(EEEE3024 UNUK)

Robotics, Dynamics and Control

Coursework 1: Exploration of Forward and Inverse Kinematics based on the DOBOT Magician Robotic Arm

STUDENT NAME: **Kexin Yu**

ID NUMBER: **20320941**

DATE: **28/03/2024**

1. **Describe and specify the robot:**

- **Define the robot type:** The Dobot Magician robotic arm is a four-axis articulated robot with all joints externally rotating. Each axis of the robot corresponds to a joint with rotatable motion. It is possible to define the robot precisely as an RRRR type, where each "R" represents a rotary joint that allows rotary motion around a single axis.

- **Outline payload, operational workspace, reach, accuracy, and the degrees of freedom of the robot:** The Dobot Magician can carry payloads up to a maximum weight of 500 g. Its operational workspace is a spherical area extending outwards from the centre point of the base with a maximum reach of 320 mm. The specific workspace can be as shown in Figure 1, with the workspace angles for each axis as listed in Table 1. Repeat positioning accuracy is $\pm$ 0.2 mm. This robot has 4 degrees of freedom.

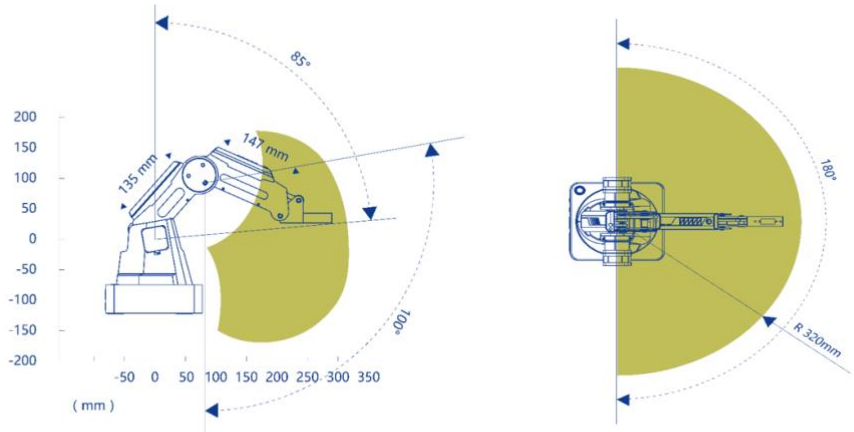

Figure 1: Workspace of Dobot Magician

Table 1: Joint Range of Motion

Axis	Range
Joint 1 base	-90° to $+90^{\circ}$
Joint 2 rear arm	0 to 85°
Joint 3 forearm	-10° to $+90^{\circ}$
Joint 4 rotation servo	-90° to $+90^{\circ}$

- **Type of actuators used and their advantages/disadvantages:** It uses servo motors as its actuators. Below is a detailed description of the advantages and disadvantages of it:

  Advantages: Servo motors have integrated encoders that provide real-time feedback for precise motion control. They are exceptionally efficient, which facilitates energy savings and maintains performance at high speeds. Torque is consistent across a wide range of speeds, and they are capable of rapid acceleration and deceleration. In addition, servo motors are good at maintaining position without the risk of step loss.

  Disadvantages: Servo motors themselves are more expensive and costly to maintain. Have complexity must be precisely adjusted and calibrated. Additionally, servo motors may be sensitive to harsh environments that may interfere with encoder feedback.

- **Describe the end effector options:** It can be equipped with grippers, suction cups, pen holders, laser heads, 3D printing heads, welding heads, sprayers, and cameras. These options allow it to perform a variety of tasks such as gripping, writing, drawing, laser engraving, 3D printing, soldering, and monitoring.
- **Do the end effectors add a degree of freedom? Provide an explanation for your answer:** End effectors do not add additional degrees of freedom. Degrees of freedom refer to the number of ways in which a robot can move independently. The degrees of freedom of Dobot Magician consist of its joints and connections, with each additional independently controllable joint adding a degree of freedom. End effectors are tools that are attached to the last moving joint of the robot, they can perform specific tasks and do not provide additional axes of rotation or movement, so they do not add the degrees of freedom.

2. **Give examples of industrial applications that this type of robot might be used for. Discuss the shortcomings of this robot and how these could be overcome:**

Industrial applications:

Assembly Tasks: It can be configured to carry out exact assembly tasks for assembling small parts with a high degree of repeatability and precision in the electronics manufacturing industry.

Pick and Place: It works well for pick-and-place jobs and can precisely regulate non-destructive part sorting, product packaging and part finishing on the assembly line.

Machine Tending: By autonomously loading and unloading items for machine handling from CNC machines or 3D printers, the robot improves the productivity of repetitive activities.

Quality Inspection: Visual quality checks are carried out by the camera-equipped Dobot to find flaws or confirm the precision of the assembly procedure in order to maintain quality control.

Shortcomings and Solutions

Limited payload and reach: Dobot Magician can only be used for light and small objects due to its modest payload capacity and restricted reach. To get around this, businesses can use Dobot robots to complete different tasks in concurrently or select a larger industrial robot for jobs that call for greater strength and reach.

Speed and efficiency: It could not be as effective as more costly industrial robots built for large-scale manufacturing. This problem can be solved by integrating the robot into operations where speed is not important, or by designing robot motion paths to minimise cycle times.

Durability and longevity: Dobot Magician may not be as durable as industrial-grade robots, which are designed to withstand harsh environments and continuous operation. Regular maintenance and careful operation can help extend its lifespan. For more demanding applications, it may be necessary to invest in an industrial robot with a higher durability rating.

Complex Tasks: The Dobot Magician has limited ability to perform complex tasks that require advanced decision-making or adaptive capabilities. Its capacity to manage more complicated tasks can be improved by integrating artificial intelligence systems.

3. Derive the D-H parameters and provide the D-H matrix for the forward kinematics of the DOBOT Arm.

The form of this robotic arm, its rotatable axiss and the rest of the places where it can move are shown in Figure 2. There are four rotary joints on this robot, and according to the DH parameter principle, the Z-axis of these joints is defined as the axis that passes through the centre of rotation of that joint, as labelled in Fig. It is worth noting that there are two other important points that move. The point O_3 is always moved in the opposite direction to the rotation of the front robotic arm to keep the end-effector parallel to the horizontal plane. The point O_5 is because the end-effector of the device is a sucker, which is always at a distance from the centre of the joint 4.

The end-effector part is not considered in the subsequent analyses because the Z-axis direction of joint 4 is downward and the rotation angle of joint 4 does not affect the end coordinates, only a fixed value of the coordinate shift due to the size of the actuator is added or subtracted. If the coordinates of point O_3 are found, then it is only necessary to move 61 cm in the direction of the x-axis and 60 cm in the direction of the z-axis to get the coordinates of O_5. Meanwhile different end-effectors have different impacts on the position, so only point O_3 is analysed here as the end coordinate.

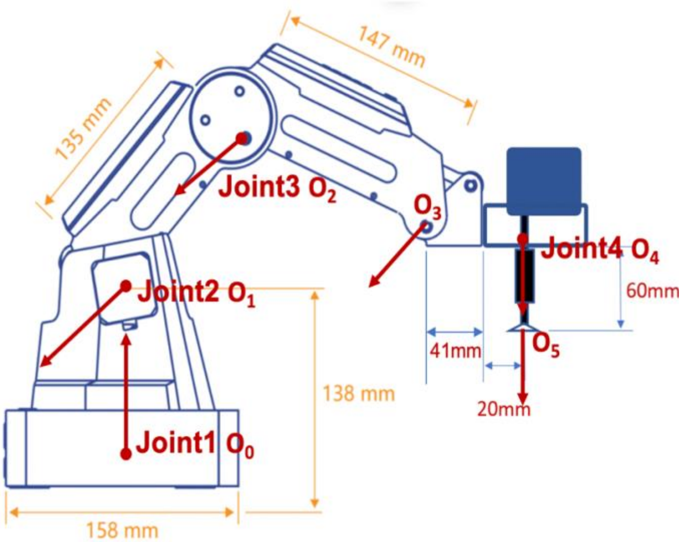

Figure 2: Diagram of the Dobot Magician Arm

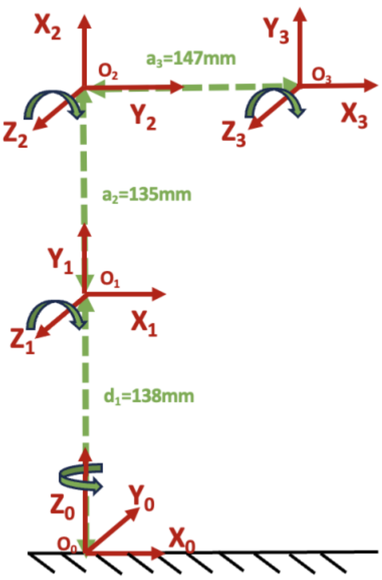

Figure 3: Axes of the different frames of the robotic arm

To obtain the DH parameters and to facilitate subsequent analyses, the robotic arm is assumed to be in an achievable initial state. According to the definition of the D-H parameters, the x-axis is determined based on the principle of making the distance from the z-axis of the previous joint to the z-axis of the current joint the smallest. It is the line perpendicular to the intersection of these two z-axes, pointing from the z-axis of the current joint to the z-axis of the next joint. If the two z-axes are parallel, the x-axis is the shortest line connecting these two axes and is orientated in a direction that makes the linkage length (a) positive. Each y-axis is determined by the right-hand rule, which is orthogonal to the z- and x-axes. Based on the above principle, the x-axis of the initial base X_0 is defined to be towards right, and the X-axis and Y-axis for the remaining different frames are plotted as shown in Figure 3. The following analyses are to find the DH parameters for joints and all the found parameters are summarised in Table 2.

For Frame 1:

- θ_1 is the angle from the x-axis of the joint 1 (O_0) to the x-axis of joint 2 (O_1), it is determined by the rotation angle of the joint 1.
- d_1 is the distance from the origin of the joint 1 (O_0) to the x-axis of joint 2 (O_1), which is the distance vertically upwards, labelled 138 mm in the Fig 3.
- a_1 is 0 because the x-axis of joint 2 (O_1) coincides ith the x-axis of the joint 1(O_0).
- α_1 is the angle from the z-axis of the joint 1 (O_0) to the z-axis of joint 2 (O_1), labelled 90°.

For Frame 2:

- θ_2 is the angle from the x-axis of the joint 2 (O_1) to the x-axis of joint 3 (O_2), it is determined by the rotation angle of the joint 2.
- d_2 is 0 because the x-axis of joint 3 (O_2) is at the same level as the z-axis of joint 2 (O_1).
- a_2 is the distance from the x-axis of joint 2 (O_1) to x-axis of joint 3 (O_2), labelled 135 mm.
- α_2 is 0 because the z-axis of joint 3 (O_2) extends along the z-axis of joint 2 (O_1).

For Frame 3:

- θ_3 is the angle from the x-axis of the joint 3 (O_2) to the x-axis of O_3, it is determined by the rotation angle of the joint 3.
- d_3 is also 0 because the x-axis of O_3 is at the same level as the z-axis of joint 3 (O_2).
- a_3 is the distance from the x-axis of joint 3 (O_2) to the x-axis of O_3, labelled 147 mm.
- α_3 is also 0 because the z-axis of O_3 extends along the z-axis of joint 3 (O_2).

It is worth noting that a quantity is defined here as the Joint Angle Offset: this offset is the difference between the actual zero position of the joint and the initial position defined in the model. If there is a fixed angular difference between the natural "zero position" of a rotating joint in its physical design (the position without external forces) and the "zero position" chosen in the D-H model, then this angle is the Joint Angle Offset. This offset needs to be considered in the program or control algorithm of the robot to ensure the accuracy of its movements. Here the two frames have non-zero offsets:

- Frame 2 has a 90°offset, which means that when the control parameter (θ_2) for the joint is set to 0, the joint is actually in the 90°position.
- Frame 3 has a $-90°$offset, which means that when the control parameter (θ_3) of the joint is set to 0, the joint is actually in the $-90°$position.

Table 2: The D-H parameters of the Dobot Magician Arm

Frame	θ_i	α_i	d_i	a_i	Offset
1	θ_1	90°	138	0	0
2	θ_2	0	0	135	90°
3	θ_3	0	0	147	-90°

According to the standard D-H representation, equation 3.1 is the transformation matrix from the previous frame to the current frame. Based on the parameters in Table 2, the transformation matrix for each successive frame can be expressed as follows:

$$^{i-1}T_i = \begin{pmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \textbf{Eqn 3.1}$$

$$^{0}T_1 = \begin{pmatrix} cos\theta_1 & 0 & sin\theta_1 & 0 \\ sin\theta_1 & 0 & -cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta_1 & 0 & sin\theta_1 & 0 \\ sin\theta_1 & 0 & -cos\theta_1 & 0 \\ 0 & 1 & 0 & 138 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{1}T_2 = \begin{pmatrix} cos\theta_2 & -sin\theta_2 & 0 & a_2 cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & a_2 sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta_2 & -sin\theta_2 & 0 & 135cos\theta_2 \\ sin\theta_2 & cos\theta_2 & 0 & 135sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{2}T_3 = \begin{pmatrix} cos\theta_3 & -sin\theta_3 & 0 & a_3 cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & a_3 sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} cos\theta_3 & -sin\theta_3 & 0 & 147cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & 147sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Therefore, for this robotic arm the total transformation equation is represented as Eqn3.2, which is computationally complex and hence MATLAB is used to assist in the calculation. The calculated total matrix result is represented as follows:

$$^{0}T_3 = {}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3 \quad \textbf{Eqn 3.2}$$

$$^{0}T_3 = \begin{pmatrix} C_{23}C_1 & -S_{23}C_1 & S_1 & C_1(a_3C_{23} + a_2C_2) \\ C_{23}S_1 & -S_{23}S_1 & -C_1 & S_1(a_3C_{23} + a_2C_2) \\ S_{23} & C_{23} & 0 & d_1 + a_3S_{23} + a_2S_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Eqn 3.3 demonstrates that the total matrix transformation equation contains information about the position of the P, so the P-point coordinates are represented as follows, where $d_1 = 138$, $a_2 = 135$, $a_3 = 147$.

$$^{0}T_3 = \begin{pmatrix} {}^{0}R_3 & P_3 \\ 0\,0\,0 & 1 \end{pmatrix} \quad \textbf{Eqn 3.3}$$

$$P_3 = \begin{pmatrix} cos\theta_1(a_3\cos(\theta_2+\theta_3) + a_2\cos(\theta_2)) \\ sin\theta_1(a_3\cos(\theta_2+\theta_3) + a_2\cos(\theta_2)) \\ d_1 + a_3\sin(\theta_2+\theta_3) + a_2\sin(\theta_2) \end{pmatrix} = \begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix}$$

3.a. For a given set of joint parameters calculate the position of the end effector mount point P(xyz). (you decide on an appropriate set of joint parameters)

Based on the limitations of the angle of rotation of each joint in Table 1 and Figure 1, it may be noted that θ_1 can be rotated counterclockwise or clockwise to 90°, θ_2 can only be rotated clockwise to 85°, and θ_3 can be rotated counterclockwise by 10° and clockwise by 90°. A clockwise turn in the table is defined as a positive value and an anti-clockwise turn is defined as a negative value, while in reality it is opposite.

Based on the above principles, θ_1 is selected here as a counterclockwise rotation of $60^{\circ}(+60^{\circ})$, θ_2 as a clockwise rotation of $45^{\circ}(-45^{\circ})$, and θ_3 as a clockwise rotation of $30^{\circ}(-30^{\circ})$. Because offset is also considered, the actual θ_2 and θ_3 are $(90^{\circ}-45^{\circ})$ and $(-90^{\circ}-30^{\circ})$ respectively. The P-point coordinates are listed as follows, this equation is still used MATLAB code to calculate, the code is shown in Figure 4, and its results are displayed in Figure 5.

$$P_3=\begin{pmatrix}X_P\\Y_P\\Z_P\end{pmatrix}=\begin{pmatrix}cos(60^{\circ})(147cos\ (90^{\circ}-45^{\circ}-90^{\circ}-30^{\circ})+135cos\ (90^{\circ}-45^{\circ})\\cos(60^{\circ})(147cos\ (90^{\circ}-45^{\circ}-90^{\circ}-30^{\circ})+135cos\ (90^{\circ}-45^{\circ})\\138+135sin(90^{\circ}-45^{\circ}-90^{\circ}-30^{\circ})+135sin\ (90^{\circ}-45^{\circ})\end{pmatrix}$$

$$\begin{pmatrix}X_P\\Y_P\\Z_P\end{pmatrix}=\begin{pmatrix}cos(60^{\circ})(147cos\ (-75^{\circ})+135cos\ (45^{\circ})\\cos(60^{\circ})(147cos\ (-75^{\circ})+135cos\ (45^{\circ})\\138+135sin(-75^{\circ})+135sin\ (45^{\circ})\end{pmatrix}$$

```
theta1 = deg2rad(60);
theta2 = deg2rad(90-45);
theta3 = deg2rad(-90-30);
d1 = 138;
a2 = 135;
a3 = 147;
% Define the position of the end-effector (P) in the base frame (0)
P = [cos(theta1) * (a3 * cos(theta2 + theta3) + a2 * cos(theta2));
     sin(theta1) * (a3 * cos(theta2 + theta3) + a2 * cos(theta2));
     d1 + a3 * sin(theta2 + theta3) + a2 * sin(theta2)];
% Display the numeric position of the end-effector
 disp(P);
```

Figure 4: Matlab code for calculating the position of the P point

```
  66.7529
 115.6194
  91.4683
```

Figure 5: Matlab code calculation results

The coordinates of point P are expressed as follows:

$$P_3=\begin{pmatrix}X_P\\Y_P\\Z_P\end{pmatrix}=\begin{pmatrix}66.7529\\115.6194\\91.4683\end{pmatrix}$$

4. Derive a possible set of inverse kinematic equations.

The inverse kinematic equations are the inverse of the known coordinates of the end P-point to find the angle rotated by different axes, where the geometrical method is used here to speculate the equations. To better demonstrate the mathematical relationship between position coordinate and angle, Figure 6 shows the state of the arm as it rotates at different axis angles, which is shown in a three-dimensional view. The following is an analysis of the effect of different angle variations on the end position.

Rotation from initial position to θ_1 rotates: The arm rotates θ_1 around the Z-axis of the base joint O_0. This determines the planar rotation of the arm in the XY plane. The end-effector (P) maintains the same height in the Z-axis, but its position in the XY-plane changes due to the rotation. The dashed green line shows the projection of the second part in the XY plane after the rotation, which is $\cos(\theta_1)$ times the length of the arm segment.

Rotation from θ_1 rotates to θ_2 rotates: The arm rotates θ_2 around the Z-axis of joint O_1. Since the angle between joint O_1 and joint O_2 does not change, joint O_2 is also rotated θ_2 degrees. This changes the position of the end-effector along the Z-axis and its projected length in the XY plane.

Rotation from θ_2 rotates to θ_3 rotates: The arm rotates θ_3 around the Z-axis of joint O_2. This also changes the position of the end-effector along the Z-axis and its projected length in the XY plane. It can be seen in the figure that the end-effector change from P' to P.

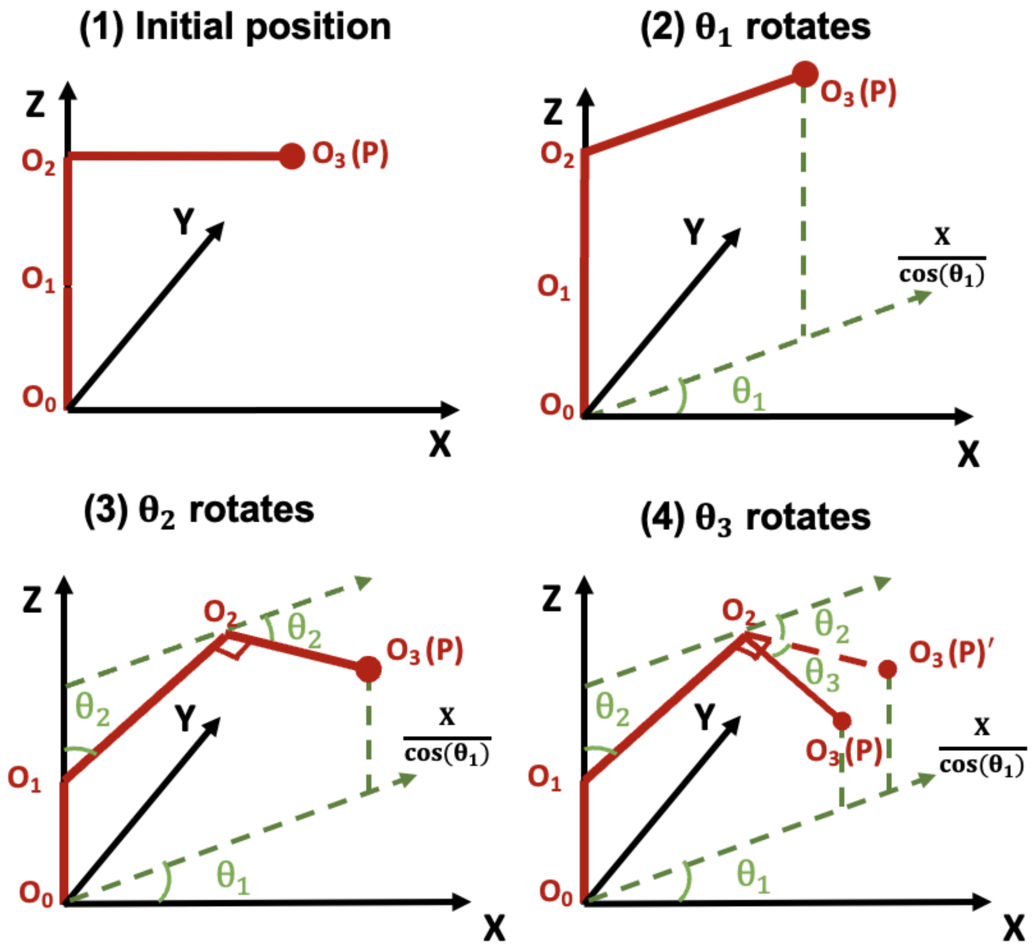

Figure 6: The state of the robotic arm in 3D view with different angles of rotation

Figure 7 illustrates the 2D view of (4) in Figure 6 to facilitate the mathematical relationship between angle and coordinate position. The figure looks at the Y and X axes from the top view and ($X_P/\cos(\theta_1)$) axis and Z axis from the side view, respectively. The lengths of the different arm segments, the coordinates of the end-effector (X_P, Y_P, Z_P), and the angles of rotation of the different joints are marked in green text. The blue text marks the angles or edge lengths that are used in later calculations.

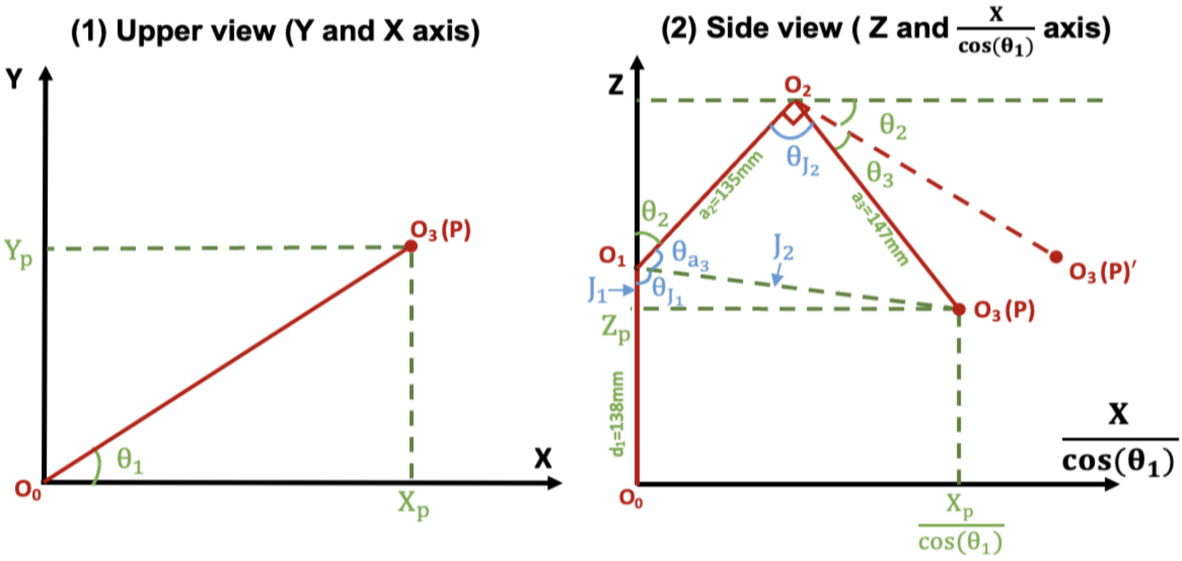

Figure 7: Different 2D views of the robotic arm

The relationship between θ_1 and Y_p, X_p can be obtained from the diagram on the left, as shown in Eqn 4.1:

$$\theta_1 = \arctan\left(\frac{Y_p}{X_p}\right) \qquad \textbf{Eqn 4.1}$$

It can be noted from the figure on the right that θ_3 is related to θ_{J2} by a sum of 90 degrees, which can be listed as Eqn 4.2. $\cos(\theta_{J_2})$ can be represented by the cosine theorem, as shown in Eqn 4.3, where the only unknown quantity J_2 is the length of the side corresponding to θ_{J2} . J_2 can be found by the relationship between the length of the side J_1, and the length of ($X_P/\cos(\theta_1)$), as shown in Eqn 4.4. J_1 is the difference between the length of the first segment of the robotic arm d_1 and the end Z-axis coordinate Z_P, as shown in Eqn 4.5. By combining all the above equations, the representation of θ_3 is obtained as Eqn 4.6.

$$\theta_3 = 90^\circ - \theta_{J_2} \qquad \textbf{Eqn 4.2}$$

$$\cos(\theta_{J_2}) = \frac{{a_2}^2 + {a_3}^2 - {J_2}^2}{2a_2a_3} \qquad \textbf{Eqn 4.3}$$

$$J_2 = \sqrt{{J_1}^2 + \left(\frac{X_p}{\cos(\theta_1)}\right)^2} \qquad \textbf{Eqn 4.4}$$

$$J_1 = d_1 - Z_p \qquad \textbf{Eqn 4.5}$$

$$\theta_3 = 90^\circ - \arccos\left(\frac{{a_2}^2 + {a_3}^2 - \left((d_1 - Z_p)^2 + \left(\frac{X_p}{\cos(\theta_1)}\right)^2\right)}{2a_2a_3}\right) \qquad \textbf{Eqn 4.6}$$

From the diagram on the right, also note that θ_2 is additively related to θ_{a3} and θ_{J1} by 180 degrees, which can be given as Eqn 4.7. $\cos(\theta_{a3})$ can be expressed by the cosine theorem as Eqn 4.8. $\tan(\theta_1)$ can be expressed as Eqn 4.9. Joining all the above equations gives the representation of θ_2 as Eqn 4.10.

$$\theta_2 = 180^{\circ} - \theta_{J_1} - \theta_{a_3} \quad \textbf{Eqn 4.7}$$

$$cos\left(\theta_{a_3}\right) = \frac{{a_2}^2 + {J_2}^2 - {a_3}^2}{2a_2J_2} \quad \textbf{Eqn 4.8}$$

$$tan\left(\theta_{J_1}\right) = \frac{\frac{X_p}{cos\,(\theta_1)}}{J_1} = \frac{X_p}{cos\,(\theta_1)J_1} \quad \textbf{Eqn 4.9}$$

$$\theta_2 = 180^{\circ} - arctan\,\left(\frac{X_p}{cos(\theta_1)(d_1 - Z_p)}\right) - arccos\,\left(\frac{{a_2}^2 + (d_1 - Z_p)^2 + (\frac{X_p}{cos(\theta_1)})^2 - {a_3}^2}{2a_2\sqrt{(d_1 - Z_p)^2 + (\frac{X_p}{cos\,(\theta_1)})^2}}\right) \quad \textbf{Eqn 4.10}$$

Based on the above calculations, the formulae for different angles can be summarised as follows, $d_1(135\text{mm})$, $a_2(138\text{mm})$, and $a_3(145\text{mm})$ are the lengths of the different robotic arms respectively:

$$\begin{cases} \theta_1 = arctan\left(\frac{Y_p}{X_p}\right) \\ \theta_2 = 180^{\circ} - arctan\,\left(\frac{X_p}{cos(\theta_1)(d_1 - Z_p)}\right) - arccos\,\left(\frac{{a_2}^2 + (d_1 - Z_p)^2 + (\frac{X_p}{cos(\theta_1)})^2 - {a_3}^2}{2a_2\sqrt{(d_1 - Z_p)^2 + (\frac{X_p}{cos\,(\theta_1)})^2}}\right) \\ \theta_3 = 90^{\circ} - arccos\,\left(\frac{{a_2}^2 + {a_3}^2 - ((d_1 - Z_p)^2 + (\frac{X_p}{\cos(\theta_1)})^2)}{2a_2a_3}\right) \end{cases}$$

4.a. Solve the resulting set of equations for the point P(xyz) calculated in task 4 and show the solutions are consistent with the joint variables chosen in task 4.

The following are the results of the coordinates obtained in Q3:

$$\begin{cases} X_p = 66.7529 \\ Y_p = 115.6194 \\ Z_p = 91.4683 \end{cases}$$

Bringing this into the equation obtained in Q4, the result can be expressed as follows:

$$\begin{cases} \theta_1 = \arctan\left(\frac{115.6194}{66.7529}\right) \\ \theta_2 = 180^{\circ} - \arctan\,\left(\frac{66.7529}{\cos\left(\arctan\left(\frac{115.6194}{66.7529}\right)\right)(138 - 91.4683)}\right) - \arccos\,\left(\frac{135^2 + (138 - 91.4683)^2 + (\frac{66.7529}{\cos\left(\arctan\left(\frac{115.6194}{66.7529}\right)\right)})^2 - 147^2}{270 \times \sqrt{(138 - 91.4683)^2 + (\frac{66.7529}{\cos\,(\arctan\left(\frac{115.6194}{66.7529}\right))})^2}}\right) \\ \theta_3 = 90^{\circ} - \arccos\,\left(\frac{135^2 + 147^2 - ((138 - 91.4683)^2 + (\frac{66.7529}{\cos\left(\arctan\left(\frac{115.6194}{66.7529}\right)\right)})^2)}{2 \times 135 \times 147}\right) \end{cases}$$

As the process involves a lot of calculations, MATLAB is used here to assist, the code run by MATLAB is shown in Figure 8. The results of its running calculations are shown in Figure 9.

```
% Define the variables
a2 = 135;
a3 = 147;
d1 = 138;
% xp, yp, and zp
xp = 66.7529;
yp = 115.6194;
zp = 91.4683;
% Calculate theta1
theta1 = atan(yp/xp);
% Calculate theta2
theta2 = 180 - rad2deg(acos((a2^2 + (d1 - zp)^2 + (xp / cos(theta1))^2 - a3^2) / ...
         (2 * a2 * sqrt((d1 - zp)^2 + (xp / cos(theta1))^2))))-
rad2deg(atan(xp / (cos(theta1) * (d1 - zp))));
% Calculate theta3
theta3 = 90 - rad2deg(acos((a2^2 + a3^2 - ((d1 - zp)^2 + (xp / cos(theta1))^2)) / ...
         (2 * a2 * a3)));
% Convert angles in degrees
theta1 = rad2deg(theta1);
% Display the results
fprintf('Theta1: %f degrees\n', theta1);
fprintf('Theta2: %f degrees\n', theta2);
fprintf('Theta3: %f degrees\n', theta3);
```

Figure 8: MATLAB code for calculating angles

```
Theta1: 59.999997 degrees
Theta2: 45.000006 degrees
Theta3: 30.000009 degrees
```

Figure 9: Calculation results of the angles

The results presented in Figure 9 are consistent with those set out in Q3, which suggests that inverse kinematic equations of motion in Q4 is correct.