

Computer-Based 4-DoF Robot Arm Simulation Design for Taking an Object as a Robotics Learning Media

Rendyansyah^{1*}, Hera Hikmarika¹, Melia Sari¹

¹University of Srinjaya, Indonesia

ABSTRACT

The development of robotics technology has become the main focus in education as a practical learning tool. Robot arm simulation with 4 degrees of freedom (DoF) can be used as a learning medium in robotics, allowing students to understand the basic concepts of design, operation, and application of robot arm systems visually and interactively. In this research, this simulation uses a visual program based on a human-machine interface that is interactive and easy to use. This research aims to design and build a computer-based 4-DoF robot arm simulation that can be used as a learning medium for students in the field of robotics. This simulation includes designing the robot's kinematic structure, calculating forward and inverse kinematics, setting end-effector movements, and illustrating object retrieval. In the simulation, the robot is visualized in top and side views to provide a comprehensive understanding to the user. The operator simulates by pointing the cursor at the box area and clicking the start and destination points, and then the operator gives the command to run the simulation. The robot moves from the original to the recorded position, returning to its original position. The test results show that the designed 4-DoF robot arm simulation can run well and be used as a practical learning medium for students in the field of robotics, providing a deep understanding of the basic concepts of design, operation, and application of a visual robot arm system. Hopefully, this research can be an effective learning medium for students to understand the concept of designing a 4-DoF robot arm.

ABSTRAK

Pengembangan teknologi robotika telah menjadi fokus utama dalam dunia pendidikan sebagai alat bantu pembelajaran yang efektif. Simulasi robot lengan 4 derajat kebebasan (DoF) dapat dimanfaatkan sebagai media pembelajaran dalam bidang robotika, memungkinkan siswa untuk memahami konsep dasar perancangan, pengoperasian, dan aplikasi sistem robot lengan secara visual dan interaktif. Dalam penelitian ini, simulasi ini menggunakan program visual berbasis Human machine Interface yang interaktif dan mudah digunakan. Penelitian ini bertujuan untuk merancang dan membangun sebuah simulasi robot lengan 4 DoF berbasis komputer yang dapat digunakan sebagai media pembelajaran bagi siswa di bidang robotika. Simulasi ini mencakup perancangan struktur kinematika robot, perhitungan forward dan inverse kinematic, pengaturan gerakan end-effector, dan ilustrasi mengambil objek. Dalam simulasi, robot divisualisasi dalam tampak atas dan samping untuk memberikan pemahaman yang komprehensif kepada pengguna. Operator menjalankan simulasi dengan mengarahkan kursor pada area kotak dan meng-klik titik awal dan tujuan, kemudian operator memberikan perintah untuk menjalankan simulasi. Robot bergerak dari posisi original menuju posisi yang telah terekam, setelah itu kembali lagi ke posisi originalnya. Hasil pengujian menunjukkan bahwa simulasi robot lengan 4 DoF yang dirancang dapat berjalan dengan baik dan mampu digunakan sebagai media pembelajaran yang efektif bagi siswa dalam bidang robotika, memberikan pemahaman yang mendalam tentang konsep-konsep dasar perancangan, pengoperasian, dan aplikasi sistem robot lengan secara visual. Penelitian ini diharapkan dapat menjadi media pembelajaran yang efektif bagi siswa dalam memahami konsep perancangan robot lengan 4 DoF.

CONTACT

rendyansyah.unsri@gmail.com

KEYWORDS

Robotic Arm, Simulation,
Taking an Object

INTRODUCTION

The development of robotics technology has progressed rapidly in recent decades. Robotics is a branch of science that studies robot design, construction, operation, and application. A robot is a mechanical entity that can perform specific tasks automatically. Artificial intelligence is one of the most significant developments in robotics technology (Afrisal et al., 2020)(Wirawan et al., 2020). Artificial intelligence allows robots to learn and adapt to their environment. The AI enables robots to perform more complex and intricate tasks. For example, robots can perform industrial tasks requiring high speed and accuracy (Sun et al., 2021). The development of robotics

technology has also affected various aspects of human life. Robots are now used in multiple fields, from industry and medicine to education. In industry, robots increase efficiency and productivity (Arents & Greitans, 2022). In medicine, robots perform very complicated and precise operations (Zemmar et al., 2020). In education, robots are used as interactive and engaging learning tools (Bravo et al., 2021).

Using robotics in the learning process can provide many benefits for students, such as increasing interest and motivation to learn, developing technological skills, and preparing them to face the world of work increasingly connected to technology. With robotics, students can learn practically and interactively, which makes learning more exciting and fun. This activity can motivate students to learn, as they can see the direct results of their knowledge. In addition, robotics can also help students develop cooperation, problem-solving, and creativity skills. By studying robotics, students learn about the basic concepts of programming, mechatronics techniques, and the latest technology in robot manufacturing. The experience equips them with the skills needed in a world of work increasingly connected to technology. In addition, learning robotics can also help students prepare themselves for the changing world of work that is increasingly automated.

There is a need for simulation in robot learning as a learning medium, one of which is a simulation of industrial robots or robot arms. Simulation of a robot arm or robot manipulator is a technique used in robotics to simulate the movements and functions of a robot arm or manipulator before it is applied in real life. This simulation usually uses specialized software that allows engineers and robotics experts to design, program, and test manipulator robots without involving actual physical robots. Satriatama & Halim (2022) discussed the initial design of an Arduino-based 3-degree-freedom robot arm that can be controlled via a smartphone. The focus of this initial design includes design drawings, block diagrams of the robot arm system, application of the Denavit-Hartenberg method, and simulation using MATLAB software. Forward Kinematics is used to determine the final position of the end effector. The 3 DOF robot arm has three links, and the robot motion simulation can be represented using the MATLAB application with the Robotic Tool Kit. Purwoto et al. (2020) analyzed the forward and inverse kinematics of a robot manipulator designed using SolidWorks to obtain a 3D model and Matlab to control the forces applied to the joint positions. The research also discusses the DH (Denavit-Hartenberg) algorithm, homogeneous transformation matrix, robot kinematics, MATLAB GUI, and Simscape Multibody Link. This research is essential in the development of science and technology, especially in the field of robotics.

On the other hand, the author has also designed a simulation of a 4-DoF robot arm in picking up objects based on remote control in a computer program (Rendyansyah et al., 2023). One of the main advantages of robot arm simulation is that it allows developers to identify problems and errors in the robot design before the production process begins. With simulation, engineers can conduct trials of various scenarios and conditions without preparing expensive and complicated physical prototypes. In the future, the development of robot manipulators will be more efficient and cost-effective.

Robot arms have a wide range of applications, from the manufacturing industry to the medical field. Simulation is one of the most effective methods for optimizing the design and function of robot arms. In this study, we designed a 4-of robot arm simulation based on a visual computer program to simulate the robot picking up objects. This simulation aims to model the behavior of the robot arm in various situations and working conditions. We can easily adjust the robot arm parameters such as arm length, joint angle, and movement speed using a visual computer program. One of the advantages of simulating a robot arm is its ability to predict the robot's performance before it is physically built. By conducting simulations, we can identify potential problems or design flaws early on, allowing us to improve before the robot arm is built. The simulation can save production time and costs and improve the efficiency of robot development. In addition, simulation is also an effective tool for training operators to operate the robot arm. By using a computer visual program, operators can practice and model the robot arm's movements without having to spend time and money to use a physical robot. The visual program can improve the operator's expertise in operating the robot arm and reduce the risk of accidents during the actual use of the robot arm.

METHODS

The 4-DoF robot arm is in the form of a simulation made in a visual computer program. This robot arm has four degrees of freedom, namely joint₁ to joint₄. Figure 1 shows an illustration of the link length setting. In this study, the link length can be adjusted according to the work area's range. In this case, the links d_1 , a_2 , a_3 , and a_4 are set to 15 cm, 20 cm, 20 cm, and 20 cm, respectively. The robot's appearance in the simulation adjusts as the value of each link changes. Joint₁ is the base of the robot that moves link_{d1}, and the robot moves left and right.

Joint_2 for movement on link_a2, joint_3 for moving link_a3, and joint_4 for movement of link_a4. Joint_1 is the horizontal movement route in the x and y axes, while joint_2 to joint_4 is the vertical route in the z-axis.

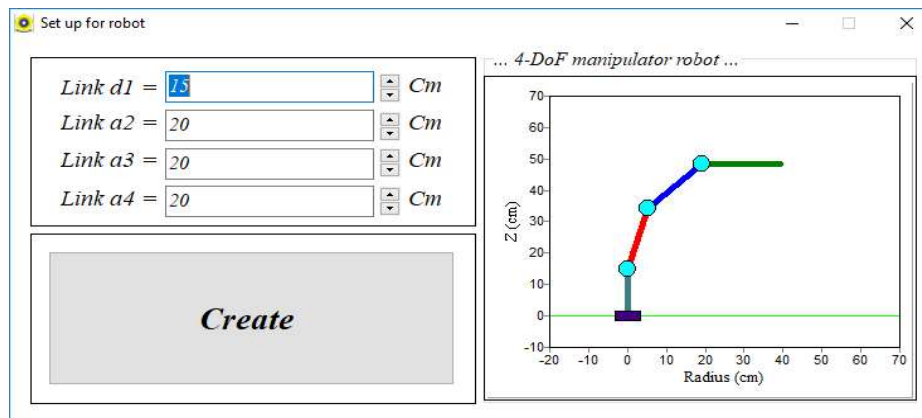


Figure 1. 4-DoF robot arm link length setting.

The movement of the 4-DoF robot arm can be set in the forward and inverse kinematic formulas. The solution in finding kinematics equations uses DH parameters and trigonometric equations (Ekaputra & Setiawan, 2017)(Satriatama & Halim, 2022)(Nurkholik et al., 2022). The forward Equation is shown in Equation (1). The inverse kinematic formula uses trigonometric rules, which, in this case, is illustrated in Figure 2. The final solution of inverse kinematics is seen in Equations (2) to (3). Please note that the orientation angle (ϕ) is set at -45° to the x-axis so that the end-effector position is always facing toward the target.

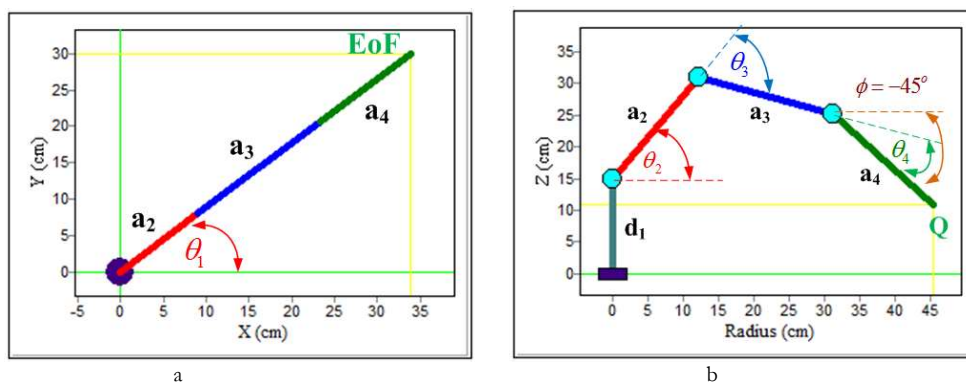


Figure 2. Joint angles of the 4-DoF robot arm, a) top view, and b) side view.

$$\begin{aligned}
 x &= (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + a_4 \cos(\theta_2 + \theta_3 + \theta_4)) \cos \theta_1 \\
 y &= (a_2 \cos \theta_2 + a_3 \cos(\theta_2 + \theta_3) + a_4 \cos(\theta_2 + \theta_3 + \theta_4)) \sin \theta_1 \\
 z &= d_1 + a_2 \sin \theta_2 + a_3 \sin(\theta_2 + \theta_3) + a_4 \sin(\theta_2 + \theta_3 + \theta_4)
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 Q_{z1} &= \sqrt{(Q_x)^2 + (Q_y)^2} \\
 P_x &= Q_{z1} - (a_4 \cdot c \phi) \\
 P_y &= Q_z - d_1 - (a_4 \cdot s \phi)
 \end{aligned} \tag{2}$$

$$\begin{aligned}\theta_3 &= \cos^{-1} \left(\frac{(P_x)^2 + (P_y)^2 - (a_2)^2 - (a_3)^2}{2a_2a_3} \right) \\ \theta_2 &= \tan^{-1} \left(\frac{P_y}{P_x} \right) - \tan^{-1} \left(\frac{a_3 \cdot s \theta_3}{a_2 + a_3 \cdot c \theta_3} \right) \\ \theta_4 &= \phi - \theta_2 - \theta_3 \\ \text{and} \\ \theta_1 &= \tan^{-1} \left(\frac{Q_y}{Q_x} \right)\end{aligned}\quad (3)$$

In this research, the 4-DoF robot arm uses the inverse kinematic function to calculate the joint angle based on the input cartesian value. Meanwhile, forward kinematics is used to get cartesian values based on joint angle input. Equation (1) shows that joint values influence the x, y, and z positions. The x and y position values represent the horizontal direction, and the z position represents the vertical direction. The kinematics equation is applied in a computer-based visual program simulation. In the simulation, there is a cursor area that the operator can use to determine the starting point and target, as shown in Figure 3.

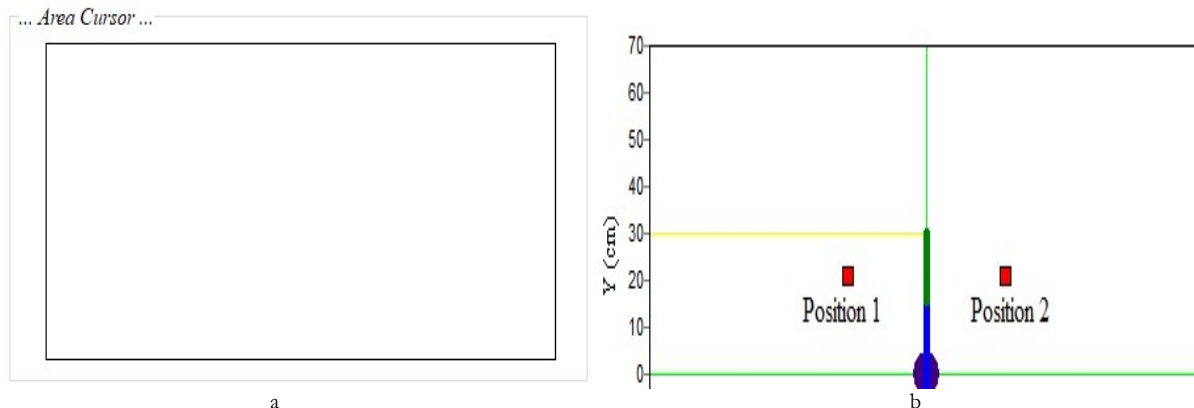


Figure 3. The cursor area used by the operator in determining the start and target points, a). Cursor area, and b) Starting and target point positions

This cursor area has a length (horizontal) of 140 cm and a width (vertical) of 70 cm. The size adjusts the area at arm's reach. The cursor area determines the position of the initial target and the final target; the operator determines the position. The operator selects the location of the object that the robot will take and places it in the desired position. The movement of the robot arm from the starting position to the end uses the Trajectory Planning method. Trajectory planning is a valuable path-planning process in developing navigation systems for robotics. Trajectory planning aims to produce an optimal path for the robot to achieve a specific goal by considering position, speed, and acceleration. (Afrisal et al., 2020).

In this research, Trajectory uses the development of a third-order polynomial, namely Cubic Trajectory Planning. The final result of the Cubic Trajectory Planning development is shown in Equation (4). The variable q_s is the initial position, and q_f is the final position. The variable t_f is the time duration for the robot to move from the starting position to the final position. The next step is to adjust Equation (4) into cartesian x, y, and z, as shown in Equation (5).

$$q(t) = q_s + 3 \left(\frac{q_f - q_s}{t_f^2} \right) t^2 - 2 \left(\frac{q_f - q_s}{t_f^3} \right) t^3 \quad (4)$$

$$\begin{aligned}
 q_x(t) &= q_{xs} + 3 \left(\frac{q_{xf} - q_{xs}}{t_{yf}^2} \right) t^2 - 2 \left(\frac{q_{xf} - q_{xs}}{t_{yf}^3} \right) t^3 \\
 q_y(t) &= q_{ys} + 3 \left(\frac{q_{yf} - q_{ys}}{t_{yf}^2} \right) t^2 - 2 \left(\frac{q_{yf} - q_{ys}}{t_{yf}^3} \right) t^3 \\
 q_z(t) &= q_{zs} + 3 \left(\frac{q_{zf} - q_{zs}}{t_{zf}^2} \right) t^2 - 2 \left(\frac{q_{zf} - q_{zs}}{t_{zf}^3} \right) t^3
 \end{aligned} \tag{5}$$

The Cubic Trajectory Planning method in Equation (5) works by adjusting the cartesian coordinates with each joint angle. This trajectory applies inverse kinematics because it requires position input (x, y, and z) and generates each joint angle value. In this research, the 4-DoF robot arm, the formulation of forward and inverse kinematics, and Cubic Trajectory Planning are programmed in a computer that forms the Human-Machine Interface.

RESULTS AND DISCUSSIONS

In this research, the 4-DOF robot arm is in the form of a computer-based simulation. Simulations are made using interface-based visual programming. The display form of the 4-DOF robot arm simulation can be seen in Figure 4. This robot arm is set for each link $d1 = 15$ cm, $a2 = 20$ cm, $a3 = 20$ cm and $a4 = 20$ cm. The length of these links can be adjusted to the furthest reach in the simulation. In running this robot simulation, starting with clicking the "Setting Robot" button, the operator can set the link length according to the farthest reach of the simulation (see Figure 1). The simulation places the original position of the robot arm in a cartesian formation $x = 0$ cm, $y = 30$ cm, and $z = 20$ cm. The x and y axes are for horizontal motion, while the z axis is for vertical motion (see Figure 5). The value of $z = 20$ cm indicates the robot releasing the object, while $z = 3$ cm for the robot to pick up the object. In this simulation, the robot is displayed in two views, namely the top and side views. The top view of the robot illustrates the robot moving left and right (x and y-axis coordinate positions). The side view demonstrates the robot's movement up and down (z-axis coordinates).

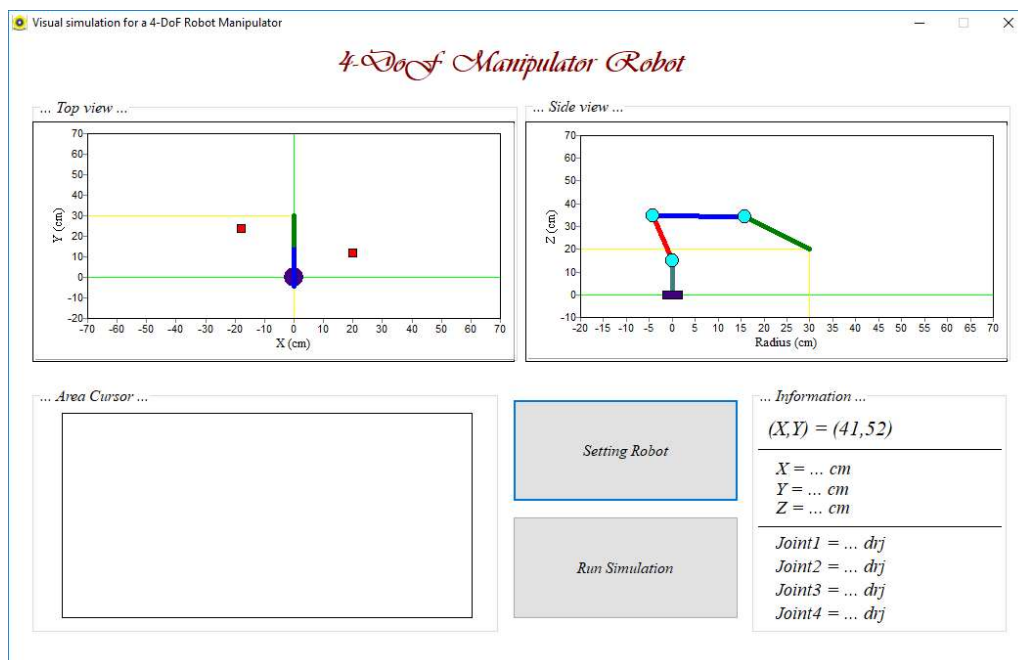


Figure 4. Simulation view of the 4-DoF robot arm.

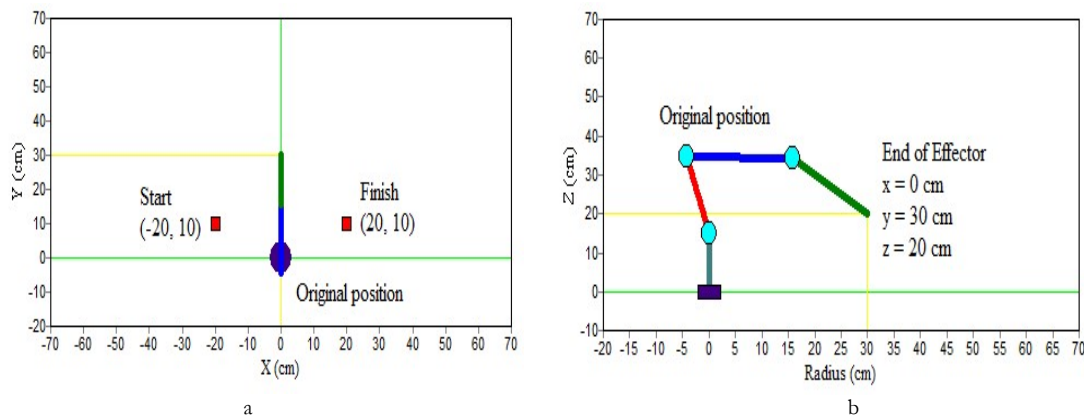


Figure 5. Illustration of the selection of the initial and final positions and the display of the original position on the robot arm. a) top view, and b) side view.

Next, the operator determines the start and end positions in the cursor area. The start position is the position of the object to be picked up by the robot towards the end position as the place to put the object. Figure 5 shows the operator's selection of the start and end positions, marked in the robot's top view. The positions are stored in a file to be read by the system, and the robot moves accordingly. Table 1 shows the positions recorded in the file, starting from the original position, picking up and moving the object, and returning to the original position. The operator chooses the x and y coordinate values to match the arm's reach, while the value $z = 20$ cm indicates the robot picking up or releasing the object, and $z = 3$ cm is the robot picking up the object. The object is illustrated with a height between 3 cm and 5 cm.

Table 1. Coordinate positions recorded in the system by the operator.

Coordinates			Information
x	y	z	
0	30	20	Original position
-	10	20	
-	10	3	The robot picks up the object
-	10	20	
20	10	20	The robot moves and places Object
20	10	3	
20	10	20	
0	30	20	Original position

After the operator presses the "Run Simulation" button, the robot moves according to the route that has been recorded. The robot's movement from one point to the next uses a Cubic Trajectory with a time interval of 1 second. The time interval of 1 second is chosen because it pauses the robot's movement in tracking the Cartesian value. Figure 6 shows the response to changes in position and speed on the robot arm read from the end-effector. Based on Figure 6, it can be seen that the robot arm can complete its task in 7.6 seconds.

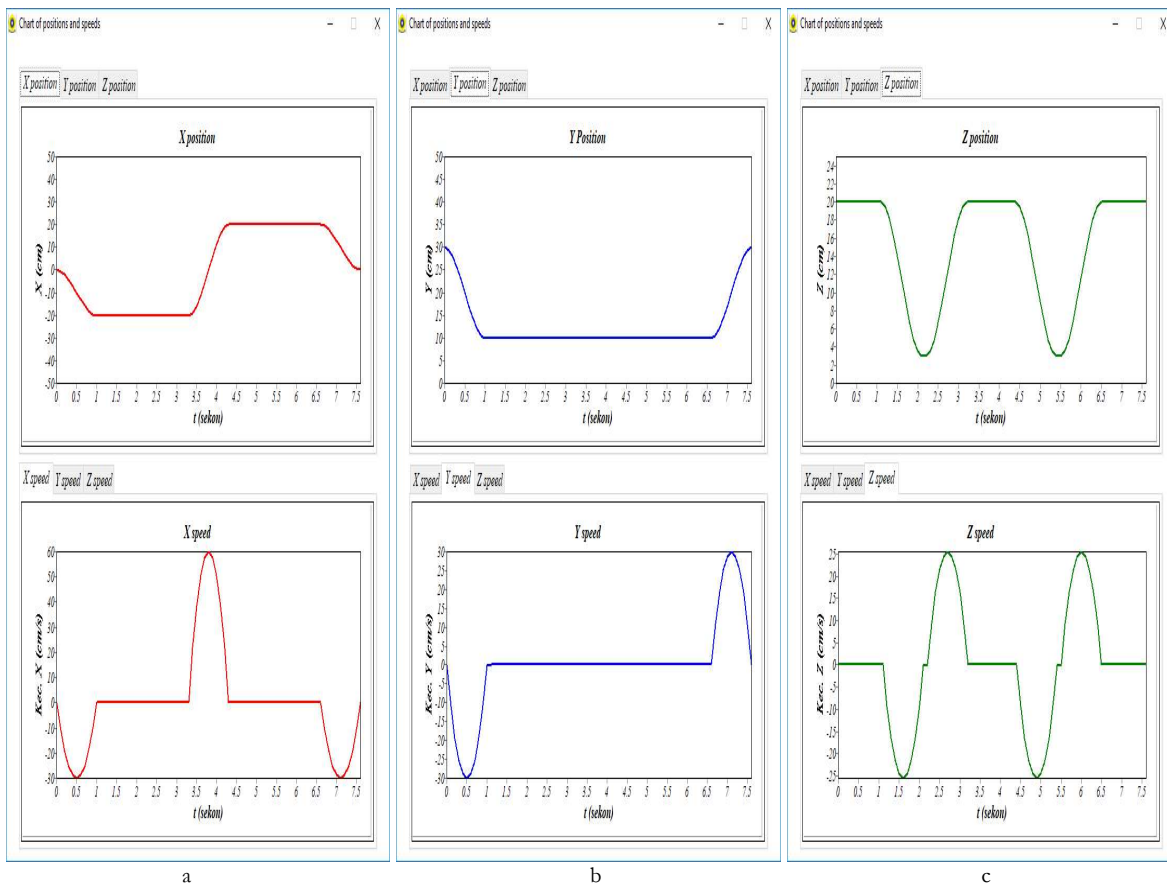


Figure 6. Changes in cartesian position and velocity values as the robot traverses the specified tracking starting position (-20, 10, 5) and ending position (20, 10, 5). a) changes in the x coordinate, b) y coordinate, and c) z coordinate.

The results of this experiment show that the robot can complete its task well, and the movement is smooth because it applies the Cubic Trajectory method. The experiment was also carried out several times, namely ten times, and provided that the coordinate point did not exceed the radius or range of the end-effector, which is 45 cm. Cubic Trajectory in the robot arm completes picking up and moving objects. If an operator selects a target that exceeds the radius, the simulation indicates, "The robot does not reach the radius." Figure 7 shows the simulation information that the robot did not reach the radius. Then, the operator can click OK to continue the simulation, and the operator can return to the original position by clicking the "Setting Robot" button. Overall, the experiment in this 4-DOF robot arm simulation runs well and can be applied to robotics learning on the theme of Manipulators or Industrial Robots.

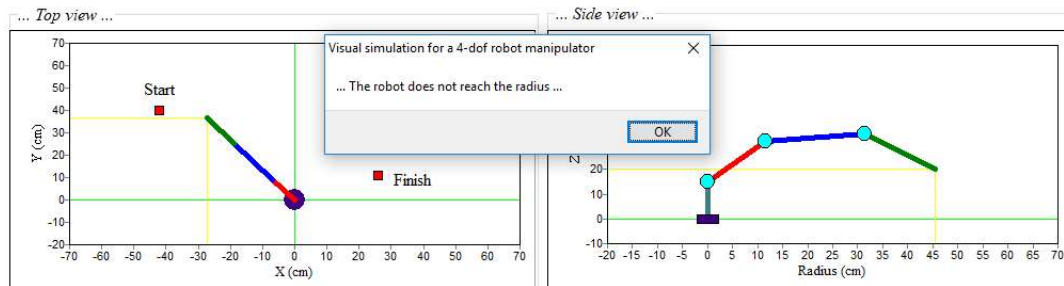


Figure 7. Simulation view when the coordinate point exceeds the end-effector radius range.

CONCLUSIONS

A 4-DoF arm robot has been designed in the form of a visual computer program-based simulation. The robot is illustrated in the task of picking up and moving objects. The robot arm can be set for each link length in this simulation. The link lengths $d1 = 15$ cm, $a2 = 20$ cm, $a3 = 20$ cm, and $a4 = 20$ cm are created by default. The link length selection is determined based on the farthest position of the End-Effector or arm reach. The robot simulation is moved by selecting the starting and ending positions in the cursor area. Then, the operator clicks the "Run Simulation" button, moves the robot arm from the original position to the initial target, moves the object, and places the object on the final target, after which the robot returns to the original position. The robot's movement in tracking the area uses the Cubic Trajectory Planning method. In this study, the simulated robot moves well according to the coordinate pattern recorded in the system. The time required for the robot to complete its task is ± 7.6 seconds. This time applies to different position patterns because the time interval from one position to another is 1 second. This simulation also displays the movement response on the coordinate route and speed change. The subsequent development is the simulation of multiple robot arms controlled in one Human-Machine Interaction.

ACKNOWLEDGEMENTS

The research/publication of this article was funded by DIPA of Public Service Agency of Universitas Sriwijaya 2024. Nomor SP DIPA 023.17.2.677515/2024, On November 24, 2023. In accordance with the Rector's Decree Number: 0013/UN9/LP2M.PT/2024, On May 20, 2024.

REFERENCE

- Afrisal, H., Setiyono, B., Yusuf, M. F., Suin, R. M., & Toirov, O. (2020). Trajectory Planning with Obstacle Avoidance of 3 DoF Robotic Arm for Test Tube Handling System. *In 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, 41–46.
- Arents, J., & Greitans, M. (2022). Smart Industrial Robot Control Trends, Challenges and Opportunities within Manufacturing. *Applied Sciences*, 12(2), 1–20.
- Bravo, F. A., Hurtado, J. A., & González, E. (2021). Using Robots with Storytelling and Drama Activities in Science Education. *Education Sciences*, 11(7), 1–16.
- Ekaputra, I. M. W., & Setiawan, J. D. (2017). Pengembangan Wearable Robotic Arm Input dan Virtual Instrument untuk Pengendalian dan Pemantauan Lengan Robot. *Jurnal Rekayasa Mesin*, 8(2), 109–119.
- Nurkholik, Z., Alif, F., & Arie, D. (2022). Rancangan Bangun Lengan Robot Arm Untuk Menggambar Menggunakan Invers Kinematik. (*JUSI*) *Jurnal Ilmiah Sistem Informasi*, 1(3), 59–68.
- Purwoto, B. H., Rhokhim, D. R., & Indraswari, D. (2020). Pemodelan Robot Kinematik Manipulator Menggunakan Matlab. *Emitor: Jurnal Teknik Elektro*, 20(2), 141–146.
- Rendyansyah, Hikmarika, H., Irmawan, Pratama, I., & Alrizki, M. F. (2023). Robot Manipulator 4-DoF Dalam Memindahkan Objek Menggunakan Remote Control Berbasis Simulasi Komputer. *Seminar Nasional AVOER*, 283–288.
- Satriatama, M. F. W., & Halim, L. (2022). Perancangan Awal dan Simulasi Lengan Robot 3 Derajat Kebebasan Berbasis Arduino yang Dikontrol dengan Aplikasi. *Jurnal Teknik Mesin Dan Mekatronika*, 7(2), 118–130.
- Sun, Z., Zhu, M., Zhang, Z., Chen, Z., Shi, Q., Shan, X., Yeow, R. C. H., & Lee, C. (2021). Artificial Intelligence of Things (AIoT) Enabled Virtual Shop Applications Using Self-Powered Sensor Enhanced Soft Robotic Manipulator. *Advanced Science*, 8(14), 1–14.
- Wirawan, N. T., Defnizal, & Ernes, R. N. (2020). Pembuatan Teknologi Robotik Dalam Dunia Militer Sebagai Media Pemantau Dan Negosiasi Berbasis Artificial Intelligence. (*JURTEKSI*) *Jurnal Teknologi Dan Sistem Informasi*, 6(2), 155–162.
- Zemmar, A., Lozano, A. M., & Nelson, B. J. (2020). The Rise of Robots in Surgical Environments During COVID-19. *Nature Machine Intelligence*, 2(10), 566–572