Project Work in Mechatronic Systems (Code: 4MECHMA032-S)

MANIPULATOR ROBOT - PICK AND PLACE

WS 2023/24
Group 17

1717750	Arjith Praison Vijaya Sankar
1702033	Guru Prasaath Pushparaj
1700943	Monish Senthil Kumar
1702059	Milton Sebaastin Clement Sepasthiyammal
1783980	Oussema Kassebi

Date : 22 January 2023
Place: Siegen

MANIPULATOR ROBOT - PICK AND PLACE

Introduction

Manipulator robots are mechanical systems designed to perform a variety of tasks, and one common application is pick-and-place operations. These robots are equipped with actuators, sensors, and links that work together to achieve precise and controlled movements. In our manipulator system, three actuating motors (Motor A, Motor B, and Motor C) drive the robot, providing the necessary power for its movements.

The manipulator consists of five links, each with specific dimensions (Link-1 to Link-4) as shown in Fig 1, and the entire system is designed to facilitate pick and place tasks with precision and efficiency. The links are interconnected to form a kinematic chain, allowing the robot to navigate its environment and interact with objects.

Fig 1: Manipulator Robot

Key Components of the Manipulator System:

1. **Actuating Motors (Motor A, Motor B, Motor C):** These motors drive the robot's movements and play a crucial role in executing pick and place tasks. The use of gearboxes, especially for Motor A, Motor B and Motor C, is essential for controlling and coordinating their rotations, ensuring synchronized motion of the robot's components.

2. **Sensors (Touch Sensor #1, Touch Sensor #2, Motor A/B/C Encoders):** The sensors provide feedback to the control system, enabling the robot to sense its environment and make real-time adjustments. Touch sensors (#1 and #2) help in setting boundaries for the robot's movements, while encoders for each motor provide information about the position of the corresponding links.

3. **Kinematic Constraints:** The manipulator system incorporates specific kinematic constraints to govern the robot's motion. For example, the rigid connection between Link-1 and Link-2 at an obtuse angle of 135°, the perpendicular orientation of Link-4 to the ground plane, and the bounded rotation of Link-3 by Link-2 and touch sensor #2 all contribute to the controlled and predictable movement of the robot.

4. **Gearboxes for Motor A, Motor B and Motor C:** The gearboxes play a crucial role in the mechanical advantage and coordination of the robot's motions.

Task Description

- Developing inverse kinematic equations for the manipulator using the geometric approach.
- Programming the manipulator to grasp and release a ball, with video proof of the manipulator picking and placing an object.
- Implementing and tuning P, PI, PD, and PID controllers for joint angle control and minimizing the error.
- Developing functions for homing and picking/placing tasks with varying station heights, supported by video evidence.
- Successfully executing pick-and-place maneuvers without collisions.

Inverse Kinematics with Geometric Approach

The workspace is constructed as a 2D Geometry as shown in the image below. Where L_0 - L_4 represent the link length. The angles theta 2 and beta can be calculated using basic trigonometry formulae as follows:

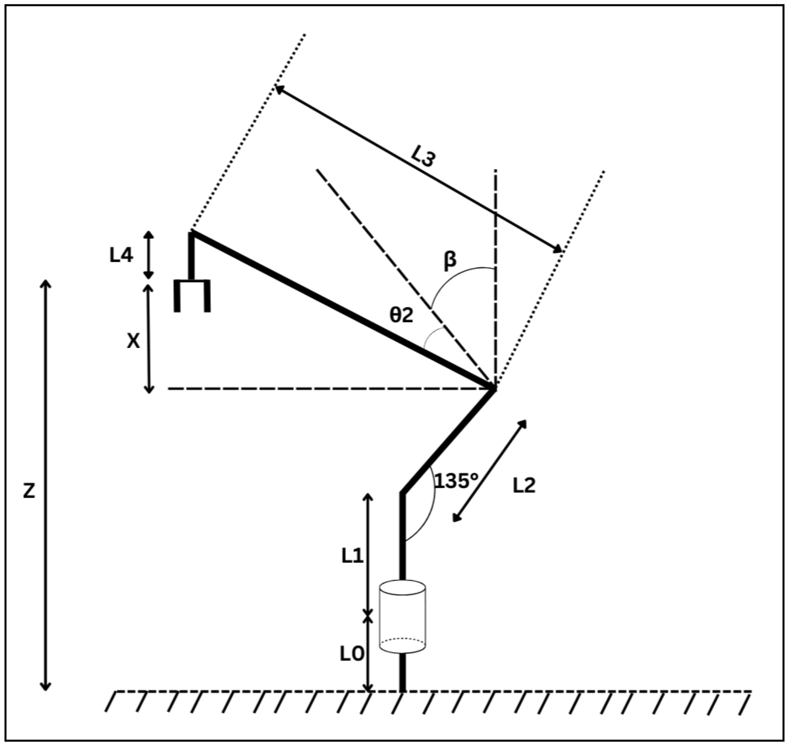


Fig 2: Workspace 2D Geometry

$\cos(\theta_2 + \beta) = (L_4 + x) / L_3$ - (1)
$\theta_2 + \beta = \cos^{-1}((L_4 + x)/L_3)$
$\theta_2 = -\beta + \cos^{-1}((L_4 + x)/L_3)$

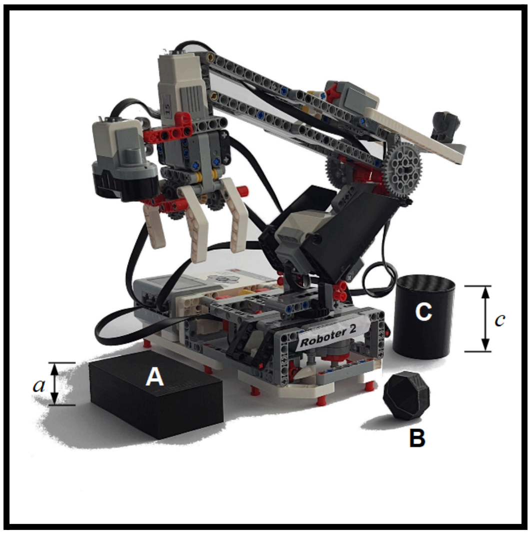
When the gripper is extended to the topmost position, the value of θ_2 tends to zero and the value of $\boldsymbol{\beta}$ can be represented as

$\beta = \cos^{-1}((L_4 + x)/L_3)$ - (2)

The value of x in terms of the height z and the link lengths is given by:
$x = z - (L_0 + L_1 + L_2 \sin 45°)$ - (3)
The value of z is measured and subsequently the value of x and the angle $\boldsymbol{\beta}$ is calculated by using the expression (3) in (2).

Workspace Description

In the operating area, there are three stations, namely Station A, Station B, and Station C. The heights of Station A and Station C are denoted by "a" and "c," respectively. The operating environment is shown in Fig 3 and Fig 4.



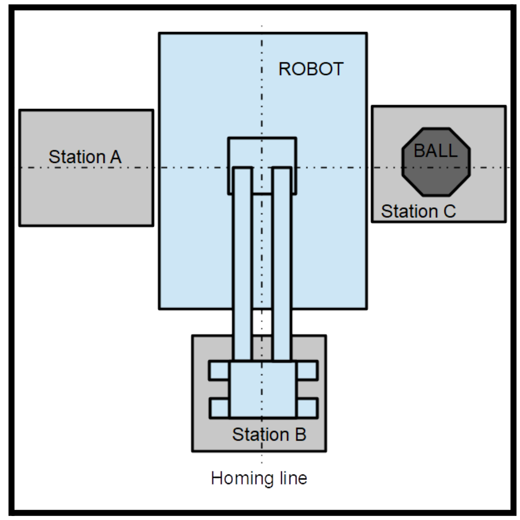


Fig 3: Stations A, B, and C.
The payload is the ball located at station B in this case.

Fig 4: Operating areas

Flow Chart - Pick, Place and Return to Home

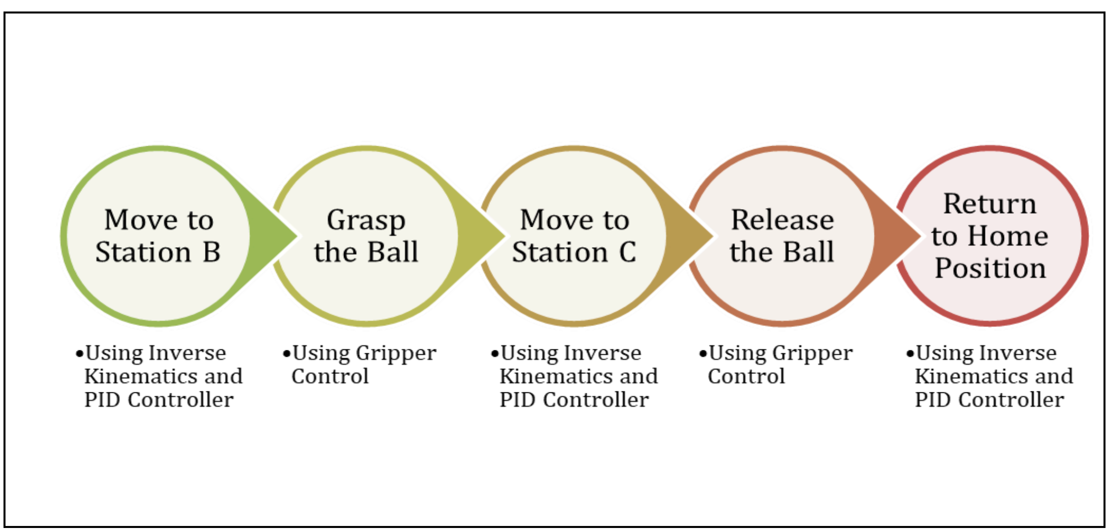


Fig 5: Flow Chart for the Task

Code Explanation

- **Homing Function** - The variables for motors, touchSensor, and sonicSensor are created initially. The top function is first called and it is responsible for moving the arm to the top position until a touch sensor is triggered, then motorC is started. The touch sensor is checked and if it is not pressed the speed of the motor is increased to 25 % of the maximum speed until the sensor is triggered. The purpose of the homing function is to initiate the homing process for the robot, which involves bringing the arm to a reference position. The process involves checking the state of a touch sensor, rotating a motor until the touch sensor is triggered, and then calling another function (***downpos***) to control the arm position further. The function ends with a pause to allow for stability.

- **Station functions** - The functions Station A, Station B, and Station C are used for picking up or placing the ball at the respective stations. These functions make use of other helper functions defined in your code, such as ***downpos***, ***toppos***, ***openA***, ***closeA***, and top, to control the robot's movements, gripper, and arm positioning. The exact actions depend on the value of the parameter ***'k'*** passed to each function, where ***'k=0'*** indicates picking up a ball, and ***'k=1'*** indicates placing a ball.

- **PID Controller Function -** The PID (Proportional-Integral-Derivative) controller is a control mechanism widely used in engineering for controlling systems. In this specific implementation, it's used to control the movement of a motor based on the error between the desired position and the current position.

  Equation of a PID controller in continuous form

  $$\mathbf{u(t) = Kp\ e(t) + Ki \int e(t)\ dt + (Kd / T)\ d/dt\ e(t)}$$

  The respective equation in digital form is

  $$\mathbf{U_k\ (Motor\ Speed) = K_p e_k + K_I\ T\ \Sigma^{i=k-1}\ e_i + K_D/T\ (e_k - e_{k-1})}$$

  - Proportional Part **('kp * ek'):**
    - The proportional term is proportional to the current error **('ek'),** which is the difference between the desired position and the current position
    - It determines the immediate response to the current error.
  - Integral Part **('ki * t * t_error')**:
    - The integral term integrates all the past errors over time **('t_error').**
    - It helps eliminate the steady-state error and brings the system to the desired position.
  - Derivative Part **('(kd / t) * de')**:

        - The derivative term reduces the rate of change of error **(‘de’)** with respect to the time **(‘t’).**
        - It prevents the system from overshooting.
    - Error Updation **(‘ek0 = ek’):**
        - The current error is stored as **(‘ek’).**
        - The previous error is stored as **(‘ek0’).**

Results

The geometric approach was successfully applied to derive the inverse kinematics equations for the manipulator. The manipulator was controlled to pick and place the ball. P, PI, PD, and PID controllers were implemented and tuned for joint angle control. Custom behaviors were modeled for homing, picking, and placing tasks at different stations with varying heights.

Reference

[1]https://www.wevolver.com/article/what-are-manipulator-robots-overview-of-types-and-applications

[2] https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313