Introduction and Forward Kinematic Control

Lab Summary and Background

In this lab, you will learn and execute basic forward kinematic control of a 6 degree of freedom (DoF) robotic arm. Executing commands through the student web platform, you will perform a simple pick and place task.

Students are not required to write servo move commands but are required to understand the provided functions in the Arm_Lib_New.py Library.

Forward Kinematics:

Forward kinematics is the simplest form of control for a 6DoF robot arm. The user writes a specific angle to each actuator and the servo motors execute these angles. The position of the claw, or end effector, can be determined mathematically with these angles and the geometry of the arm. It is easiest to visualize this using a simplified example. Below is the geometry of a 2 DoF robot arm in 2-dimensional space. Throughout this lab, you will utilize forward kinematics to move individual servo motors and perform a pick and place task.

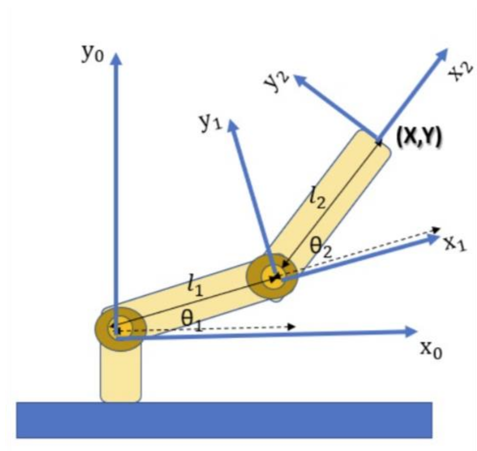


Figure 1: 2 DoF Robot Arm Geometry

Trigonometric equations can be used to describe the position of the end effector as a function of servo angles and link lengths.

$$x = x_2 = l_1 cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \qquad (1)$$

$$y = y_2 = l_1 sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \qquad (2)$$

Following a similar procedure, the position of the end effector of the robot in this lab can be calculated with link lengths and servo motor angles. Below, the numbering convention of each servo motor, angle and link is visible.

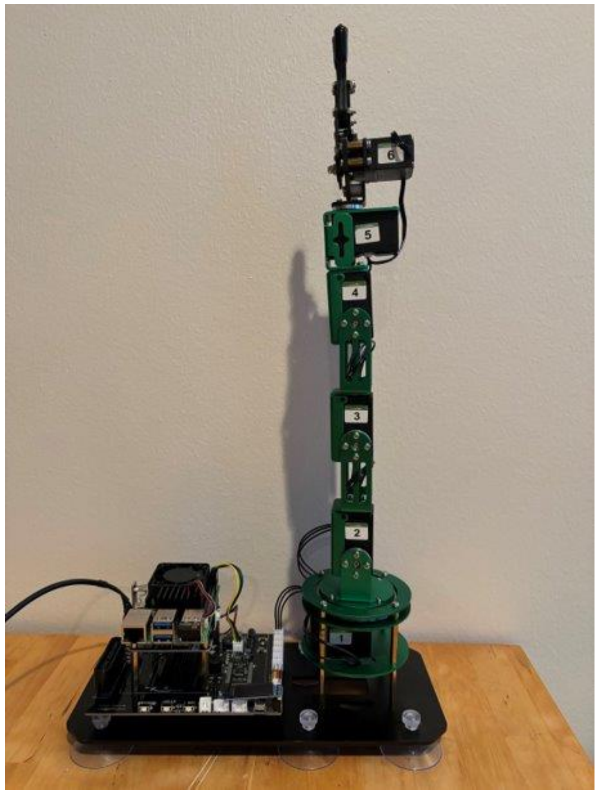


Figure 2: Servo Labels – Yahboom Robotic Arm

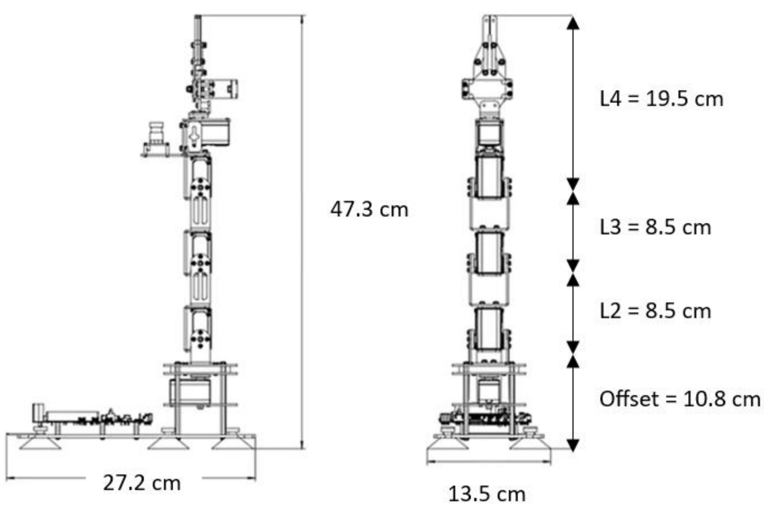


Figure 3: Robot Link Dimensions

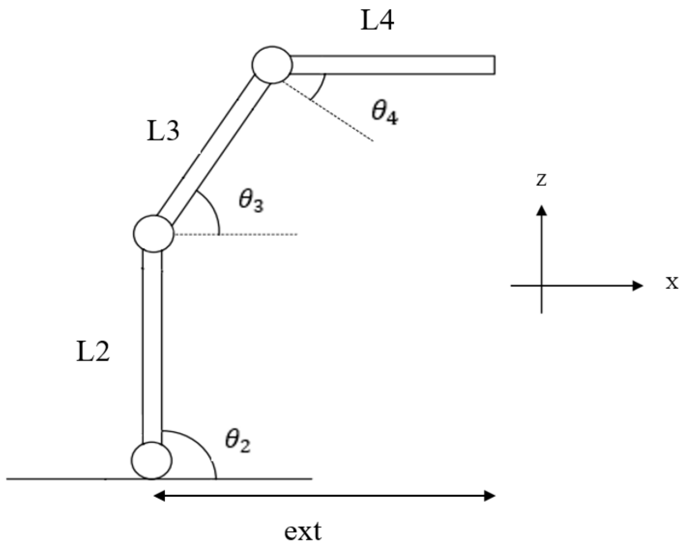


Figure 4: 2-D Servo Angle Geometry X-Z Plane

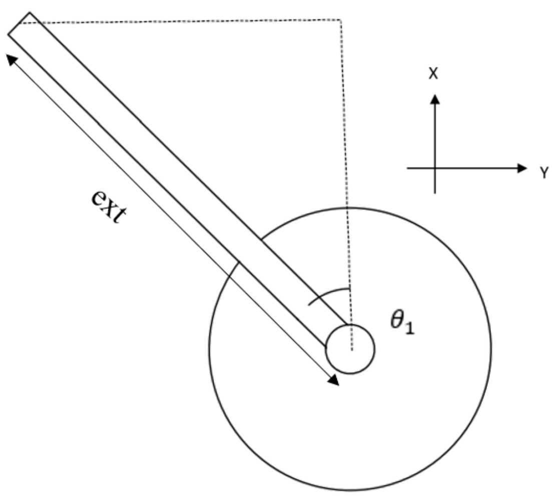


Figure 5: 2-D Servo Angle Geometry X-Y Plane

In order to easily change between coordinate frames, rotational matrices must be defined for each servo motor. Servo 1 rotates the entire arm about the Z axis.

Servo 1 rotational matrix:

$$R_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Servo motors 2-4 introduce rotation in the XZ plane.

Servo 2 rotational matrix:

$$R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_2) & -\sin(\theta_2) \\ 0 & \sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \tag{4}$$

Servo 3 rotational matrix:

$$R_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_3) & -\sin(\theta_3) \\ 0 & \sin(\theta_3) & \cos(\theta_3) \end{bmatrix}$$

Servo 4 rotational matrix:

$$R_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_4) & -\sin(\theta_4) \\ 0 & \sin(\theta_4) & \cos(\theta_4) \end{bmatrix}$$

Using these values, the complete rotational matrix for the system can be defined as follows:

$$R = R_1 \cdot R_2 \cdot R_3 \cdot R_4$$

$$R = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1)\cos(\theta_2+\theta_3+\theta_4) & \sin(\theta_1)\sin(\theta_2+\theta_3+\theta_4) \\ \sin(\theta_1) & \cos(\theta_1)\cos(\theta_2+\theta_3+\theta_4) & -\cos(\theta_1)\sin(\theta_2+\theta_3+\theta_4) \\ 0 & \sin(\theta_2+\theta_3+\theta_4) & \cos(\theta_2+\theta_3+\theta_4) \end{bmatrix}$$

Concatenating R with a final column of the X, Y, Z position equations produced a complete transformation matrix in the base to end effector frame.

$$T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1)\cos(\theta_2+\theta_3+\theta_4) & \sin(\theta_1)\sin(\theta_2+\theta_3+\theta_4) & X \\ \sin(\theta_1) & \cos(\theta_1)\cos(\theta_2+\theta_3+\theta_4) & -\cos(\theta_1)\sin(\theta_2+\theta_3+\theta_4) & Y \\ 0 & \sin(\theta_2+\theta_3+\theta_4) & \cos(\theta_2+\theta_3+\theta_4) & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Where X, Y and Z are defined using the arm geometry as:

$$ext = l_2\cos(\theta_2)+l_3\cos(\theta_2+\theta_3)+l4\cos(\theta_2+\theta_3+\theta_4) \quad (5)$$

$$Z = l_2\sin(\theta_2)+l_4\sin(\theta_2+\theta_3)+l_4\sin(\theta_2+\theta_3+\theta_4)+ offset \quad (6)$$

$$X = ext\cdot\cos(\theta_1) \quad (7)$$

$$Y = ext\cdot\sin(\theta_1) \quad (8)$$

With this, it is easy to change between the end-effector frame and the base frame. This can be done by multiplying the end-effector frame to base frame transformational matrix by the position in the end-effector frame.

$$P_{base} = T_{ee-to-base}\cdot P_{ee}$$

Similarly, to convert from an end – effector to base frame position to end-effector position, the transformation matrix inverse can be utilized.

$$P_{ee} = T^{-1}_{ee-to-base}\cdot P_{base}$$

The above X Y and Z equations are utilized in the get position function calls that you will use in this lab.

<u>Notes:</u>

- Servo motors are numbered 1-6. This number is the ID of the servo used in function calls.
  - 1: Base
  - 2: Shoulder
  - 3: Elbow

- 4: Wrist Pitch
  - 5: Wrist Roll
  - 6: Gripper
- Servo motors range from 0-180 degrees. Their rotation is defined as follows in their relative coordinate frames:
  - Servo 1: rotates entire arm about the Z axis
    - 0 degrees: 90 degrees ccw

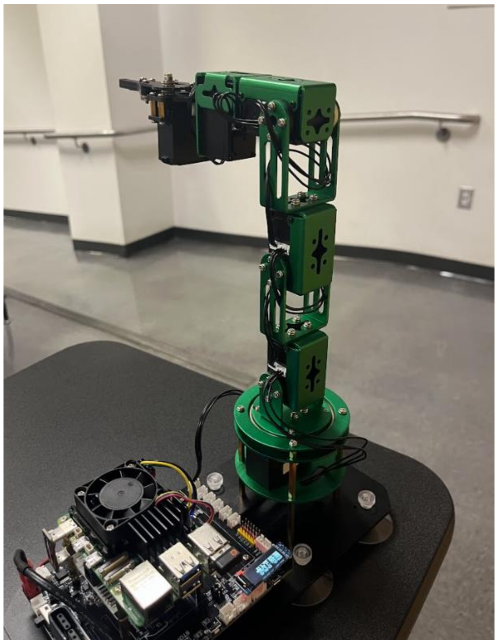


Figure 6: Servo 1 180 Deg

    - 90 degrees: home (forward facing)
    - 180 degrees: facing user

Figure 7: Servo 1 0 Deg

- Servos 2-4: Rotates in XZ Plane
  - 0 degrees: right angle forward

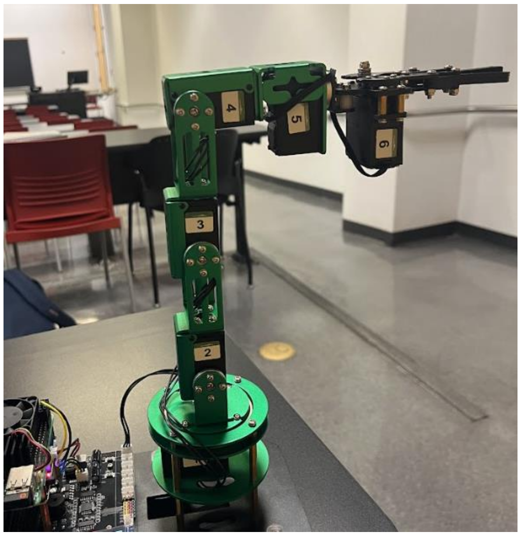


Figure 8: Servo 4 0 Deg

  - 90 degrees: home position
  - 180 degrees: right angle backwards

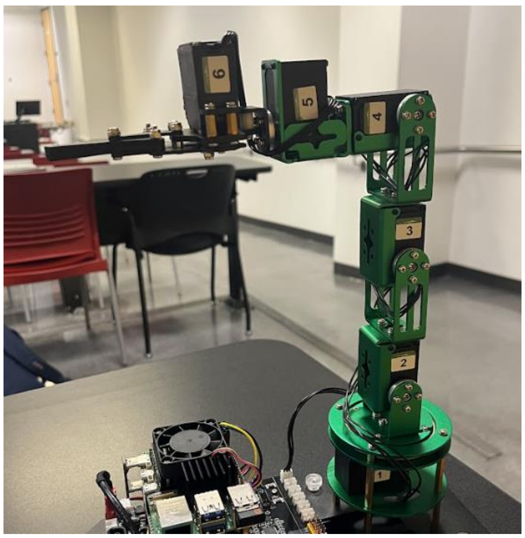


Figure 9: Servo 4 180 Deg

- Servo 5: Rotates in XZ plane
  - 0 degrees: claw oriented vertically, rotated away from user

Figure 10: Servo 6 0 Deg

- 90 degrees: home
- 180 degrees: claw oriented vertically, rotated towards user

Figure 11: Servo 6 180 Deg

- Servo 6: Rotates in XZ plane
  - 30 degrees: fully open claw

Figure 12: Servo 6 30 Deg

- 90 degrees: intermediate position

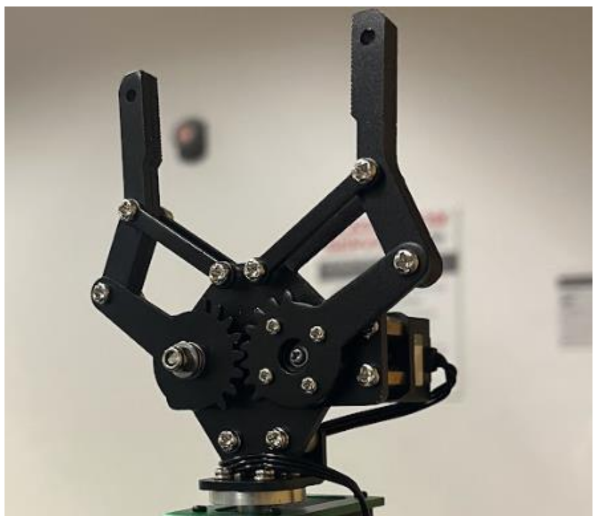


Figure 13: Servo 6 90 Deg

- 180 degrees: closed claw

- Servo write commands write execute delta angle
    - Ex) Executing servo write command of 90 degrees to servo in current position 90 degrees will produce no change.

Necessary write commands:

Arm_serial_servo_write(id, angle, time)

- Sends angle command to one servo at a time
- ID: number corresponding to servo
- Desired angle
- Time: execution time in milliseconds

Arm_serial_servo_write6(angle1, angle2, angle3, angle4, angle5, time)

- Angles 1-5: desired angle of each servo motor
- Time: execution time in milliseconds

Arm_serial_servo_write6_array(joints_array, time)

- Joints_array: 1 by 6 array of desired servo angles
- Time: execution time in milliseconds

Necessary read commands:

Arm_serial_servo_read(id)

- Returns current angle 0-180 of id

Get_pos_xyz()

- Returns array [x, y, z, ext] of arm where x, y, and z define the position in space of the tip of the end effector (assuming closed claw) and ext is the forward extension of the arm.

Pos_array()

- Returns array of all servo motor current angles 0-180 degrees [theta1 theta2 theta3 theta4 theta5 theta6]

Useful helper functions

checklim(id, angle)

- Bool function that makes sure projected position is within the mechanical limits of the arm
- Id: servo id
- Angle: desired angle command
- Called in Arm_serial_servo_write

checklim6(joints)

- Bool function that makes sure projected position is within the mechanical limits of the arm
- Called in Arm_serial_servo_write6_array(joints_array, time)

get_pos_xyz_proj(joints)

- Joints: array of servo 1-6 angles
- Returns [x, y, z, ext] where x, y, z is the theoretical position in space of the closed end effector and ext is the theoretical forward extension of the arm for angles joints

Lab Supplies

- Monitor
- Yahboom robotic arm
- Student web platform login credentials
- 4 pieces of tape

Files Needed

Intro Lab.py

Arm_Lib_New.py

Part 1: Connection and Individual Servo Movement

1. Connect to robotic arm using ssh connection
2. Manipulate servos 1, 3, and 6 individually
    - Should we have them record/ measure something as proof of completion?

- Include screenshots from web app

1. Open Student web platform on monitor
2. Input login credentials
3. Select robot corresponding to lab station
4. Open Intro_Lab.py
5. Press F1 button on robot to send to home position
    - Include pic
    - Note EStop
6. Utilize servo read commands to determine what servo angles correspond to the home position
7. Utilize get_pos_xyz to determine the position in space that corresponds to home

Home position angles:

Servo 1:	**Servo 2:**	**Servo 3:**	**Servo 4:**	**Servo 5:**	**Servo 6:**

Home position coordinates:

X:	**Y:**	**Z:**	**EXT:**

8. Send servos 2-4 individually to 0 and 180 degrees. Record the position in space for each condition. Send arm to home position before moving to next servo.

Servo 2 - 0 degrees:	**Servo 3 – 0 degrees:**	**Servo 4 – 0 degrees:**
X: **Y:** **Z:**	**X:** **Y:** **Z:**	**X:** **Y:** **Z:**

EXT:	EXT:	EXT:
Servo 2 - 180 degrees:	**Servo 3 – 180 degrees:**	**Servo 4 – 180 degrees:**
X: **Y:** **Z:** **EXT:**	**X:** **Y:** **Z:** **EXT:**	**X:** **Y:** **Z:** **EXT:**

9. Send robot home
10. Send servo 4 to 90 degrees
11. Send servo 1 to 180 degrees and 90 degrees. Record position in space for both conditions

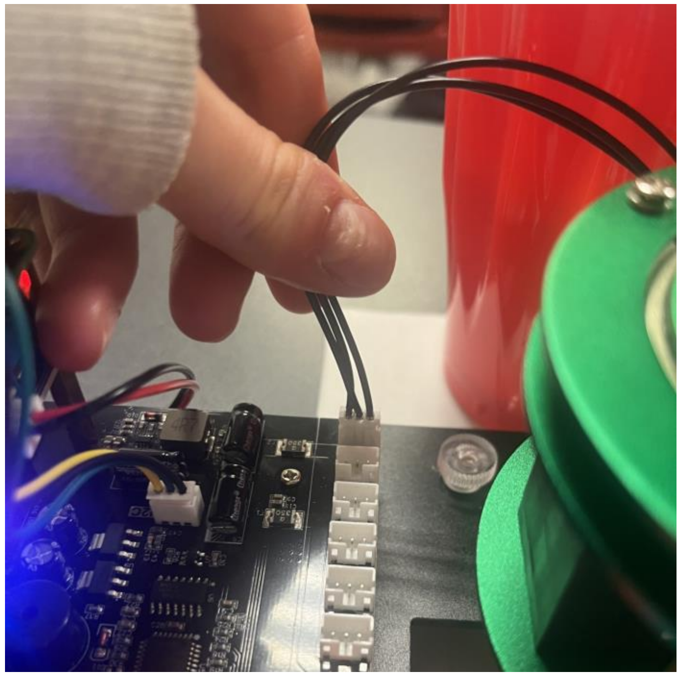
Servo 1 – 0 degrees **Servo 4 – 90 degrees**	**Servo 1 - 180 degrees** **Servo 4 – 90 degrees**
X: **Y:** **Z:** **EXT:**	**X:** **Y:** **Z:** **EXT:**

Part 2: Pick and Place with Forward Kinematics

Objective: Move robotic arm to a pre-determined point using forward kinematics.

- From home position, pick up a block, return home, place block in second position, return home.

1. Use the provided tape to create x's on two locations within the mechanical range of the arm. This defines position 1 and position 2.
2. You must determine sets of servo angles necessary to pick up a block in position 1, return to home position, place the block in position 2, and return home.
    a. To determine servo angles, remove servo bus connection and manipulate the robot manually



   b. After manual manipulation of servo motors, plug servo bus in and execute read commands

3. Once servo angles are determined, write code to perform the following:
   a. Start in home position
   b. Open gripper (servo 6) to 180 degrees
   c. Execute servo positions determined to pick up block in position 1
   d. Close grippers (servo 6) to XXX degrees
   e. Return to home position. Do not further close grippers
   f. Execute servo positions determined to place block in position 2
   g. Open grippers (servo 6) to 180 degrees)
   h. Return to home position
   i. Multiple iterations of manual manipulation/angle adjustments may be necessary to find effective servo angles
   j. Record final angle sets and xyz position in space of end effector in positions 1 and 2

	Position 1:	**Position 2:**	**X, Y, Z, EXT position 1:**
Servo 1:			
Servo 2:			
Servo 3:			**X, Y, Z, EXT position 2:**
Servo 4:			
Servo 5:			

Successful execution of task observed by lab monitor:

Lab Monitor Initials: Date:

Discussion Questions

1. What is a rotary potentiometer and how does it give us angle feedback in our servo motors?
2. What is forward kinematics and how does it differ from inverse kinematics when moving the robotic arm?
3. In terms of end position and end effector orientation, how do servo motors 1, 2 and 3 differ from servo motors 4, 5 and 6?
4. What are a few important physical measurements on the robotic arm and why are they important when performing forward kinematic calculations?