

Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics

Adnan Rafi Al Tahtawi ^{1*}, Muhammad Agni ², Trisiani Dewi Hendrawati ³

¹ Department of Electrical Engineering, Politeknik Negeri Bandung, Bandung, Indonesia

^{2,3} Department of Computer Engineering, Politeknik Sukabumi, Sukabumi, Indonesia

Email: ¹ adnan.raf@polban.ac.id, ² agniid@live.com, ³ trisianidewi@polteksmi.ac.id

*Corresponding Author

Abstract— Robot arm is often used in industry with various tasks, one of which is a pick and place. Rapid prototyping of robot arms is needed to facilitate the development of many industrial tasks, especially on a laboratory scale. This study aims to design a small-scale three degree of freedom (3-DoF) robot arm for pick and place mission using the inverse kinematics method. The mechanical robotic arm system is designed using Solidworks with four servo motors as the actuator. Arduino Mega 2560 is used as a microcontroller in which the inverse kinematic method is embedded. This method is used to move the robot based on the coordinates of the destination pick and place. The test results show that the robot arm can carry out the pick and place mission according to the target coordinates given with the largest average error of about 5 cm. While the error generated between the calculation and computation results is around 3%.

Keywords—robot arm, small-scale, pick and place, inverse kinematics

I. INTRODUCTION

The robot arm is a type of robot that is used for various industrial processes. The robot can move the position from one point to another by determining the position of the coordinates. This robot control configuration can be done manually or automatically and can be used statically or mobile [1]. Inverse kinematic is an algorithm that is often used in the operation of a robot arm. This method allows the conversion of coordinates into the angular motion of each joint robot. In solving the inverse kinematic problem, several methods can be used such as Denavit-Hartenberg (D-H) [2], screw theory, and iterative method [3]. Besides, soft computing methods can also be used to solve inverse kinematic equations [4], [5]. In essence, some of these methods are used to find the best solution for calculating joint angles based on target coordinates. Several researchers have discussed the application of inverse kinematic in the robot arm. One of them is research [6] which uses this method to control the position of the robot arm movement when carrying out a pick and place mission. Besides, other researchers also use this method for conveyor systems [7], sorting objects by shape [8], integration with computer vision [9], and rapid prototyping [10]. However, to facilitate the development of robot arms, especially in studying the kinematics of their movements, a small-scale prototype of a robot arm is needed.

Several researchers have also developed a low-cost and small-scale prototype robot arm. Research [11] discussed the fabrication of light robot arm parts which was carried out for 6 to 10 hours. Each part of the robot is designed using 3D simulation and realized in hardware using 3D printing. The result is that the robot arm has the criteria for raw material, metallic reinforcements, accessibility, and flexible couplings. In the study [12], a low-cost robot arm with a counterbalance mechanism was developed that can balance the robot while performing services. The proposed mechanism can effectively reduce the movement torque of the robot so that a relatively cheap price motor part can be used. Reference [13] proposes a flexible robot arm for pick and place that uses two DC motors and a silicon bar as a flexible link. The test results show that the robot can hold more than two times its weight. Several studies have also designed a low-cost robot arm with recognition features. As in research [14] which applies the voice recognition method to control the robot arm. Also, other recognition methods such as visual tracking [15]-[18] and gesture control [19]-[22] have been integrated into the robot arm. Currently, robot arm research has developed along with the emergence of the Internet of Things (IoT) technology. As in research [23]-[25] that have developed an internet robot arm that allows robot operation through internet connectivity. This system is very much needed considering the development of IoT technology, especially in the industrial sector. Finally, the robot arm has also been integrated with Unmanned Aerial Vehicle (UAV) technology known as Unmanned Aerial Manipulator (UAM), as has been developed in research [26].

The purpose of this research is to design a small-scale robot arm for learning needs in the laboratory. The main contribution of this paper is to provide a simple design for a robot arm that can be used to study the behavior and tasks of a robot on a small scale. The robot arm was designed using Solidworks software and then realized using 3D printing. The robot is designed using low-cost parts, such as MG servo motors and Arduino microcontrollers [27]. In this study, the robot is then given an automatic pick and place mission based on inverse kinematics [28]-[30]. This research is a continuation of the paper [31] which discusses the design of robotic mechanical and electrical systems. In this paper, the inverse kinematics method is applied to the robot and tested for automatic pick and place missions.



II. RESEARCH METHOD

A. System Design

The block diagram of the system design can be seen in Fig. 1. This system consists of an input section, a control section, and an output section. In the input section, there is the MATLAB GUI and Arduino IDE which functions to determine the coordinates of the robot arm movement. The coordinates used are picked coordinates and place coordinates. The robot arm control part is Arduino Mega 2560 with an inverse kinematic method embedded in it. At the output, there is a 180-degree servo motor as an actuator. Table 1 shows the technical specifications of the designed robot arm.

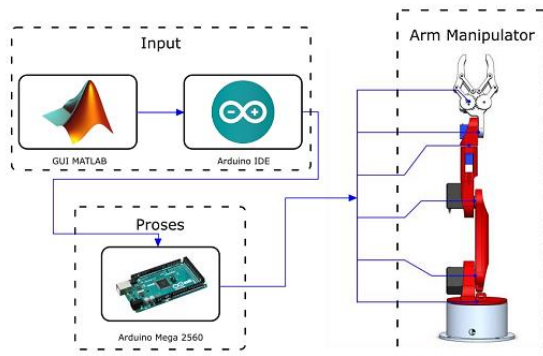


Fig. 1. Robot arm system design

TABLE I. ROBOT SPECIFICATION

Specification	Type/Value
Microcontroller	Arduino Mega 2560 ATMega 2560, operating voltage 5V, 54 digital I/O pins, 16 analog input pin, clock speed 16 MHz
Servo Motor	MG996R (base and shoulder) -Torque 9.4 Kgf.cm (4.8V) -Torque 11 Kgf.cm (6V) -Operating voltage 4.8-7.2V SG90 (elbow and gripper) -Torque 1.2 Kgf.cm (4.8V) -Torque 1.6 Kgf.cm (6V) -Operating voltage 3-7.2V
Joint	4 (base, shoulder, elbow, gripper)
Power Supply	12 V
Height (vertical position)	45.94 cm
Weight	0.418 kg
Reachability	35 cm, 0-180 degrees

B. Hardware Design

The hardware design of this robot arm includes mechanical and electronic parts. The mechanical robotic arm system is designed with three degrees of freedom. The robot arm configuration used is articulated with three joint revolute and one end-effector. The mechanical system of the robot arm is very influential and dominates the system because the shape and movement of the mechanics will affect the mission being carried out. The material used in the manufacture of this robot is a 3D printing PLA (Polylactic Acid) filament which is made from corn starch essence. The

material used has an Infill Density of 5%, meaning that the fill thickness of the manufacturing material is bigger and will be lighter than 100% Infill Density which has a very tight fill thickness or no holes at all. By using a lower infill, the joint motion system on the servo motor actuators will be lighter and will not work stronger because the difference in weight of the part link affects the motion performance of the servo motor. Fig. 2 presents the overall shape of the three-dimensional model designed using Solidworks 3D modeling software. There are four joint and link structures of the 3 DOF robot arm, namely the base, waist, elbow, and gripper. Each joint movement is controlled by a servo motor. The MG996R type servo motor is placed on the joint base and waist, while the SG90 type is placed on the elbow and gripper joint. Figure 3 shows the schematic of the robot arm being designed. Four servo motor units are connected to the microcontroller, where three servo motors act as driving the joint and one servo drives the gripper. The four servo motors are controlled using the `servo.write(degrees)` command, where degrees are the angular data generated by inverse kinematics calculations. The circuit also adds a 9 V power supply for the servo motor, a 5 V voltage regulator for the microcontroller, and several LEDs as indicators.

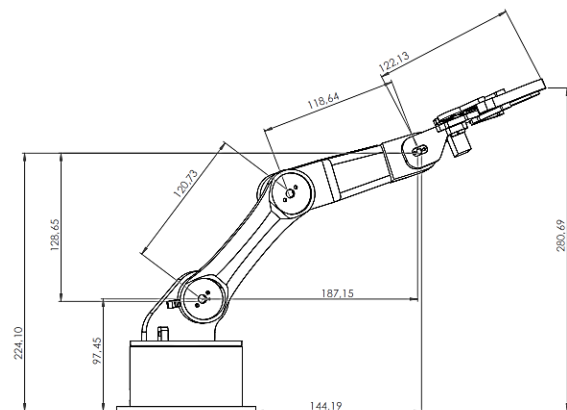


Fig. 2. Robot hardware design

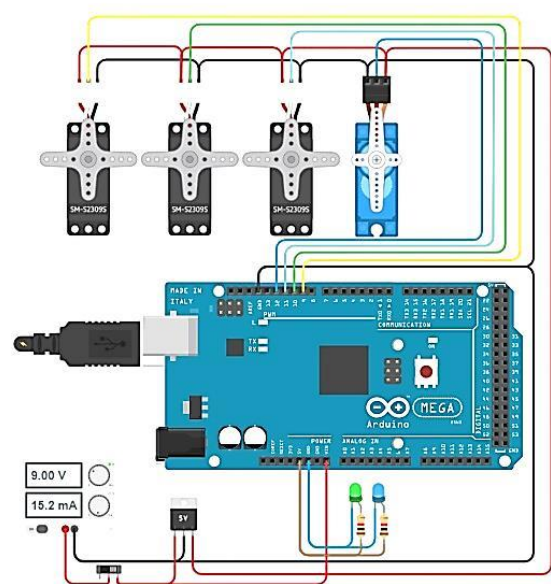


Fig. 3. Electrical schematic design

C. Software Design

The designed system is equipped with software as an interface using the MATLAB GUI. The interface is designed to function as a place to provide the pick and place coordinates of the work piece. The designed interface is shown in Fig. 4. The coordinates given on the interface will be sent via serial communication to the Arduino Mega 2560. These coordinates are then processed using the inverse kinematic method so that the angle value for each servo motor is generated. The angle value is then sent to each motor so that it can move the robot to the pick and place coordinates.

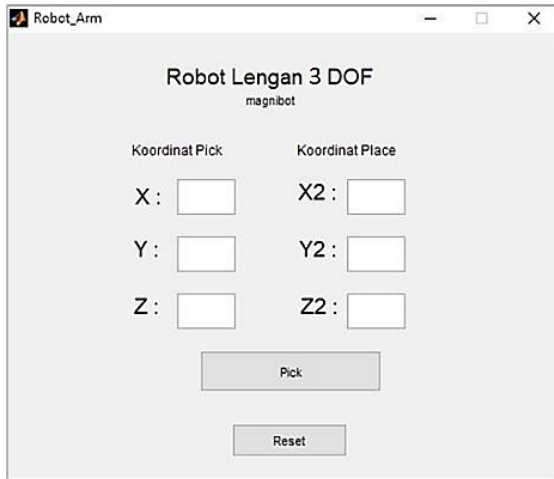


Fig. 4. Robot control interface

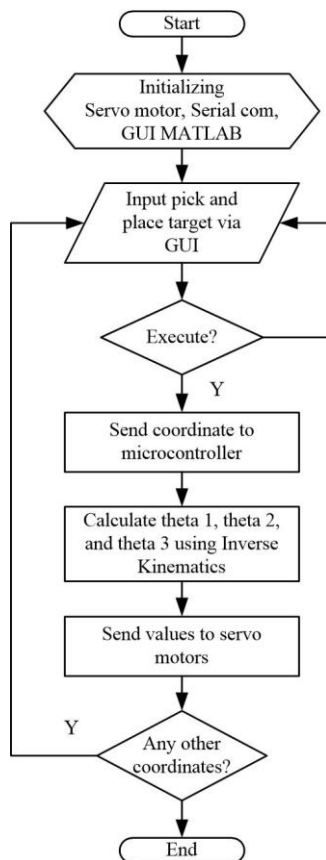


Fig. 5. Flowchart of the robot task

The flowchart of the robot arm during the operation can be seen in Fig. 5. Generally, the way this system works starts from the input of the pick and place coordinates given through the MATLAB GUI. The coordinates are then sent to Arduino Mega 2560 and then processed with inverse kinematics calculations. The result of the calculation is the value of the degree of angle for the joint base, shoulder, and elbow on the robot. After the angle degree value is obtained, the value is sent to each servo motor at each joint of the robot arm. Thus, the robot arm end effector can go to the target point and the robot arm gripper can pick up and place objects at the intended coordinates.

D. Inverse Kinematics

Inverse kinematics is a method used to find corner variables for joint robots based on the robot's position coordinates. The inverse kinematics method will look for the parameter values that must be given to each actuator to achieve the final destination coordinates. The block diagram of inverse kinematics is shown in Fig. 6. In determining the final coordinates of the end-effector, inverse kinematics must be adjusted to the workspace boundary of the robot's reach. Solving the inverse kinematics can be done using Pythagoras law and trigonometric rules. Inverse kinematics can be solved by looking at the two sides of the robot, namely the top view and the side view. The top side is used to find the degree angle θ_1 of the base joint and the joint with the vertical rotating axis. The sides are used to find the degrees of angle θ_2 of the joint shoulder and the degrees of angle θ_3 of the joint elbow. For the degree of angle on the gripper, it is enough to use the ON-OFF algorithm. The illustration of the top and side diagram of the robot can be seen in Fig. 7 and Fig. 8.

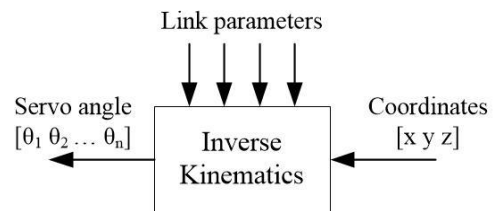


Fig. 6. Inverse kinematics mechanism

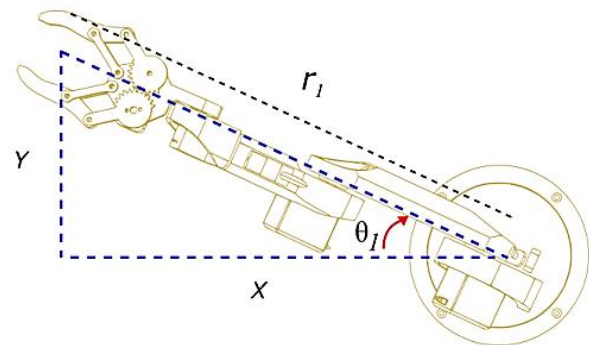


Fig. 7. Robot top view

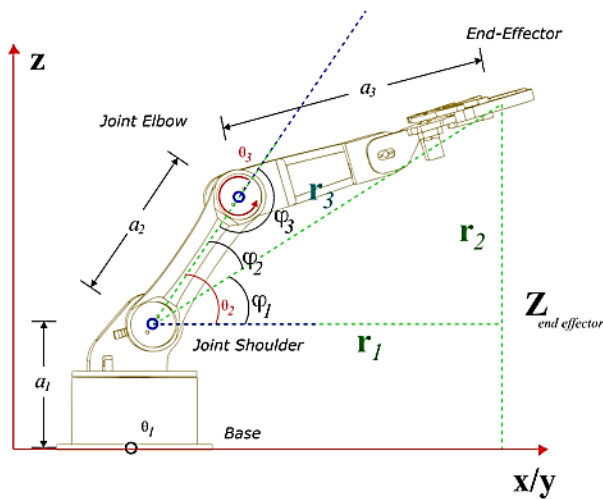


Fig. 8. Robot side view

To determine the degree of the axis of the joint base, the following tangent equation can be used

$$\theta_1 = \tan^{-1} \left[\frac{Y_{ef}}{X_{ef}} \right] \quad (1)$$

where X_{ef} is the position of the end-effector coordinates on the X-axis, Y_{ef} is the position of the end-effector coordinates on the Y-axis, and θ_1 is the degree of the angle of the joint base.

To find the degree of angle θ_2 , a triangle guideline is needed which describes the lengths r_1 , r_2 , and r_3 as shown on the dashed green line in Fig. 7. Thus, r_1 , r_2 , and r_3 can be calculated based on the following equation.

$$r_1 = \sqrt{X_{ef}^2 + Y_{ef}^2} \quad (2)$$

$$r_2 = Z_{ef} - a_1 \quad (3)$$

$$r_3 = \sqrt{r_2^2 + r_1^2} \quad (4)$$

where Z_{ef} is the position of the end-effector coordinates on the Z-axis and a_1 is the link length between base and shoulder.

Furthermore, by looking at the side of the 3-DOF robot arm structure, it can be found the value of the angle degrees θ_2 and θ_3 . From the side, several variables can be defined including a_1 , a_2 , a_3 , Z_{ef} , θ_2 , and θ_3 . a_2 is the link length between the shoulder and elbow. a_3 is the link length between the elbow and the end-effector. θ_2 is formed from the horizontal line axis of the joint shoulder and the link between the shoulder and elbow. θ_3 is formed from the horizontal line axis of the joint shoulder and elbow.

Referring to Fig. 7, θ_2 can be calculated by the following equation.

$$\theta_2 = \varphi_1 + \varphi_2 \quad (5)$$

where

$$\varphi_1 = \tan^{-1} \left[\frac{r_2}{r_1} \right] \quad (6)$$

$$\varphi_2 = \cos^{-1} \left[\frac{(a_2)^2 + (r_3)^2 - (a_3)^2}{2a_2r_3} \right] \quad (7)$$

Meanwhile, θ_3 can be calculated using the following equation.

$$\theta_3 = 180^\circ - \varphi_3 \quad (8)$$

where

$$\varphi_3 = \cos^{-1} \left[\frac{(a_2)^2 + (a_3)^2 - (r_3)^2}{2a_2a_3} \right] \quad (9)$$

The inverse kinematics equation is then implemented using programming language that embedded in Arduino board microcontroller. Below shows a list of inverse kinematics programs in C language on Arduino IDE compiler.

```
void inverse() {
//calculate thetal
if (x>-1 ) {
thetal = atan(y/x);
thetal = thetal*180/PI;
}
else {
minusx = -x;
thetal = atan(y/minusx);
thetal = thetal*180/PI;
thetal = 180-thetal;
}

//calculate r1,r2,r3
r1 = sqrt((sq(x))+(sq(y)));
r2 = z-a1;
r3 = sqrt((sq(r2))+(sq(r1)));

//calculate pi1,pi2,pi3
pi1 = atan(r2/r1)*180/PI;
pi2 = acos(((sq(a2))+(sq(r3))-(sq(a3)))/(2*a2*r3))*180/PI;
pi3 = acos(((sq(a2))+(sq(a3))-(sq(r3)))/(2*a2*a3))*180/PI;

//calculate theta2,theta3,theta4
theta2 = pi1+pi2;
theta3 = 180-pi3;
}
```

III. RESULT AND DISCUSSION

The overall system design is then implemented according to the system block diagram as shown in Fig. 9. The system that has been created is then tested to determine its performance. Tests were carried out on servo motors and pick and place missions based on inverse kinematics.

A. Servo Motor Testing

Servo motor testing is carried out to determine whether the motor is functioning properly or not. The test is carried out on each servo motor at each joint. Servo motor testing is done by measuring the voltage when the servo motor is given a PWM signal and when the servo motor is not given a PWM signal. No-load voltage (V_{NL}) is the voltage when the servo motor is not given a PWM signal. Full load voltage (V_{FL}) is the voltage when the servo motor is given a PWM signal which will make the servo motor have a load. The servo motor test results can be seen in Table 2.

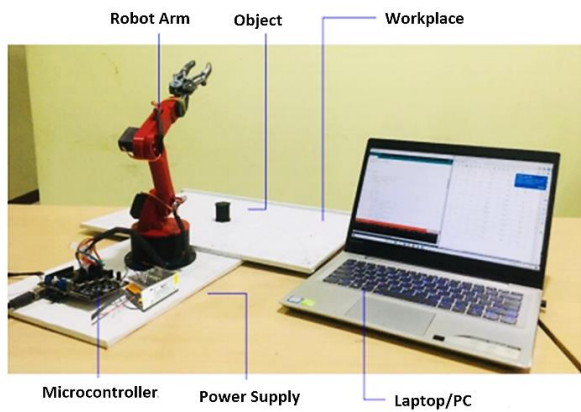


Fig. 9. System implementation

TABLE II. SERVO MOTOR TESTING

No	Joint	V _{NL} (V)	V _{FL} (V)
1	Base	5.07	4.99
2	Shoulder	5.06	4.98
3	Elbow	5.04	4.93
4	Gripper	5.00	4.73
Average		5.04	4.90

From the results, it can be seen that the V_{NL} and the V_{FL} have an average difference of 0.08 V. This is because there is power dissipation when the servo motor operates. The existence of power dissipation will affect the position of the end-effector inaccurately.

B. Pick and Place Coordinate Testing

This test is carried out to determine the coordinates of the end-effector of the robot arm to reach the destination coordinates. The test is done by dividing the workspace into quadrant 1 for the negative end-effector X position and quadrant 2 for the positive end-effector X position. Measurements are made using a coordinate board and the aid of a ruler to find out the difference between the targets and the results achieved. The results of the pick coordinate test can be seen in Fig. 10, while the place coordinate test can be seen in Fig. 11.

Based on the results of the pick coordinate test, it can be seen that the robot managed to get to the coordinates according to the given target even though there was an error. The largest error average value occurs on the Y-axis, which is 3.5 cm, while the smallest error value is obtained on the X-axis, which is 1.74 cm. When viewed from the test, the largest average error occurs in the third test with a coordinate error of 5.4 cm, while the smallest average error occurs in the fifth test with a value of 1.27 cm.

In testing the place coordinates, it can also be seen that the robot managed to get to the coordinates according to the target even though there was an error. The largest error average value occurs on the X-axis, which is 2.68 cm, while the smallest error value is obtained on the Z-axis, which is 1.78 cm. When viewed from the test, the largest average error occurs in the first test with a coordinate error of 2.87 cm, while the smallest average error occurs in the third test with a value of 1.93 cm.

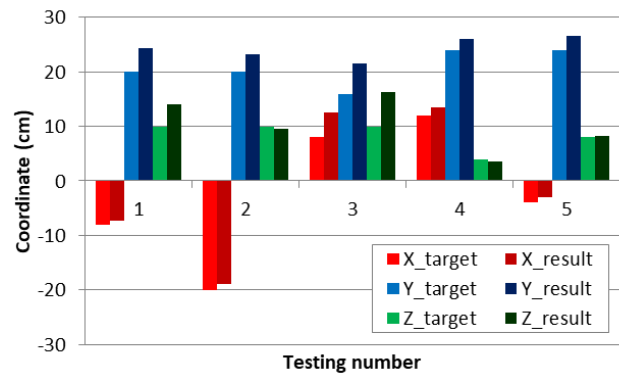


Fig. 10. Pick coordinate testing result

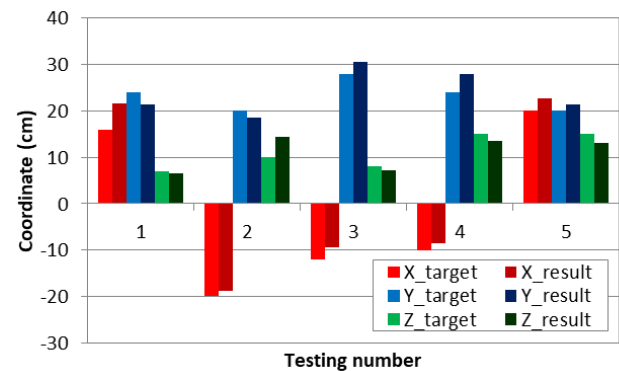


Fig. 11. Place coordinate testing result

C. Servo Angle Testing Based Inverse Kinematics

Servo angle testing aims to determine the angle value generated by the results of microcontroller computation. The resulting value is then compared with the results of calculations manually using Ms. Excel. The test is carried out by providing five test coordinates for the position (x, y, z) in the pick and place conditions. The test coordinates in the pick condition are (-8,20,10), (-20,20,10), (8,16,10), (12,24,4), and (-4,24,8). The test coordinates in place conditions are (16,24,7), (-20,20,10), (-12,28,8), (-20,24,15), and (20,20,15). Then the results of the computation of θ_1 , θ_2 , and θ_3 angles are displayed on the Arduino IDE Serial Monitor. We also compare the servo angles resulting by computational with Arduino microcontroller and the calculation based on inverse kinematics equations. The error is calculated using the error equation as follows.

$$Error (\%) = \frac{Abs(\theta_{Cal} - \theta_{Comp})}{\theta_{Cal}} \times 100\% \quad (10)$$

TABLE III. SERVO ANGLE ON PICK MOVEMENT

No.	θ_1 (degree)		θ_2 (degree)		θ_3 (degree)	
	Cal.	Comp.	Cal.	Comp.	Cal.	Comp.
	111.8	111.8	78.7	80.1	197.9	192.0
2	135.0	135.0	49.3	48.8	160.8	152.7
3	63.43	63.4	95.7	97.5	215.5	209.6
4	63.43	63.4	41.3	44.3	167.2	163.9
5	99.4	99.4	62.0	60.0	183.8	178.7
E	0.00 %		3.03%		3.10%	

TABLE IV. SERVO ANGLE ON PLACE MOVEMENT

No.	θ_1 (degree)		θ_2 (degree)		θ_3 (degree)	
	Cal.	Comp.	Cal.	Comp.	Cal.	Comp.
1	56.3	56.3	40.4	41.2	156.8	150.5
2	135.0	135.0	79.3	81.2	170.9	179.0
3	113.2	113.2	34.5	33.6	135.2	139.4
4	129.8	129.8	29.2	29.1	113.3	110.5
5	45.0	45.0	46.4	41.2	156.8	150.5
E	0.00 %		3.70%		3.67%	

Table 3 and Table 4 show the angles generated by the inverse kinematic for each servo under pick and place conditions. Based on the test data, it is found that there are two angles generated that exceed the maximum limit of 180 degrees, namely the first test of the θ_3 angle of the pick condition and the third test of the θ_3 angle of the place condition. Each of these conditions results in an angle of 192.02° and 209.61° . When this condition occurs, the servo movement will saturate at 180° , which can cause an error in the pick coordinates.

The results show that there is an error between the angle generated by the computation and the calculation of about 3% on θ_2 and θ_3 both when making a pick and place. The θ_1 angle yields the same value between the computation and calculation results. This happens because the θ_1 equation is not that complicated when compared to the θ_2 and θ_3 equations. The error value may be caused by differences in the resolution of the data types processed in the computation process.

This study designed a small-scale robot arm by applying the inverse kinematic method for the pick and place mission. Solving the inverse kinematic solution in this study still uses generic equations. Several methods of solving the inverse kinematic have been developed, such as Denavit-Hartenberg (D-H), Screw Theory, and Iterative Method. Each of these three methods has advantages and disadvantages. As in terms of computation time, the D-H method has the fastest computation time and the Iterative method has the longest computation time [3]. However, the use of this inverse kinematic solving method can be adapted to the specifications and tasks of the robot to be designed.

IV. CONCLUSION

A small-scale robot arm with a pick and place mission based on inverse kinematics has been designed, implemented, and tested successfully. The inverse kinematics method that is applied can make the robot perform a pick and place mission for several coordinate targets. The test results show that the robot has the largest average coordinate error as far as 5.4 cm in the third pick test. The largest average error servo angle between computational and calculation is 3.7%. This happens because of the saturation in the servo motor and the differences in the resolution data type. The next research is to design a closed-loop control algorithm by considering the saturation condition of the actuator.

REFERENCES

- [1] R. Gautam, A. Gedam, A. Zade, and A. Mahawadiwar, "Review on development of industrial robotic arm," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 03, 2017.
- [2] Y. Becerra, M. Arbulu, S. Soto, and F. Martinez, "A comparison among the Denavit-Hartenberg, the screw theory, and the iterative methods to solve inverse kinematics for assistant robot arm," In *International Conference on Swarm Intelligence*, Springer, Cham, 2019, pp. 447-457.
- [3] A. Patil, M. Kulkarni, and A. Aswale, "Analysis of the inverse kinematics for 5 DOF robot arm using DH parameters," In *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2017, pp. 688-693.
- [4] C. Lopez-Franco, J. Hernandez-Barragan, A. Y. Alanis, and N. Arana-Daniel, "A soft computing approach for inverse kinematics of robot manipulators," *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 104-120, 2018.
- [5] G. B. Mahanta, B. B. V. L. Deepak, M. Dileep, B. B. Biswal, and S. K. Pattanayak, "Prediction of inverse kinematics for a 6-DOF industrial robot arm using soft computing techniques," In *Soft computing for problem solving*, Singapore, 2019, pp. 519-530.
- [6] K. M. Myint, Z. M. M. Htun, and H. M. Tun, "Position control method for pick and place robot arm for object sorting system," *International journal of scientific & technology research*, vol. 5, no. 06, pp. 57-61, 2016.
- [7] S. D. Han, S. W. Feng, and J. Yu, "Toward Fast and Optimal Robotic Pick-and-Place on a Moving Conveyor," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 446-453, 2019.
- [8] H. S. Nu'man, Y. Sofyan, and A. R. Al Tahtawi, "Pengendalian Robot Lengan Pemilih Benda Berdasarkan Bentuk Menggunakan Teknologi Computer Vision. In SEMNASTERA (Seminar Nasional Teknologi dan Riset Terapan), vol. 2, 2020, pp. 42-48.
- [9] N. F. Zulfardi, D. I. Saputra, A. D. A. Ahkam, "Aplikasi Deteksi Benda Menggunakan Metode Image Substraction Sebagai Masukan Koordinat Pada Robot Lengan 3 DOF," In *Prosiding Seminar Nasional Teknologi dan Riset Terapan (SEMNASTERA)*, vol. 1, 2019, pp. 30-37.
- [10] I. Ishak, J. Fisher, and P. Larochelle, "Robot arm platform for rapid prototyping: Concept," In *Proceedings in Florida Conference on Recent Advances in Robotics*, 2015.
- [11] S. C. Gutiérrez, R. Zotovic, M. D. Navarro, and M. D. Meseguer, "Design and manufacturing of a prototype of a lightweight robot arm," *Procedia Manufacturing*, vol. 13, pp. 283-290, 2017.
- [12] H. S. Kim and J. B. Song, "Low-cost robot arm with 3-DOF counterbalance mechanism," In *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4183-4188.
- [13] G. Kato, D. Onchi and M. Abarca, "Low cost flexible robot manipulator for pick and place tasks," *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Jeju, 2013, pp. 677-680.
- [14] N. Andrews, S. Jacob, S. M. Thomas, S. Sukumar and R. K. Cherian, "Low-Cost Robotic Arm for differently abled using Voice Recognition," *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019, pp. 735-739.
- [15] Y. Wu, M. Wang and N. M. Mayer, "A new type of eye-on-hand robotic arm system based on a low-cost recognition system," *2017 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, Taipei, 2017, pp. 110-114.
- [16] F. Šuligoj, B. Jerbić, M. Švaco, B. Šekoranja, D. Mihalinec and J. Vidaković, "Medical applicability of a low-cost industrial robot arm guided with an optical tracking system," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 3785-3790.
- [17] O. Kroeger, F. Wollschläger and J. Krüger, "Low-Cost Embedded Vision for Industrial Robots: A Modular End-of-Arm Concept," *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, 2020, pp. 1301-1304.
- [18] T. H. Luu and T. H. Tran, "3D vision for mobile robot manipulator on detecting and tracking target," *2015 15th International Conference on*

- Control, Automation and Systems (ICCAS), Busan, 2015, pp. 1560-1565.
- [19] P. P. Sarker, F. Abedin and F. N. Shimim, "R3Arm: Gesture controlled robotic arm for remote rescue operation," 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 2017, pp. 428-431.
- [20] R. Mardiyanto, M. F. R. Utomo, D. Purwanto and H. Suryoatmojo, "Development of hand gesture recognition sensor based on accelerometer and gyroscope for controlling arm of underwater remotely operated robot," 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, 2017, pp. 329-333.
- [21] L. Aggarwal, V. Gaur and P. Verma, "Design and implementation of a wireless gesture controlled robotic arm with vision," International Journal of Computer Applications, vol. 79, no. 13, pp. 39-43, 2013.
- [22] S. A. Khajone, S. W. Mohod and V. M. Hame, "Implementation of a wireless gesture controlled robotic arm," International Journal of innovative research in computer and communication engineering, vol. 3, no. 1, pp. 375-379, 2015.
- [23] W. M. H. W. Kadir, R. E. Samin, and B. S. K. Ibrahim, "Internet controlled robotic arm," Procedia Engineering, vol. 41, pp. 1065-1071, 2012.
- [24] R. A. Atmoko and D. Yang, "Online Monitoring & Controlling Industrial Arm Robot Using MQTT Protocol," 2018 IEEE International Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics), Bandung, Indonesia, 2018, pp. 12-16.
- [25] P. Siagian and K. Shinoda, "Web based monitoring and control of robotic arm using Raspberry Pi," 2015 International Conference on Science in Information Technology (ICSITech), Yogyakarta, 2015, pp. 192-196.
- [26] G. Zhang, Y. He, B. Dai, F. Gu, J. Han and G. Liu, "Robust Control of an Aerial Manipulator Based on a Variable Inertia Parameters Model," IEEE Transactions on Industrial Electronics, vol. 67, no. 11, pp. 9515-9525, 2020.
- [27] K. Kunal, A. Z. Arfianto, J. E. Poetro, F. Waseel, and R. A. Atmoko, "Accelerometer Implementation as Feedback on 5 Degree of Freedom Arm Robot," Journal of Robotics and Control (JRC), vol. 1, no. 1, pp. 31-34, 2020.
- [28] A. Abdulkareem, O. Ladenegan, A. F. Agbetuyi, and C. O. A. Awosope, "Design and Implementation of a Prototype Remote-Controlled Pick and Place Robot," International Journal of Mechanical Engineering and Technology, vol. 10, no. 2, 2019.
- [29] K. Ghadge, S. More, P. Gaikwad, and S. Chillal, "Robotic arm for pick and place application," International Journal of Mechanical Engineering and Technology, vol. 9, no. 1, pp. 125-133, 2018.
- [30] V. S. Manjula and R. I. Karamagi, "Automatic pick and place robot manipulation using a microcontroller," J Appl Computat Math, vol. 7, no. 2, pp. 408, 2018.
- [31] M. Agni and A. R. Al Tahtawi, "Perancangan Robot Lengan Lima Derajat Kebebasan untuk Misi Pick and Place Berbasis Inverse Kinematics, In SEMNASTERA (Seminar Nasional Teknologi dan Riset Terapan), vol. 2, 2020, pp. 87-92.