# Assignment 4

**Prajjwal Dutta**[1]
**1229914740**
[1]Robotics and Autonomous systems, Arizona State University, Arizona, USA

**Article Info**

Digital Twin of MyCobot in MATLAB

***Keywords:***

Digital Twin
Alphabet Writing
Serial Communication
MyCobot280M5

**ABSTRACT** (10 PT)

This project establishes a reliable serial communication link between MATLAB and the Mycobot 280M5 robotic arm using the UART standard. Configured for bidirectional data flow, MATLAB commands control the Mycobot's movements, while the robotic arm provides real-time feedback. A user-friendly MATLAB script encapsulates the communication protocol, enabling seamless integration and intuitive control through a graphical interface. This interface enhances the Mycobot's versatility for applications in manufacturing, research, and education.

## 1. INTRODUCTION (10 PT)

The objective of this project is to establish a communication interface between the MyCobot robotic arm and MATLAB using serial communication. Our goal is to use MATLAB's serial communication features to operate the MyCobot robot with an intuitive graphical user interface that allows for fine manipulation and movement control. This project demonstrates how MATLAB and hardware can work together seamlessly to develop a variety of control algorithms and analyze data in real time.

## 2. OBJECTIVE

-Serial Communication Setup: To ensure effective data transfer and command execution, establish a stable and dependable serial communication link between MATLAB and the MyCobot robot.
-Control Algorithm Implementation: Create a reliable MATLAB control algorithm that will allow users to precisely manipulate the joints and end-effector of the MyCobot, giving them the ability to define desired positions and orientations.
- Development of Graphical User Interfaces (GUIs): Create and apply a user-friendly MATLAB GUI to give users a smooth and interactive platform to control the MyCobot robot. Users will be able to easily input precise joint angles and carry out intricate motion sequences thanks to the GUI.
-Real-Time Feedback Integration: Allow the MyCobot to provide users accurate and timely feedback about the robot's present state, joint angles, and commands that have been executed. This characteristic

## 3. METHODOLOGY

3.1. Communication Setup: Use a C type USB-to-serial converter to connect Mycobot to a laptop.

• COM11 was the port for my gadget.

3.2. MATLAB's Serial Communication

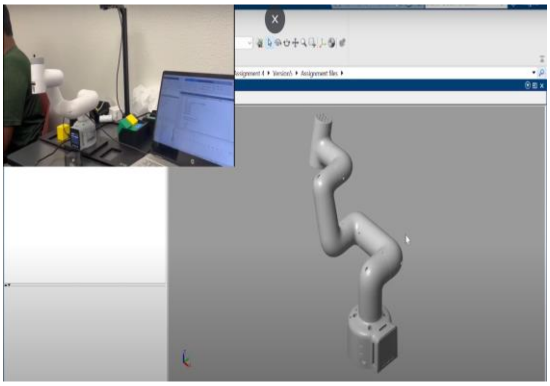
• Establish a serial port connection at a baud rate of 115200 using device "COMM11".

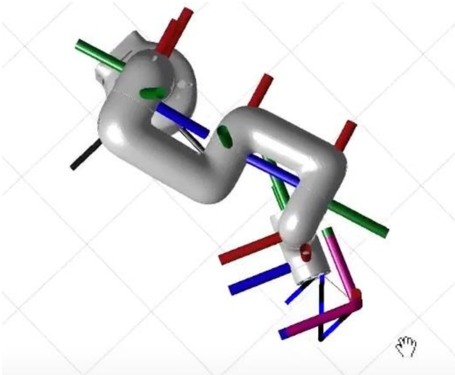
• Build a data array that contains particular values that correspond to a command frame.

• Read the device's response in six bytes after writing the data array there.

• Using a conversion function, define the coordinates (x, y, z, rx, ry, rz), speed, and send them.

• Convert coordinates, speed, and additional frame markers to create a new data array, which is then written to the device.





3.3. Forward Kinematics

```
a1= 131.56; a2 = 110.4; a3 =96; a4 =73.18; a5 = 66.39 ; a6 = 43.6;
```

```
>> Cobot_6DOF_Forward_Kinematics_Prajjwal

H01 =

[0, -cos(t1),  sin(t1),       0]
[0, -sin(t1), -cos(t1),       0]
[1,        0,        0, 3289/25]
[0,        0,        0,       1]


H12 =

[cos(t2), -sin(t2), 0, (552*cos(t2))/5]
[sin(t2),  cos(t2), 0, (552*sin(t2))/5]
[      0,        0, 1,               0]
[      0,        0, 0,               1]


H23 =

[cos(t3), -sin(t3), 0, 96*cos(t3)]
[sin(t3),  cos(t3), 0, 96*sin(t3)]
[      0,        0, 1,          0]
[      0,        0, 0,          1]
```

```
H34 =

[ 0, -sin(t4), cos(t4),        0]
[ 0,  cos(t4), sin(t4),        0]
[-1,        0,       0, 6639/100]
[ 0,        0,       0,        1]


H45 =

[cos(t5), 0,  sin(t5),       0]
[sin(t5), 0, -cos(t5),       0]
[      0, 1,        0, 3659/50]
[      0, 0,        0,       1]


H56 =

[cos(t6), -sin(t6), 0,     0]
[sin(t6),  cos(t6), 0,     0]
[      0,        0, 1, 218/5]
[      0,        0, 0,     1]
```

```
H06 =

[- cos(t6)*(cos(t5)*sin(t1) + cos(t2 + t3 + t4)*cos(t1)*sin(t5)) - sin(t2 + t3 + t4)*cos(t1)*sin(t6),   sin(t6)*(cos(t5)*sin(t1) + cos(t2 + t3 + t4)*cos(t1
[  cos(t6)*(cos(t1)*cos(t5) - cos(t2 + t3 + t4)*sin(t1)*sin(t5)) - sin(t2 + t3 + t4)*sin(t1)*sin(t6), - sin(t6)*(cos(t1)*cos(t5) - cos(t2 + t3 + t4)*sin(t1
[                                    cos(t2 + t3 + t4)*sin(t6) - sin(t2 + t3 + t4)*cos(t6)*sin(t5),                                                    cos(t2 + t3 + t
[                                                                                                0,
```

H06 =
[- cos(t6)*(cos(t5)*sin(t1) + cos(t2 + t3 + t4)*cos(t1)*sin(t5)) - sin(t2 + t3 + t4)*cos(t1)*sin(t6), sin(t6)*(cos(t5)*sin(t1) + cos(t2 + t3 + t4)*cos(t1)*sin(t5)) - sin(t2 + t3 + t4)*cos(t1)*cos(t6), cos(t2 + t3 + t4)*cos(t1)*cos(t5) - sin(t1)*sin(t5), (6639*sin(t1))/100 - (552*cos(t1)*sin(t2))/5 - (218*sin(t1)*sin(t5))/5 + (218*cos(t2 + t3 + t4)*cos(t1)*cos(t5))/5 - (3659*cos(t2 + t3)*cos(t1)*sin(t4))/50 - (3659*sin(t2 + t3)*cos(t1)*cos(t4))/50 - 96*cos(t1)*cos(t2)*sin(t3) - 96*cos(t1)*cos(t3)*sin(t2)]
[ cos(t6)*(cos(t1)*cos(t5) - cos(t2 + t3 + t4)*sin(t1)*sin(t5)) - sin(t2 + t3 + t4)*sin(t1)*sin(t6), - sin(t6)*(cos(t1)*cos(t5) - cos(t2 + t3 + t4)*sin(t1)*sin(t5)) - sin(t2 + t3 + t4)*cos(t6)*sin(t1), cos(t1)*sin(t5) + cos(t2 + t3 + t4)*cos(t5)*sin(t1), (218*cos(t1)*sin(t5))/5 - (6639*cos(t1))/100 - (552*sin(t1)*sin(t2))/5 - 96*cos(t2)*sin(t1)*sin(t3) - 96*cos(t3)*sin(t1)*sin(t2) + (218*cos(t2 + t3 + t4)*cos(t5)*sin(t1))/5 - (3659*cos(t2 + t3)*sin(t1)*sin(t4))/50 - (3659*sin(t2 + t3)*cos(t4)*sin(t1))/50]
[cos(t2 + t3 + t4)*sin(t6) - sin(t2 + t3 + t4)*cos(t6)*sin(t5),cos(t2 + t3 + t4)*cos(t6) + sin(t2 + t3 + t4)*sin(t5)*sin(t6),sin(t2 + t3 + t4)*cos(t5),96*cos(t2 + t3) + (552*cos(t2))/5 - (3659*sin(t2 + t3)*sin(t4))/50 + cos(t5)*((218*cos(t2 + t3)*sin(t4))/5 + (218*sin(t2 + t3)*cos(t4))/5) + (3659*cos(t2 +

t3)*cos(t4))/50 + 3289/25]
[0,0,0,1]

END EFFECTOR POSITIONS

1. X - Coordinate:

   (6639*sin(t1))/100 - (552*cos(t1)*sin(t2))/5 - (218*sin(t1)*sin(t5))/5 + (218*cos(t2 + t3 + t4)*cos(t1)*cos(t5))/5 - (3659*cos(t2 + t3)*cos(t1)*sin(t4))/50 - (3659*sin(t2 + t3)*cos(t1)*cos(t4))/50 - 96*cos(t1)*cos(t2)*sin(t3) - 96*cos(t1)*cos(t3)*sin(t2)

2. Y - Coordinate:

   (218*cos(t1)*sin(t5))/5 - (6639*cos(t1))/100 - (552*sin(t1)*sin(t2))/5 - 96*cos(t2)*sin(t1)*sin(t3) - 96*cos(t3)*sin(t1)*sin(t2) + (218*cos(t2 + t3 + t4)*cos(t5)*sin(t1))/5 - (3659*cos(t2 + t3)*sin(t1)*sin(t4))/50 - (3659*sin(t2 + t3)*cos(t4)*sin(t1))/50

3. Z - Coordinate:

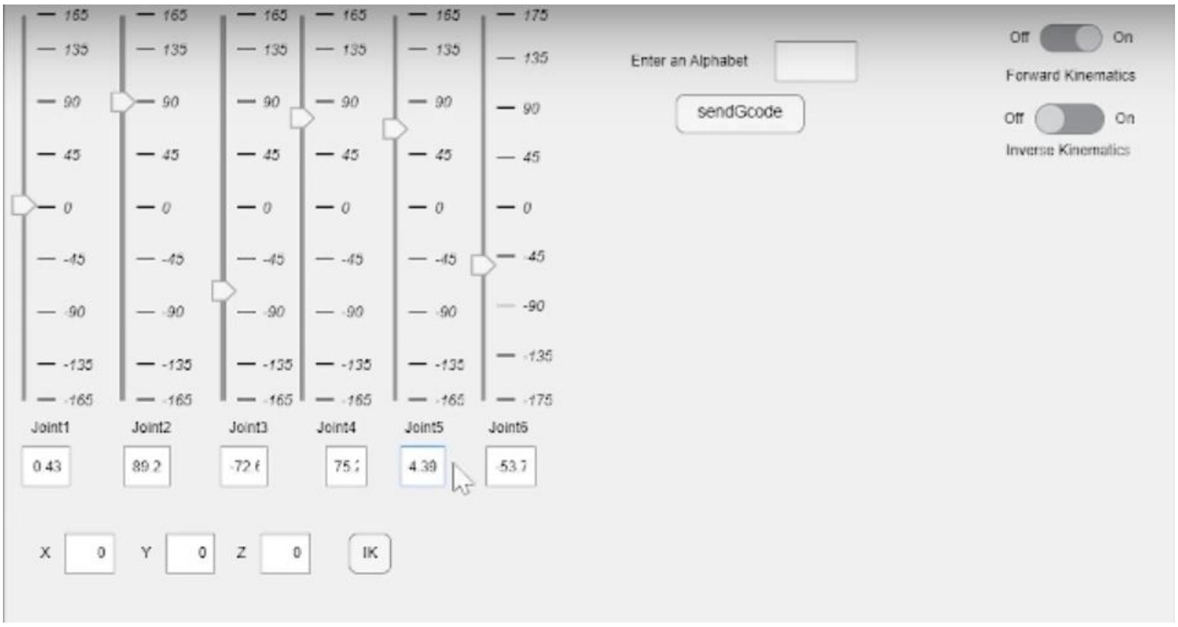
   96*cos(t2 + t3) + (552*cos(t2))/5 - (3659*sin(t2 + t3)*sin(t4))/50 + cos(t5)*((218*cos(t2 + t3)*sin(t4))/5 + (218*sin(t2 + t3)*cos(t4))/5) + (3659*cos(t2 + t3)*cos(t4))/50 + 3289/25

We can get the end effector positions from these equations for given joint angles. Using MATLAB we get →
Taking the Joint angles Provided by the grader →
$\theta 1 = 0°$; $\theta 2 = 90°$; $\theta 3 = -45°$; $\theta 4 = 0°$; $\theta 5 = 0°$; $\theta 6 = 0°$
Putting these values in the previous equations we get the end effector coordinates for these joint angles. They are →

```
px_new =

-199.198469581406638114258034993l3


py_new =

-66.39


pz_new =

282.01818090087358224201166336867
```

## 4. GUI Design

- Using sliders or direct input, the GUI can be used to display joint angles.
- Using forward kinematics, the robot's digital counterpart adjusts its position in real time to match the real robot's orientation.
- Another feature of the GUI is the ability to alternate between the digital twin's forward and inverse kinematics simulation in Simulink.
- The GUI includes a dialog box where users can enter the alphabet to be written. • It includes a robot figure that depicts the robot's movements as it writes a letter.




5. **RESULTS**

MyCobot's robotic arm was successfully controlled thanks to the application of the control algorithm and the graphical user interface (GUI) in MATLAB. Users had the ability to precisely and precisely execute motions and specify joint angles. The robot's real-time feedback verified that it interpreted user inputs as intended.

Verification of the Forward Kinematic Solution: The forward kinematic solution underwent extensive testing and verification. The real positions and orientations attained by the MyCobot were compared to the end-effector positions and orientations determined by the control algorithm. The outcomes showed a high degree of accuracy, confirming the applicability of the forward kinematic solution that the system had used.

6. **Alphabet Drawing**

- Presentation of the alphabet Z drawings executed by the simulation and the actual MyCobot robot.

7. **Implementation on the Actual Robot**

- The robot receives the alphabet's waypoints, which were determined during the simulation, via serial communication.
- To transform the numerical value into a serial communication technique that the cobot could understand, the waypoints had to be converted to hexadecimal values.

8. **Video Link**

- **https://youtu.be/eLKifsDTy-c?si=L3degN1w8z-Fpc82**

9. **Conclusion**

The seamless control and manipulation of physical hardware through an intuitive interface is demonstrated by the successful integration of MATLAB's serial communication with the MyCobot robotic arm. The effectiveness of the applied control algorithm is demonstrated by the validated forward kinematic solution, which guarantees the precise conversion of human inputs into precise robotic motions. This project demonstrates how MATLAB can be used in real-world scenarios to facilitate effective and seamless communication with physical robotic systems.