ELEC0129 – Final Report – Team 14

Table of Contents

Section 1: Forward Kinematics

This section of the report will focus on calculating the Forward Kinematics of the robot. The goal would be to calculate the end effector of the robot given a set of join angles.

Firstly, we will define the Denavit-Hartenverg Parameters, which include the robots Link Lengths, a_{i-1}, Link Twists, α_{i-1}, Joint Angles, θ_i and Link Offsets, d_i. These are shown in table 1.1.

Before drawing the axes of the robot, we will first determine the direction in which the robot moves that would correspond to a +ve angle change. This is done by changing the angles in the Arduino from $0°$ to $+30°$, then observing the direction in which the robot moves. This is done for each joint angle and the results are summed up in the figures below.

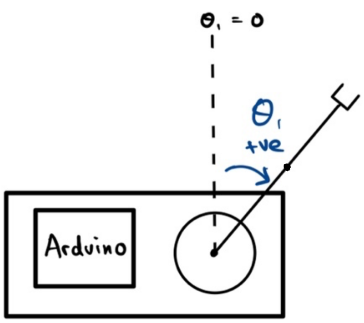


Figure 1.1a: Top View of the robot and orientation of θ_3

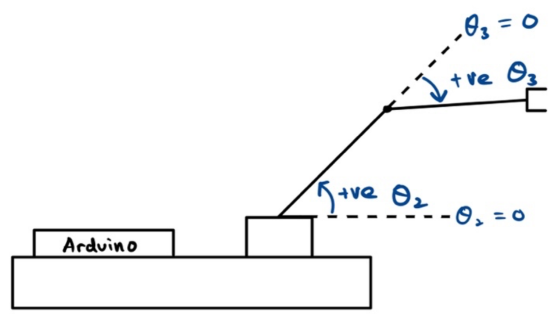


Figure 2.1b: Side View of the robot and orientation of θ_2 & θ_3

Next, we can draw the axes of the robot as shown in the scehmatic in figure 1.2.

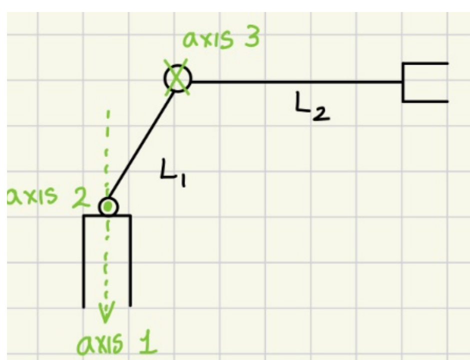


Figure 1.2: Axes of the robot

Based on figure 1.2, we can measure the length of the robot's arm L_1 and L_2 $100mm$ and $170mm$ respectively. Following this, we can add the mutual perpendicular line to the diagram and obtain figure 1.3.

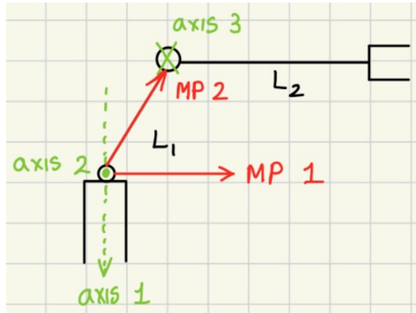


Figure 1.3 Drawing of the mutual prependicular line

Based on figure 1.3, we can attach the frames {1} and {2} to the robot. The frames are based on the following rules:

- Z_i pointing along the i-th joint
- X_i pointing along the mutual perpendicular

After that, we attach the frame {0} such that it would match {1} when the first joint variable, θ_1, is equal to zero. The result after applying the frames to the robot is shown in figure 1.4.

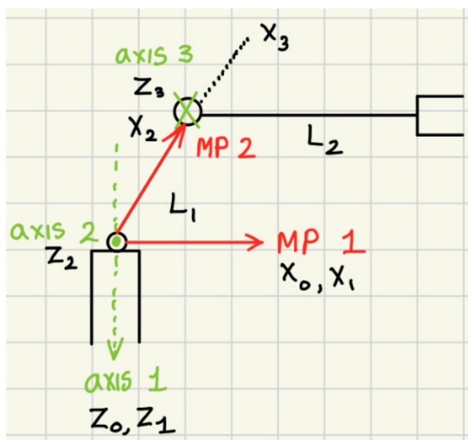


Figure 1.4 Drawing of the axis

By using the frames defined in figure 1.4, we can define the Denavit-Hartenverg (DH) Parameters as:

- Link length a_{i-1} (mm): Distance between the z_{i-1} and z_i along the x_{i-1}
- Link twist α_{i-1} $(°)$: Angle between the z_{i-1} and z_i along the x_{i-1}
- Joint angle θ_i $(°)$: Angle between x_i and x_{i+1} along the z_{i+1}
- Link Offsets, d_i (mm): Distance between x_i andx_{i+1} along the z_{i+1}

Now, we can measure the individual DH parameters as shown below:

For $i = 1$:

- a_0 is the distance between the Z_0 and Z_1 along the X_0, therefore $a_0 = 0$
- α_0 is the angle between the Z_0 and Z_1 along X_0, therefore $\alpha_0 = 0$
- θ_1 is the angle between the X_0 and X_1 along Z_1. Since joint 1 is a revolute joint, θ_1 is a variable.
- d_1 is the distance between X_0 and X_1 along Z_1, therefore $d_1 = 0$

For $i = 2$:

- a_1 is the distance between the Z_1 and Z_2 along the X_1, therefore $a_1 = 0$
- α_1 is the angle between the Z_1 and Z_2 along the X_1, therefore $\alpha_1 = -90$
- θ_2 is the angle between the X_1 and X_2 along Z_2. Since joint 2 is a revolute joint, θ_2 is a variable.
- d_2 is the distance between X_1 and X_2 along Z_2, therefore $d_2 = 0$

For $i = 3$:

- a_2 is the distance between the Z_2 and Z_3 along X_2, therefore $a_2 = 100mm$ as $L_1 = 100mm$
- α_2 is the angle between the Z_2 and Z_3 along X_2, therefore $\alpha_2 = 180°$
- θ_3 is the angle between the X_2 and X_3 along Z_3. Since joint 3 is a revolute joint, so θ_3 is a variable.
- d_3 is the distance between X_2 and X_3 along the Z_3, therefore $d_3 = 0$

These measurements are summed up in table 1.1 shown below:

Denavit-Hartenverg (DH) Parameters

Link Lengths, a_{i-1} (mm)	Link Twists, α_{i-1} (°)	Joint Angles, θ_i (°)	Link Offsets, d_i (mm)
$a_0 = 0$	$\alpha_0 = 0$	$\theta_1 = Variable$	$d_1 = 0$
$a_1 = 0$	$\alpha_1 = -90$	$\theta_2 = Variable$	$d_2 = 0$
$a_2 = 100$	$\alpha_2 = -180$	$\theta_3 = Variable$	$d_3 = 0$

Table 1.1: Denavit-Hartenverg (DH) Parameters

End effector

The end effector of the robot represents the relative position of the end tip to joint 3 to the base. This can be calculated using a series of transformation matrix, which we shall derive in this section.

The end effector of the robot represents the relative position of the end tip of the robot with respect to frame {3}. According to the figure, the end effector is given as:

$$^{3}P = \begin{bmatrix} L2 \\ 0 \\ 0 \end{bmatrix}$$

The transformation matrix of the neighboring frames of the robot is shown below:

$$^{i-1}_{\ \ i}T = \begin{bmatrix} c\theta_1 & -s\theta_i & 0 & \alpha_{i-1} \\ c\alpha_{i-1}s\theta_i & c\alpha_{i-1}c\theta_i & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\alpha_{i-1}s\theta_i & s\alpha_{i-1}c\theta_i & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Based in this equation, we can derive the transformation matrix $^{0}_{1}T,\ ^{1}_{2}T,\ ^{2}_{3}T$ as below:

$$^{0}_{1}T = \begin{bmatrix} cos\theta 1 & -sin\theta 1 & 0 & 0 \\ sin\theta 1 & cos\theta 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

$$^{1}_{2}T = \begin{bmatrix} cos\theta 2 & -sin\theta 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -sin\theta 2 & -cos\theta 2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

$$^{2}_{3}T = \begin{bmatrix} cos\theta 3 & sin\theta 3 & 0 & 100 \\ -sin\theta 3 & -cos\theta 3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

The transformation matrix from 0 to 3 $^{0}_{3}T$ is the multiple of the $^{0}_{1}T,\ ^{1}_{2}T$ and $^{2}_{3}T$

$$^{0}_{3}T = {}^{0}_{1}T * {}^{1}_{2}T * {}^{2}_{3}T$$

Therefore, the transformation matrix $^{0}_{3}T$ will be:

$$^{0}_{3}T = \begin{bmatrix} \cos\theta 1 * \cos(\theta 2 - \theta 3) & -cos\theta 1 * \sin(\theta 3 + \theta 2) & sin\theta 1 & cos\theta 1 * cos\theta 2 * 100 \\ sin\theta 1 * \cos(\theta 2 + \theta 3) & sin\theta 1 * \sin(\theta 3 + \theta 2) & -cos\theta 1 & sin\theta 1 * cos\theta 2 * 100 \\ \sin(\theta 3 - \theta 2) & -\cos(\theta 2 - \theta 3) & 0 & -sin\theta 2 * 100 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.4)$$

Since we know the position of the end-effector relative to frame {3} is:

$$ {}^{3}P = \begin{bmatrix} 170 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{1.5} $$

The final position of the end-effector is the multiple of the P and the ${}^{0}_{3}T$

$$ {}^{0}_{3}P = {}^{0}_{3}T * {}^{3}P $$

$$ {}^{0}_{3}P = \begin{bmatrix} 170 * cos\theta 1 * \cos(\theta 2 - \theta 3) + cos\theta 1 * cos\theta 2 * 100 \\ 170 * sin\theta 1 * \cos(\theta 2 + \theta 3) + sin\theta 1 * cos\theta 2 * 100 \\ 170 * \sin(\theta 3 - \theta 2) - sin\theta 2 * 100 \\ 1 \end{bmatrix} \tag{1.6} $$

After that, we implemented the code into MATLAB which gave position of the end effector with respect to frame {0} for a set of input angles. Then, we entered the angles into the Arduino and measured the real position in which the robot is moved to with respect to frame {0}. The results are shown in table 1.2.

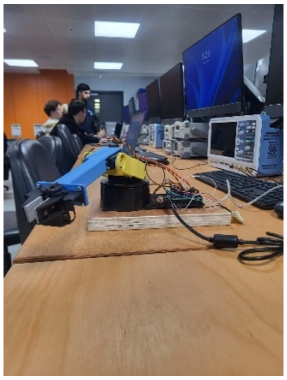
Joint 1 Angle $(\theta_1)(°)$	Joint 2 Angle $(\theta_2)(°)$	Joint 3 Angle $(\theta_3)(°)$	Calculated Cartesian Coordinates (mm)	Measured coordinates(mm)
30	30	30	$\begin{pmatrix} 222.23 \\ 128.00 \\ -50 \end{pmatrix}$	$\begin{pmatrix} 215 \\ 124 \\ -45 \end{pmatrix}$
45	45	45	$\begin{pmatrix} 170.21 \\ 170.21 \\ -70.71 \end{pmatrix}$	$\begin{pmatrix} 170 \\ 165 \\ -66 \end{pmatrix}$
60	60	60	$\begin{pmatrix} 110 \\ 190.53 \\ -86.60 \end{pmatrix}$	$\begin{pmatrix} 114 \\ 180 \\ -83 \end{pmatrix}$
90	90	90	$\begin{pmatrix} 0 \\ 170.00 \\ -100 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 170 \\ -100 \end{pmatrix}$
120	120	120	$\begin{pmatrix} -60.75 \\ 105.22 \\ -86.60 \end{pmatrix}$	$\begin{pmatrix} -57 \\ 103 \\ -90 \end{pmatrix}$

Table 1.2: Overview of calculated results against measured results

Overall, we can see that the measured coordinates closely match the theoretical cartesian coordinates we calculated. Although we see some small differences between the actual measured position of the end effector and the calculated position, these differences are marginal and within ±*5mm*. These measured differences can be accounted for by parallax error when measuring end effectors position by hand. Additionally, the end effector of the robot is an imaginary floating point in the middle f the gripper. Therefore, the perception of the end effector point would be different for every measurement, contributing to the error.

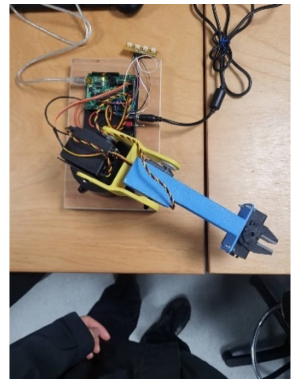
The figures below show the robots position for different sets of input angles.

Input angle: $\theta_1 = 30°$, $\theta_2 = 30°$, $\theta_3 = 30°$



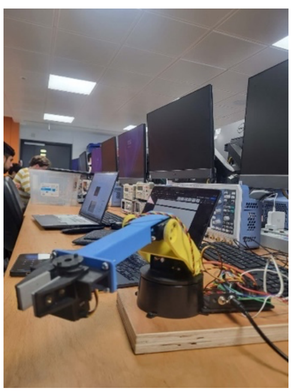
Picture 1.5 view looking into x-axis

Picture 1.6 view ooking into from y-axis



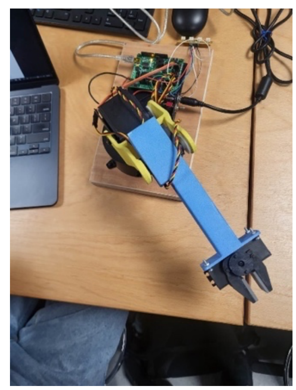
Picture 1.7 view looking into from z-axis

Input angle: $\theta_1 = 45°$, $\theta_2 = 45°$, $\theta_3 = 45°$



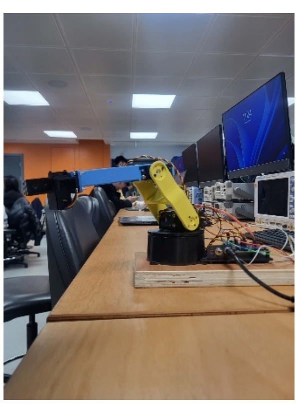
Picture 1.8 view looking into x-axis

Picture 1.9 view looking into y-axis

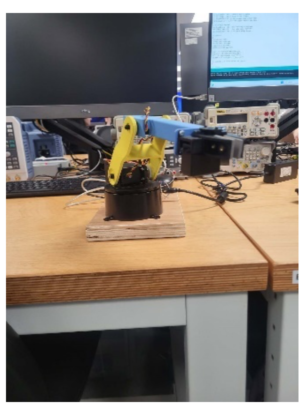


Picture 1.10 view looking into z-axis

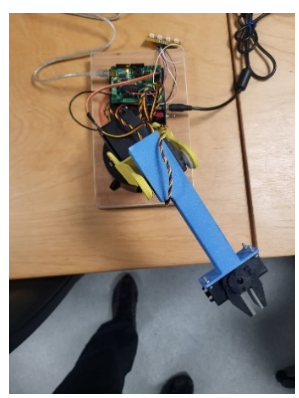
Input angle: $\theta_1 = 60°$, $\theta_2 = 60°$, $\theta_3 = 60°$



Picture 1.11 view looking into x-axis

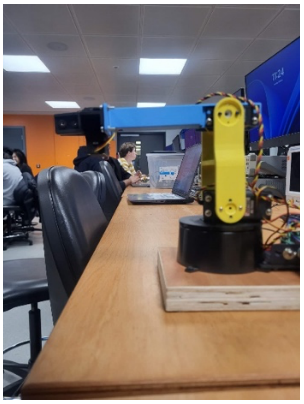


Picture 1.12 view looking into y-axis

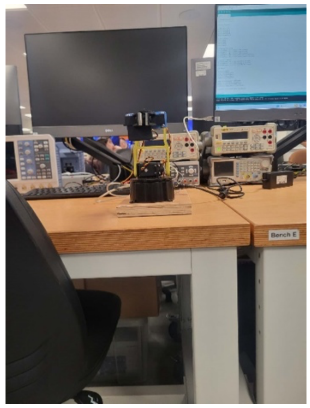


Picture 1.13 view looking into z-axis

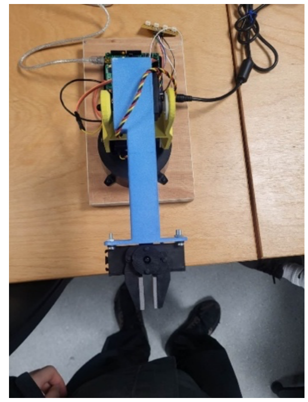
Input angle: $\theta_1 = 90°,\ \theta_2 = 90°,\ \theta_3 = 90°$



Picture 1.14 view looking into x-axis



Picture 1.15 view looking into y-axis

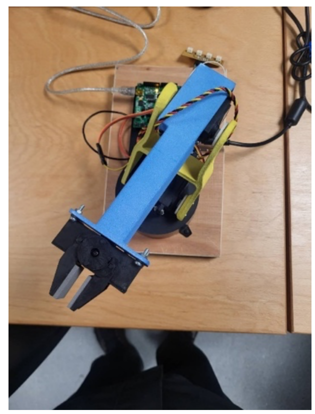


Picture 1.16 view looking into z-axis

Input angle: $\theta_1 = 120°,\ \theta_2 = 120°,\ \theta_3 = 120°$

Picture 1.17 view looking into x-axis

Picture 1.18 view looking into y-axis



Picture 1.19view looking into z-axis

Section 2: Inverse Kinematics

This section of the report will focus on calculating the inverse kinematics of the robotic arm. The end goal is to be able to calculate the angles of joints 1 to 3, given that we know the length of the robot's arm and the cartesian position, (x, y, z) of the grippers gripping point with respect to the robot's base frame. We first start by establishing the angles and the lengths of the robot by drawing the diagrams as shown below.

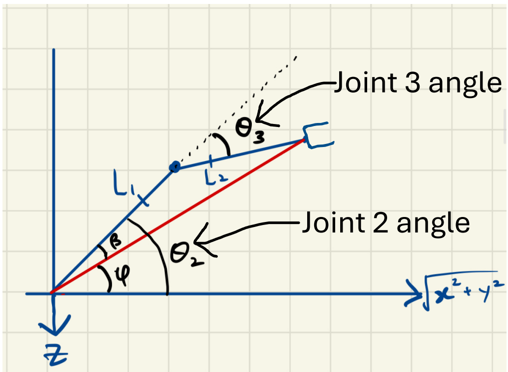


Figure 2.1: Side View (Case 1, elbow up, non inversed base)

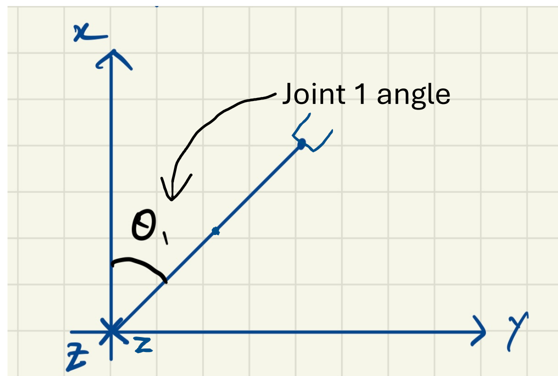


Figure 2.2: Top View (Case 1 & 2, non-inversed base)

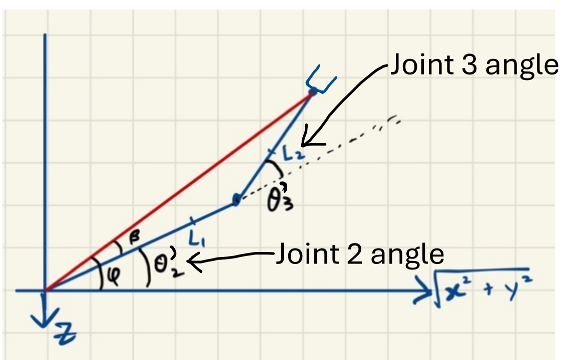


Figure 2.3: Side View (Case 2, elbow down, non-inversed base)

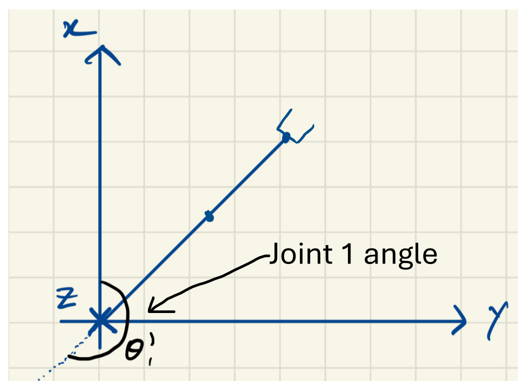


Figure 2.4: Top view (Case 3 & 4, Inversed Base)

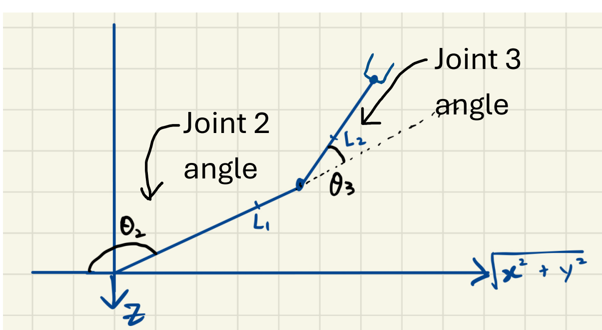


Figure 2.5: Side (Case 4, elbow down, inversed base)

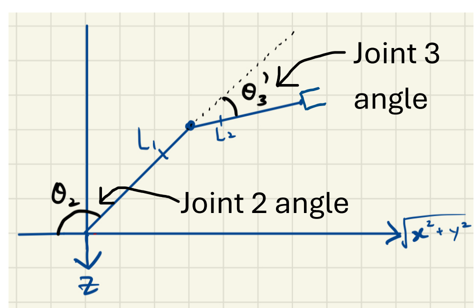


Figure 2.6: Side view (Case 1, elbow up, inversed base)

Derivation of Inverse Kinematics: Geometric Approach

Now, we will calculate the inverse kinematics of the robot using the geometric approach.

We will first calculate the solution for joint angle 1 (θ_1), which can be given by 2 values depending on the position of the robot (Inversed or non-inversed base).

We will first consider the case for the non-inversed base shown in figure 2.2, which can be calculated using simple trigonometry as:

$$\theta_1 = \cos^{-1}\left(\frac{x}{\sqrt{x^2+y^2}}\right) \tag{2.1}$$

Next, we will consider the case for the robot with inversed base, as shown in figure 2.4. From this, we can deduce that the expression for θ_1' will be given by:

$$\theta_1' = \cos^{-1}\left(\frac{x}{\sqrt{x^2+y^2}}\right) + 180° \tag{2.2}$$

Moving forward, we will now calculate the angle θ_2, which can have two values depending on the position of the robot (elbow up or elbow down).

We will first consider the elbow up solution, which is represented in figure 2.1.

Next, we can calculate the equations for joint angle 2 (θ_2), which is defined as the angle from L_2 to the x-y plane. To calculate θ_2 for elbow up solution, we can write θ_2 as:

$$\theta_2 = \varphi + \beta \tag{2.3}$$

However, for the elbow down solution, θ_2 is instead written as:

$$\theta_2' = \varphi - \beta \tag{2.4}$$

Using the same cosine equation, we used to derive equation 2.1, we can calculate the angle β as:

$$L_2^2 = L_1^2 + \left(\sqrt{x^2+y^2+z^2}\right)^2 - 2L_1\sqrt{x^2+y^2+z^2}\cos(\beta)$$

$$\beta = \cos^{-1}\left(\frac{L_1^2 + \left(\sqrt{x^2+y^2+z^2}\right)^2 - L_2^2}{2L_1\sqrt{x^2+y^2+z^2}}\right)$$

Using trigonometry, we can calculate φ as:

$$\varphi = \tan^{-1}\left(\frac{-z}{\sqrt{x^2+y^2}}\right)$$

Now that we have derived the equation for φ and β, we can substitute them into equation (2.3) and obtain the elbow up solution for θ_2 as:

$$\theta_2 = \cos^{-1}\left(\frac{L_1^2 + \left(\sqrt{x^2+y^2+z^2}\right)^2 - L_2^2}{2L_1\sqrt{x^2+y^2+z^2}}\right) + \tan^{-1}\left(\frac{-z}{\sqrt{x^2+y^2}}\right) \qquad (2.5)$$

To obtain the elbow down solution for joint angle 2 (θ_2), we substitute the expression for β and φ into equation (2.4) and obtain:

$$\theta_2' = -\cos^{-1}\left(\frac{L_1^2 + \left(\sqrt{x^2+y^2+z^2}\right)^2 - L_2^2}{2L_1\sqrt{x^2+y^2+z^2}}\right) + \tan^{-1}\left(\frac{-z}{\sqrt{x^2+y^2}}\right) \qquad (2.6)$$

Finally, we will now calculate the value for joint angle 3 (θ_3), which will also have two values depending on the position of the robot (elbow up or elbow down).

We will first consider the elbow up solution, for which the value of θ_3 can be calculated using the cosine rule as:

$$x^2 + y^2 + (-z)^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(180 - \theta_3)$$

Rearranging this equation, we will get:

$$\theta_3 = \cos^{-1}\left(\frac{x^2+y^2+z^2-L_1^2-L_2^2}{2L_1L_2}\right) \qquad (2.7)$$

Using symmetry for the elbow down case, we can find the value for θ_3' as:

$$\theta_3' = -\cos^{-1}\left(\frac{x^2+y^2+z^2-L_1^2-L_2^2}{2L_1L_2}\right) \qquad (2.8)$$

At this stage, we have already derived the equations which would give us $\theta_1, \theta_2\ and\ \theta_3$ for one set of input coordinates, $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$.

By considering the multiplicities of solutions, we would obtain 4 different sets of angles for a given set of coordinates (4 cases). The equations for the joint angles for each cases are given by the equations shown in Table 2.1:

Case	Joint 1 angle Equation	Joint 2 angle Equation	Joint 3 angle Equation
Case 1: Elbow up, non-inversed base	θ_1	θ_2	θ_3
Case 2: Elbow down, non-inversed base	θ_1	θ_2'	θ_3'
Case 3: Elbow up, Inversed base	θ_1'	$180 - \theta_2$	θ_3'
Case 4: Elbow down, inversed base	θ_1'	$180 - \theta_2'$	θ_3

Table 2.1: Equations used to find the corresponding angles for each case

The next stage involves the implementation of the code into MATLAB, for which the code can be found in the attached file called "InverseKinematics.m". Using MATLAB, we calculated the 4 possible sets of angles of which the robot can reach the end point given by coordinates (x,y,z). Table 2.2 shows the result for case for a set input coordinate.

Input Coordinates (mm)	Cases:	Calculated Joint 1 Angle (°)	Calculated Joint 2 Angle (°)	Calculated Joint 3 Angle (°)
$\begin{pmatrix} 90 \\ 90 \\ -90 \end{pmatrix}$	Case 1 Case 2 Case 3 Case 4	45 45 225 225	115.2 -44.8 64.8 224.8	115.4 -115.4 -115.3 115.4
$\begin{pmatrix} 90 \\ 150 \\ -90 \end{pmatrix}$	Case 1 Case 2 Case 3 Case 4	59 59 239 239	87 -32.6 93 212.6	90.3 -90.3 -90.3 90.3
$\begin{pmatrix} 60 \\ 150 \\ -60 \end{pmatrix}$	Case 1 Case 2 Case 3 Case 4	68.2 68.2 248.2 248.2	92.1 -51.3 87.9 231.3	105.7 -105.7 -105.7 105.7
$\begin{pmatrix} -90 \\ 100 \\ -140 \end{pmatrix}$	Case 1 Case 2 Case 3 Case 4	132 132 312 312	107.2 -14.9 72.8 194.9	92 -92 -92 92
$\begin{pmatrix} -60 \\ 60 \\ -60 \end{pmatrix}$	Case 1 Case 2 Case 3 Case 4	135 135 315 315	148.2 -77.7 31.8 257.7	145.7 -145.7 -145.7 145.7

Table 2.2: Table of Results with calculated and measured results

The next step involves implementing the equations for each angle into the Arduino. Due to the physical limitations of the robot, we could only implement the equations for case 1. To check for the accuracy of our equations, we calculated the angles for each joint for 5 sets of input coordinate, then entered the angles into the Arduino. Then, we measure the obtained coordinates and compare it with our input coordinates.

Input Coordinates (mm)	Calculated Joint 1 Angle (°)	Calculated Joint 2 Angle (°)	Calculated Joint 3 Angle (°)	Measured Cartesian Coordinates (mm)
$\begin{pmatrix} 90 \\ 90 \\ -90 \end{pmatrix}$	45	115.2	115.4	$\begin{pmatrix} 88 \\ 92 \\ -90 \end{pmatrix}$
$\begin{pmatrix} 90 \\ 150 \\ -90 \end{pmatrix}$	59	87	90.3	$\begin{pmatrix} 91 \\ 149 \\ -89 \end{pmatrix}$
$\begin{pmatrix} 60 \\ 150 \\ -60 \end{pmatrix}$	68.2	92.1	105.7	$\begin{pmatrix} 55 \\ 153 \\ -65 \end{pmatrix}$
$\begin{pmatrix} -90 \\ 100 \\ -140 \end{pmatrix}$	132	107.2	92	$\begin{pmatrix} -90 \\ 105 \\ -130 \end{pmatrix}$
$\begin{pmatrix} -60 \\ 60 \\ -60 \end{pmatrix}$	135	148.2	145.7	$\begin{pmatrix} -65 \\ 65 \\ -58 \end{pmatrix}$

Table 2.3: table of results

Observations

The figures shown below are our observations for different input coordinates. Refer to the axis drawn in figure 2.7, 2.8, 2.9 for reference.

Input Coordinates: $\begin{pmatrix} 90 \\ 90 \\ -90 \end{pmatrix}$

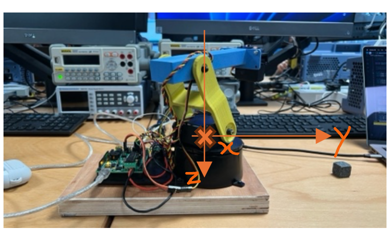


Figure 2.7: View looking into x-axis

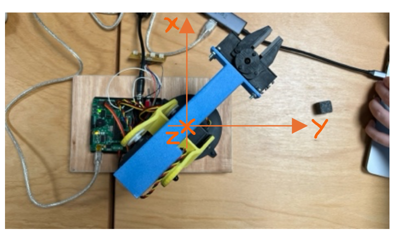


Figure 2.8: View looking into z-axis

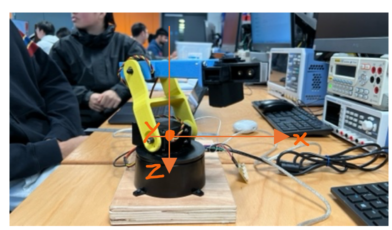


Figure 2.9: View looking into y-axis

Input Coordinates: $\begin{pmatrix} 90 \\ 150 \\ -90 \end{pmatrix}$

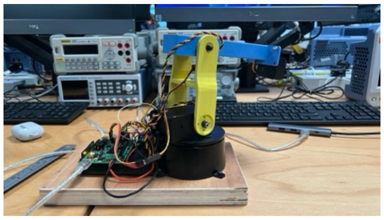


Figure 2.10: View looking into x-axis

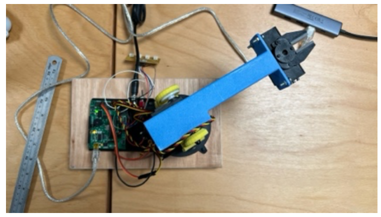


Figure 2.11: View looking into z-axis

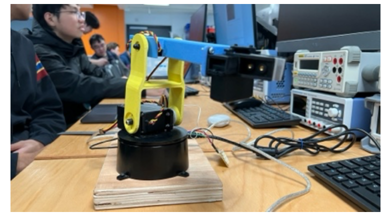


Figure 2.12: View looking into y axis

Input Coordinates: $\begin{pmatrix} 60 \\ 150 \\ -60 \end{pmatrix}$

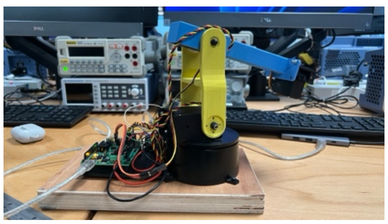


Figure 2.13: View looking into x-axis

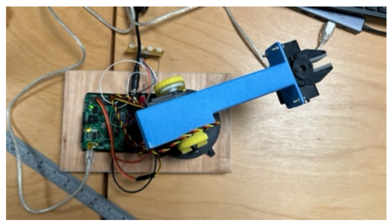


Figure 2.14: View looking into z-axis


Figure 2.15: View looking into y axis

Input Coordinates: $\begin{pmatrix} -90 \\ 100 \\ -140 \end{pmatrix}$

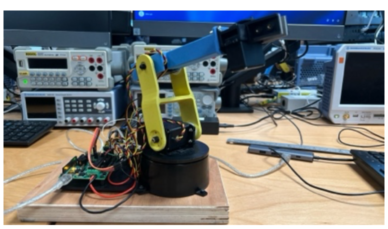


Figure 2.16: View looking into x-axis

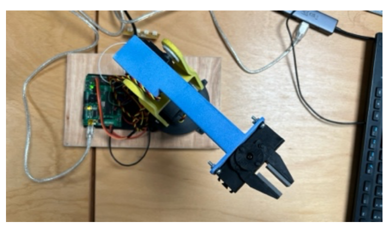


Figure 2.17: View looking into z-axis

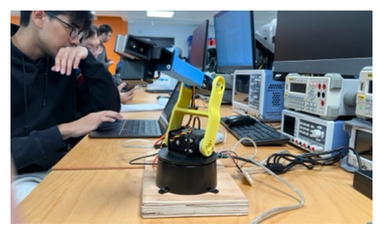


Figure 2.18: View looking into y axis

Input Coordinates: $\begin{pmatrix} -60 \\ 60 \\ -60 \end{pmatrix}$

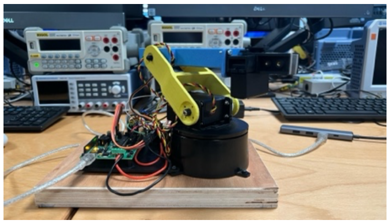


Figure 2.19: View looking into x-axis

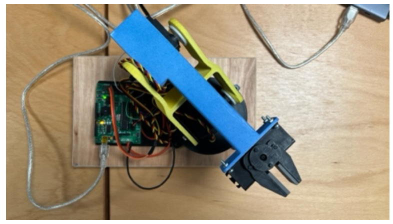


Figure 2.20: View looking into z-axis

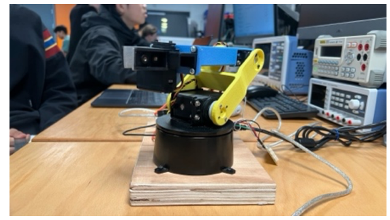


Figure 2.21: View looking into y axis

Based on our observations, we concluded that the equations that we derived for joint angles 1, 2 and 3 are accurate. The differences in measured output coordinates and input coordinates can be explained by parallax error when taking measurements.

Additionally, as the x,y and z axis used as the point of reference to measure the position of the end effector is an imaginary line, hence there exist an inaccuracy for each measurement. This limitation is overcome by taking several measurements for each coordinate and taking the average value.

Moreover, the mechanical error of the robot, such as manufacturing tolerances, wear and tear, and imperfect alignment of components can lead to deviations in the robot's positioning accuracy. Hence, causing the end position to deviate from the specified coordinate.

Finally, we implemented this into the Arduino. In which the movement of the robot with respect to each axis is programmed to be controlled by a potentiometer, with each potentiometer controlling one axis.

Section 3: Trajectory Planning

This section of the report focuses on designing a time profile of position, velocity, and acceleration of our robot's movement. Given the start position of the robot, end position of the robot and time to reach final position, we can generate a cubic polynomial which will describe the trajectory of the robot arm. The

The robot starts from an initial position (μ_0), waits for 10 seconds, then moves in a straight line to the final position (μ_f) according to the cubic polynomial over a period of 5 seconds, with a final time $t_f = 5\ seconds$. Then, the robot waits another 10 seconds and repeats the same trajectory but in the opposite direction instead. Again, the robot moves across a period of 5 seconds with a final time $t_f = 5\ \ seconds$.

The motion repeats until the power is turned off. We generate trajectory points every 0.1 seconds, so we have 50 points from initial to final position and another 50 points from final to initial position.

Derivation of Cubic Polynomial

Since we are using a cubic polynomial to describe the position of the robot, the position of the robot in a specific direction (e.g. x-direction) can be described in the general form below:

$$u(t) = \begin{cases} a_0 + a_1 t + a_2 t^2 + a_3 t^3 & t \leq t_f \\ u_f & t > t_f \end{cases}$$

Where a_i is a coefficient to be derived for $i = 1{,}2{,}3{,}4$.

To derive the expressions for a_i, we can use the four boundary conditions specified below:

$$\begin{gathered} u(0) = u_0 \\ u(t_f) = u_f \\ \dot{u}(0) = 0 \\ \dot{u}(t_f) = 0 \end{gathered} \tag{3.1}$$

where μ_0 is the initial position and μ_f is the final position of the end effector.

Equating the four boundary constraints to the general cubic polynomial form, we get the following equations:

$$u(0) = a_0 + a_1(0) + a_2(0)^2 + a_3(0)^3 = a_0 = u_0 \tag{3.2}$$

$$u(t_f) = a_0 + a_1(t_f) + a_2(t_f)^2 + a_3(t_f)^3 = u_f \tag{3.3}$$

$$\dot{u}(0) = \frac{d}{dt}(a_0 + a_1 t + a_2 t^2 + a_3 t^3) = a_1 + 2a_2(0) + 3a_3(0)^2 = a_1 = 0 \tag{3.4}$$

$$\dot{u}(t_f) = \frac{d}{dt}(a_0 + a_1 t + a_2 t^2 + a_3 t^3) = a_1 + 2a_2(t_f) + 3a_3(t_f)^2 = 0 \tag{3.5}$$

Simplifying and rearranging the above four equations will give us:

Rearranging equation (3.2) gives us:

$$a_0 = u_0 \tag{3.6}$$

Rearranging equation (3.3) gives us:

$$a_1 = 0 \tag{3.7}$$

Rearranging equation (3.4) gives:

$$a_1 + 2a_2(t_f) + 3a_3(t_f)^2 = 2a_2(t_f) + 3a_3(t_f)^2 = 0$$

$$a_2 = -\frac{3a_3 t_f}{2}$$

$$a_2 = -\frac{3}{2}\left(\frac{-2}{t_f^3}(u_f - u_0)\right)t_f$$

$$a_2 = \frac{3}{t_f^2}(u_f - u_0) \tag{3.8}$$

Rearranging equation (3.5) gives us:

$$a_0 + a_2(t_f)^2 + a_3(t_f)^3 = u_0 + a_2(t_f)^2 + a_3(t_f)^3 = u_f$$

$$a_2(t_f)^2 + a_3(t_f)^3 = u_f - u_0$$

$$-\frac{3a_3 t_f}{2}(t_f)^2 + a_3(t_f)^3 = u_f - u_0$$

$$-\frac{1}{2}a_3 = \frac{u_f - u_0}{t_f^3}$$

$$a_3 = \frac{-2}{t_f^3}(u_f - u_0) \tag{3.9}$$

Equations (3.6), (3.7), (3.8) and (3.9) are the expressions used to calculate the coefficient a_i when we are given the initial and final position of the robot.

Now that we have the coefficients for the equation of $u(t)$, we can express the movement of the robot in a specific direction (e.g. x-direction). However, since our robot moves in three-dimensional space, this equation would not be able to account for the other 2 directions (y and z direction).

To overcome this limitation, we can split the equation $u(t)$ into three parts, to account for the x direction, y-direction, and z-direction. Then, we can form 3 equations, $u_x(t), u_y(t)$ and $u_z(t)$ to describe the trajectory of the robot in 3 dimensions.

Next, to test the equations, we choose three different sets of movements, with each set of movements being the robot moving to and from three distinct coordinates.

First Set of Movements

The robot moves from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 100 \\ -90 \\ -140 \end{pmatrix}$, then back to $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$

Table 3.0 shows the values of a_i for the robot's movement from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 100 \\ -90 \\ -140 \end{pmatrix}$:

	a_0	a_1	a_2	a_3
x	$u_0 = 175$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(100 - 175) = -9$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(100 - 175) = 1.2$
y	$u_0 = 0$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-90 - 0) = -10.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-90 - 0) = 1.44$
z	$u_0 = -100$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-140 + 100)$ $= -4.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-140 + 100) = 0.64$

Table 3.0: Table of Results for Cubic Polynomial Coefficients in Forward Direction

Below is a table of 50 coordinates in which the robot passes through in forward direction from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 100 \\ -90 \\ -140 \end{pmatrix}$ in 5 seconds. The robot uses the same array but in the opposite direction for the reverse movement:

0 - X: 174.911194, Y: -0.106560, Z: -100.047363	25 - X: 135.251205, Y: -47.698551, Z: -121.199356
1 - X: 174.649597, Y: -0.420480, Z: -100.186882	26 - X: 133.009613, Y: -50.388470, Z: -122.394875
2 - X: 174.222397, Y: -0.933120, Z: -100.414719	27 - X: 130.782410, Y: -53.061108, Z: -123.582718
3 - X: 173.636795, Y: -1.635840, Z: -100.727043	28 - X: 128.576813, Y: -55.707825, Z: -124.759033
4 - X: 172.899994, Y: -2.520000, Z: -101.120003	29 - X: 126.400017, Y: -58.319981, Z: -125.919991
5 - X: 172.019196, Y: -3.576960, Z: -101.589760	30 - X: 124.259216, Y: -60.888939, Z: -127.061752
6 - X: 171.001602, Y: -4.798081, Z: -102.132477	31 - X: 122.161621, Y: -63.406059, Z: -128.180466
7 - X: 169.854401, Y: -6.174721, Z: -102.744324	32 - X: 120.114426, Y: -65.862694, Z: -129.272308
8 - X: 168.584793, Y: -7.698242, Z: -103.421440	33 - X: 118.124825, Y: -68.250214, Z: -130.333435
9 - X: 167.199997, Y: -9.360003, Z: -104.160004	34 - X: 116.200027, Y: -70.559975, Z: -131.359985
10 - X: 165.707199, Y: -11.151362, Z: -104.956161	35 - X: 114.347229, Y: -72.783333, Z: -132.348145
11 - X: 164.113602, Y: -13.063684, Z: -105.806084	36 - X: 112.573624, Y: -74.911652, Z: -133.294067
12 - X: 162.426392, Y: -15.088325, Z: -106.705925	37 - X: 110.886429, Y: -76.936287, Z: -134.193909
13 - X: 160.652802, Y: -17.216644, Z: -107.651840	38 - X: 109.292824, Y: -78.848610, Z: -135.043823
14 - X: 158.800003, Y: -19.440006, Z: -108.639999	39 - X: 107.800026, Y: -80.639969, Z: -135.839996
15 - X: 156.875198, Y: -21.749767, Z: -109.666565	40 - X: 106.415222, Y: -82.301735, Z: -136.578552
16 - X: 154.885590, Y: -24.137287, Z: -110.727684	41 - X: 105.145622, Y: -83.825256, Z: -137.255676
17 - X: 152.838394, Y: -26.593927, Z: -111.819527	42 - X: 103.998421, Y: -85.201897, Z: -137.867508
18 - X: 150.740799, Y: -29.111050, Z: -112.938248	43 - X: 102.980820, Y: -86.423019, Z: -138.410233
19 - X: 148.599991, Y: -31.680006, Z: -114.080002	44 - X: 102.100021, Y: -87.479980, Z: -138.879990
20 - X: 146.423203, Y: -34.292164, Z: -115.240959	45 - X: 101.363220, Y: -88.364143, Z: -139.272964
21 - X: 144.217606, Y: -36.938881, Z: -116.417282	46 - X: 100.777618, Y: -89.066864, Z: -139.585281
22 - X: 141.990402, Y: -39.611519, Z: -117.605118	47 - X: 100.350410, Y: -89.579506, Z: -139.813126
23 - X: 139.748810, Y: -42.301437, Z: -118.800636	48 - X: 100.088806, Y: -89.893433, Z: -139.952637
24 - X: 137.500000, Y: -44.999992, Z: -120.000000	49 - X: 100.000008, Y: -90.000000, Z: -140.000000

Table 3.1: Table of Results Containing 50 Coordinates of Robot in Forward Direction

Table 3.2 shows the values of a_i for the robot's movement from $\begin{pmatrix} 100 \\ -90 \\ -140 \end{pmatrix}$ to $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$:

	a_0	a_1	a_2	a_3
x	$u_0 = 100$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(175 - 100) = 9$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(175 - 100) = -1.2$
y	$u_0 = -90$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(0 + 90) = 10.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(0 + 90) = -1.44$
z	$u_0 = -140$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-100 + 140) = 4.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-100 + 140)$ $= -0.64$

Table 3.2: Table of Results Calculating Cubic Polynomial Coefficients in Reverse Direction

Final Cubic Polynomial Forms for each axis for the entire movement:

$$u_x(t) = \begin{cases} 175 & t \leq 10 \\ 175 - 9t^2 + 1.2t^3 & 10 < t \leq 15 \\ 100 & 15 < t \leq 25 \\ 100 + 9t^2 - 1.2t^3 & 25 < t \leq 30 \\ 175 & t > 30 \end{cases}$$

$$u_y(t) = \begin{cases} 0 & t \leq 10 \\ -10.8t^2 + 1.44t^3 & 10 < t \leq 15 \\ -90 & 15 < t \leq 25 \\ -90 + 10.8t^2 - 1.44t^3 & 25 < t \leq 30 \\ 0 & t > 30 \end{cases}$$

$$u_z(t) = \begin{cases} -100 & t \leq 10 \\ -100 - 4.8t^2 + 0.64t^3 & 10 < t \leq 15 \\ -140 & 15 < t \leq 25 \\ -140 + 4.8t^2 - 0.64t^3 & 25 < t \leq 30 \\ -100 & t > 30 \end{cases}$$

MATLAB Plots for Position, Velocity & Acceleration In Different Planes

Figures 3.0-3.2 display how the robot vary in terms of position, velocity, and acceleration in the X plane.

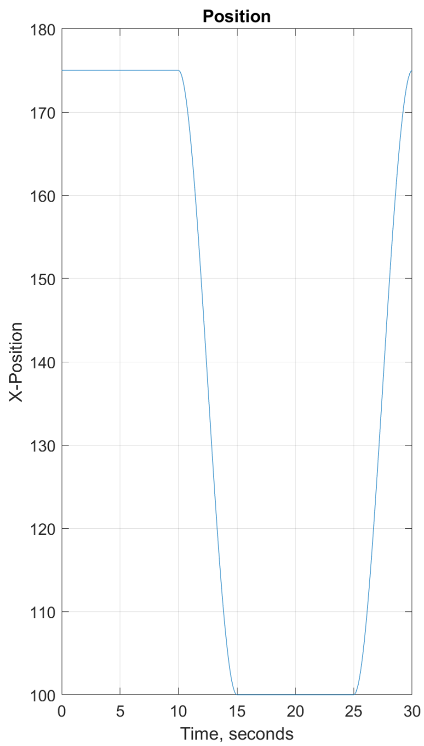


Figure 3.0: Position(mm) against time plot in X-Plane

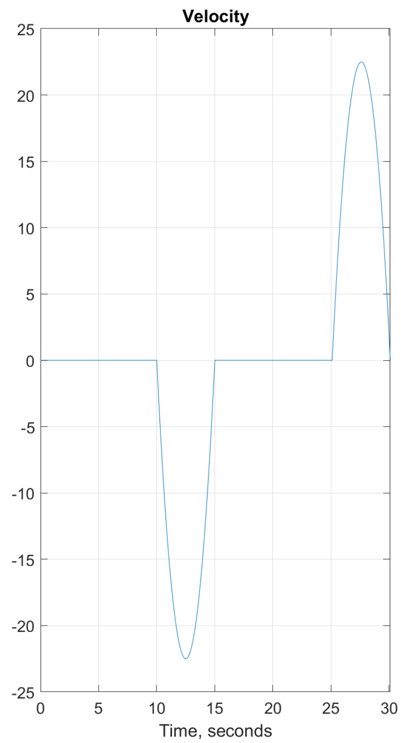


Figure 3.1: Velocity(mm/s) against time plot in X-Plane

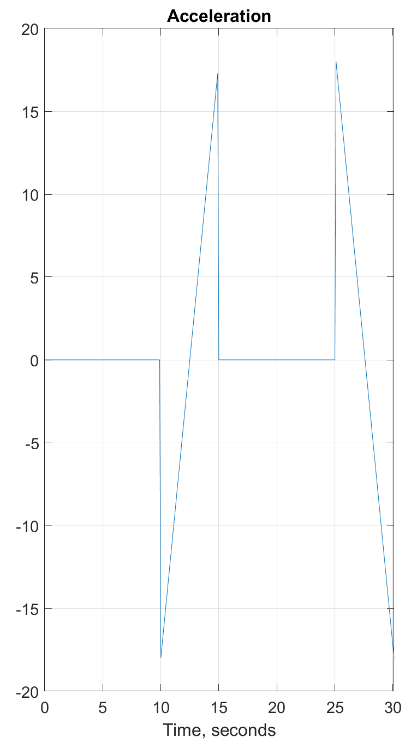


Figure 3.2: Acceleration $\left(\frac{mm^2}{s}\right)$ *against time plot in X-Plane*

Figures 3.3-3.5 display how the robot vary in terms of position, velocity, and acceleration in the Y plane.

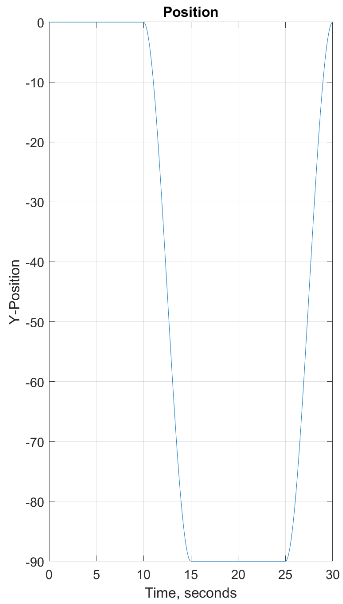


Figure 3.3: Position(mm) against time plot in Y-Plane

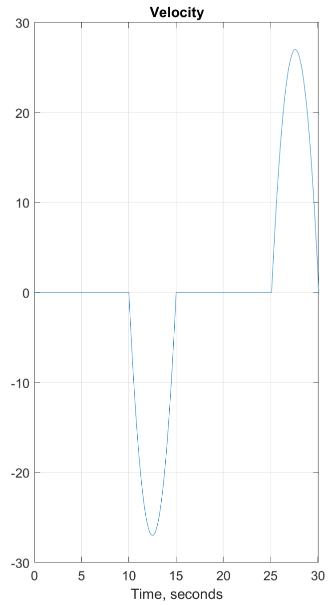


Figure 3.4: Velocity(mm/s) against time plot in Y-Plane

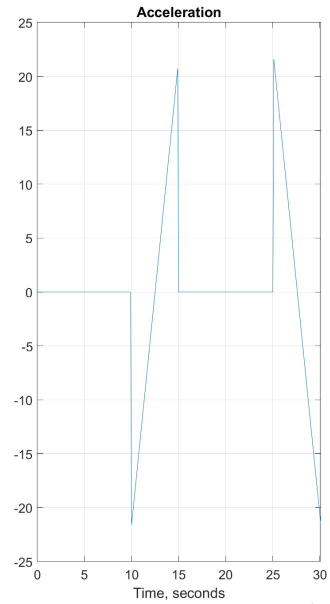


Figure 3.5: Acceleration $\left(\frac{mm^2}{s}\right)$ *against time plot in Y-Plane*

Figures 3.6-3.8 display how the robot vary in terms of position, velocity, and acceleration in the Z plane.

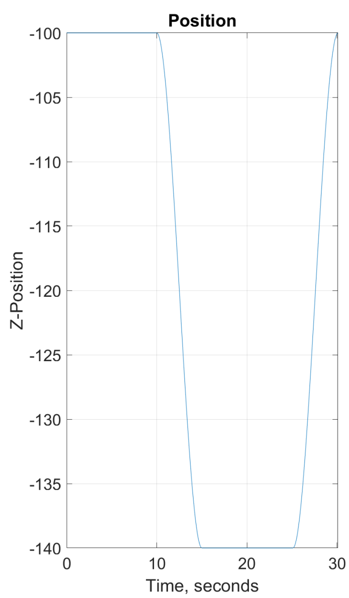


Figure 3.6: Position(mm) against time plot in Z-Plane

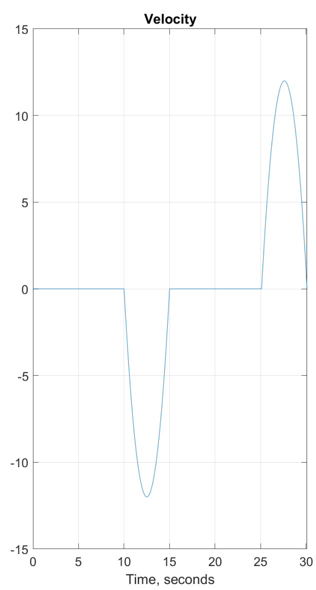


Figure 3.7: Velocity(mm/s) against time plot in Z-Plane

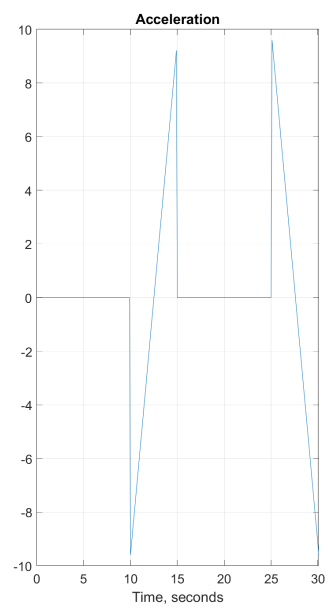


Figure 3.8: Acceleration $\left(\frac{mm^2}{s}\right)$ *against time plot in Z-Plane*

Interpretation of Results

In this section, we will analyze the plot for displacement, velocity, and acceleration for the trajectory of our robot in the x-axis.

We first look at the displacement plot in the x-direction shown in figure 3.0. For the time $10s < t < 15s$, we can see that the robot moves from $x = 175mm$ to $x = 100mm$, starting with an increasing negative gradient, followed by a decreasing negative gradient. This implies that the robot slowly increase its velocity up until a midway point, then slows down before it reaches the end point, resembling an inverse s shape.

For the velocity plots in Figures 3.1, 3.4 & 3.7, we observed the same trend of increasing velocity in the negative direction from 10s to a maximum velocity at 12.5s followed by a decrease in negative velocity. The trend is reciprocated in the positive velocity direction from 25s to 30s with a maximum positive velocity at 27.5s. The maximum positive and negative velocity are equal. The velocity of the robot remained at 0 from 0-10s and 15-25s.

Finally, the acceleration plots in Figure 3.2, 3.5 & 3.8 too showed similar trends of a sharp drop to a maximum negative acceleration at 10s and a constant increase to a maximum positive acceleration at 15s, followed by a sharp drop down to 0 acceleration. The following trend is repeated but from positive acceleration to negative acceleration. These sharp changes in acceleration result in very jerky movements observed by the robot.

Second Set of Movements:

The robot moves from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 150 \\ 90 \\ -90 \end{pmatrix}$, then back to $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$

Table 3.3 shows the values of a_i for the robot's movement from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 150 \\ 90 \\ -90 \end{pmatrix}$:

	a_0	a_1	a_2	a_3
x	$u_0 = 175$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(150 - 175) = -3$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(150 - 175) = 0.4$
y	$u_0 = 0$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(90 - 0) = 10.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(90 - 0) = -1.44$
z	$u_0 = -100$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-90 + 100) = 1.2$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-90 + 100) = -0.16$

Table 3.3: Table of Results for Calculating Cubic Polynomial in Forward Direction

Table 3.4 shows the values of a_i for the robot's movement from $\begin{pmatrix} 150 \\ 90 \\ -90 \end{pmatrix}$ to $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$:

	a_0	a_1	a_2	a_3
x	$u_0 = 150$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(175 - 150) = 3$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(175 - 150) = -0.4$
y	$u_0 = 90$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(0 - 90) = -10.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(0 - 90) = 1.44$
z	$u_0 = -90$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-100 + 90) = -1.2$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-100 + 90) = 0.16$

Table 3.4: Table of Results for Calculating Cubic Polynomial in Reverse Direction

Final Cubic Polynomial Forms for each axis for the entire movement:

$$u_x(t) = \begin{cases} 175 & t \leq 10 \\ 175 - 3t^2 + 0.4t^3 & 10 < t \leq 15 \\ 150 & 15 < t \leq 25 \\ 150 + 3t^2 - 0.4t^3 & 25 < t \leq 30 \\ 175 & t > 30 \end{cases}$$

$$u_y(t) = \begin{cases} 0 & t \leq 10 \\ 10.8t^2 - 1.44t^3 & 10 < t \leq 15 \\ 90 & 15 < t \leq 25 \\ 90 - 10.8t^2 + 1.44t^3 & 25 < t \leq 30 \\ 0 & t > 30 \end{cases}$$

$$u_z(t) = \begin{cases} -100 & t \leq 10 \\ -100 + 1.2t^2 - 0.16t^3 & 10 < t \leq 15 \\ -90 & 15 < t \leq 25 \\ -90 - 1.2t^2 + 1.6t^3 & 25 < t \leq 30 \\ -100 & t > 30 \end{cases}$$

Third Set of Movements

The robot moves from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 150 \\ 60 \\ -60 \end{pmatrix}$, then back to $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$.

Table 3.5 shows the values of a_i for the robot's movement from $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$ to $\begin{pmatrix} 150 \\ 60 \\ -60 \end{pmatrix}$:

	a_0	a_1	a_2	a_3
x	$u_0 = 175$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(150 - 175) = -3$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(150 - 175) = 0.4$
y	$u_0 = 0$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(60 - 0) = 7.2$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(60 - 0) = -0.96$
z	$u_0 = -100$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-60 + 100) = 4.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-60 + 100)$ $= -0.64$

Table 3.5: Table of Results for Calculating Cubic Polynomial in Forward Direction

Table 3.6 shows the values of a_i for the robot's movement from $\begin{pmatrix} 150 \\ 60 \\ -60 \end{pmatrix}$ to $\begin{pmatrix} 175 \\ 0 \\ -100 \end{pmatrix}$:

	a_0	a_1	a_2	a_3
x	$u_0 = 150$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(175 - 150) = 3$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(175 - 150) = -0.4$
y	$u_0 = 60$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(0 - 60) = -7.2$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(0 - 60) = 0.96$
z	$u_0 = -60$	0	$\frac{3}{t_f^2}(u_f - u_0) = \frac{3}{5^2}(-100 + 60)$ $= -4.8$	$\frac{-2}{t_f^3}(u_f - u_0) = -\frac{2}{5^3}(-100 + 60) = 0.64$

Table 3.6: Table of Results for Calculating Cubic Polynomial in Reverse Direction

Final Cubic Polynomial Forms for each axis for the entire movement:

$$u_x(t) = \begin{cases} 175 & t \leq 10 \\ 175 - 3t^2 + 0.4t^3 & 10 < t \leq 15 \\ 150 & 15 < t \leq 25 \\ 150 + 3t^2 - 0.4t^3 & 25 < t \leq 30 \\ 175 & t > 30 \end{cases}$$

$$u_y(t) = \begin{cases} 0 & t \le 10 \\ 7.2t^2 - 0.96t^3 & 10 < t \le 15 \\ 60 & 15 < t \le 25 \\ 60 - 7.2t^2 + 0.96t^3 & 25 < t \le 30 \\ 0 & t > 30 \end{cases}$$

$$u_z(t) = \begin{cases} -100 & t \le 10 \\ 100 + 4.8t^2 - 0.64t^3 & 10 < t \le 15 \\ -60 & 15 < t \le 25 \\ 60 - 4.8t^2 + 0.64t^3 & 25 < t \le 30 \\ -100 & t > 30 \end{cases}$$

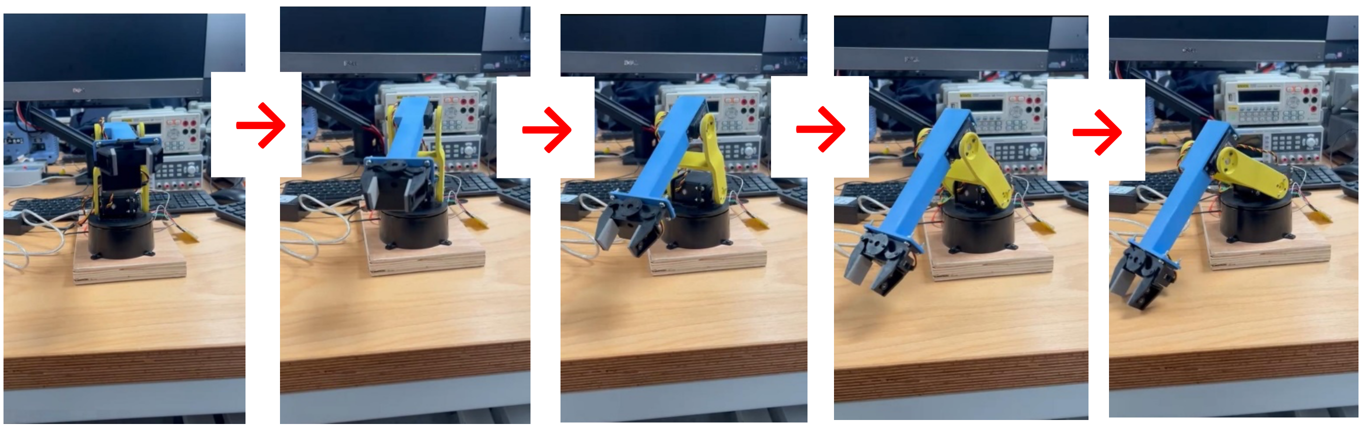


Figure 3.9: Series of Pick & Place Still Images of Trajectory Planning in Forward Direction

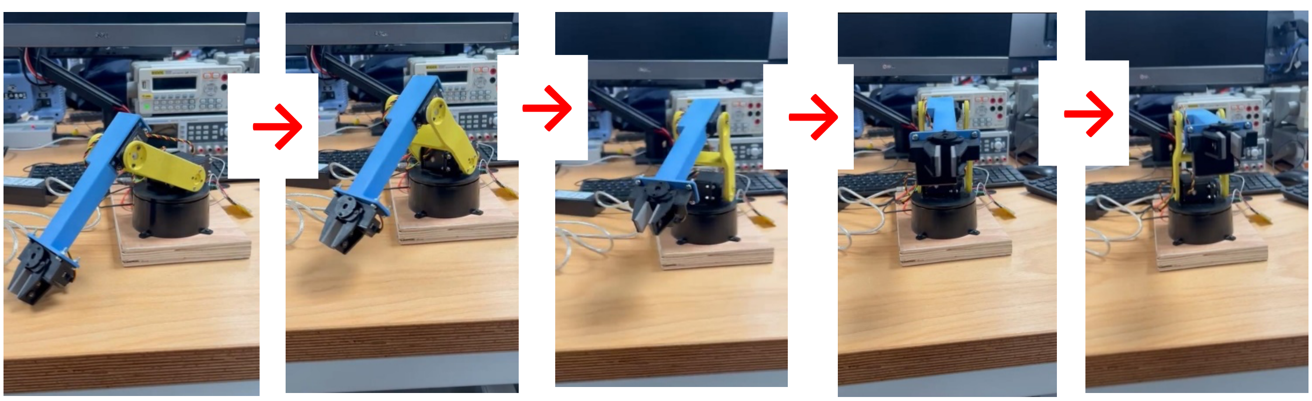


Figure 4.0: Series of Pick & Place Still Images of Trajectory Planning in Reverse Direction

In conclusion, our robot was able to perfectly pick and place objects via trajectory planning through defined initial, final points and cubic polynomials. However, if we were to repeat this task, we would have done a quintic polynomial instead of cubic to avoid jagged and rough movements from the robot, which would result in wear and tear. This is to avoid the sudden changes in acceleration as seen in Figures 3.2, 3.5 & 3.8.