ISSN 2355-3286

Modeling and Simulation of Manipulator Robot Using MATLAB

Tesya Permadi[1], Silmi Ath Thahirah Al Azhima[2], Mariya Al Qibtya[3], Nurul Fahmi Arief Hakim[4]
[1, 2, 3, 4]Industrial Automation and Robotic Engineering Education, Indonesia University of Education, Bandung, Indonesia
[1]tesya@upi.edu, [2]silmithahirah@upi.edu, [3]mariyalqibtiya@upi.edu, [4]nurulfahmi@upi.edu

Accepted on 24 May 2024
Approved on 13 June 2024

***Abstract*— In recent decades, the rapid development of robotic technology has increased the demand for efficient and affordable robots, especially in the manufacturing industry. One type of robot that is commonly used is the robotic arm, which is capable of performing complex tasks with high precision and speed. This research focuses on the modeling and simulation of a 4-DOF RRPR robot manipulator using MATLAB software, including the SimScape Multibody Toolbox and Robotic System Toolbox. This study investigates various aspects of robot performance, such as joint angles, end-effector coordinates, and robot dynamics. With an emphasis on simulation, this research aims to accelerate the development of robotic technology and minimize the risks associated with physical implementation in the field. The simulation results provide valuable insights for improving the efficiency, precision, and reliability of robot manipulators in various applications.**

***Index Terms*— MATLAB Simulation; SimScape Multibody Link; Robotic Arm; Robot Manipulator; Robotic System Toolbox.**

I. Introduction

In recent decades, the development of robotic technology has experienced rapid progress. The need for robots that can assist human work, especially in the manufacturing industry, has driven high demand for the development of efficient and affordable robots [1]. One type of robot that is widely used is the robotic arm, which is capable of performing complex tasks with high precision and speed. However, the development process of robotic arms often requires significant time and resources. Therefore, computer simulation becomes crucial to accelerate this process and save costs [2, 3].

The mathematical and kinematic modeling of robot manipulators can provide a solid foundation for understanding and evaluating the performance of robotic systems before physical implementation in the field, as done by [4].

Previous research on robot manipulators includes the work of [5], who modeled the motion of a 4DOF (degree of freedom) robot manipulator using the MATLAB Graphic User Interface (GUI). Motion control of robots has also been explored to reduce position control errors, such as using the PID method [5, 6], Fuzzy Logic [7], and combining PID and Fuzzy Logic control [8].

The simulation of robot manipulators using MATLAB software has also been explored by [9]. In this research, the SimScape Multibody toolbox was leveraged for the visualization of the robot manipulator. This approach can help accelerate the robot development process and enhance the efficiency in utilizing resources.

Furthermore, the research on Manipulator robots has been extensively investigated by [10]. This study utilized a 4-DOF RRRR Manipulator Robot model and the Robotic System Toolbox library to compute the inverse kinematics of the manipulator. The findings of this research provide valuable insights into the development and control of such robotic systems.

This research will focus on the simulation testing of the kinematics of a Manipulator. This simulation allows researchers to evaluate various scenarios and operational conditions without having to implement the physical system in the field. By using simulation software such as the SimScape Multibody Toolbox and the MATLAB Robotic System Toolbox, This research can investigate aspects such as the joint angle values, the end-effector position coordinates, and the robot dynamics. Thus, researchers can gain a deeper understanding of the system performance and identify ways to improve the efficiency, precision, and reliability of the manipulator robot in various applications, such as in the fields of industry, healthcare, agriculture, and others.

With an emphasis on simulation, this research aims to accelerate the development process of robotic technology and minimize the risks associated with physical implementation in the field.

II. Methods

A. Forward and Inverse Kinematics of 4-DOF RRRP Manipulator

In this research, the kinematics of the robot manipulator is divided into two types, namely Inverse Kinematics and Forward Kinematics. Inverse kinematics is used to obtain the joint angle values for each robot joint, if the desired position (*x*, *y*, *z*) is known. Meanwhile, forward kinematics is used to obtain the position (*x*, *y*, *z*) of the robot, if the values of each joint are known. The explanation regarding the kinematics in this research can be seen in the flowchart shown in Figure 1.

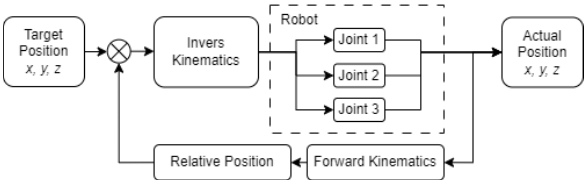

Fig. 1. Robot Kinematics

In this research, a 4-DOF RRRP manipulator was used as the model for simulation in MATLAB. The forward kinematics and inverse kinematics will be discussed in this chapter. The kinematics of the 4-DOF RRPR robot can be described in detail using the D-H parameter notation for each joint. According to [15-17], the D-H parameters are widely used in research because they are simple and easier to understand in describing the kinematics of robotic arms. Figure 2 shows the structure of the 4-DOF RRPR robot, where joints 1, 2, and 3 are revolute joints, while joint 4 is a prismatic joint. The D-H parameter values can be seen in Table 1.

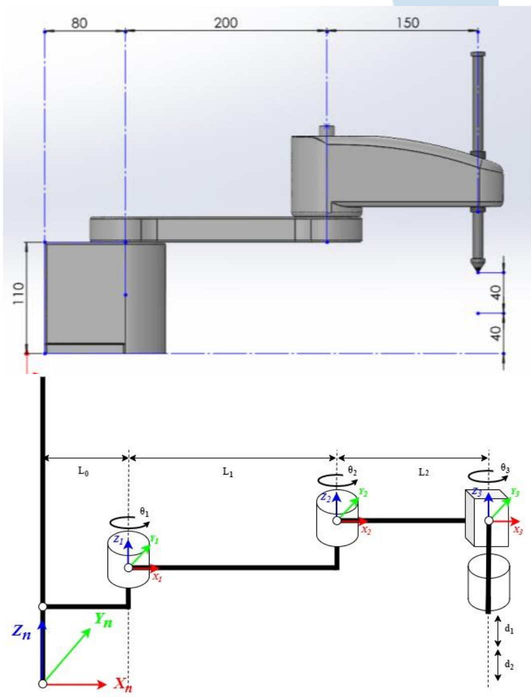

Fig. 2. Free Body Diagram of 4-DOF RRRP Manipulator

TABLE I. D-H PARAMETER OF 4-DOF RRRP MANIPULATOR

Link	a_i (m)	α_i (deg)	d_i (m)	θ_i (deg)
1	L1	0	0	θ_1
2	L2	π	0	θ_2
3	0	0	D1	0
4	0	0	D2	θ_3

From the D-H parameters shown in Table 1, the values can be transformed into a D-H matrix form as shown in Equation (1). The four values generated by each joint are combined into a single matrix, which results in an equation to find the coordinates x, y, and z. Furthermore, the lengths of the shoulder arm (L_1), elbow arm (L_2), and the length of the end-effector's rise are incorporated into that equation.

$$A_i = \begin{bmatrix} cos\theta_i & -cos\alpha_i sin\theta_i & sin\alpha_i sin\theta_i & a_i cos\theta_i \\ sin\theta_i & cos\alpha_i cos\theta_i & -sin\alpha_i cos\theta_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The four values from each joint that have been transformed into matrix form are shown in Equations (2), (3), (4), and (5). These four matrices are then combined into a single matrix as shown in Equation (6). In these equations, Sin is represented by S, and Cos is represented by C.

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1C_1 \\ S_1 & C_1 & 0 & a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_2 = \begin{bmatrix} C_2 & S_2 & 0 & a_2C_2 \\ S_2 & -C_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_4 = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_4^0 = A_1 \dots A_4 = \begin{bmatrix} C_{12}C_4S_{12}S_4 & -C_{12}S_4 + S_{12}C_4 & 0 & a_1C_1 + a_2C_{12} \\ S_{12}C_4 - C_{12}S_4 & -S_{12}S_4 - C_{12}C_4 & 0 & a_1S_1 + a_2C_{12} \\ 0 & 0 & -1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Using Equation (6), it is possible to calculate the values of the coordinates x, y dan z. The coordinate value of x can be determined through Equation (7). The coordinate value of y can be determined through Equation (8). And the coordinate value of z can be determined through Equation (9).

$$x = a_1C_1 + a_2C_{12} \quad (7)$$

$$y = a_1S_1 + a_2S_{12} \quad (8)$$

$$z = d_3 \quad (9)$$

The inverse kinematics in this study is to find the values of θ_1, θ_2, and d_1. The solution to this inverse kinematics can be solved using the Pythagorean theorem and the cosine rule. The kinematic solution in the 4-DOF RRPRR robot is solved using a single side, namely the top view of the robotic arm structure. If we look at Figure 2, we can see that the position on the Zaxis only depends on the translation movement of d_1, so it can be concluded that $d_1 = Z$. Figure 3 provides a representation to obtain an equation to generate the values of θ_1 and θ_2.

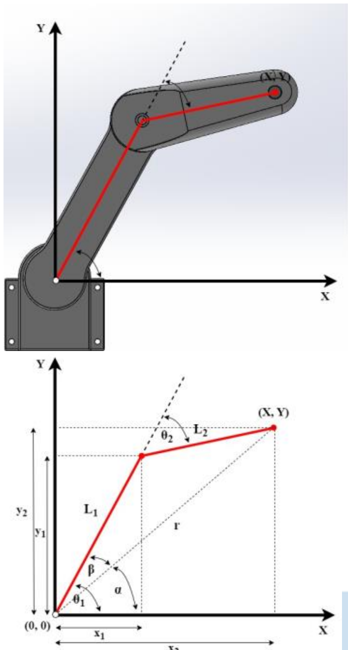

Fig. 3. Cartesian Coordinate 2-DOF Manipulator

As shown in Figure 3, L_1 represents the length of the shoulder arm and L_2 represents the length of the elbow arm. θ_1 represents the angle of the shoulder, and θ_2 represents the angle of the elbow. Using the tangent law, an equation is obtained as shown in Equation (10). To find the value of β, Equation (11) is used. Thus, the value of θ_1 will be obtained from Equation (12). The value of θ_2 is obtained using the cosine law, as shown in Equation (13).

$$\alpha = tan^{-1}\frac{y_2}{x_2} \quad (10)$$

$$\beta = tan^{-1}\frac{L_2 sin\theta_2}{L_1 + L_2 cos\theta_2} \quad (11)$$

$$\theta_1 = tan^{-1}\frac{y_2}{x_2} + tan^{-1}\frac{L_2 sin\theta_2}{L_1 + L_2 cos\theta_2} \quad (12)$$

$$\theta_2 = -cos^{-1}\frac{x_2^2 + y_2^2 - L_1^2 - L_2^2}{2L_1L_2} \quad (13)$$

B. *Setup Experiment*

In this research, the robot manipulator is driven using MATLAB Simulink. Simulink serves as the workspace for simulating the movement of the robot manipulator. In the robot manipulator motion simulation, a task will be given to reach a predetermined position. The input provided is the coordinate point on the x, y, and z axes, which refer to the end-effector position. Subsequently, it will be observed whether the simulated motion is in accordance with the inputted trajectory.

Before conducting the simulation, the information related to the robot must be defined first using SimScape Multibody, which is a library in the MATLAB software. The defined robot dimensions include the joint positions, link lengths, and frame positions on the x, y, and z axes. The block diagram that defines the robot dimensions in Simulink can be seen in Figure 4.

To calculate the Inverse Kinematics, Forward Kinematics, joint torque, and joint force, the Robotic System Toolbox library will be used. The overall system block diagram can be seen in Figure 5.

In this research, we focused on the use of computer simulation to model the movement of the robot manipulator. However, this research has not yet considered the specifications of the robot actuators used, such as the rotational degree resolution, actuator torque, and others.

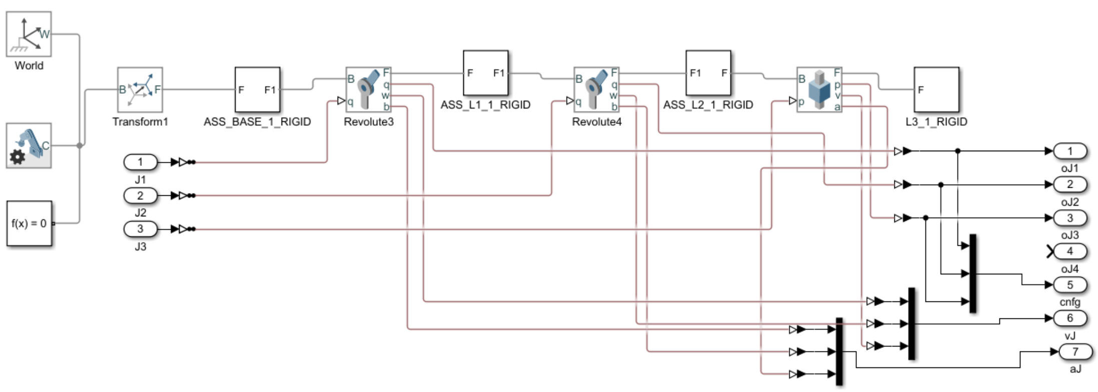

Fig. 4. Robot Block Diagram

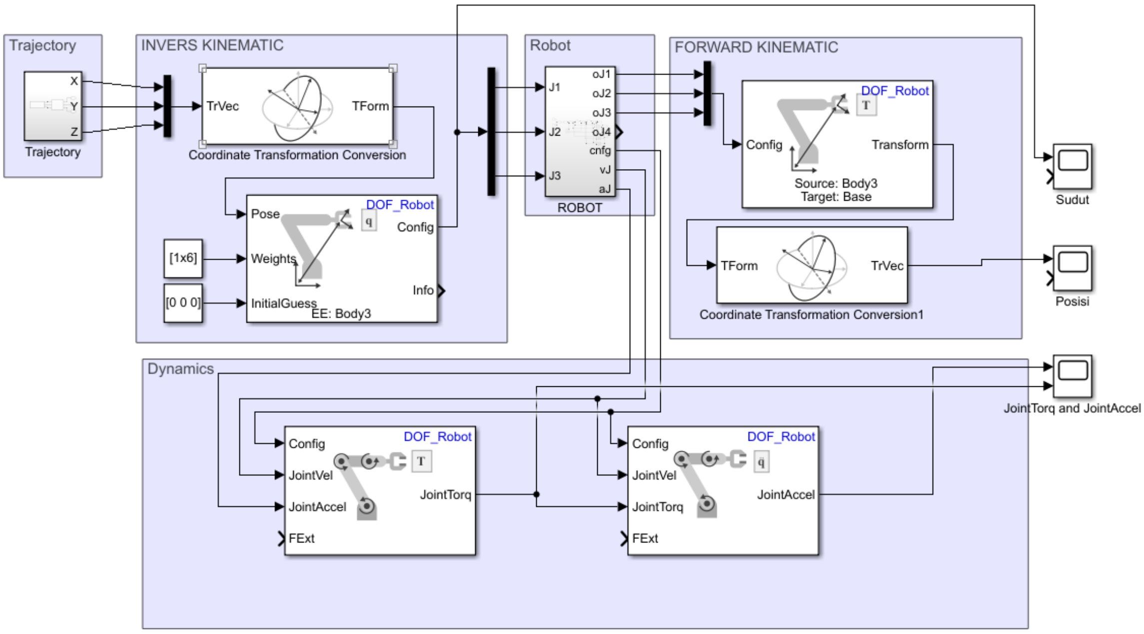

Fig. 5. System Block Diagram

III. RESULT AND DISCUSSIONS

The testing is carried out by moving the robot along a trajectory as shown in Figure 6, and the details can be seen in Table 2. This will provide the values of each joint angle obtained from the Inverse Kinematics block, as well as the End-Effector position. This research will also display the torque and acceleration values at each joint.

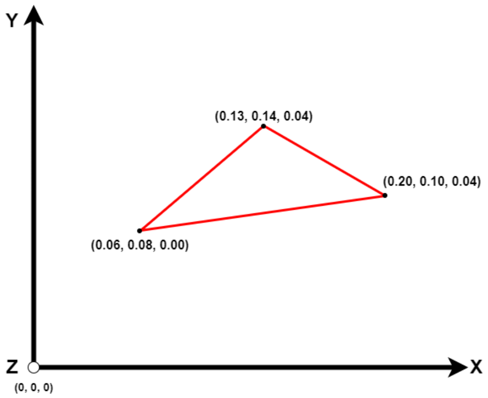

Fig. 6. Trajectory of Robot Manipulator

TABLE II. TRAJECTORY OF ROBOT MANIPULATOR

Time (s)	Coordinate (m)		
	X	Y	Z
0	0.06	0.08	0.00
2	0.06	0.08	0.00
4	0.13	0.14	0.00
5	0.13	0.14	0.04
6	0.13	0.14	0.00
8	0.20	0.10	0.00
9	0.20	0.10	0.04
10	0.20	0.10	0.00
12	0.06	0.08	0.00

In this research, it is assumed that joint 3 does not rotate, so the value of $\theta_3 = 0$. After simulating the robot movement using the trajectory shown in Table 2 and Figure 6, the joint angle values were obtained as shown in Figure 7, and the End-Effector position coordinates were obtained as shown in Figure 8.

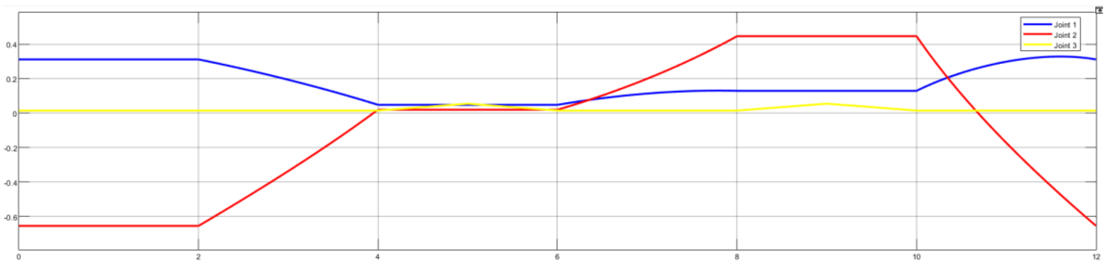

Fig. 7. Value Angle of Joint 1, joint 2 and joint 3

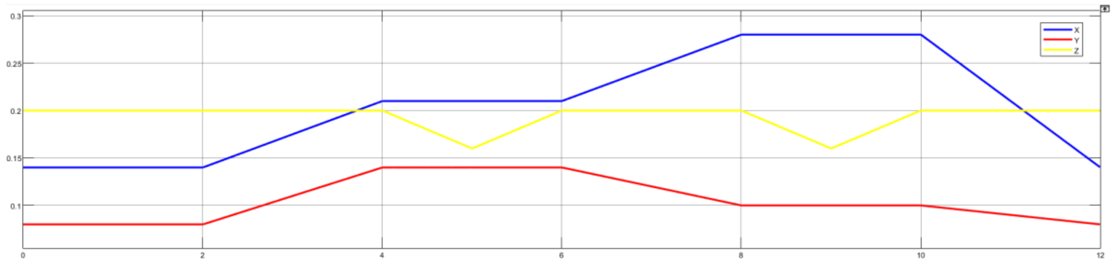

Fig. 8. Position of End-Effector

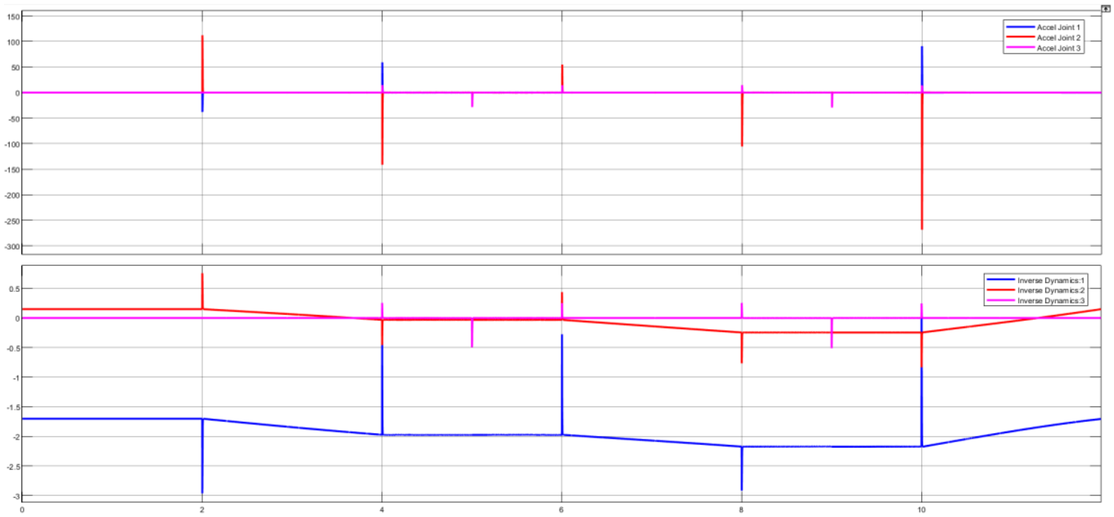

Fig. 9. Value Acceleration and Torque Robot Manipulator

From Figure 7, it can be seen that in the initial or standby position (t = 0 - 2 seconds), joint 1 is around 0.312, joint 2 is around -0.65, and joint 3 is around 0. Joint 3 only moves at (t = 4, 5, 8, 9) and when joint 3 moves, joint 1 and joint 2 do not move. Then at t = 2 - 4 seconds, joint 1 and joint 2 are around 0 - 0.05. At t = 6 - 8 seconds, joint 1 and joint 2 move up successively, approaching values of 0.13 and 0.45. When t = 10 - 12 seconds, the values of joint 1 and joint 2 return to the initial standby position.

For the End-Effector position, as shown in Figure 8, the x, y, z positions obtained from Forward Kinematics are relatively the same as the input trajectory shown in Table 2.

According to the data provided in Figure 9, the following can be observed. Joint 3 only experiences acceleration at specific time points. At t = 4, 5, 6, 8, 9, and 10 seconds, joint 3 has acceleration values of 14.7, -28.33, 14.5, 14.70, -29.22, and 14.10, respectively, along with corresponding torque values of 0.25, -0.50, 0.25, 0.25, -0.51, and 0.24.

At t = 2 seconds, joint 1 has an acceleration of -38 and a torque of -2.96, while joint 2 has an acceleration of 112.25 and a torque of 0.75. At t = 4 seconds, joint 1 accelerates at 59.25 with a torque of -0.14, and joint 2 accelerates at -141.00 with a torque of -0.45. Moving on to t = 6 seconds, joint 1 has an acceleration of 31.8 and a torque of -0.27, while joint 2 has an acceleration of 55.00 and a torque of 0.43. At t = 8 seconds, joint 1 accelerates at 4.6 with a torque of -2.91, and joint 2 accelerates at 54.88 with a torque of 0.43. Lastly, at t = 10 seconds, joint 1 has an acceleration of 90.83 and a torque of 0.01, and joint 2 has an acceleration of -268.26 and a torque of -0.83.

Overall, the data shows that the robotic system has a complex dynamic behavior with coordinated joint movements. Further analysis is required to understand the relationship between the inputs, kinematics, and dynamics of this robotic system.

IV. Conclusion

Based on the analysis conducted in this study, it can be concluded that the approach used provides a significant contribution to the development of robotic technology. The kinematic modeling of the robot manipulator enables the accurate identification of the joint angle values and the end-effector position coordinates. Additionally, the simulation using software such as the SimScape Multibody Toolbox and the MATLAB Robotic System Toolbox provides a deep understanding of the performance of the robot manipulator system.

The simulation accelerates the development process of robotic technology and reduces the risks associated with physical implementation in the field. The simulation results allow researchers to evaluate various scenarios and operational conditions without having to perform physical implementation, which in turn can improve the efficiency, precision, and reliability of robot manipulators in various applications. Thus, this research provides valuable insights into the development of modern robotics and emphasizes the importance of modeling and simulation in understanding and enhancing the performance of robot manipulator systems.

For future research, researchers may consider investigating various other disturbance factors, such as electromagnetic interference, unexpected external forces, or unstable surface conditions, which can affect the performance and accuracy of the robot manipulator's movement. By identifying these sources of disturbance, the research can focus on the development of more advanced and adaptive control systems to dynamically compensate for various disturbances, thus enhancing the efficiency, precision, and reliability of robot manipulators. Additionally, exploring robot manipulators with higher degrees of freedom can also provide new insights into the development of more advanced and adaptive robotic technologies for challenging operational conditions.

REFERENCES

[1] G. Z. Yang *et al.*, "The grand challenges of Science Robotics," *Sci Robot,* vol. 3, no. 14, Jan 31 2018, doi: 10.1126/scirobotics.aar7650.

[2] W.-c. Lee and S.-a. Kuo, "Simulation and control of a robotic arm using MATLAB, simulink and TwinCAT," in *2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, 2020: IEEE, pp. 1-5.

[3] Z. Liqiu, A. Juan, Z. Ronghao, and M. Hairong, "Trajectory Planning and Simulation of Industrial Robot Based on MATLAB and RobotStudio," presented at the 2021 IEEE 4th International Conference on Electronics Technology (ICET), 2021.

[4] G. R. Cahyono, M. F. Setiawan, Y. Rizal, and J. Riadi, "Comparison of 4 DOF Arm Robot for Trajectory Planning with 3rd and 5th Polynomial Orders," in *2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, 2022: IEEE, pp. 281-286.

[5] M. Tamre, R. Hudjakov, D. Shvarts, A. Polder, M. Hiiemaa, and M. Juurmaa, "Implementation of integrated wireless network and MatLab system to control autonomous mobile robot," *International Journal of Innovative Technology and Interdisciplinary Sciences,* vol. 1, no. 1, pp. 18-25, 2018.

[6] T. H. Noventino, M. R. Rosa, and A. Z. Fuadi, "Pid control design and kinematic modelling of 3-dof robot manipulator," in *2022 International Conference on Electrical Engineering, Computer and Information Technology (ICEECIT)*, 2022: IEEE, pp. 88-94.

[7] D. R. Ramadhan, A. R. Al Tahtawi, and K. Wijayanto, "Kendali Posisi Robot Lengan pada Misi Pick and Place dengan Metode Fuzzy Logic Control," in *Prosiding Industrial Research Workshop and National Seminar*, 2021, vol. 12, pp. 96-102.

[8] M. F. El-Khatib and S. A. Maged, "Low level position control for 4-DOF arm robot using fuzzy logic controller and 2-DOF PID controller," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, 2021: IEEE, pp. 258-262.

[9] B. H. Purwoto, "Pemodelan robot kinematik manipulator menggunakan matlab," *Emitor: Jurnal Teknik Elektro,* vol. 20, no. 2, pp. 141-146, 2020.

[10] M. F. Setiawan and A. Hasanah, "Pemodelan dan Implementasi Gerak Robot Manipulator dengan Menggunakan Robotics System Toolbox pada MATLAB," *JURNAL SURYA ENERGY,* vol. 8, no. 1, pp. 12-19, 2023.

[11] Tian, F., Liu, G. M., & Tao, K. (2012). Kinematics Analysis for a PRRR Manipulator. Applied Mechanics and Materials, 271-272, 1578-1581. https://doi.org/10.4028/www.scientific.net/AMM.271-272.1578

[12] "Robotics Toolbox - Peter Corke," Peter Corke. [Online]. Available: https://petercorke.com/toolboxes/robotics-toolbox/

[13] F. C. Park and K. M. Lynch, Modern Robotics: Mechanics, PlannZing and Control. 2017.