Kinematics Analysis of Four Degree-of-Freedom Articulated Manipulator Using MATLAB GUI

Ther Htet Aung[#1], Wai Mar Myint[#2]

#1,2 Mechatronic Department, Mandalay Technological University

[#1]therhtetaung4@gmail.com
[#2]mcwaimar@gmail.com

***Abstract* - This paper presents the design, development, and control of a four Degree-of-Freedom (DOF) articulated manipulator utilizing a MATLAB graphical user interface (GUI) and an Arduino UNO microcontroller. Denavit-Hartenberg (D-H) analysis was employed to model the forward and inverse kinematics of the manipulator. MATLAB was used to verify the end-effector position and compute the joint angles using the developed algorithms. A user-friendly GUI was created in MATLAB to provide intuitive interaction with the manipulator, allowing users to adjust joint angles, visualize movements, and execute predefined tasks. The GUI communicated with the Arduino microcontroller, which controlled the servo motors attached to the manipulator's joints. Despite the successful implementation, several limitations were encountered, including potential inaccuracies in the kinematic model due to imprecise D-H parameters and communication latency between MATLAB and Arduino. Future research could focus on addressing these limitations to enhance the manipulator's accuracy and reliability.**

***Keywords* -D-H Parameters, Four-DOF Articulated Manipulator, Forward Kinematics, MATLAB GUI, Inverse Kinematics, Robotics Toolbox**

I. INTRODUCTION

The rapid advancement of robotics technology has impacted many industries, from manufacturing and healthcare to construction and space exploration. Articulated manipulators, known for their flexibility and ability to mimic human arm movements, are increasingly used for tasks requiring precision. These robots consist of interconnected joints and links that can perform complex motions. This paper focuses on the design, simulation, and control of a four Degrees-of-Freedom articulated manipulator. Both forward and inverse kinematics are used to examine the movement of the manipulator. MATLAB is used for mathematical modelling, and AutoCAD is used for detailed mechanical design. Inverse kinematics computes the joint parameters required for a given end-effector position, whereas forward kinematics uses joint parameters to determine the end effector's location.

A key challenge in this research is solving the kinematic equations that guide the manipulator's movements. These equations, which relate the joint angles of the manipulator to its end-effector position and orientation, are inherently nonlinear and can exhibit multiple solutions or singularities. Additionally, developing a control system that allows real-time control, supported by a MATLAB graphical interface (GUI), is critical for practical use. This interdisciplinary mechanical design and dynamic simulation, making the system useful for both research and industry. Figure 1 shows the overall block diagram of the system.

Figure 1. Block Diagram of Four DOF Articulated Manipulator

II. DESIGN OF FOUR DEGREE OF FREEDOM ARTICULATED MANIPULATOR

A 4-DOF articulated manipulator is a robotic arm with four degrees of freedom, allowing it to perform complex tasks by moving and positioning objects in 3D space. These manipulators are commonly used in industrial automation and research due to their versatility and relatively simple design compared to higher-DOF systems.

The design and construction of a 4-DOF articulated manipulator involve integrating multiple rotating joints, each contributing one degree of freedom. Typically, these joints are driven by servo motors, allowing precise control over the arm's movement. 3D printing technology has made it more accessible to design and create custom robotic arms, as it allows for rapid prototyping and customization of parts. By using CAD software, one can design each component, print it with PLA materials, and then assemble the manipulator.

Once assembled, the manipulator can be controlled using an Arduino Uno, which drives the servo motors. After implementing control algorithms, including forward and inverse kinematics, enables the arm to perform specific tasks, such as picking and placing objects. The 3D-printed 4-DOF articulated manipulator is shown in Figure 2.

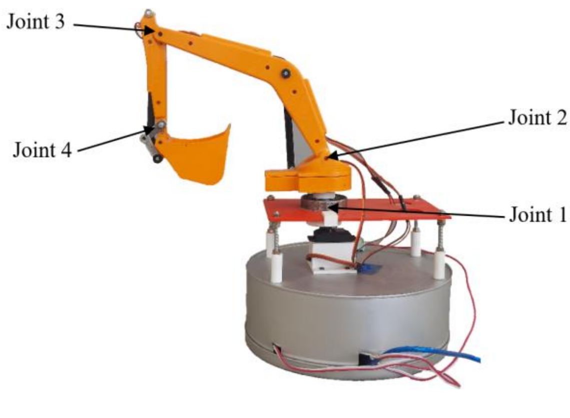


Figure 2. 3D-Printed 4-DOF Articulated Manipulator

This 4-DOF manipulator manages to outperform larger counterparts and offers features not found elsewhere. The robot is portable and easily integrated. This compact four-axis can handle a payload of 0.5 kg and can reach a height of 366 mm, making it suitable for light pick and place, machine tending, laboratory work, training, and demonstrations. The robot motion speed and motion range are shown in Table 1.

TABLE 1. SPEED AND RANGE OF ROBOTIC MOTION

Joint No.	Range	Speed	Type of Motion
Joint 1	+90° to -90°	5.24 rad/s	Rotation Motion
Joint 2	+15° to	10.47 rad/s	Arm Motion
Joint 3	-60° to -	10.47 rad/s	Arm Motion
Joint 4	-25° to -90°	10.47 rad/s	Arm Motion

The design incorporates four degrees of freedom (4-DOF), providing a versatile range of motion that enhances its functionality in various applications. Figure 3 illustrates the axes and link dimensions of the 4-DOF articulated manipulator, detailing the geometric relationships and dimensions that define its range of motion.

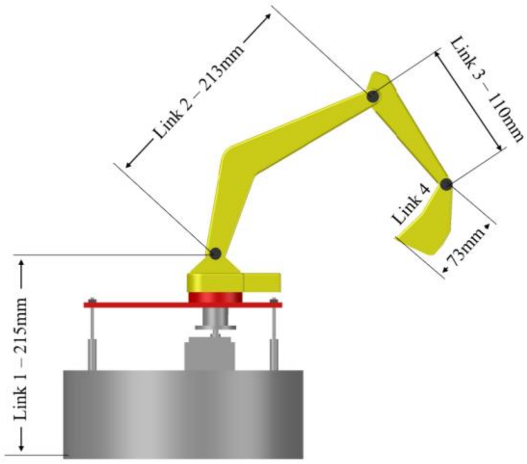


Figure 3. Dimensions of 4-DOF Articulated Manipulator

In addition, Figure 4 showcases the compact design of the manipulator, highlighting its efficient use of space and the elegance of its articulated structure. The manipulator's design and construction facilitate precise control and effective handling, making it suitable for complex tasks requiring both speed and accuracy.

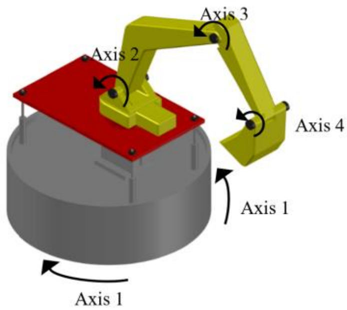


Figure 4. Axes Of 4-DOF Articulated Manipulator

III. KINEMATIC MODELING OF 4-DOF ARTICULATED MANIPULATOR

In robot systems, kinematic modeling plays an essential role in the design and implementation of robot arm controls. The kinematic analysis of the robot arm deals with the relationships between the end-effector's variables (i.e., the position and orientation) and joint variables (i.e., the rotational and/or translational displacements). Although forward kinematic equations describe the position and orientation of the end-effector in terms of the given joint variables and link lengths and corresponding inverse kinematic equations express the joint variables in terms of the position and orientation of the end-effector and link lengths.

A. Forward Kinematics Analysis

Forward or direct kinematics determines the position and orientation of the end-effector with respect to the base of the robotic arm as a function of joint angles. The Denavit-Hartenberg (DH) parameter is presented to build the homogeneous transformation matrices between the robot joint axes. The illustration of how frame {i} is related to the previous frame {i-1} and the description of the frame parameters are shown in Figure 5. The four parameters of the classic D-H convention are shown in Figure 3.14, which are (θ_i, a_i, d_i, α_i). It can be translated the coordinates from O_{i-1}, X_{i-1}, Y_{i-1}, Z_{i-1} to O_i, X_i, Y_i, Z_i using those four parameters.

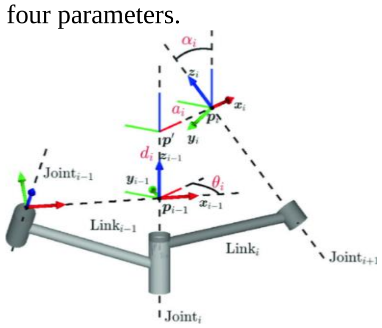


Figure 5. The Description of Frame {i} with respect to Frame {i-1}

The four transformations between the two axes can be defined using the Denavit-Hartenberg as:

$$
\begin{aligned}
{}^{i-1}T_i &= \text{Rot}(z, \theta_i).\ \text{Trans}(z, d_i).\ \text{Trans}(x, d_i).\ \text{Rot}(x, \alpha_i) \\
&= \begin{bmatrix} \cos\theta_i & -\sin\theta_i \,.\, \cos\alpha_i & \sin\theta_i \,.\, \sin\alpha_i & a_i \,.\, \cos\theta_i \\ \sin\theta_i & \cos\theta_i \,.\, \cos\alpha_i & -\cos\theta_i \,.\, \sin\alpha_i & a_i \,.\, \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \quad (1)
$$

The frame assignment for the 4-DOF articulated manipulator is illustrated in Figure 6, which provides a visual representation of the coordinate frames at each joint. The D-H coordinate system is established for each link, as detailed in Table 2, and a homogeneous transformation matrix is developed, which is fundamental in transforming the coordinates of one frame to the next, effectively linking the motion between frame {i-1} and frame {i}. By applying this transformation, the relative positions and orientations of the manipulator's links can be systematically calculated.

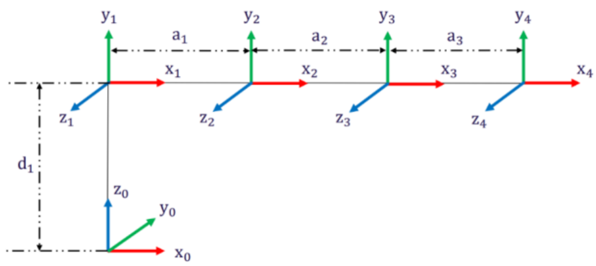


Figure 6. Frame Assignment of Four Degree-of-Freedom Manipulator

TABLE 2. D-H PARAMETERS OF 4-DOF ARTICULATED MANIPULATOR

Axis (i)	Joint Angle	Joint Offset	Link Length (a_i)	Twist Angle (α_i)
1	θ_1	d_1	0	90°
2	θ_2	0	a_1	0
3	θ_3	0	a_2	0
4	θ_4	0	a_3	0

The final homogeneous transformation of the 4-DOF articulated manipulator robot is calculated via the product of the basic transformations as:

$$^0T_4 = {}^0T_1\, {}^1T_2\, {}^2T_3\, {}^3T_4 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} & {}^0R_4 & & {}^0P_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

B. Inverse Kinematics Analysis

For the inverse kinematics problem, the end-effector's location and orientation are given, and the problem is to find the joint angles θ_i, i = 1, 2, 3, 4, necessary to bring the end-effector to the desired location. Contrary to the forward kinematics, the end-effector orientation and pose are always unique for the same set of joint position variable values. The inverse kinematic model of the 4-DOF articulated manipulator can be summarized as follows:

$$\theta_1 = \text{atan2}\,(p_y, p_x) \quad (3)$$

$$\phi = \theta_2 + \theta_3 + \theta_4 \quad (4)$$

$$A = p_x - (a_3 .\cos\phi .\cos\theta_1) \quad (5)$$

$$B = p_y - (a_3 .\cos\phi .\sin\theta_1) \quad (6)$$

$$C = p_z - d_1 - (a_3 .\sin\phi) \quad (7)$$

$$\theta_3 = \pm \cos^{-1}\left(\frac{A^2 + B^2 + C^2 - a_1^2 - a_2^2}{2a_1a_2}\right) \quad (8)$$

$$a = a_2 .\sin\theta_3 \quad (9)$$

$$b = a_1 + a_2 .\cos\theta_3 \quad (10)$$

$$\theta_2 = \tan^{-1}\left(\frac{C}{\sqrt{a^2 + b^2 - C^2}}\right) - \tan^{-1}\left(\frac{a}{b}\right) \quad (11)$$

$$\theta_4 = \phi - \theta_2 - \theta_3 \quad (12)$$

IV. WORKSPACE AND GUI DEVELOPMENT OF 4-DOF ARTICULATED MANIPULATOR

The workspace is a critical aspect of the manipulator's design, representing the volume within which the end-effector (the part of the manipulator that interacts with the environment) can operate. The 3D workspace of a 4-DOF articulated manipulator for this system is visualized in a three-dimensional coordinate system with X, Y, and Z axes as shown in Figure 7. The different colors in the plot correspond to the orientation of the end-effector, measured in degrees, as indicated by the color bar on the right side of the image.

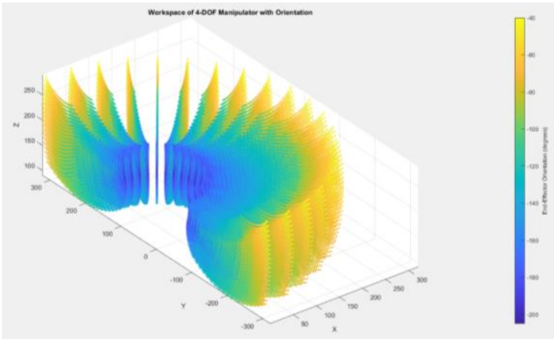
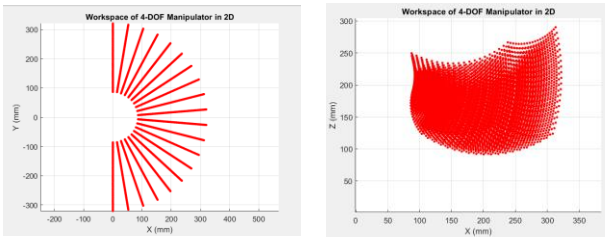


Figure 7. Three-Dimensional Workspace of the 4-DOF Articulated Manipulator

The shape of the workspace is determined by the range of motion of each joint in the manipulator, as well as the combined effect of these movements. In this case, the workspace appears as a complex, symmetric shape with multiple lobes, indicating that the manipulator can reach various positions in space with different end-effector orientations. The outer boundaries of the workspace, as projected onto the XY-Plane and XZ-Plane, are shown in Figure 8.



(A) (B)

Figure 8. Outer Boundary of Workspace in (A) XY-Plane (B) XZ-Plane

Peter Corke's Robotic Toolbox is used for graphical user interface (GUI) control for 4-DOF articulated manipulator. The graphical user interface (GUI) for controlling the manipulator is shown in Figure 9.

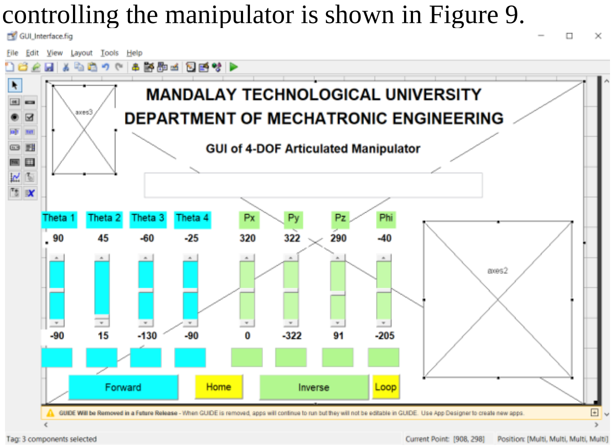


Figure 9. GUI for Manipulator

V. EXPERIMENTAL RESULTS

Firstly, the geometric model of the 4-DOF articulated manipulator is validated in MATLAB. After finding the homogeneous matrix 0T4 that describes the end-effector position and orientation with respect to the robot's global coordinate frame, the position and orientation of the robot, which gives the position vector p, and the orientation of the end-effector as in equations 13, 14, 15 and 16 are as follows:

$$p_x = \cos\theta_1 . a_2 . \cos(\theta_2+\theta_3) + a_1 . \cos\theta_2 + a_3 . \cos(\theta_2+\theta_3+\theta_4) \quad (13)$$

$$p_y = \sin\theta_1 . a_2 . \cos(\theta_2+\theta_3) + a_1 . \cos\theta_2 + a_3 . \cos(\theta_2+\theta_3+\theta_4) \quad (14)$$

$$p_z = d_1 + a_2 . \sin(\theta_2+\theta_3) + a_1 . \sin\theta_2 + a_3 . \sin(\theta_2+\theta_3+\theta_4) \quad (15)$$

$$\phi = \theta_2+\theta_3+\theta_4 \quad (16)$$

Given: d_1 = 215 mm, a_1 = 213 mm, a_2 = 110mm and a_3 = 73 mm.

The above equations are programmed in MATLAB, allowing for precise computation of the forward kinematic model. To validate this model, a set of joint angles were selected, ensuring that the computations accurately reflect the intended robotic movements. The final computed matrices, derived from these joint angles, provide detailed information on both the positions and orientations within the robotic system.

These matrices were computed for several distinct robot positions, including the "Home" position, as well as "Position-1" and "Position-2." For each of these positions, the joint angles were manually inputted by the user into the MATLAB script. This manual input process enabled the calculation of the p_x, p_y, and p_z vectors, which represent the precise coordinates in 3D space. The results of these calculations are documented in Tables 3, 4 and 5.

TABLE 3. THE COMPUTED HOME POSITION OF THE FORWARD KINEMATIC MODEL IN MATLAB

Entered Joint Angles		Calculated Positions and Orientations		Transformation Matrix ($^0T_4 = {^0T_1}.{^1T_2}.{^2T_3}.{^3T_4}$)			
θ_1	0	p_x	308	0.9997	0.0213	0	308
θ_2	15	p_y	-1	0	0	-1	-1
θ_3	-60	p_z	123	-0.0213	0.9997	0	123
θ_4	-25	ϕ	-70	0	0	0	1

TABLE 4. THE COMPUTED POSITION (POSITION-1) OF THE FORWARD KINEMATIC MODEL IN MATLAB

Entered Joint Angles		Calculated Positions and Orientations		Transformation Matrix ($^0T_4 = {^0T_1}.{^1T_2}.{^2T_3}.{^3T_4}$)			
θ_1	90	p_x	0	0	0	1	0
θ_2	45	p_y	87	-0.9999	0.0015	0	87
θ_3	-130	p_z	249	-0.0015	-0.9999	0	249
θ_4	-90	ϕ	-176	0	0	0	1

TABLE 5. THE COMPUTED POSITION (POSITION-2) OF THE FORWARD KINEMATIC MODEL IN MATLAB

Entered Joint Angles		Calculated Positions and Orientations		Transformation Matrix ($^0T_4 = {^0T_1}.{^1T_2}.{^2T_3}.{^3T_4}$)			
θ_1	45	p_x	155	-0.1830	0.6830	0.7071	155
θ_2	30	p_y	155	-0.1830	0.6830	-0.7071	155
θ_3	-90	p_z	155	-0.9659	-0.2588	0	155
θ_4	-45	ϕ	-105	0	0	0	1

The positions and orientations of the robot were selected to validate the inverse kinematic solution. By applying the inverse kinematic equations, a set of joint angles was obtained. These calculations were carried out using MATLAB, which resulted in two sets of solutions. The joint angles corresponding to the "Home" position are detailed in Table 6, while the solutions for "Position-1" and "Position-2" are presented in Tables 8 and 10, respectively. Moreover, the expected and obtained results for inverse kinematics are shown in Table 7 (for "Home" position), Table 9 (for "Position-1) and Table 11 (for "Position-2).

TABLE 6. COMPUTED RESULTS OF THE INVERSE KINEMATIC MODEL IMPLEMENTED IN MATLAB FOR HOME POSITION (P_X = 308, P_Y = -1, P_Z = 123 AND Φ = -70)

Compared Joint Angles	θ_1	θ_2	θ_3	θ_4
Joint Limitation	+90, -90	+15, +45	-60, -130	-25, -90
Solution 1	0	15	-60	-25
Solution 2	0	-24.14	60	-105.86

TABLE 7. THE DESIRED JOINT ANGLES AND COMPUTED JOINT ANGLES USING INVERSE KINEMATIC IN MATLAB FOR HOME POSITION

Position and Orientation of Home Position		Desired Joint Angles for Home Position		Computed Joint Angles using Inverse Kinematic	
p_x	308	θ_1	0	θ_1	0
p_y	-1	θ_2	15	θ_2	15
p_z	123	θ_3	-60	θ_3	-60
ϕ	-70	θ_4	-25	θ_4	-25

TABLE 8. COMPUTED RESULTS OF THE INVERSE KINEMATIC MODEL IMPLEMENTED IN MATLAB FOR POSITION-1 (P_X = 0, P_Y = 87, P_Z = 249 AND Φ = -176)

Compared Joint Angles	θ_1	θ_2	θ_3	θ_4
Joint Limitation	+90, -90	+15, +45	-60, -130	-25, -90
Solution 1	90	45	-130	-90
Solution 2	90	-17.96	122	-279.04

TABLE 9. THE DESIRED JOINT ANGLES AND COMPUTED JOINT ANGLES USING INVERSE KINEMATIC IN MATLAB FOR POSITION-1

Position and Orientation of Position-1		Desired Joint Angles for Position-1		Computed Joint Angles using Inverse Kinematic	
p_x	0	θ_1	90	θ_1	90
p_y	87	θ_2	45	θ_2	44.29
p_z	249	θ_3	-130	θ_3	-129.72
ϕ	-176	θ_4	-90	θ_4	-90.85

TABLE 10. COMPUTED RESULTS OF THE INVERSE KINEMATIC MODEL IMPLEMENTED IN MATLAB FOR POSITION-2 (P_X = 155, P_Y = 155, P_Z = 155 AND Φ = -105)

Compared Joint Angles	θ_1	θ_2	θ_3	θ_4
Joint Limitation	+90, -90	+15, +45	-60, -130	-25, -90
Solution 1	45	29.46	-87.67	-46.78
Solution 2	45	-24.17	87.67	-168.5

TABLE 11. THE DESIRED JOINT ANGLES AND COMPUTED JOINT ANGLES USING INVERSE KINEMATIC IN MATLAB FOR POSITION-2

Position and Orientation of Position-2		Desired Joint Angles for Position-2		Computed Joint Angles using Inverse Kinematic	
p_x	155	θ_1	45	θ_1	45
p_y	155	θ_2	30	θ_2	29.46
p_z	155	θ_3	-90	θ_3	-87.67
ϕ	-105	θ_4	-45	θ_4	-46.78

First, the kinematic model of the 4-DOF articulated manipulator is loaded into the MATLAB GUI. Users can input joint angles or specify the end-effector's position through text boxes or sliders. By pressing the "Forward" or "Inverse" button, the system computes either joint angles or end-effector position, depending on the selected operation. The GUI provides real-time visual feedback and error-checking, alerting users if inputs exceed the robot's workspace or joint limits. Sliders offer an interactive way to adjust joint angles, providing instant visual updates as the manipulator moves in real-time. Additionally, the interface includes a "Home" push button commands the manipulator to return to its predefined home position, set at "$\theta_1 = 0$, $\theta_2 = 15$, $\theta_3 = -60$, and $\theta_4 = -25$." The MATLAB GUI displaying the manipulator in its home position is shown in Figure 10.

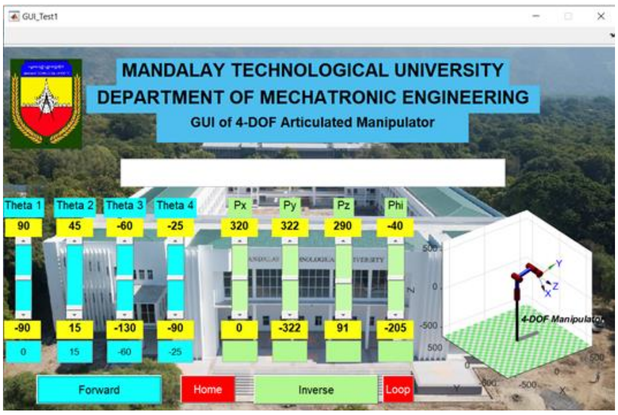


Figure 10. MATLAB GUI of Home Position

The MATLAB GUI display of the manipulator for “Position-1” is shown in Figure 11. When the joint angles $\theta_1 = 90$, $\theta_2 = 45$, $\theta_3 = -130$ and $\theta_4 = -90$ are applied, the end-effector’s position and orientation of the “Position-1” of the 4-DOF articulated manipulator is measured as $p_x = 1$ mm, $p_y = 92$ mm, $p_z = 244$ mm and $\phi = -176°$.

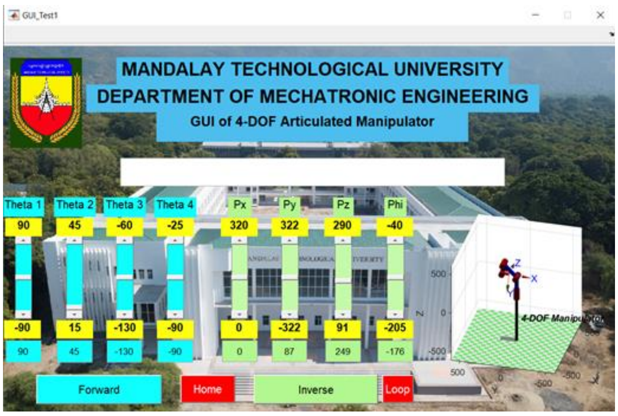


Figure 11. MATLAB GUI of Position-1

The MATLAB GUI of “Position-2” is shown in Figure 12. When the angles $\theta_1 = 45$, $\theta_2 = 30$, $\theta_3 = -90$ and $\theta_4 = -45$ are applied, the measured results of the “Position-1” of the 4-DOF articulated manipulator is $p_x = 158$ mm, $p_y = 158$ mm, $p_z = 164$ mm and $\phi = -104°$.

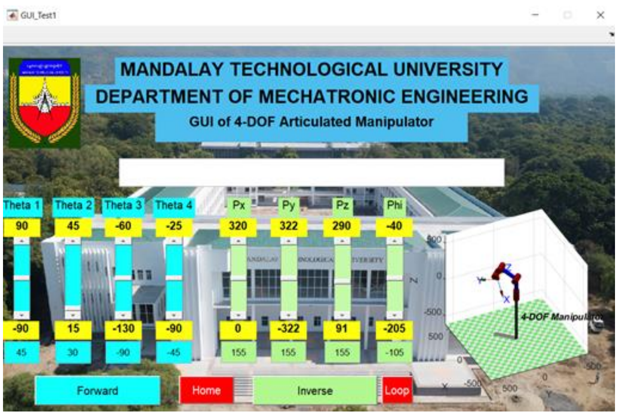


Figure 12. MATLAB GUI of Position-2

VI. CONCLUSIONS

A 4-DOF articulated manipulator is a robotic arm with four independent movements that allow it to position and orient objects in space for tasks like picking, placing, or assembly. The calculation process in this paper includes both the DH transformation matrix and inverse kinematics. The DH transformation matrix is used to determine the position of the manipulator, while inverse kinematics is used to calculate the joint angles based on position. To ensure the reliability of the system, the model is tested with both forward and inverse kinematics. The calculation results are compared with experimental results to validate the accuracy and performance of the system. The main strengths are its systematic approach and practical implementation. This research shows the advantage of using MATLAB GUI in controlling a 4-DOF articulated manipulator is the user-friendly interface, which simplifies the management of complex robotic operations.

ACKNOWLEDGEMENTS

Firstly, the author wishes to express her sincere gratitude to Dr. San Yu Khaing, Pro-Rector, Mandalay Technological University, for granting the invaluable permission necessary to undertake this paper. The author extends his deepest appreciation to Dr. Wut Yi Win, Professor and Head, Department of Mechatronic Engineering, Mandalay Technological University, for her guidance, invaluable suggestions, shared experiences. The author is sincerely grateful to his supervisor, Dr. Wai Mar Myint, Associate Professor, Department of Mechatronic Engineering, Mandalay Technological University, for her valuable suggestion, sharing experience and correction on this paper as well as her willingness to share ideas for this paper. The author especially appreciates and thanks all his teachers for their supports and guidance.

REFERENCES

[1] Senthilkumar & Parthiban, 2016, Modeling, Kinematic Analysis and Simulation of 4-DOF TRLR Polar Robot Manipulator.

[2] Payal Agnihotri, 2015, ANFIS Based Forward and Inverse Kinematics of Robot Manipulator with Five Degrees of Freedom.

[3] Ravirkumar Mourya, Amit Shelke, Sourabh Satpute, Sushant Kakade and Manoj Botre, 2015, Designed and Implemented Four Degrees of Freedom Articulated Robotic Arm.

[4] Arian Faravar, 2014, Design Implementation and Control of a Robotic Arm Used for All Terrain Robot.

[5] K. J. Parekh, 2012, Kinematic Analysis of 3-DOF R-R-R Type Manipulator Arm.

[6] Ei Thwe Thwe Theint, 2023, Validation of Geometric Model for ABB IRB 120 Robot Using MATLAB and ABB RobotStudio Software.