Solving Kinematics Problems of a 6-DOF Robot Manipulator

Alireza Khatamian
Computer Science Department, The University of Georgia, Athens, GA, U.S.A

Abstract - *This paper represents an analytical approach for solving forward kinematics problem of a serial robot manipulator with six degrees of freedom and a specific combination of joints and links to formulate the position of the gripper by a given set of joint angles. In addition, a direct geometrical solution for the robot's inverse kinematics problem will be defined in order to calculate the robot's joint angles to pose the gripper in a given coordinate. Furthermore, the accuracy of the two solutions will be shown by comparing the results in a developed simulation program which uses the Unified System for Automation and Robot Simulation (USARSim).*

Keywords: Kinematics, Geometric, Robot Manipulators

1 Introduction

Automation and hazard reduction in the workplace are two important factors of driving the use of robotics, specially articulated arm robots, in order for decreasing human deaths and injuries, economically affordable production plus saving time. Therefore, high precision control is the basic and primary step of using these robots.

In brief, six degrees of freedom, three of which are used for posing the gripper at a specific position and the other three for orientation adjustment, is an essential dexterity characteristic of a space robot manipulator. Hence, the focus of this paper is on controlling a six degree-of-freedom arm robot with a particular series of links and joints.

[3] represents the kinematics of nine common industrial robots including forward kinematics, inverse kinematics plus Jacobian matrix which is used in higher level arm robots calculations. In [4], general equations for a human-arm-like robot manipulators have been presented and [5] introduces kinematics solutions for robot manipulators based on their structures. A novel recurrent neural network controller with learning ability to maintain multiple solutions of the inverse kinematics is introduced in [8]. [9] proposes a nonlinear programming technique for solving the inverse kinematics problem of highly articulated arm robots in which finding the joint angles is hard for a desired joint configuration (inverse kinematics). In [10] a new combined method is proposed to solve inverse kinematics problem of six-joint Stanford robotic manipulator by using genetic algorithms and neural networks in order to obtain more precise solutions and to minimize the error at the end effector final position. A novel heuristic method, called Forward And Backward Reaching Inverse Kinematics (FABRIK), is described and compared with some of the most popular existing methods regarding reliability, computational cost and conversion criteria in [11]. [12] represents a hybrid combination of neural networks and fuzzy logic intelligent technique to solve the inverse kinematics problem of a three degree of freedom robotic manipulator. In [13], an algorithm in which the independent components of link lengths are used as a medium to analyze the forward kinematics of a six-degree-of-freedom Stewart platform can be found and an extension to closed –loop inverse kinematics (CLIK) algorithm is introduced in [14] to meet some applications that require the joint acceleration.

A fundamental phase for controlling a robot manipulator is solving the robot's kinematics problems. Kinematics is the science of motion that treats the subject without regard to the forces that cause it [1]. Kinematics has two aspects: (1) Forward Kinematics and (2) Inverse Kinematics. The former is getting the position and orientation of the end effector of the robot by having the joint angles. The latter is setting joint angles to place the end effector of the robot in a given position and orientation.

Forward kinematics problem of a serial manipulator has a straight forward solution which formulates the position and orientation of the robot's end effector by a series of joint angles. On the other hand, having more degrees of freedom on a serial manipulator elaborates solving the inverse kinematics problem of the robot. There are two approaches to solve the inverse kinematics problem: (1) numerical method and (2) geometrical approach.

The purpose of this paper is to present an analytical method and a geometrical approach for solving forward and inverse kinematics problem of a particular six degree-of-freedom serial manipulator, accordingly. The target robot is a KUKA KR60 which is an industrial manipulator production of KUKA Corporation.

The rest of this paper is organized as follows: in section II, robot parameters and specifications are described, section III and IV represent forward and inverse kinematics solutions, accordingly. Experimental results and precision assessment are provided in section V, and section VI concludes the paper.

2 Robot Specification

To describe a robot manipulator specifically, joint types and the relation between links, which is the exposure mode of two operational axes of two consecutive joints (parallel or perpendicular), have to be explained.

KUKA KR60 is an industrial robot designed by seven links which are connected to each other by six revolute joints. All the joints of this robot are the same and there is no prismatic, cylindrical, planar or any other type of joint in the structure of the robot.

The functional state of each joint related to its successive joint in the design of this robot is as follows:

$$R_1 \perp R_2 \parallel R_3 \perp R_4 \perp R_5 \perp R_6 \tag{1}$$

in which R indicates a revolute joint and the indices describe the position of the joint relative to the base of the robot.

Figure 1 displays a symbolic structure of a KUKA KR60 and the attached frames to the joints in addition to the effective axis of the joints which is always along the Z axis and also length of the links.

A robot manipulator's crucial aspect is its Denavit-Hertenberg parameters or D-H parameter table. It specifies all the characteristics of a robot manipulator including link lengths and relative orientation of the joints. Table 1 shows the D-H parameters of a KUKA KR60.

3 Forward Kinematics

Forward kinematics problem is finding the position and orientation of the end effector of the robot by a given set of joint angles and also having D-H parameters of the robot. This section explains an analytical method for solving the forward kinematics problem of a KUKA KR60.

A robot manipulator's forward kinematics problem is solved by attaching a single frame to each joint along with the robot's base. Each frame describes the position and orientation of each joint of the robot relative to the base or any other global coordinate. Attaching these frames to the joints reduces the calculation of the robot's end effector's position and orientation to a coordinate translation problem which is solved by transformation matrices.

Therefore, every joint has a position and orientation relative to its previous joint. These relations are described by transformation matrices. A general formulation for calculation of these matrices is as follows:

TABLE 1
D-H parameters of a KUKA KR60

i	***a***	***α***	***d***	***θ***
0	350	0	–	–
1	0	$\frac{\pi}{2}$	−815	$0+\theta_1$
2	850	0	0	$-\frac{\pi}{2}+\theta_2$
3	145	$\frac{\pi}{2}$	0	$0+\theta_3$
4	0	$-\frac{\pi}{2}$	−820	$\pi+\theta_4$
5	0	$\frac{\pi}{2}$	0	$\pi+\theta_5$
6	–	–	170	$0+\theta_6$

$${}^{i-1}_{\ \ i}T = R_X(\alpha_{i-1})D_X(a_{i-1})R_Z(\theta_i)D_Z(d_i) \tag{2}$$

in which $R_X(\alpha_{i-1})$ is a rotation matrix about the X axis by α_{i-1}, $D_X(a_{i-1})$ is translation matrix along the X axis by a_{i-1}, $R_Z(\theta_i)$ is a rotation matrix about the Z axis by θ_i, $D_Z(d_i)$ is translation matrix along the Z axis by d_i and α, a, θ and d are D-H parameters of the robot. So we have:

$$R_X(\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i-1} & -\sin\alpha_{i-1} & 0 \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$D_X(a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$R_Z(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$D_Z(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

and ${}^{i-1}_{\ \ i}T$ is,

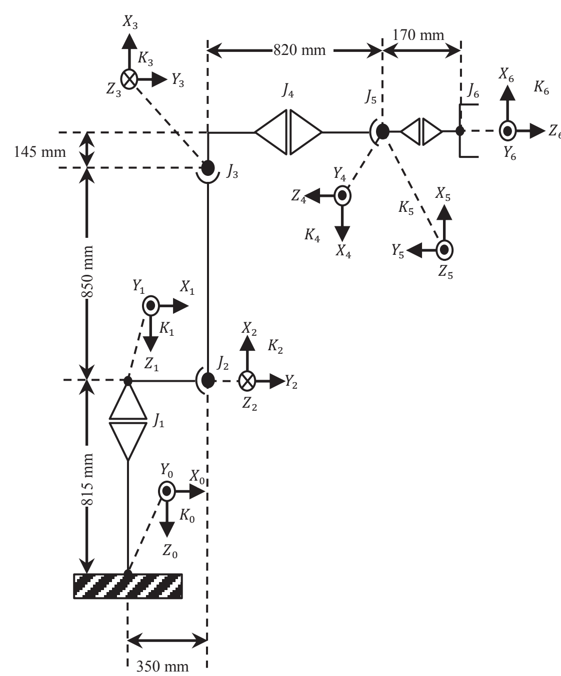

Figure 1. Symbolic structure of a KUKA KR60, link length, the attached frames to the joints and the operational axis which is along the Z axis

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\alpha_{i-1}\sin\theta_i & \cos\alpha_{i-1}\cos\theta_i & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\alpha_{i-1}\sin\theta_i & \sin\alpha_{i-1}\cos\theta_i & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Hence, the following matrix multiplication computes the final transformation matrix that gives the position and orientation of the robot's end effector relative to its base.

$${}^0_6T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T \quad (8)$$

Let,

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

The position and the orientation of the end effector in roll-pitch-yaw representation are as follows:

$${}^0\mathrm{P}_6 = \begin{bmatrix} \mathrm{r}_{14} \\ \mathrm{r}_{24} \\ \mathrm{r}_{34} \end{bmatrix} \quad (10)$$

$$\mathrm{pitch} = \mathrm{Atan2}\left(\mathrm{r}_{13}, \sqrt{\mathrm{r}_{23}^2 + \mathrm{r}_{33}^2}\right) \quad (11)$$

$$\mathrm{roll} = \begin{cases} 0 & pitch = \frac{\pi}{2}, -\frac{\pi}{2} \\ \mathrm{Atan2}\left(-\frac{\mathrm{r}_{23}}{\cos(\mathrm{pitch})}, \frac{\mathrm{r}_{33}}{\cos(\mathrm{pitch})}\right) & o.w \end{cases} \quad (12)$$

$$\mathrm{yaw} = \begin{cases} \mathrm{Atan2}(\mathrm{r}_{32}, \mathrm{r}_{22}) & pitch = \frac{\pi}{2} \\ -\mathrm{Atan2}(\mathrm{r}_{32}, \mathrm{r}_{22}) & pitch = -\frac{\pi}{2} \\ \mathrm{Atan2}\left(-\frac{\mathrm{r}_{12}}{\cos(\mathrm{pitch})}, \frac{\mathrm{r}_{11}}{\cos(\mathrm{pitch})}\right) & o.w \end{cases} \quad (13)$$

4 Inverse Kinematics

Inverse kinematics problem of a robot manipulator is finding the joint angles of the robot by having the position and orientation of the end effector of the robot. Inverse kinematics problem of a serial manipulator is more important than the forward kinematics, as it is essential to move the gripper of the robot to a required position with a defined orientation in order to, for instance, grab an object in that position and orientation.

There are two approaches to solve the inverse kinematics problem of a robot manipulator; mathematical or algebraic and geometrical. The higher degrees of freedom requires the more complicated algebraic solution. Therefore, this section has been devoted to present a geometrical solution for the inverse kinematics problem of a KUKA KR60.

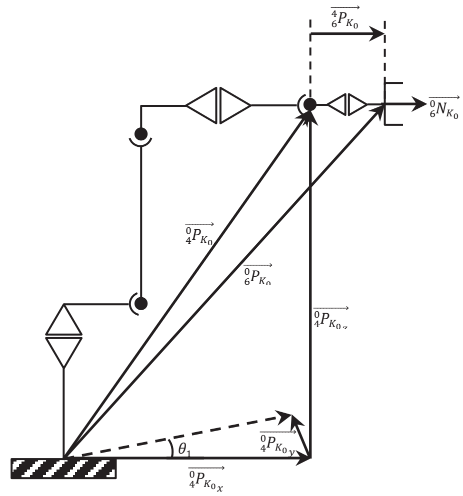

Figure 2. Geometrical representation of first joint angle calculation

In a geometrical method, vectors describe the robot's state to solve the problem which is the calculation of the joint angles of the robot. This section is divided into five subsections to illustrate the joint angles' computing method.

A. Joint 1

The first joint angle's calculation, as shown in Figure 2, is accomplished by the projection of a vector which originates from the origin of frame K_0 and ends to the origin of frame K_4 ($\overrightarrow{{}^0_4\mathrm{P}_{K_0}}$) on the $\mathrm{X-Y}$ plane of frame K_0.

Let ${}^0_\mathrm{G}\mathrm{T}$ be the target transformation matrix relative to the base which defines the target position and orientation.

$${}^0_GT = \begin{bmatrix} {}^0_GT_{11} & {}^0_GT_{12} & {}^0_GT_{13} & {}^0_GT_{14} \\ {}^0_GT_{21} & {}^0_GT_{22} & {}^0_GT_{23} & {}^0_GT_{24} \\ {}^0_GT_{31} & {}^0_GT_{32} & {}^0_GT_{33} & {}^0_GT_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \overrightarrow{{}^0_6N_{K_0}} = \begin{bmatrix} {}^0_GT_{13} \\ {}^0_GT_{23} \\ {}^0_GT_{33} \end{bmatrix} \quad (14)$$

then we have,

$$\begin{cases} \overrightarrow{{}^4_6P_{K_0}} = d_6 \times \overrightarrow{{}^0_6N_{K_0}} \\ \overrightarrow{{}^0_6P_{K_0}} = \begin{bmatrix} {}^0_GT_{14} \\ {}^0_GT_{24} \\ {}^0_GT_{34} \end{bmatrix} \end{cases} \quad (15)$$

$$\Rightarrow \overrightarrow{{}^0_4P_{K_0}} = \overrightarrow{{}^0_6P_{K_0}} - \overrightarrow{{}^4_6P_{K_0}} = \begin{bmatrix} {}^0_GT_{14} - d_6\,{}^0_GT_{13} \\ {}^0_GT_{24} - d_6\,{}^0_GT_{23} \\ {}^0_GT_{34} - d_6\,{}^0_GT_{33} \end{bmatrix}$$

so,

$$\theta_1 = \begin{cases} Atan2({}^0_GT_{24} - d_6\,{}^0_GT_{23}, {}^0_GT_{14} - d_6\,{}^0_GT_{13}) \\ Atan2({}^0_GT_{24} - d_6\,{}^0_GT_{23}, {}^0_GT_{14} - d_6\,{}^0_GT_{13}) + \pi \end{cases} \quad (16)$$

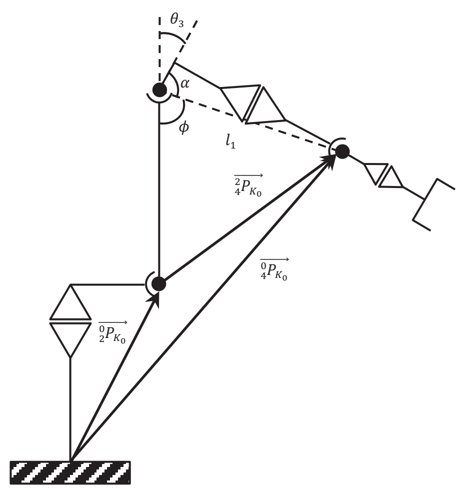

Figure 3. Visual representation of joint 3 angle calculation

B. *Joint 3*

Based on Figure 3's illustration, to calculate θ_3, first $\overrightarrow{{}^2_4P_{K_0}}$ needs to be calculated. In order to compute $\overrightarrow{{}^2_4P_{K_0}}$, $\overrightarrow{{}^0_2P_{K_0}}$ should be available, beforehand. By having $\overrightarrow{{}^2_4P_{K_0}}$ and l_1, ϕ can be calculated and then by using a simple geometric rule, which helps to compute the angle of between two edges of a triangle, α will be quantified.

Let $\theta_2 = 0$ and,

$$
{}^0_2T = \begin{bmatrix} {}^0_2T_{11} & {}^0_2T_{12} & {}^0_2T_{13} & {}^0_2T_{14} \\ {}^0_2T_{21} & {}^0_2T_{22} & {}^0_2T_{23} & {}^0_2T_{24} \\ {}^0_2T_{31} & {}^0_2T_{32} & {}^0_2T_{33} & {}^0_2T_{34} \end{bmatrix} \Rightarrow \overrightarrow{{}^0_2P_{K_0}} = \begin{bmatrix} {}^0_2T_{14} \\ {}^0_2T_{24} \\ {}^0_2T_{34} \end{bmatrix} \tag{17}
$$

thus,

$$
\overrightarrow{{}^2_4P_{K_0}} = \overrightarrow{{}^0_4P_{K_0}} - \overrightarrow{{}^0_2P_{K_0}} = \begin{bmatrix} \overrightarrow{{}^2_4P_{K_0}}_x \\ \overrightarrow{{}^2_4P_{K_0}}_y \\ \overrightarrow{{}^2_4P_{K_0}}_z \end{bmatrix} \tag{18}
$$

$$
\phi = Asin\left(\frac{\left(l_1^2 - a_2^2 + \left|\overrightarrow{{}^2_4P_{K_0}}\right|^2\right)}{2\left|\overrightarrow{{}^2_4P_{K_0}}\right| l_1}\right) + Asin\left(\frac{\left|\overrightarrow{{}^2_4P_{K_0}}\right| - \dfrac{l_1^2 - a_2^2 + \left|\overrightarrow{{}^2_4P_{K_0}}\right|^2}{2\left|\overrightarrow{{}^2_4P_{K_0}}\right|}}{a_2}\right) \tag{19}
$$

$$
\alpha = Atan2(-d_4, a_3) \tag{20}
$$

So,

$$
\theta_3 = \begin{cases} \pi - \phi - \alpha \\ \pi + \phi - \alpha \end{cases} \tag{21}
$$

C. *Joint 2*

θ_2 is computed by $\overrightarrow{{}^2_4P_{K_2}}$, β_1 and β_2, as Figure 4 displays.

$$
\overrightarrow{{}^2_4P_{K_2}} = {}^2_0R\,\overrightarrow{{}^2_4P_{K_0}} = {}^0_2R^{-1}\,\overrightarrow{{}^2_4P_{K_0}} \tag{22}
$$

$$
{}^0_2T = \begin{bmatrix} {}^0_2R & {}^0_2P_{ORG} \\ 0 & 1 \end{bmatrix} \Rightarrow
$$

$$
{}^0_2R = \begin{bmatrix} {}^0_2T_{11} & {}^0_2T_{12} & {}^0_2T_{13} \\ {}^0_2T_{21} & {}^0_2T_{22} & {}^0_2T_{23} \\ {}^0_2T_{31} & {}^0_2T_{32} & {}^0_2T_{33} \end{bmatrix} = {}^2_0R^{-1} \tag{23}
$$

$$
\overrightarrow{{}^2_4P_{K_2}} = \begin{bmatrix} {}^0_2T_{11} & {}^0_2T_{12} & {}^0_2T_{13} \\ {}^0_2T_{21} & {}^0_2T_{22} & {}^0_2T_{23} \\ {}^0_2T_{31} & {}^0_2T_{32} & {}^0_2T_{33} \end{bmatrix} \begin{bmatrix} \overrightarrow{{}^2_4P_{K_0}}_x \\ \overrightarrow{{}^2_4P_{K_0}}_y \\ \overrightarrow{{}^2_4P_{K_0}}_z \end{bmatrix} \tag{24}
$$

Thus,

$$
\beta_1 = Atan2\left(\overrightarrow{{}^2_4P_{K_2}}_x, \overrightarrow{{}^2_4P_{K_2}}_y\right) \tag{25}
$$

$$
\beta_2 = Asin\left(\frac{a_2^2 - \left|\overrightarrow{{}^2_4P_{K_0}}\right|^2 + l_1^2}{2l_1a_2}\right) + Asin\left(\frac{l_1 - \dfrac{a_2^2 - \left|\overrightarrow{{}^2_4P_{K_0}}\right|^2 + l_1^2}{2l_1}}{\left|\overrightarrow{{}^2_4P_{K_0}}\right|}\right) \tag{26}
$$

and then,

$$
\theta_2 = \begin{cases} \dfrac{\pi}{2} - (|\beta_1| + \beta_2) \\ \dfrac{\pi}{2} + (|\beta_1| - \beta_2) \end{cases} \tag{27}
$$

D. *Joint 5*

In order to calculate θ_5, ${}^0_4T$ is computed by assuming θ_4 is equal to 0. Then by using the definition of dot product of two normal vectors which are shown in Figure 5, θ_5 is obtained.

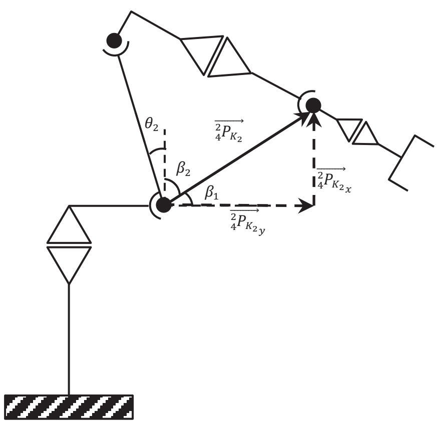

Figure 4. Joint 2 angle calculation vector representation

$$ {}^0_4T = \begin{bmatrix} {}^0_4T_{11} & {}^0_4T_{12} & {}^0_4T_{13} & {}^0_4T_{14} \\ {}^0_4T_{21} & {}^0_4T_{22} & {}^0_4T_{23} & {}^0_4T_{24} \\ {}^0_4T_{31} & {}^0_4T_{32} & {}^0_4T_{33} & {}^0_4T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \overrightarrow{{}^0_4N_{K_0}} = \begin{bmatrix} {}^0_4T_{13} \\ {}^0_4T_{23} \\ {}^0_4T_{33} \end{bmatrix} \quad (28) $$

So we have,

$$ \theta_5 = \pi - Acos(\overrightarrow{{}^0_4N_{K_0}} . \overrightarrow{{}^0_6N_{K_0}}) \quad (29) $$

E. Joint 4 and 6

To obtain θ_4 and θ_6, rotation matrix ${}^4_6$R is used. On the one hand, ${}^4_6$R is:

$$ {}^4_6R = {}^0_4R^{-1}{}^0_6R = {}^4_0R\,{}^0_6R \quad (30) $$

and on the other hand,

$$ {}^4_6R = Rot_z(\theta_4)Rot_y(\theta_5+\pi)Rot_z(\theta_6) \quad (31) $$

in which,

$$ Rot_y\,(\theta_5+\pi) = \begin{bmatrix} \cos(\pi+\theta_5) & 0 & \sin(\pi+\theta_5) \\ 0 & 1 & 0 \\ -\sin(\pi+\theta_5) & 0 & \cos(\pi+\theta_5) \end{bmatrix} \quad (32) $$

$$ Rot_z(\theta_6) = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33) $$

thus,

$$ {}^4_6R = \begin{bmatrix} -c_4c_5c_6 - s_4s_6 & c_4c_5s_6 - s_4c_6 & -c_4s_5 \\ -s_4c_5c_6 + c_4s_6 & s_4c_5s_6 + c_4c_6 & -s_4s_5 \\ s_5c_6 & -s_5s_6 & -c_5 \end{bmatrix} \quad (34) $$

in which c_4 is corresponding to $\cos(\theta_4)$, s_4 is $\sin(\theta_4)$ and so forth.

For the sake of simplicity, let:

$$ {}^4_6R = \begin{bmatrix} {}^4_6R_{11} & {}^4_6R_{12} & {}^4_6R_{13} \\ {}^4_6R_{21} & {}^4_6R_{22} & {}^4_6R_{23} \\ {}^4_6R_{31} & {}^4_6R_{32} & {}^4_6R_{33} \end{bmatrix} \quad (35) $$

So, we have,

$$ \theta_4 = Atan2(-{}^4_6R_{23}, -{}^4_6R_{13}) \quad (36) $$

$$ \theta_6 = Atan2(-{}^4_6R_{32}, {}^4_6R_{31} \quad (37) $$

5 Experimental Results

In order for testing the forward and inverse kinematics solutions, a simulation program has been developed under the Unified System for Automation and Robot Simulation (USARSim).

A feedback testing approach by the following steps has been selected to estimate the accuracy of the proposed forward and inverse kinematics solutions:

(1) Moving the end effector of the robot to a specific location and orientation.
(2) Calculating the joint angles by the inverse kinematics solution.
(3) Changing the joint angles to the calculated values.
(4) Getting the gripper position and orientation by using the forward kinematics solution.
(5) Finally, computing the Euclidean distance between the initial position/orientation and the final position/orientation to get the error of the two solutions.
(6) Running the above steps 50 times with random generated initial position and orientation.

Figure 6 shows the calculated errors during 50 runs of the program. As it can be seen, the accuracy of these solutions is reasonably fair. The 50 runs only manifest three drastic errors in the position and orientation calculation method.

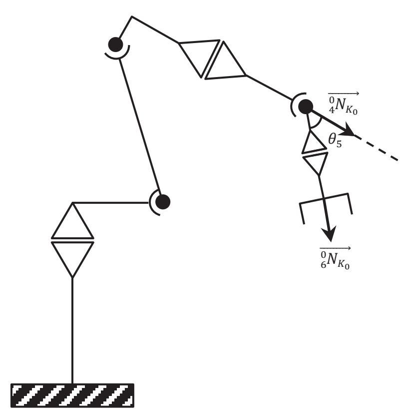

Figure 5. Joint 5 angle calculation geometrical visualization

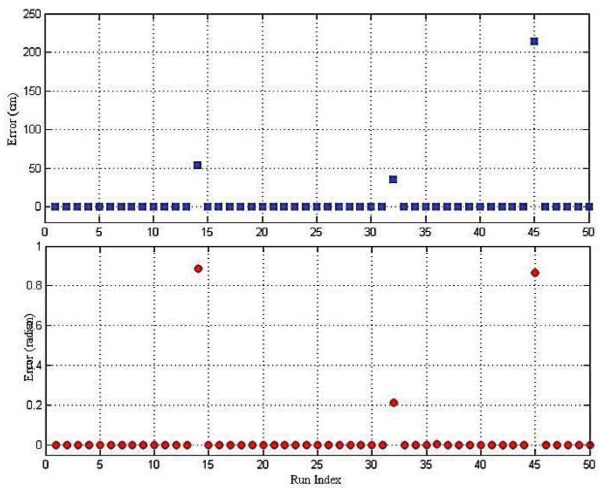

Figure 6. Position and orientation calculation error in 50 random runs in a simulated area using feedback loop testing

6 Conclusion

To conclude, this paper proposed a mathematical method and a geometrical approach for solving the forward and the inverse kinematics problems of an industrial arm robot, KUKA KR60. The experimental result obtained by feedback testing showed these solutions are less erroneous and more accurate. In the simulated programming application which with this method has been tested, all the steps have been implemented and therefore the result is based on the accuracy of the models in the simulation environment. Even the results are based upon simulation; one can conclude that the measurement has enough accuracy for practical usage.

7 Acknowledgment

I would like to thank Dr. Behrouz Minaei who was my advisor in my bachelor’s degree at the Iran University of Science and Technology and Dr. Nasser Mozayani whose unforgettable effort was an absolute motivation and support during my bachelor’s study. I also want to thank Dr. Hamid R. Arabnia and Dr. Walter D. Potter, faculty members of The University of Georgia, and Mr. Jeffrey Thompson who helped me to write this paper.

8 References

[1] J. J. Craig, Intrduction to Robotics: Mechanics and Manipulations, 3rd ed., 2004.

[2] A. Khatamian, "Virtual Manufacturing Automation by 6-DOF Manipulator in USARSim Simulation Area under Unreal Engine 3 (BSc. Thesis in Persian)," Iran University of Science and Technology, Tehran, 2011.

[3] V. H. John Lloyd, "Kinematics of common industrial robots," vol. 4, no. 2, pp. 169-191, June 1988.

[4] R. L. Ahmad Hemami, "Kinematic equations and solutions of a human-arm-like robot manipulator," vol. 4, no. 1, pp. 65-72, March 1988.

[5] T. K. L. Kesheng Wang, "Structure design and kinematics of a robot manipulator," vol. 6, no. 4, pp. 299-309, October 1988.

[6] A. P. M. L.-W. Tsai, "Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods," pp. 189-200, 01 Jun 1985.

[7] C. R. T. A. E. S. Stuart R. Lucas, "On the Numerical Solution of the Inverse Kinematic Problem," 01 Dec 200.

[8] R. F. a. J. J. S. Reinhart, "Neural Learning and Dynamical Selection of Redundant Solutions for Inverse Kinematic Control," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference*, 2011.

[9] J. a. N. I. B. Zhao, "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures," *ACM Transaction on Graphics (TOG),* vol. 13, no. 4, pp. 313-336, 1994.

[10] R. Köker, "A Genetic Algorithm Approach to a Neural-Network-Based Inverse Kinematics Solution of Robotic Manipulators Based on Error Minimization," *Information Science,* vol. 222, pp. 528-543, 2013.

[11] J. L. Andreas Aristidou, "FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem," *Graphical Models,* vol. 73, no. 5, pp. 243-260, 2011.

[12] S. V. A. a. B. C. N. Manjaree, "Inverse Kinematics Using Neuro-Fuzzy Intelligent Technique for Robotic Manipulator," *International Journal of Computers, Communications and Control–Communicated,* 2013.

[13] J. H. H. G. Zhelong Wang, "Forward Kinematics Analysis of a Six-Degree-of-Freedom Stewart Platform Based on Independent Component Analysis and Nelder–Mead Algorithm," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions,* vol. 41, no. 3, pp. 589-597, 2011.

[14] Y. L. a. X. Z. Jingguo Wang, "Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm," *International Journal of Advanced Robotics Systems,* 2010.