HO CHI MINH CITY UNIVERSITY OF
TECHNOLOGY AND EDUCATION

FACULTY OF MECHANICAL ENGINEERING

MODELLING AND CONTROL OF INDUSTRIAL MANIPULATORS

FORWARD AND INVERSE KINEMATICS MODEL OF INDUSTRIAL ROBOT MANIPULATOR IRB 1200-7/0.7

Lecturer: Dr. Tran Minh Thien
Student: Tran Nhat Hoang - 21134008
Le Duc Phap Anh - 21134001

Ho Chi Minh City, 7 November

CONTENTS

FORWARD AND INVERSE KINEMATICS MODEL OF INDUSTRIAL ROBOT MANIPULATOR IRB 1200-7/0.7

1. INTRODUCTION

An industrial manipulative robot, commonly referred to as an industrial robot, is officially defined by the International Organization for Standardization (ISO) as an automatically controlled, reprogrammable, multipurpose manipulator featuring three or more programmable axes. These robotic systems can exhibit both stationary and mobile configurations and find extensive application within the realm of industrial automation. They have been globally adopted to undertake tasks characterized by repetitive actions and a high potential for human injury, thereby serving as substitutes for human labor in such contexts. Industrial robots come in a variety of mechanical configurations, each differing in terms of their degrees of freedom (DOF) and types of joints. The selection of a particular configuration is contingent upon the robot's intended task and operational environment.

To effectively program the robot arm, the establishment of a comprehensive mathematical model is imperative, particularly with regard to forward and inverse kinematics. Numerous methodologies are available for computing forward kinematic equations, ranging from direct computation for simpler structures to the utilization of transformation matrices, where each matrix represents a link or joint. Nevertheless, the former approach proves unsuitable for addressing the intricacies of this robot's structure. Regarding the latter technique, while it yields precise results, the large number of links and joints in this robot necessitates a computation process characterized by numerous steps (the number of matrices equating to the sum of the number of links and joints). Consequently, to compute the forward kinematics for the Fanuc industrial robot, the Denavit-Hartenberg convention is employed. This convention offers a lower level of complexity and a reduced number of matrices (equivalent to the number of degrees of freedom), thereby simplifying the derivation of forward kinematic equations.

Following the acquisition of forward kinematics for the robot, the subsequent step entails solving these equations to establish the relationship between manipulated/input variables (position and orientation of the end effector) and controlled/output variables (angles of each joint). This process is termed inverse kinematics, with the corresponding equations being referred to as inverse kinematic equations. However, due to the wide array of robot configurations, there exists no universally applicable method for deriving inverse kinematic equations. Consequently, fundamental mathematical techniques, such as trigonometric identities, are employed to solve for inverse kinematics.

2. METHODOLOGY

The ABB IRB 1200-7/0.7 is a high-performance, compact industrial robot designed for a wide range of applications, including machine tending, material handling, and assembly. It features a 7 kg payload, a 703 mm reach, and a 0.02 mm repeatability, making it ideal for precision tasks. The ABB IRB 1200-7/0.7 is a versatile and reliable robot that can be used to improve efficiency and productivity in a wide range of industries.

2.1. TECHNICAL INFORMATION

Before building mathematical model for the robot, we must first study its specifications. The Table 1 are the technical specifications of the robot arm (official datasheet), including number of axes, reach, range of angle of rotation, load capacity, allow, drive method, mass and installation environment.

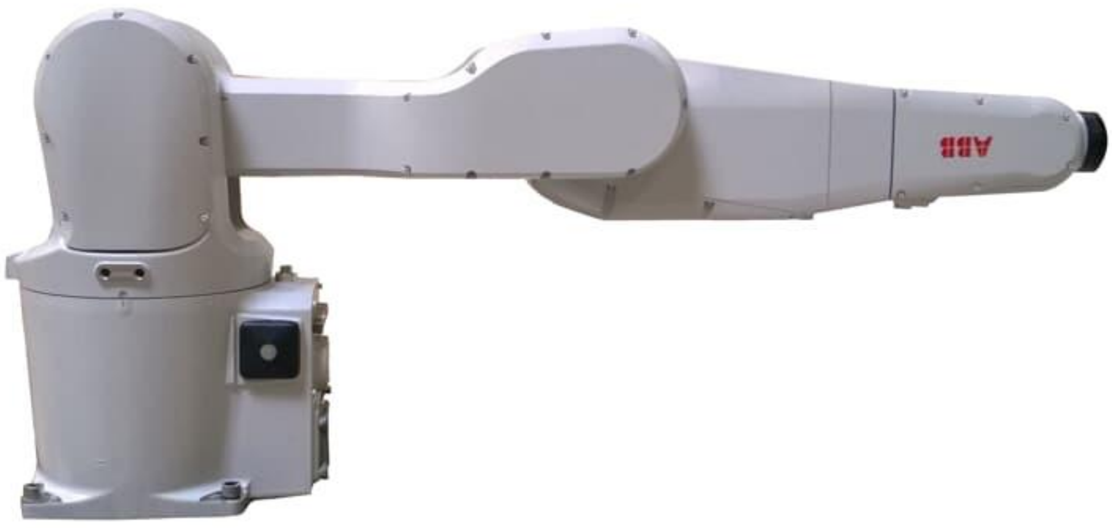


Figure 1: IRB 1200-7/0.7

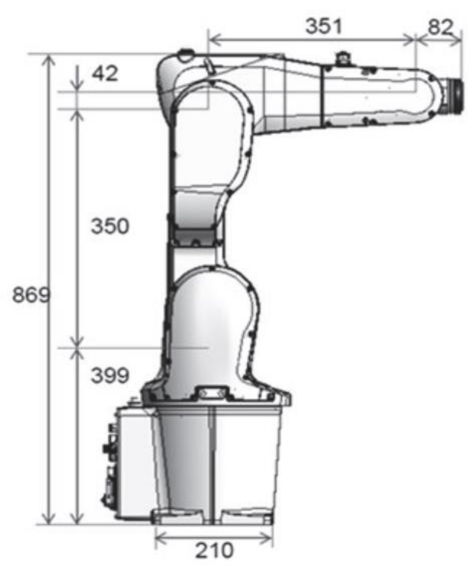


Figure 2: Dimension of IRB 1200-7/0.7

Specification			
Robot version	*Reach (m)*	*Payload (kg)*	*Armload (kg)*
IRB 1200-7/0.7	0.7	7	0.3
Number of axes	6		
Protection	Standard robot (IP40) or IP67, Foundry Plus 2 based on IP67, Food Grade Lubricant and Clean Room (150 class 3) based on IP67		
Mounting	Any angle		
Controller	IRC5 Compact/IRC5 Single Cabinet		
Integrated signal and power supply	10 signals on wrist		
Integrated air supply	4 air on wrist (5 Bar)		
Integrated ethernet	One 100/10 Base-TX ethernet port		
Performance (according to ISO 9283)			
1 kg picking cycle			
25 x 300 x 25 mm			
Max. TCP Acceleration	35 m/s		
Acceleration time 0-1 m/s	0.06 s		
Position repeatability	0.02 mm		
Technical information			
Electrical Connections			
Supply voltage	200-600 V, 50-60 Hz		
Rated power transformer rating	0.39 kW		
Power consumption	0.39 kW		
Physical	*Base (mm)*	*Height (mm)*	*Weight (kg)*
IRB 1200-7/0.7	210 x 210	869	54
Movement			
Axis movement	*Working range*	*Axis max. speed*	
Axis 1 rotation	$-170° \div 170°$	$288°/s$	
Axis 2 arm	$-100° \div 135°$	$240°/s$	
Axis 3 arm	$-200° \div 70°$	$300°/s$	
Axis 4 wrist	$-270° \div 270°$	$400°/s$	
Axis 5 bend	$-130° \div 130°$	$405°/s$	
Axis 6 turn	Default: $-400° \div 400°$ Max. rev: ± 242	$600°/s$	

Table 1: IRB 1200-7/0.7 Specification

Besides specifications, the datasheet also provides a technical drawing with links length and workspace (Figure 3)

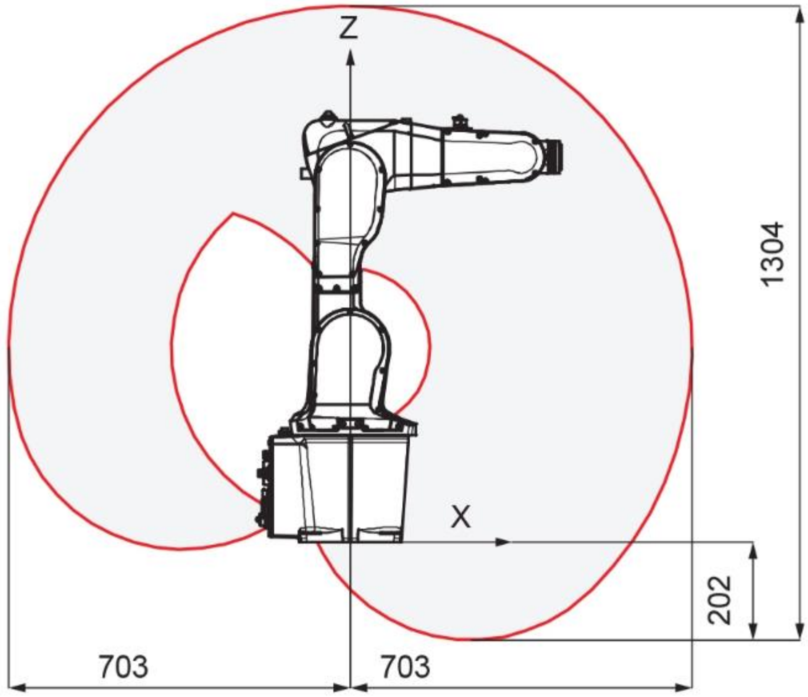


Figure 3: IRB 1200-7/0.7 workspace

2.2. ROTATION AND POSITION TRANSFORM

Assume the set-up position of the robot is in the form showed in the figure below:

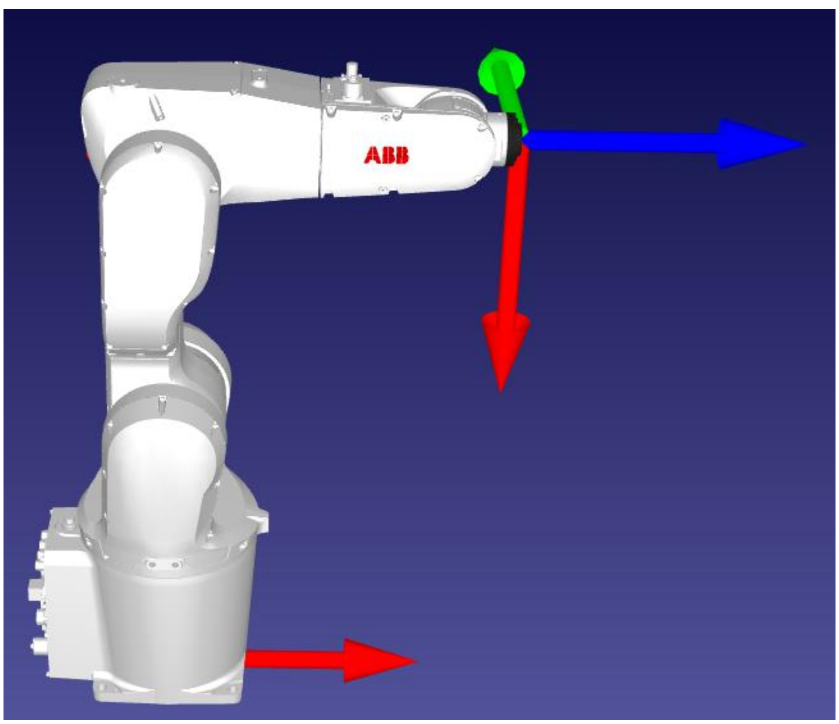


Figure 4:Model 3D of IRB 1200-7/0.7

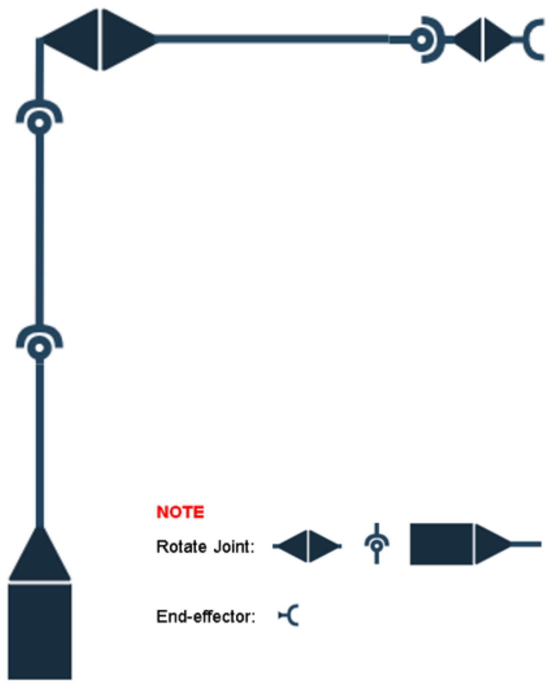


Figure 5:Model 2D of IRB 1200-7/0.7

Now we will testing the rotation of each joint, and see the final result after the transformation. When calculating, we take the standard units for length (mm) and angle (radian).

<u>Joint 1:</u>

We have the rotation matrix when rotating around the z matrix:

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

In the local frame:

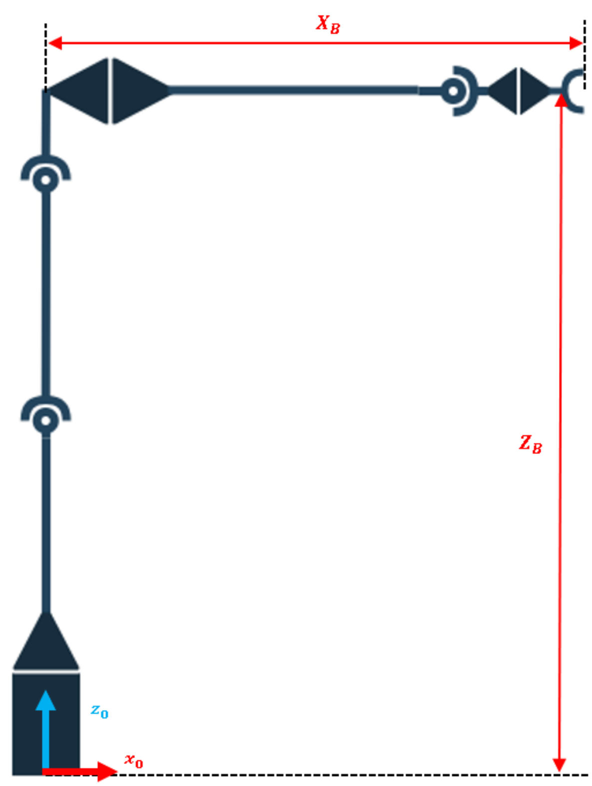
$$r_p^B = [X_B, 0, Z_B]^T$$

With $X_B = 433,\ Z_B = 791$

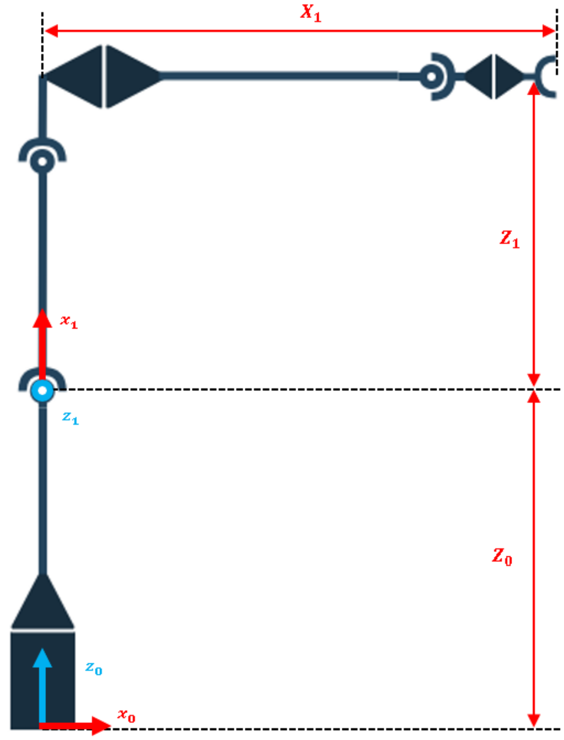
$$r_p^G = A_z^{-1} r_p^B$$

Finally, we get the equation:

$$r_p^G = [X_P, Y_p, Z_P]^T = \begin{pmatrix} X_B cos(\varphi) \\ X_B sin(\varphi) \\ Z_B \end{pmatrix}$$



Joint 2:



We have the translation vector for the ($x_1y_1z_1$) coordination:

$$d_G = [0, 0, Z_0]^T$$

We have the rotation matrix when rotating around the y matrix:

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

In the local frame:

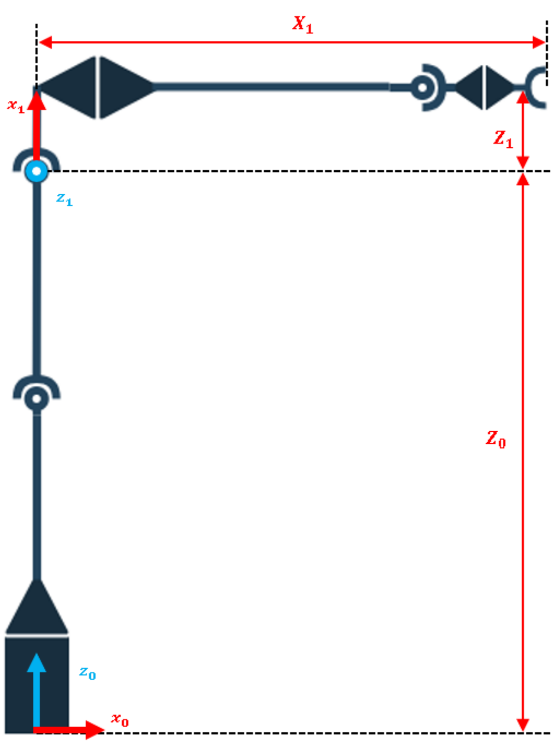
$$r_p^B = [X_1, 0, Z_1]^T$$

With $Z_0 = 399,\ X_1 = 433,\ Z_1 = 392$

$$r_p^G = A_y^{-1} r_p^B + d_G$$

Finally, we get the equation:

$$r_p^G = [X_P, Y_p, Z_P]^T = \begin{pmatrix} X_1\, cos(\varphi) + Z_1 \sin(\varphi) \\ 0 \\ -X_1\, sin(\varphi) + Z_1 \cos(\varphi) + Z_0 \end{pmatrix}$$



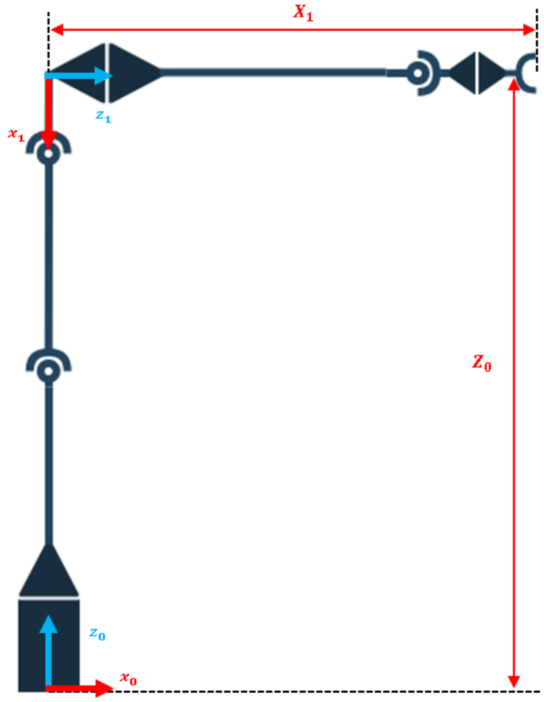
Joint 3:

Similar to Joint 2, with the parameters is:

$$Z_0 = 749,\ X_1 = 433,\ Z_1 = 42$$

Apply the index into the equation below:

$$r_p^G = [X_P, Y_p, Z_P]^T = \begin{pmatrix} X_1\, cos(\varphi) + Z_1 \sin(\varphi) \\ 0 \\ -X_1\, sin(\varphi) + Z_1 \cos(\varphi) + Z_0 \end{pmatrix}$$



Joint 4:

We have the translation vector for the ($x_1y_1z_1$) coordination:

$$d_G = [0, 0, Z_0]^T$$

We have the rotation matrix when rotating around the x matrix:

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

In the local frame:

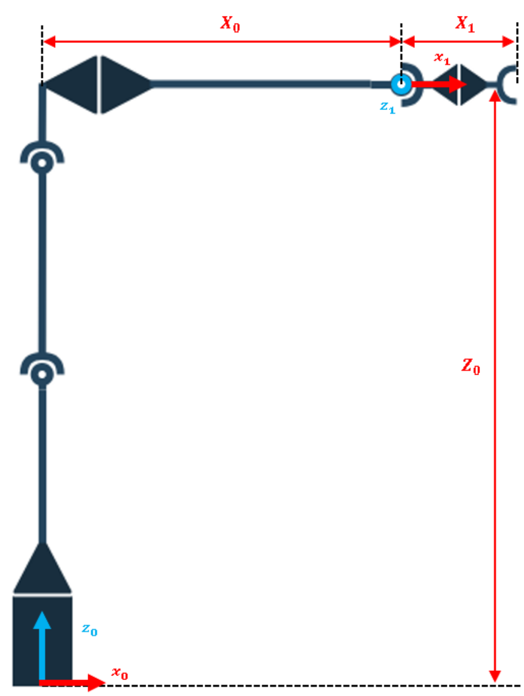
$$r_p^B = [X_1, 0{,}0]^T$$

$$\text{With } Z_0 = 791,\ X_1 = 433$$

$$r_p^G = A_x^{-1} r_p^B + d_G$$

Finally, we get the equation: $r_p^G = [X_P, Y_p, Z_P]^T = \begin{pmatrix} X_1 \\ 0 \\ Z_0 \end{pmatrix}$

Joint 5:



We have the translation vector for the ($x_1y_1z_1$) coordination:

$$d_G = [X_0, 0, Z_0]^T$$

We have the rotation matrix when rotating around the y matrix:

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

In the local frame:

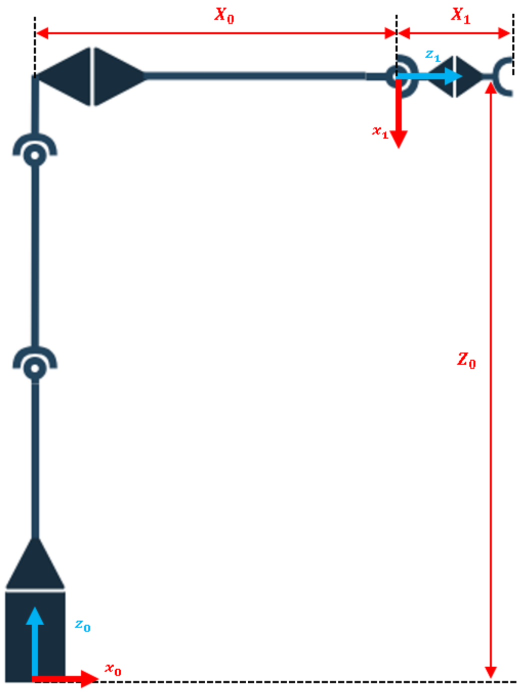
$$r_p^B = [X_1, 0{,}0]^T$$

With $X_0 = 351, Z_0 = 791,\ X_1 = 82$

$$r_p^G = A_y^{-1} r_p^B + d_G$$

Finally, we get the equation:

$$r_p^G = [X_P, Y_p, Z_P]^T = \begin{pmatrix} X_0 + X_1\, cos(\varphi) \\ 0 \\ -X_1\, sin(\varphi) + Z_0 \end{pmatrix}$$



Joint 6:

Similar to Joint 4, we have the translation vector for the ($x_1y_1z_1$) coordination:

$$d_G = [X_0, 0, Z_0]^T$$

We have the rotation matrix when rotating around the x matrix:

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

In the local frame:

$$r_p^B = [X_1, 0,0]^T$$

With $X_0 = 351, Z_0 = 791,\ X_1 = 82$

$r_p^G = A_x^{-1} r_p^B + d_G$

Finally, we get the equation: $r_p^G = [X_P, Y_p, Z_P]^T = \begin{pmatrix} X_0 + X_1 \\ 0 \\ Z_0 \end{pmatrix}$

3. KINEMATICS ANALYSIS

3.1. ROBOT VARIABLES, PARAMETERS & DENAVIT HARTENBERG TABLE

This 6-DOF robot manipulator has six links and all of them are revolute joints. By changing the angle of these revolute links, a robot can reach and fulfill its function as expected. Therefore, variables of the kinematics system of the introduced robot manipulator are these revolute joint angles and their starting value accepted as 0 (zero) at home position, and it should be bigger than -180° and less than or equal to 180°. To be simpler for kinematic analysis, we use the schematic diagram as Figure 7

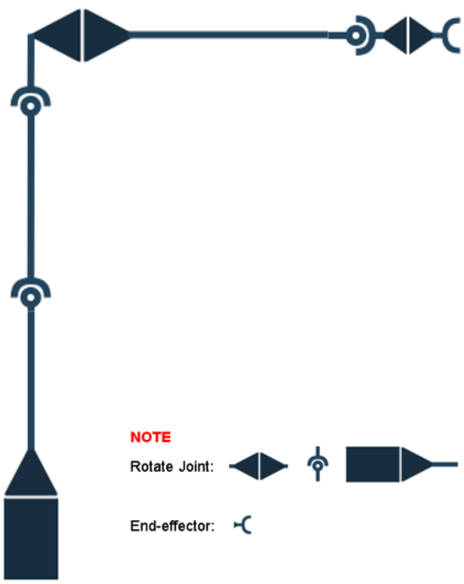


Figure 7: Model 2D of IRB 1200-7/0.7

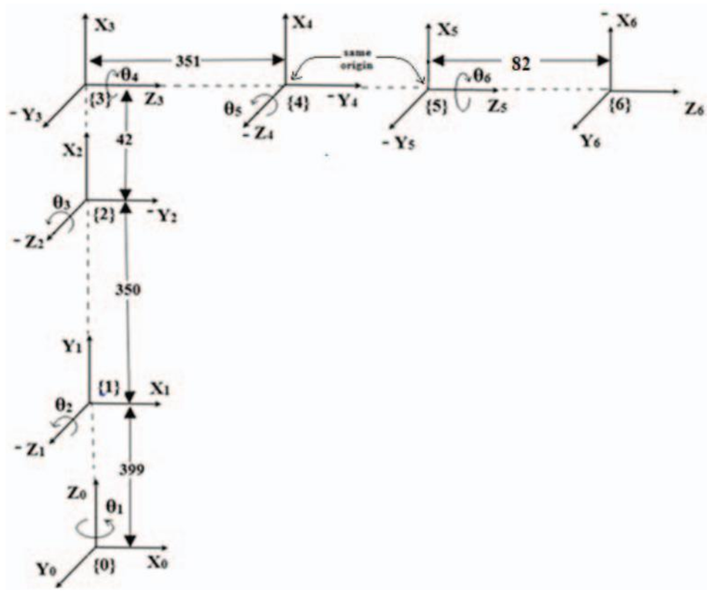


Figure 6: 2D Representation of robot manipulator with key dimensions

We place the local coordinates following rules below:

- z-axis is always the joint axis on which the joint rotates about if it is a rotational joint or moves along if it is a translational joint.
- x-axis must be perpendicular and intersect both new z & old z
- y-axis's placement follows the right-hand rule based on the x & z axes

Referring to Figure 6, beginning at joint 1, z_0 represents the first joint, which is a revolute joint. x_0 is chosen to be parallel to the reference frame x-axis. Next, z_1 is assigned at joint 2. x_1 will be normal to z_0 and z_1, since these two axes are intersecting. x_2 will be in the direction of the common normal between z_1 and z_2. So, there are infinite way to chose x_2 , we will chose follow the direction of the link connect joint 2 and 3. x_3 is in the direction of the common normal between z_2 and z_3. Similarly, x_4 is in the direction of the common normal between z_3 and z_4. Finally, z_5 and z_6 are in same direction because they are parallel and collinear. z_5 represent the motions of joint 6, while z_6 represents the motions of the end effectors.

When the diagram is obtained, we are then able to determine D-H parameters. D-H parameters include link length a_i, link twist α_i, link offset d_i and twist angle θ_i. The four parameters are defined as below:

- Link length a_i: distance between O_{i-1} and O_i measured along x_i axis.
- Link twist α_i: angle between z_{i-1} and z_i measured about x_i axis.
- Link offset d_i: distance between O_{i-1} and O_i measured along z_{i-1} axis.
- Twist angle θ_i: angle between x_{i-1} and x_i measured about z_{i-1} axis.

The assigned coordinate frames are followed to fill out the parameters as shown in below.

#	θ_i	d_i (mm)	a_i (mm)	α_i
1	θ_1	$d_1 = 399$	$a_1 = 0$	$\alpha_1 = -\pi/2$
2	$\theta_2 - \pi/2$	$d_2 = 0$	$a_2 = 350$	$\alpha_2 = 0$
3	θ_3	$d_3 = 0$	$a_3 = 42$	$\alpha_3 = -\pi/2$
4	θ_4	$d_4 = 351$	$a_4 = 0$	$\alpha_4 = \pi/2$
5	θ_5	$d_5 = 0$	$a_5 = 0$	$\alpha_5 = -\pi/2$
6	$\theta_6 - \pi$	$d_6 = 82$	$a_6 = 0$	$\alpha_5 = 0$

Table 2: DH Table for IRB 1200-7/0.7

3.2. FORWARD KINEMATICS

In 1955, Denavit and Hartenberg introduced a matrix-based approach for establishing coordinate systems on individual links within robot joint chains. This method was devised to articulate the translation or rotation relationships between adjacent links. This kinematic model for robots is rooted in the D-H (Denavit-Hartenberg) coordination system. The transformations between consecutive joints can be readily expressed by substituting the parameters from a designated table into the matrix.

$$^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For abbreviation and simplicity following notation substitutions will be used throughout this report:

$$c_n = \cos\theta_n$$

$$s_n = \sin\theta_n$$

$$c_{ab} = \cos a\cos b - \sin a\sin b$$

$$s_{ab} = \sin a\cos b + \cos a\sin b$$

where the upper left 3×3 rotation matrix which is used to determine three Euler angles, and fourth column of the matrix which represents the position vector (Cartesian coordinates). According to the general matrix, six homogenous transformation matrices are as below:

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

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

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

$$^{3}T_4 = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & 0 \\ \sin\theta_4 & 0 & -\cos\theta_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{4}T_5 = \begin{bmatrix} \cos\theta_5 & 0 & -\sin\theta_5 & 0 \\ \sin\theta_5 & 0 & \cos\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{5}T_6 = \begin{bmatrix} -\cos\theta_6 & \sin\theta_6 & 0 & 0 \\ -\sin\theta_6 & -\cos\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The total transformation matrix from the robot base to the hand is as follows:

$$^{0}T_6 = {}^{0}T_1\,{}^{1}T_2\,{}^{2}T_3\,{}^{3}T_4\,{}^{4}T_5\,{}^{5}T_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where:

- $n = [n_x \quad n_y \quad n_z]^T$ (normal vector) represents the direction of x_6 axis in the 0 frame.
- $o = [o_x \quad o_y \quad o_z]^T$ (orientation vector) represents the direction of y_6 axis in the 0 frame.
- $a = [a_x \quad a_y \quad a_z]^T$ (approach vector) represents the direction of z_6 axis in the 0 frame.
- $p = [p_x \quad p_y \quad p_z]^T$ represents the position of final point in the 0 frame.

In which:

$$n_x = -c_1 s_{23} c_4 c_5 c_6 - s_1 s_4 c_5 c_6 - c_1 c_{23} s_5 c_6 + c_1 s_{23} s_4 s_6 - s_1 c_4 s_6 \quad (1)$$

$$n_y = -s_1 s_{23} c_4 c_5 c_6 + c_1 s_4 c_5 c_6 - s_1 c_{23} s_5 c_6 - s_1 s_{23} s_4 s_6 + c_1 c_4 s_6 \quad (2)$$

$$n_z = -c_4 c_{23} c_5 c_6 + s_{23} s_5 c_6 + c_{23} s_4 s_6 \quad (3)$$

$$o_x = c_1 s_{23} c_4 c_5 s_6 + s_1 s_4 c_5 s_6 + c_1 c_{23} s_5 s_6 + c_1 s_{23} s_4 c_6 - s_1 c_4 c_6 \quad (4)$$

$$o_y = s_1 s_{23} c_4 c_5 s_6 - c_1 s_4 c_5 s_6 + s_1 c_{23} s_5 s_6 + s_1 s_{23} s_4 c_6 + c_1 c_4 c_6 \quad (5)$$

$$o_z = c_4 c_{23} c_5 c_6 - s_{23} s_5 s_6 + c_{23} s_4 c_6 \quad (6)$$

$$a_x = -c_1 s_{23} c_4 s_5 - s_1 s_4 s_5 + c_1 c_{23} c_5 \quad (7)$$

$$a_y = -s_1 s_{23} c_4 s_5 + c_1 s_4 s_5 + s_1 c_{23} c_5 \quad (8)$$

$$a_z = -s_5 c_4 c_{23} - s_{23} c_5 \quad (9)$$

$$p_x = -c_1 s_{23} c_4 s_5 d_6 - s_1 s_4 s_5 d_6 + c_1 c_{23} c_5 d_6 + c_1 c_{23} d_4 + a_3 c_1 s_{23} + a_2 c_1 s_2 \quad (10)$$

$$p_y = -s_1 s_{23} c_4 s_5 d_6 + c_1 s_4 s_5 d_6 + s_1 c_{23} c_5 d_6 + s_1 c_{23} d_4 + a_3 s_1 s_{23} + a_2 s_1 s_2 \quad (11)$$

$$p_z = -s_5 c_4 c_{23} d_6 - s_{23} c_5 d_6 - s_{23} d_4 + a_3 c_{23} + a_2 c_2 + d_1 \quad (12)$$

Example, to evaluate the accuracy of equation, we use the function of Matlab robotic toolbox for comparison.

Matlab code 1: Robotoolbox source code, the angles will be contained in the list variables named 'angles'

```
[a,alpha,d] = get_dh();
      %Link([theta, d, a, alpha, jointType(0: Rot; 1:Trans])
S(1) = Link([0   d(1)   a(1)   alpha(1)  0]);
S(2) = Link([0   d(2)   a(2)   alpha(2)  0]);
S(3) = Link([0   d(3)   a(3)   alpha(3)  0]);
S(4) = Link([0   d(4)   a(4)   alpha(4)  0]);
S(5) = Link([0   d(5)   a(5)   alpha(5)  0]);
S(6) = Link([0   d(6)   a(6)   alpha(6)  0]);
Rob = SerialLink(S);
Rob.name = '6R';
angle_offset = [pi/6 pi/3-pi/2 pi/6 pi/4 pi/3 0-pi];
a_point = Rob.A(1:6,angles);
```

Try to keys in the list of angles: Example: $\left(\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, 0\right)$

This is the robotic toolbox function result:

$$a_{point} = \begin{bmatrix} -0.483 & 0.2588 & -0.8365 & 230.3 \\ 0.1294 & 0.9659 & 0.2241 & 190.9 \\ 0.866 & 0 & -0.5 & 182 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This is the result of our computation:

Matlab code 2:

```
>> testfwd = myForward(pi/6, pi/3, pi/6, pi/4, pi/3, 0)
```

$$testfwd = \begin{bmatrix} -0.483 & 0.2588 & -0.8365 & 230.28 \\ 0.1294 & 0.9659 & 0.2241 & 190.93 \\ 0.866 & 0 & -0.5 & 182 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Matlab code 3:

```
>> postest = myForwardPos(pi/6, pi/3, pi/6, pi/4, pi/3, 0)
```

$$postest = \begin{bmatrix} 230.28 \\ 190.93 \\ 182 \\ 1 \end{bmatrix}$$

3.3. INVERSE KINEMATICS

In our previous session, we determined the end-effector's position in Cartesian space by providing joint rotation angles as input. However, in the real world, robots work in the opposite manner: the robot controller must ascertain the joint angles required to achieve a specific end-effector position. This process is accomplished through inverse kinematics calculations, which we will delve into in this session. The input variables for the inverse calculation are as follows:

$$^{0}T_6 = \begin{bmatrix} \mathrm{n}_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Before going to the solution for inverse kinematic, it is necessary to unified the sign represent for the length of links of the robot arm.

<u>*Joint 1:*</u>

From $^{0}T_6$, we have:

$$p'_x = p_x - d_6 a_x = d_4 c_1 c_{23} + a_3 c_1 s_{23} + a_2 c_1 s_2 \quad (13)$$

$$p'_y = p_y - d_6 a_y = d_4 s_1 c_{23} + a_3 s_1 s_{23} + a_2 s_1 s_2 \quad (14)$$

$$\rightarrow \frac{p'_y}{p'_x} = \frac{s_1(d_4c_{23}+a_3s_{23}+a_2s_2)}{c_1(d_4c_{23}+a_3s_{23}+a_2s_2)} \tag{15}$$

Hence, joint angle $\theta_1 is$ given by

$$\theta_1 = atan2(p'_y, p'_x) = atan2(p_y - d_6a_y, p_x - d_6a_x) \tag{16}$$

Joint 3:

We have:

$$p'_z = p_z - d_6a_z = -s_{23}d_4 + a_3c_{23} + a_2c_2 + d_1 \tag{17}$$

Squaring and adding $p'_x, p'_y, p'_z - d_1$ and rearranging

$$\frac{(p'_x)^2 + (p'_y)^2 + (p'_z - d_1)^2 - d_4^2 - a_3^2 - a_2^2}{2a_2} = a_3c_3 - d_4s_3$$

$$Let\ K = \frac{(p'_x)^2 + (p'_y)^2 + (p'_z - d_1)^2 - d_4^2 - a_3^2 - a_2^2}{2a_2} \rightarrow K = a_3c_3 - d_4s_3 \tag{18}$$

Solve this equation we have solution:

$$\theta_3 = \tan^{-1}\left(\frac{a_3}{d_4}\right) - \tan^{-1}\left(\frac{K}{\pm\sqrt{a_3^2 + d_4^2 - K^2}}\right) \tag{19}$$

Joint 2:

We have:

$${}^0T_3^{-1}\,{}^0T_6 = \begin{bmatrix} n_{11} & n_{12} & n_{13} & n_{14} \\ n_{21} & n_{22} & n_{23} & n_{24} \\ n_{31} & n_{32} & n_{33} & n_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

$${}^3T_6 = {}^3T_4\,{}^4T_5\,{}^5T_6 \qquad = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

Where:

$$n_{33} = c_1c_{23}a_x + s_1c_{23}a_y - s_{23}a_z \tag{23}$$

$$n_{34} = c_1c_{23}p_x + s_1c_{23}p_y - s_{23}p_z + a_2S_3 + d_1s_{23} \tag{24}$$

$$m_{33} = c_5 \tag{25}$$

$$m_{34} = d_6 c_5 + d_4 \tag{26}$$

From (23), (25):

$$m_{33} = n_{33} \Leftrightarrow c_5 = c_1 c_{23} a_x + s_1 c_{23} a_y - s_{23} a_z \tag{27}$$

From (24), (26):

$$m_{34} = n_{34} n_{34} \Leftrightarrow d_6 c_5 + d_4 = c_1 c_{23} p_x + s_1 c_{23} p_y - s_{23} p_z + a_2 S_3 + d_1 s_{23} \tag{28}$$

Substitute (27) in (28) and rearranging, we have:

$$c_{23}\left(c_1 p_x + s_1 p_y - d_6 c_1 a_x - d_6 s_1 a_y\right) - s_{23}(p_z - d_1 - d_6 a_z) = d_4 - a_2 s_3$$
$$\Leftrightarrow c_{23} M_1 - s_{23} M_2 = M \tag{29}$$

Where:

$$M_1 = c_1 p_x + s_1 p_y - d_6\left(c_1 a_x + s_1 a_y\right) \tag{30}$$

$M_2 = p_z - d_1 - d_6 a_z$

$M = -a_2 s_3 + d_4$

Solving equation (29), we have result:

$$\theta_{23} = atan2(M_1, M_2) - atan2\left(M, \pm\sqrt{M_1^2 + M_2^2 - M^2}\right)$$
$$\rightarrow \theta_2 = \theta_{23} - \theta_3 \tag{31}$$

Joint 4:

From (20), (21) with:

$$n_{13} = s_1 a_x - c_1 a_y \tag{32}$$

$$n_{23} = c_1 s_{23} a_x + s_1 s_{23} a_y + c_{23} a_z \tag{33}$$

$$m_{13} = -c_4 s_5 \tag{34}$$

$$m_{23} = -s_4 s_5 \tag{35}$$

From (32), (34):

$$m_{13} = n_{13} \Leftrightarrow -c_4 s_5 = s_1 a_x - c_1 a_y \tag{36}$$

From (33), (35):

$$m_{23} = n_{23} \Leftrightarrow -s_4 s_5 = c_1 s_{23} a_x + s_1 s_{23} a_y + c_{23} a_z \tag{37}$$

From (36), (37):

$$\rightarrow \theta_4 = atan2(s_4, c_4) \tag{38}$$

Joint 5:

From (27), we can calculate θ_5:

$$\theta_5 = \cos^{-1}\left(c_1 c_{23} a_x + s_1 c_{23} a_y - s_{23} a_z\right) \tag{39}$$

Joint 6:

From (20), (21) with:

$$n_{31} = c_1 c_{23} n_x + s_1 c_{23} n_y - s_{23} n_z \tag{40}$$

$$n_{32} = c_1 c_{23} o_x + s_1 c_{23} o_y - s_{23} o_z \tag{41}$$

$$m_{31} = -c_6 s_5 \tag{42}$$

$$m_{32} = s_6 s_5 \tag{43}$$

From (40), (42):

$$m_{31} = n_{31} \Leftrightarrow -c_6 s_5 = c_1 c_{23} n_x + s_1 c_{23} n_y - s_{23} n_z \tag{44}$$

From (41), (43):

$$m_{32} = n_{32} \Leftrightarrow s_6 s_5 = c_1 c_{23} o_x + s_1 c_{23} o_y - s_{23} o_z \tag{45}$$

From (36), (37):

$$\rightarrow \theta_6 = atan2(s_6, -c_6) \tag{46}$$

To test the inverse function, we will forward a list of joints angle, then reverse for comparison.

Matlab code 4: test the inverse function

```
testfwd = myForward(pi/6, pi/3, pi/6, pi/4, pi/3, 0);
postest = myForwardPos(pi/6, pi/3, pi/6, pi/4, pi/3, 0);
[theta1,theta2,theta3,theta4,theta5,theta6] = myInverse(testfwd);
test_inverse=rad2deg([theta1,theta2,theta3,theta4,theta5,theta6]);
```

Result:

$$test_{inverse} = [30, 60, 30, 45, 60, 0]$$

4. EXPERIMENT AND SIMULATION.

In theory, once we have derived the inverse kinematic equations, we gain the capability to control the robot arm. However, to prevent potential consequences like collisions and impacts resulting from erroneous equations, it is essential to undergo a process of simulation and validation before conducting real-world tests.

Using function support by robotic system toolbox, firstly, we verify the equation in this report with the function support by Matlab, the result of Forward is similar, the result of Inverse is sometimes different in angles, but when we do the forward kinematic with these angles, the state matrix is similar.

Matlab code 4: Robotoolbox source code support for test

```
[a,alpha,d] = get_dh();
      %Link([theta, d, a, alpha, jointType(0: Rot; 1:Trans])
S(1) = Link([0   d(1)   a(1)   alpha(1)  0]);
S(2) = Link([0   d(2)   a(2)   alpha(2)  0]);
S(3) = Link([0   d(3)   a(3)   alpha(3)  0]);
S(4) = Link([0   d(4)   a(4)   alpha(4)  0]);
S(5) = Link([0   d(5)   a(5)   alpha(5)  0]);
S(6) = Link([0   d(6)   a(6)   alpha(6)  0]);
Rob = SerialLink(S);
Rob.name = '6R';
angle_offset = [pi/6 pi/3-pi/2 pi/6 pi/4 pi/3 0-pi];
a_point = Rob.A(1:6,angles);
```

Also, we build a robot arm rely on the DH parameters of the IRB 1200-7/0.7 robot manipulator, control it using the forward and inverse kinematic function in this paper.

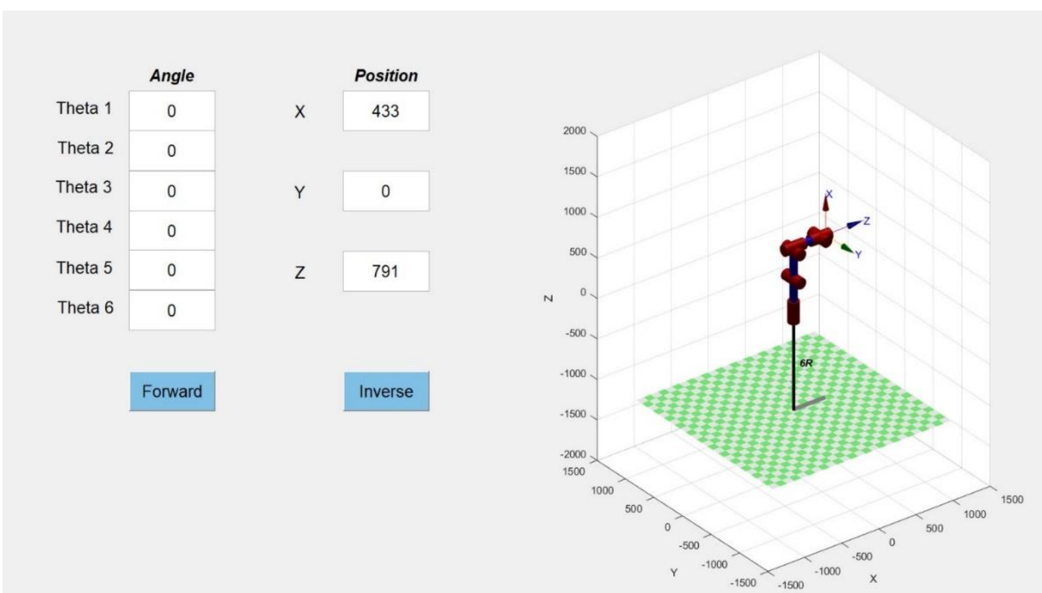


Figure 8: Simulation in Matlab robotic toolbox

5. CONCLUSION

In the present work, we investigate about model of IRB 1200-7/0.7 robot manipulator, include the technical information, structure of robot, the forward and inverse kinematics parametric analytical model was discussed. We also model the IRB 1200-7/0.7 robot in Matlab, then can understand how they word in their workspace.

6. REFERENCE

[1] Adrian-Florin NICOLESCU, Florentin-Marian ILIE, Tudor-George ALEXANDRU, " Forward and inverse kinematics study of industrial robots taking into account constructive and functional parameter's modeling" , *Proceedings in Manufacturing Systems, Volume 10, Issue 4, 2015, 157−164*.

[2] S. DİKMENLİ, "FORWARD & INVERSE KINEMATICS SOLUTION OF 6-DOF ROBOTS THOSE HAVE OFFSET & SPHERICAL WRISTS," *Eurasian Journal of Science Engineering and Technology,* vol. 3, no. 1, pp. 14-28, 2022.