International Journal on Interactive Design and Manufacturing (IJIDeM)
https://doi.org/10.1007/s12008-023-01552-w

ORIGINAL PAPER

Six DOF robot: inverse kinematics solution to path planning for intersecting pipes for welding operation and inverse Jacobian comparison

Avantsa V. S. S. Somasundar[1] · G. Yedukondalu[1]

Received: 22 May 2023 / Accepted: 10 September 2023

Abstract

Robot's movement along a continuous path, point-to-point operations are programmed. In continuous path welding operation, the path is planned at the identified weld seam. This paper deals with inverse Kinematics and inverse Jacobian approaches for welding path planning of intersecting pipe models. The straight, circular and axi-symmetric pipe models were considered for welding operation. The intersecting curve equations based on the model have been produced which is the path of welding robot. The welding paths by Kuka KR5 Arc Robot were computed, simulated and analyzed in both MATLAB and RoboAnalyzer software. Both Inverse Kinematics and Inverse Jacobian methods were compared graphically and numerically. It was found that, the resulting weld path geometries were similar in both methods. However very less errors and very close to expected positions during the path tracing were observed in inverse Kinematics approach as compared to inverse Jacobian method where errors were higher due to its intrinsic linear interpolation nature in path tracing.

Keywords Inverse kinematics · Inverse Jacobian · Path planning · Pipe welding

1 Introduction

A serial robot contains of several links as well as joints. The robot joints are actuated by the motors, which ultimately cause the robot end effector to change position and orientation. In practical applications, the mechanism is governed by Cartesian control to produce the required rotation of joints which uses Jacobian. Inverse Kinematics and Jacobian Inverse produce joint co-ordinates by using End effector Position and orientation and derivations. These are the best methods to compare with Cartesian control approach. Also, singularity, joint motions are obtained.

Welding of intersecting pipes joint is taken in this paper (Figs. 1a and 2). The profile of intersection of two pipes is to be generated (Fig. 1b). A graphical model is required to visualize and to begin programming of path tracing by robot. An analytical record is also equally important to generate various points' co-ordinates along the path by software program. Manufacturing technologies for accurate preparation of pipes i.e. cutting and drilling, prior to welding are assumed to exist. Hence ideal models of the intersection path that are without any deviations are considered for path generation.

This present work will help industry work, also in the design phase of path planning for operations like welding, painting, assembly. Suitability of given Configuration of the Robot (using D-H parameters) for the required path profile can be assessed by simulation and singularity findings from this work. Iteratively a more suitable Robot Configuration may be suggested if given configuration is not suitable [1, 2]. Since it is simulation by MATLAB programs used as compared to physical robotic trials and testing which requires to purchase or rent a Physical Robot, the cost of this estimation and selection of more suitable Robot Configuration is cheaper and easy to use [3, 4]. Comparison of Inverse Jacobian and Inverse Kinematics approaches is not available much in the literature which is essential in applying path planning to real geometries. To estimate Robotic path tracing in design phase even without the availability of Physical robots, we need software tools. The current research is an effort to fill that gap. One of the limitations are multiple solutions of Inverse

✉ Avantsa V. S. S. Somasundar
somasundar.av@gmail.com

G. Yedukondalu
yedukondalu@kluniversity.in

1 Department of Mechanical Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh 522502, India

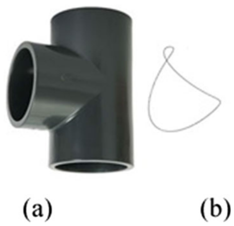


Fig. 1 **a** Pipe T-joint, pipes intersect at 90 degrees angle to one another. **b** Equal radii and 90 degrees intersection are the conditions under which the pipes' intersection path is shown in the example above

Fig. 2 Industrial application example of intersecting pipes in chemical engineering and process Industry. (*Courtesy*: gillins.com-Project in slough)

Kinematics approach. This is resolved inherently in the Robotic Toolbox Plugin used with MATLAB. Also, real time adaptability may be limited due to thermal expansion of joint due to Welding, and joint angles for the path are calculated based on Room temperature dimension of pipes and intersecting pipe joint [5, 6].

Taken as the intersecting curve is the curve which forms on the surface of intersecting pipes that cross. We discuss in this paper, the two types of programmable approaches to achieve motion along this 3-dimensional curved path by robot end effector. These are inverse kinematics and inverse Jacobian. Algebraic solution using Maple was proposed to trace the intersecting pipes joint for different physical parameters [7]. Deviation between pipelines and their ideal models were assumed and quantified to adjust for path deviations of intersecting curve [8]. Different methods orientation [9]. A Robot system was developed by fixing to branch pipe and perform welding along intersection joint [10]. Deep Deterministic Policy Gradient (DDPG) is applied to get joint trajectories [11]. As control inputs are computed from the generated trajectory in each update, they are applicable to achieve closed-loop control similar to model predictive controller [12]. Inverse Kinematics Algorithms using analytical methods and control-based methods were presented [13]. Singular value decomposition-based analysis of Jacobian transpose method, the pseudoinverse method, and the damped least squares methods for inverse kinematics (IK) were detailed [14]. A versatile, scalable, general-purpose robot simulation framework called V-REP is introduced [15]. Some of the applications covered by V-rep are rapid prototyping, safety/remote monitoring, training and education, hardware control, and factory automation simulation. Inverse kinematics learning is presented for resolved motion rate control (RMRC) in the case when optimization criterion to resolve kinematic redundancies is applied [16]. Robot operating System (ROS) and its relation to robot software frameworks and ROS architecture are outlined [17]. FK of industrial robots with revolute joints, three-dimensional simulation and plots of RoboAnalyzer tool were discussed [18]. Methodology of Singularity study of Kuka Industrial robot was outlined for preliminary paths [19]. Path profiles of various joints were analyzed by Jacobian Inverse by [20].

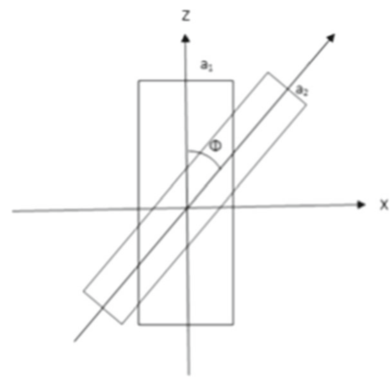


Fig. 3 Intersecting pipes model, general Scenario

2 Methodology

2.1 Path geometry

Figure 3 shows a model of cylindrical pipes which intersect at an angle of Φ.

Here.

- Φ is pipes axes intersection angle
- θ_1 and θ_2 are parametric angles (0 to 360 degrees)
- a_1,a_2 are radii of pipes
- z_1 and z_2 are distances along each pipe's respective axes

Parametric equations of 1st cylinder

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_1 \mathrm{Cos}\theta_1 \\ a_1 \mathrm{Sin}\theta_1 \\ z_1 \end{bmatrix} \tag{1}$$

For 2nd cylinder

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{pmatrix} \mathrm{Cos}\phi & 0 & -\mathrm{Sin}\phi \\ 0 & 1 & 0 \\ \mathrm{Sin}\phi & 0 & \mathrm{Cos}\phi \end{pmatrix} \begin{bmatrix} a_2\mathrm{Cos}\theta_2 \\ a_2\mathrm{Sin}\theta_2 \\ z_2 \end{bmatrix}
= \begin{bmatrix} a_2\mathrm{Cos}\theta_2\mathrm{Cos}\phi - z_2\mathrm{Sin}\phi \\ a_2\mathrm{Sin}\theta_2 \\ a_2\mathrm{Cos}\theta_2\mathrm{Sin}\phi + z_2\mathrm{Cos}\phi \end{bmatrix} \quad (2)
$$

equating x, y, z components of each pipe surface yields the intersecting curves.

Here a_1, a_2, φ, θ_2 are known variables and θ_1, z_1 and z_2 are unknown variables.

$$
\begin{aligned}
z &= a_2\mathrm{Cos}\theta_2 + \frac{\mathrm{Cos}\phi\sqrt{a_1^2 - a_2^2 + a_2^2\mathrm{Cos}\theta_2^2}}{\mathrm{Sin}\phi} \\
y &= a_2\mathrm{Sin}\theta_2 \\
x &= -\sqrt{a_1^2 - a_2^2 + a_2^2\mathrm{Cos}\theta_2^2}
\end{aligned} \quad (3)
$$

2.2 Inverse kinematics

The array of successive homogeneous transformation matrices is found out and input to the MATLAB based Robotic ToolBox plug-in. Each of these matrices correspond to successive poses along the required path. to. As a result set of joint angles at each pose is obtained to plot the path in simulation.The many alternative configurations (joint angles) for a given EE pose may be calculated theoretically or more easily by RoboAnalyzer like softwares and seen which alternative configuration is chosen by plug-in to trace the path. This analysis will help to see how singularity is avoided and how minimum joints rotation is achieved over the entire path.

2.3 Inverse Jacobian

The Jacobian matrix at a given pose (joint angles) of the robot is obtained. Inverse Jacobian is found from Jacobian matrix. Then change in joint angles (though very small) is found from product of inverse Jacobian and required incremental change in EE pose (x,y,z, roll, pitch, yaw angles). The above steps are iterated in a loop till the end point of the required path is reached. As it can be inferred from above procedure, Inverse Jacobian is a linear approximation method. Care is to be taken that the path does not cross any singular points. If EE has to pass through singularity, at that singular pose the determinant of Jacobian matrix will be zero, so the path planner will be aware of the singular points on the path and have to redesign the path or change co-ordinates of starting and ending of path.

Any offline path planning algorithm and its results that usually based on Cartesian approach for path planning can be compared with these two fundamental approaches for optimum Joint angles, joint trajectories and hence power consumption, cost of operation may be standardized.

2.4 Software used

Matlab R2015a is used with Robotic Toolbox plug-in 9.10 in the current work. In the current work matrix multiplication, arrays of 1-D,2-D matrices are widely used. Robotic tool box is a Plug-in to Matlab and supports Robotic research. Robotic Toolbox plug-in's following built-in API is used in the current work. Create virtual robot based on D-H Parameters, find out

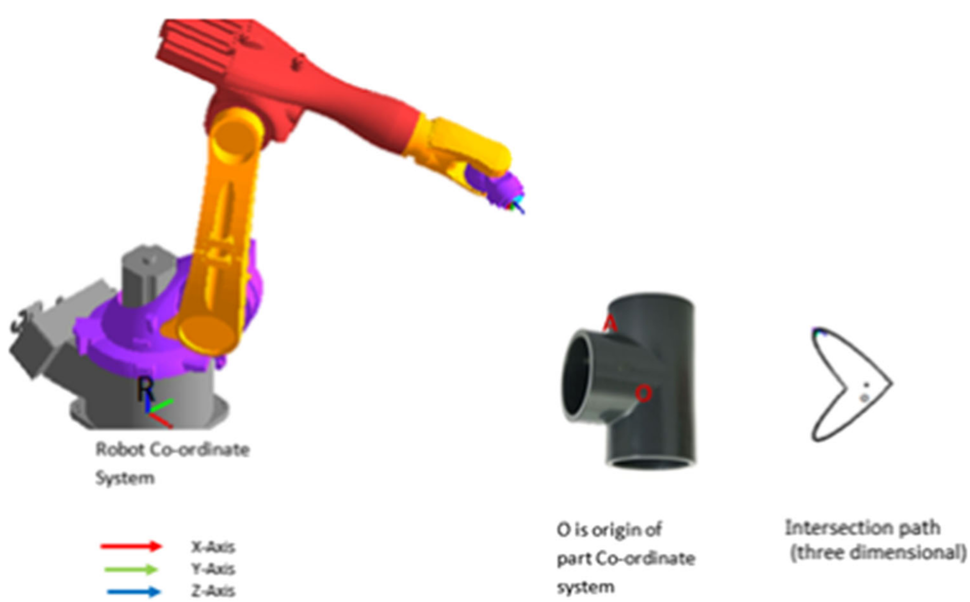
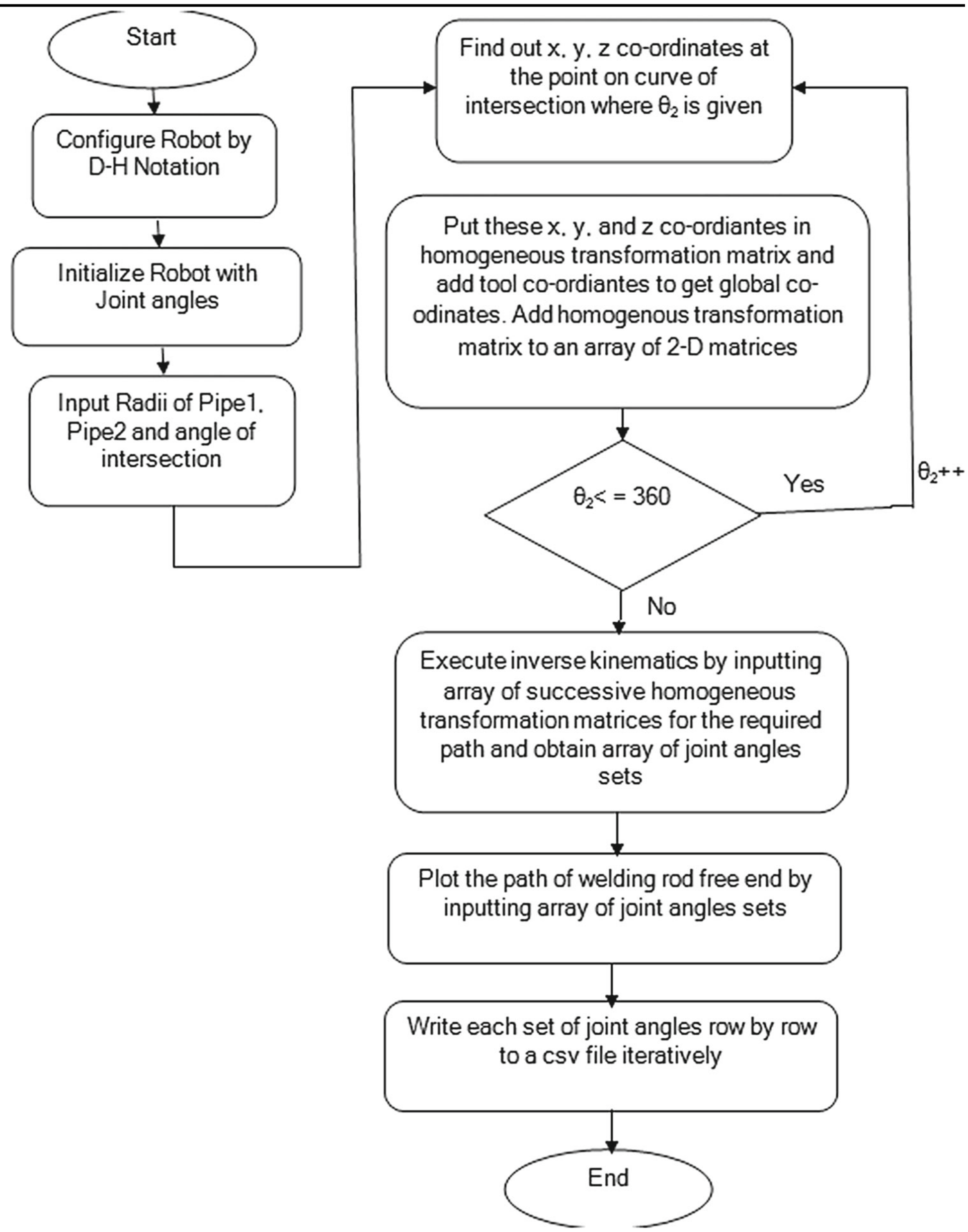


Fig. 4 Robot, Model of Intersecting pipes, Part Co-ordinate Systems and Weld curve

Fig. 5 Inverse Kinematics



Forward Kinematics, Inverse Kinematics results, Jacobian matrix for a given pose, Check singularity of any Pose and Animation/Simulation of path tracing.

RoboAnalyzer software is used to visualize robotic operations. RoboAnalyzer features are FKine for path tracing, EE Config for EE pose at intermediate positions of path, Joint trajectory to read Joint trajectory.CSV file are used in the present work. RoboAnalyzer's Ikine feature is used to cross validate Matlab results of Inverse Kinematics for some EE poses.

3 Path planning

The Work positioning and Robot initial pose are derived and same as [21].

Model of intersecting pipes in 3-D (Fig. 4), was used as reference to produce simulation of continuous weld path by 6-DOF KUKA KR5 Arc Robot model. 3D Cartesian co-ordinate (x,y,z) of 360 intermediate points for the path were evaluated. These co-ordinates of via points were given to the Inverse Kinematics function call. There by joint angles of joints of robot were returned. Then FK function was invoked and path was produced in graphics window as shown in Fig. 6. These angles were written to a CSV file. Same path of the model was also produced in RoboAnalyzer software with.CSV file input (Fig. 6).

3.1 Path generation flowchart

See Fig. 5.

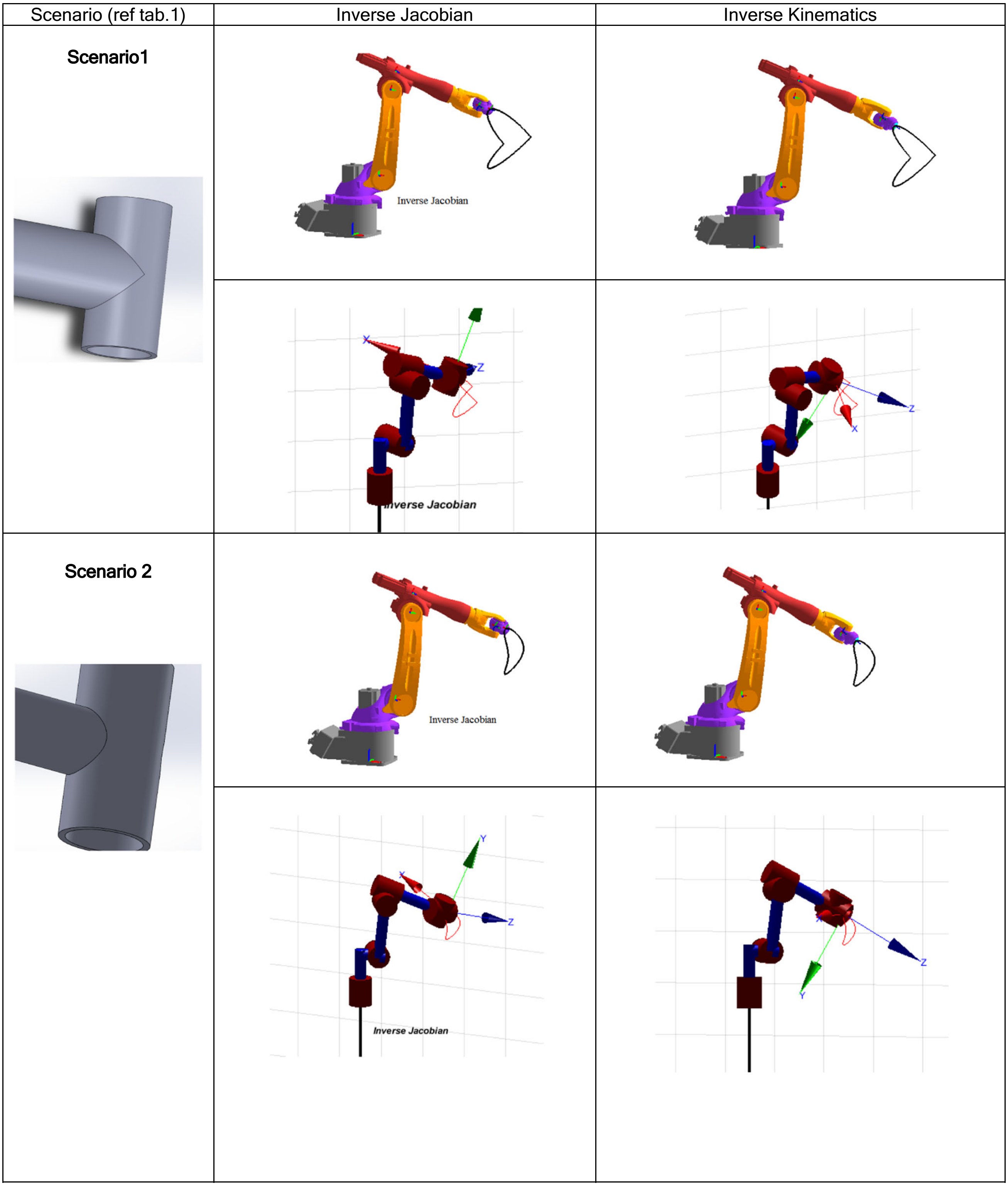


Fig. 6 Path tracing results in Various Welding Scenarios, Resulting paths traced in All four welding Scenarios are given. Both Inverse Kinematics and Inverse Jacobian approach based paths are given side by side for comparison. Paths are traced in Roboanalyzer software and MATLAB also. Refer Table 2 For Welding Scenario details

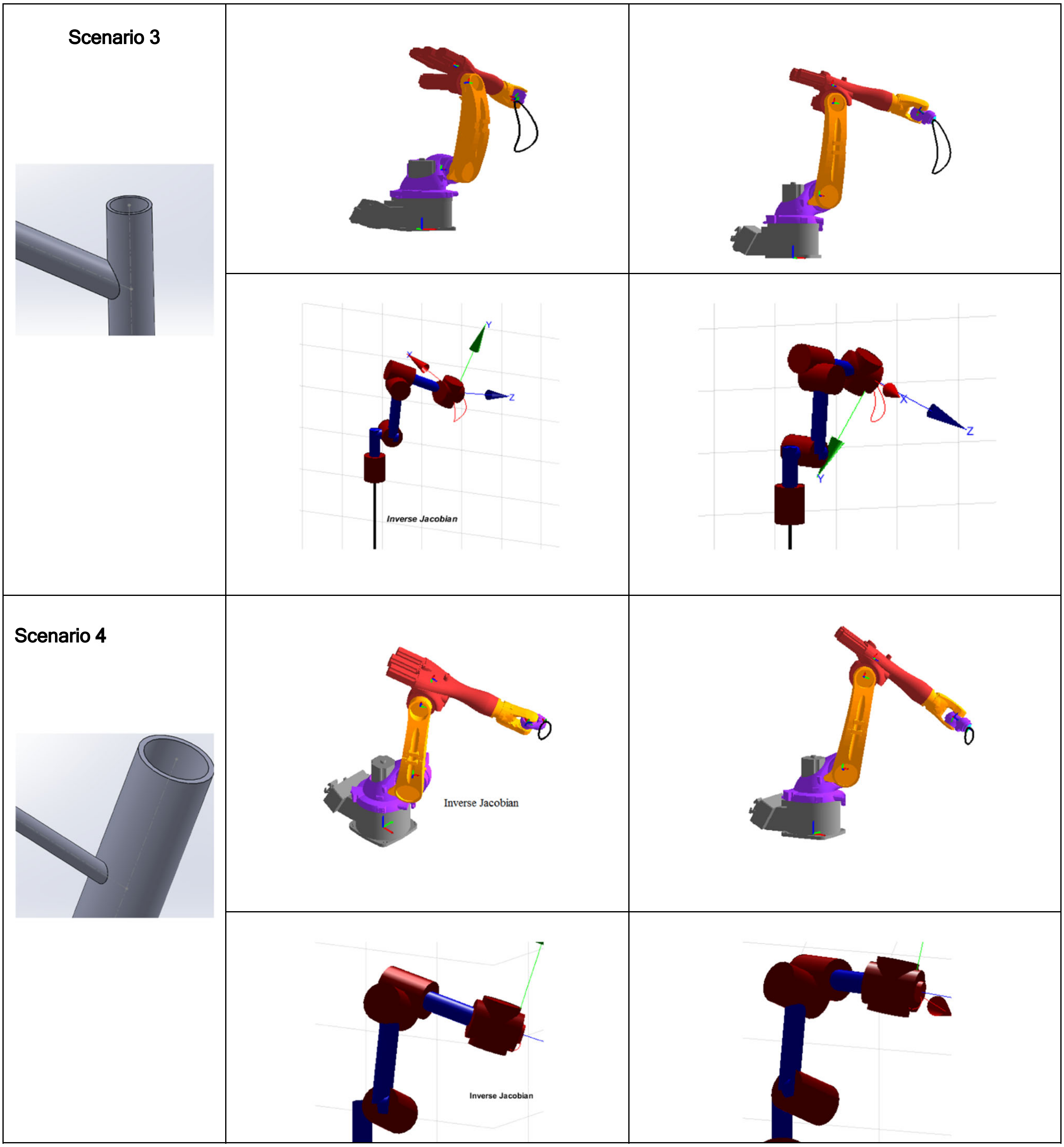


Fig. 6 continued

4 Results and discussions

Results for various scenarios in RoboAnalyzer and MATLAB by Inverse Jacobian, Inverse kinematics (Figs. 6, 7, 8, 9, 10, 11, 12).

5 Comparison

MATLAB program is run separately for different no. of Steps as shown in Table 2. Each time, the output CSV file is supplied to Robo Analyzer. Robo analyzer traces the path till the End point. This point from Robo Analyzer is the actual position obtained from Inverse Kinematics algorithm. Scenario1

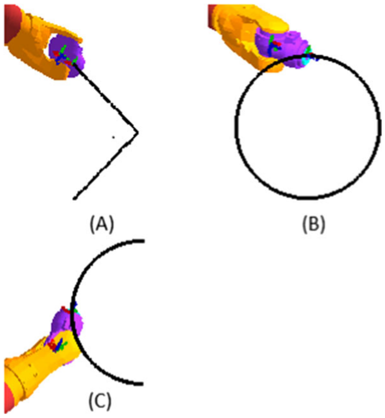


Fig. 7 Path produced (Scenario 1)

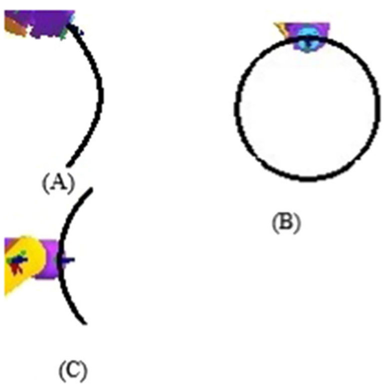


Fig. 8 Path produced (Scenario 2)

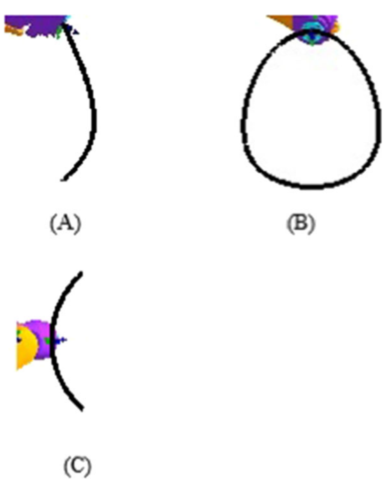


Fig. 9 Path produced (Scenario 3)

(Ref. Table 1) is chosen for comparing Inverse Jacobian and Inverse Kinematics methods by simultaneously changing number of steps in path. The 180,360,540 segments that make up the joint curve of the pipes were separated into IK, IJ programmes in MATLAB, which yielded 181,361,541 sets

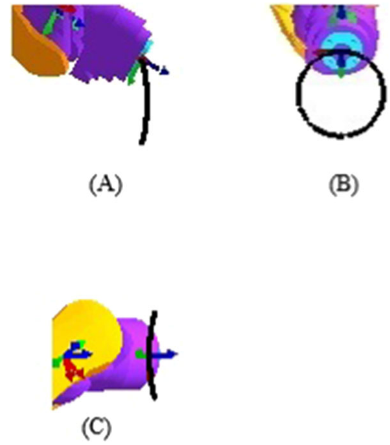


Fig. 10 Path produced (Scenario 4)

Table 1 Welding scenarios, work is performed on all these scenarios by Inverse Kinematics and Inverse Jacobian approached and followed by comparision

Scenario	Radius of pipe 1 (a_1) in cm	Radius of pipe 2 (a_2) in cm	Angle of intersection in degrees
1	20	20	90
2	20	15	90
3	20	15	60
4	20	5	90

of 6 joints. End Effector X, Y, Z locations at 6 even steps are taken while simulating path tracing in RoboAnalyzer. Real Position and Intended Position were compared using the difference between the two (positional error). Plotted are the positioning errors for each of the 6 evenly spaced steps for 180,360,540 steps. Inverse kinematics approach showed very small positional error and with increase in number of steps the error got reduced. Error values were higher using the inverse Jacobian technique. IJ is a linear approx. solution, and the current study involves a 3D curve.

As shown in the Table 2, the entire path is divided into 180, 360 and 540 steps successively and in each of them, the positional error is recorded after every 1/6th portion of the path is traced and tabulated for comparison. Positional error is the distance between expected and actual points after each step is passed.

the positional errors are high. Also, the positional errors changed from high to low when the path is changing direction from positive to negative co-ordinates and vice versa. Since the path is cyclic, the error change also revealed to be of cyclic in nature.

For Scenario 1, graphs of Joint angles for path execution using the IK technique are provided. (Figs. 13 and 14

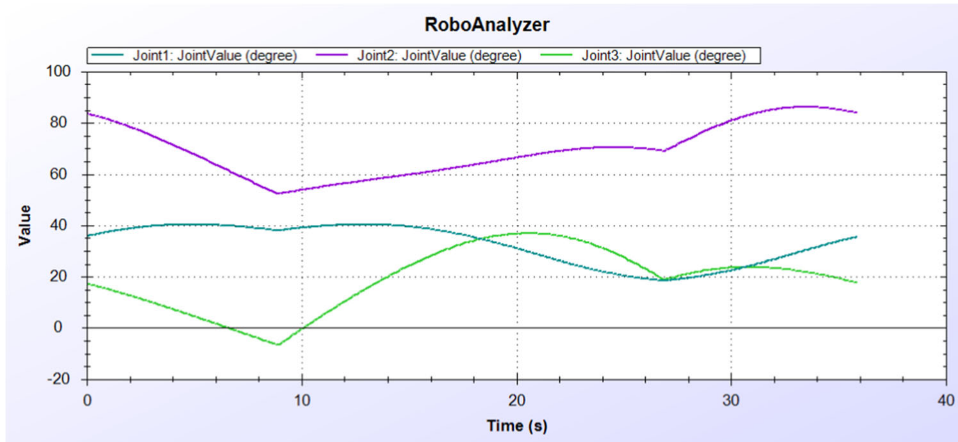


Fig. 11 Joint 1, 2, 3's angles produced in IK (produced in RoboAnalyzer)

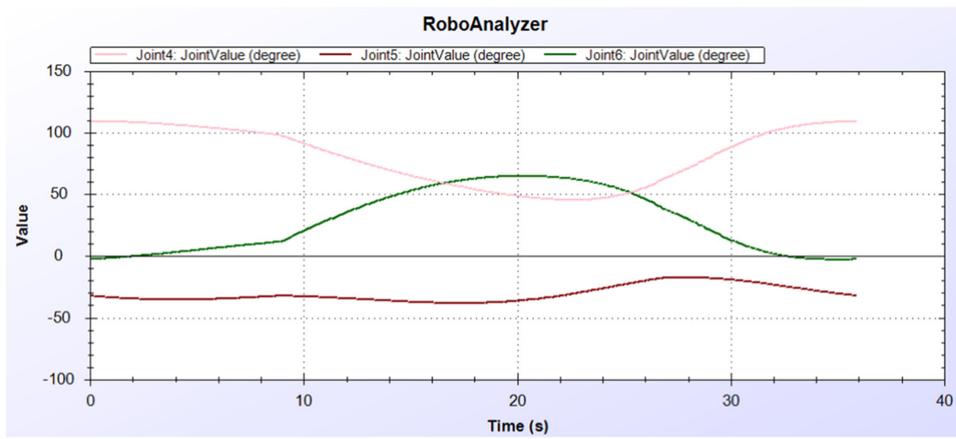


Fig. 12 4, 5, 6 Joint angles produced in IK (produced in RoboAnalyzer)

Table 2 Positional error (mm) of EE—Inverse Kinematics method

	Error in EE position(mm)		
Path step	180 steps	360 steps	540 steps
0	0	0	0
1	− 0.223078	− 0.08868	− 0.04393
2	0.0785037	− 0.09296	− 0.04599
3	− 0.232187	− 0.03632	− 0.52046
4	− 0.088762	0.048268	− 0.08876
5	0.0294426	0.31402	0.082301
6	0.3721936	0.224714	0.085252

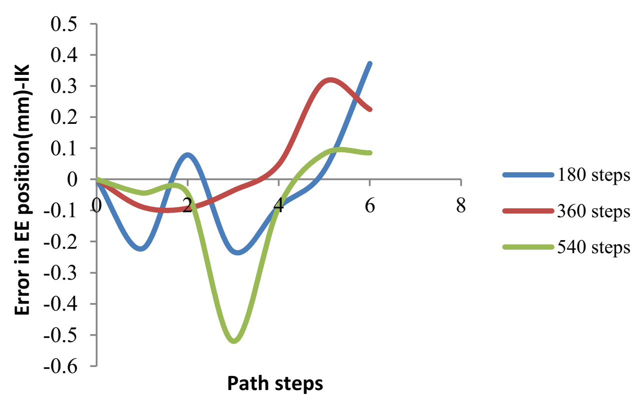


Fig. 13 Positional error in tests with different numbers of steps (IK). Range of error has mostly reduced when no of steps increased

with graphs from [20]. Comparing Graphs for Joint angles by Inverse Kinematics and Inverse Jacobian approach, they have resulted to be very similar. It is because of very small difference of errors obtained by each method. Although the total time required to complete the journey (Time has changed linearly with the number of steps), shape of curves was consistent with observations if we compare graphs of joint angles with all steps. It indicates Accuracy and convergence of the process that is used.

6 Conclusions

Following the above experiments, Inverse Kinematics approach is found to be more accurate with less error in the current application of three-dimensional curved paths as compared to the Inverse Jacobian approach. For paths that are curves in 2D or 3D Inverse Kinematics approach may

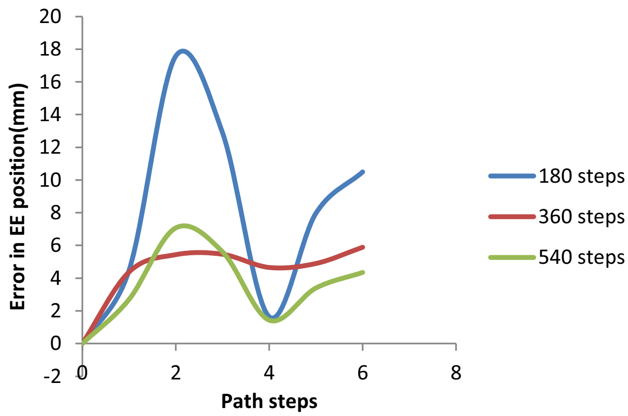


Fig. 14 Positional error in tests with different numbers of steps (IJ). Range of error has mostly reduced with no of steps increased [21]

be opted where the computational error is very less. Inverse Jacobian approach which is based on linear approximation, should be avoided for continues paths where 2D or 3D curves are involved. However, Inverse Jacobian matrix method may be used additionally to test for singularities since it involves Jacobian Matrix finding. A physical Robot setup, where we give Joint angles CSV file to trace the Path, will be required for experimental validation. However, in subsequent work We carry out Inverse Kinematic based Simulation for the above Scenarios and compare with current results for validation.

The Inferences are that the positional errors in IK Program considerably lower than that of Inverse Jacobian approach, and IK approach can be recommended for One dimensional, two dimensional or even three-dimensional paths with very negligible error. The above work was also carried out for different combinations of Pipe radii and angle of intersection and result trend has been repeated.

The Major Conclusions are that IK program is better than Inverse Jacobian Program when it comes to positional errors and IK program is recommended over Inverse Jacobian approach. The future scope is that more number of profiles may be analyzed and simulated using this program by incorporating their parametric data into the program. It requires a slight 10 to 15% modifications to the program. The programs are very generalized.

7 Programming

We wrote more than 400 lines of MATLAB code for this work. Separate programs are written to execute Inverse Kinematics and Inverse Jacobian approaches. The programs take Radii of intersecting pipes and intersection angle from the user as Input and perform calculations and produce graphs. Hence the programs are claimed to be very generic and take up any values for dimensions of the pipes. We measured time taken to execute a program and it is found to be in few minutes. As the no of steps in the path increased, the time taken to execute increased linearly. When we measured time taken to execute a program, it turned out to be less than 5 min. As the no of steps divided into in the required path increased, the time taken to execute the MATLAB program increased linearly. Also, if the no. of. Joints of the Robot are more, the IK Solution computation time increases exponentially with no. of. Joints of the Robot.

References

1. Singh, R., Pathak, V.K., Sharma, A., Chakraborty, D., Saxena, K.K., Prakash, C., Buddhi, D., hazim Salem, K.: Caster walker GAIT trainer (CGT): a robotic assistive device. Robot. Auton. Syst. **159**, 104302 (2023)
2. Chakraborty, D., Rathi, A., Singh, R., Pathak, V.K., Srivastava, A.K., Sharma, A., Saxena, K.K., Kumar, G., Kumar, S.: External six-bar mechanism rehabilitation device for index finger: development and shape synthesis. Robot. Auton. Syst. **1**(161), 104336 (2023)
3. Sarkhel, P., Dikshit, M.K., Pathak, V.K., Saxena, K.K., Prakash, C., Buddhi, D.: Robust deflection control and analysis of a fishing rod-type flexible robotic manipulator for collaborative robotics. Robot. Auton. Syst. **1**(159), 104293 (2023)
4. Sharma, S.K., Grewal, H.S., Saxena, K.K., Mohammed, K.A., Prakash, C., Davim, J.P., Buddhi, D., Raju, R., Mohan, D.G., Tomków, J.: Advancements in the additive manufacturing of magnesium and aluminum alloys through laser-based approach. Materials **15**(22), 8122 (2022)
5. Sharma, S.K., Grewal, H.S., Saxena, K.K., Mohammed, K.A., Prakash, C., Davim, J.P., Buddhi. D., Raju, R., Mohan, D.G., Tomków, J.: Advancement of magnesium and aluminum alloys in the role of additive manufacturing
6. Kale, B.S., Bhole, K.S., Dhongadi, H., Oak, S., Deshmukh, P., Oza, A., Ramesh, R.: Effect of polygonal surfaces on development of viscous fingering in lifting plate Hele-Shaw cell. Int. J. Interact. Des. Manuf. (IJIDeM). **1**, 1–8 (2022)
7. Stockie, J.: The geometry of intersecting tubes applied to controlling a robotic welding torch, (1999)
8. Liu, Y., Liu, J., Tian, X.: An approach to the path planning of intersecting pipes weld seam with the welding robot based on non-ideal models. Robot. Comput.-Integr. Manuf. **55**, 96–108 (2019)
9. Angeles, J., Rojas, A., Lopez-Cajun, C.S.: Trajectory planning in robotic continuous-path applications. IEEE J. Robot. Autom. **4**(4), 380–385 (1988)
10. Yao, T., Gai, Y., Liu, H.: Development of a robot system for pipe welding. Int. Conf. Meas. Technol. Mech. Autom. **1**, 1109–1112 (2010)
11. Singamaneni, P., Dewangan, P., Guhan, P., et al.: A deep reinforcement learning approach for dynamically stable inverse kinematics of humanoid robots, (2018)
12. Bipin, K., Duggal, V., Krishna, M.: Autonomous navigation of generic Quadrocopter with minimum time trajectory planning and control. In: 2014 IEEE international conference on vehicular electronics and safety, ICVES 2014. 66–71 (2015). https://doi.org/10.1109/ICVES.2014.7063726
13. Amici, C., Cappellini, V.: Inverse Kinematics of a serial robot. MATEC Web Conf. **53**, 01060 (2016)

14. Buss, S.: Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods. In: IEEE transactions in robotics and automation 17 (2004)
15. Rohmer, E., Singh, S., Freese, M.: V-REP: A versatile and scalable robot simulation framework (2013)
16. D'Souza, A., Vijayakumar, S., Schaal S.: Learning inverse Kinematics (2001)
17. Quigley, M., Conley, K., Gerkey, B., et al.: ROS: an open-source robot operating system (2009)
18. Othayoth, R., Chittawadigi, R., Joshi, R., et al.: Robot kinematics made easy using RoboAnalyzer software. Comput. Appl. Eng. Educ. (2017). https://doi.org/10.1002/cae.21828
19. Somasundar, A.V.S.S., Gandham, Y., Kesavulu, K.: Singularity analysis of Kuka 6 DOF robot for motion simulation. Int. J. Mech. Product. Eng. Res. Dev. **9**, 223–228 (2019)
20. Somasundar, A.V.S.S., Yedukondalu, G.: Inverse Jacobian programming approach to robotic path planning of various path profiles. Int. J. Interact. Des. Manuf. (IJIDeM) (2022). https://doi.org/10.1007/s12008-022-01155-x
21. Somasundar, A., Gandham, Y.: Robotic path planning and simulation by Jacobian inverse for industrial applications. Procedia Comput. Sci. **133**, 338–347 (2018)