SETSCI Conference Proceedings,
4 (1): 383-390, 2019

ISAS 2019
3rd International Symposium on Innovative Approaches in Scientific Studies
Section: Engineering and Natural Sciences
April 19 – 21, 2019, Ankara, Turkey

Inverse Kinematic Analysis of IRB120 Robot Arm

Oğuz Faik Seven[1*] and Arif Ankaralı [2]

[1]Mechanical Engineering Department, Gazi University, Ankara, Turkey
[2] Mechanical Engineering Department, Ankara Yıldırım Beyazıt University, Ankara, Turkey
[*]Corresponding author and Speaker: o.faikseven@gazi.edu.tr
Presentation/Paper Type: Oral / Full Paper

Abstract **–** Robot algorithms require faster and more efficient computing due to increasing integration of robot arms to automation systems and expanding capabilities of robotic systems. Inverse kinematics calculations used in trajectory generation is one of these algorithms. The inverse kinematics problem can be defined as determination of joint angular positions for a desired position and orientation of the end-effector. The aim of this paper is to present a closed form solution to the inverse kinematics problem of IRB120 robot arm. To this end, forward kinematics is formulated using Denavit-Hartenberg representation and analytic solution for inverse kinematics is obtained with a geometric approach. The inverse kinematics solution is tested for different robot arm configurations in a simulation environment written in GNU Octave. The analytical solution successfully produced all possible robot configurations. Also, using a sampling rate of $0.001s$, cartesian space trajectories are successfully mapped to the joint space. It is concluded that presented analytical solution can be used in real time control applications of IRB120 robot arm.

Keywords – Inverse Kinematics, IRB120, D-H Representation, Industrial Manipulators, Robot Trajectory

I. Introduction

Inverse kinematic solution of a manipulator holds primary importance in the control applications. Trajectory planning in control applications is mostly achieved by mapping the task space trajectories to the joint space using inverse kinematics. For a given set of position and orientation, the inverse kinematic solution is not unique and depending on the robot structure, various robot postures might give the desired result. A general 6R manipulator will have at most 16 such postures [1]. However, due to mechanical restrictions IRB120 has four different configurations.

Although, there are numerous analytical, numerical and soft-computing solution methods available in the literature, every method has its own advantages. If the robot structure has a closed form solution, analytical methods will yield the fastest result. This is the reason that most of the manipulators are designed to give a closed form solution. For example, manipulators having a Euler wrist structure (three joint axis forming the wrist intersect at a common point) will always have a closed form solution [2]. One of the analytical solution methods is to use the robot geometry. Trigonometric functions resulting from the geometric projection of the robot structure can be used to obtain the desired angular positions [3], [4]. Another analytical solution method is the use of algebraic manipulation on the equations resulting from the kinematic model. One such method which utilizes vector analysis theory and dual-number algebra can be found in [5].

The analytic methods become insufficient in the analysis of manipulators which does not yield a closed form solution. In such cases, numerical methods can be utilized. Unlike analytical methods, numerical counterparts use iterative procedures to converge the results. Most numerical methods applied in the inverse kinematic problem differ in solution methods. It is possible to apply known numerical solution methods to set of nonlinear equations such as Gauss-Newton [6], cyclic coordinate descent [7], inverse jacobian [8], interval analysis [9], and polynomial continuation [10]. However, the numerical methods lack speed and in complex mechanisms, the solution might not converge for all possible configurations. Also, they are known to perform poorly around the singular points of the manipulator.

In recent years, the focus of the study shifted towards soft computing methods. Soft computing can be roughly defined as deduction of input output relations in a complex system via approximate calculations. The methods include artificial neural networks(ANN) [11], adaptive neuro-fuzzy inference system(ANFIS) [12], genetic algorithm [13], and particle-swarm optimization algorithm [14]. It is also possible to use evolutionary symbolic regression algorithm to produce a closed form solution [15]. One important comparative study on the performance of these soft computing methods when applied to the inverse kinematics problem can be found in [16]. Although, ANN and ANFIS gives sufficiently fast results, they suffer from accuracy. Also, optimization algorithms are similar to the numerical methods which comes with a heavy computation cost.

In this study, closed form inverse kinematic solution for IRB120 robot arm is obtained using a geometric approach which was first proposed by Lee and Ziegler [3]. The solution proved successful in obtaining all possible robot configurations. Also, a sample task space trajectory is mapped to the joint space using the inverse kinematics equations presented. It is important to highlight that the obtained solution falls under the analytical solution methods category. Compared with algebraic solution methods, it presents a simpler formulation, because the method does not deal with the manipulation of heavy algebraic equations. Also, the method yields faster results than its numerical and soft-computing counterparts due to its analytic nature.

II. Materials and Method

Kinematic analysis of robot manipulators consists of two parts, namely forward and inverse kinematics. Forward kinematics is the study of link positions and orientations for a given set of joint angular positions$(\theta_1, \theta_2, \ldots, \theta_n)$. This concept forms the basis of geometric representation of the system. Thus, it will be the starting point for inverse kinematics solution of the manipulator.

A. Kinematic Representation

Although, there are numerous approaches regarding the kinematic representation of robot manipulators, we shall use the formulation presented by Denavit and Hartenberg [17]. Figure 1 gives the established Denavit-Hartenberg(D-H) link coordinate systems illustrated both on the manipulator and in the simulation environment.

After establishing the coordinate system, the structure of the robot can be represented with four D-H parameters associated with each link: the joint angle(θ_i), link length(d_i), offset distance(a_i), and twist angle(α_i). Table 1 gives the D-H parameters associated with each link.

Table 1. IRB 120 Robot Arm D-H Kinematic Parameters

Joint i	**θ_i(deg)**	**α_i(deg)**	**a_i(mm)**	**d_i(mm)**
1	90	-90	0	124
2	-90	0	270	0
3	0	-90	70	0
4	0	-90	0	302
5	0	90	0	0
6	0	0	0	72

B. Forward Kinematics

Let a_i, α_i, d_i, and θ_i be the D-H parameters of link i, then a joint to joint homogeneous transformation matrix ${}^{i-1}\boldsymbol{A}_i$ which maps the coordinates from link $i-1$ to link i can be defined as [18]:

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

Then, the kinematic equation of a manipulator which fully describes the position and orientation of every link with respect to a base coordinate system can be expressed by the successive joint to joint mapping of adjacent links.

$$ {}^0\boldsymbol{T}_i = \prod_{j=1}^{i} {}^{j-1}\boldsymbol{A}_j \, , for\ i = 1, 2, \ldots, n \quad (2) $$

For easy calculation, it is possible to represent the same formula by dividing into two calculation steps. Let $\boldsymbol{T}_1 = {}^0\boldsymbol{A}_1 {}^1\boldsymbol{A}_2 {}^2\boldsymbol{A}_3$ and $\boldsymbol{T}_2 = {}^3\boldsymbol{A}_4 {}^4\boldsymbol{A}_5 {}^5\boldsymbol{A}_6$. After successive mappings of the respective joints,$\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ becomes,

$$ \boldsymbol{T}_1 = \begin{bmatrix} C_1C_{23} & S_1 & -C_1S_{23} & C_1(a_2C_2 + a_3C_{23}) \\ S_1C_{23} & -C_1 & -S_1S_{23} & S_1(a_2C_2 + a_3C_{23}) \\ -S_{23} & 0 & -C_{23} & d_1 - a_2S_2 - a_3S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

$$ \boldsymbol{T}_2 = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -C_4C_5S_6 - S_4C_6 & C_4S_5 & d_6C_4S_5 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_4S_5 & d_6S_4S_5 \\ -S_5C_6 & S_5S_6 & C_5 & d_4 + d_6C_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

where $C_i = cos\theta_i$; $S_i = \sin\theta_i$; $C_{ij} = \cos(\theta_i + \theta_j)$; $S_{ij} = \sin(\theta_i + \theta_j)$. Then, the orientation and the position of the end-effector can be expressed as,

$$ {}^0\boldsymbol{T}_6 = \boldsymbol{T}_1\boldsymbol{T}_2 \quad (5) $$

If the operation is carried out further, the mapping between the end-effector and the base coordinate systems can be represented without any matrix multiplication. Let normal($\boldsymbol{n}$),sliding($\boldsymbol{s}$), approach($\mathbf{a}$), and position($\boldsymbol{p}$) vectors represent the orientation and position of the end-effector coordinate system. Then, the homogeneous transformation matrix which maps between the end-effector and the base coordinate systems is,

$$ {}^0\boldsymbol{T}_6 = \boldsymbol{T}_1\boldsymbol{T}_2 = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6) $$

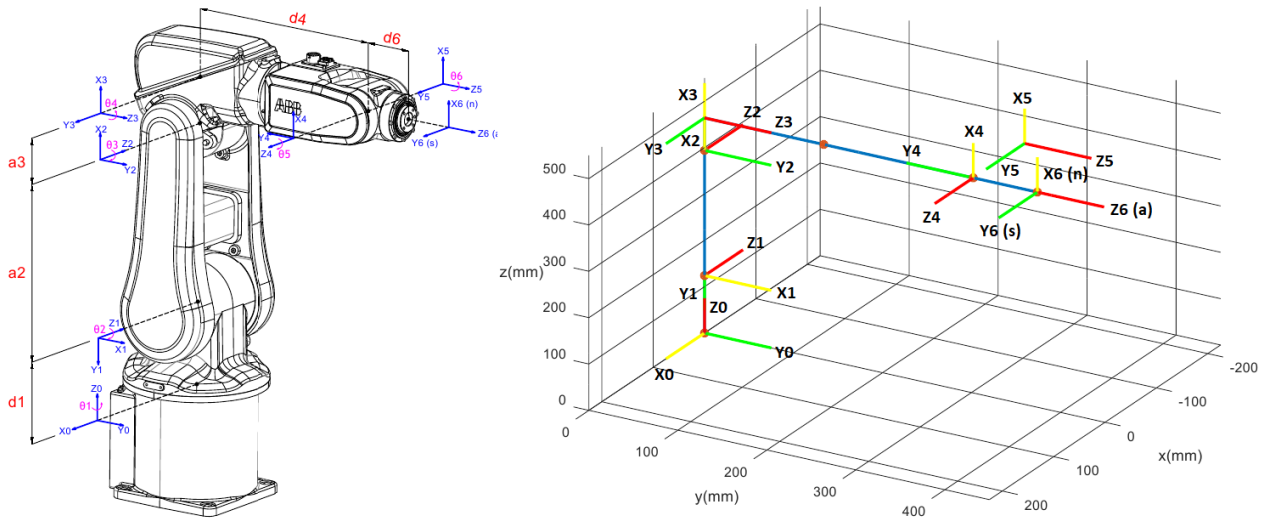

Fig. 1 D-H Link Coordinate System

C. Inverse Kinematics

In robotics, the inverse kinematics problem refers to the calculation of joint angles for a given position and orientation of the end-effector. Unlike forward kinematics, inverse kinematics problem is not straight forward and there is not a general solution method which is applicable to every manipulator type. Moreover, depending on the geometric configuration of the manipulator, there might exist multiple configurations/solutions for a desired position end orientation.

Lee and Ziegler have proposed a geometric approach in solving inverse kinematics problem for PUMA type robots [2]. Their approach utilizes *atan2* function by generation two analytic solutions for every joint angle, one for sine of the angle and the other one for the cosine. The inverse kinematics solution presented in this study is an extension of their approach for PUMA type robots.

As the manipulator is designed to imitate human arm, the robot configurations are also defined according to the human arm structure. Four different configurations of IRB120 are defined as,

Above Arm: Position of the end effector has a positive coordinate in the direction of $\boldsymbol{y}_2$ axis.

Below Arm: Position of the end effector has a negative coordinate in the direction of $\boldsymbol{y}_2$ axis.

Wrist Down: $\boldsymbol{s} \cdot \boldsymbol{y}_5 > 0$

Wrist Up: $\boldsymbol{s} \cdot \boldsymbol{y}_5 < 0$

These four configurations can be implemented into the mathematical formulation using two operators. The values of these two operators must be assigned for inverse kinematic solution depending on the desired configuration of the robot as the following,

$$ELBOW = \begin{cases} -1, Above\ arm \\ +1, Below\ arm \end{cases} \tag{7}$$

$$WRIST = \begin{cases} +1, Wrist\ down \\ -1, Wist\ up \end{cases} \tag{8}$$

As the $WRIST$ operator defined by comparing two vectors, its value does not have a clear visual indication. Figure 2 illustrates different arm configurations for $ELBOW$ operator.

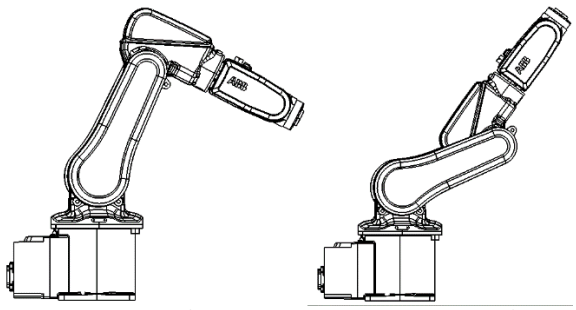

Fig. 2 Above arm(on the Left) and Below Arm(on the Right) Configurations of IRB120

First, the isolation of the first three joints is needed by eliminating the wrist motion out of the equation. Since the last two links of the robot always moves parallel to each other, it is possible to calculate the coordinates of the fourth coordinate system without considering the wrist solution. Let $\boldsymbol{p_6}$ define the position vector of the sixth coordinate system and $\boldsymbol{a}$ is the desired approach vector for the end-effector. Then, the coordinates of the fourth coordinate system, which is also the position vector of the homogeneous transformation matrix ${}^0\boldsymbol{T}_4$, can be expressed as,

$$\boldsymbol{p_4} = \boldsymbol{p_6} - d_6\boldsymbol{a} = [p_x \quad p_y \quad p_z]^T \tag{9}$$

Joint 1 Solution:

Considering only the first three links of the robot, if the kinematic representation of the robot given in Figure 1 is projected onto $\boldsymbol{x}_0\boldsymbol{y}_0$ plane, the solution for the first joint can be obtained.

$$\theta_1 = atan2(p_y, p_x) \tag{10}$$

The use of $atan2$ function places the solution to the correct quadrant, and eliminates the possible complications which may arise from the tangent function.

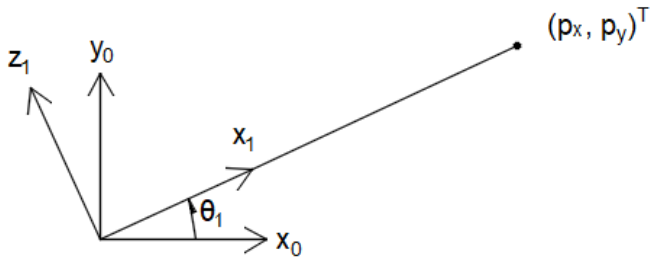

Fig. 3 Joint 1 Solution

Joint 2 Solution:

By following the same procedure, angular position of the second joint can be solved by projecting the robot arm onto the $\boldsymbol{x}_1\boldsymbol{y}_1$ plane. Unlike the first joint, there are two possible arm configurations. From Figure 4, following geometric relations can be obtained,

$$r = \sqrt{p_x^2 + p_y^2} \tag{11}$$

$$R = \sqrt{r^2 + (p_z - d_1)^2} \tag{12}$$

$$sin\alpha = \frac{p_z - d_1}{R} \tag{13}$$

$$cos\alpha = \frac{r}{R} \tag{14}$$

$$cos\beta = \frac{a_2^2 + R^2 - a_3^2 - d_4^2}{2a_2R} \tag{15}$$

$$sin\beta = \sqrt{1 - cos^2\beta} \tag{16}$$

$$\alpha = atan2(sin\alpha, cos\alpha) \tag{17}$$

$$\beta = atan2(sin\beta, cos\beta) \tag{18}$$

In Table 2, the values of $ELBOW$ operator in the second and third joint solutions for different robot configurations are given.

Table 2. Arm Configurations for Joint 2 and 3

Configuration	$\boldsymbol{\theta_2}$	$\boldsymbol{\theta_3}$	$\boldsymbol{ELBOW}$
Below Arm	$-\alpha + \beta$	$\pi - \phi - \psi$	+1
Above Arm	$-\alpha - \beta$	$-\pi - \phi + \psi$	-1

It is possible to express the formulas given in Table 2 in a more compact format.

$$\theta_2 = -\alpha + ELBOW \cdot \beta \tag{19}$$

Then, sine, cosine and the actual value of θ_2 can be calculated:

$$sin\theta_2 = -sin\alpha\, cos\beta + ELBOW \cdot cos\alpha\, sin\beta \tag{20}$$

$$cos\theta_2 = cos\alpha\, cos\beta + ELBOW \cdot sin\alpha\, sin\beta \tag{21}$$

$$\theta_2 = atan2(sin\theta_2, cos\theta_2) \tag{22}$$

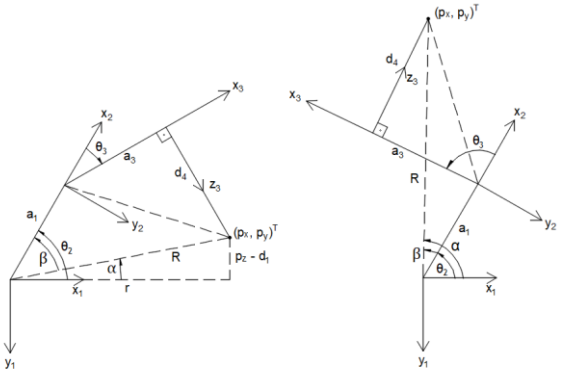

Fig. 4 Joint 2 Solution for Above Arm(on the Left) and Below Arm(on the Right)

Joint 3 Solution:

The third joint can be solved by projecting the arm of the robot onto $\boldsymbol{x_2 y_2}$ plane. Similar to the second joint, there are two possible configurations. From Figure 5, necessary geometric relations can be obtained as:

$$cos\psi = \frac{a_2^2 + a_3^2 + d_4^2 - R^2}{2a_2\sqrt{a_2^2 + d_4^2}} \tag{23}$$

$$sin\psi = \sqrt{1 - cos^2\beta} \tag{24}$$

$$cos\phi = \frac{a_3}{\sqrt{a_3^2 + d_4^2}} \tag{25}$$

$$sin\phi = \frac{d_4}{\sqrt{a_3^2 + d_4^2}} \tag{26}$$

$$\psi = atan2(sin\psi, cos\psi) \tag{27}$$

$$\phi = atan2(sin\phi, cos\phi) \tag{28}$$

Using the operator values given in Table 2, θ_3 can be expressed in terms of the desired configuration of the robot:

$$\theta_3 = -\phi - ELBOW \cdot (\pi - \psi) \tag{29}$$

$$sin\theta_3 = sin\phi cos\psi - ELBOW \cdot cos\phi\, sin\psi \tag{30}$$

$$cos\theta_3 = -cos\phi cos\psi - ELBOW \cdot sin\phi sin\psi \tag{31}$$

$$\theta_3 = atan2(sin\theta_3, cos\theta_3) \tag{32}$$

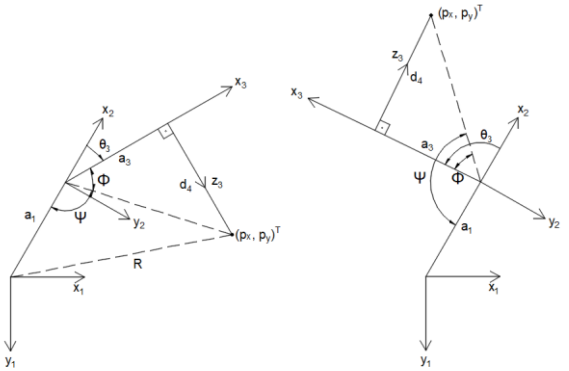

Fig. 5 Joint 3 Solution for Above arm(on the Left) and Below Arm(on the Right)

Thus far, we presented the solution of the first three joints. With this knowledge, it is possible to calculate the orientation of the third coordinate system. From there we can assure the necessary orientation of the end-effector is obtained. As it was suggested in [3], for the desired end-effector orientation following criteria should be met:

$$\boldsymbol{z_4} = \frac{\pm(\boldsymbol{z_3} \times \boldsymbol{a})}{\|\boldsymbol{z_3} \times \boldsymbol{a}\|}, for\ joint\ 4\ solution \tag{33}$$

$$\boldsymbol{a} = \boldsymbol{z_5}, for\ joint\ 5\ solution \tag{34}$$

$$\boldsymbol{s} = \boldsymbol{y_6}, for\ joint\ 6\ solution \tag{35}$$

Joint 4 Solution:

Depending on the $WRIST$ operator's value, the desired orientation can be obtained via setting $\boldsymbol{z_4}$ vector to either $\boldsymbol{z_3} \times \boldsymbol{a}$ or $-(\boldsymbol{z_3} \times \boldsymbol{a})$. Unfortunately, it is not possible to determine the sign of $\boldsymbol{z_4}$ beforehand. Since, the $WRIST$ operator is defined by comparing the end-effector and fifth coordinate systems, the true sign of the cross product cannot be determined without referring to this comparison. Also, when the approach vector aligns with $\boldsymbol{z_3}$ vector, the cross product of the two vector becomes zero and the degenerate case occurs. These, complications can be overcome by assuming the sign of the cross product is positive and defining another sign operator.

$$\Omega = \begin{cases} 0 & degenerate\ case \\ \boldsymbol{s} \cdot \boldsymbol{y_5} & if\ \boldsymbol{s} \cdot \boldsymbol{y_5} \neq 0 \\ \boldsymbol{n} \cdot \boldsymbol{y_5} & if\ \boldsymbol{s} \cdot \boldsymbol{y_5} = 0 \end{cases} \tag{36}$$

From Figure 1, it can be seen that $\boldsymbol{y_5}$ and $\boldsymbol{z_4}$ are always parallel to each other. Also, using the $\boldsymbol{z_4}$ definition given in (36), the sign operator can be redefined as,

$$\Omega = \begin{cases} 0 & degenerate\ case \\ \boldsymbol{s} \cdot \dfrac{(\boldsymbol{z_3} \times \boldsymbol{a})}{\|\boldsymbol{z_3} \times \boldsymbol{a}\|} & if\ \boldsymbol{s} \cdot (\boldsymbol{z_3} \times \boldsymbol{a}) \neq 0 \\ \boldsymbol{n} \cdot \dfrac{(\boldsymbol{z_3} \times \boldsymbol{a})}{\|\boldsymbol{z_3} \times \boldsymbol{a}\|} & if\ \boldsymbol{s} \cdot (\boldsymbol{z_3} \times \boldsymbol{a}) = 0 \end{cases} \tag{37}$$

If the initial sign assumption of the vector cross product is correct, Ω and $WRIST$ operator will have the same sign. As a result, by combining these two operator and defining a new corrected sign operator M, true orientation of the wrist can be obtained. Table 3 gives the corrected sign for all possible configurations.

Table 3. Wrist Configurations

Wrist Orientation	$\boldsymbol{\Omega}$	$\boldsymbol{WRIST}$	$\boldsymbol{M = WRIST \cdot sign(\Omega)}$
Down	≥ 0	+1	+1
Down	< 0	+1	-1
Up	≥ 0	-1	-1
Up	< 0	-1	+1

Using the sign operator M and projecting the last three links of the robot onto $\boldsymbol{x_4 z_4}$ plane, fourth joint can be solved.

$$sin\theta_4 = -M(z_4 \cdot x_3) \tag{38}$$

$$cos\theta_4 = M(z_4 \cdot y_3) \tag{39}$$

$$\theta_4 = atan2(sin\theta_4, cos\theta_4) \tag{40}$$

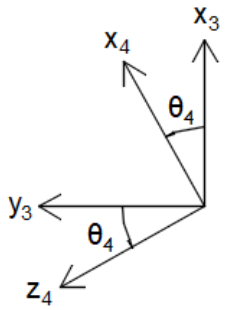

Fig. 6 Joint 4 Solution

Joint 5 Solution:

Using the projection of the wrist onto $\boldsymbol{x_4 y_4}$ frame given in Figure 7, it is possible to obtain required geometric relations for the fifth joint solution.

$$sin\theta_5 = \boldsymbol{a} \cdot \boldsymbol{x}_4 \tag{41}$$

$$cos\theta_5 = -\boldsymbol{a} \cdot \boldsymbol{y}_4 \tag{42}$$

$$\theta_5 = atan2(sin\theta_5, cos\theta_5) \tag{43}$$

Note that $\boldsymbol{x}_4$ and $\boldsymbol{y}_4$ vector correspond to the x and y column of the homogeneous transformation matrix ${}^0\boldsymbol{T}_4$. With the knowledge of the first four joint angles and using the procedure described in the forward kinematics section, it is possible to calculate this matrix.

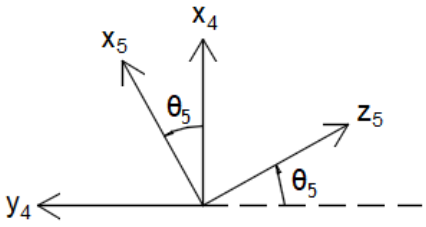

Fig. 7 Joint 5 Solution

Joint 6 Solution:

With the solution of the first five joints, the manipulator end-effector is in the desired coordinates, and with the sixth joint solution, the orientation will also be aligned with the desired orientation. The solution for the last joint can also be obtained by following same procedure. Figure 8 gives the projection of the wrist onto the $\boldsymbol{x_5 y_5}$ plane, and the solution for this joint becomes,

$$sin\theta_6 = \boldsymbol{n} \cdot \boldsymbol{y}_5 \tag{44}$$

$$cos\theta_6 = \boldsymbol{s} \cdot \boldsymbol{y}_5 \tag{45}$$

$$\theta_6 = atan2(sin\theta_6, cos\theta_6) \tag{46}$$

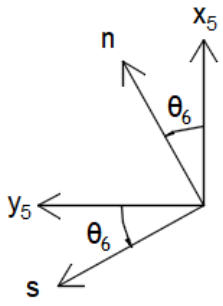

Fig. 8 Joint 6 Solution

Note that when $\boldsymbol{z}_3 \times \boldsymbol{a} = 0$, degenerate case occurs. When a desired position and orientation coincides with the degenerate case, individual values of θ_4 and θ_6 loses importance and only their sum $(\theta_4 + \theta_6)$ matters. While controlling the robot, generally, θ_4 is set to its current value and the required rotation is obtained with the rotation of θ_6 only.

D. Trajectory Generation with Inverse Kinematics

Suppose the manipulator is required to track the space curve which is defined in its parametric form.

$$\vec{\boldsymbol{r}}(t) = a\,cos(t)\,\vec{\boldsymbol{i}} + a\,sin(t)\,\vec{\boldsymbol{j}} + bt\vec{\boldsymbol{k}} \tag{47}$$

where $a = 400\ mm$, $b = 120\ mm$, and $0 \leq t \leq \pi$. Then, the curve can be expressed in terms of arc length(s).

$$\vec{\boldsymbol{r}}(s) = a\,cos(cs)\,\vec{\boldsymbol{i}} + a\,sin(cs)\,\vec{\boldsymbol{j}} + bcs\vec{\boldsymbol{k}} \tag{48}$$

where $c = 1/\sqrt{a^2 + b^2}$. The task space trajectory defined by (47) can be seen in Figure 9.

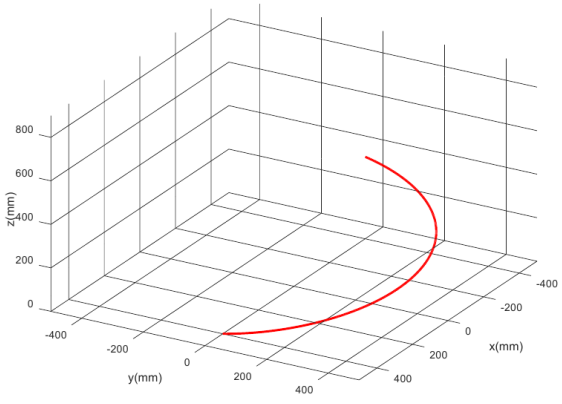

Fig. 9 The Task Space Trajectory

In obtaining inverse kinematics solution, the orientation of the end-effector is also needed. Although, it is possible to define a constant orientation here, it is defined using the fundamental properties of the curve to make the simulation more realistic. Unit tangent vector of the trajectory can be expressed as,

$$\vec{T}(s) = \frac{d\vec{r}/ds}{\|d\vec{r}/ds\|} = -ac\,sin(cs)\,\vec{\imath} + ac\,cos(cs)\,\vec{\jmath} + bc\vec{k} \tag{49}$$

Then, the unit normal vector of the trajectory becomes,

$$\vec{N}(s) = \frac{d\vec{T}/ds}{\|d\vec{T}/ds\|} = -cos(cs)\,\vec{\imath} - sin(cs)\,\vec{\jmath} \tag{50}$$

Using unit tangent and unit normal vectors, binormal of the trajectory can be defined to complete the right handed coordinate system.

$$\vec{B} = \vec{T} \times \vec{N} = bc\,sin(cs)\,\vec{\imath} - bc\,cos(cs)\,\vec{\jmath} + ac\vec{k} \tag{51}$$

Now, the homogeneous transformation matrix can be expressed to give the desired trajectory.

$${}^{0}T_6 = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{-T} & \boldsymbol{-N} & \boldsymbol{r} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{52}$$

$${}^{0}T_6 = \begin{bmatrix} bc\,sin(cs) & ac\,sin(cs) & cos(cs) & a\,cos(cs) \\ -bc\,cos(cs) & -ac\,cos(cs) & sin(cs) & a\,sin(cs) \\ ac & -bc & 0 & bcs \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{53}$$

III. Results

For kinematics simulation, a computer program which covers both forward and inverse kinematics is written in GNU Octave. In the Figure 10, results of the forward kinematics simulation for individual joints can be seen. The robot pose is obtained via forward kinematics procedure explained in the previous chapter. Figures are obtained by moving each joint in the specified range while keeping other joints stationary. For each pose of the robot, angular positions are used to calculate the coordinates of each joint. Then, the program uses these coordinates to generate the posture of the manipulator. Yellow, green, and red coordinate axes represents x, y, and z coordinate axes of the respective joints.

The program can also simulate four different arm configurations using the procedure presented. The desired position and orientation of the end-effector is supplied to the program in the homogeneous transformation matrix format ${}^{0}T_6$. Then, the program calculates necessary angular positions of each joint for four different configurations. In order to illustrate these configurations, following arbitrary end-effector position and orientation is used.

$${}^{0}T_6 = \begin{bmatrix} 0.3394 & 0.9402 & 0.0312 & 2.17 \\ -0.4771 & 0.1444 & 0.8670 & 275.38 \\ 0.8107 & -0.3087 & 0.4975 & 595.79 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{54}$$

Angular positions which correspond to this position and orientation is calculated and expressed in Table 4, and also, calculated angular positions are simulated using forward kinematics. The results can be seen in Figure 11.

Table 4. Angular Positions for Different Configurations

Robot Config.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
Above Arm Wrist Down	90.0	-27.8	-143.9	-2.2	-51.9	21.2
Above Arm Wrist Up	90.0	-27.8	-143.9	177.8	51.9	-158.8
Below Arm Wrist Down	90.0	-100.0	-10.0	10.0	10.0	10.0
Below Arm Wrist Up	90.0	-100.0	-10.0	-170.0	-10.0	-170.0

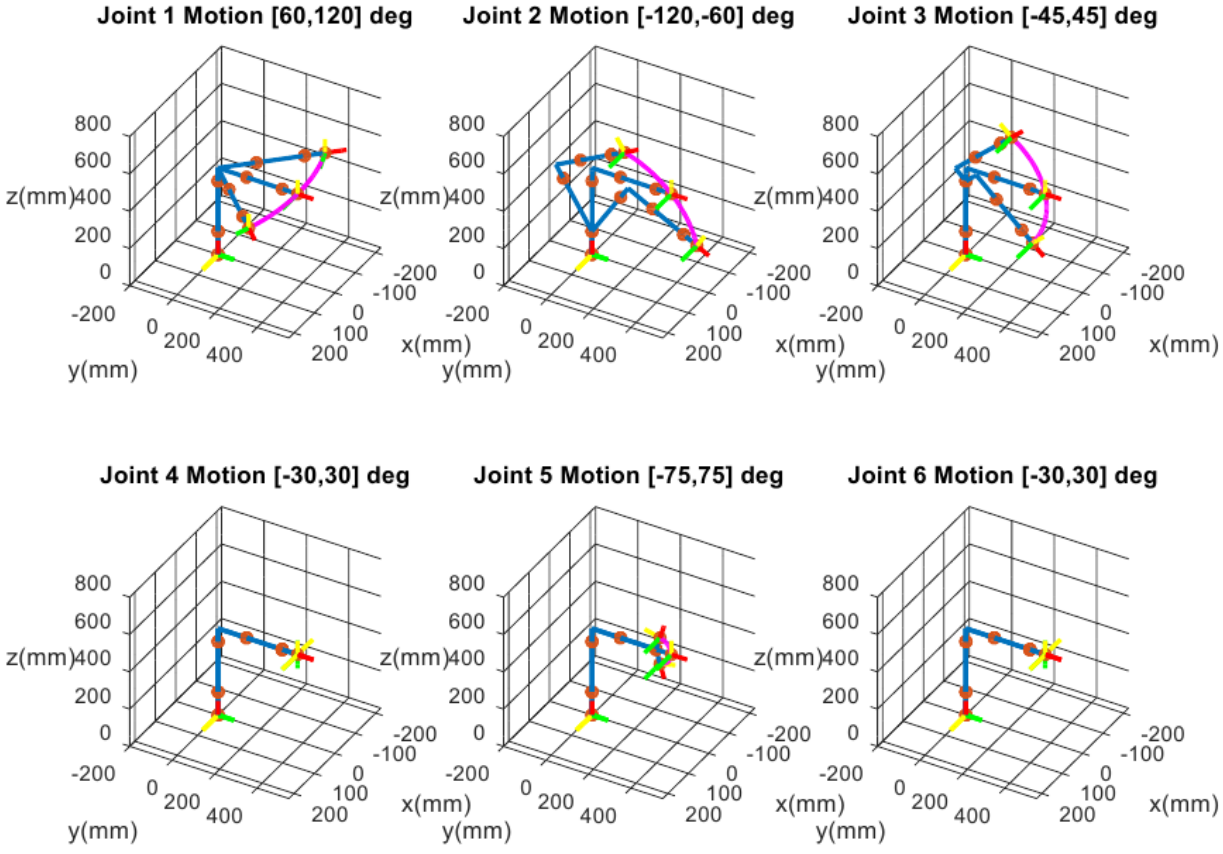

Fig. 10 Forward Kinematics Simulation

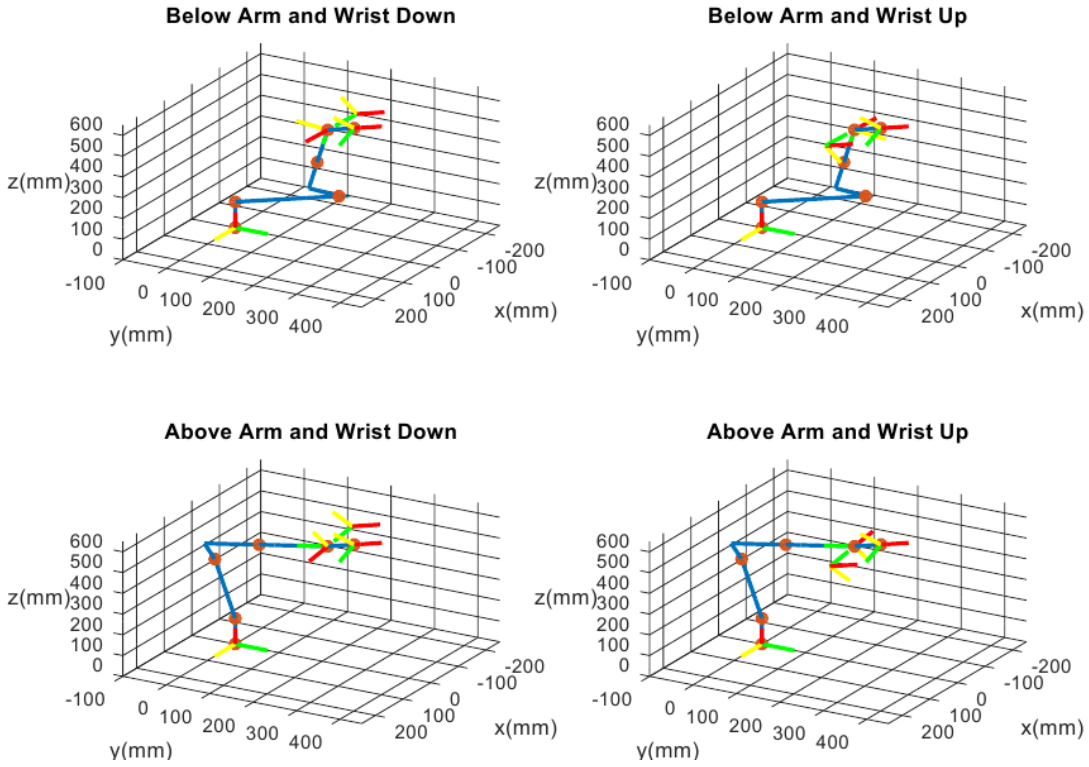

Fig. 11 Simulation of Different Manipulator Configurations

With the desired orientation which was defined for the trajectory generated, the approach vector will be always normal and the sliding vector will be always tangent to the trajectory. To illustrate the robot movement, using sample points along the trajectory, plot given in Figure 12 is generated for above arm and wrist down configurations.

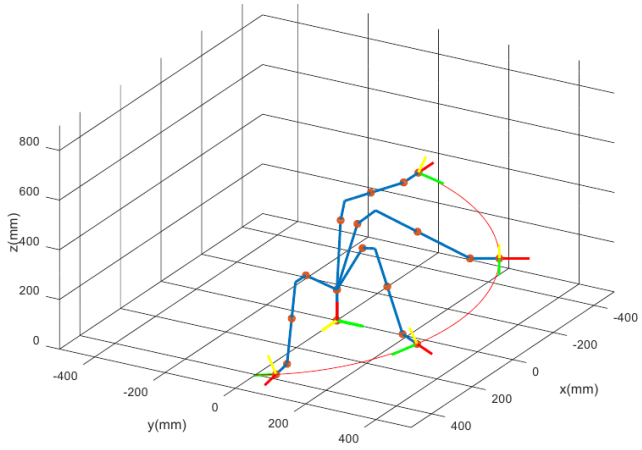

Fig. 12 Trajectory Tracking with Inverse Kinematics

As it can be seen in the figure, the end-effector passes the trajectory with the desired orientation. Using a sampling rate of $t = 0.001s$, joint space trajectories which would give the desired task space trajectory can be generated. Figure 13 shows the joint space trajectory of each link.

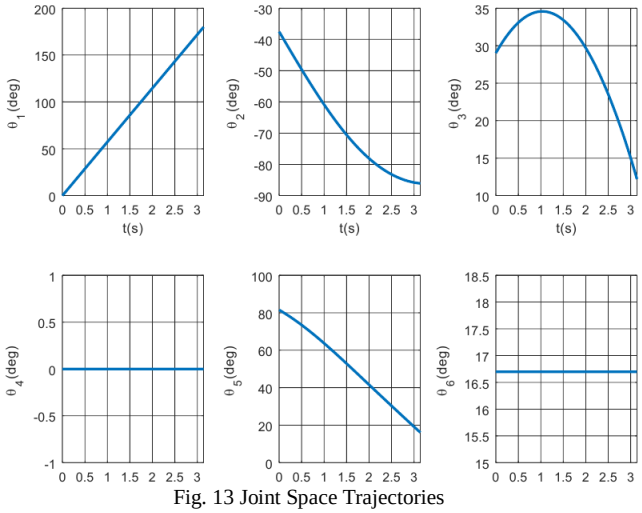

Fig. 13 Joint Space Trajectories

IV. Discussion

In Figure 10, individual joint motion is simulated. Motion generated from each joint moves the end-effector through a circular path. This indicates that the forward kinematics formulation is successfully implemented. Also, D-H kinematic representation presented itself in a simpler format than the ones encountered in the literature like screw theory.

It can be seen from Figure 11 the program could solve for all possible robot arm configurations. The desired configuration can easily be defined using $ELBOW$ and $WRIST$ operators. Note that for a general 6R manipulator, backswept configuration of the arm should be considered, but it is generally not used in pick and place applications. For this reason, the solution was not extended to include this configuration. However, the solution method can easily be extended by defining another operator which covers backswept positions as well.

Figure 12 and Figure 13 shows a sample trajectory generation using inverse kinematics. Since the trajectory

represents a circular helix, a linear increase in θ_1 is expected. Also, required rotation in the end-effector is obtained with the first joint and joints four and six stayed stationary. Using the angular positions given in Figure 12, it is possible generate joint space velocities and accelerations. Therefore, the solution method presented enables to describe a fully defined trajectory which can be used in trajectory tracking.

V. Conclusion

In this study, closed form solution for inverse kinematics problem of IRB120 is obtained via a geometric approach. The solution is tested using computer simulation and described method could calculate joint angles for all possible robot configurations. Also, a sample task space trajectory is used to generate joint space trajectories. Compared to the numerical and soft-computing solution methods, the closed form solution performs faster due to its analytical nature. As a result, real time control applications of IRB120 is possible with the solution presented.

Acknowledgment

The authors would like to thank M. Erbaş for language editing and proof reading of this document.

References

[1] D. Manocha, and J. F. Canny, "Efficient inverse kinematics for general 6R manipulators," *IEEE transactions on robotics and automation*, vol. 10, pp. 648-657, Oct. 1994.

[2] S. Küçük, and Z. Bingül. "The inverse kinematics solutions of industrial robot manipulators," in *Proc. ICM'04*, 2004, p. 274.

[3] C. S. G. Lee, and M. Ziegler, "Geometric approach in solving inverse kinematics of PUMA robots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 20, pp. 695-706, Dec. 1984.

[4] J. Cote, C. M. Gosselin, and D. Laurendeau, "Generalized inverse kinematic functions for the Puma manipulators," *IEEE transactions on robotics and automation*, vol. 11, pp. 404-408, June 1995.

[5] H. Y. Lee, and C. G. Liang, "A new vector theory for the analysis of spatial mechanisms," *Mechanism and Machine Theory*, vol. 23, pp. 209-217, May 1988.

[6] J. Angeles, "On the numerical solution of the inverse kinematic problem," *The International Journal of Robotics Research*, vol. 4, pp. 21-37, June 1985.

[7] A. L. Olsen, and H. G. Petersen, "Inverse kinematics by numerical and analytical cyclic coordinate descent," *Robotica*, vol. 29, pp. 619-626, Dec. 2011.

[8] Y. Zhao, T. Huang, and Z. Yang, "A new numerical algorithm for the inverse position analysis of all serial manipulators," *Robotica*, vol. 24, pp. 373-376, May 2006.

[9] R. S. Rao, A. Asaithambi, and S. K. Agrawal, "Inverse kinematic solution of robot manipulators using interval analysis," *Journal of Mechanical Design*, vol. 120, pp. 147-150, Mar. 1998.

[10] C. Wampler, and A. Morgan, "Solving the 6R inverse position problem using a generic-case solution methodology," *Mechanism and Machine Theory*, vol. 26, pp. 91-106, Jan. 1991.

[11] R. Köker, T. Çakar, and Y. Sari, "A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators," *Engineering with Computers*, vol. 30, pp. 641-649, Jan. 2014.

[12] A. Ankaralı, "ANFIS inverse kinematics and precise trajectory tracking of a dual arm robot," *Proc. MSV'12*, 2012, p. 270.

[13] P. Kalra, P. B. Mahapatra, and D. K. Aggarwal, "An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots," *Mechanism and machine theory*, vol. 41, pp. 1213-1229, Oct. 2006.

[14] N. Rokbani, and A. M. Alimi, "Inverse kinematics using particle swarm optimization, a statistical analysis," *Procedia Engineering*, vol. 64, pp. 1602-1611, Sept. 2013.

[15] F. Chapelle, and P. Bidaud, "Closed form solutions for inverse kinematics approximation of general 6R manipulators," *Mechanism and machine theory*, vol. 39, pp. 323-338, Mar. 2004.

[16] A. El-Sherbiny, M. A. Elhosseini, and A. Y. Haikal, "A comparative study of soft computing methods to solve inverse kinematics problem," *Ain Shams Engineering Journal*, vol. 9, pp. 2535-2548, Dec. 2018.

[17] R. S. Hartenberg, and J. Denavit, "A kinematic notation for lower pair mechanisms based on matrices," *Journal of applied mechanics*, vol. 77, pp. 215-221, June 1955.

[18] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision and Intelligence*, 1st ed., H. Freeman, Ed. New York, USA: McGraw-Hill, 1987.