Analysis of Numerical Method on Inverse Kinematics of Robotic Arm Welding with Artificial Intelligence

Syed Yaseen\(^1\)*, Jaya Prakash\(^2\)
Department of Mechanical Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India
Email: \(^*\)yaseeny1998@gmail.com

Abstract. The key component of versatile production processes is robots. They are found in diverse systems where it is possible to bypass and optimize human function. I simulated a remote-control trickster in MATLAB with 6 degrees of freedom in this research. A robot is used in numerous applications, such as decoration, carpentry and equipment testing. In hardware verification labs, robotic arms are used to hold passive and power rail probes that connect from instruments like scopes and power supplies to PCB boards to protect the PCB layout from rip off due to sudden movement of the probes. Robot kinematics uses the geometry (position and orientation) of rigid bodies (links) and joints to control the movement of the robot. In this project, to control its movement, I have illustrated either forward or reverse enzyme kinetics of a device. Through Kinematics uses the positions of a joint to measure the end-effector position of the robot. The angles of the joints with the end-effector position are determined by inverse kinematics as the reference. There are several methods to calculate the forward and inverse kinematics such as analytical methods, numerical hit and trial, and iterative methods.

Keywords: Numerical Method, Welding Robot, Robot Kinematics, Artificial Intelligence, Robot Manipulator.

1. Introduction
In this chapter, a detailed discussion of the robot's arm, the basic components of the robotic arm (joints and links) and the degree(s) of freedom is provided.

Robotic arms are probably the most mathematically complex robots to analyze, simulate, and build. A robotic arm has links/frames and joints as the basic building blocks. The links are also known as rigid bodies which are chained together by revolute or prismatic joints to construct an open or closed chain robotic manipulator [4].

Robot anatomy discusses the various movements and relations and other elements of the construction process of the manipulator. A robotic joint gives relative movement across two robot contacts. A certain level of free movement (DOF) is given in any joint or axis. In most cases, each joint is connected with just one degree of independence. The sophistication of the robot can then be classified by the cumulative number of individual liberty degrees that it possesses [4]. Figure 1 displays frames / links and joints graphical depiction and placing [2].

Each joint is linked to three interfaces, an entry link and an output connexion, as seen in Figure 1. The joint allows for regulated rotational motion between the source and the destination attachment. The rigid part of the robot handling machine [1] is a robotic link. Many robots, such as the board, are mounted on a fixed platform.
1.1. Joint-link numbering convention
The robotic base is called relation 0, and its attachment to the first joint. In this series the first joint is joint 1. Connection 0 is a junction 1 input path and the 1 output connexion, leading to a junction 2. Link 1 is also the output link in combination with joint 1 and the reference link in joint 2 at the same time. For all joints and ties in the robotic system, this connect-numbering scheme is adopted [3].

1.2. Degree of freedom
The degree of freedom, or DOF, is an important concept in robotics. In general, each degree of freedom is a joint, which can be controlled along with the other joints to have a particular arm position. It is important that a robotic arm has a fewer degrees of freedom, because each degree of freedom adds more complexity to the control algorithm. Also, each degree of freedom requires a motor and an encoder [17].

1.3. Robot workspace
The environment of the robot is all locations, also recognized as the accessible room, which are accessible by the end effector (gripper). The working space depending on the angles / transcription limits of the limbs, the distance of the arm connexion, the angle to catch something anyway. It depends heavily on the setup of the robot [5].

2. Six DOF Robot manipulator
In this experiment a six DOF robotic arm consisting of six connexions is built using the Simulation Platform MATLAB, which further carries out forward and reverse film analyses.

The six DOF manipulators will have six joints from joint 1 to joint 6 and six from relation 0 to link 5, the joint link conventions has already been addressed. Figure 2 shows the graphical representation of a six DOF manipulator [18].

The organization of links and joints of the six DOF manipulator is shown in the above figure. Even the direction of the final of the mechanical system is seen above. In this project, the base (first link) is fixed [6]. The aim is to find a connection in the first and last link (gripper link). The connection can be achieved by defining the interactions relate between both the communication frames connected to the
3. Application of Denavit-Hartenberg Convention

In the collection of orientations for application domains, the Denavit Hartenberg Conventions (known as the DH Standard) is used [6].

Synchronise frameworks are linked in the tradition to the links between two positions, so that \( Z \) is bound to the joint and the second to the branch \( X \). The coordination transforms of the n-connected serial robot shape the cinematic formulas of the robot [7]:

\[
[T] = [Z_1][X_1][Z_2][X_2]...[X_n][Z_n],
\]

where \([T]\) is the transformation locating the end-link.

Each transformation from link \( i-1 \) to link \( i \) has six parameters. There are three parameters for rotation and three parameters for displacement. As I developed a six DOF robot using MATLAB robotic tool, it becomes complex to calculate the transformation matrix with three parameters each for rotation and displacement. In general, DH convention converts six parameters to four parameters (three parameters for rotation and one for displacement) for considering the transformation links from \( i-1 \) to \( i \).

The parameters of the DH convention are:
- \( A_i \) denotes the link length of link \( i' \)
- \( \alpha_i \) denotes the link twist of link \( i' \)
- \( \theta_i \) denotes the joint angle of joint \( i' \), revolute variable.
- \( d_i \) denotes the link offset, prismatic variable.

The link length and link twist are fixed for a given link. For a prismatic robot, \( d_i \) is the varying parameter and \( \theta_i \) is fixed. For a revolute robot, \( \theta_i \) is the varying parameter and \( d_i \) is fixed [8].

The DH convention is used in the project and the six DOF robots follows the DH rules mentioned for assigning the frames [9]:

**Rule 1:** \( Z_{i-1} \) is the axis of actuation of joint \( i \).
- Axis of transition of a democratic uprising
- Axis of transition of a prismatic joint

**Rule 2:** Axis \( X_i \) is perpendicular to \( Z_{i-1} \).

**Rule 3:** Axis \( Y_i \) is derived from \( X_i \) and \( Z_i \).

The DH parameters for the six DOF robots are shown in Table 1 [18]:

The table shows the joint angle configuration for the six DOF robot used in this project. The joint offset, link length and twist angle for each link are shown in the table. The angle limit for each link is given. A joint angle which is outside the specified limits is considered to be invalid.

| Joint Angle | Joint Offset (di) | Link Length (ai) | Twist Angle (\( \alpha_i \)) | Angle limits |
|-------------|------------------|------------------|-----------------------------|--------------|
| \( \theta_1 + \pi/4 \) | 0 | 0 | -\( \pi/6 \) | -185 to 185 |
| \( \theta_2 + \pi/3 \) | 0.5 | 0.699 | 0 | -155 to 35 |
| \( \theta_3 \) | 0.0948 | 0.0948 | \( \pi/2 \) | -130 to 154 |
| \( \theta_4 + \pi/3 \) | 0.68 | 0 | -\( \pi/2 \) | -350 to 350 |
| \( \theta_5 \) | 0 | 0 | \( \pi/2 \) | -130 to 130 |
| \( \theta_6 \) | 0.853 | 0 | 0 | -350 to 350 |

4. Application of forward kinematics and inverse kinematics

The method of evaluating the position and position of the end effector in Cartesian coordinates with the aid of the coordinate system is forward kinematics [10]. With the aid of the location of the output shaft, dynamic model is the method of measuring the joint angles. The graphical relationship among forward and reverse enzyme kinetics is seen in Figure 3.
Figure 3: Relationship between forward and inverse kinematics

Forward kinematics gives only one solution to the given input angles, shown in Figure 4 [11]. On the other hand, inverse kinematics can give multiple solutions to a given input, shown in Figure 5. This implies that inverse kinematics is more suitable to real word applications [12].

\[
\begin{align*}
\mathbf{p}_e &= \mathbf{p}_z + \mathbf{a}_0 \mathbf{a}_1 \\
&= \begin{bmatrix} n_x & 0 & n_z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}
\end{align*}
\]
5. Matlab Robot Creation Flowchart

Throughout this section we explore in depth how to introduce, build and evaluate the robotic arm only for future and the reverse kinematics using the MATLAB platform. [13] The flow chart in figure 6, illustrates how the robot is first built. Controlling the robot with input joint angles, forward kinematics and inverse kinematics functions used for the implementation.

![Flow chart](image.png)

Figure 6: Flow chart

In this project, I used MATLAB R2016B for robot creation and simulation. In the MATLAB tool, RVC feature which consists of the robotic 3D capability is initialized. With the RVC feature, the robot arm can be developed, controlled and manipulated [16]. I have a file, startup_rvm.m which calls several robot functions by calling the respective .m files of each function.

A matlab file, LINK.m performs the process of link creation. [14] This file contains all robot connection information, like simulated values, rigid body electromagnetic variables, engine specifications, etc. In addition, there are classes and functions to receive the four parameters of the DH convention and construct the link. The fifth parameter of a link determines its type, revolute (0) or prismatic (1). The robot is created by connecting all links together in a serial fashion. The MATLAB file, Serial Link. m does connect the vector of link objects and forms a serial-link robot object. The output of Serial Link. m gives all the links information and robot dimensions. Figure 7 shows the output of Serial Link. m (Serial-Link robot). [15] The DH parameters of each link and also shown in Figure7.
Figure 7: Links of six DOF robot by serial link

To view the 3D-representation of the robot created, Plot.m function is executed on the robot links to display the graphical animation based on the kinematic model. In the file, the plot function connects the roots of the connection points and reveals its 3D view. The movement of the robot can be controlled by adjusting the joints.

The file, fkine.m has a function which outputs the robot-end effector position as a homogenous transformation for the joint configuration. The result is provided in a matrix format as described in the forward kinematic chapter [17]. The fkine.m file uses the transformation matrix method to provide the result. Similarly, the file ikine.m outputs the joint co-ordinates corresponding to the robot end effector position [18].

6. Results

Example 1: Six link revolute robot

Example 1 gives the parameters of the links that constitute the six DOF robots. It also gives the configuration of the links (Prismatic or Revolute). In this project, I developed a robot with six links in revolute configuration. The DH parameters of the six DOF robots are obtained from the result as represented in Figure 8.

Example 2.1: Robot position with joint angles (11, 16, 21, 29, 41, 51)
Example 2.1 shows the robot with joint angles (11, 16, 21, 29, 41, 51). This sequence of coordinate system positioned the joints in various directions to achieve the output values effector seen in Figure 8. The same reflects the theme can also be achieved using inverse kinematics via different joint angles.

Example 2.2: Moving kinematics of positions of joints (11, 16, 21, 29, 41, 51)

Example 2.2 shows that the obtained in 4x4 matrix form from which the values of the position vectors are obtained. In addition, the position vectors of the end-effector are all positive. The second orientation vector along the three axes is negative. An illustration is shown in Figure 9.
Example 3.1: Robot position with joint angles (9, 18, 19, 17, 27, 36)

Figure 10: Robot position with joint angles (9, 18, 19, 17, 27, 36)
Example 3.1 shows the robot with joint angles (9, 18, 19, 17, 27, 36). In this example, the end-effector position is pointing downwards. Link 5 is perpendicular to link 6 and also, link 6 is parallel to link 2.

Link 5 is perpendicular to link 2. Also, link 2, link 5 and link 6 forms a rectangular shape with one side open. The configuration of the robot is shown in Figure 10.

Example 3.2: Forward kinematics with joint angles (9, 18, 19, 17, 27, 36)

Figure 11: Forward kinematics with joint angles (9, 18, 19, 17, 27, 36)
Example 3.2 shows the forward kinematics of the robot with joint angles (9, 18, 19, 17, 27, 36). The second orientation vector along the three axes is negative. The orientations of the end-effector along the z-axis are negative. The position vector along the three axes is positive as shown in Figure 11.

Example 4.1: Robot position with joint angles (13, 18, 25, 37, 44, 50)
**Example 4.1** shows the robot with joint angles (13, 18, 25, 37, 44, 50). In this example, joints 3 and 5 are perpendicular to each other. Link 5 and link 6 are serially attached to each other. Under these values of the joint variables, the end-effector position is directed towards the upper right corner as shown in Figure 12.

**Example 4.2**: Forward kinematics with joint angles (13, 18, 25, 37, 44, 50)

*Example 4.2* shows the forward kinematics of the robot with joint angles (13, 18, 25, 37, 44, 50). In this example, joints 3 and 5 are perpendicular to each other. The positive vectors along the z-axis oz, αz, pz are all positive except the position vector nz which has a negative value as shown in Figure 13.

*Example 5.1*: Robot position with joint angles (6, 11, 21, 30, 39, 46)
Example 5.1 shows the robot with joint angles (6, 11, 21, 30, 39, 46). In this example, the end-effector position is pointing downwards similar to example 3.1 with an angle variation in the end effector. This is mainly due to the angle of joint 6. Also, joint 3 is perpendicular to joint 5. These are illustrated in Figure 14.

Example 5.2: Forward kinematics with joint angles (6, 11, 21, 30, 39, 46)

Example 5.2 shows the forward kinematics of the joint angles (6, 11, 21, 30, 39, 46). In this example, the orientation and position vectors along the y-axis are negative and all the orientation vectors along z-axis are positive except $\alpha_z$. The position vector along x and z axes are positive whereas it is negative along the y-axis as shown in Figure 15.

Example 6.1: Robot position with joint angles (50, 48, 42, 31, 23, 15)
Example 6.1 shows the robot with joint angles (50, 48, 42, 31, 23, 15). In this example, the end-effector position is pointing towards the upper-left and the configuration of the joints is very different as shown in Figure 16. Joints 2, 3, 6 and joints 4, 6, 5 form two different triangles. Link 0 and link 5 are parallel to each other.

Example 6.2: Forward kinematics with joint angles (50, 48, 42, 31, 23, 15)

In example 6.2 the position vector along the x and y axis is negative whereas it is positive along the y-axis. The orientation vectors nx, ox, ax and the position vector px are negative. The orientation vector oxis negative along the three axes as shown in Figure 17.
Example 7.1: Robot position with joint angles (9, 16, 25, 35, 43, 51)

Figure 18: Robot position with joint angles (9, 16, 25, 35, 43, 51)

Example 7.1 shows the robot with joint angles (9, 16, 25, 35, 43, 51). In this example, the end-effector position is parallel to the x-axis. Also, joint 5 and joint 6 are perpendicular to each other. The link 5 and link 6 are perpendicular to each other respectively. This configuration is shown in Figure 18.

Example 7.2: Forward kinematics with joint angles (9, 16, 25, 35, 43, 51)

Figure 19: Forward kinematics with joint angles (9, 16, 25, 35, 43, 51)

Example 7.2 shows the forward kinematics of the joint angles (9, 16, 25, 35, 43, 51). The orientation vectors nx, ny and nz are negative. The diagonal vectors nx, oy and az of the orientation parameters of the end-effector are positive, negative, and negative respectively. Figure 19 shows this configuration.

Example 8.1: Robot position with joint angles (7, 14, 20, 23, 34, 49)
Example 8.1 shows the robot with joint angles (7, 14, 20, 23, 34, 49). In this example, link 2 and link 6 are parallel to each other. Also, the end effector position is pointing towards joint 2. Link 2 and link 6 are perpendicular to link 5 as shown in Figure 20.

Example 8.2: Forward kinematics with joint angles (7, 14, 20, 23, 34, 49)

Example 8.2 shows the result of forward kinematics for the joint angles (7, 14, 20, 23, 34, 49). The orientation vectors $n_x$, $n_y$, $n_z$ of the forward kinematics are positive which is opposite to the robot with the joint angle (9, 16, 25, 35, 43, 51). The orientation parameters along the z axis of the end-effector are positive. This configuration is shown in Figure 21.
7. Result Summary

Table 2: Robot link output

| LINK  | Joint angle | Joint offset (d_i) | Link length (a_i) | Twist angle (\(\alpha_i\)) | Prismatic or Revolute |
|-------|-------------|-------------------|------------------|-----------------------------|----------------------|
| Link 1 | Q1          | 0                 | 0                | -\(\pi/6\)                 | Revolute             |
| Link 2 | Q2          | 0.5               | 0.699            | 0                           | Revolute             |
| Link 3 | Q3          | 0.0948            | 0.0948           | \(\pi/2\)                   | Revolute             |
| Link 4 | Q4          | 0.68              | 0                | -\(\pi/2\)                 | Revolute             |
| Link 5 | Q5          | 0                 | 0                | \(\pi/2\)                   | Revolute             |
| Link 6 | Q6          | 0.853             | 0                | 0                           | Revolute             |

Table 2 shows the parameters of each link of the six-link robotic arm. The parameters are joint offset, link length, twist angle and joint angle. In addition, it also shows the configuration of each link (prismatic or revolute). All these parameters together form the DH parameters for the six link robotic arm used in this project.

Table 3: Input joint angle

| Joint angle | Input 1 | Input 2 | Input 3 | Input 4 | Input 5 | Input 6 | Input 7 |
|-------------|---------|---------|---------|---------|---------|---------|---------|
| \(\theta_1\) | 11      | 9       | 13      | 6       | 50      | 9       | 7       |
| \(\theta_2\) | 16      | 18      | 18      | 11      | 48      | 16      | 14      |
| \(\theta_3\) | 21      | 19      | 25      | 21      | 42      | 25      | 20      |
| \(\theta_4\) | 29      | 17      | 37      | 30      | 31      | 35      | 23      |
| \(\theta_5\) | 41      | 27      | 44      | 39      | 23      | 43      | 34      |
| \(\theta_6\) | 51      | 36      | 50      | 46      | 15      | 51      | 49      |

Table 3 shows the six joint angles for each input (example). The robot position for each input (example) is shown in the previous chapter. It is also proven that changing the angle of any joint would result in a different end-effector position.

Table 4: Forward kinematics result

| Position vector | Output 1 | Output 2 | Output 3 | Output 4 | Output 5 | Output 6 | Output 7 |
|-----------------|----------|----------|----------|----------|----------|----------|----------|
| Nx              | -0.6359  | 0.5071   | 0.6414   | 0.9878   | -0.7086  | -0.0487  | 0.4069   |
| Ny              | -0.6871  | -0.8599  | -0.5989  | -0.0348  | 0.6655   | -0.1410  | 0.2719   |
| Nz              | 0.3515   | -0.0581  | -0.4795  | 0.1521   | 0.2346   | -0.9888  | 0.8721   |
| Ox              | -0.4250  | -0.8597  | 0.5456   | -0.1251  | -0.6426  | 0.8466   | -0.7750  |
| Oy              | -0.0684  | -0.4999  | -0.0833  | -0.7594  | -0.4712  | 0.5195   | -0.4026  |
| OZ              | -0.9026  | -0.1052  | 0.8339   | 0.6385   | -0.6042  | -0.1158  | 0.4871   |
| Ax              | 0.6442   | 0.0614   | -0.5394  | 0.0932   | -0.2916  | 0.5300   | 0.4835   |
| Ay              | -0.7234  | 0.1033   | -0.7965  | -0.6497  | -0.5789  | -0.8428  | -0.8741  |
| Az              | -0.2485  | -0.9928  | 0.2733   | -0.7545  | 0.7615   | 0.0940   | 0.0469   |
| Px              | 0.1644   | 0.2368   | -0.2747  | 0.3078   | -0.0904  | 0.9605   | -0.2511  |
| Py              | 0.4157   | 0.7299   | -1.2023  | -1.4058  | -0.3611  | -1.7079  | 0.5865   |
| Pz              | 0.6945   | 0.2215   | 1.2389   | 0.4786   | 1.2384   | 0.3677   | -0.1047  |

Table 4 shows the forward kinematics result for the input combinations (with six different joint angles) mentioned in Table 3. Vectors N, O and the position of the finished of the robot reflects A. The location of the orientation of the end is defined by Vector P.

Table 5 shows the inverse kinematics result of different input combinations of the position vectors.
The joint angles were measured as output using the angle and location variables as data. It proves that forward and inverse kinematics is inverse function of each other.

| Joint angle | Input 1 | Input 2 | Input 3 | Input 4 | Input 5 | Input 6 | Input 7 |
|-------------|---------|---------|---------|---------|---------|---------|---------|
| $\theta_1$  | 11      | 9       | 13      | 6       | 50      | 9       | 7       |
| $\theta_2$  | 16      | 18      | 18      | 11      | 48      | 16      | 14      |
| $\theta_3$  | 21      | 19      | 25      | 21      | 42      | 25      | 20      |
| $\theta_4$  | 29      | 17      | 37      | 30      | 31      | 35      | 23      |
| $\theta_5$  | 41      | 27      | 44      | 39      | 23      | 43      | 34      |
| $\theta_6$  | 51      | 36      | 50      | 46      | 15      | 51      | 49      |

8. Conclusion

In this project, I have simulated a six DOF robot using MATLAB simulation tool. The robot's motion was regulated using different angular position configurations in various directions. To evaluate the end-effector orientation for fixed coordinate system, and the steering angle for a fixed end-effector, the principle of inverse kinematics has been used. Through research and implementation of kinematics concept in the project, it became clear that inverse kinematics is widely used in practical applications. The result of the forward kinematics was also discussed for each joint angle combination emphasizing the difference in the configuration of the position vector. The results confirmed that the inverse function of the inverse kinematics.

9. References

[1] Peter Corke, Resources for robotics education, February 2017 http://petercorke.com/wordpress/
[2] Vijayabaskar V. and Rajendran V. (2011), “Analysis and modeling of wind dependence of ambient noise in shallow water of Arabian sea”, European Journal of Scientific Research, Vol.50, pp.28-34, SJR IF: 0.21, Nov. 2010.
[3] Robot Kinematics, February 2017 http://www.wikipedia.com.
[4] NPTEL, February 2017 http://nptel.ac.in/courses/112101098/31.
[5] Crowder, R.M, Automation and Robotics, February 2017 http://www.springer.com/cda/content/document/cda_downloaddocument/9789400705784c2.pdf?SGWID=0-0-45-1338704-p174080776
[6] Er. Harpreet Singh, Dr. Naveen Dhillon and Er. Imran Ansari, “Forward and inverse Kinematics solution for six DOF,” IJAEM, ISSN 2319-4847, 2013.
[7] Kinematic and dynamic analysis of a robotic positioning arm, February 2017 http://bpl.ee.washington.edu/eprints/110/1/Th033.pdf
[8] Craig, J.J.(1989). Introduction to Robotics Mechanics and Control, USA: Addison Wesley Publishing Company.
[9] Forward kinematics: The Denavit- Hartenberg convention, March 2017 https://www.cs.duke.edu/bnd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf.
[10] Homogenous transformation matrices, March 2017 http://me.umn.edu/courses/me5286/manipulator/LectureNotes/2017/ME5286CourseNotes3-2017.pdf
[11] W. W. Melek. "ME 547: Robot Manipulators: Kinematics, Dynamics, and Control". Waterloo, ON, 2010, University of Waterloo.
[12] L-W. Tsai, "Robot Analysis: The Mechanics of Serial and Parallel Manipulators" NY, 1999, John Wiley & Sons, Inc.
[13] Zhang, Dan Lei, Jianhe Kinematic analysis of a novel 3-DOF actuation redundant parallel manipulator using artificial intelligence approach, Robotics and Computer- Integrated Manufacturing, 2011, P.157-163.
[14] B Purmus, H. Temurtas & A. Gan An inverse kinematics solution using practical swarm optimization 6th International Advanced Technologies Symposium (IATS '11), 16-18 May 2011, Elazığ, Turkey.
[15] B. Pattanaik and S. Chandrasekaran, "Recovery and reliability prediction in fault tolerant
automotive embedded system," 2012 International Conference on Emerging Trends in Electrical Engineering and Energy Management (ICETEEEEM), Chennai, 2012, pp. 257-262, doi: 10.1109/ICETEEEEM.2012.6494477.

[16] Mustafa JabbarHayawi analytical inverse kinematics algorithm Of A 5-DOF robot arm Journal of education of college, no.4 vol.1march./2011.

[17] Pattanaik B, Murugan S. “Cascaded H-Bridge Seven Level Inverter using Carrier Phase Shifted PWM with Reduced DC sources," International Journal of MC Square Scientific Research, 9(3):30-9, 2017.

[18] Stewart D. A Platform with Six Degrees of Freedom, Proc.Insn.Mech.Engrs.(Part I), 180(15), P.371-386,1965.