

IMPLEMENTATION OF INVERSE KINEMATIC AND TRAJECTORY PLANNING ON 6-DOF ROBOTIC ARM FOR STRAIGHT-FLAT WELDING MOVEMENT

1,2) Departement of Mechanical Engineering, State Polytechnic of Malang, Jl. SoekarnoHata 9, Malang, Indonesia

3)Departement of Mechanical Engineering, State Polytechnic of Jakarta, Jl. Prof. DR. G.A. Siwabessy, Kampus Universitas Indonesia, Depok 16425

Correponding email ¹⁾ :
Arifnurhuda1997@gmail.com

Muhammad Arif Nur Huda ¹⁾, Sugeng Hadi Susilo ²⁾, Pribadi Mumpuni Adhi ³⁾

Abstract. Robotic arms have been used in various processes such as for moving goods, welding, assembling, and painting. In the case of welding and painting, it is necessary to move the end-effector robot accurately and smoothly to follow the specified trajectory. In robotic arm control, 2 things are important to be analyzed and implemented in controlling the motion of the robotic arm, namely inverse kinematic and trajectory planning. In this study, the inverse kinematic and trajectory planning algorithms are implemented to the robotic arm controller in the form of an Arduino Mega 2560 microcontroller. The inverse kinematic solution uses geometric and algebraic analytical methods. while the trajectory planning method is using LSPB (Linear Segment Parabolic Blend) Trajectory in Cartesian Space. Data retrieval is done by giving 2 input coordinates of the desired position and orientation, then the data in the form of the joint angle value will be measured using a rotary encoder as an angle sensor. Furthermore, the joint angle measurement value is converted in cartesian coordinates to get the end-effector position. Data analysis is done by comparing the data value of each joint angle with the calculated value so that the error value appears. The results showed that the inverse kinematic and trajectory planning algorithms were successfully applied to the 6-DOF robotic arm to perform straight-flat welding movements. Inverse kinematic testing on both input coordinates, the average error value for joints 2, 3, and 5 is 1.82°, 1.26°, and 2.08°. Meanwhile, the average error of the end-effector position at the x and z coordinates is 2.08 mm and 12.9 mm, respectively. Then for the trajectory planning test, the error value for the end-effector position in the x and z coordinates is 2.25 mm and 10.7 mm.

Keywords : inverse kinematic, trajectory planning, robotic arm, DOF, welding movement.

1. INTRODUCTION

In the era of the industrial revolution 4.0, many manufacturing industries have implemented several cutting-edge technologies. One technology that is gaining popularity in the manufacturing industry is a robotic arm manipulator. The use of this technology is because today's industry wants to further increase effectiveness and productivity. The increase in effectiveness and productivity is because the robotic arm can increase speed and accuracy [1]. In addition, a robot can replace jobs that were previously difficult and dangerous to become easier and safer [2]. This technology in the form of robotic arms has been used in various industrial processes, such as moving goods, cutting, casting, welding, painting, assembling, and others.

In the case of applying robots for certain purposes such as welding, and painting, it is necessary to move the end-effector robot accurately and smoothly to follow the specified trajectory [3]. In this study, a robot motion control will be discussed to perform a simple straight welding movement in the flat position. 2 things are

important to be analyzed and implemented in controlling the motion of the Robotic Arm. The two things are inverse kinematic and trajectory planning. From the design process to the experiment it is important to analyze the kinematic solution and plan the trajectory of the robot [4]. In robotics, inverse kinematic is an equation to find joint parameters with known the desired position and orientation of the end-effector [5]. While the trajectory is a specification of the position of the robot in a function of time and the trajectory should be quite smooth in a function of time [6]. By applying a combination of the inverse kinematic and trajectory planning algorithm, it can produce an accurate and smooth of the robotic arm movement.

There are previous studies that have been used as literature studies, such as the inverse kinematic solution on the 6 Degrees of Freedom (DOF) robotic arm analytically using algebraic and geometric methods [7-9]. Then some studies discuss trajectory planning methods such as the LSPB trajectory method [10], [11], and polynomial trajectory [12], [13]. Several studies have also implemented inverse kinematic and trajectory planning methods on robotic arms. Such as research on the application of inverse kinematic and fuzzy logic on a 4-DOF robotic arm for pick and place purposes [14], the application of inverse kinematic on a 3-DOF robotic arm [15], research on the application of inverse kinematic and trajectory planning on a 5 DOF Scorbot robotic arm based on simulation using Matlab software [16], and research on controlling 4 DOF robots uses Cartesian-space trajectory planning [17]. However, some of the above application studies are still applied to small-scale robots and for purposes other than welding.

By looking at previous studies, in this study, the inverse kinematic and trajectory planning algorithms are applied in a 6 DOF robotic arm motion control to perform simple welding movements in the form of a straight line.

2. METHODS

This research is experimental and applied research by implementing the inverse kinematic algorithm and trajectory planning to the 6-DOF robotic arm controller as a control for straight-flat welding motion. The Robotic Arm controller in this study is the Arduino Mega 2560.

2.1 System Implementation

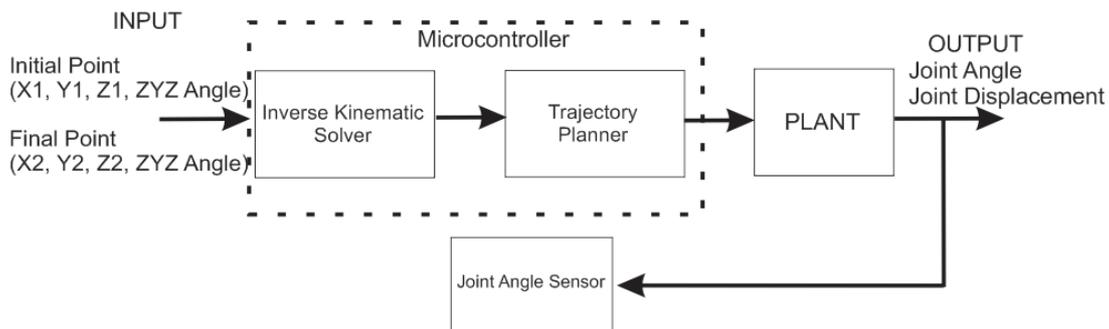


Figure 1. Implementation System Diagram

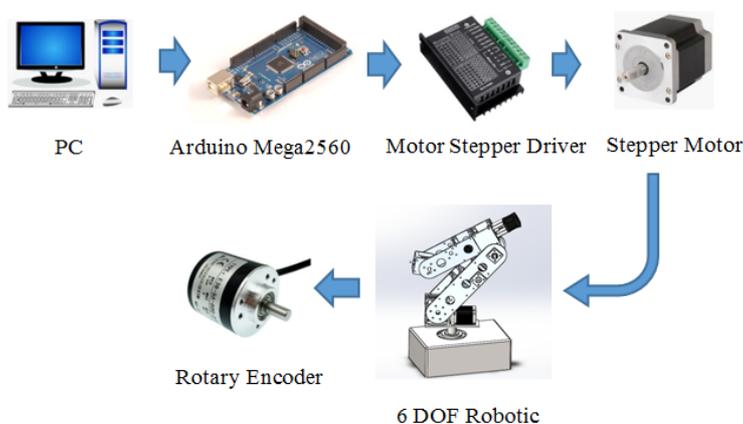


Figure 2. Hardware Implementation

The implemented system in Figure 1 is an open-loop system without any feedback from sensors. The way this implementation system works is that initially 2 input coordinate points are given along with the desired

orientation, $(x_1, y_1, z_1, \text{zyz angle } 1)$ and $(x_2, y_2, z_2, \text{zyz angle } 1)$. Then with the inverse kinematic algorithm, the microcontroller will calculate the angle value at each joint for the two positions. Then the trajectory planning algorithm will calculate how much the joint angle changes each time it changes. Then the calculation is converted by the microcontroller into a digital control voltage signal sent by the stepper motor driver. Furthermore, the motor driver will send voltage to the stepper motor. Finally, the stepper motor converts the voltage signal into rotation which can finally rotate the joint robot mechanism. To monitor the joint angle of the robotic arm, a rotary encoder has been installed on each robot joint as an angle sensor. The movement of the end-effector robot is expected to gently traverse a straight line between the 2 input coordinates to perform straight welding. Full Hardware Implementation diagram is shown in Figure 2.

2.2 Robot Specification

In this study, a robotic arm that has 6 DOF has been designed using a stepper motor as the actuator. Each stepper motor is connected to a worm gear mechanism and a timing belt as shown in Figure 3b. The kinematic structure model of the robotic arm is shown in Figure 3a.

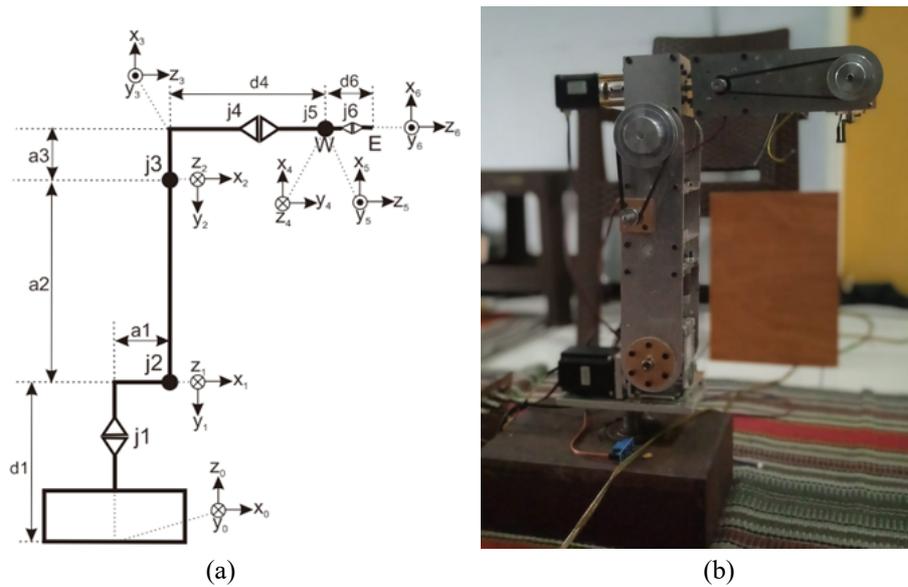


Figure 3. Kinematic Model (a) and Real Robotic Arm (b)

The letters W and E in the Figure 3a are wrist points and end-effector points, respectively. The end-effector in this research robot arm is a Mig welding torch. Table 1 is the kinematics specification of the robot which is indicated by the Denavit Hartenberg (DH) Parameters.

Table 1. DH) Parameter of 6-DOF Robotic Arm

Joint	Alpha (α_{i-1})	Arm Length (a_{i-1})	Offset Joint (d_i)	Theta (θ_i)
1	-90°	a1=55 mm	d1=200	θ_1
2	0°	a2=280 mm	0	θ_2
3	-90°	a3=60 mm	0	θ_3
4	90°	0	d4=220 mm	θ_4
5	-90°	0	0	θ_5
6	0°	0	d6=40 mm	θ_6

2.3 Inverse Analysis Method

In this study, the inverse kinematic algorithm implemented on the robotic arm controller uses analytical methods, namely geometric and algebraic methods. Although there are many inverse kinematic solution methods such as analytic, numeric, or soft computing in some literature, if the structure of the robot has a closed-form solution, analytical methods will be the fastest calculation [10]. A geometrical method is used to calculate joint

1, joint 2, and joint 3. While the algebraic method is used to calculate joints 4, 5, and 6. The following is the solution for joints 1, 2, and 3.

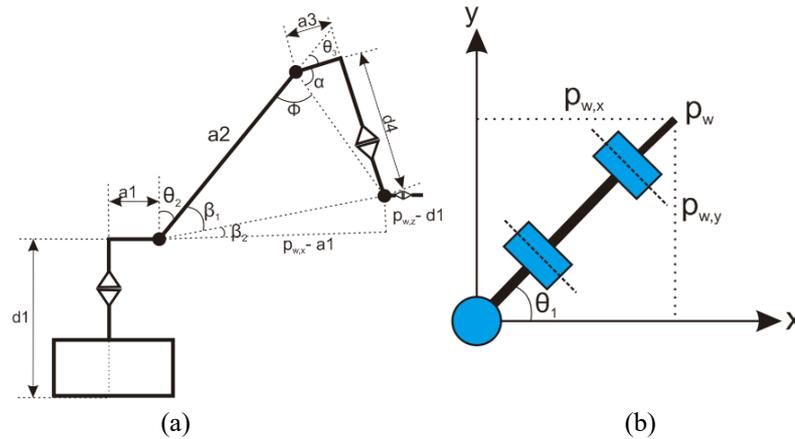


Figure 4. Kinematic Structure (a) Side view (b) Top View

Joint 1

$$\theta_1 = \text{Atan} \left(\frac{p_{w,y}}{p_{w,x}} \right) \tag{1}$$

Joint 3

$$\theta_3 = \pi - \phi - \alpha \tag{2}$$

As for finding the values of and the tone is as follows:

$$\alpha = \text{Atan} \left(\frac{d_4}{a_3} \right) \tag{3}$$

$$\phi = \text{Acos} \left(\frac{(a_2^2 + (d_4^2 + a_3^2) - ((P_{w,z} - d_1)^2 + (P_{w,x} - a_1)^2))}{2a_2(d_4^2 + a_3^2)} \right) \tag{4}$$

Joint 2

$$\theta_2 = 90 - \beta_1 - \beta_2 \tag{5}$$

The values of 1 and 2 can be found with the following equation.

$$\beta_2 = \text{Atan} \left(\frac{P_{w,z} - d_1}{P_{w,x} - a_1} \right) \tag{6}$$

$$\beta_1 = \text{Acos} \left(\frac{a_2^2 + ((P_{w,z} - d_1)^2 + (P_{w,x} - a_1)^2) - (d_4^2 + a_3^2)}{2a_2 \sqrt{(P_{w,x} - a_1)^2 + (P_{w,z} - d_1)^2}} \right) \tag{7}$$

Here are the solutions for joints 4, 5, and 6 using the algebraic method. To get $\theta_4, \theta_5, \theta_6$, value is needed calculations to find the rotation matrix from frames 3 to 6.

$$R_6^3 = (R_3^0)^{-1} R_6^0 \tag{8}$$

$$R_6^3 = \begin{bmatrix} R_{611}^3 & R_{612}^3 & R_{613}^3 \\ R_{621}^3 & R_{622}^3 & R_{623}^3 \\ R_{631}^3 & R_{632}^3 & R_{633}^3 \end{bmatrix} \tag{9}$$

$$\theta_4 = \text{Atan} (-R_{623}^3, -R_{613}^3) \tag{10}$$

$$\theta_5 = \text{Acos}(R_{633}^3) \tag{11}$$

$$\theta_6 = \text{Atan}(-R_{632}^3, -R_{631}^3) \tag{12}$$

Table 2 is the result of analytical calculation of the joint angle value of the 2 input coordinates that will be used in this study.

Table 2. Inverse Kinematic calculation Using Analytical Method

Input Coordinate (Px, Py, Pz, ZYZ Angle)	Joint Angle Calculation (°)					
	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
400, 0, 300, -90, 180, 90	0	-59.578	17.639	0	41.918	0
300, 0, 300, -90, 180, 90	0	-76.377	43.880	0	32.477	0

Through calculations using analytical methods (geometrical and algebraic method) at the two input coordinates that will be used for this research, there are 3 moving joints. The moving joints are joint 2, joint 3, and joint 5. Validation of the joint solution values for the 2 coordinate positions has also been carried out with an inverse kinematic solution using a numerical method. Table 3 is the result of calculating the joint value solution using the numerical method with the help of the Matlab Robotic Toolbox software.

Table 3. Inverse Kinematic calculation Using Numerical Method (Matlab)

Koordinat Input (Px, Py, Pz, ZYZ Angle)	Joint Angle Calculation (°)					
	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
400, 0, 300, -90, 180, 90	0	-59.577	17.730	0	41.847	0
300, 0, 300, -90, 180, 90	0	-76.379	43.974	0	32.406	0

In table 2 and table 3, It can be seen that the difference in the value of the inverse kinematic solution with the value of the geometric solution is very small. These results prove that the analytical method inverse kinematic algorithm can be implemented into the 6 DOF Robotic Arm controller.

2.4 Trajectory Planning Method

The trajectory used in this robotic arm is LSPB (Linear Segment Parabolic Blend) trajectory in Cartesian-space. The Robotic Arm is expected to be able to move from the initial coordinate point to the final coordinate point smoothly. The trajectory time is planned for 27 seconds with a maximum end-effector speed of 4.5 m/s. With the help of the Matlab Robotics Toolbox software, the robot movement plan can be described in graphical form as shown in Figure 5. The circle line is the end-effector movement plan at the x-coordinate (Px), while the triangle line is the end-effector trajectory plan at the z-coordinate (Pz).

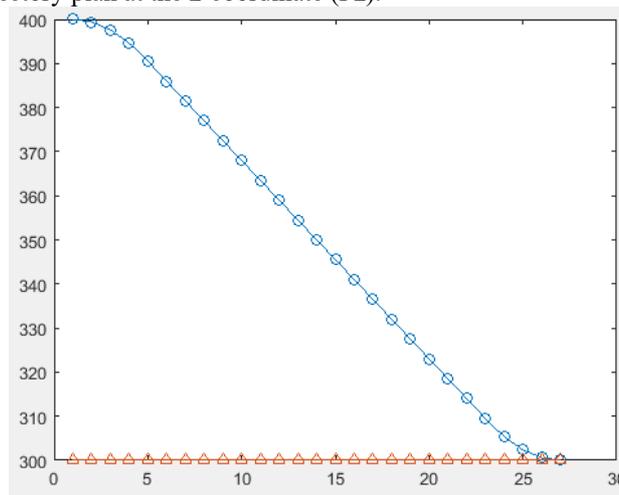


Figure 5. Trajectory Planning

2.5 Data Collection

Data collection is divided into 2 stages. The first stage of data collection is to test the inverse kinematic algorithm when the position coordinates are input. The data to be measured is in the form of the joint angle of the Robotic Arm. Then to find the position of the end-effector, forward kinematic calculations are carried out with the measurement value of the joint angle. Meanwhile, for the second stage, trajectory planning data was collected by measuring changes in joint angles per unit time. The angle value can also be converted into a

change in the position of the end-effector in Cartesian coordinates using forward kinematics so that the end-effector trajectory can be known. The value of the end-effector trajectory is then compared with the value of the trajectory planning, using the matlab robotics toolbox

3. RESULTS AND DISCUSSION

Here is the real movement of the 6-DOF Robotic Arm in carrying out straight welding movements in a flat position.

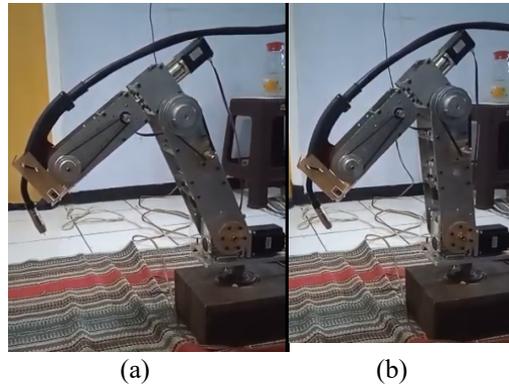


Figure 6. Robot Movement Testing (a) Position 1 (b) Position 2

In Figure 6, the robot arm moves from position 1, which is the coordinate point 400, 0, 300, to position 2, which is the coordinate point 300, 0, 300 by maintaining the orientation of the robot arm, namely -90° , 180° , 90° in Euler angle or zyz angle. In Figure 6, the MIG Welding Torch has also been installed as the end-effector of the robotic arm.

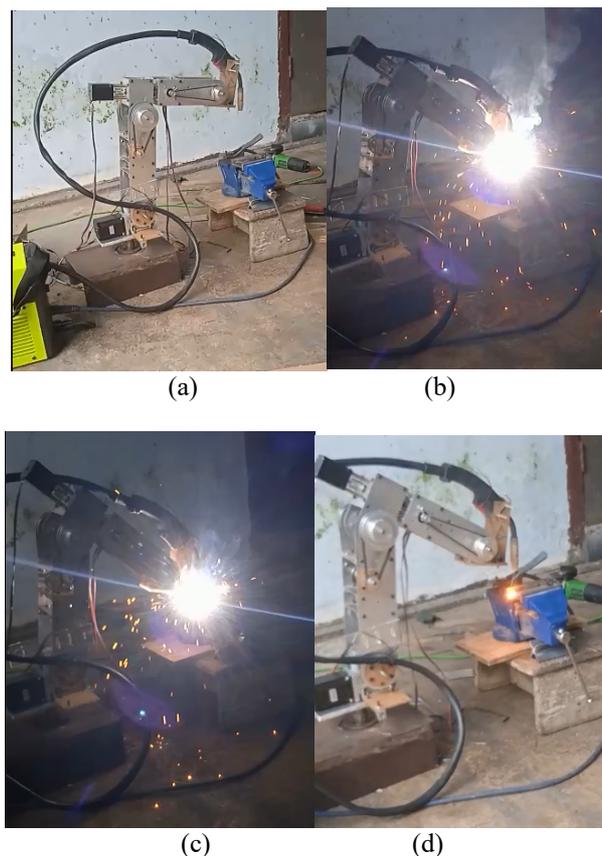


Figure 7. Straight Welding Task Using Robotic Arm (a) Home Position, (b) Start Welding Process, (c) Final Welding process, (d) Back to Home Position.

After testing the movement of the robot, in Figure 7 the robot arm is also used in a real environment, namely to perform straight welding tasks in a flat position. The movement of the robot starts from the home position, then

to position 1 to start welding process, then to position 2 as the end point of welding process, and returns to the home position by maintaining the orientation of the end-effector.

3.1 Inverse Kinematic Testing

Inverse kinematic testing is carried out by inputting 2 coordinate positions along with their orientation. The coordinates of 1 or initial position are (400mm, 0mm, 300mm, -90°, -180°, 90°) while coordinates 2 or the final position are (300mm, 0mm, 300mm, -90°, -180°, 90°). Real data in the form of the angle value of each joint will be measured using a rotary encoder sensor for joint angle readings. Then the angle data can be used to calculate the position of the end-effector using forward kinematic. The sensor reading data is compared with the calculated value. While the position of the end-effector will be compared with the desired input coordinates. From the test results, 3 joints play a role to form the two end-effector positions. Table 4 and Table 5 are observation data for joints 2, 3, and 5 as well as the position of the end-effector at x and z coordinat.

Table 4. Input Coordinate (400mm, 0mm, 300mm, -90°, -180°, 90°)

Sample	Joint Angle Measurement (°)			Joint Angle Error (°)			End-Effector Position (mm)		End-Effector Position Error (mm)	
	Joint 2	Joint 3	Joint 5	Joint 2	Joint 3	Joint 5	Px	Pz	Px	Pz
1	-57.15	18.68	40.28	2.63	0.96	1.57	400.7	282.5	0.7	17.5
2	-56.93	18.13	40.88	2.86	0.40	0.97	402.2	283	2.2	17
3	-57.83	18.68	40.73	1.95	0.96	1.12	399.7	286.5	0.3	13.5
4	-57.38	18.38	39.6	2.40	0.66	2.24	401.9	284.8	1.9	15.2
5	-57.15	18.17	40.05	2.63	0.45	1.79	402.7	284.1	2.7	15.9
6	-56.93	19.63	41.4	2.85	1.91	0.44	397	278.1	3	21.9
7	-57.15	19.07	39.83	2.63	1.35	2.02	399.7	281.2	0.3	18.8
8	-57.15	18.56	40.05	2.63	0.85	1.79	401.2	282.8	1.2	17.2
9	-57.38	19.35	40.05	2.40	1.63	1.79	401	281.66	1	18.34
10	-57.60	18.45	40.73	2.18	0.73	1.12	400.6	285.9	0.6	14.1

In Table 4, it can be observed that from 10 attempts at input coordinates (400mm, 0mm, 300mm, -90°, -180°, 90°) the highest t error occurred at joint 2, which was 2.86°. While the smallest error occurs at joint 3 which is 0.4°.

Table 5. Input Koordinat (300mm, 0mm, 300mm, -90°, -180°, 90°)

Sample	Joint Angle Measurement (°)			Joint Angle Error (°)			End-Effector Position (mm)		End-Effector Position Error (mm)	
	Joint 2	Joint 3	Joint 5	Joint 2	Joint 3	Joint 5	Px	Pz	Px	Pz
1	-74.93	18.68	40.28	1.23	1.42	2.7	297.1	290.8	2.9	9.2
2	-74.93	18.13	40.88	1.45	0.42	1.43	301	292.7	1	7.3
3	-75.83	18.68	40.73	0.55	1.69	2.93	296.5	292.8	3.5	7.2
4	-75.38	18.38	39.6	1.00	1.476	3.60	298.2	291.5	1.8	8.5
5	-75.15	18.17	40.05	1.23	1.476	2.93	298.2	291.5	1.8	8.5
6	-74.93	19.63	41.4	1.45	2.376	2.26	294.4	287	5.6	13
7	-75.15	19.07	39.83	1.23	1.814	3.38	297	289.6	3	10.4
8	-75.15	18.56	40.05	1.23	1.139	1.58	298.2	291.5	1.8	8.5
9	-75.38	19.35	40.05	1.00	2.094	3.16	295.6	289.7	4.4	10.3
10	-75.60	18.45	40.73	0.78	1.307	2.93	298.1	293	1.9	7

In table 5, it can be observed that out of 10 attempts at the input coordinates (300mm, 0mm, 300mm, -90°, -180°, 90°) the highest error occurred at joint 5, which was 3.6°. While the smallest error occurs at joint 3 which is equal to 0.42°.

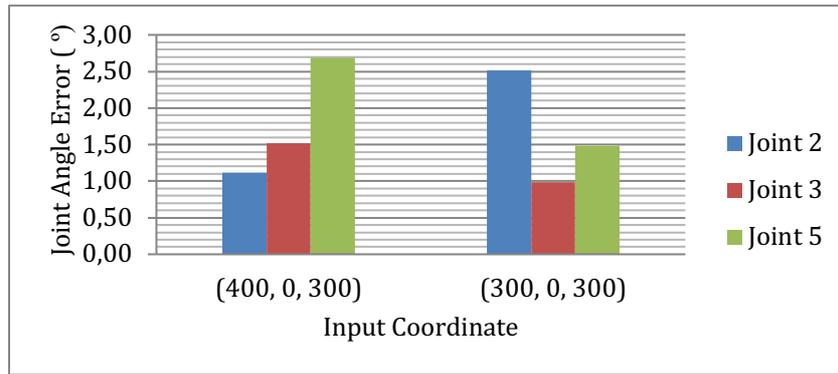


Figure 8. Joint Angle Mean Error

In Figure 8, it can be seen that the average error values for joint 2, joint 3, joint 5 at the input coordinates of 400, 0, 300 with Euler angles of -90, 180, 90 are 1.12°, 1.52°, and 2.69°, respectively. While the average error values for joint 2, joint 3, and joint 5 at the input coordinates of 300, 0, 300 are 2.51°, 0.99°, and 1.48°, respectively. The average error for both positions is 1.82° for joint 2, 1.26° for joint 3 and 2.08° for joint 5.

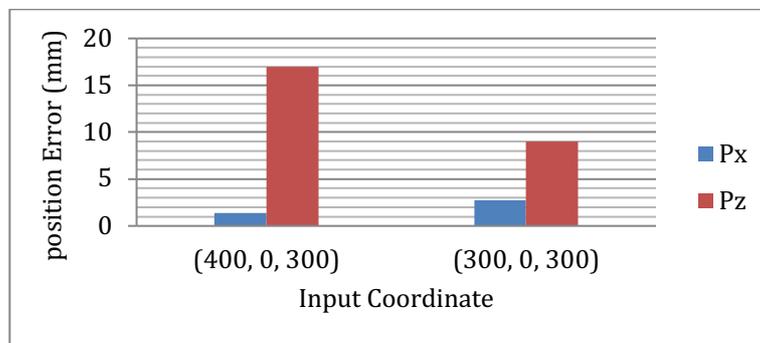


Figure 9. End-Effector Position Mean Error

By performing forward kinematic calculations, the measurement data can be converted into the end-effector position. It can be seen in Figure 9, that the average error value at the end effector position at the input coordinates (400, 0, 300). By comparing the position of the end-effector with the input coordinates, the average error value is obtained at the x and y coordinates. while the average error value for the end-effector position (400, 0, 300) in the x and y coordinates is 1.39 mm and 16.94 mm, respectively. Meanwhile, the average error value for the position of the end-effector (300, 0, 300) in the x and y coordinates is 2.77 mm and 8.99 mm, respectively. The average error for the two coordinate positions when compared with the input coordinates we want is for Px, and Pz is 2.08 mm and 12.9 mm.

3.2 Trajectory Planning Testing

The trajectory planning test was carried out by monitoring changes in the angles of joints 2, 3, and 5 of the Robotic Arm every second for 27 seconds. To obtain the displacement of end-effector data, the joint value measurement data is converted into Cartesian coordinates using forward kinematic calculations. Table 6 is data on changes in joint angle and end-effector position for 27 seconds with sampling every second.

Table 6. Experimental data of robotic arm trajectory

Time (s)	Joint Angle (°)			End-Effector Position (mm)		
	Joint 2	Joint 3	Joint 5	Px	Py	Pz
0	-57.15	18.68	40.28	400.7	0	282.5
1	-57.15	18.80	40.05	400.1	0	282
2	-57.38	19.24	40.05	398.8	0	282
3	-58.05	20.03	39.60	396	0	283.3
4	-58.73	21.09	39.38	392	0	283.8
5	-59.40	22.05	38.70	388.8	0	284.6

Time (s)	Joint Angle (°)			End-Effector Position (mm)		
	Joint 2	Joint 3	Joint 5	Px	Py	Pz
6	-60.30	22.78	38.25	385.8	0	286.3
7	-60.75	23.79	37.58	382	0	286.5
8	-61.65	24.64	36.90	379.3	0	288.8
9	-62.55	25.71	36.68	375.1	0	290.3
10	-63.45	27.00	35.78	370.5	0	291.1
11	-64.13	28.13	35.55	366.2	0	291.1
12	-64.80	29.59	35.10	360	0	290
13	-65.48	30.60	34.65	357	0	290.4
14	-66.38	31.95	34.20	351.9	0	290.8
15	-67.28	33.19	33.98	346.8	0	291.4
16	-67.95	34.43	33.75	341.9	0	290.9
17	-68.85	35.55	33.30	337.5	0	291.8
18	-69.75	36.73	33.08	332.5	0	292.5
19	-70.65	37.91	32.63	327	0	293.1
20	-71.55	39.09	32.18	322.8	0	293.8
21	-72.23	40.33	31.50	318	0	293
22	-72.90	41.46	31.28	313.6	0	292.7
23	-73.58	42.64	30.60	309.1	0	292.1
24	-74.25	43.43	30.38	305	0	292.7
25	-74.70	44.21	30.15	302.5	0	292.2
26	-74.93	44.78	29.93	300	0	292.5
27	-74.93	45.11	29.93	299.9	0	290.7

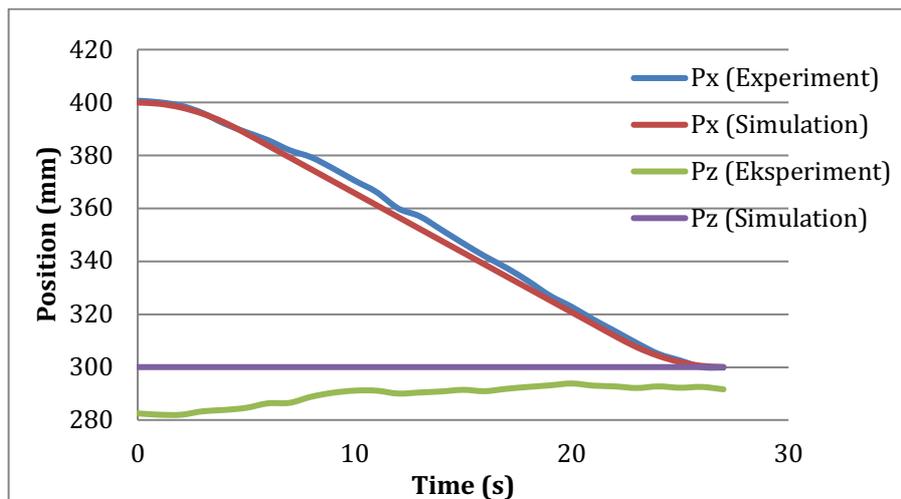


Figure 10. End-effector trajectory Eksperimen vs Simulation

Figure 10 is a comparison of the trajectory end-effector in Cartesian coordinates. The red and gray lines represent the trajectory of the robotic arm on the x and z axes in the simulation using the Matlab Robotics toolbox, or in other words, an illustration of the trajectory planning. While the red and orange lines are the actual end-effector trajectories from the calculation results on the x and z axes with the joint robot measurement values obtained. If we observe the movement of the end-effector on the x-axis from the coordinates $P_x=400\text{mm}$ to $P_x=300\text{mm}$ it looks quite smooth, this is evidenced by the presence of a parabolic curve at the beginning and end of each robot's movement. However, when compared to the trajectory planning chart, it can be seen that there are still error values. Because not all parts of the end-effector movement graph coincide with the desired trajectory graph, or in other words, there is an error value in the movement of the Robotic Arm. Then on the z-axis, there is a fairly large error value. It can be seen that the movement of the end-effector has not been able to fully maintain the position of the end-effector at a height of 300 mm on the z-axis. In addition, it can be seen that the movement of the end-effector in the z-axis changes which tends to increase, in the sense that the movement of the end-effector tends to increase. The following is a graph of the error value of the end-effector position when moving

from the coordinate point (400, 0, 300)mm to the coordinate point (400, 0, 300) while maintaining the Euler angle orientation value (-90°, 180°, 90°).

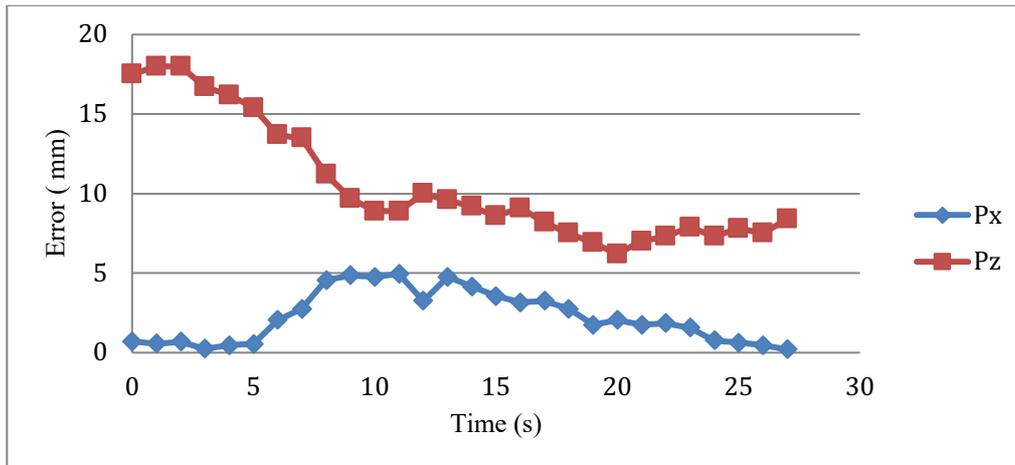


Figure 11. End-Effector Absolute Error Trajectory

The graph in Figure 11 above is a comparison of the trajectory end-effector in Cartesian coordinates. It can be seen at the end-effector position that at the end-effector position in the x-coordinate, the largest absolute error value that occurs is 4.95 mm, while the largest error that occurs in the y-coordinate is 18 mm. The average x-coordinate error value is 2.25 mm and the z-coordinate average error is 10.7 mm.

4. CONCLUSION

The inverse kinematic algorithm with analytical method (geometrical and algebraic) and LSPB trajectory planning method in cartesian space has been successfully applied to the 6-DOF robotic arm to perform straight flat welding movements. Inverse Kinematic testing on both input coordinates, the average error value for joints 2, 3, and 5 is 1.82°, 1.26°, and 2.08°. Meanwhile, the average error of the end-effector position at the x and z coordinates is 2.08 mm and 12.9 mm, respectively. Then for the trajectory planning test, the average error value for the end-effector position in the x and z coordinates is 2.25 mm and 10.7 mm.

5. ACKNOWLEDGMENT

The first author thanks the Director of the State Polytechnic of Malang for granting scholarships to study in the Applied Masters Program in Manufacturing Technology Engineering, Department of Mechanical Engineering, State Polytechnic of Malang for two years.

6. REFERENCES

- [1] Y. Jadega and B. Pandya, "Design and Development of 5-DOF Robotic Arm Manipulators," *International Journal of Scientific & Technology Research*, vol. 8, no.11, pp.2158-2167, 2019.
- [2] J. Wisnu, *Robotika: Teori dan Aplikasi*. Jakarta: Universitas Indonesia, 2012.
- [3] Y. Chen, L. Li, X. Ji, "Smooth and Accurate Trajectory Planning for Industrial Robots," *Advances in Mechanical Engineering (AIME)*, 2014.
- [4] M. Dahari and J.D. Tan, "Forward and Inverse Kinematics Model for Robotics Welding Process Using KR-16KS KUKA Robot," *Fourth International Conference on Modeling, Simulation and Applied Optimization*, 2011.
- [5] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. London: Pearson Education Ltd, 2005.
- [6] Kevin M. Lynch and Frank C. Park, *Modern Robotics Mechanics, Planning, and Control*. Cambridge : Cambridge University Press, 2017.
- [7] A. Khatamian, "Solving Kinematics Problems of a 6-DOF Robot Manipulator," *Int'l Conf. Scientific Computing*, 2015.
- [8] O. F. Sevem and A. Ankarah, "Inverse Kinematic Analysis of IRB120 Robotic Arm," in *SETSCI Conference Proceedings, 3rd International Symposium on Innovative Approaches in Scientific Studies, Section: Engineering and Natural Sciences*, 2019.
- [9] T. Aravinthkumar, M. Suresh, B. Vinod, "Kinematic Anlysis of 6 DOF Articulated Robotic Arm," *International Research Journal Of Multidiciplinary Technovation*, vol. 3, no. 1, pp. 1-5, 2021.

- [10] S. Z. S Al-khayyt “Creating Through Points in Liniear Function with Parabolic Blends Path by Optimization Method,” *2nd International Conference on Control, Automation, and Artificial Intelligence* vol. 14, no. 1, pp. 77-89, 2018.
- [11] A. Walch, C. Eitzinger, S. Zambal, W. Palfinger, “LSPB Trajecotry Planning Using Quadratic Splines,” *3rd International Conference on Mechatronics and Robotics Engineering*, 2017.
- [12] S. Zribi, M. Mejerbi, H. Tlijani, J. Knani, V. Puig, “Smooth Motion Profile for Trajectory Planning of a Flexible Manipulator,” in *Proceeding International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 2018.
- [13] X. Zhao, M. Wang, N. Liu, Y. Tang “Trajectory Planning for 6-DOF Robotic Arm Based on Quintic Polynomial,” *2nd International Conference on Control, Automation, and Artificial Intelligence* vol. 134, 2017.
- [14] F. R. Iskandar, I. Sucahyo, M. Yantidewi, “Penerapan Metode Inverse Kinematik Pada Kontrol Gerak Robot Lengan Tiga Derajat Bebas,” *Jurnal Inovasi Fisika Indonesia (IFI)*, vol. 09, no. 02, pp. 64-71, 2020.
- [15] T. Dewi, S. Nurmaini, P Risma, Y Oktarina, and M. Roriz, “Inverse Kinematic Analysis of 4 DOF pick and place arm Robot Manipulator using Fuzzy Logic”, *International Journal of Electric and Computer Engineering (IJECE)*, vol. 10, no.02, pp.1376-1386, 2020.
- [16] R. Jhala, “Inverse Kinematics and Trajectory Planning of 5 DOF Scorbob-ER Vplus,” *International Journal of Engineering Development and Research*, vol. 8, no.3, pp.442-449, 2020.
- [17] M. F. Faris, A. Triwijayanto, I. Setiawan, “Perancangan Arm Manipulator 4 DOF Dengan Menggunakan Pengendalian Cartesian Space-Trajectory Planning,” *Transient*, vol. 1, no. 4, 2012.