Comparison of 4 DOF Arm Robot for Trajectory Planning with 3rd and 5th Polynomial Orders

Gunawan Rudi Cahyono
Mechanical Engineering Department
Lambung Mangkurat University
Banjarbaru, Indonesia
gunawan.cahyono@ulm.ac.id

Nurmahaludin
Electronics Engineering Department
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
mahaludin@poliban.ac.id

Muhammad Farouk Setiawan
Mechanical Engineering Department
Lambung Mangkurat University
Banjarbaru, Indonesia
mfarouksetiawan@gmail.com

Yusie Rizal
Electronics Engineering Department
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
yusie.rizal@poliban.ac.id

Joni Riadi
Electronics Engineering Department
Politeknik Negeri Banjarmasin
Banjarmasin, Indonesia
oni_riadi@poliban.ac.id

2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS) | 978-1-6654-0648-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/EECCIS54468.2022.9902924

***Abstract*— Technology's concept of automation has a big impact on life. One of the applications of this technology is the robotic arm. In this study, a 4 DOF (Degree of Freedom) arm robot with a microcontroller is tested for its movement results using the inverse kinematic method for calculating the angle of the robot joint and using a polynomial order with a reference point for the robot trajectory. The order of the polynomials used is 3rd and 5th order polynomials. Comparing the two polynomial orders is used to determine the best robot movement performance. The results are an error value for the accuracy of the robot's motion to move the object, respectively 0.5% and 0.9% on the x-axis, while 0.1% and 0.5% on the y-axis. The resulting average speed is 0.0402 m/s and 0.0431 m/s. The trajectory curve formed has a stable up and downtrend in each polynomial order.**

Keywords*— *Arm robot 4 DOF, inverse kinematic, trajectory, polynomial order

I. Introduction

Technological developments centered on automation currently have a good impact on adding efficiency values to a work environment. Time management is considered something that needs to be considered and needed by an industrial company. One of these automation technologies is applied in manufacturing a robot, namely an arm robot. The robot arm consists of three parts, namely the mechanical structure, actuator, and motion control system [1]. In the robotic arm, the mechanical structure is in the form of a joint or referred to as a joint, and the connection between the joints is called a link. Joints on the robotic arm consist of 2 types, namely revolute joints, which have a rotating movement, and prismatic joints, which have forward or backward movements [2].

To calculate the movement of jointed robots such as robotic arms, the inverse kinematic method can be a very effective solution to help calculate the angle of each joint robot, with the known variable being the coordinates of the destination [3]. The application of the inverse kinematic method in calculating the magnitude of the change in the angle of each joint aims to facilitate the programming process so that the end of the arm of the robot can reach the desired coordinate point, and a hybrid approach[4] The application of inverse kinematics in research [5] on the Design of Arm Manipulator 4 DOF using Cartesian Space-Trajectory Planning control says that the robot arm can follow the trajectory with the smallest and largest error values on average at the coordinate values of 0.5 cm, respectively, and 3 cm. Another study on the application of the Inverse Kinematic method to the motion control of a robotic arm with three degrees of freedom said that by entering the input variable in the form of coordinates (x, y, and z), the average error percentage for the x-axis is 0.42%, the Y-axis is 0.42%. 5.03%, and the Z-axis of 3.46% [2].

Both studies used the inverse kinematic method with two reference points: the starting point and the endpoint. Consideration of the use of trajectories can result in smooth robotic arm movements. The trajectory is a trajectory of motion from one point to another [6]. Using a polynomial function approach to the power of n-1, the intermediate points of n points are P1 (x1, y1), P2 (x2, y2), and P3 (x3, y3), Pn (xn,yn) can be determined [7]. And used for determining the robot trajectory points. The application of the trajectory is based on a curve composed of many points formed by polynomial equations. Observations on several orders and polynomial reference points can be applied, and it is still possible to improve the average percentage error of the robot's movement.

Another thing that is also considered is how the robot's speed and smooth movement parameters can be presented. This study aims to test the motion of the 4 DOF arm robot using the inverse kinematic method and add the 3rd and 5th order polynomial equation methods with 4 and 6 reference points, respectively, in the calculation. Testing the motion of the robot to see the level of accuracy and speed in moving objects from one point to another. The calculation results in the form of a trajectory curve can be used as a reference to observe the smooth level of the robot's movement. So that the comparison of the variation of the polynomial equation and the reference point becomes the basis for determining which variation produces the best performance for the 4 DOF arm robot motion application.

II. System Design Method

The steps of the procedure carried out in this study can be seen in the flow chart in Fig.1 as follows.

A. Designing Robot Arm

The 4 DOF arm robot is designed with propulsion and has four rotating joints : a waist joint, a shoulder joint, an elbow joint, and a wrist joint [8]. In this case, each robotic arm joint

978-1-6654-0648-2/22/$31.00 ©2022 IEEE

has angles that can be calculated using inverse kinematics. Determination of this angle with the aim that the end of the robot arm can reach the desired end-point position. The design of the robotic arm used in this study is shown in Fig. 2.

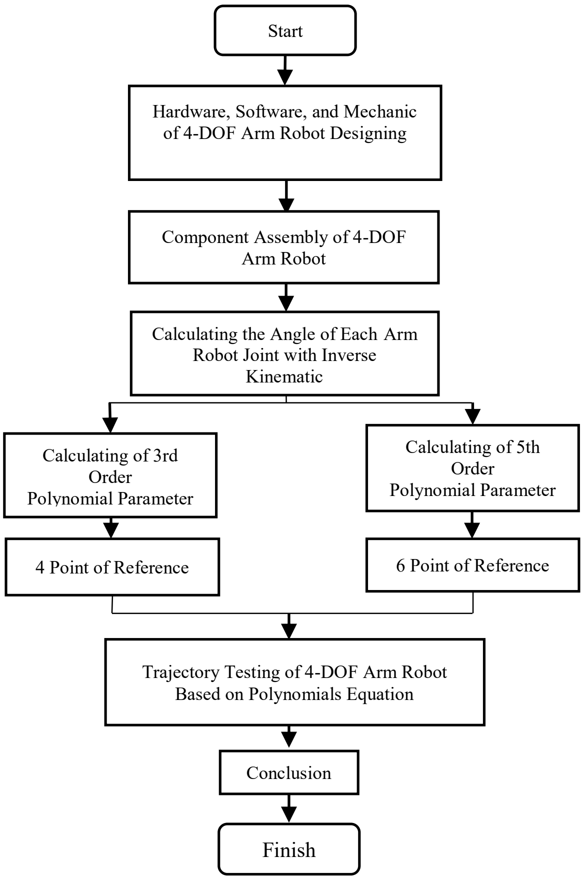

Start
Hardware, Software, and Mechanic of 4-DOF Arm Robot Designing
Component Assembly of 4-DOF Arm Robot
Calculating the Angle of Each Arm Robot Joint with Inverse Kinematic
Calculating of 3rd Order Polynomial Parameter
Calculating of 5th Order Polynomial Parameter
4 Point of Reference
6 Point of Reference
Trajectory Testing of 4-DOF Arm Robot Based on Polynomials Equation
Conclusion
Finish

Fig. 1. Research Flowchart

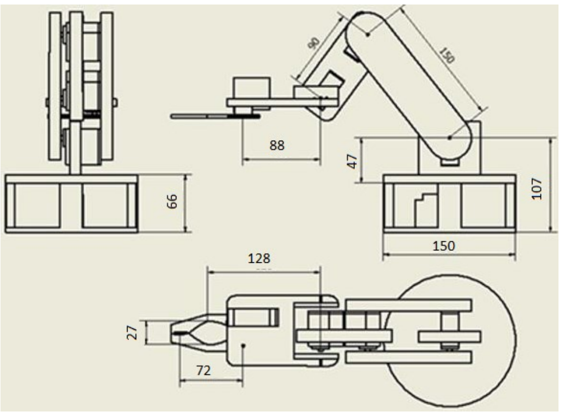


Fig. 2. Robot Arm Design and Dimensions

B. Determining the Reference Point

Determination of the position of the reference point is based on planning the path of the robotic arm, which consists of a collection of points that will be traversed by the robotic arm or known as a trajectory. For the determination of the reference points are 4 points for the 3rd order polynomial equation and 6 points for the 5th order polynomial equation from the robotic arm motion path planning. This number of reference points is the number needed to determine the polynomial equation. There are no special provisions in determining the reference point. As long as it is within reach of the robotic arm, it can still be used as a reference point.

C. Determining the Inverse Kinematic

Inverse kinematic has an algorithm with kinematic analysis method of robot transformation from Cartesian space to joint space [9]. The inverse kinematic equation produces parameter values for each robot joint at the given or desired end-effector position and orientation [10]. Robot motion control can be made easier by using inverse kinematics because the motion of the robot actuator can be known from the robot's position. This method is also very useful for shortening the processing time because there is no need to do a trial-error method to determine the position and orientation of the robot [2]. The implementation of inverse kinematics can also speed up the definition of the robot shape and has a small maximum error [11].

In an inverse kinematic calculation, one of the methods used is the geometric method. Geometric methods are generally used for robots with a small DOF [12]. This method uses calculations by looking from two two-dimensional (2D) points of view, namely from the top viewpoint and the front point of view of the robot, so that the angle of each joint robot can be determined [13]. Describe the geometry of the robot's movement; it can be done using Cartesian coordinates [14]. In this study, the inverse kinematics calculation with the geometric method is used. From the robot arm structure in Fig. 2, the data used in the inverse kinematic algorithm is the length of each link that connects each joint. For more details, it is described as follows.

1) Rotation cap (L1) = 107 mm = 0.107 m
2) Arm 1 (L2) = 150 mm = 0.15 m
3) Arm 2 (L3) = 90 mm = 0.09 m
4) Clamp (L4) = 128 mm = 0.128 m

The arrangement of the axes of the robot arm's coordinate plane is made to facilitate programming on the robot. In the inverse kinematic algorithm, the center of the coordinates is at the top of the rotation cover. Fig. 3 is the coordinate plane of the robot arm.

Furthermore, the inverse kinematic equation needs to be calculated to determine the angle of each joint robot. The calculation for each of these angles can be explained in Fig. 4 and Fig. 5.

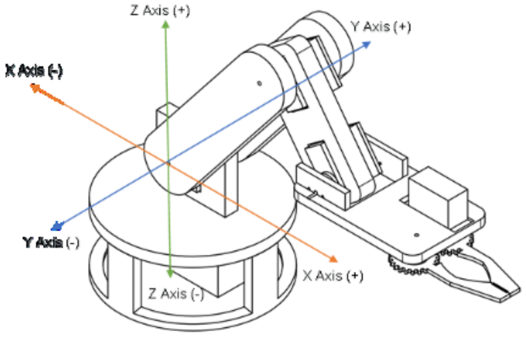


Fig. 3. Robot Arm Coordinate Field

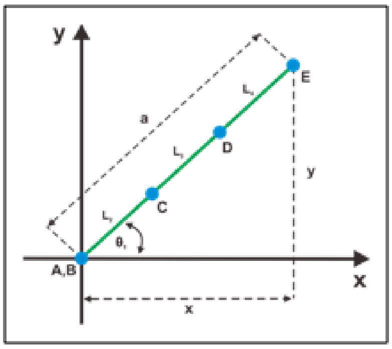


Fig. 4. Joint Robot Angle Calculation (XY plane Viewpoint)

In Fig. 4, it can be seen that the equation for calculating angle 1 in the XY plane is as follows.

$$\tan\theta_1 = \frac{y}{x} \quad (1)$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \quad (2)$$

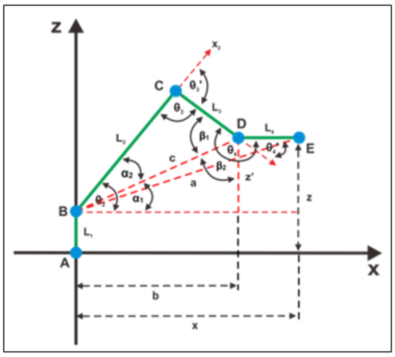


Fig. 5. Joint Robot Angle Calculation (XZ plane of view angle)

In Fig. 5, it can be seen the equations for calculating angle 2, angle 3, and angle 4 in the XZ plane. To determine the measure of the angle 2. Then it is necessary to find the measure of the alpha angles 1 and 2. The equations used are as follows.

$$z' = z - L_1 \quad (3)$$

$$b = x - L_4 \quad (4)$$

$$c = \sqrt{z' + b^2} \quad (5)$$

$$\alpha_1 = \tan^{-1}\left(\frac{z'}{b}\right) \quad (6)$$

By using the law of cosines in the triangle BCD. So that the following equation is obtained.

$${L_3}^2 = {L_2}^2 + c^2 - 2L_2 c\cos\alpha_2 \quad (7)$$

$$\alpha_2 = \cos^{-1}\left(\frac{{L_2}^2 + c^2 - {L_3}^2}{2L_2 c}\right) \quad (8)$$

$$\theta_2 = \alpha_1 + \alpha_2 \quad (9)$$

Using the law of cosines in the triangle BCD, it can be determined that angle 3 is as follows.

$$c^2 = {L_2}^2 + {L_3}^2 - 2L_2L_3\cos\theta_3 \quad (10)$$

$$\theta_3 = \cos^{-1}\left(\frac{{L_2}^2 + {L_3}^2 - c^2}{2L_2L_3}\right) \quad (11)$$

As for determining the size of the angle 4, it is necessary to find the measure of the beta angles 1 and 2. The equations used are as follows.

$${L_2}^2 = {L_3}^2 + c^2 - 2L_2 c\cos\beta_1 \quad (12)$$

$$\beta_1 = \cos^{-1}\left(\frac{{L_3}^2 + c^2 - {L_2}^2}{2L_3 c}\right) \quad (13)$$

$$\beta_2 = \tan^{-1}\left(\frac{b}{z\prime}\right) \quad (14)$$

So the angle value of angle 4 is as follows.

$$\theta_4 = \beta_1 + \beta_2 + 90 \quad (15)$$

The input angle for each servo motor can be determined as follows.

$${\theta_1}' = \theta_1 \quad (16)$$

$${\theta_2}' = \theta_2 \quad (17)$$

$${\theta_3}' = (180 - \theta_3) \quad (18)$$

$${\theta_4}' = 180 - (360 - \theta_4) \quad (19)$$

D. Determining the Inverse Kinematic

Inverse kinematic has an algorithm with kinematic analysis method of robot transformation from Cartesian space to joint space [9]. The inverse

Polynomial equations can be used to find points between n points P1(x1,y1), P2(x2,y2), P3(x3,y3), Pn(xn,yn) by using a polynomial function approximation to the power of n-1 [7]. The polynomial equation in this study serves to obtain the value of the angle (θ) from a known point. So that the robot's motion trajectory (trajectory) can be formed, or it can be called an angular function based on time [15]. The following equation is the general form of a polynomial.

$$\theta_i(t) = \alpha_0 + \alpha_0 t + \alpha_2 t^2 + \alpha_3 t^3 \ldots + \alpha_n t^n \quad (20)$$

The algorithm for calculating polynomial equations is as follows.

1) Describe the number of n-point and the iteration of the factor.
2) Insert of known points to the θi = (θi, for i = 1, 2, 3,…, n).
3) Describe the coefficient of a polynomial function based on the function of the known points.
4) Insert the coefficient of a polynomial function.
5) Calculating θ(t) from the polynomials function which had been generated.
6) Showing (θ1, θ2, θ3, θ4) from each time.

E. Determining the Inverse Kinematic

The hardware design for controlling the robot uses a microcontroller-based on Arduino Shield ATMega 2560, servo motors as a driving force for the robot arm, a power supply as the electric power, an Adapter Step Down Converter Module as a voltage reducer from the power supply to the servo motor, and PC as interface. The input programs are entered from a PC, then processed by the microcontroller to drive each servo motor.

The software design is arranged and entered in the Arduino as its programming language. The program which entered in

Arduino is the result of polynomials equations by the angles of each joint robot.

Designing mechanic involves how to describe the placement of robot components so the robot movement as planned. Then, if the designs have been cleared, the robot components can be assembled.

III. Experiment Results and Analysis

A. *Kinematic Calculation of Robot Arm*

The kinematic calculation using the inverse kinematic method on the movement of the robot arm aims to determine the value of each angle at the joint robot. In this study, a 4 DOF arm robot was tested using the inverse kinematic method designed in Part II to move the test object from point A to point B. Before calculating the kinematic inverse, it is necessary to determine the point to be addressed, namely point A and point B. Then, between these two points, another reference point is given to form a robot's path of motion (trajectory). In this case, to calculate the kinematics of the 4 DOF arm robot, it can be determined as follows.

1) *Determine A, B, and Reference Points*

It should be noted that determining the point is based on the provisions of the robot arm design made previously. This study uses 4 and 6 reference points for each polynomial equation later. In the reference point, it is also necessary to determine the iteration factor in the time variable. The iteration factor in the time function used in this study is 1 of t = 0 to 10. So that the points A, B, and their respective references (in meters) can be written as follows.

a) The starting point (t=0) or point A is (x = 0.28 ; y = 0 ; z = 0.12);

b) Point (t=2) (x = 0.25 ; y = 0.03 ; z = 0.15);

c) Point (t=4) (x = 0.22 ; y = 0.06 ; z = 0.19);

d) Point (t=6) (x = 0.19 ; y = 0.10 ; z = 0.17);

e) Point (t=8) (x = 0.18 ; y = 0.14 ; z = 0.13);

f) Endpoint (t=10) or Point B (x = 0.16 ; y = 0.16 ; z = 0.12).

Based on this point, it is known that the distance from point A to point B if a straight line is drawn is 20 cm. These points are measured by ruler on the robot’s pad.

2) *Calculating Inverse Kinematic*

The results of the inverse kinematic calculation defined by the reference point as the time variable based on (1) to (19) can be seen in Table I.

TABLE I. Calculation Result of Invers Kinematik

t	θ_1’	θ_2’	θ_3’	θ_4’
0	0°	39.4863°	105.746°	66.2595°
2	6.8427°	55.7271°	119.819°	64.0921°
4	15.2551°	76.3079°	120.501°	44.1932°
6	27.7585°	72.3799°	135.062°	62.6821°
8	37.875°	48.8738°	137.997°	89.1232°
10	45°	42.9832°	140.298°	97.3148°

3) *Calculating Polynomial Equations*

Calculate the inverse kinematic of each joint angle value for t at the reference point. The data from the joint angle value will be entered into the polynomial equation to get another angle value of t (0-10), with the iteration factor being 1. This study uses two types of polynomial equations: 3rd order and 5th order polynomials. The results of the calculation of the polynomial equation are as follows.

a) *3rd Orde Polinomial*

To calculate the 3rd order polynomial, input four reference points into the Equation. The four reference points taken are t=0, t=4, t=6, and t=10. Based on (20), the calculation results for the 3rd order polynomial can be seen in Table II as follows.

b) *5th Orde Polinomial*

To calculate the 5th order polynomial, input six reference points into the Equation. The 6 reference points taken are at t=0, t=2, t=4, t=6, t=8 and t=10. Based on (20), the calculation results for the 5th order polynomial can be seen in Table III as follows.

TABLE II. Calculation Result of 3rd Order Polynomial

t	θ_1	θ_2	θ_3	θ_4
0	0°	39.4863°	105.746°	66.2595°
1	1.5°	55.7225°	105.248°	49.5256°
2	4.83448°	66.8859°	108.179°	41.293°
3	9.56551°	73.555°	113.582°	40.0271°
4	15.2551°	76.3079°	120.501°	44.1932°
5	21.4655°	75.7238°	127.98°	52.2562°
6	27.7585°	72.3799°	135.062°	62.6821°
7	33.6966°	66.8562°	140.791°	73.9353°
8	38.8415°	59.7298°	144.211°	84.4816°
9	42.7554°	51.5797°	144.366°	92.7863°
10	45°	42.9832°	140.298°	97.3148°

TABLE III. Calculation Result of 5th Order Polynomial

t	θ_1	θ_2	θ_3	θ_4
0	0°	39.4863°	105.746°	66.2595°
1	4.14092°	43.9002°	120.663°	76.7632°
2	6.8427°	55.7271°	119.819°	64.0921°
3	10.3099°	68.0875°	117.847°	49.7586°
4	15.2551°	76.3079°	120.501°	44.1932°
5	21.3639°	77.8732°	127.45°	49.5753°
6	27.7585°	72.3799°	135.062°	62.6821°
7	33.4657°	61.4876°	139.198°	77.7062°
8	37.875°	48.8738°	137.997°	89.1232°
9	41.2156°	40.1825°	134.674°	94.4832°
10	45°	42.9832°	140.298°	97.3148°

B. *Realization of Robot Arm Motion*

Real plant testing is carried out in three stages, namely testing reference points based on the results of calculations using the inverse kinematic method, and testing the accuracy of the point where the test object is dropped, testing the travel time of the robot's motion by entering polynomial equations at each corner in time t. Then, from the travel time, the robot's speed can be calculated. To test the smooth movement of the robot is based on the graph of the angle value generated from the polynomial equation. The results of the real plant test of the 4 DOF arm robot are as follows.

1) *Reference Point Test*

Based on the data obtained in Table I, the Inverse Kinematic Calculation Results were tested for the accuracy of the resulting points. The results of the reference point testing can be seen in Table IV as follows. In Table IV, it can be seen that the errors that occur at the reference point, from the smallest to the largest, are 0% and 16.667%, respectively. This error can occur because of several things, including the 3D Printer printouts that there are parts of the robot that are not the right size according to the design, the servo motor installation process is not right at 0°, the electrical voltage is not stable on each servo motor, and the load a robot that a servo motor can’t restrain.

TABLE IV. PERCENTAGE ERROR VALUE AT REFERENCE POINTS

t	Calculation Result (m)			Test Result (m)			Error (%)		
	x	y	z	x	y	z	x	y	z
0	0.28	0	0.12	0.28	0	0.117	0	0	2.5
2	0.25	0.03	0.15	0.25	0.03	0.145	0	0	3.333
4	0.22	0.06	0.19	0.207	0.07	0.19	5.909	16.667	0
6	0.19	0.10	0.17	0.174	0.11	0.165	8.421	10	2.941
8	0.18	0.14	0.13	0.18	0.131	0.132	0	6.428	6.154
10	0.16	0.16	0.12	0.16	0.16	0.117	0	0	2.5

In this case, it is also necessary to pay attention to the calculation of the robot's load and the electrical circuit to run properly and not damage the servo motor. Then it is also necessary to pay attention to the normalization process or make the servo motor angle at 0°. So that when entering the angle, it is right at the desired point. In contrast to research [5], the movement of the 4 DOF arm robot has the smallest average error value at 0.5 cm coordinates and the largest error value at 3 cm coordinates. The error value tends to be large compared to this study.

2) *Movement Robot Speed Testing*

The real plant test carried out on the polynomial equation is by inserting the polynomial equation at each corner in time t, which will then calculate the travel time of the robot motion to obtain the robot's speed. The recorded time and the resulting speed based on the motion of the robot with a delay on Arduino of 200 milliseconds are as follows.

TABLE V. TIME AND SPEED OF ROBOT MOVEMENT

Test	3rd Order (second)	5th Order (second)
1	4.34	4.66
2	5.38	4.53
3	5.19	4.73
Average	**4.97**	**4.64**
Average Speed (m/s)	**0.0402**	**0.0431**

In testing the motion of 3rd order and 5th order polynomials, it is known that the average travel time needed to move as far as 20 cm or 0.2 m from point A (t = 0) to point B (t = 10) with a delay on Arduino of 200 milliseconds respectively are 4.97 seconds and 4.64 seconds. As for the travel time, it can be seen that the speeds of the 3rd order and 5th order polynomials are 0.0402 m/s and 0.0431 m/s, respectively. So it is known that 5th order polynomials are faster than 3rd order polynomials. In calculating this speed, it is very influential on the delay value entered on the Arduino. So the travel time and speed will change depending on the delay used.

3) *Robot Motion Accuracy Test*

The accuracy, in this case, is how the comparison between 3rd order polynomials and 5th order polynomials in placing the test object from the initial position or point A (t=0) (x = 0.28 and y = 0) to the specified position is at the endpoint or point B (t=10) (x = 0.16 and y = 0.16). The test results are as follows.

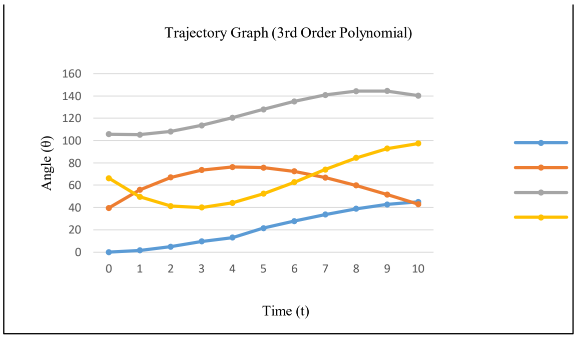
TABLE VI. THE PRECISION OF OBJECT’S POSITION FROM ROBOT MOVEMENT

Test	3rd Order		5th Order	
	x (m)	y (m)	x (m)	y (m)
1	0.164	0.159	0.167	0.159
2	0.167	0.158	0.167	0.154
3	0.164	0.159	0.174	0.153
Average (m)	**0.165**	**0.159**	**0.169**	**0.155**
Error (%)	**3.125**	**0.625**	**5.625**	**3.125**

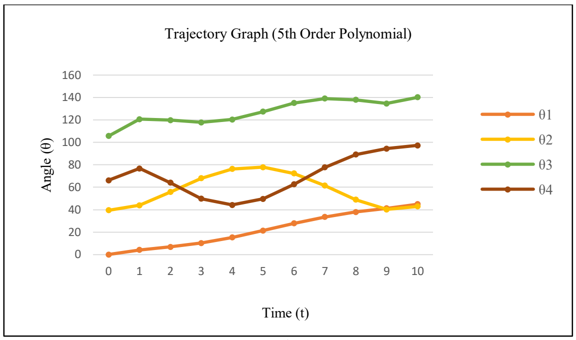
As for testing the accuracy of the robot's motion, it can be seen in Table VI, where the comparison between 3rd order and 5th order of polynomials in the test object from the initial position or point A (t = 0) (x = 0.28 and y = 0) to the specified position namely at the endpoint or point B (t = 10) (x = 0.16 and y = 0.16) with tests carried out three times where it is recorded that for the 3rd order polynomial the average position of objects, namely (x = 0.165 and y = 0.159) and for a 5th order polynomial the average object position is (x = 0.169 and y = 0.155). It can be seen here that neither of them is right at the position determined according to the calculation (x = 0.16; y = 0.16). There are errors in order three and order five polynomials are 3.125% and 5.625% on the x-axis, respectively, while 0.625% and 3.125% on the y-axis. This can happen because of the same thing as what happened at the previous reference point. Then it can also be generated from the use of the wrong type of servo motor. In the study [2], the error percentage for the timeliness of the largest robot motion for the X-axis is 3%, and for the Y-axis is 68.8%. This shows that the error value tends to be very large compared to this study.

4) *Robot Smooth Motion Test*

The smooth motion of the robot on the polynomial equation can be seen from the trajectory graph of the polynomial of each angle compared to the time (t) obtained from Fig. 6.



(a)



(b)

Fig. 6. Trajectory Graph (a) 3rd Order Polynomial (b) 5th Order Polynomial

Fig. 6 shows that the 3rd order polynomial has a more regular graph of reducing the angle over time than the 5th order polynomial, which tends to be unstable. So from this graph, it can be seen that the 3rd order polynomial has a smoother motion when compared to the 5th order polynomial. In this study, in every robot movement, it is necessary to avoid significant changes in the angle so that there is no erratic movement. However, it goes back to how the location of the reference point is also which can have a big impact because this polynomial equation will offer changes of each value that will remain on track. So that if each path that has been determined will be close to a value far enough away, it will cause unstable conditions.

IV. Conclusion

The robot test results produce motion that has an error value resulting from each x, y, and z position from its reference point, from the smallest to the largest, respectively, 0% and 16.667%. Errors for the accuracy of the robot's motion in moving the test object on the 3rd order and 5th order polynomials are also seen, namely 3.125% and 5.625% on the x-axis, while 0.625% and 3.125% on the y-axis. Then the average speed produced is 0.0402 m/s and 0.0431 m/s. Judging from the points generated from the two polynomial equations, they form a trajectory curve that goes up and down with changes in the angle value that are not too significant from each point. However, 3rd order polynomial has a more regular graph of increasing or decreasing the angle at any time than 5th order polynomial, so it has a smoother movement. In this case, based on these three indicators, the performance of the 3rd order polynomial is better than 5th order polynomial.

References

[1] M. Didi, E. D. Marindani, and A. Elbani, “Rancang Bangun Pengendalian Robot Lengan 4 DOF dengan GUI (Graphical User Interface) Berbasis Arduino Uno,” J. Tek. Elektro, vol. 2, no. 3, pp. 1–11, 2015, [Online]. Available: https://jurnal.untan.ac.id/index.php/jteuntan/article/view/14580.

[2] F. Rokhman Iskandar, I. Sucahyo, M. Yantidewi, P. Studi Fisika, and U. Negeri Surabaya, “Penerapan Metode Invers kinematik Pada Kontrol Gerak Robot Lengan Tiga Derajat Bebas,” J. Inov. Fis. Indones. (IFI, vol. 09, pp. 64–71, 2020.

[3] I. Farozi, R. Maulana, W. Kurniawan, and G. E. Setyawan, “Implementasi Robot Lengan Pemindah Barang 3 DOF Menggunakan Implementasi Robot Lengan Pemindah Barang 3 DOF Menggunakan Metode Inverse Kinematics,” J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya, vol. 2, no. April, pp. 2810–2816, 2018.

[4] P. H. Setiawan, “Analisis Inverse Kinematics Tersegmentasi Berbasis Geometris Pada Robot Humanoid Saat Walking Gait System for Humanoid Robot,” Institut Teknologi Sepuluh Nopember, 2016.

[5] F. Faris, A. Triwiyatno, and I. Setiawan, “Arm Manipulator 4 Dof Dengan Cartesian Space-Trajectory Planning,” Transient, vol. 1, no. 4, pp. 151–158, 2012.

[6] I. A. Kurniawan, Feriyonika, and S. Pramono, “Inverse dan Body Kinematics pada Robot Hexapod,” Ind. Res. Work. Natl. Semin., vol. 9, pp. 115–123, 2018.

[7] D. Levy, “Introduction to Numerical Analysis,” CSCAMM, University of Maryland. 2010, doi: 10.1080/00401706.1966.10490427.

[8] G. Bao, S. Liu, and H. Zhao, “Kinematics Simulation of 4 DOF Manipulator,” vol. 123, no. Msmee, pp. 400–408, 2017, doi: 10.2991/msmee-17.2017.81.

[9] J. W. Kusuma, P. Shinta, and H. Dedy, “Pergerakan Kaki Pada Robot Hexapod,” Robotics, p. 10, 2015.

[10] E. K. Frisyras, V. C. Moulianitis, and N. A. Aspragathos, “ANNs to approximate all the inverse kinematic solutions of non-cuspidal manipulators,” IFAC-PapersOnLine, vol. 51, no. 22, pp. 418–423, 2018, doi: 10.1016/j.ifacol.2018.11.586.

[11] N. E. Khomariah and S. Huda, “Udiebot: Implementasi Inverse Kinematic pada Quadruped Robot,” J. Ilmu Komput. dan Desain Komun. Vis., vol. 3, no. 1, pp. 12–21, 2018.

[12] S. Gómez et al., “Design of a 4-Dof Robot Manipulator with Optimized Algorithm for Inverse Kinematics Design of a 4-DOF Robot Manipulator with Optimized Algorithm for Inverse Kinematics,” Int. J. Mech. Mechatronics Eng., vol. 9, no. 6, pp. 929–934, 2016.

[13] I. D. Nugraha, “Pendekatan Geometri untuk Perhitungan Inverse Kinematics Gerakan Lengan Robot 4 Derajat Kebebasan,” J. Tek. Mesin ITI, vol. 5, no. 1, p. 1, 2021, doi: 10.31543/jtm.v5i1.572.

[14] S. Setiawan, Firdaus, B. Rahmadya, and Derisma, “Penerapan Invers Kinematika Untuk Pergerakan Kaki Robot Biped,” in Seminar Nasional Sains dan Teknologi, 2015, no. November, pp. 1–9.

[15] H. R. Nurhakim, “Sistem Kendali Gerak Continuous Path Tracking dengan Menggunakan Cubic Trayektori Planning pada Robot Manipulator 4 DOF,” Universitas Indonesia, 2010.