UNIVERSITÄT SIEGEN

Department
Elektrotechnik und Informatik

Naturwissenschaftlich
Technische Fakultät

Lehrstuhl für Regelungs- und Steuerungstechnik

Mechatronic Systems Laboratory
Winter term 2023-24

Report
Manipulator Robot – Pick & Place

Submitted by: Group 5

Name of the Student	Matriculation Number
Aakash Deshpande	1701953
Chirag Angadi	1725283
Raaj Nihaal Jeevannath Kotian	1702017
Sanketh Ganesh	1712320
Shri Hari Murali Krishna	1711109

Table of Contents

1. Objective 3
2. Overview of Robot 3
    2.1 Robot Operating Range 3
3. Robot Inverse Kinematics 4
    3.1 Equation for θ_1 4
    3.2 Equation for θ_2 4
4. PI Controller for Joints 5
5. Robot pick and place process 5
    5.1 Flow chart for pick and place 6

Table of Figures

1. Figure 1: Sensors and Actuators -------- 3
2. Figure 2: Gear ratio -------- 3
3. Figure 3: Operating Areas -------- 3
4. Figure 4: Robot links Configuaration 4
5. Figure 5: Top View -------- 4
6. Figure 6: Side View of robot -------- 4
7. Figure 7: Side View case 1 -------- 4
8. Figure 8: Side View case 2 -------- 5

1. Objective

- The objective of robot manipulator is to complete the sequential task by picking an object from desired loaction and placing it at respective locations.
- To implement the PID controller at joint angles in program for efficient movement of robot links.

2. Overview of Robot

- A robotic manipulator consists of a sequence of rigid links joined by articulated joints, designed with one end fixed and the other end free to create an arm-like structure. This mechanism enables the manipulation of objects within a specified range of motion, typically dictated by the number of degrees of freedom. In the context of a Pick and Place task, a specific robotic manipulator comprises four links and three motors to execute its functions.

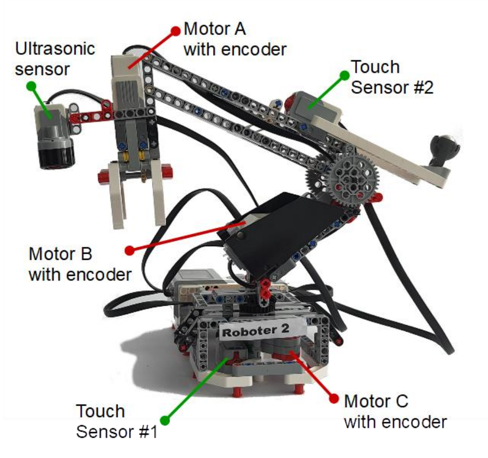

Figure 1: Sensors and Actuators
Source: Task Sheet

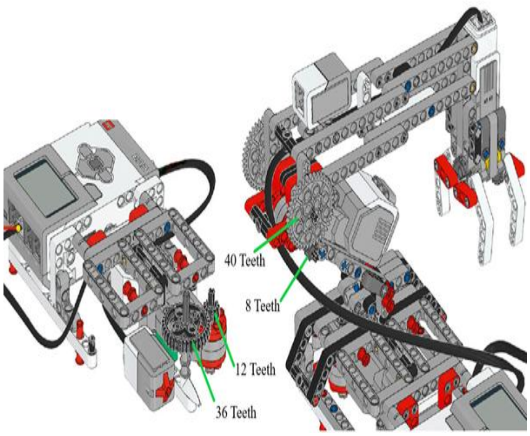

Figure 2: Gear ratio
Source: Task Sheet

2.1 Robot Operating Range

- The operating range of robot is restricted between three stations namely A, B and C by touch sensor indications. Considering B position as homing postion, A and C stations are at +90deg and -90deg respectively.

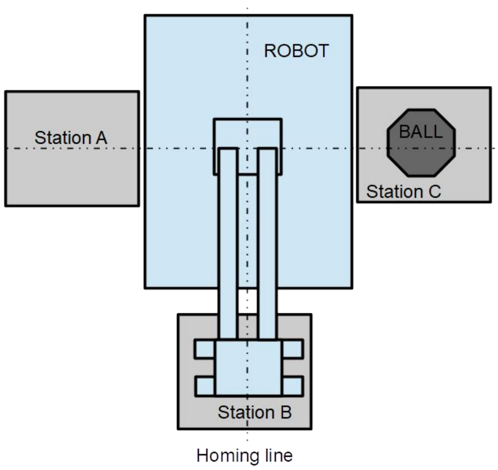

Figure 3: Operating Areas
Source: Task Sheet

3. Robot Inverse Kinematics

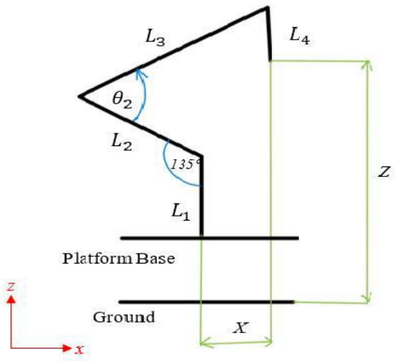

Figure 4: Robot Links Configuaration

- As the robot has two motors (i.e B and C) which are mainly involved for movement of robot links. The the two degrees of freedom (i.e angles θ_1 and θ_2) are calculated using inverse kinematics analytical approch.
- θ_1 and θ_2 are obtained in terms of (x,y,z) coordinates of manipulator.

3.1 Equation for θ_1

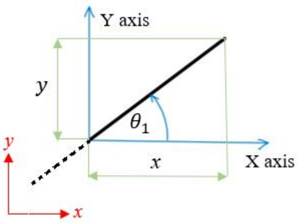

Figure 5: Top View

- In the top view of the manipulator robot, the angle θ_1 represents the rotation of link L_1 about the z-axis.
- This angle, calculated using the arctangent function of the ratio (y/x), determines the orientation of the first link relative to the Cartesian coordinates (x,y).
- $\theta_1 = \tan^{-1}(y/x)$

3.2 Equation for θ_2

Case 0:

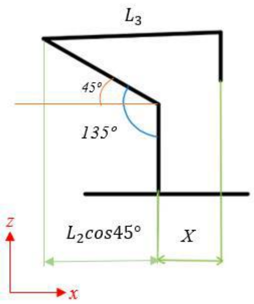

Figure 6: Side View of robot

- Consider a position where L3 is parallel to the ground and θ1 = 0o.
- Considering X from the diagram

  $X = L_3 - L_2 \cos 45^o$
  $X = 117.9$ mm

Case 1: Considering Link 3 (L_3) to be above the horizontal axis of joint 2.

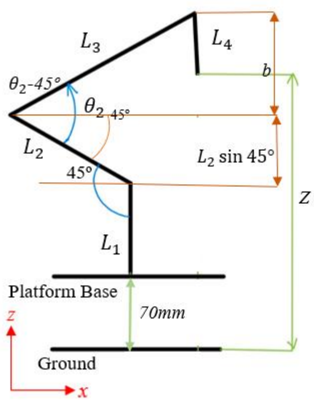

Figure 7: Side View case 1

From Fig 7, equations can be obtained:

$L_4 + Z = b + L_2 \ \text{Sin}\ 45^o + L_1 + 70$

$b = L_4 + Z - (L_2/\sqrt{2}) - L_1 - 70$ ---------------- 1

$\text{Sin}(\theta_2 - 45^o) = b/L_3$ ---------------- 2

From equation 1 & 2:

$$\boldsymbol{\theta_2 = \text{Sin}^{-1}\left((L_4 + Z - (L_2/\sqrt{2}) - L_1 - 70)/L_3 \right) + 45^o}$$

Case 2: Considering Link 3 (L_3) to be below the horizontal axis of joint 2.

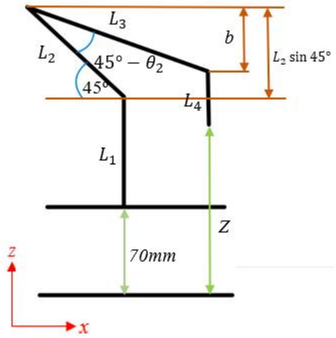

Figure 8: Side View case 2

From Fig 8, equations can be obtained:

$$L_4 + Z + b = L_2 \sin 45^o + L_1 + 70$$

$$b = (L_2/\sqrt{2}) + L_1 + 70 - L_4 - Z \quad \text{----------------} \ 1$$
$$\sin(45^o - \theta_2) = b/L_3 \quad \text{----------------} \ 2$$

From equation 1 & 2:

$$\boldsymbol{\theta_2 = 45^o - \sin^{-1}((L_2/\sqrt{2}) + L_1 + 70 - L_4 - Z)/L_3)}$$

4. PI Controller for Joints

- PI controller is implemented in operations of robot while the robot is moving from one station to other station and while picking placing of object at respective stations.
- The controller is designed as function in MATLAB script (**Function name: PIController**) which called in other MATLAB functions namely, **position, pick and place**. (refer MATLAB code file: Group5.m)
- The **PIController** function is designed in such a way that it will take error and total error as arguments and provide controlled value (Motor speed):
  Controlled value = Kp*error + Ki*total error

  Kp = 0.05 Ki = 10
- An if-else statments are used to make sure controlled value (motor speed) will not go beyond certain limit to ensure the robot resitance from wirings.
- If **0 < controlled value < 20** then function **returns 20**, if **-20 < controlled value < 0** then function **returns**
- **-20** as speed.
- The error is difference between the desired position(encoder value) and currrent position (encoder value).
  error = Desired encoder vaue - Current encoder value
- The total error is calculated from accumulation of error till the operation is changed to other.
  total error = total error + error

5. Robot pick and place process

- The task for picking an object from one station and placing it at other station is accomplised by sequential tasks performed.
- Initially the robot determine the height of the each station and store the data in variables. (Function name: **initial_height)**
- An function **home_posi** is defined in code to position the manipulator at certain position, considered as initially position. A combination of motor B and motor C will lead motor to be in home position from any other position.
- Further function namely **position** is defined for positioning of robot arm at desired station using base motor C.
- Functions like **pick and place** are used to actuate motor B for arm movement and motor A for endeffector opening and closing.
- **Station_A, Station_B, Station_C** functions are defined to perform set of operations like picking object and placing object at there respective stations.

5.1 Flow chart for pick and place