Received: 10.04.2016
Accepted: 02.03.2017

Forward and Inverse Kinematic Analysis and Validation of the ABB IRB 140 Industrial Robot

Mohammed Almaged[1]

Abstract - The main goal of this paper is to derive the forward and inverse kinematic model of the ABB IRB 140 industrial manipulator. Denavit-Hartenberg analysis (DH) is presented to write the forward kinematic equations. Initially, a coordinate system is attached to each of the six links of the manipulator. Then, the corresponding four link parameters are determined for each link to construct the six transformation matrices (${}^{i-1}_{i}T$) that define each frame {i} relative to the previous one {i-1}. While, to develop the kinematics that calculates the required joint angles ($\theta_1 - \theta_6$), both geometrical and analytical approaches are used to solve the inverse kinematic problem. After introducing the forward and inverse kinematic models, a MATLAB code is written to obtain the solutions of these models. Then, the forward kinematics is validated by examining a set of known positions of the robot arm, while the inverse kinematics is checked by comparing the results obtained in MATLAB with a simulation in Robot Studio.

Keywords - *Robotics, forward kinematics, inverse kinematics, ABB IRB 140 manipulator*

1. Introduction

'Kinematics is the science of geometry in motion' [1]. This means it deals only with geometrical issues of motion such as the position and orientation regardless the force that causes them. There are two types of kinematics, the forward and inverse kinematics. Forward kinematic analysis is concerned with the relationship between the joint angle of the robot manipulator and the position and orientation of the end-effector [2]. In other words, it deals with finding the homogeneous transformation matrix that describes the position and orientation of the tool frame with respect to the global reference frame. On the other hand, inverse kinematics is used to calculate the joint angles required to achieve the desired position and orientation. The same transformation matrix which resulted from the forward kinematics in order to describe the position and the orientation of the tool frame relative to the robot base frame is used here in the inverse kinematics to solve for the joint angles.

The IRB 140, shown in Figure 1 below, is compact six axes (6 DOF) industrial manipulator. It is designed with six revolute joints providing a flexible use at an outstanding accuracy to be suitable for a wide range of applications such as welding, packing, assembly, etc.

[1] *Department of Mechatronics, Newcastle University, School of Mechanical & Systems Engineering Newcastle Upon Tyne, UK, m_engineer89@yahoo.com*

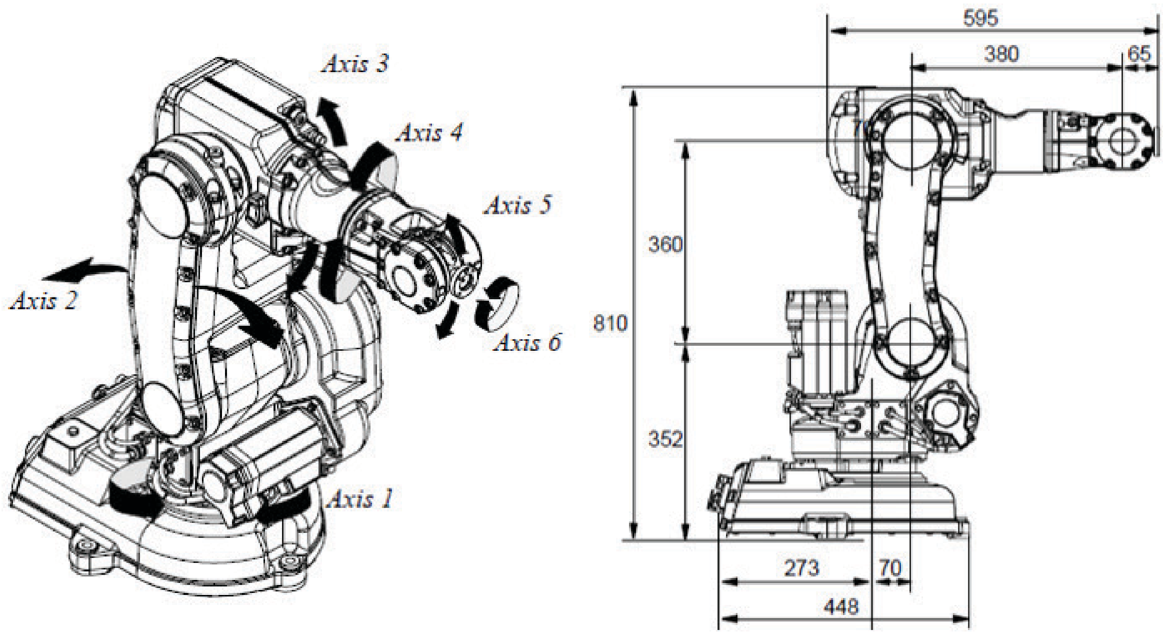

Figure 1. The ABB IRB 140 manipulator

2. Forward Kinematics

To mathematically model a robot and hence determine the position and orientation of the end effector with respect to the base or any other point, it is necessary to assign a global coordinate frame to the base of the robot and a local reference frame at each joint. Then, the Denavit-Hartenberg analysis (DH) is presented to build the homogeneous transformations matrices between the robot joint axes [3]. These matrices are a function of four parameters resulted from a series of translations and rotations around different axes. The illustration of how frame {i} is related to the previous frame {i −1} and the description of the frame parameters are shown in Figure 2 below.

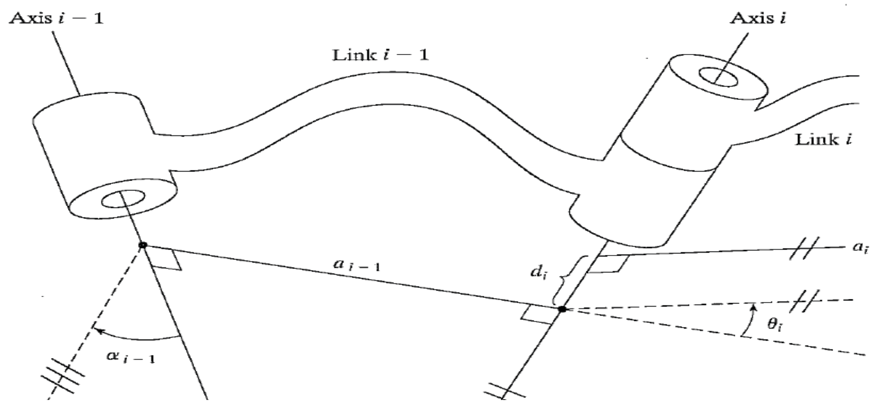

Figure 2. The description of frame {i} with respect to frame {i −1} [3]

From Figure 2, the modified D-H parameters can be described as:

- α_{i-1}: Twist angle between the joint axes Z_i and Z_{i-1} measured about X_{i-1}.
- a_{i-1}: Distance between the two joint axes Z_i and Z_{i-1} measured along the common normal.
- θ_i: Joint angle between the joint axes Xi and Xi-1 measured about Zi.
- d_i: Link offset between the axes Xi and Xi-1 measured along Zi.

Thus, the four Transformations between the two axes can be defined as:

$$ {}^{i-1}_{\;\;i}T = Rot(X_{i-1}, \alpha_{i-1}) \text{ x } Trans(X_{i-1}, a_{i-1}) \text{ x } Rot(Z_i, \theta_i) \text{ x } Trans(0,0,d_i) $$

After finishing the multiplication of these four transformation, the homogeneous transform can be obtained as:

$$ {}^{i-1}_{\;\;i}T = \begin{pmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & a_{i-1} \\ s_{\theta_i}c_{\alpha_{i-1}} & c_{\theta_i}c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & -d_i s_{\alpha_{i-1}} \\ s_{\theta_i}s_{\alpha_{i-1}} & c_{\theta_i}s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & d_i c_{\alpha_{i-1}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \textbf{(1.1)} $$

The ABB IRB140 frames assignment is shown below in Figure 3.

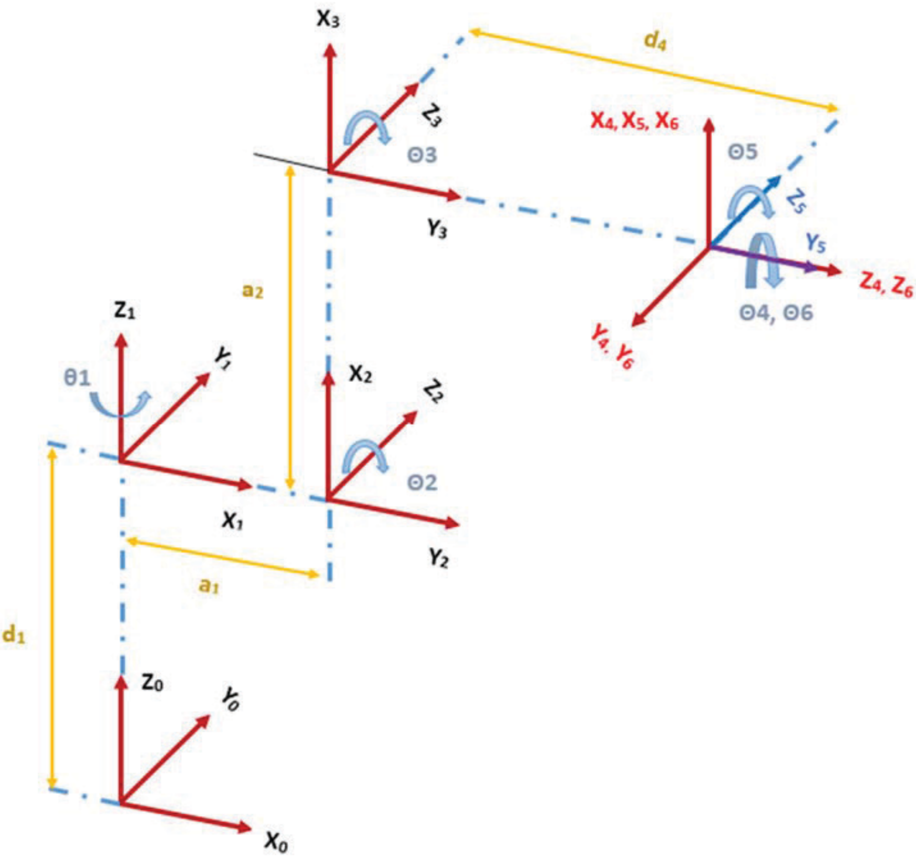

Figure 3. ABB IRB140 frames assignment

According to our particular frame assignment, the modified D-H parameters are defined in Table 1 below.

Table 1. Hata! Belgede belirtilen stilde metne rastlanmadı.. The ABB IRB 140 D-H parameters

Axis (i)	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	$d_1 = 352$	θ1
2	-90	$a_1 = 70$	0	θ2-90
3	0	$a_2 = 360$	0	θ3
4	-90	0	$d_4 = 380$	θ4
5	90	0	0	θ5
6	-90	0	0	θ6

For the simplicity of calculations and matrix product, it can be assumed that $s_2 = \sin(\theta 2\text{-}90)$, $c_2 = \cos(\theta 2\text{-}90)$. After achieving the D-H Table 1, the individual transformation matrix for each link is achieved by substituting the link parameters into the general homogeneous transform derived above in (1.1).

$$ {}^0_1T = \begin{pmatrix} c_{\theta_1} & -s_{\theta_1} & 0 & a_0 \\ s_{\theta_1}c_{\alpha_0} & c_{\theta_i}c_{\alpha_{01}} & -s_{\alpha_0} & -d_1 s_{\alpha_0} \\ s_{\theta_1}s_{\alpha_0} & c_{\theta_1}s_{\alpha_0} & c_{\alpha_0} & d_1 c_{\alpha_0} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^0_1T = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^1_2T = \begin{pmatrix} c_{\theta_2} & -s_{\theta_2} & 0 & a_1 \\ s_{\theta_2}c_{\alpha_1} & c_{\theta_2}c_{\alpha_1} & -s_{\alpha_1} & -d_2 s_{\alpha_1} \\ s_{\theta_2}s_{\alpha_1} & c_{\theta_2}s_{\alpha_1} & c_{\alpha_1} & d_2 c_{\alpha_1} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^{01}_{\ 2}T = \begin{pmatrix} c_2 & -s_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^2_3T = \begin{pmatrix} c_{\theta_3} & -s_{\theta_3} & 0 & a_2 \\ s_{\theta_3}c_{\alpha_2} & c_{\theta_3}c_{\alpha_2} & -s_{\alpha_2} & -d_3 s_{\alpha_2} \\ s_{\theta_3}s_{\alpha_2} & c_{\theta_3}s_{\alpha_2} & c_{\alpha_2} & d_3 c_{\alpha_2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^2_3T = \begin{pmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^3_4T = \begin{pmatrix} c_{\theta_4} & -s_{\theta_4} & 0 & a_3 \\ s_{\theta_4}c_{\alpha_3} & c_{\theta_4}c_{\alpha_3} & -s_{\alpha_3} & -d_4 s_{\alpha_3} \\ s_{\theta_4}s_{\alpha_3} & c_{\theta_4}s_{\alpha_3} & c_{\alpha_3} & d_4 c_{\alpha_3} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^3_4T = \begin{pmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^4_5T = \begin{pmatrix} c_{\theta_5} & -s_{\theta_5} & 0 & a_4 \\ s_{\theta_5}c_{\alpha_4} & c_{\theta_5}c_{\alpha_4} & -s_{\alpha_4} & -d_5 s_{\alpha_4} \\ s_{\theta_5}s_{\alpha_4} & c_{\theta_5}s_{\alpha_4} & c_{\alpha_4} & d_5 c_{\alpha_4} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^4_5T = \begin{pmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^5_6T = \begin{pmatrix} c_{\theta_6} & -s_{\theta_6} & 0 & a_5 \\ s_{\theta_6}c_{\alpha_5} & c_{\theta_6}c_{\alpha_5} & -s_{\alpha_5} & -d_6 s_{\alpha_5} \\ s_{\theta_6}s_{\alpha_5} & c_{\theta_5}s_{\alpha_5} & c_{\alpha_5} & d_6 c_{\alpha_5} \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad {}^5_6T = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Once the homogeneous transformation matrix of each link is obtained, forward kinematic chain can be applied to achieve the position and orientation of the robot end-effector with respect to the global reference frame (robot base).

$${}^0_2T = {}^0_1T \text{ X } {}^1_2T$$

$${}^0_2T = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{X} \begin{pmatrix} c_2 & -s_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_1c_2 & -c_1s_2 & -s_1 & c_1a_1 \\ s_1c_2 & -s_1s_2 & c_1 & s_1a_1 \\ -s_2 & -c_2 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0_3T = {}^0_2T \text{ X } {}^2_3T$$

$${}^0_3T = \begin{pmatrix} c_1c_2 & -c_1s_2 & -s_1 & c_1a_1 \\ s_1c_2 & -s_1s_2 & c_1 & s_1a_1 \\ -s_2 & -c_2 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0_3T = \begin{pmatrix} c_1c_2c_3 - c_1s_2s_3 & -(c_1c_2s_3 + c_1s_2c_3) & -s_1 & c_1c_2a_2 + c_1a_1 \\ s_1c_2c_3 - s_1s_2s_3 & -(s_1c_2s_3 + s_1s_2c_3) & c_1 & s_1c_2a_2 + s_1a_1 \\ -(s_2c_3 + c_2s_3) & s_2s_3 - c_2c_3 & 0 & -s_2a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0_3T = \begin{pmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & c_1(c_2a_2 + a_1) \\ s_1c_{23} & -s_1s_{23} & c_1 & s_1(c_2a_2 + a_1) \\ -s_{23} & -c_{23} & 0 & -s_2a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^4_6T = {}^4_5T \text{ X } {}^5_6T$$

$${}^4_6T = \begin{pmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_5c_6 & -c_5s_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^3_6T = {}^3_4T \text{ X } {}^4_6T$$

$${}^3_6T = \begin{pmatrix} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -s_4 & -c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_5c_6 & -c_5s_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^3_6T = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0_6T = {}^0_3T \text{ X } {}^3_6T$$

$${}^0_6T = \begin{pmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & c_1(c_2a_2 + a_1) \\ s_1c_{23} & -s_1s_{23} & c_1 & s_1(c_2a_2 + a_1) \\ -s_{23} & -c_{23} & 0 & -s_2a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & 0 \\ s_5\,c_6 & -s_5\,s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0_6T = \begin{pmatrix} r11 & \mathrm{r12} & \mathrm{r13} & \mathrm{x} \\ \mathrm{r21} & \mathrm{r22} & \mathrm{r23} & \mathrm{y} \\ \mathrm{r31} & \mathrm{r32} & \mathrm{r33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$r_{11} = c_1c_{23}\,(c_4c_5c_6 - s_4s_6) - c_1s_{23}s_5\,c_6 + s_1(s_4c_5c_6 + c_4s_6)$
$r_{12} = c_1c_{23}(-c_4c_5s_6 - s_4c_6) + c_1s_{23}s_5\,s_6 - s_1(s_4c_5s_6 - c_4c_6)$
$r_{13} = -c_1c_{23}c_4s_5 - c_1s_{23}c_5 - s_1s_4s_5$
$r_{21} = s_1c_{23}\,(c_4c_5c_6 - s_4s_6) - s_1s_{23}s_5\,c_6 - c_1(s_4c_5c_6 + c_4s_6)$
$r_{22} = s_1c_{23}(-c_4c_5s_6 - s_4c_6) + s_1s_{23}s_5\,s_6 + c_1(s_4c_5s_6 - c_4c_6)$
$r_{23} = -s_1c_{23}c_4s_5 - s_1s_{23}c_5 + c_1s_4s_5$
$r_{31} = -s_{23}\,(c_4c_5c_6 - s_4s_6) - c_{23}s_5\,c_6$
$r_{32} = -s_{23}\,(-c_4c_5s_6 - s_4c_6) + c_{23}s_5\,s_6$
$r_{33} = s_{23}c_4s_5 - c_{23}c_5$
$x = -d_4c_1s_{23} + c_1(c_2a_2 + a_1)$
$y = -d_4s_1s_{23} + s_1(c_2a_2 + a_1)$
$z = -s_2a_2 + d_1 - d_4c_{23}$

Now, it is also possible to find the position of the tip (TCP) with respect to the robot base. According to the robot frame assignment, it is simply a transition along the z axis of frame {6} by d6 (65 mm). Therefore, the final position of the end effector with respect to the robot global reference frame can be expressed as:

$\text{Ptip} = {}^0_6T\ X\ \text{P}^6$

$$\text{Ptip} = \begin{pmatrix} r11 & \text{r12} & \text{r13} & \text{x} \\ \text{r21} & \text{r22} & \text{r23} & \text{y} \\ \text{r31} & \text{r32} & \text{r33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} 0 \\ 0 \\ \text{d6} \\ 1 \end{pmatrix} = \begin{pmatrix} \text{d6 X r13} + \text{x} \\ \text{d6 X r23} + \text{y} \\ \text{d6 X r33} + \text{z} \\ 1 \end{pmatrix}$$

3. Forward Kinematic Validation

After finding the homogeneous transformation matrix (${}^0_6T$) that describes the end effector position and orientation with respect to the robot global reference frame, the position of the robot in space is expressed by the vector ${}^0P_{6ORG}$ which gives the values of x, y and z vectors as follow:

$x = -d_4c_1s_{23} + c_1(c_2a_2 + a_1)$
$y = -d_4s_1s_{23} + s_1(c_2a_2 + a_1)$
$z = -s_2a_2 + d_1 - d_4c_{23}$

ı that: $S_2 = \sin(\theta 2\text{-}90)$, $C_2 = \cos(\theta 2\text{-}90)$, d1 = 352 mm,
d4 = 380 mm, a1 = 70 mm and a2 = 360 mm.

These equations are programmed in Matlab and a set of eight positions, illustrated below in Figure 4, were chosen randomly to validate the forward kinematic model. The joint angles of each position are entered manually by the user to obtain the x, y and z vectors as shown in Table 2 below. It can be clearly seen that there is no y component corresponding to these particular positions because Θ1 is always given to be zero. The same joint angle values were entered through the robot operating software in the lab and the results were similar to the x, y and z vectors obtained from Matlab which proves the validity of this model.

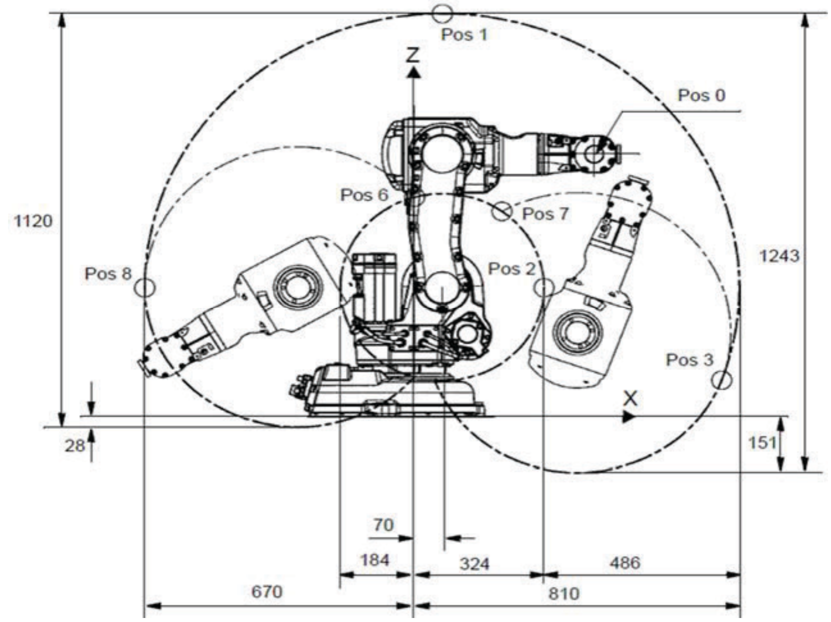

Figure 4. Set of different robot positions

4. Inverse Kinematics

Inverse kinematics is used to calculate the joint angles required to achieve the desired position and orientation in the robot workspace. In general, there are two methods of solution, the analytical and geometrical approaches. Since three consecutive axes of the robot intersect at a common point, Pieper's solution can be applied. Pieper's approach works on the principle of separating the position solution for ϴ1, ϴ2 and ϴ3 from the orientation solution to solve for ϴ4, ϴ5 and ϴ6 [4]. Therefore, a geometrical approach is initially implemented to find the joint variables ϴ1, ϴ2 and ϴ3 that define the end effector position in space, while an analytical solution is applied to calculate the angles ϴ4, ϴ5 and ϴ6 which describe the end-effector orientation.

4.1 Geometrical solution

According to the frame assignment shown in Figure 1, x and y components of frame {1} is the same as frame {0} because there is only a Z-directional offset between the two frames.

Table 2. Numerical calculation for the values x, y and z of each positions

Position	Joint angles	X vector	Y vector	Z vector
0	ϴ1 = 0, ϴ2 = 0, ϴ3 = 0	450	0	712
1	ϴ1 = 0, ϴ2 = 0, ϴ3 = -90	70	0	1092
2	ϴ1 = 0, ϴ2 = 0, ϴ3 = 50	314	0	420.9
3	ϴ1 = 0, ϴ2 = 110, ϴ3 = -90	765	0	98.9
6	ϴ1 = 0, ϴ2 = -90, ϴ3 = 50	1.1	0	596
7	ϴ1 = 0, ϴ2 = 110, ϴ3 =-230	218	0	558
8	ϴ1 = 0, ϴ2 = -90, ϴ3 = -90	-670	0	352

Therefore, the projection of the wrist components on x-y plane of frame {0} has the same components on frame {1} [5, 6]. In addition, since both link two and three are planar, the position vector in y direction changes with respect to θ1 only. Thus, two possible solutions for θ1 can be achieved by simply applying the arctangent function.

$$\theta 1 = \operatorname{atan2}(Py, Px), \tag{4.1}$$
$$\theta 11 = \Pi + \theta 1. \tag{4.2}$$

The solutions of θ2 and θ3 are obtained by considering the plane, shown in Figure 5, formed by the second and third planar links with respect to the robot global reference frame.

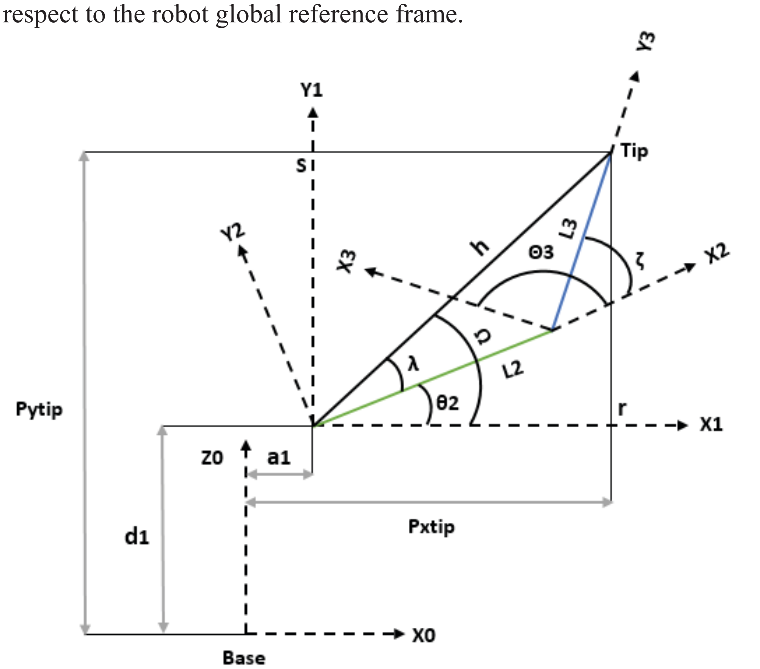

Figure 5. Projection of links two and three onto the x y plane

The cosine low is used to solve for θ3 as follow:

$$h^2 = (L_2)^2 + (L_3)^2 — 2 \text{ x } L_2 \text{ x } L_3 \cos(180 — \zeta)$$

Since the position is given with respect to the robot tip (TCP), L_3 should be equal to d4 + d6.
While, $L_2 = a_2$, $h^2 = s^2 + r^2$, $\cos(180 - \zeta) = -\cos(\zeta)$.

$$s^2 + r^2 = (a_2)^2 + (d_4 + d_6)^2 + 2 \text{ x } a_2 \text{ x } (d_4 + d_6) \cos(\zeta)$$
$$\operatorname{Cos}(\zeta) = \frac{[s2 + r2 — (a2)2 — (d4 + d6)\ 2]\ ]}{2 \text{ x a2 x } (d4 + d6)} \tag{4.3}$$

Now, we should have the value of **(s)** and **(r)** in term of $\mathbf{P_{xtip}}$, $\mathbf{P_{ytip}}$, $\mathbf{P_{ztip}}$ and **θ1**.
$S = (P_{ztip} — d1)$

$r = \pm\sqrt{(\text{Pxtip} — \text{a1} \cos(\theta 1))2 + (\text{Pytip} — \text{a1} \sin(\theta 1))2}$, Sub. **(s)** and (r) in (4.3) yield:

$$\operatorname{Cos}(\zeta) = \frac{[(\text{Pztip} — \text{d1})2 + (\text{Pxtip} — \text{a1}\cos(\theta 1))2 + (\text{Pytip} — \text{a1}\sin(\theta 1))2 — (\text{a2})2 — (\text{d4} + \text{d6})\ 2]}{2 \text{ x a2 x } (\text{d4} + \text{d6})}$$

$$\operatorname{Sin}(\zeta) = \pm\sqrt{1 - \operatorname{Cos2}(\zeta)}$$

$$\zeta = \operatorname{atan2}(\operatorname{Sin}(\zeta), \operatorname{Cos}(\zeta))$$

Finally, $\theta 3 = -(90 + \zeta)$ **(4.4)**

The negative sign in θ3 indicates that the rotation occurred in the opposite direction. Likewise, we can follow the same procedure to solve for θ2 using similar trigonometric relationships.

$\theta2 = \Omega - \lambda$
$\Omega = \text{atan2}\,(s, r)$
$\lambda = \text{atan2}\,((d_4+d_6)\sin(\zeta), a_2 + (d_4+d_6)\cos(\zeta))$
$\theta2 = \text{atan2}\,(s, r) - \text{atan2}\,[(d_4+d_6)\sin(\zeta), a_2 + (d_4+d_6)\cos(\zeta)]$, sub the values of (s) and (r) yield:
$\theta2 = \text{atan2}\,[(P_{ztip} - d1), \pm\sqrt{(Pxtip - a1\cos(\theta1))2 + (Pytip - a1\sin(\theta1))2}]$
$\quad - \text{atan2}\,[(d_4+d_6) \text{ x } \sin(\zeta), a_2 + (d_4+d_6)\cos(\zeta)]$.

Again the rotation occurred in the opposite direction of the z axis as well as there are an initial rotation of 90^0 between axis 1 and axis 2. Therefore, the final value of θ2 equal to:

$$\theta2 = -((\Omega - \lambda) - 90). \quad \textbf{(4.5)}$$

It is important to say that any position within the robot workspace can be achieved with many orientations. Therefore, multiple solutions exist for the variables ϴ1, ϴ2 and ϴ3 due to the nature of trigonometric functions.

As noticed above, every solution step resulted in two values that will be used in the next step, and so on. For example, there are four solutions for ζ that resulted from two different values of θ1 (θ1 and θ11), this procedure gives four solutions for θ3, each solution corresponds to different robot configurations of elbow-up and elbow-down representations. These solutions can be listed in Table 3 below to illustrate all the possible solution set.

Table 3. Possible solution set

Solution	THETA1	THETA3	THETA2	Set
1	ϴ1	ϴ3	ϴ2	SET 1
2	ϴ1	ϴ3	ϴ22	
3	ϴ1	ϴ33	ϴ2i	SET 2
4	ϴ1	ϴ33	ϴ22i	
5	ϴ11	ϴ3i	ϴ2j	SET 3
6	ϴ11	ϴ3i	ϴ22j	
7	ϴ11	ϴ33i	ϴ2k	SET 4
8	ϴ11	ϴ33i	ϴ22k	

4.2 Analytical solution

After solving the first inverse kinematic sub-problem which gives the required position of the end effector, the next step of the inverse kinematic solution will deal with the procedure of solving the orientation sub-problem to find the joint angles ϴ4, ϴ5 and ϴ6. This can be done using Z-Y-X Euler's formula. As the orientation of the tool frame with respect to the robot base frame is described in term of Z-Y-X Euler's rotation, this means that each rotation will take place about an axis whose location depends on the previous rotation [3]. The Z-Y-X Euler's rotation is shown below in Figure 6.

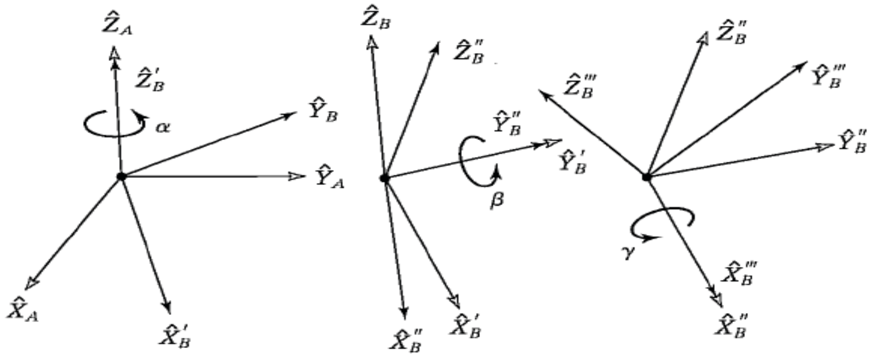

Figure 6. Z—Y—X Euler rotation [3]

The final orientation matrix that results from these three consecutive rotations will be as follow:

$${}^0_6R = R_{z'y'x'} = R_z(\alpha)\, R_y(\beta)\, R_x(\gamma)$$

$${}^0_6R = \begin{pmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{pmatrix} X \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{pmatrix}$$

$${}^0_6R = \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix}$$

Recall the forward kinematic equation,

$${}^0_3R = \begin{pmatrix} c_1 c_{23} & -c_1 s_{23} & -s_1 \\ s_1 c_{23} & -s_1 s_{23} & c_1 \\ -s_{23} & -c_{23} & 0 \end{pmatrix}$$

$${}^3_6R = ({}^0_3R)^T \; {}^0_6R$$

$${}^3_6R = \begin{pmatrix} c_1 c_{23} & s_1 c_{23} & -s_{23} \\ -c_1 s_{23} & -s_1 s_{23} & -c_{23} \\ -s_1 & c_1 & 0 \end{pmatrix} x \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix}$$

$${}^3_6R = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}$$

However, it can be concluded that the last three intersected joints form a set of ZYZ Euler angles with respect to frame {3}. Therefore, these rotations can be expressed as:

$$R_{z'y'z'} = {}^3_6R = R_z(\alpha)\, R_y(\beta)\, R_z(\gamma)$$

$${}^3_6R = \begin{pmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} X \begin{pmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{pmatrix} X \begin{pmatrix} c_\gamma & -s_\gamma & 0 \\ s_\gamma & c_\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$${}^3_6R = \begin{pmatrix} c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ s_\alpha c_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{pmatrix}$$

Where ${}^3_6R$ is given above as

$${}^{3}_{6}R = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}$$

It is possible now to use the ZYZ Euler's angles formula to obtain the solutions for Θ4, Θ5 and Θ6 where

$$\theta_5 = \beta = atan2\left(+\sqrt{g_{31}^2 + g_{32}^2}, g_{33}\right)$$

$$\theta_4 = \alpha = atan2\left(\frac{g_{32}}{s_\beta}, \frac{-g_{31}}{s_\beta}\right)$$

$$\theta_6 = \gamma = atan2\left(\frac{g_{23}}{s_\beta}, \frac{g_{13}}{s_\beta}\right)$$

For each of the eight solutions achieved from the geometric approach for Θ1, Θ2 and Θ3, there is another flipped solution of Θ4, Θ5 and Θ6 that can be obtained as:

$\theta_{55} = \beta' = atan2\left(-\sqrt{g_{31}^2 + g_{32}^2}, g_{33}\right)$, Or simply Θ55 = − Θ5

$\theta_{44} = \alpha = atan2\left(\frac{g_{32}}{s_{\beta'}}, \frac{-g_{31}}{s_{\beta'}}\right)$, Or simply Θ44 = 180 + Θ5

$\theta_{66} = \gamma = atan2\left(\frac{g_{23}}{s_{\beta'}}, \frac{g_{13}}{s_{\beta'}}\right)$, Or simply Θ66 = 180+ Θ6

Now, if β = 0 or 180, this means that the robot in a singular configuration where the joint axes 4 and 6 are parallel. This results in a similar motion of the last three intersection links of the robot manipulator.

Alternatively:

If β = θ_5 = 0, the solution will be

$\theta_4 = \alpha = 0$

$\theta_6 = \gamma = atan2\,(-g12, g11)$

If β = θ_5 = 180, the solution will be

$\theta_4 = \alpha = 0$

$\theta_6 = \gamma = atan2\,(g12, -g11)$

5. Inverse Kinematic Validation

The home position of the robot in space is chosen to check the validity of the inverse kinematic solution. This position can be represented by a point (P_{tip}) in the robot workspace. This point describes the position of the end effector (TCP) with respect to the robot base frame. By applying the inverse kinematic equations derived above, a set of joint angles is achieved. However, some of these angles do not yield a valid solution which is simply due to the fact that not all the joints can be rotated by 360^0.

P_{tip} (Home Position) = $[pxtip\ \ pytip\ \ pztip]^T$ = $[515\ 0\ 712]^T$

After performing the calculations in MATLAB, four sets of solution were obtained as follow:

Table 4. Inverse kinematic solution sets

Θ1	Θ3	Θ2	Set
0	-180	102	SET 1
0	-180	0	
0	0	0	SET 2
0	0	**-102**	
180	-153	93.7	SET 3
180	-153	-23	
180	-27	23	SET 4
180	-27	**-93.7**	

However, because of the limitation on the joint angle range of movement [7], especially joints 2 and 3, some of these solutions (marked in **red**) are not valid. The ABB IRB 140 joint angle limits are listed below in Table 5.

Table 5. ABB IRB 140 joint angle limits [7]

Joint Angle	MAX	MIN
ϴ1	180	-180
ϴ2	110	-90
ϴ3	50	-230
ϴ4	200	-200
ϴ5	115	-115
ϴ6	400	-400

After checking all the possible solutions with joint angle limitation table, only three valid solutions [(0, 0, 0), (180, -23, -153), (0, 102, -180)] were achieved which represent different robot configurations of the home position, elbow-up and elbow-down representations. The elbow-up configuration that corresponds to joint angles (180, -23, -153) is shown in Figure 7 below, while Figure 8 shows the elbow-down configuration that corresponds to joint angles (0, 102, and 180). Finally, the set (0, 0, 0) represents the home position by default. It is important to note that the position vector in Robot Studio is given for the TCP with respect to the robot global reference frame. Thus to match our solution with the simulation in Robot Studio, the inverse kinematics was solved with respect to the robot TCP.

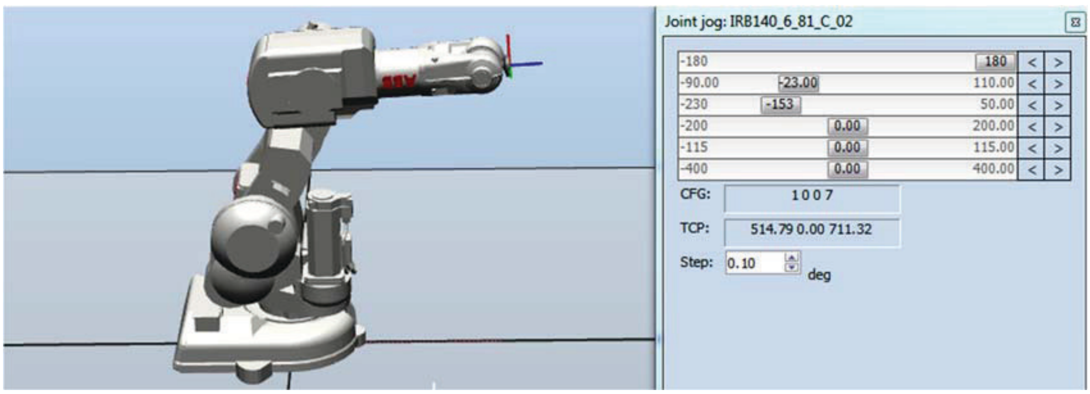

Figure 7. Elbow-up configuration

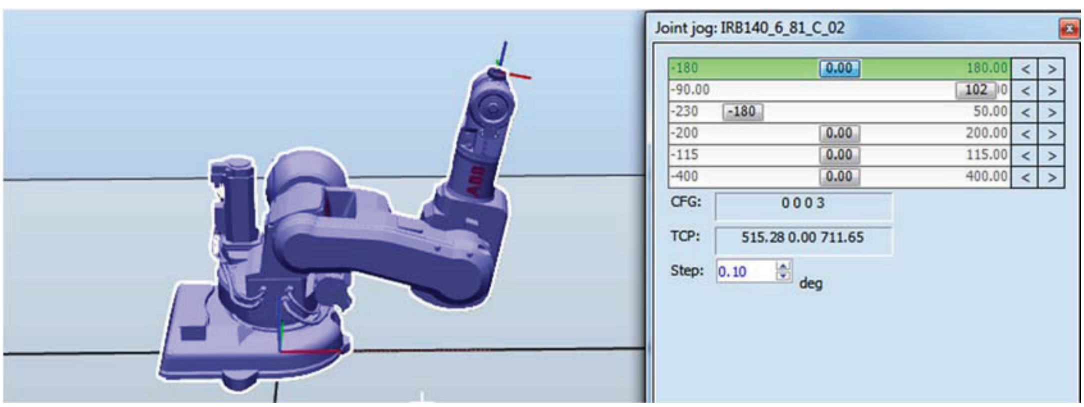

Figure 8. Elbow-down configuration

6. Conclusion

This work was undertaken to build the forward and inverse kinematic models of the ABB IRB 140 industrial manipulator. The Denavit-Hartenberg analysis (DH) is introduced to form the homogeneous transformation matrices. From the derived kinematic equations, it can be concluded that the position of the robot is given as a function of ϴ1, ϴ2 and ϴ3 only, while the three last intersection joint angles (ϴ4, ϴ5 and ϴ6) are used to give the desired orientation in space. The position vectors (x, y and z) obtained from the kinematic equations were matched with the actual robot position in the lab for the same joint angle input. Therefore, it can be declared that the kinematic derivation was carried out successfully. Two approaches have been presented to solve the inverse kinematic problem. Those were the geometrical and analytical approaches. Multiple solutions have been produced due to the nature of trigonometric functions. However, it has been shown that not all the solutions that resulted from the inverse kinematics were valid. This is basically due to the physical restrictions on the joint angle range of movement. A simulation of the manipulator in Robot Studio has been introduced to prove the validity of the inverse kinematic model. It is also used to validate the written Matlab code.

References

[1] R. N. Jazar, *Theory of applied robotics* vol. 1: Springer, 2010.
[2] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control* vol. 3: Wiley New York, 2006.
[3] J. J. Craig, *Introduction to robotics: mechanics and control*: Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2005.
[4] J. N. Pires, *Industrial robots programming: building applications for the factories of the future*: Springer, 2007.
[5] T. J. Carter, "The Modeling of a Six Degree-of-freedom Industrial Robot for the Purpose of Efficient Path Planning," The Pennsylvania State University, 2009.
[6] D. B. Vicente, *Modeling and Balancing of Spherical Pendulum Using a Parallel Kinematic Manipulator*, 2007.
[7] ABB Robotics, *IRB 140 M2000 Product Specification*, Retrieved from http://www.abb.com/ on December 01, 2015.

Appendices

I. Forward kinematics script

```
% THIS PROGRAM IS USED TO SOLVE THE FORWARD KINEMATIC OF THE ABB IRB140

% NON RETURN FUNCTION OF THE MAIN PROGRAM TO COMBINE ALL THE FUNCTIONS
    TOGETHER IN ONE SCRIPT

function [ NONRETURNFN ] = FORWARD( )

% DECLARATION OF THE MDH PARAMETERS

a0 = 0;      d1 = 352;    alpha0 = 0;
a1 = 70;     d2 = 0;      alpha1 = -pi/2;
a2 = 360;    d3 = 0;      alpha2 = 0;
a3 = 0;      d4 = 380;    alpha3 = -pi/2;
a4 = 0;      d5 = 0;      alpha4 = pi/2;
a5 = 0;      d6 = 0;      alpha5 = -pi/2;

% USER INTERFACE

theta1 = input ('PLEASE ENTER THE VALUE OF THETA1 IN DEGREE = ');
theta2 = input ('PLEASE ENTER THE VALUE OF THETA2 IN DEGREE = ');
theta3 = input ('PLEASE ENTER THE VALUE OF THETA3 IN DEGREE = ');
```

```
theta4 = input ('PLEASE ENTER THE VALUE OF THETA4 IN DEGREE = ');
theta5 = input ('PLEASE ENTER THE VALUE OF THETA5 IN DEGREE = ');
theta6 = input ('PLEASE ENTER THE VALUE OF THETA6 IN DEGREE = ');

% CALL THE DH FUNCTION TO CALCULATE THE HOMOGENOUS TRANSFORMATION MATRICES

T10 = DHFUNCTION(a0,alpha0,d1,theta1*pi/180)
T21 = DHFUNCTION(a1,alpha1,d2,(theta2-90)*pi/180)
T32 = DHFUNCTION(a2,alpha2,d3,theta3*pi/180)
T43 = DHFUNCTION(a3,alpha3,d4,theta4*pi/180)
T54 = DHFUNCTION(a4,alpha4,d5,theta5*pi/180)
T65 = DHFUNCTION(a5,alpha5,d6,theta6*pi/180)
T20 = T10*T21;
T30 = T20*T32;
T64 = T54*T65;
T63 = T43*T64;
T60 = T30*T63

% THE POSTION OF THE END EFEECTOR AT JOINT 6

Xw = T60(1,4);
Yw = T60(2,4);
Zw = T60(3,4);
P6 = [Xw;Yw;Zw]

% THE POSTION OF THE END EFEECTOR AT THE TCP

PTCP= T60*[0;0;65;1]

% Modifed DH TRANSFORM FUNCTION

function T = DHFUNCTION(ai,alphai,di,thetai)

T = [ cos(thetai),                -1.*sin(thetai),            0,                 ai ;

      sin(thetai).*cos(alphai),   cos(thetai).*cos(alphai),   -1.*sin(alphai),   1*di.*sin(alphai);

      sin(thetai).*sin(alphai),   cos(thetai).*sin(alphai),   cos(alphai),       di.*cos(alphai);

      0,                          0,                          0,                 1          ];

end

end
```

II. Inverse kinematics script

```
% THIS PROGRAM IS USED TO SOLVE THE INVERSE KINEMATIC OF THE ABB IRB 140

% DEFINE A NON RETURN FUNCTION TO COMBINE ALL THE INVERSE FUNCTIONS TOGETHER
  IN ONE SCRIPT

function [ NONRETURNFUNCTION] = INVERSE(  )
```

```
% DECLERATION OF THE ROBOT PARAMETER
d1 = 352;
a1 = 70;
a2 = 360;
d4 = 380;
NOSOLUTION=1000;

% THIS PROGRAM IS DESIGNED TO SOLVE THE INVERSE WITH RESPECT TO Porg6 OR TCP
ACCORDING TO USER SELECTION
sel = input ('TO SOLVE THE INVERSE WITH RESPECT TO FRAME6 PRESS 1 WHILE, TO SOLVE THE
INVERSE WITH RESPECT TO TCP ENTER 2: ');
if (sel == 1)
d6 = 0;
elseif (sel == 2)
d6 = 65;
else
d6= 65;
end

% USER INTERFACE

xtip = input ('PLEASE ENTER THE GOAL POSTION X = ');
ytip = input ('PLEASE ENTER THE GOAL POSTION y = ');
ztip = input ('PLEASE ENTER THE GOAL POSTION z = ');
alpha= input ('PLEASE ENTER THE VALUE OF alpha IN DEGREE = ');
beta = input ('PLEASE ENTER THE VALUE OF beta IN DEGREE  = ');
gama = input ('PLEASE ENTER THE VALUE OF gama IN DEGREE  = ');

% CALCULATING ALL THE POSSIBLE VALUES FOR THETA1

theta1= atan2 (ytip,xtip);
theta11= pi + theta1;
THETA1 = theta1 * 180/pi;
THETA11= theta11 * 180/pi;

% CALCULATING ALL THE POSSIBLE VALUES FOR THETA3

s = (ztip - d1);
r =  sqrt((xtip - a1*cos (theta1))^2 +(ytip - a1*sin(theta1))^2);
czeta = (r^2 + s^2  - (a2)^2 - (d4 + d6)^2)/(2 * a2 *(d4 + d6));

% SINGULARTIY CONDTION, CHECK IF THE POSTION WITHIN THE WORKSPACE OR NOT

if (abs(czeta) <= 1)
szeta = sqrt(1-(czeta)^2);
szeta1 = -szeta;
zeta= atan2(szeta,czeta);
zeta1= atan2(szeta1,czeta);
theta3 = -(pi/2 + zeta);
theta33 = -(pi/2 + zeta1);
THETA3 = conversion( theta3,50,-230);
THETA33 = conversion( theta33,50,-230);
else
theta3 = NOSOLUTION;
```

```
theta33= NOSOLUTION;
THETA3 = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR THETA3');
THETA33 = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR  THETA33');
End

s = (ztip - d1);
r =  sqrt((xtip - a1*cos (theta11))^2 +(ytip - a1*sin(theta11))^2);
czetai = (r^2 + s^2  - (a2)^2 - (d4 + d6)^2)/(2 * a2 *(d4 + d6));

% SINGULARTIY CONDTION, CHECK IF THE POSTION WITHIN THE WORKSPACE OR NOT

if (abs(czetai) <= 1)
szetai = sqrt(1-(czetai)^2);
szeta1i = -szetai;
zetai= atan2(szetai,czetai);
zeta1i= atan2(szeta1i,czetai);
theta3i = -(pi/2 + zetai);
theta33i = -(pi/2 + zeta1i);
THETA3i = conversion( theta3i,50,-230);
THETA33i = conversion( theta33i,50,-230);
else
theta3i=NOSOLUTION;
theta33i=NOSOLUTION;
THETA3i = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR THETA3i');
THETA33i = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR  THETA33i');
end

% CALCULATING ALL THE POSSIBLE VALUES FOR THETA2

if (theta3 == NOSOLUTION)
THETA2 = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR THETA2');
THETA22 = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR  THETA22');
else
theta2 = THE2(xtip,ytip,ztip,theta1,zeta);
theta22 = THE2COMP(xtip,ytip,ztip,theta1,zeta);
THETA2 = conversion( theta2,110,-90);
THETA22 = conversion( theta22,110,-90);
end

if (theta33 == NOSOLUTION)
THETA2i = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR THETA2i');
THETA22i = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR  THETA22i');
else
theta2i = THE2(xtip,ytip,ztip,theta1,zeta1);
theta22i = THE2COMP(xtip,ytip,ztip,theta1,zeta1);
THETA2i = conversion( theta2i,110,-90);
THETA22i = conversion( theta22i,110,-90);
end

if (theta3i == NOSOLUTION)
THETA2j = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR THETA2j');
THETA22j = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR  THETA22j');
else
theta2j = THE2(xtip,ytip,ztip,theta11,zetai);
```

```
theta22j = THE2COMP(xtip,ytip,ztip,theta11,zetai);
THETA2j = conversion( theta2j,100,-90);
THETA22j = conversion( theta22j,100,-90);
end

if (theta33i == NOSOLUTION)
THETA2k = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR THETA2k');
THETA22k = ('GOAL OUT OF WORKSPACE, THERE IS NO VAILD VALUS FOR  THETA22k');
else
theta2k = THE2(xtip,ytip,ztip,theta11,zeta1i);
theta22k = THE2COMP(xtip,ytip,ztip,theta11,zeta1i);
THETA2k = conversion( theta2k,110,-90);
THETA22k = conversion( theta22k,110,-90);
end

% DISPLAY ALL THE POSSIBLE EIGHT SOLUTIONS, NOTE THAT EVERY TWO SOLUTIONS FORM
   ONLY ONE SOLUTION SET

disp ( ' THETA 1,2,3 SOLUTIONS')
disp ( ' SET 1')
SOL1 =[ THETA1, THETA2, THETA3]
SOL2 =[ THETA1, THETA22, THETA3]
disp ( ' SET 2')
SOL3 =[ THETA1, THETA2i, THETA33]
SOL4 =[ THETA1, THETA22i, THETA33]
disp ( ' SET 3')
SOL5 =[ THETA11, THETA2j, THETA3i]
SOL6 =[ THETA11, THETA22j, THETA3i]
disp ( ' SET 4')
SOL7 =[ THETA11, THETA2k, THETA33i]
SOL8 =[ THETA11, THETA22k, THETA33i]

% SOLVING THE SECOND KINEMATIC SUB-PROBLEM (ORIENTATION)

alpha = alpha * pi/180;
beta = beta * pi/180;
gama = gama * pi/180;

R60 = [cos(alpha).*cos(beta),           (cos(alpha).*sin(beta).*sin(gama))- sin(alpha).*cos(gama),
       (cos(alpha).*sin(beta).*cos(gama)) + sin(alpha).*sin(gama) ;

        sin(alpha).*cos(beta),          (sin(alpha).*sin(beta).*sin(gama)) + cos(alpha).*cos(gama),
        (sin(alpha).*sin(beta).*cos(gama)) - cos(alpha).*sin(gama) ;

        - sin (beta),                        cos (beta).*sin (gama),           cos (beta).*sin (gama)]


R30 = [cos(theta1).*cos(theta2+theta3),              -cos(theta1).*sin(theta2+theta3),                      sin(theta1);

        sin (theta1).*cos(theta2+theta3),            -sin(theta1).*sin(theta2+theta3),                      cos(theta1);

       -sin(theta2+theta3),                          -cos(theta2+theta3),                                    0     ] ;
```

```
RT30= transpose (R30);
R63 = RT30 * R60 ;
g11 = R63 (1,1);
g12 = R63 (1,2);
g23 = R63 (2,3);
g31 = R63 (3,1);
g32 = R63 (3,2);
g33 = R63 (3,3);

% THETA 4,5,6 CALCULATION

theta5 = atan2 ( sqrt((g31)^2 +(g32)^2), g33);
if(theta5 == 0)
THETA4=  0
THETA5=  0
theta6 = atan2 (-g12, g11);
THETA6=  theta6*180/pi
elseif (theta5 == pi)
THETA4=  0
THETA5=  0
theta6 = atan2 (g12,-g11);
THETA6=  theta6*180/pi
else
theta4 = atan2 (g32/ sin (theta5), - g31/ sin (theta5));
theta6 = atan2 (g23/ sin (theta5), g31/ sin (theta5));
THETA4=  conversion( theta4,200,-200);
THETA5=  conversion( theta5,115,-115);
THETA6=  conversion( theta6,400,-400);

% FLIPPED POSTION

theta44 = theta4 + pi;
theta55 = -theta5;
theta66 = theta6+pi;
THETA44=  conversion( theta44,200,-200);
THETA55=  conversion( theta55,115,-115);
THETA66=  conversion( theta66,400,-400);
disp ( ' THETA 4,5,6 SOLUTIONS')
Solution1 = [THETA4,THETA5,THETA6]
Solution2 = [THETA44,THETA55,THETA66]
end

% FIRST POSSIBLE SOLUTION OF THETA2 FUNCTION

function RES = THE2(xtip,ytip,ztip,theta1,zeta)
 s = (ztip - d1);
 r =  sqrt((xtip - a1*cos (theta1))^2 +(ytip - a1*sin(theta1))^2);
omega = atan2 (s, r);
lenda = atan2 (( d4+d6) * sin (zeta) , a2+( d4+d6)* cos (zeta));
RES =  - ((omega - lenda)- ( pi/2)) ;
End
```

```
% SECOND POSSIBLE SOLUTION OF THETA2 FUNCTION

function RES1 = THE2COMP(xtip,ytip,ztip,theta1,zeta)
s = (ztip - d1);
r =  - sqrt((xtip - a1*cos (theta1))^2 +(ytip - a1*sin(theta1))^2);
omega = atan2 (s, r);
lenda = atan2 (( d4+d6) * sin (zeta) , a2+( d4+d6)* cos (zeta));
RES1 = - ((omega - lenda) - ( pi/2));
end

% JOINT ANGLES LIMIT FUNCTION

function OUT = conversion( theta,upperlimit,lowerlimit)
upperlimit = upperlimit * pi / 180;
lowerlimit = lowerlimit * pi / 180;
if (theta > upperlimit)
OUT = (' THE SOLUTION OUT OF JOINT ANGLE LIMIT ');
elseif (theta < lowerlimit)
OUT = (' THE SOLUTION OUT OF JOINT ANGLE LIMIT ');
else
OUT = theta * 180 / pi;
end

end

end
```