

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/301451210>

# Robot manipulation through inverse kinematics

Conference Paper · July 2015

DOI: 10.1145/2783449.2783497

---

CITATIONS

14

---

READS

4,061

3 authors, including:



[Abdullah Aamir Hayat](#)

Singapore University of Technology and Design

70 PUBLICATIONS 743 CITATIONS

[SEE PROFILE](#)



[Subir Kumar Saha](#)

Indian Institute of Technology Delhi

114 PUBLICATIONS 771 CITATIONS

[SEE PROFILE](#)

# Robot Manipulation through Inverse Kinematics

Abdullah Aamir Hayat  
Department of Mechanical  
Engineering

Indian Institute of Technology Delhi  
New Delhi, India  
[aamir\\_hayat@rediffmail.com](mailto:aamir_hayat@rediffmail.com)

Ratan Sadanand O. M  
Department of Mechanical  
Engineering

Indian Institute of Technology Delhi  
New Delhi, India  
[ratan.sadan@gmail.com](mailto:ratan.sadan@gmail.com)

Subir. K. Saha  
Department of Mechanical  
Engineering

Indian Institute of Technology Delhi  
New Delhi, India  
[saha@mech.iitd.ac.in](mailto:saha@mech.iitd.ac.in)

## ABSTRACT

Inverse kinematics of a robot is very essential to find the joint variables that satisfy the desired pose of the robot during its manipulation. This is used in controlling the robot position, animation of the robot, etc. In this paper, step-by-step explanation and comparison of two widely used methods, namely, inverse kinematics and Jacobian inverse methods, for robot manipulation are presented. For this purpose a six degrees-of-freedom wrist-partitioned industrial robot KUKA KR5 Arc was used to illustrate the methods. A novel approach has been proposed for selecting the appropriate set of joint angles among the several inverse kinematic solutions. It is based on weight of each link and manipulability. The comparison of these approaches for linear and circular trajectory is presented. Their advantages, limitations, applications, and computations involved are also highlighted.

## Categories and Subject Descriptors

I.2.9 Robotics

## General Terms

Theory, Verification.

## Keywords

Inverse kinematics, Jacobian, Manipulability, Robot manipulation

## 1. INTRODUCTION

The goal of a robot controller is to generate an acceptable motion of the end-effector by precisely actuating its joints for a specified task. To study the geometry of motion without considering its cause comes under the subject of kinematics. Robot kinematics is divided into forward and inverse kinematics which are depicted in Figure 1. Forward kinematics (FK) utilizes kinematic equations to find the pose, i.e., position and orientation of the end-effector (EE), given the joint angles, while the inverse kinematics (IK) computes the joint angles for a desired pose of the

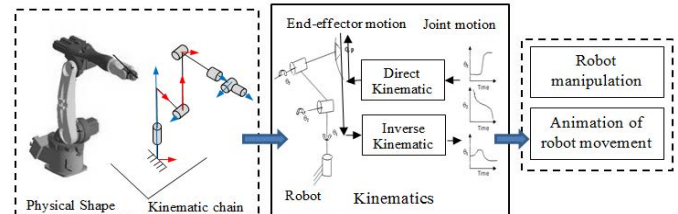


Figure 1. Robot kinematics and its utility

EE. Joint angles are then input to the actuators attached to move the links of the robot according to the specified trajectory. So there is a requirement of accurate joint angle values for precise

Table 1. Number of inverse kinematic solutions

DOF	Number of solutions
6 (6R, 5RP)	8
6R (intersecting wrist)	16
Greater than 6 (Redundant manipulator)	$\infty$

robot manipulation and/or kinematic animation.

For serial robots, FK is straightforward, which has unique solution, while the inverse kinematics has multiple solutions satisfying a particular pose. One approach to the inverse kinematics problem is to find a closed-form solution using algebraic or geometric method. Another approach is to find a numerical solution by a successive approximation algorithm. Although the former approach is generally more desirable in applying the solution to real-time control of robots, it is not always possible to obtain the closed-form solutions for the manipulators with arbitrary architectures [1]. There exist several methods of modeling and solving IK of a robot. The kinematic modeling of serial-chain robots are commonly done using the Denavit-Hartenberg (DH) parameters [2]. The analytical method for solving inverse kinematics for six degrees of freedom (DOF) robot manipulator with three consecutive axes either intersecting or parallel was reported in [3]. Such architecture have eight inverse kinematic solutions. A general six DOF robots, have sixteen solutions [4]. The number of analytical solutions for different degrees of freedom robots are listed in Table 1. Hence, for continuous motion of a robot, an accurate set of solutions among several inverse kinematic solutions is required. Another method for solving IK problem is by the use of Jacobian inverse [5, 6]. The numerical method for solving the inverse of Jacobian matrix was presented in [7, 8]. The analytical solution to the inverse kinematics problem for six-DOF robot manipulator is presented in [9]. The advantage of analytical expression is that it gives formulas for relationship between joint angle and link parameters. These relationship can be directly embedded in the robotics controller.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

AIR '15, July 02 - 04, 2015, Goa, India

© 2015 ACM. ISBN 978-1-4503-3356-6/15/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2783449.2783497>

Serial robots with wrist-partitioned feature, i.e., the last three link axes intersecting, are widely used in industries [2]. An algebraic method for solving the IK of a six-DOF robot with wrist-partitioned structure with all revolute joints, abbreviated here as *6R-WP-robot*, presented in [1] is used in this paper. Further, step-by-step procedure for implementing the IK for continuous robot manipulation is explained.

The strategy for resolving the conflict in selection of a set of joint angles from the eight inverse kinematics solutions of an industrial robot KUKA KR5 Arc is proposed in this paper. These strategies can also be followed for robots with multiple inverse kinematic solutions for manipulation and control. The analytical and inverse Jacobian methods are compared by tracing the given linear and circular trajectories. Such comparisons are not easily traceable in open literature, even though the industrial robot manufacturers may use them. Hence, this paper will help those researchers who would like to take forward the proposed concepts and implement them.

This paper is divided into three sections. In Section 2, the general inverse kinematic solutions and Jacobian inverse methods are presented for 6R-WP robot. In Section 3, the trajectories traced by the robot for a given task using the two algorithms are shown. Finally the conclusions are given in Section 5.

## 2. METHODOLOGY

This section explains in brief the formulation used behind the inverse kinematics of a 6R-WP robot. Two methods, namely, analytical method and the Jacobian inverse method are discussed.

### 2.1 Inverse Kinematics

Geometric model of the serial robot with wrist-partitioned feature i.e., the DH parameters is given in Table 2. Kinematic diagram of the corresponding robot is shown in the Figure 2(a). The DH parameters are taken from its identified value presented in [10]. The net transformation matrix to obtain the position and orientation of the EE can be given as,

$$\mathbf{T} = \prod_{i=1}^6 \mathbf{T}_i = \underbrace{(\mathbf{T}_1 \mathbf{T}_2 \mathbf{T}_3)}_{\text{Arm}} \underbrace{\mathbf{T}_4 \mathbf{T}_5 \mathbf{T}_6}_{\text{Wrist}} \quad (1)$$

where the  $4 \times 4$  matrix  $\mathbf{T}_i$  is given by

$$\mathbf{T}_i \equiv \begin{bmatrix} \mathbf{Q}_i & \mathbf{p}_i \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & b_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The position  $\mathbf{p}_{EE}$  and orientation  $\mathbf{Q}_{EE}$  of the EE for six-DOF robot can be explicitly expressed as:

$$FK(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = (X, Y, Z, \phi, \theta, \psi) \quad (3)$$

$$\mathbf{p}_{EE} = \sum_{i=1}^6 \mathbf{a}_i \text{ and } \mathbf{Q}_{EE} = \prod_{i=1}^6 \mathbf{Q}_i \quad (4)$$

where  $\mathbf{p}$  is the position vector,  $\mathbf{Q}$  is the  $3 \times 3$  rotation or orientation matrix,  $\mathbf{0} \equiv [0, 0, 0]^T$  is the three-dimensional vector of zeros, whereas the  $4 \times 4$  matrix,  $\mathbf{T}$ , is called the homogeneous transformation matrix (HTM) of the end-effector. Based on Equation (1) – (4), once can define the IK solutions as :

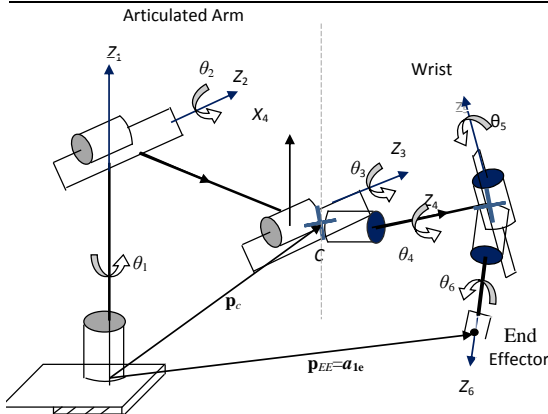
$$IK(X, Y, Z, \phi, \theta, \psi) = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) \quad (5)$$

The detailed derivation to find the joint angles has been discussed in [11],[1]. Note that the point  $C$  in Figure 2(a) shows the intersection of axes 4, 5, and 6. Hence it is independent of those joints and function only the first three joints. First, joints angles 1, 2, and 3 are solved before obtaining the joint angles 4, 5, and 6. The motion of the robot end-effector can be obtained by dividing the path into a number of intermediate points and moving the end effector from point-to-point.

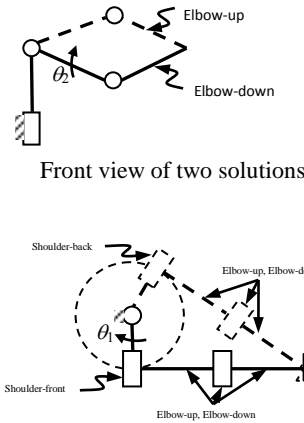
**Table 2. DH parameters of a 6R-WP robot KUKA KR5 Arc [9].**

No	$b_i$ (mm)	$\theta_i$ ( $^\circ$ )	Minimum joint limit ( $^\circ$ )	Maximum joint limit ( $^\circ$ )	$a_i$ (mm)	$\alpha_i$ ( $^\circ$ )
1	400	$\theta_1$	-155	155	180	$\pi/2$
2	0	$\theta_2$	-65	180	600	$\pi/2$
3	0	$\theta_3$	-68	105	120	0
4	620	$\theta_4$	-350	350	0	$\pi/2$
5	0	$\theta_5$	-130	130	0	$\pi/2$
6	0	$\theta_6$	-350	350	0	0

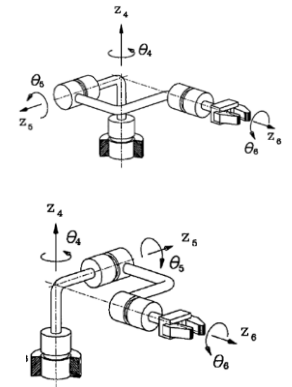
In the inverse kinematics solution, the required joint angles corresponding to the subsequent points are calculated. This yield



a) Kinematic model of 6R robot with last three links intersecting



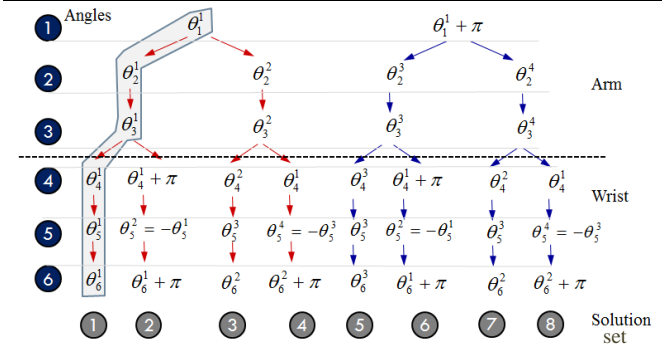
b) Top view of other two solutions [1]



c) Two solutions for the wrist [10]

**Figure 2. Kinematic diagram of a six-DOF wrist-partitioned robot with its inverse kinematic solutions**

multiple solutions. Figure 2(b) and Figure 2(c) gives four sets of  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  by solving a quartic equation and two sets for  $\theta_4$ ,  $\theta_5$ , and  $\theta_6$  respectively. Hence, a 6R-WP robot can have up to eight possible configurations for a given pose of the EE. They are indicated in Figure 3.



**Figure 3. Eight inverse kinematics solutions**

Each link joint has some mechanical limit which was mentioned in Table 2. The IK solutions for an EE pose specified by the user, needs to be checked for the joint limits first. If it exceeds the given value of joint limits of the robot, those values get discarded.

The wrapping of angles is widely used in the robots where the angles are obtained from the encoder values. Wrapping of the IK solutions in the range of  $(-\pi, \pi)$  is necessary for the continuous and smooth motion of the robot. The algorithm for wrapping the angles in the interval is given below.

With several IK solutions, the appropriate set of solution among the eight solutions, as shown in Figure 3, must be selected in order to have a smooth motion of the robot. This is very important

*Algorithm #1: Wrap angles in interval  $[-\pi, \pi]$*

Theta [i][j]

*<comment> 'i' is number of IK solutions, i.e., 8*

*<comment> 'j' is number of joint angles, i.e., 6 here*

**for** i=1to8

**for** j=1to6

Theta [i][j]= Theta [i][j] % (2 $\pi$ ) *<comment> '%' is modulo operator*

**if** Theta[i][j] < pi

Theta[i][j]= Theta [i][j] + (2 $\pi$ )

**elseif** Theta[i][j] > pi

Theta[i][j]= Theta [i][j] - (2 $\pi$ )

**return** Theta[i][j]

to avoid sudden change in robot configuration while moving from one point to another. Four methods are listed for selecting one of the eight possible solutions.

### 2.1.1 Minimum deviation of all joint angles

Minimum deviation in joint angles is desired to avoid any sudden change in the configuration of the robot while in motion. This index utilizes the previous joint states to select the next set of solution among the eight IK solutions.

$$\eta = \min \left( \sum_{i=1}^6 \left\| \theta_{i,current} - \theta_{i,next} \right\|^2 \right) \quad (6)$$

where  $\theta_{i,current}$  is the column vector of joint angles corresponding to the current configuration of end-effector, and  $\theta_{i,next}$  is the  $i^{th}$  joint angle for the next configuration. Minimum norm set of angles are selected among the eight IK solutions of the next step.

### 2.1.2 Minimum deviation in first three joints

The first three joints of a six axis WP robot is generally used for positioning of the end-effector of a robot. For most of the industrial robots, first three joints supply the maximum torque for the motion of the end-effector. Hence it is preferable to minimize the motion of the first three joints while moving from the current to the next point, without compromising on the accuracy. Using this rationale, the next joint configuration is chosen out of the possible multiple inverse kinematics. The solution that has minimum variation is selected as the joint values for the next point in the path. In this way, elbow-up and elbow-down configurations are insured.

$$\eta = \min \left( \sum_{i=1}^3 \left\| \theta_{i,current} - \theta_{i,next} \right\|^2 \right) \quad (7)$$

### 2.1.3 Weighted deviation of joints

As discussed in Section 2.1.2, the first three joints have to supply a relatively higher joint torques for robot motion. It is natural that while going from one configuration to the next, the motion of those joints must be minimized. In order to account for the same while choosing the next solution, the following index is utilized, by putting some weights to the first three joints, i.e.

$$\eta = \min \left( \sum_{i=1}^3 \left\| (\theta_{i,current} - \theta_{i,next}) w_i \right\|^2 \right) \quad (8)$$

where  $w_i$  is the weight which was chosen proportionally to the inverse of the mass of each link.

### 2.1.4 Manipulability

The manipulability of a robot [12] can be used as a criterion to select an appropriate solution from the multiple inverse kinematics solutions. Applying the maximum manipulability criterion in each step of the trajectory may result in flip configuration in between. Hence, it is applied once for selecting the pose for the starting position only.

$$\eta = \max \left( \sqrt{\det(\mathbf{J}(\theta_i) \mathbf{J}^T(\theta_i))} \right) \quad (9)$$

## 2.2 Jacobian inverse method

In robotics, the Jacobian can be interpreted as the time derivative of the kinematic equations which relates the velocity of the end-effector to the joint rates. The expression for the Jacobian is given as follows

$$\omega_e = \mathbf{J}_\omega \dot{\theta} \quad \text{and} \quad \mathbf{v}_e = \mathbf{J}_v \dot{\theta} \quad (10)$$

where  $\mathbf{J}_\omega$  and  $\mathbf{J}_v$  are the  $3 \times 6$  matrices relating the contribution of the joint velocities or rates  $\dot{\theta}$  to the end-effector angular

velocity  $\omega_e$  and velocity  $\mathbf{v}_e$  respectively. Also  $\mathbf{J}_\omega$  and  $\mathbf{J}_v$  are the function of  $\boldsymbol{\theta}$ . Equation (10) can be re-written as

$$\Delta \mathbf{x}_e = \mathbf{J} \Delta \boldsymbol{\theta} \quad (11)$$

where  $[\Delta \phi \Delta \theta \Delta \psi \Delta x \Delta y \Delta z]^T = \mathbf{J} [\Delta \theta_1 \Delta \theta_2 \Delta \theta_3 \Delta \theta_4 \Delta \theta_5 \Delta \theta_6]^T$

The term  $\Delta \mathbf{x}_e$  is the change in the pose of the end-effector corresponding to the change in joint angles  $\Delta \boldsymbol{\theta}$ . The Jacobian  $\mathbf{J}$  is the 6x6 matrix for a 6-DOF robot, which is obtained by the recursive relation of the angular velocity and linear velocities for 6R manipulator. It is given as

$$\mathbf{J} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_6 \\ \mathbf{e}_1 \times \mathbf{a}_{1,e} & \mathbf{e}_2 \times \mathbf{a}_{2,e} & \cdots & \mathbf{e}_6 \times \mathbf{a}_{6,e} \end{bmatrix} \quad (12)$$

in which,  $\mathbf{a}_{1,e}$  is the vector as shown in Figure 2 (a) and  $\mathbf{e}_1$  is the joint axis direction of joint 1. Similarly  $\mathbf{e}_2$  is the joint axis direction and  $\mathbf{a}_{2,e}$  is the position of the EE from joint 2, and so on. Each column of the Jacobian consists of the differential rotation and translation vector corresponding to the differential change in the joint rates. In order to obtain the joint variation for the desired increment at the  $k^{\text{th}}$  point pose is given by the Jacobian inverse as below,

$$\Delta \boldsymbol{\theta}_k = \mathbf{J}_k^{-1} \Delta \mathbf{x}_e \quad (13)$$

where  $\Delta \mathbf{x}_e$  is the desired increment in the pose of the EE. The efficient way to decide the increment was given in [13]. The joint angles required to reach the  $k^{\text{th}}$  point, which are similar to the first order Taylor series expansion of the joint angle evaluated as

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \mathbf{J}_k^{-1} \Delta \mathbf{x}_e \quad (14)$$

$$\text{In general, } \boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \mathbf{J}^\dagger \Delta \mathbf{x} + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}) \Delta \boldsymbol{\phi} \quad (15)$$

where  $\mathbf{J}^\dagger = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}$  is pseudo inverse of  $\mathbf{J}$  matrix and  $(\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})$  is the null space of  $\mathbf{J}$  with  $\Delta \boldsymbol{\phi}$  as objective function. Here for manipulator with DOF=6, (14) is used as the Jacobian matrix is square and it can be inverted.

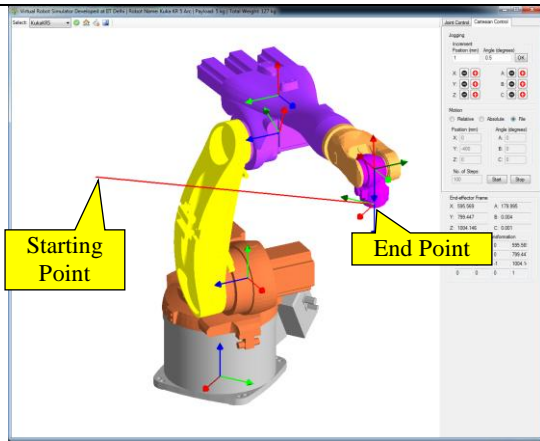
### 3. RESULTS AND DISCUSSIONS

The angular positions obtained from the IK are updated by the controller of the robot. The control frequency of a typical industrial robot generally varies between 10 to 50 Hz that determines the time instance in which the joint position values are updated and commanded by the controller.

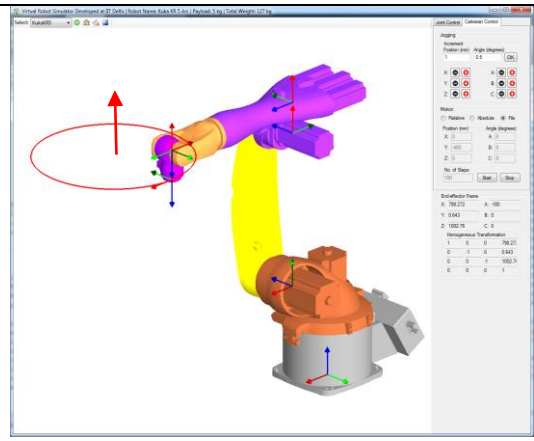
The inverse kinematics and the Jacobian inverse method explained in Sections 2.1 and 2.2 respectively, are implemented in MATLAB environment for a linear and circular trajectory. For animation of the robot motion, the joint angles obtained using IK were provided to Virtual Robot Module (VRM) [14] of RoboAnalyzer software. They are shown Figure 4. The joint angles calculated were given as input to the VRM for the animation and visualization purpose. These algorithms were tested on Intel® Core, i7-3770 CPU and 3.40 GHz with 8 GB RAM, 64 bit Windows 7 OS. The comparisons of the approaches given in Section 2 are now discussed in following sections.

#### 3.1 Straight line motion

Generally, a robot programmer specifies the initial and final points, as shown in Figure 4(a). The values of  $[X, Y, Z, \phi, \theta, \psi]$  at the starting and at the end positions are  $[0.8, -0.4, 1, 0, 0, \pi]$  and  $[0.6, 0.4, 1, 0, 0, \pi]$  respectively. The desired motion between the two points is a straight line. The values of X, Y and Z are in meters and the orientations  $\phi, \theta$ , and  $\psi$  are in radians.

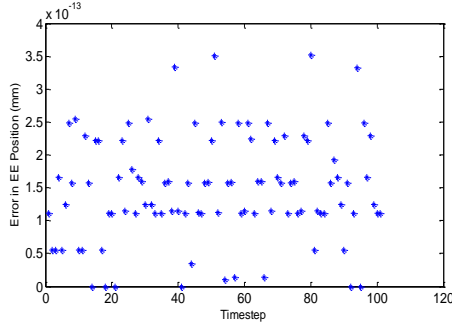


a) Straight line motion

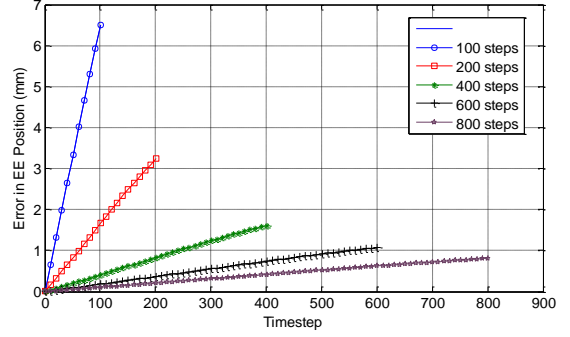


b) Circular motion

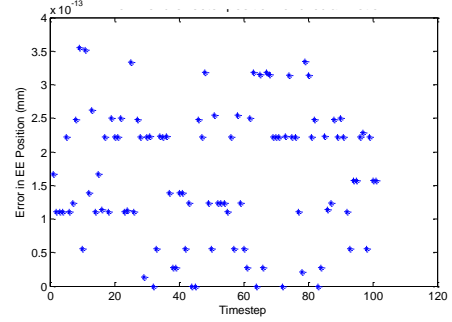
Figure 4. The desired path to be traced by the robot.



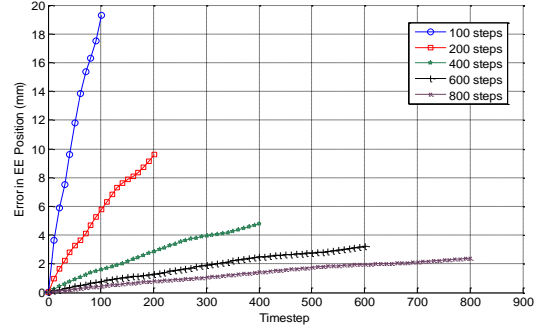
(a) Error in end-effector position (straight line, inverse kinematics)



(b) Error in end-effector position (straight line, Jacobian inverse)



(c) Error in end-effector position (circular, inverse kinematics)



(d) Error in end-effector position (circular, Jacobian inverse)

**Figure 5. Comparison of two methods for straight line and circular trajectories.**

Figure 4(a) shows the straight line motion traced by the robot. The inverse kinematic method gave positioning errors of the order of  $10^{-13}$  mm between commanded and desired position of the end-effector for straight line motion as shown in Figure 5 (a). Table 4 lists the time taken using different selection criteria proposed in Section 2.1.4 for tracing the straight line.

**Table 3. Results for linear trajectory using Jacobian inverse**

S.No	No. of steps	Straight line motion		Circular motion	
		Error (mm)	Time (s)	Error (mm)	Time (s)
1	100	6.49	0.508	19.42	0.534
2	200	3.25	0.941	9.87	0.957
3	400	1.86	1.770	4.73	1.821
4	600	1.01	2.606	3.47	2.663
5	800	0.89	3.465	2.43	3.502

The Jacobian inverse method was also tested. The difference between the desired EE position and the position reached using the Jacobian inverse method for 100 steps is 6.49 mm. Note that it is an approximate method. Hence, if the number of steps is increased it is expected to reduce the errors. This is evident from Table 3 and Figure 5(b). When the number of steps is 800, the error is only 0.89mm. However, the computation time increases. Table 3 also lists the time taken to generate the results, i.e., the joint angles to achieve the desired path (in this case, the straight line).

### 3.2 Circular motion

Here the robot programmer needs to specify the center, radius and normal of the circle which was required to be traced by the robot. The center  $C$  of the circle was taken as (0.8, 0, 0.9) m and the radius of the circle specified as 0.3 m.

The axis of the circle was kept parallel to the joint axis 1 which is shown in Figure 4(b). The end effector was at a constant orientation. Using the inverse kinematics method an error of the order  $10^{-13}$  mm was observed. Table 4 lists the time taken.

Here the robot programmer needs to specify the center, radius and normal of the circle which was required to be traced by the robot. The center  $C$  of the circle was taken as (0.8, 0, 0.9) m and the radius of the circle specified as 0.3 m. The axis of the circle was kept parallel to the joint axis 1 which is shown in Figure 4(b). The end effector was at a constant orientation. Using the inverse kinematics method an error of the order  $10^{-13}$  mm was obtained as shown in Figure 5(c). Table 4 lists the time taken with different step size using different selection criteria of IK solutions.

The Jacobian inverse method resulted in higher positioning errors (19.42 mm) compared to inverse kinematics method, where the positioning errors were almost negligible. Figure 5(d) illustrates that by increasing the number of intermediate points or reducing the step size the error of EE positioning reduces.

It was found that the Jacobian inverse solution has greater error while tracing a circular trajectory than the straight line motion. This is quite obvious as the method is based on straight line approximations. Hence, more the curvature of a path, more error is expected. The inverse kinematic method on the other hand is



not affected by the step size as the calculations are exact. The step size for calculating the intermediate configurations here depends on the control frequency and the resolution of the robot.

**Table 4. Time taken in inverse kinematics method (100 steps)**

Criterion for inverse kinematics	Straight line (in sec)	Circular motion (in sec)
Minimum deviation of all joint angles	4.338	4.468
Minimum deviation in first three joints	4.319	4.341
Weighted deviation of joints	4.338	4.423
Manipulability	4.412	4.452

## 4. CONCLUSIONS

This work has reported the comparison between two widely used methods for manipulation and animation of a typical industrial robot, namely the inverse kinematic and the Jacobian inverse methods. Linear and circular trajectories were selected for comparing the two approaches. The step by step method to obtain IK solutions and selecting criteria of one solution among eight solutions was given. Among the four criteria for selecting solutions, least computation time is for checking minimum deviation in first three joint angles, but the user can select other criteria depending on the requirement and task. The wrist partitioned six-DOF robot results in attaining far better accuracy in positioning the robot by commanding them calculated IK joint space than the numerical method, i.e., Jacobain inverse technique. Though increasing the step size in numerical methods reduces the error at the cost of increased computation. In animation environment these algorithms were successfully implemented and compared. In future adaptive learning techniques for IK will be interesting to implement and compare.

## 5. ACKNOWLEDGMENTS

The financial support to the first two authors is from the sponsored project entitled “Adaptive Force Control of an Industrial Robot (KUKA KR5) Equipped with Force/Torque Sensor” by BRNS/BARC Mumbai under the setting up of “Programme for Autonomous Robotics Lab” at IIT Delhi is sincerely acknowledged.

## 6. REFERENCES

- [1] Saha, S. K., 2014. Introduction to Robotics, 2nd ed.: Tata McGraw-Hill Education.
- [2] Siciliano, B., and Khatib, O., 2008. Springer Handbook of Robotics: Springer.
- [3] Peiper, D. L., 1968. The kinematics of manipulators under computer control, DTIC Document.
- [4] Manseur, R., and Doty, K.L., 1989. A robot manipulator with 16 real inverse kinematic solution sets, *The International Journal of Robotics Research*, vol. 8, no. 5, pp. 75-79.
- [5] Goldenberg, A. A., Benhabib, B., and Fenton, R. G., 1985. A complete generalized solution to the inverse kinematics of robots, *Robotics and Automation, IEEE Journal of*, vol. 1, no. 1, pp. 14-20.
- [6] Wampler, C. W., 1986. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-

Squares Methods, *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 16, no. 1, pp. 93-101.

[7] Lapresté, J.T., Jurie, F. M., Dhome, M., and Chaumette, F., 2004. An efficient method to compute the inverse jacobian matrix in visual servoing. *IEEE International Conference on Robotics and Automation*, pp. 727-732.

[8] Fijany, A., and Bejczy, A.K., 1988. Efficient Jacobian inversion for the control of simple robot manipulators. *IEEE International Conference on Robotics and Automation*, vol.2, pp. 999-1007.

[9] Vasilyev, I.A. and Lyashin, A.M., 2010. Analytical solution to inverse kinematic problem for 6-DOF robot-manipulator, *Automation and Remote Control*, vol. 71, no. 10, pp. 2195-2199

[10] Hayat, A.A., Chittawadigi, R.G., Udai, A.D., and Saha, S.K., 2013. Identification of Denavit-Hartenberg Parameters of an Industrial Robot. pp. 1-6.

[11] Angeles, J., 1997. *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*, Springer, 1997.

[12] Yoshikawa, T., 1985. Manipulability of robotic mechanisms, *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3-9.

[13] Castellet, A., and Thomas, F., 1997. Towards an efficient interval method for solving inverse kinematic problems, *IEEE International Conference on Robotics and Automation*, pp. 3615-3620.

[14] Rajeevlochana, C.G., and Saha, S.K., 2011. RoboAnalyzer: 3D model based robotic learning software. *International Conference on Multibody dynamics*, pp 3-13.