Robotic Systems I: Final Project

Chinmay Amrutkar[1]
[1]Arizona State University, Tempe, AZ

Keywords:

MyCobot Pro 600
Maze Problem
Open CV
Inverse Kinematics
Path Planning
MATLAB
AI kit camera

ABSTRACT

This report details the integration of the MyCobot Pro 600 collaborative robot with its digital twin, enabling precise motion planning and execution. The project began by using Python and OpenCV to solve 4x4 maze and get the waypoints of the solution, which were further calibrated to the robot's coordinate system using linear interpolation. These calibrated points served as starting and ending positions for the robot's trajectory. The planned path was divided into multiple intermediate points, and an inverse kinematics model implemented in MATLAB was used to compute joint angles for each point. These angles were transmitted to the robot via TCP communication, allowing seamless execution of the planned path. It was very important to simulate and verify the digital twin of the robot's kinematics before the actual execution to assure precision and dependability. This project highlights the synergy between digital twin technology, computer vision, and robotics, demonstrating its effectiveness in real-world automation tasks.

Corresponding Author:

Chinmay Ravindra Amrutkar
Arizona State University, Tempe, AZ
Email: camrutka@asu.edu

I. INTRODUCTION

Elephant Robotics' myCobot Pro-600 is a versatile, lightweight collaborative robot designed for various commercial and industrial applications. The development of automation systems benefits significantly from the integration of robotics with digital twin technology. Digital twins provide a virtual environment to simulate, analyze, and refine robotic operations before deployment, reducing errors and enhancing precision. This project aims to connect the MyCobot Pro 600 collaborative robot with its digital twin to perform and validate motion planning tasks.

Robotic automation serves as a vital tool for addressing complex real-world problems, and navigating a maze is a classical challenge that tests the integration of image processing, kinematics, and autonomous control. This project explores an innovative approach to solving a 4x4 rectangular maze using the MyCobot Pro 600 robotic arm. Leveraging advanced technologies such as digital twins, vision systems, and socket programming, the project aims to showcase the seamless interaction between simulation and physical execution.

The methodology begins with maze detection and solution path generation using an AI Kit camera and image processing algorithms. The maze's solution path is then transformed into waypoints for robot navigation. A digital twin, developed using Simscape Multibody in MATLAB, validates the robot's movements before physical execution. The calibrated path ensures precise navigation while minimizing potential errors during real-world deployment. This project emphasizes accuracy, adaptability, and efficiency, highlighting the potential of robotics in overcoming structured challenges.

II. LAB PROCESS

1. Kinematic Diagram: We created a kinematic diagram for the MyCobot Pro 600, which visually represented the robot's structure and the relationships between its joints.

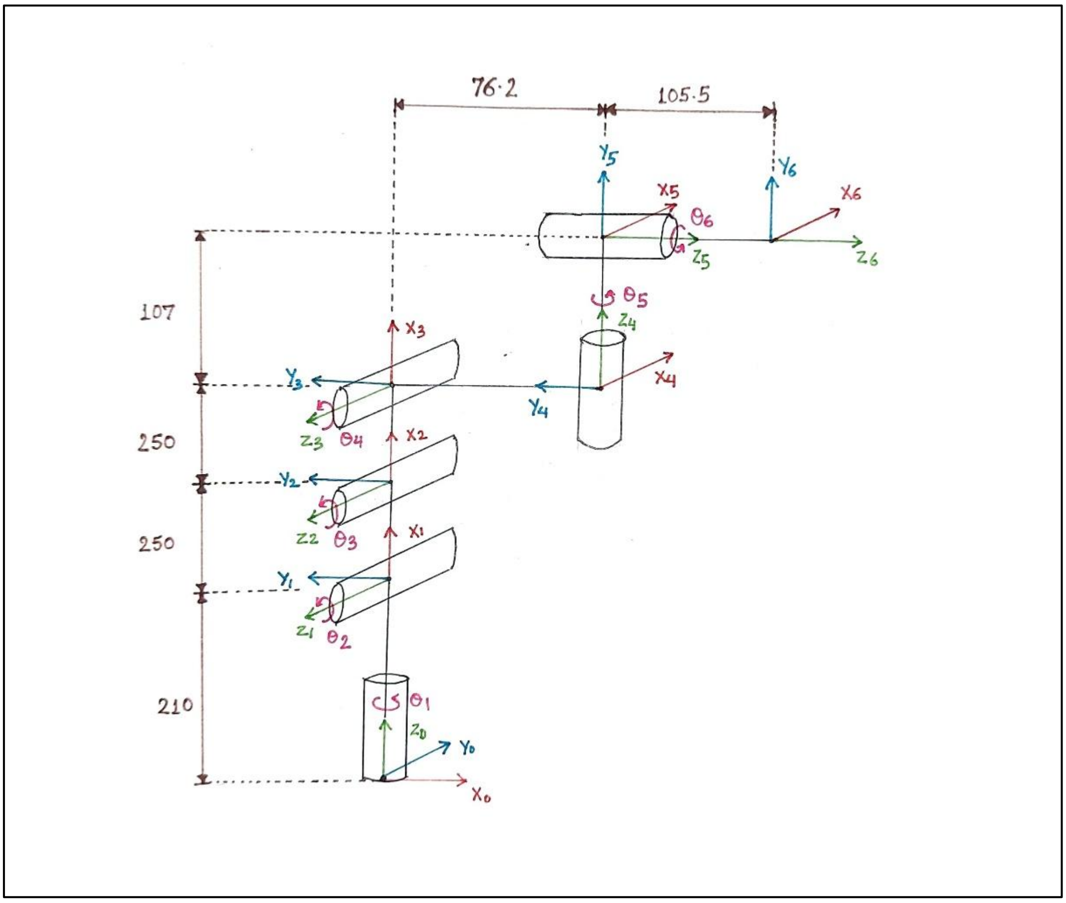

Fig 1.1 Kinematic Diagram of mycobot Pro 600

2. Homogeneous Transformation Matrices: The process began by calculating the homogeneous transformation matrices for the MyCobot Pro 600 to understand and model its kinematics.

- Joint Definition: Axes and coordinate frames were assigned to each joint of the robot.
- Rotation Matrices and Translation Vectors: Rotation matrices were derived based on the axes of rotation and respective joint angles. Translation vectors were determined using link lengths and offsets.
- Transformation Matrices: Rotation matrices and translation vectors were combined to create transformation matrices for each joint.
- Using MATLAB, the individual matrices were multiplied to obtain the overall transformation matrix ${}^{0}H_{6}$, representing the position and orientation of the end-effector relative to the base frame.
- Homogeneous Transformation Matrix for transformation from Joint 0 coordinate frame to Joint 1 coordinate frame is as follows:

$$
{}^{0}H_{1} = \begin{array}{cccc} 0 & -\cos(\theta_1) & \sin(\theta_1) & 0 \\ 0 & -\sin(\theta_1) & -\cos(\theta_1) & 0 \\ 1 & 0 & 0 & a1 \\ 0 & 0 & 0 & 1 \end{array}
$$

- Homogeneous Transformation Matrix for transformation from Joint 1 coordinate frame to Joint 2 coordinate frame is as follows:

$$^{1}H_{2} = \begin{matrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a2cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a2sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

- Homogeneous Transformation Matrix for transformation from Joint 2 coordinate frame to Joint 3 coordinate frame is as follows:

$$^{2}H_{3} = \begin{matrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a3cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a3sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

- Homogeneous Transformation Matrix for transformation from Joint 3 coordinate frame to Joint 4 coordinate frame is as follows:

$$^{3}H_{4} = \begin{matrix} 0 & -\sin(\theta_4) & \cos(\theta_4) & 0 \\ 0 & \cos(\theta_4) & \sin(\theta_4) & 0 \\ -1 & 0 & 0 & a4 \\ 0 & 0 & 0 & 1 \end{matrix}$$

- Homogeneous Transformation Matrix for transformation from Joint 4 coordinate frame to Joint 5 coordinate frame is as follows:

$$^{4}H_{5} = \begin{matrix} \cos(\theta_5) & 0 & \sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & -\cos(\theta_5) & 0 \\ 0 & 1 & 0 & a5 \\ 0 & 0 & 0 & 1 \end{matrix}$$

- Homogeneous Transformation Matrix for transformation from Joint 5 coordinate frame to Joint 6 coordinate frame is as follows:

$$^{5}H_{6} = \begin{matrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & a6 \\ 0 & 0 & 0 & 1 \end{matrix}$$

We multiplied the individual HTMs in MATLAB to obtain:

$$^{0}H_{6} = {}^{0}H_{1}{}^{1}H_{2}{}^{2}H_{3}{}^{3}H_{4}{}^{4}H_{5}{}^{5}H_{6}$$

- Further, In the MATLAB code we substituted specific numerical values for the joint angles to calculate the position and orientation of the end-effector.
- Further from $^{0}H_{6}$ we extracted and displayed the position of the end-effector from the base frame i.e. p_x , p_y *and* p_z

4. Forward and Inverse Kinematics

- **Forward Kinematics:** The homogeneous transformation matrix ($^{0}H_{6}$) was used to calculate the end-effector position (Px, Py, Pz) based on specific joint angles. These computations verified the robot's ability to achieve desired positions.
- **Inverse Kinematics:** MATLAB was utilized to develop an inverse kinematics (IK) model that computed the joint angles required to reach specified points in the workspace. This model was essential for executing motion paths.

5. Digital Twin Development

- A digital twin of the MyCobot Pro 600 was developed to simulate its behavior and validate the planned paths before physical execution.
- The STEP file of myCobot Pro 600 was imported into Fusion 360, where joints were assigned in alignment with the kinematic diagram.
- The model was exported as a URDF file using the Simscape Multibody Link tool and imported into MATLAB.
- In Simulink and a gain function was added to enhance the simulation's realism, ensuring accurate representation of the robot's motion in the digital twin.

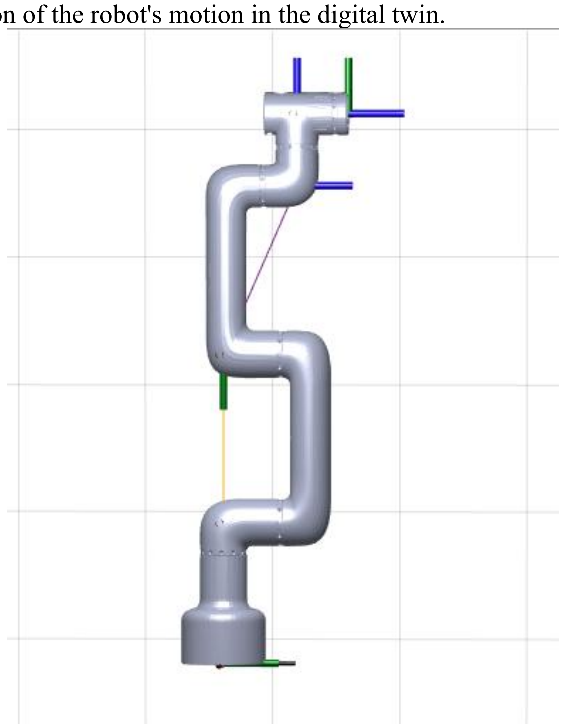

Fig 2 Digital Twin of mycobot Pro 600

6. Robot Connection and AI Kit Integration

- The MyCobot Pro 600 was connected to the computer via an Ethernet connection, enabling TCP communication using Python socket programming.
- The robot was powered on, and the TCP server was configured through the RoboFlow OS.
- An AI kit camera was integrated into the setup to provide real-time workspace visualization. This camera detected arUco markers placed within the workspace to define key points for motion planning.

7. Maze Generation and its Solution

- Used the maze generation tool to create a random 4x4 maze

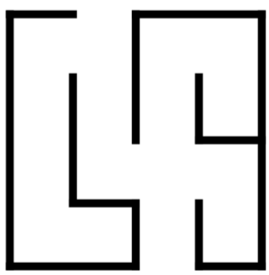

Fig 3 Random maze generated

- Captured an image of the maze using the AI Kit camera.
- Implemented an image processing algorithm to:
    - Detect the maze boundaries and paths.
    - Identify the entrance and exit points.
    - Determine the solution path using a BFS algorithm.
- Converted the solution path from pixel coordinates to real-world coordinates by mapping the maze's camera zone.

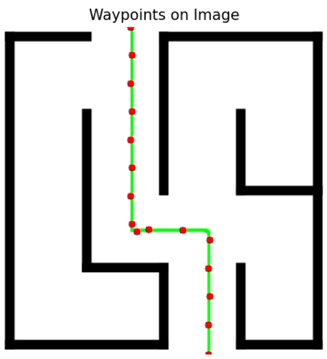

Fig 5. Solution Path and waypoints

8. Path Planning and Verification

- Generated waypoints representing straight-line segments along the solution path.
- The path was divided into several intermediate points to ensure smooth motion.
- Each point was processed through the inverse kinematics model in MATLAB to compute the corresponding joint angles.
- The planned path was tested and verified in the digital twin environment to ensure accuracy and feasibility.

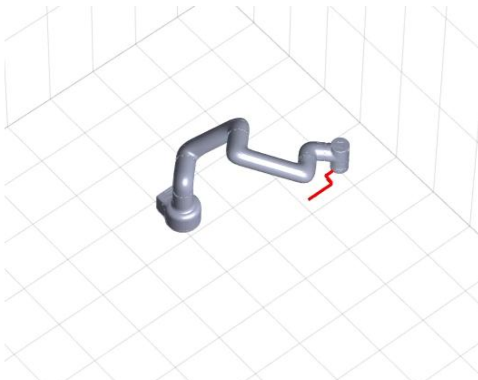

Fig 6 Path planning of mycobot Pro 600

9. Motion Execution

- The calculated joint angles from the inverse kinematics model were transmitted to the MyCobot Pro 600 using TCP communication.
- Python was used to send commands, enabling the robot to successfully execute the planned trajectory in a straight line between the two markers.

10. Comparative Analysis and Final Validation

- The end-effector positions obtained during physical execution were compared against the positions calculated through forward kinematics.
- Discrepancies, if any, were analyzed and addressed, ensuring alignment between the digital twin and the physical robot.

III. CONCLUSION

This project successfully addressed the challenge of solving a 4x4 rectangular maze using the MyCobot Pro 600 robotic arm. By integrating image processing, robotic kinematics, and digital twin simulation, the system was able to autonomously plan and execute precise navigation through the maze. The digital twin played a pivotal role in validating motion paths, reducing errors, and ensuring safe real-world deployment.

Key achievements include the development of a robust maze-solving algorithm, accurate mapping of image coordinates to the robot's workspace, and seamless execution of the solution path using efficient socket programming. These results demonstrate the potential of combining robotics and simulation technologies to solve structured challenges with adaptability and precision.

This project not only highlights the versatility of the MyCobot Pro 600 but also emphasizes the importance of simulation and systematic testing in robotic automation. The outcomes provide a strong foundation for exploring more complex scenarios, advancing the integration of digital twins and robotics in real-world applications.

IV. BIOGRAPHIES OF AUTHORS

	**Chinmay Amrutkar** is a master's student at Arizona State University pursuing Robotics and Autonomous Systems (AI). He has completed his Bachelors of Technology in Robotics and Automation at MIT World Peace University, Pune, India. He can be contacted via email at camrutka@asu.edu