

Physical Modeling of a 4-DOF Manipulator for Learning Purposes Using CAD and GUI

Emmanuel Agung Nugroho^{1,2,*}, Joga Dharma Setiawan², and M. Munadi²

¹ Mechatronics Engineering Department, Indorama Engineering Polytechnic, Purwakarta, Indonesia

² Department of Mechanical Engineering, Faculty of Engineering, Diponegoro University, Semarang, Indonesia
Email : emmanuel.agung@pei.ac.id (E.A.N.); joga.setiawan@gmail.com (J.D.S.); munadi@ft.undip.ac.id (M.M.)

*Corresponding author

Abstract—The growing popularity of manipulators in industry has increased the need for mastery of knowledge about the kinematics and dynamics of manipulators. On the other hand, manipulators are not affordable items for learning purposes, so modeling is one of the right solutions as well as a new form of contribution to introduce physical manipulators without having to have many manipulators in the laboratory. Through this work, a 4-Degree of Freedom (DOF) manipulator can be physically modeled similarly to its original form as an educational robot, and a dashboard can be designed to control its movement. The mechanical physical model of the manipulator is developed using Autodesk Inventor, and the setting of the manipulator parameter is done using the Graphical User Interface (GUI) MATLAB. The manipulator model used is Dobot Magician which has four revolute joints. The advantages of modeling using Autodesk Inventor design tools are an intuitive user interface, easy to understand, and student-free license, so it is more friendly to students than other design software such as in the existing study. As a learning media, this modeling has been very complex to learn mechanical design, converting into MATLAB with XML (extensible markup language) extension on Simscape Multibody, setting kinematics and dynamics on Simscape, and designing control interface using GUI on MATLAB. This work proves the accuracy of robot movement with the path planning method set through the GUI based on the forward kinematics and reverse kinematics approaches.

Keywords—dobot magician, educational robot, physical modeling, forward kinematic, inverse kinematic

I. INTRODUCTION

Currently, manipulators in the industrial world are increasingly widespread for welding, painting, pick and place tasks, other manufacturing tasks and have even developed for the role of vision robots whose function is to monitor, supervise, and quality control for a product [1–3]. This very high development has attracted the attention of researchers and academics to make the control mechanism simpler but still has high precision [4, 5]. Some researchers have done a similar modeling research including Li and Wang [6] wrote the results of their research modeling the four Degree of Freedom (DOF) SCHUNK robot and analyzing kinematics in redundant space with Denavit

Hartenberg (DH) parameters to get the movement coordinates of the modeled robot. Mohammed [7] has modeled the four DOF RA-02 manipulator and analyzed the kinematics using covariance (DH) and exponential product formulas using MATLAB toolbox to obtain a visual manipulator configuration. Arya *et al.* have also modeled a four DOF robot using solidworks and concluded that the modeling made has greatly helped students analyze the kinematics of the robot practically [8].

By modeling a manipulator, the physical characteristics, kinematics and dynamics of a manipulator can be known [9–12]. This research models the Dobot Magician 4 DOF manipulator which is done by redesigning the shape and characteristics of the Dobot Magician using Autodesk Inventor software [13]. The assembly model that has been generated by Inventor is then exported to Simulink MATLAB's Simscape Multibody. In Simscape Multibody, Motion, and torque parameters are set to determine the movement trajectory of the manipulator [14–17]. The Dobot Magician manipulator movement is controlled using a GUI through forward and inverse kinematics equations based on the physical parameters of the Dobot Magician [18–20]. Therefore, a manipulator can be modeled with a similar shape to the original, as well as being able to design its control dashboard based on the kinematics equations of manipulator movement. Design of robot mechanical parts and body assembly is done using Inventor software, setting the dynamics and kinematics of the manipulator is done using Simulink and the manipulator control system is done using MATLAB GUI [21, 22]. A slight difference between this research compared to previous research is combining two different research processes into one integrated process so as to prove the design of the mobile manipulator in accordance with the path planning.

II. RELATED WORKS

A. Dobot Magician

Dobot Magician is a manipulator with four revolute joints consisting of the base, shoulder, elbow, and end of the effector at the wrist. The end of the effector consists of a pneumatic vacuum suction cup mechanism, gripper, laser printing, 3D printing hot-end, and pen holder for drawing

graphs [23–26]. Dobot Magician specifications are presented in Table I.

TABLE I. DOBOT MAGICIAN SPECIFICATIONS [6]

Parameter Item	Specification
Number of Axes	4+1 (Base, Shoulder, Elbow, and wrist + EoF)
Payload	500 g
Maximum reach	320 mm
Position repeatability	0.2 mm
Length of base L_1	85 mm
Length of Shoulder L_2	135 mm
Length of Elbow L_3	147 mm
Length of wrist L_4	59 mm
Length of EoF	50 mm

The Dobot Magician structure diagram consists of four joints: $J_1, J_2, J_3,$ and J_4 . The J_2 position is the starting point with coordinates $(0, 0, 0)$, and the movement position is defined as (x,y,z) [24, 27]. The mechanical structure of Dobot Magician is shown in Fig. 1.

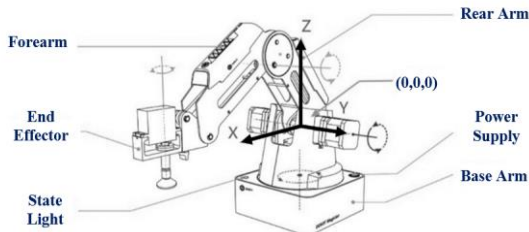


Fig. 1. Mechanical structure of Dobot Magician [2].

B. Kinematics of Four-DOF Manipulator

Kinematics is defined as the mechanism of robot movement from the aspects of position, velocity and acceleration by ignoring the forces that affect the movement of the robot. Robot kinematics is generally divided into two mechanisms, Forward kinematic and inverse kinematic. Forward kinematic is a movement analysis to get position coordinates (x, y, z) if the position of each joint on the robot is known. While inverse kinematics is a movement analysis to get the value/angle of each joint if we know the position coordinate data (x, y, z) [28, 29].

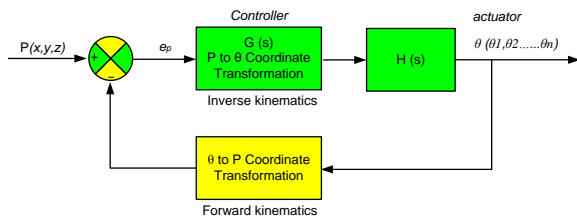


Fig. 2. Robot kinematics system control diagram.

Fig. 2 is a description of the working mechanism of the inverse kinematic robot control system with input in the form of position control that produces the angle of movement of the robot joints and forward kinematic with input in the form of movement angles that produce the final position of the robot movement. Changes in the angle and movement of each joint actuator are carried out with the DH method.

The DH method is expressed in terms of four parameters: twisting angle α_i , link length L , link offset d_i , and joint angle θ_i [30–33]. DH parameters for the 4-DOF Dobot magician manipulator are:

- L_i Link Length; length in the X axis measured from $Z J_2$
- α_i Link Twist; angle on the X axis measured from $Z J_2$
- d_i Link Offset; distance on the Z axis measured from X J_2
- θ_i Joint Angle; angle at the Z axis measured at X J_2

Based on the explanation of these parameters, the modification of DH Dobot Magician parameters can be formulated to obtain a homogeneous transformation matrix that represents the position and orientation of the reference frame with respect to the fixed reference frame of Dobot Magician. The intended modification of DH parameters on Dobot Magician is shown in Table II.

TABLE II. D-H PARAMETER MODIFICATION OF DOBOT MAGICIAN

Frame i	L_i	α_i	d_i	θ_i
1	0	90°	L_1	θ_1
2	L_2	0°	0	θ_2
3	L_3	0°	0	θ_3
4	L_4	0°	0	θ_4
5*	0	90	$-L_5$	$\theta_5 = 0^\circ$

* An optional end of effector attached to wrist frame 4.

C. Homogeneous Transformation Matrices

From the modification of DH parameters in Table III, the relative homogeneous transformation matrix can be formulated for each frame corresponding to [34–36]. DH Parameter-based homogeneous transformation as follows:

$$A_1^0 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & L_1 C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & L_1 S\theta_1 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$A_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & L_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & L_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_3^2 = \begin{bmatrix} C\theta_3 & S\theta_3 & 0 & L_3 C\theta_3 \\ S\theta_3 & -C\theta_3 & 0 & L_3 S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_4^3 = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & L_4 C\theta_4 \\ S\theta_4 & C\theta_4 & 0 & L_4 S\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_5^4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

The homogeneous matrix connecting Frame 1 to Frame 5 is obtained by the following formula:

$$A_5^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 \quad (6)$$

$$A_5^0 = \begin{bmatrix} R_5^0 & P_5^0 \\ 0 & 1 \end{bmatrix} \quad (7)$$

From Eqs. (6) and (7), the rotation matrix and vector location of Frame 5 with respect to Frame 1 are obtained as follows:

$$R_5^0 = \begin{bmatrix} C(\theta_1 + \theta_5) & -S(\theta_1 + \theta_5) & 0 \\ S(\theta_1 + \theta_5) & C(\theta_1 + \theta_5) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

And

$$P_5^0 = \begin{bmatrix} C\theta_1 (L_4 + L_3C(\theta_2 + \theta_3) + L_2C\theta_2) \\ S\theta_1 (L_4 + L_3C(\theta_2 + \theta_3) + L_2C\theta_2) \\ L_1 - L_5 + L_3S(\theta_2 + \theta_3) + L_2S\theta_2 \end{bmatrix} \quad (9)$$

D. Forward Kinematics and Inverse Kinematics

Forward kinematics requires closed-loop vector analysis to derive position equations. The number of equations is determined by the number of joints and the number of manipulator arms. From the derivative of Eq. (9), the forward kinematics equation is obtained as follows:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} C\theta_1 (L_3C(\theta_2 + \theta_3) + L_2C\theta_2) \\ S\theta_1 (L_3C(\theta_2 + \theta_3) + L_2C\theta_2) \\ L_1 + L_3S(\theta_2 + \theta_3) + L_2S\theta_2 \end{bmatrix} \quad (10)$$

The inverse kinematics of the manipulator is always more complex than the forward kinematics because the equations solved are non-linear and result in infinite solutions. Based on Eq. (10), the inverse kinematic parameter approach to obtain $(\theta_1, \theta_2, \theta_3)$ is as follows:

(1) Approximate θ_1

$$P_x = x - L_4; P_y = y \text{ and } P_z = z + L_5$$

$$\frac{S\theta_1}{C\theta_1} = \frac{P_y}{P_x}, \tan \theta_1 = \frac{P_y}{P_x} \quad (11)$$

$$\theta_1 = \tan^{-1} \left[\frac{P_y}{P_x} \right] = \tan^{-1}(P_x, P_y) \quad (12)$$

(2) Approximate θ_3

To simplify the finding of θ_3 , the cosine equation is expressed as follows:

$$a = \frac{P_x}{C\theta_1} = L_3C(\theta_2 + \theta_3) + L_2C\theta_2 \quad (13)$$

$$b = P_z - L_1 = L_3S(\theta_2 + \theta_3) + L_2S\theta_2 \quad (14)$$

Therefore, it is obtained:

$$a^2 + b^2 = L_2^2 + L_3^2 + 2L_2L_3C\theta_3 \quad (15)$$

Then:

$$C\theta_3 = \frac{a^2 + b^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (16)$$

Assuming the object is still within the working range of the Dobot, then:

$$S\theta_3 = \pm \sqrt{1 - C\theta_3^2}$$

Then:

$$\theta_3 = \tan^{-1} \left(\frac{S\theta_3}{C\theta_3} \right) \quad (17)$$

(3) Approximate θ_2

While θ_2 is obtained by the following equation:

$$a = (L_3C\theta_3 + L_2)C\theta_2 - (L_3S\theta_3)S\theta_2 \quad (18)$$

$$b = (L_3S\theta_3 + L_2)S\theta_2 - (L_3C\theta_3)C\theta_2 \quad (19)$$

Eqs. (18) and (19) can be written in another form:

$$a = m C\theta_2 - n S\theta_2$$

$$b = m S\theta_2 - n C\theta_2$$

When

$$m = L_3C\theta_3 + L_2 \text{ and } n = L_3S\theta_3, \text{ if } \\ p = \pm \sqrt{m^2 + n^2} \text{ obtained by}$$

$$\varphi = \tan^{-1} \left(\frac{n}{m} \right) \quad (20)$$

If:

$$m = p \cos \varphi \text{ if } n = p \sin \varphi$$

Then:

$$a = p C\varphi C\theta_2 - p S\varphi S\theta_2 \quad (21)$$

$$b = p C\varphi S\theta_2 - p S\varphi C\theta_2 \quad (22)$$

From Eqs. (21) and (22) obtained by

$$a = p C(\varphi + \theta_2)$$

$$b = p S(\varphi + \theta_2)$$

Then:

$$\varphi + \theta_2 = \tan^{-1} \left(\frac{b}{a} \right) \quad (23)$$

So as to obtain θ_2 from Eqs. (19) and (22) is:

$$\theta_2 = \tan^{-1} \left(\frac{b}{a} \right) - \tan^{-1} \left(\frac{n}{m} \right) \quad (24)$$

III. METHODOLOGY

The purpose of this research is to model the 4-DOF Dobot Magician manipulator using Simscape multibody and control its movement using forward kinematics and reverse kinematics programmed using Simscape Multibody on Simulink and GUIDE MATLAB. The research diagram is shown in Fig. 3.

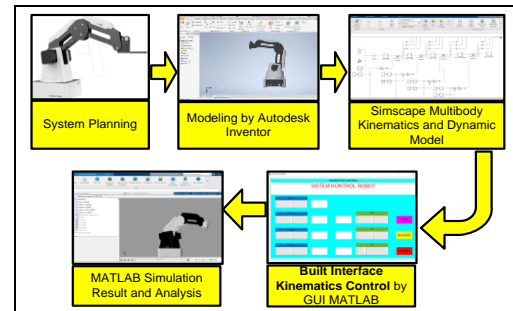


Fig. 3. Block diagram of 4-DOF manipulator modeling.

Fig. 3 explains the stages of the research starting from modeling using Autodesk Inventor to create a Dobot Magician manipulator model. The 3-dimensional model of Dobot Magician is then exported into Simscape Multibody and set the actuator parameters. Control is performed by providing torque and movement parameters at each joint of the manipulator model. The kinematics characteristic equation of Dobot Magician is entered in the MATLAB GUI program script. The Dobot Magician manipulator model can be controlled more interestingly and interactively using MATLAB GUI images. A flowchart of system planning for modeling Dobot Magician using Autodesk Inventor and Simscape MATLAB is shown in Fig. 4.

Fig. 4 explains the flowchart of making the Dobot Magician model:

1. Part design and continue with the assembly process using the Autodesk Inventor application. The assembly file is saved with the file name DOBOT.iam.
2. Export the DOBOT.iam file by clicking Add_ins and save the export in Simscape to the destination folder, save the file with the extension DOBOT.xml.
3. Open the command window of the MATLAB software and open the Dobot file by typing smimport "DOBOT", this will open a Simulink window that displays the Multibody Simscape diagram of the Dobot model which includes the link body tree and rigid body tree diagram.
4. Setting the joint diagram by adding sliders and gains to the sensing and actuation to control the movement of the Dobot model.
5. Adjust the sensing and actuating values so that the Dobot Model's movement matches the Dobot Magician's actual movement.
6. Create a MATLAB GUI view that represents the rigid body tree controller at all joints of the Dobot model.
7. Kinematics testing based on the model's goal coordinates against the Dobot Magician.

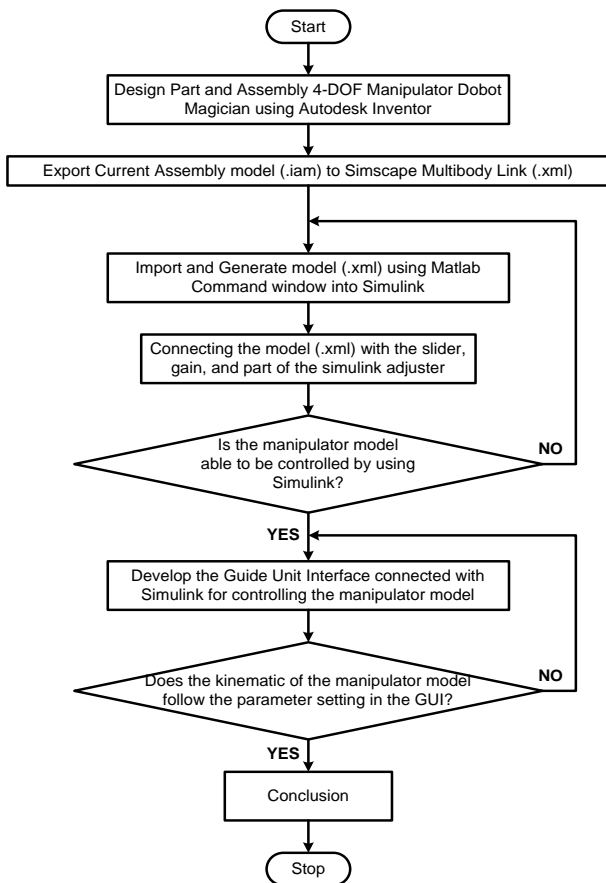


Fig. 4. Flowchart system planning.

A. Autodesk Inventor Dobot Magician Modeling

The main part of Dobot Magician consists of 5 parts: static base, rotation base, shoulder, elbow, and wrist. These five parts are drawn according to the dimensions and shape using Autodesk Inventor and assembled according to the frame coordinate structure on the Dobot Magician model [37, 38]. The following is the assembly structure and frame coordinates of Dobot Magician shown in Fig. 5.

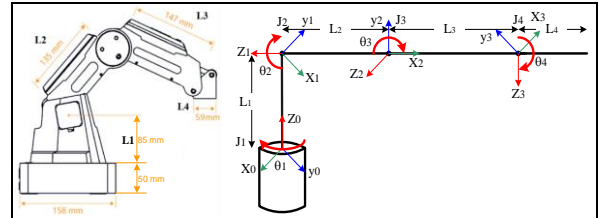


Fig. 5. Structure and frame coordinates of Dobot Magician.

Based on Fig. 5, the Dobot Magician modeling displays 4-frames without End of the Effector on the wrist. The parameters according to Fig. 5 are shown in Table III.

TABLE III. DOBOT MAGICIAN PART MODEL PARAMETERS

Frame i	L_i	α_i	d_i	θ_i
1	0	90°	85	$\theta_1 = -135^\circ \text{ to } 135^\circ$
2	135	0°	0	$\theta_2 = 0^\circ \text{ to } 85^\circ$
3	147	0°	0	$\theta_3 = -10^\circ \text{ to } 95^\circ$
4	59	0°	0	$\theta_4 = 90^\circ \text{ to } -90^\circ$
5*		90°	-70	$\theta_5 = 0^\circ$

* Frame 5 is not installed in the modeling.

Based on Table III, to calculate the forward kinematic is determined by $L_1 = 85 \text{ mm}$, $L_2 = 135 \text{ mm}$, $L_3 = 147 \text{ mm}$ and $L_4 = 59 \text{ mm}$ and θ_i . When θ_i is the motion value generated from the manipulator arm controller slider based on the Dobot Magician characteristic value as in Table IV.

B. Simscape Multibody-Physical Modeling

The Dobot Magician 3D model is imported from Autodesk Inventor into Simscape Multibody MATLAB, resulting in a link model in XML format. This format displays the rigid body tree, which is the body structure and the rigid body joint, which is the joint that connects the rigid body tree. Device control is shown complementing this link model as a regulator of revolute joint movement. The physical model of the Dobot Magician manipulator in Simscape Multibody is shown in Fig. 6.

Fig. 6 shows the overall rigid bodies and rigid body joints of the 4-DOF manipulator model. Five parts of the rigid body tree, namely the base expressed by base_1_RIGID, base rotation expressed by upper1_RIGID, shoulder expressed by upper2_RIGID, elbow expressed by upper3_RIGID and wrist expressed by upper4_RIGID. Two subsystems in the form of Simulink positions control to adjust the angle of movement of the revolute joint that connects rotationally to each rigid body tree. The rigid body tree of Dobot Magician which consists of several parts is described in Table IV.

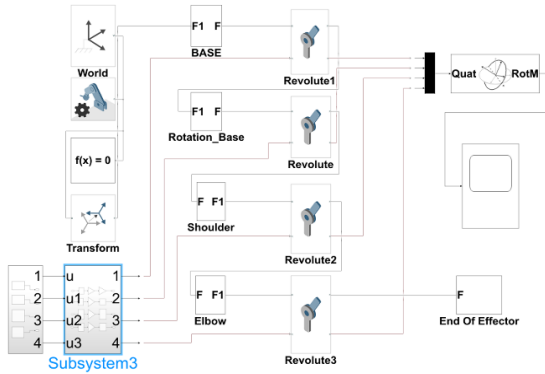


Fig. 6. Physical model of Dobot Magician Manipulator.

From Table IV, there is a world base for each rigid body tree, which is the foundation point for the structure of a robot body model. All joints are revolute that move rotationally relative to the parent frame and define a rotational position in radians around the axis of motion.

TABLE IV. RIGID BODY TREE OF DOBOT MAGICIAN MANIPULATOR

Body Name	Joint Type	Parent Frame	Child Frame
Base	Revolute	Base_1	Upper1_2
Base rotating	Revolute1	Upper1_2	Upper2_2
Shoulder	Revolute2	Upper2_2	Upper_3_rev4
Elbow	Revolute3	Upper3_rev4	Upper_4_1
Wrist	-	Upper_4_1	-

C. Kinematics Control by Graphical User Interface MATLAB

The MATLAB GUI is designed as an interface to control each rigid body joint of the Dobot Magician manipulator and evaluate the position of the End of Effector [39–41]. The rigid body joint controller is presented in the forward kinematic system, and the End of Effector position is evaluated in the Inverse kinematic system. The interface design of the Dobot Magician manipulator model is shown in Fig. 7.

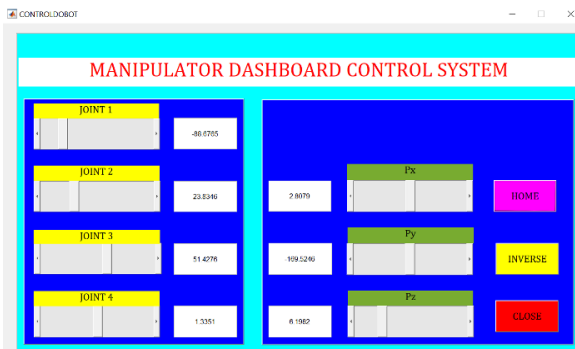


Fig. 7. Kinematics manipulator dashboard control using GUI MATLAB.

The MATLAB GUI will function as the controller of the Dobot Magician manipulator model after running the model on Simscape Multibody. Theta ($\theta_1, \theta_2, \theta_3, \theta_4$) is obtained from inverse kinematics based on the slider parameters in the Simscape Multibody function, while (P_x, P_y, P_z) represents the position of the End of Effector.

IV. RESULT AND DISCUSSING

In this research, several stages of testing were carried out, namely:

1. Modeling joint parameters to determine the trajectory of the end of the effector manipulator Dobot Magician. Visualize the trajectory using MATLAB; this is needed to help analyze the optimization of the Dobot Magician motion control system.
2. Testing the forward kinematics equation by giving each joint a particular joint angle value to obtain X, Y, and Z coordinates at the end of the effector.
3. Testing the inverse kinematics equation by giving a particular value to the X, Y, and Z coordinates to obtain the angular movement value of each Dobot Magician joint.
4. Comparing the results of testing the Dobot Magician model calculated using MATLAB with the value data obtained from Dobot Studio as the main driver of Dobot Magician.

A. Manipulator Movement Trajectory

Referring to the kinematics equations for the 4-DOF manipulators and the parameters of the Dobot Magician, the trajectory graph of the Dobot Magician End-Effector can be visualized using MATLAB programming. This analysis is important to identify the movement route to avoid the risk of collision with other objects around the manipulator area. The sample coordinate values used: X = 47.552, Y = -193.864, and Z = 31.935. The trajectory graphs are displayed in 2-dimensional and 3-dimensional visualizations shown in Figs. 8 and 9.

Fig. 8 shows the 2D motion and Fig. 9 shows the 3D motion modeled using MATLAB based on the arm length parameters and the angle of motion that the Dobot Magician can perform.

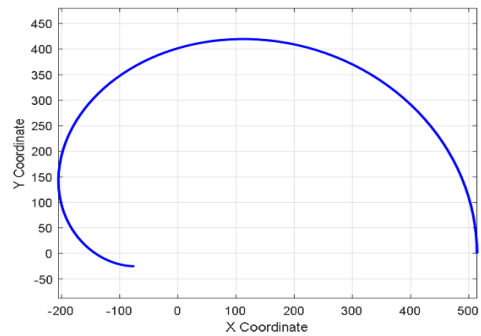


Fig. 8. Dobot Magician 2-dimensional trajectory.

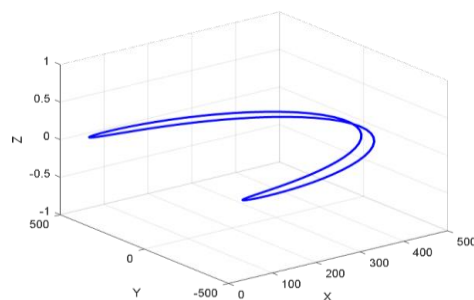


Fig. 9. Dobot Magician 3-dimensional trajectory.

B. Forward Kinematic Approach

The Forward kinematic approach was carried out in this test by determining the value of the joint angle and then entering the equation calculated using MATLAB so that the position (P_x , P_y and P_z) was obtained. The first case uses the theta value parameter, $t_1 = 10$; $t_2 = 60$; $t_3 = 55$; $t_4 = -115$; $t_5 = 0$. The instruction for the forward kinematic MATLAB equation used is as follows:

```
% Enter the DH parameters
a = [0, 135, 147, 59, 70];
alpha = [pi/2, 0, 0, 0, pi/2];
d = [85, 0, 0, 0, 93];
theta = [10, 60, 55, -115, 0];

% Forward kinematics implementation
t1=10; t2=60; t3=55; t5=0; t4=(-t2-t3);
T = eye(4);
for i = 1:5
A1=[cos(t1) -sin(t1) 0 L1*cos(t1) ; sin(t1) cos(t1) 0
L1*sin(t1) ; 0 -1 0 L1 ; 0 0 0 1];
A2=[cos(t2) -sin(t2) 0 L2*cos(t2) ; sin(t2) cos(t2) 0
L2*sin(t2) ; 0 0 1 0 ; 0 0 0 1];
A3=[cos(t3) -sin(t3) 0 L3*sin(t3) ; sin(t3) cos(t3) 0
L3*sin(t3) ; 0 0 1 0 ; 0 0 0 1];
A4=[cos(t4) -sin(t4) 0 L4*sin(t4) ; sin(t4) cos(t4) 0
L4*sin(t4) ; 0 0 1 0 ; 0 0 0 1];
A5=[1 0 0 0 ; 0 1 0 0 ; 0 0 1 0 ; 0 0 0 1];
A=A1*A2*A3*A4*A5;

T = T * A;
end

disp(Transformation Matrices:')
disp(T);

position = T(1:3, 4);
orientation = tform2eul(T);

disp('Position:')
disp(position);
disp('Orientation:')
disp(orientation);
```

The MATLAB program instructions for the forward kinematic equation with the Dobot Magician object and using the joint angle parameter (θ_i) $t_1 = 10$; $t_2 = 60$; $t_3 = 55$; $t_4 = -115$; $t_5 = 0$, then produce a graph of the position and orientation of the end of effector coordinates as shown in Fig. 10.

Based on the MATLAB program instructions and the graph shown in Fig. 10, the following equation results are obtained:

$$\text{Transformation Matrices:} \begin{bmatrix} -0.8391 & -0.5440 & 0 & -10.7634 \\ -0.5440 & 0.8391 & 0 & 103.8582 \\ 0 & 0 & -1 & 182.8296 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Position:
X = -10.7634; Y = 103.8582; Z = 182.8296;

Orientation:
[-2.5664 0 3.1416]

Information about position and orientation helps in complex movement path planning. This is required when the manipulator has to move through a number of points or plan a movement that involves changing position and orientation along a path. By understanding and controlling position and orientation, manipulators can be geared to perform a variety of tasks with the precision, efficiency,

and flexibility required in various industrial or research applications.

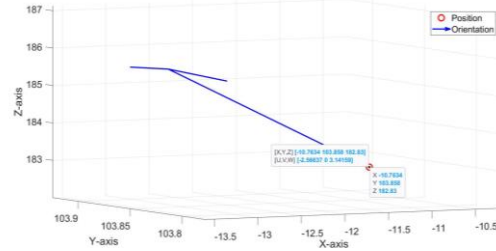


Fig. 10. Graph of position and orientation of the end of the effector with forward kinematic equation approach.

C. Inverse Kinematic Approach

The inverse kinematic approach is performed by modeling the position value obtained based on the forward kinematic approach as the length of the body dimension to calculate the angles ($\theta_1, \theta_2, \theta_3, \theta_4$). For this case, the position parameters used are $P_x = 45$ mm; $P_y = 25$ mm; $P_z = 60$ mm. MATLAB equations are used as follows:

```
target_position = [45, 25, 60];
joint_angles = inverse_kinematics(target_position);
function joint_angles =
inverse_kinematics(target_position)
% DH parameter definition
a = [85, 135, 147, 59];
alpha = [-pi/2, 0, 0, 0];
d = [0, 0, 0, -70];
A = [cos(theta), -sin(theta) * cos(alpha), sin(theta) *
sin(alpha), a * cos(theta);
sin(theta), cos(theta) * cos(alpha), -cos(theta) *
sin(alpha), a * sin(theta);
0, sin(alpha), cos(alpha), d;
0, 0, 0, 1];

% joint angle calculation based on target_position
x = target_position(1);
y = target_position(2);
z = target_position(3);

% Calculate joint angle 1
theta1 = atan2(y, x);

% Calculate joint angle 3
d4 = z - d(1);
r = sqrt(x^2 + y^2);
L = sqrt(r^2 + d4^2);
cosTheta3 = (L^2 - a(2)^2 - a(3)^2) / (2 * a(2) *
a(3));
sinTheta3 = sqrt(1 - cosTheta3^2);
theta3 = atan2(sinTheta3, cosTheta3);

% Calculate joint angle 2
alpha_val = atan2(d4, r);
beta = asin((a(3) * sinTheta3) / L);
theta2 = alpha_val - beta;

% Calculate joint angle 4
theta4 = 0;

% Adjust according to the actual kinematics model

joint_angle = [theta1, theta2, theta3, theta4];
disp('Joint angles:');
disp(joint_angles);

% Visualization of the robot arm
visualize_robot_arm(joint_angles, a, alpha_values,
d);
end
```

Inverse Kinematics analysis using MATLAB on Dobot Magician is very important because it provides an understanding of how to adjust the joint angles of the robot manipulator to achieve a certain position and orientation of the end effector. This analysis is important in planning and controlling the movement of the robot for a specific purpose. With target position $P_x = 45$ mm; $P_y = 25$ mm; $P_z = 60$ mm, the MATLAB inverse kinematic graph is shown in Fig. 11.

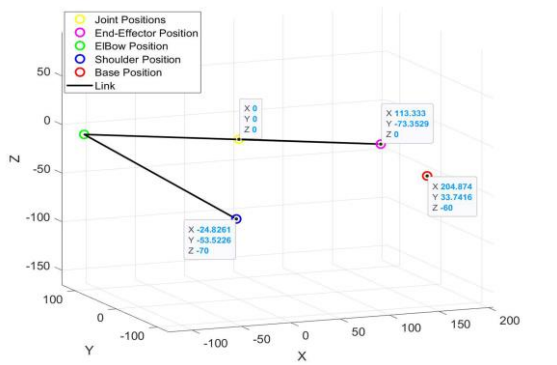


Fig. 11. Graph of joints and links Dobot Magician with Inverse kinematics approach.

Based on the inverse kinematic equation, the Dobot Magician with the specified target position produces Joint angles $(\theta_1, \theta_2, \theta_3, \theta_4)$ as follows:

Joint Angles (rad):

$$\theta_1 = 0.5071; \theta_2 = -0.5744; \theta_3 = 2.5795; \theta_4 = 0$$

Joint Angles (deg):

$$\theta_1 = 29.05^\circ; \theta_2 = -32.9^\circ; \theta_3 = 147.8^\circ; \theta_4 = 0^\circ$$

From the inverse kinematic equations implemented in your MATLAB program, the joint angles results obtained provide information about the configuration of the joints of the robotic manipulator (Dobot Magician) required to achieve a specific position and orientation of the end of the effector. By analyzing the joint angles generated by the inverse kinematic equations, it is possible to identify potential problems, improve the validity of solutions, and ensure that the Dobot Magician behaves in accordance with system constraints.

D. Testing Forward and Inverse Kinematics Approach

The two kinematics equation approaches above provide a more concrete picture that the control modeling of the Dobot Magician manipulator is empirically successful through calculations using MATLAB. Forward kinematic control by giving parameter values $(\theta_1, \theta_2, \theta_3, \theta_4)$ has resulted in Wrist positions $P_x, P_y,$ and P_z at certain values. The inverse kinematic control using the parameters $P_x, P_y,$ and P_z after being calculated based on the model equation results in the value of $(\theta_1, \theta_2, \theta_3, \theta_4)$ at the revolute joint position according to the original position. More complete will be presented in the calculation data table using MATLAB as shown in Table V.

Table V shows that $t_1, t_2,$ and t_3 are values in the revolute joint that have an impact on the position value of the x, y, and z axes through the forward kinematic model equation. While $t_{11}, t_{21},$ and t_{31} are the angular position of the revolute joint after the inverse kinematics mechanism is carried out from the $P_x, P_y,$ and P_z values obtained in the previous forward kinematics equation.

TABLE V. CALCULATION OF FORWARD AND INVERSE KINEMATICS

t_1 (deg)	t_2 (deg)	t_3 (deg)	P_x (mm)	P_y (mm)	P_z (mm)	t_{11} (deg)	t_{21} (deg)	t_{31} (deg)
-90	0	90	0.00	-135.00	232.00	-90	0	90
-70	15	80	40.22	-110.50	266.38	-70	15	80
-50	35	70	46.63	-55.57	304.42	-50	35	70
-30	55	60	13.26	-7.65	328.81	-30	55	60
0	80	50	-71.05	0.00	330.56	0	80	50
20	70	40	-3.86	-1.40	349.99	20	70	40
40	60	30	51.71	43.39	348.91	40	60	30
60	50	20	68.53	118.69	326.55	60	50	20
80	40	10	34.37	194.89	284.38	80	40	10
90	30	0	0.00	244.22	226.00	90	30	0

Based on Table V, the comparison of the initial angular position and final angle of the forward kinematics and inverse kinematics equations is obtained. This comparison can be seen in Figs. 12–14.

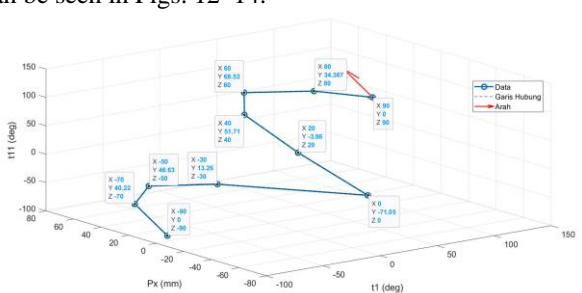


Fig. 12. Comparison of joint positions θ_1 .

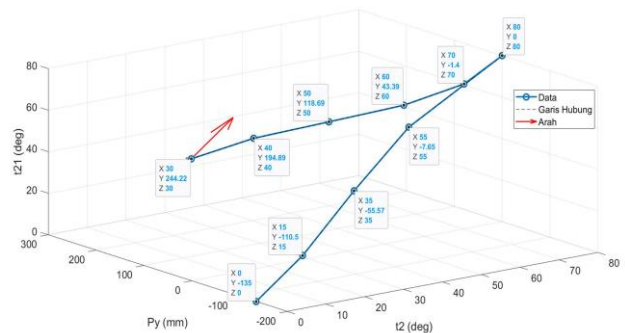


Fig. 13. Comparison of joint positions θ_2 .

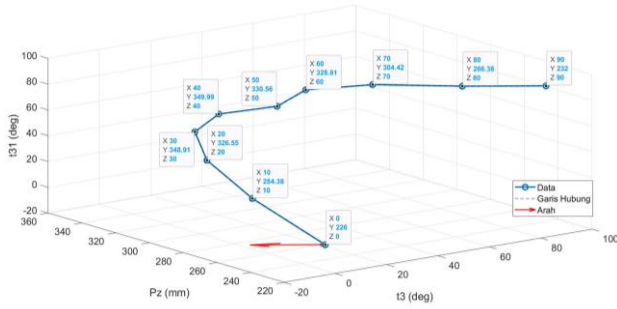


Fig. 14. Comparison of joint positions θ_3 .

Fig. 12 presents a comparison graph of the joint angle at the base rotation with an initial value of t_1 and an end value of t_{11} against the wrist position on the x-axis (P_x). Fig. 13 displays a comparison graph of the joint angle at the shoulder with the initial value of t_2 and the final value of t_{21} against the position of the wrist on the y-axis (P_y). Fig. 14 displays a comparison graph of the joint angle at the elbow with the initial value t_3 and the final value t_{31} against the wrist position on the z-axis (P_z). The expression that can be explained from Figs. 12–14 is:

1. The graph displays X, Y, and Z coordinates where the X and Z coordinates represent the joint angle at the end of the effector and the Y coordinate represents the position of the axis of interest (P_x , P_y , or P_z). it is expressed as in the following program instructions:

```
% Plotting
figure;
plot3(t1, Px, t11, 'o-', 'Linewidth', 2,
'MarkerSize', 8);
title('Grafik Hubungan t1, Px, dan t11');
xlabel('t1 (deg)');
ylabel('Px (mm)');
zlabel('t11 (deg)');
grid on;
```

2. Comparison of angular positions at t_1 , t_2 and t_3 convincingly states that the equations of the Dobot Magician manipulator model are accurate, so the data from the inverse kinematics calculations used as position data for forward kinematics calculations are the same.
3. The dashboard control via MATLAB GUI has represented the arm movement angle and the end position of the effector based on the control system installed on the Simscape Multibody.

E. Comparison of Model with Real Object

In experiments using the MATLAB GUI dashboard, modeling is done by giving joint angles to each joint to get the End of Effector coordinates (x, y, z). Validation of the data resulting from the model controlled by the GUI will be done by comparing to the joint angle data and the position of the end of the effector of the Dobot Magician manipulator displayed by Dobot Studio. The compared parameters is shown in Fig. 15.

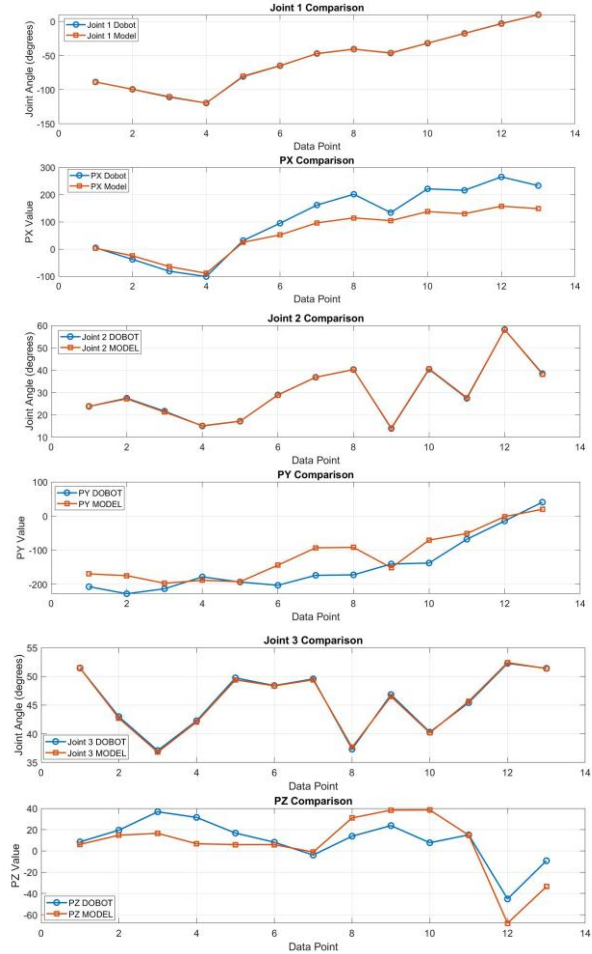


Fig. 15 Comparison graph of joint angle and EoF position.

Fig. 15 is a comparison graph of the joint angle and position of the end effector of the manipulator model with Dobot Magician. This graph is obtained from 12 sample data through the Dobot Magician manipulator joint setting obtained through the teaching playback method, and the dashboard on the MATLAB GUI duplicates the joint position. The two end-off effector position data were compared, and the deviation at each X, Y, and Z coordinate was obtained. On the X-axis, the average error deviation was 36%. On the Y-axis the average error deviation was 31%, and on the Z-axis, the error deviation was 46%.

V. CONCLUSIONS AND RECOMMENDATIONS

Through this work, the control of Dobot Magician robot has been developed for learning purposes. The robot model was designed using Autodesk Inventor while the GUI for parameter setting was achieved using MATLAB. The equations of the forward kinematics and inverse kinematics models in this study have been tested for calculation accuracy, as evidenced by empirical data displayed in tables and graphs of calculation results using MATLAB. By comparing the position deviation between the model coordinates and the real object coordinates of Dobot Magician, the average value for all axes is 37.6%. The differences in position between the actual movement and the movement model in Dobot Magician are referred to as “movement error” or “kinematic error”. This occurs due to

factors that are not read by the model, such as imprecision in mechanical or electronic components, dynamics of drive components and friction that can affect the actual movement, and differences in control system algorithms used in real objects and MATLAB. To overcome this deviation, it can be improved by adding accurate sensors and adjusting mechanical specifications that are oriented toward the precision of object coordinates.

This study is still open to be continued in the next few stages including research by testing the comparison of angular parameters ($\theta_1, \theta_2, \theta_3, \theta_4$) and position parameters (P_x, P_y, P_z) modeling results with Dobot Magician parameters that have been embedded in Dobot Studio. The integration research between MATLAB with microprocessor embedded add-ons such as Arduino and Raspberry Pi is possible so that a method of controlling Dobot Magician is obtained other than using Dobot Studio.

CONFLICT OF INTEREST

The authors declare there are no conflicts of interest.

AUTHOR CONTRIBUTIONS

Emmanuel Agung Nugroho conducted the research; Joga Dharma Setiawan guided by providing research methodology considerations; M. Munadi rechecked all data and journal writing. All authors have approved the final version.

REFERENCES

- [1] M. R. Woodside, J. Fischer, P. Bazzoli, D. A. Bristow, and R. G. Landers, "A kinematic error controller for real-time kinematic error correction of industrial robots," *Procedia Manuf.*, vol. 53, pp. 705–715, 2021. doi: 10.1016/j.promfg.2021.06.069
- [2] K. Paes, W. Dewulf, K. Vander Elst, K. Kellens, and P. Slaets, "Energy efficient trajectories for an industrial ABB robot," *Procedia CIRP*, vol. 15, pp. 105–110, 2014. doi: 10.1016/j.procir.2014.06.043
- [3] Z. Liu, Q. Liu, W. Xu, L. Wang, and Z. Zhou, "Robot learning towards smart robotic manufacturing: A review," *Robot. Comput. Integr. Manuf.*, vol. 77, pp. 102360, April 2022. doi: 10.1016/j.rcim.2022.102360
- [4] S. C. Akkaladevi, M. Plasch, A. Pichler, and M. Ikeda, "Towards reinforcement based learning of an assembly process for human robot collaboration," *Procedia Manuf.*, vol. 38, pp. 1491–1498, 2019. doi: 10.1016/j.promfg.2020.01.138
- [5] K. Peta, J. Włodarczyk, and M. Maniak, "Analysis of trajectory and motion parameters of an industrial robot cooperating with a numerically controlled machine tools," *J. Manuf. Process.*, vol. 101, pp. 1332–1342, 2023. doi: 10.1016/j.jmapro.2023.06.063
- [6] Y. Li and L. Wang, "Kinematic model and redundant space analysis of 4-DOF redundant robot," *Mathematics*, vol. 10, no. 4, Feb. 2022. doi: 10.3390/math10040574
- [7] A. A. Mohammed and M. Sunar, "Kinematics modeling of a 4-DOF robotic arm," in *Proc. International Conference on Control, Automation and Robotics, ICCAR 2015*, Institute of Electrical and Electronics Engineers Inc., Jul. 2015, pp. 87–91. doi: 10.1109/ICCAR.2015.7166008
- [8] R. Arya, J. P. S. Chauhan, and I. Khan, "Kinematic analysis and modelling of a 4-Dof revolute robot manipulator using solidworks," *Int. Res. J. Mod. Eng. Technol. Sci.*, Mar. 2023. doi: 10.56726/irjmets34556
- [9] M. Khairudin, Z. Mohamed, A. R. Husain, and M. A. Ahmad, "Dynamic modelling and characterisation of a two-link flexible robot manipulator," *J. Low Freq. Noise Vib. Act. Control*, vol. 29, no. 3, pp. 207–219, 2010. doi: 10.1260/0263-0923.29.3.207
- [10] S. Gale, H. Rahmati, J. T. Gravidahl, and H. Martens, "Improvement of a robotic manipulator model based on multivariate residual modeling," *Front. Robot. AI*, vol. 4, no. July, pp. 1–17, 2017. doi: 10.3389/frobot.2017.00028
- [11] A. S. Reddy, V. V. M. J. S. Chembuly, and V. V. S. K. Rao, "Modelling and simulation of a redundant agricultural manipulator with virtual prototyping," *J. Robot. Control*, vol. 4, no. 1, pp. 83–94, 2023. doi: 10.18196/jrc.v4i1.17121
- [12] M. Kalyoncu, "Mathematical modelling and dynamic response of a multi-straight-line path tracing flexible robot manipulator with rotating-prismatic joint," *Appl. Math. Model.*, vol. 32, no. 6, pp. 1087–1098, 2008. doi: 10.1016/j.apm.2007.02.032
- [13] M. W. Azhar, "Kinematics modelling of robot manipulator using solidworks," *Univ. Muhammadiyah Surakarta*, vol. 1, 14, 2019.
- [14] L. N. Truc and N. T. Lam, "Quasi-physical modeling of robot IRB 120 using Simscape multibody for dynamic and control simulation," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 28, no. 4, pp. 1949–1964, 2020. doi: 10.3906/ELK-1909-131
- [15] R. S. R. AlNajjar, "Modelling and control of a two-link rigid-flexible," Master thesis, Arab Emirates university College of Engineering Department of Mechanical Engineering, 2021.
- [16] A. N. A. E. Gaber, S. F. Rezeka, and H. A. El Gamal, "Design and position control of arm manipulator; experimentally and in MATLAB sim mechanics," *Int. J. Eng. Res. Technol.*, vol. 5, issue 8, pp. 352–359, 2016. doi: 10.17577/IJERTV5IS080277
- [17] I. V. Krechetov, A. A. Skvortsov, P. S. Lavrikov, and D. V. Yatskin, "Development of an anthropomorphic gripping manipulator: The study of kinematics and virtual modeling of grip," *Am. J. Appl. Sci.*, vol. 13, no. 1, pp. 14–27, 2016. doi: 10.3844/ajassp.2016.14.27
- [18] J. Luis and A. Trujillo, "Kinematic model of the scorbot 4PC manipulator implemented in MATLAB's guide hardware description," vol. 11, no. 4, pp. 183–199, 2018.
- [19] K. Cherifi and K. Hariche, "Design of a MATLAB toolbox and GUI for minimal realizations," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 2, pp. 929–940, 2019. doi: 10.11591/ijeecs.v17.i2.pp929-940
- [20] B. H. Purwoto, "Pemodelan robot kinematik manipulator menggunakan MATLAB," *Emit. J. Tek. Elektro*, vol. 20, no. 2, pp. 141–146, 2020. doi: 10.23917/emit.v20i02.11345
- [21] S. H. B. Ishak, "Design of robotic arm controller using MATLAB," Bachelor thesis, Universiti Teknikal Malaysia Melaka, April, 2011.
- [22] S. M. Vallinayagam, "4A GUI based kinematic model development of 6 DOF manipulator using MATLAB," *Int. J. Eng. Dev. Res.*, vol. 6, pp. 2321–9939, 2019.
- [23] P. S. Tsai, T. F. Wu, J. Y. Chen, and F. H. Lee, "Drawing system with dobot magician manipulator based on image processing," *Machines*, vol. 9, no. 12, Dec. 2021. doi: 10.3390/machines-9120302
- [24] Dobot Magician User Manual. (2018). 1.2.5. pp. 1–135. [Online]. Available: www.dobot.cc
- [25] R. Islam, A. Rahaman, and M. Habibur, "Cartesian trajectory based control of dobot robot," *IEOM Soc. Int.*, no. 2017, pp. 1507–1517, 2019.
- [26] J. D. Sanjuan De Caro, M. Rahman, and I. Rulik, "Forward kinematic analysis of Dobot using closed-loop method," *IAES Int. J. Robot. Autom.*, vol. 9, no. 3, p. 153, Sep. 2020. doi: 10.11591/ijra.v9i3.pp153-159
- [27] M. R. Sheikh, "Trajectory tracking of a four degree of freedom robotic manipulator," Master thesis, The University of Wisconsin, 2019.
- [28] E. A. Nugroho, N. R. Wibowo, and R. Rizalludin, "Fuzzy system as four-joint robot movement control for moving goods based on time and," *Jurnal Ramatekno*, vol. 2, no. 2, pp. 7–15, 2022.
- [29] O. Hock and J. Šedo, "Forward and inverse kinematics using pseudoinverse and transposition method for robotic arm dobot," in *Kinematics*, E. G. Hurtado, Ed. IntechOpen, 2017. doi: 10.5772/intechopen-71417
- [30] M. R. Sheikh. (2019). Trajectory tracking of a four degree of freedom robotic trajectory tracking of a four degree of freedom robotic manipulator manipulator. [Online]. Available: https://dc.uwm.edu/etd/2252
- [31] M. N. Wai, T. T. Htoo, and L. Maw, "4-DOF robot arm for pick and place process," *Int. J. Adv. Sci. Res. Eng.*, vol. 6, no. 11, pp. 55–64, 2020. doi: 10.31695/ijasre.2020.33927
- [32] A. Widyacandra, A. R. Al Tahtawi, and M. Martin, "Forward and inverse kinematics modeling of 3-DoF AX-12A robotic

- manipulator,” *JITEL (Jurnal Ilm. Telekomun. Elektron. dan List. Tenaga)*, vol. 2, no. 2, pp. 139–150, Sep. 2022. doi: 10.35313/jitel.v2.i2.2022.139-150
- [33] A. Renfrew, “Book review: Introduction to robotics: Mechanics and control,” *Int. J. Electr. Eng. Educ.*, vol. 41, no. 4, pp. 388–388, 2004. doi: 10.7227/ijee.41.4.11
- [34] H. Zhong Ting, M. Hairi Mohd Zaman, M. Faisal Ibrahim, and A. Mohamed Moubark. Kinematic analysis for trajectory planning of open-source 4-DoF robot arm. [Online]. Available: www.ijacsa.thesai.org
- [35] T. Q. Nguyen V.-T. Phan, D.-T. Vo, V.-H. Trinh, H.-V. Nguyen, M.-S. Tran, and D.-T. Tran, “Kinematics, dynamics and control design for a 4-DOF robotic manipulator,” in *Proc. 2021 International Conference on System Science and Engineering, ICSSE 2021*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021, pp. 138–144. doi: 10.1109/ICSSE52999.2021.9538418
- [36] J. Alexandre Gonçalves Manuel Braz-César João Paulo Coelho Editors. *Lecture Notes in Electrical Engineering*. 695. [Online]. Available: <http://www.springer.com/series/7818>
- [37] A. N. Ghaniwala, M. A. Khan, M. M. Akhtar, A. Shaikh, M. Affan, and R. Uddin. (2021). Analytical Kinematic analysis of Multi-DOF serial link robot arm. in *Proc. 2021 6th International Electrical Engineering Conference*. [Online]. Available: https://ieec.neduet.edu.pk/2021/papers_2021/IEEC_2021_34.pdf
- [38] P. Jha, “Inverse kinematic analysis of robot manipulators a thesis submitted in fulfilment of the requirement for the award of the degree,” Doctoral dissertation, National Institute of Technology Rourkela, India, 2015. <https://core.ac.uk/download/pdf/80147797.pdf>
- [39] S. Zodey and S. K. Pradhan, “MATLAB toolbox for kinematic analysis and simulation of dexterous robotic grippers,” *Procedia Eng.*, vol. 97, pp. 1886–1895, 2014. doi: 10.1016/j.proeng.-2014.12.342
- [40] A. M. San and W. P. Ei. (2014). Interface control system for 3-DOF parallel kinematic manipulator. [Online]. 3(19). pp. 3929–3933. Available: <https://ijsetr.com/uploads/235461IJSETR1816-678.pdf>
- [41] D. Sivasamy, M. Dev Anand, and K. Anitha Sheela, “Robot forward and inverse kinematics research using MATLAB,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 2, special issue 3, pp. 29–35, 2019. doi: 10.35940/ijrte.B1006.0782S319

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License ([CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.