + Input Power: 9V – 42V
     + Maximum Supply Current: 4A
     + Optical Isolation: High-speed input
     + Overcurrent and Overvoltage Protection
   * **Connection**: The driver is connected to the stepper motor and PLC. It receives control signals from the PLC and adjusts the motor's movements accordingly.
8. **Solid State Relay (Solid State Relay)**:
   * **Figure 11: Solid State Relay**
   * **Description**: Solid-state relays are used to control electrical devices within the system. Specifications include:
     + Max Operating Temperature: 7500C
     + Current Rating: 25A
     + Input Type: 5 ~ 60 VDC
     + Voltage – Load: 5 ~ 60V
   * **Connection**: The relay is connected between the PLC and the DC motor. It allows the PLC to control the motor by switching it on and off.
9. **Robot Frame**:
   * **Figure 13: Solid-work Model of Robotic Arm**
   * **Description**: The robot frame is designed using SolidWorks software and 3D printed using PLA plastic. The frame supports various functions such as gripping objects, 3D printing, and laser engraving.
   * **Connection**: The frame holds all mechanical components and connects to the actuators and sensors, providing structural integrity to the system.

## 3.3. Connection Diagram

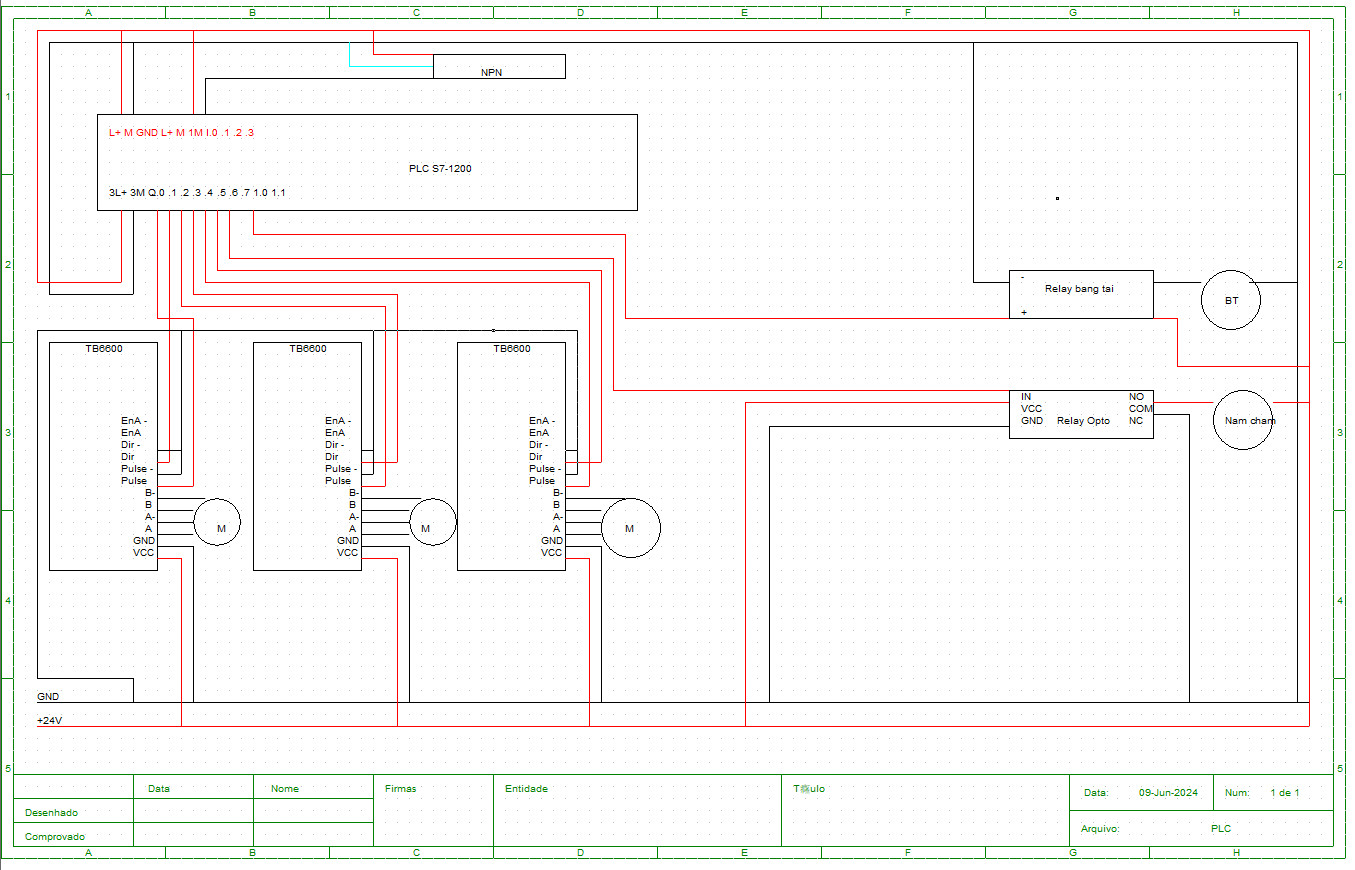


Figure 17: Connection Diagram

**Description:** The connection diagram (Figure 17) illustrates the interconnections between various hardware components of the 3DOF robotic arm system. Each connection is labeled and detailed to ensure clear understanding of how the components interact with each other to form a functional robotic system.

1. **Power Supply (LITEON EPS-5)**
   * **Role**: Supplies 24V DC power to the entire system.
   * **Connections**:
     + Outputs 24V DC to the PLC Siemens S7-1200.
     + Provides power to the DC motor for the conveyor belt.
     + Powers the Driver TB6600.
     + Supplies power to the Opto relay and SSR relay controlling the electronic magnet.
2. **PLC Siemens S7-1200 CPU 1214C DC/DC/DC**
   * **Role**: Central control unit that processes input from sensors and sends control signals to actuators.
   * **Connections**:
     + **Digital Inputs**:
       - Connected to the Optical Sensor E3F-DS10C4.
     + **Digital Outputs**:
       - Connected to Driver TB6600 for step motor control.
       - Connected to Opto relay for electronic magnet control.
       - Connected to a relay for controlling the DC motor of the conveyor belt.
     + **Power Supply**: Receives 24V DC from the LITEON EPS-5 power supply.
3. **DC Motor for Belt Conveyor**
   * **Role**: Drives the conveyor belt to transport items.
   * **Connections**:
     + Powered by 24V DC from the LITEON EPS-5.
     + Controlled by a relay connected to the PLC’s digital output.
4. **Optical Sensor E3F-DS10C4**
   * **Role**: Detects the position of products on the conveyor belt.
   * **Connections**:
     + Connected to the digital input ports of the PLC for real-time data transmission.
5. **Step Motor 42**
   * **Role**: Rotates the robot joints for positioning.
   * **Connections**:
     + Controlled by the Driver TB6600, which receives signals from the PLC.
6. **Driver TB6600**
   * **Role**: Controls the step motor based on signals from the PLC.
   * **Connections**:
     + Receives control signals from the PLC’s digital output.
     + Powered by the 24V DC from the LITEON EPS-5.
     + Drives the Step Motor 42.
7. **Opto Relay (Solid State Relay)**
   * **Role**: Controls the electronic magnet by switching it on and off based on PLC signals.
   * **Connections**:
     + Receives control signals from the PLC’s digital output.
     + Powers the electronic magnet.
     + Connected to the 24V DC supply from the LITEON EPS-5.
8. **Electronic Magnet**
   * **Role**: Used for picking up and holding objects.
   * **Connections**:
     + Controlled by the Opto relay.
     + Receives power through the Opto relay from the LITEON EPS-5.

**Detailed Wiring and Port Connections:**

* **Power Connections**:
  + The LITEON EPS-5 power supply is connected to all components requiring 24V DC power.
* **Control Connections**:
  + The PLC’s digital input ports are connected to the optical sensors to receive detection signals.
  + The PLC’s digital output ports are connected to the Driver TB6600 for step motor control, to the Opto relay for electronic magnet control, and to a relay for the DC motor control.
* **Actuator Connections**:
  + The DC motor for the conveyor belt is connected to a relay, which is controlled by the PLC.
  + The step motor is connected to the Driver TB6600, which in turn is controlled by the PLC.
  + The electronic magnet is connected to the Opto relay, which is controlled by the PLC.

**Summary:** Figure 17 provides a comprehensive view of the interconnections between the power supply, PLC, sensors, motors, and actuators. It shows how the PLC serves as the central hub, processing input from the sensors and controlling the actuators to perform the desired operations. The clear labeling and detailed connections ensure that each component is correctly integrated into the system for optimal performance.

# Chapter 4: Matlab Code and Simulation

## 4.1. Matlab Simulink

In this chapter, the robot’s configure was setting for compute and simulate on Matlab Simulink. The forward/inverse kinematics was computed earlier, now the velocity and the robot dynamic has been computed and simulated in the following figure, with the specific parameters.

*The first case is:*

Testing Inverse Kinematics by giving Px = 25, Py = -15, Pz = 15 (cm).

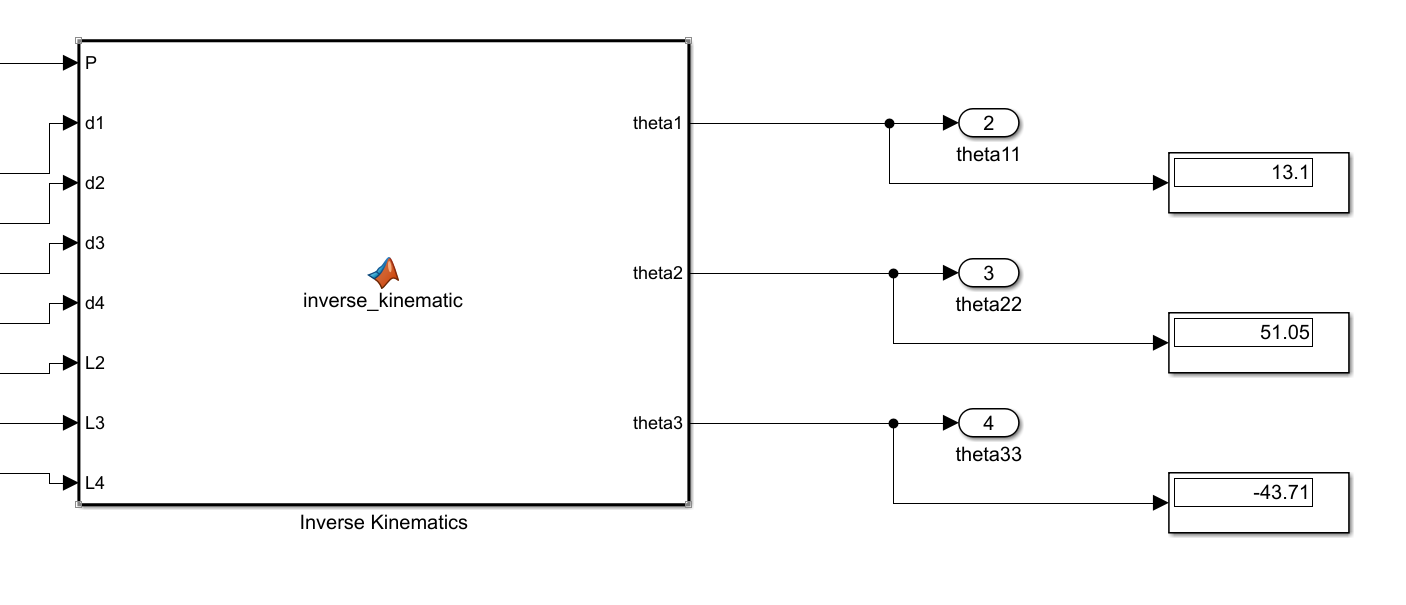


Figure 18: Inverse Kinematics

Apply the computed angle = 13.1 , = 51.05, = -43.71.

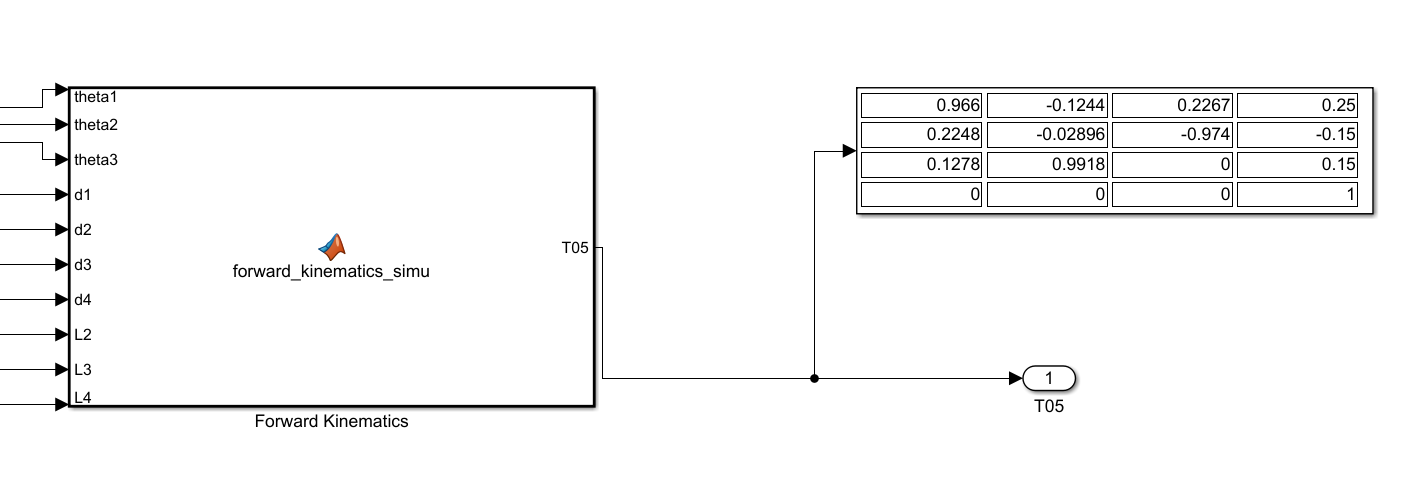


Figure 19: Forward Kinematic with the inverse kinematics inputs

The outputs are Px = 25, Py = -15, Pz = 15 (cm)

Giving ;10;

= 5; = 5; = 5;

; ; ; ; ;

; ; ; ; ;

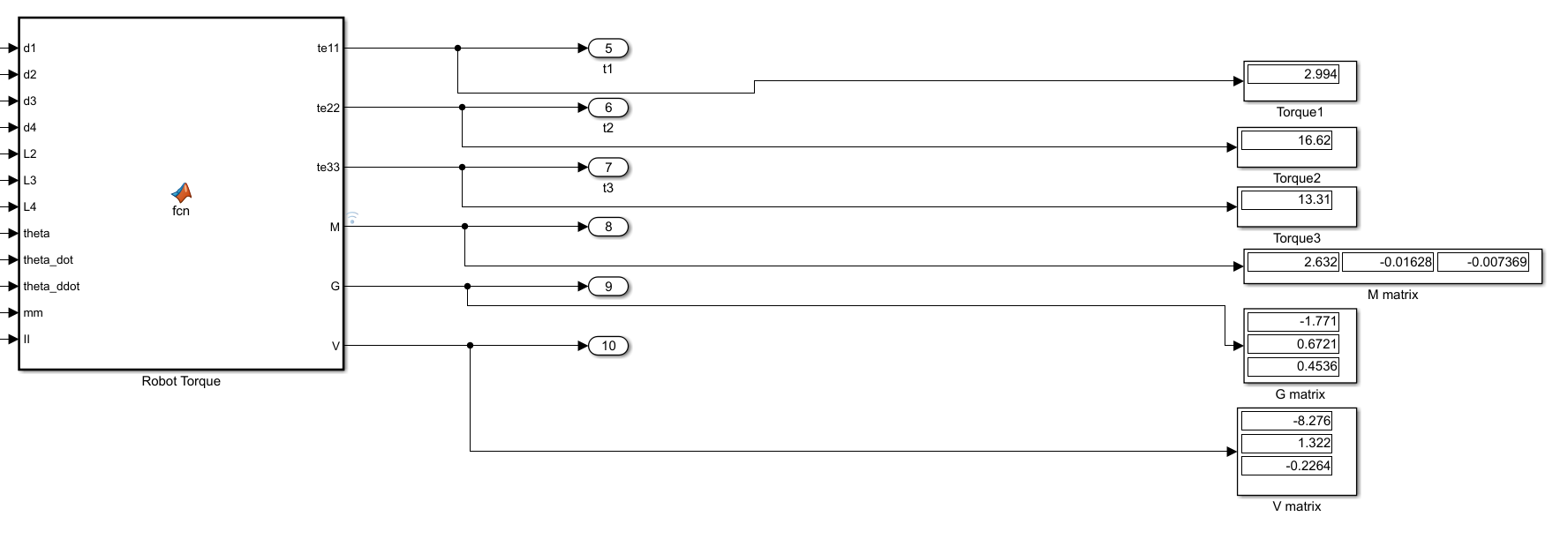


Figure 20: Robot Dynamic

The outputs are:

* Torque 1 = 2.994 Nm
* Torque 2 = 16.62 Nm
* Torque 3 = 13.31
* Mass matrix = [2.63, -0.01626, -0.007369]
* Gravity Matrix = [-1.771; 0.6721; 0.4536]
* Coriolis Matrix =[-8.276;1.322;-0.2264]

*The second case is:*

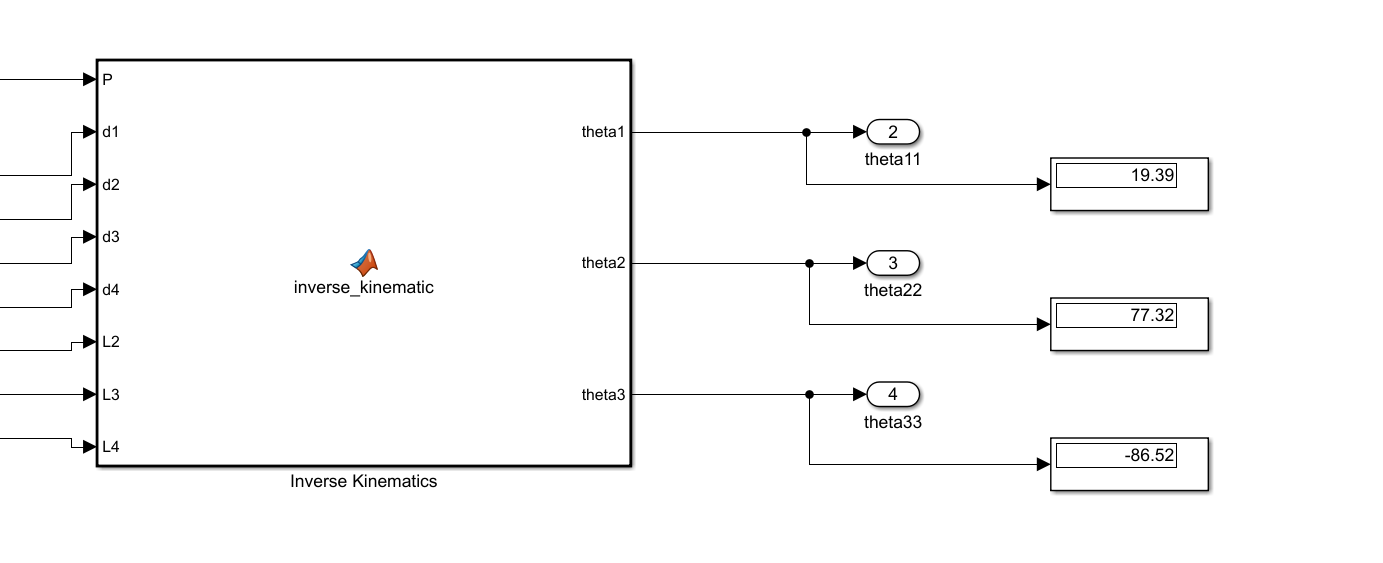
Px = 23, Py = -13.4, Pz = 12.3. (cm)

Figure 21: Inverse Kinematics

Apply the computed angle = 19.39 , = 77.32, = -86.52.

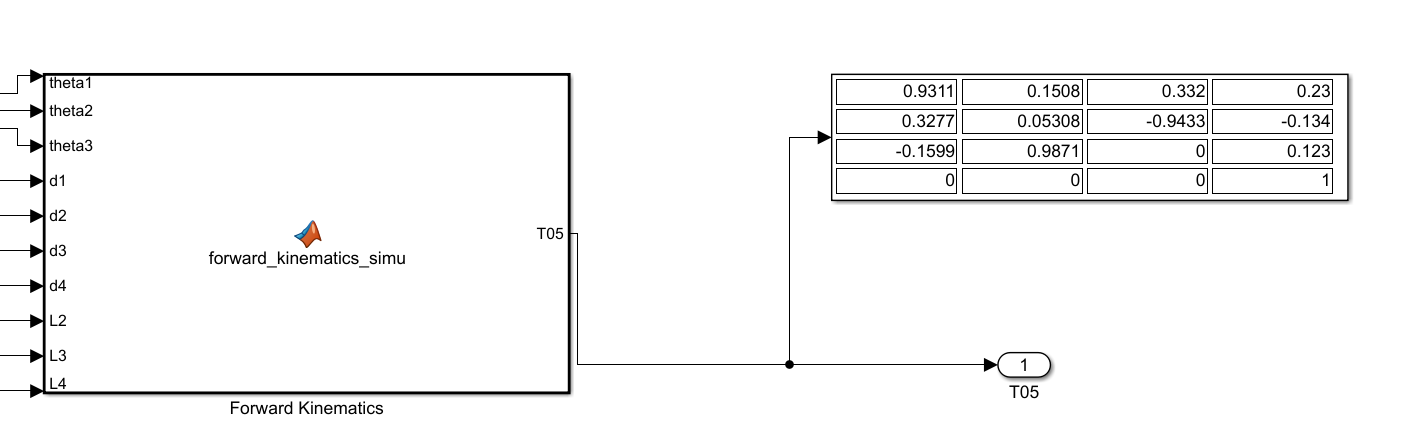


Figure 22: Forward Kinematics

The outputs are Px = 23, Py = -13.4, Pz = 12.3. (cm)

Giving ;

= 0.1; = 0.2; = 0.3;

; ; ; ; ;

; ; ; ; ;

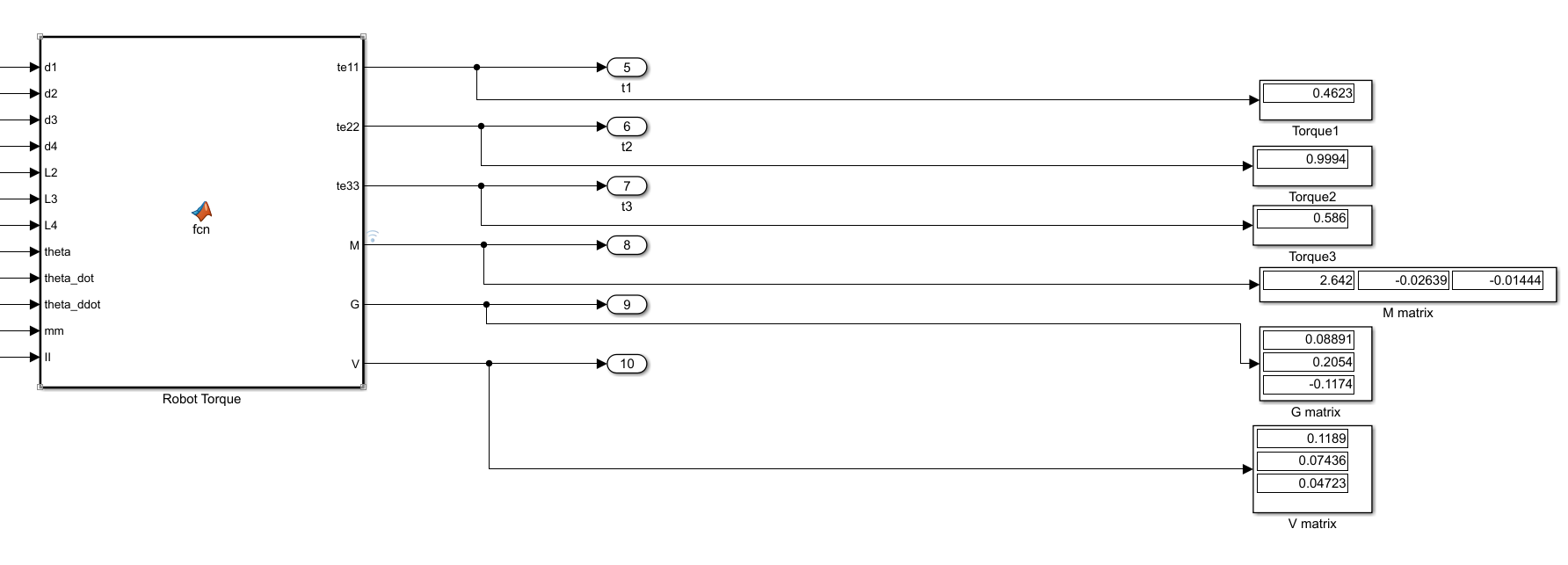
****

Figure 21: Robot Dynamic

The outputs are:

* Torque 1 = 0.4623 Nm
* Torque 2 = 0.9994 Nm
* Torque 3 = 0.586 Nm
* Mass matrix = [2.642, -0.02639, -0.01444]
* Gravity Matrix = [0.08891; 0.2054; -0.1174]

Coriolis Matrix = [0.1189; 0.07436; 0.04723]

# CHAPTER 5: CONCLUSION AND FUTURE ORIENTATION

## 5.1 Conclusion

After studying and researching, my group's system has worked properly according to the requirements that we set. The function blocks operate correctly, ensure accurate data updates, and the system runs stably.

* **System advantages:**
* Stable operation
* Funds at an appropriate level and can commercialize the product.
* Easy to study and expand further.
* **System disadvantages:**
* There is no independent power source, so it cannot operate during a power outage.
* There is no set home button for the robotic arm.
* There is no fixed base and electrical cabinet for the system.
* It is not possible to adjust the distance and angle for the robot to pick up objects.

## 5.2 Future orientation

Design gripper in order to picking up objects instead of electronic magnet. Attach color detection or shape detection into the palletizer system. Add HMI monitor in order to maintaining the system, and report errors to the users.

# APPENDIX

%%

%Deg

%clear all

%clc

L2 = 8.872; L3 = 11.5; L4 = 4; d1 = 6.1214; d2 =3; d3 = 12.015;d4=5.26 ; theta1 = 30; theta2 = 45 ; theta3 = -40;

%CT\_JcraigDEG(alpha, a, d, theta)

T01 = CT\_JcraigDEG(0,0,d1,theta1);

%T12 = CT\_JcraigRAD(sym(pi/2),L1,0,theta2);

T12 = CT\_JcraigDEG(90,0,d2,theta2);

T23 = CT\_JcraigDEG(0,L2,d3,theta3);

T34 = CT\_JcraigDEG(0,L3,d4,0);

T45 = CT\_JcraigDEG(0,L4,0,0);

T05 = T01\*T12\*T23\*T34\*T45;

P = T05(:,4);

theta23 = theta2+theta3;

%%

Px = T05(1,4);Py= T05(2,4); Pz =T05(3,4);

%%

%DEG

alpha =0;

alpha1 = asind(-Py/sqrt((Px^2 +Py^2)));

alpha2 = acosd(Px/sqrt((Px^2+Py^2)));

if Px>0

alpha = alpha1;

else

alpha = alpha2;

end

theta11 = asind((d2+d3+d4)/sqrt((Px^2 +Py^2)))-alpha;

K2 = Pz - d1;

K1 = Px\*cosd(theta11) + Py\*sind(theta11) ;

c3 = ((K1^2 + K2^2-(L3+L4)^2-L2^2)/(2\*L2\*(L3+L4)));

s3 = sqrt(1-(c3^2));

theta33 = -acosd(c3);

theta333 = atan2d(s3,c3);

s2 = ((K2\*(L2+(L3+L4)\*cosd(theta33)))-(K1\*(L3+L4)\*sind(theta33)))/(((L3+L4)\*sind(theta33))^2+(L2+(L3+L4)\*cosd(theta33))^2);

c2 = (K1+(L3+L4)\*s2\*sind(theta33))/(L2+(L3+L4)\*cosd(theta33));

theta22 = atan2d(s2,c2);

%% Robot Dynamic

%%

syms L2 L3 L4 m1 m2 m3 m4 m5 theta1 theta2 theta3 d1 d2 d3 d4 alpha alpha1 alpha2 I1 I2 I3 I4 I5 fx fy fz theta1\_dot theta1\_ddot theta2\_dot theta2\_ddot theta3\_dot theta3\_ddot omega0 v0

assume(m4,'real'); assume(I3,'real');

assume(L2,'real'); assume(m1,'real'); assume(m5,'real'); assume(I4,'real');

assume(L3,'real'); assume(m2,'real'); assume(I1,'real'); assume(I5,'real');

assume(L4,'real'); assume(m3,'real'); assume(I2,'real');

assume(theta1,'real'); assume(theta1\_dot,'real'); assume(theta1\_ddot,'real');

assume(theta2,'real'); assume(theta2\_dot,'real'); assume(theta2\_ddot,'real');

assume(theta3,'real'); assume(theta3\_dot,'real'); assume(theta3\_ddot,'real');

assume(d1,'real');

assume(d2,'real');

assume(d3,'real');

assume(d4,'real');

assume(alpha,'real');

assume(alpha1,'real');

assume(alpha2,'real');

assume(fx,'real');

assume(fy,'real');

assume(fz,'real');

assume(omega0,'real');

%% Settiing Parameters

R01 = T01(1:3,1:3); R011 = R01'; P01 = T01(1:3,4);

R12 = T12(1:3,1:3); R122 = R12'; P12 = T12(1:3,4);

R23 = T23(1:3,1:3); R233 = R23'; P23 = T23(1:3,4);

R34 = T34(1:3,1:3); R344 = R34'; P34 = T34(1:3,4);

R45 = T45(1:3,1:3); R455 = R45'; P45 = T45(1:3,4);

R02 = R01\*R12; T02 = T01\*T12;

R03 = R02\*R23; T03 = T02\*T23;

R04 = R03\*R34; T04 = T03\*T34;

R05 = R04\*R45; T05 = T04\*T45;

%%

%% LINK VELOCITy

omega0 = [0;0;0]; v0 = [0;0;0];

P0c = [0 0 d1]'; P1c = [0 -d2 0]'; P2c = [L2/2 0 d3]'; P3c = [L3/2 0 d4]'; P4c = [L4/2 0 0]';

theta\_dot = [theta1\_dot;theta2\_dot;theta3\_dot;0;0];

R = {R011,R122, R233, R344, R455};

Rc = {R01,R02,R03,R04,R05};

P = {P01,P12,P23,P34,P45};

Pc = {P0c,P1c,P2c,P3c,P4c};

T = {T01,T02,T03,T04,T05};

omega = sym(zeros(3, 5));

v = sym(zeros(3, 5));

vc = sym(zeros(3,5));

Pcc = sym(zeros(4,5));

for i = 0:4

switch i

case 0

thetadoti = [0;0;theta\_dot(1)];

case 1

thetadoti = [0;0;theta\_dot(2)];

case 2

thetadoti = [0;0;theta\_dot(3)];

case 3

thetadoti = [0;0;theta\_dot(4)];

case 4

thetadoti = [0;0;theta\_dot(5)];

end

if i == 0

omega(:, i+1) = R{i+1} \* omega0 +thetadoti;

v(:, i+1) = R{i+1} \* v0 + SKMN(omega0) \* P{i+1};

vc(:,i+1) = Rc{i+1} \* v(:,i+1) + SKMN(omega(:,i+1)) \* Pc{i+1};

else

omega(:, i+1) = R{i+1} \* omega(:, i)+thetadoti;

v(:, i+1) = R{i+1} \* v(:, i) + SKMN(omega(:, i)) \* P{i+1};

vc(:,i+1) = Rc{i+1}\*v(:,i+1)+SKMN(omega(:,i+1))\*Pc{:,i+1};

end

omega(:, i+1) = simplify(omega(:, i+1));

v(:, i+1) = simplify(v(:, i+1));

vc(:,i+1) = simplify(vc(:,i+1));

end

Pcc(:,1)= T{1}\*[P0c;1];

Pcc(:,2)= T{2}\*[P1c;1];

Pcc(:,3)= T{3}\*[P2c;1];

Pcc(:,4)= T{4}\*[P3c;1];

Pcc(:,5)= T{5}\*[P4c;1];

Pcc = simplify(Pcc);

%% Larange

syms g t;

K = sym(zeros(1, 5));

U = sym(zeros(1, 5));

m = [m1, m2, m3, m4, m5];

I = [I1, I2, I3, I4, I5];

g = [0, g, 0];

for i = 1:5

% Kinetic Energy

K(i) = 1/2 \* m(i) \* vc(:,i)' \* vc(:,i) + I(i) \* omega(:,i)' \* omega(:,i);

% Potential energy

U(i) = -m(i) \* g \* Pcc(1:3,i);

end

% Kinetic and Potential Energy

Ktotal = sum(K);

Utotal = sum(U);

% Largarian equation

L = simplify(Ktotal - Utotal);

%%

% Define symbolic variables for theta\_dot and theta\_ddot

theta\_dot = [theta1\_dot; theta2\_dot; theta3\_dot];

theta\_ddot = [theta1\_ddot; theta2\_ddot; theta3\_ddot];

theta = [theta1; theta2; theta3];

% Calculate the derivatives of the Lagrangian with respect to theta\_dot and theta

dLdtheta\_dot = jacobian(L, theta\_dot).';

dLdtheta = jacobian(L, theta).';

% Calculate the time derivatives of dLdtheta\_dot

dLdtheta\_dot\_dt = jacobian(dLdtheta\_dot, theta) \* theta\_dot + jacobian(dLdtheta\_dot, theta\_dot) \* theta\_ddot;

% Calculate the equations of motion

tau = simplify(dLdtheta\_dot\_dt - dLdtheta);

% Extract individual torques

te11 = tau(1);

te22 = tau(2);

te33 = tau(3);

%%

% Extract the inertia matrix M

M = jacobian(tau, theta\_ddot);

M = simplify(M);

% Extract the Coriolis and centrifugal matrix V

C = sym(zeros(3, 3));

for k = 1:3

for j = 1:3

for i = 1:3

C(k, j) = C(k, j) + 0.5 \* (diff(M(k, j), theta(i)) + diff(M(k, i), theta(j)) - diff(M(i, j), theta(k))) \* theta\_dot(i);

end

end

end

V = simplify(C \* theta\_dot);

% Extract the gravity vector G

G = jacobian(Utotal, theta).';

G = simplify(G);

% Display the results

disp('Inertia Matrix (M):');

disp(M);

disp('Coriolis and Centrifugal Matrix (V):');

disp(V);

disp('Gravity Vector (G):');

disp(G);

function [t] = CT\_JcraigDEG(alpha, a, d, theta)

t = [cosd(theta) -sind(theta) 0 a;

cosd(alpha)\*sind(theta) cosd(alpha)\*cosd(theta) -sind(alpha) -sind(alpha)\*d;

sind(alpha)\*sind(theta) sind(alpha)\*cosd(theta) cosd(alpha) cosd(alpha)\*d

0 0 0 1];

end

function [t]= SKMN(B)

x = B(1,1);y = B(2,1);z = B(3,1);

B = [x;y;z];

t = [0 -z y;

z 0 -x;

-y x 0];

end

# REFERENCES

[1] Craig, J. J. (2004). Introduction to Robotics: Mechanics and Control (3rd ed.). Pearson. (Chapters 3, 4, 5, 6).

[2] Nguyen, V. (2022, October 1). 3DOF Robotic Arm System Overview [Video]. YouTube. <https://youtu.be/_RtMGaMUV6g>