

Forward and inverse kinematics modeling of 3-DoF AX-12A robotic manipulator

Ayu Widyacandra^{1*}, Adnan Rafi Al Tahtawi², Martin³

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Bandung

Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat, Indonesia

^{1*}ayu.widyacandra.toi18@polban.ac.id, ²adnan.rafi@polban.ac.id, ³martin@polban.ac.id

ABSTRACT

The presence of robots that can assist humans with heavy or dangerous work makes the need for robots more pressing at the moment. One type of robot needed is a robot arm, which is widely used in the manufacturing industry, such as in the assembly process and pick and place. The types of robotic arms used vary both in terms of configuration and the number of degrees of freedom. However, with different types of robotic arms, different models of movement are used. Therefore, research related to the modeling of the robotic arm continues to be carried out to obtain the appropriate movement of the robotic arm. One of the methods used as a first step in designing a robotic arm movement model is kinematics analysis. Kinematics analysis aims to analyze the movement of the robot arm without knowing what force causes the movement. This paper aims to produce an ideal movement model for the AX-12A 3-DoF robotic arm using forward kinematic and inverse kinematic analysis using two methods, the Denavit-Haterberg method and the geometric approach method. The difference from other papers is that this paper makes the kinematics model using Robotic, Vision, and Control (RVC) tools based on the Peter I. Corke model on MATLAB software first before implementing it on hardware. The results show that the error percentage for the forward kinematic model is 1.04% and the inverse kinematic is 0.76%, which means the two models achieved the target that the model's error maximum must be less than 2%.

Keywords: robot arm, forward kinematic, inverse kinematic, MATLAB

ABSTRAK

Kehadiran robot yang dapat membantu manusia dalam pekerjaan yang berat atau berbahaya membuat kebutuhan akan robot semakin mendesak saat ini. Salah satu jenis robot yang dibutuhkan adalah lengan robot, yang banyak digunakan dalam industri manufaktur, seperti dalam proses perakitan dan untuk mengambil dan menempatkan barang. Jenis lengan robot yang digunakan bervariasi baik dari segi konfigurasi maupun jumlah derajat kebebasannya. Namun, dengan berbagai jenis lengan robot, model gerakan yang berbeda digunakan. Oleh karena itu, penelitian terkait pemodelan lengan robot terus dilakukan untuk mendapatkan gerakan lengan robot yang tepat. Salah satu metode yang digunakan sebagai langkah awal dalam merancang model gerakan lengan robot adalah analisis kinematika. Analisis kinematika bertujuan untuk menganalisis pergerakan lengan robot tanpa mengetahui gaya apa yang menyebabkan pergerakan tersebut. Makalah ini bertujuan untuk menghasilkan model gerakan ideal lengan robot AX-12A 3-DoF menggunakan analisis kinematika maju dan kinematik terbalik menggunakan dua metode, yaitu metode Denavit-Haterberg dan metode pendekatan geometrik. Perbedaan dari makalah lain adalah makalah ini membuat model kinematika menggunakan tools Robotic, Vision, and Control (RVC) berdasarkan model Peter I. Corke pada software MATLAB terlebih dahulu sebelum mengimplementasikannya pada hardware. Hasil penelitian menunjukkan bahwa persentase error untuk model kinematik maju adalah 1,04% dan kinematik terbalik adalah 0,76%, yang berarti kedua model mencapai target bahwa kesalahan model maksimum harus kurang dari 2%.

Kata kunci: robot lengan, forward kinematic, inverse kinematic, MATLAB

1. INTRODUCTION

Robot arm or robot manipulator is a robot that has a mechanical configuration and movement like a human arm. Robotic arms have been used massively in various industrial manufacturing processes, such as for the process of pick and place [1], sorting [2], assembling and packing [3], and welding [4]. Robot arms that are used in industrial processes have various and different types from one process to another which can be categorized based on the configuration and the number of degrees of freedom.

With the differences in these types, it is necessary to have a mathematical model that is appropriate to the needs of the movement of each robot arm.

The first step involved in any robotic system is the analysis and modeling of the kinematics robot arm [5]. There are two ways of kinematics analysis, they are forward kinematics and inverse kinematics. Kinematics analysis on robot arms has been used in many studies using several methods. The methods used in forward kinematics are geometric, homogeneous transformation matrices, quaternions, and Euler-angles, while in inverse kinematics the methods are geometric approach, kinematic decoupling, and algebraic method [6]. Research [7] and [8] used the Denavit-Haternberg method in analyzing forward kinematics on a 6-DoF robot arm. The forward kinematic solution is simpler than the inverse kinematic solution because the inverse kinematic solution is more complex and non-linear. Therefore, development is carried out in finding a solution to the inverse kinematic solution. For example, the analysis of the inverse kinematic solution on a 5-DoF robot arm using an algebraic approach with DH Parameters and the result that the robot arm can move to the desired position [9]. Meanwhile, [10] uses a geometric approach on a 3-DoF planar arm robot that successfully reaches the intended coordinates with an average error of 5.7 mm. After modeling with both kinematics analysis has been carried out, the results can be implemented into the program and simulated using several software such as [11] with simulation using V-REP to analyze a 3-DoF robot arm with forward kinematics that obtains the error percentage is 4.99% for the x-coordinate, 5.57% for the y-coordinate, and 3.18% for the z-coordinate. This is different from what was done in research [12] and [13] which chose kinematics modeling on the PUMA 560 robot arm and AL5A robot arm using MATLAB simulation with the RVC robotics toolbox feature from Peter I. Corke.

This study aims to identify the 3-DoF AX-12A robotic arm and design the modeling of the robotic arm kinematically in two ways, they are forward kinematic using the Denavit-Haternberg method and inverse kinematic using a geometric approach method that can be useful as a learning module for robotics courses and also become the basis for further development of robotic arm research.

2. RESEARCH METHOD

2.1 Forward Kinematic Model

The Denavit-Hartenberg (DH) method is systematic and the application of this method is easy to model the serial manipulators. The DH method is used to develop kinematic models of robots due to its versatility and acceptance for modeling any number of joints and links of the arm regardless of complexity. In this method, we want to calculate the position (from basic coordinates) of the robot arm by assigning an angle value to the DH-Parameter. DH-Parameter is used to model the articulated robot type by describing the parameters of the relationship between one joint and another.

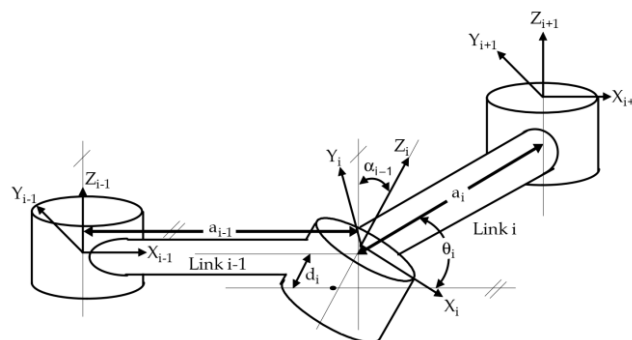


Figure 1. Common frame assignment for a general manipulator [14]

The parameters are a_i , α_i , d_i , and θ_i , while

- a_i : The distance from Z_{i-1} to Z_i is measured along X_i
- α_i : The angle between Z_{i-1} and Z_i is measured along X_i
- d_i : The distance from X_{i-1} to X_i is measured along Z_i
- θ_i : The angle between X_{i-1} and X_i is measured along Z_i

where X_i axis is pointing along the common normal and Z_i axis of the coordinate frame is pointing along the rotary or sliding direction of the joints. Table 1 shows the DH-Parameter of the 3-DoF robotic manipulator.

Table 1. DH-Parameter of 3 links robot arm

Links	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

After the DH-Parameter values have been obtained, then these values are put into a homogeneous transformation matrix. The matrix is based on rotation and translation of the x and z axes. The homogeneous transformation matrix equation based on DH-Parameter on the robot arm can be seen in equation (1).

$${}^{i-1}_i T = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i}$$

$${}^{i-1}_i T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{i-1}_i T = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

And the final result of the calculation will get a matrix like equation (2).

$$\begin{bmatrix} u & v & w & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} ux & vx & wx & px \\ uy & vy & wy & py \\ uz & vz & wz & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2 Inverse Kinematic Model

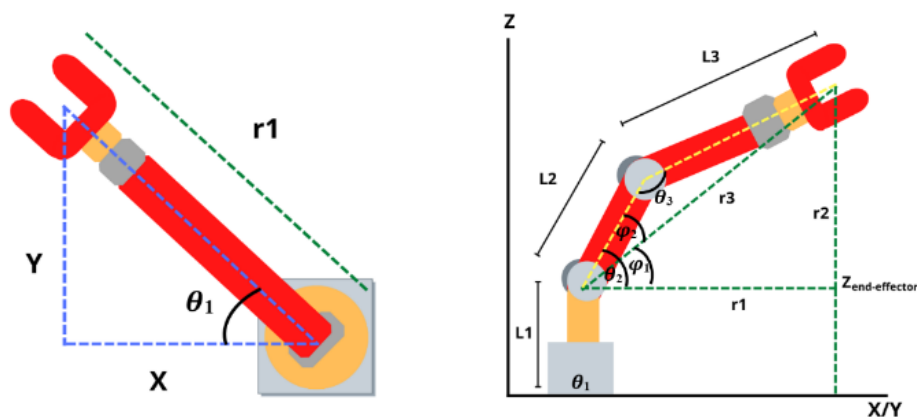


Figure 2. The 3-DoF robot arm inverse kinematic model: top view (left), side view (right)

In solving the inverse kinematic problem can use the geometric approach method. To get a certain end-effector position angle, this method uses the cosine rule of the angle in a triangle (Pythagoras' law and trigonometric rules). This model is adapted from [14] who has made a small-scale 3 DoF arm robot using the inverse kinematic approach for pick and place missions. Therefore, each link on the robotic arm is made into a triangular shape like in Figure 2 and the solution for the inverse kinematic model is based on the following equations.

a) The solution to find the base joint angle (θ_1)

- If X is positive, then :

$$\theta_1 = \tan^{-1} \left[\frac{Y_{ef}}{X_{ef}} \right] \quad (3)$$

- If X is negative, then :

$$\theta_1 = 180^\circ - (\tan^{-1} \left[\frac{Y_{ef}}{X_{ef}} \right]) \quad (4)$$

b) The solution to find the shoulder joint angle (θ_2)

- The solution to find r_1 , r_2 , and r_3 :

$$r_1 = \sqrt{X_{ef}^2 + Y_{ef}^2} \quad (5)$$

$$r_2 = Z_{ef} - a_1 \quad (6)$$

$$r_3 = \sqrt{r_2^2 + r_1^2} \quad (7)$$

- The solution to find φ_1 and φ_2 :

$$\varphi_1 = \tan^{-1} \left[\frac{r_2}{r_1} \right] \quad (8)$$

$$\varphi_2 = \cos^{-1} \left[\frac{(a_2)^2 + (r_3)^2 - (a_3)^2}{2a_2r_3} \right] \quad (9)$$

- So the value of θ_2 :

$$\theta_2 = \varphi_1 + \varphi_2 \quad (10)$$

c) The solution to find elbow joint angle (θ_3)

$$\varphi_3 = \cos^{-1} \left[\frac{(a_2)^2 + (a_3)^2 - (r_3)^2}{2a_2a_3} \right] \quad (11)$$

$$\theta_3 = -(180^\circ - \varphi_3) \quad (12)$$

2.3 The hardware of AX-12A Robotic Arm

The Smart Arm Robotic AX-12A robot has 4 DoF and a gripper (end-effector) as shown in Figure 3. The parts of the robotic arm include links made by aluminum metal rods, 6 servo motors that function as actuators (including 1 on the base rotation, 2 on the shoulder, 2 on the elbow, 1 on the wrist), and an ON-OFF servo motor on the clamp/gripper which functions as an end effector. But in this paper, only 3 DoF was used where the wrist was considered to be inactive with the consideration that modeling forward kinematic and inverse kinematics becomes easier and less complex.

The Smart Arm Robotic AX-12A uses a Dynamixel AX-12A servo motor which is an intelligent servo motor that has an operating angle of 300° , a resolution of $0.29^\circ/\text{step}$ (0-1023 steps), and a torque of up to 12 kg.cm. On this motor two connectors have 3 pins installed, they are GND, VDD, and Data. The connection system is a daisy chain and the communication system is via UART TTL half-duplex multiple-drop interface.

To control the arm robot is assisted by a microcontroller Arduino UNO and PC as an interface. Arduino UNO is a type of microcontroller based on the ATmega 328 board. And for the power supply, an external power supply is chosen, that is the Travo Power Supply Adapter 12V DC 5A.

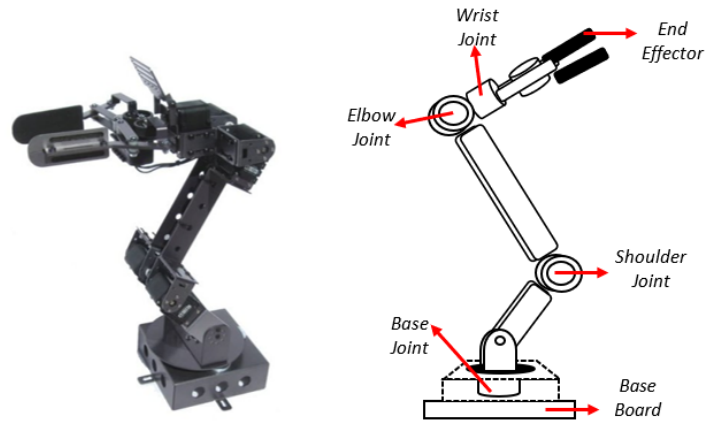


Figure 3. Smart Arm Robotic AX-12A

2.4 Giving Position Codes

The position codes are made to facilitate testing both for simulation and at the time of validation on hardware. The position code is made up of five codes, named from the letters A to E with each code determining the input angle at each joint and also the coordinate position (X, Y, Z) to be reached by the end-effector. The position codes are listed in Table 2.

Table 2. Position codes for testing

Position Code	Angle (degree)			Coordinate (cm)		
	θ_1	θ_2	θ_3	X	Y	Z
A	30	90	90	17	31	16
B	45	100	80	8	33	18
C	90	110	70	-16	28	21
D	120	120	60	-26	14	22
E	180	130	50	-23	-13	24

3. RESULT AND DISCUSSION

3.1 Forward Kinematic Modelling Result

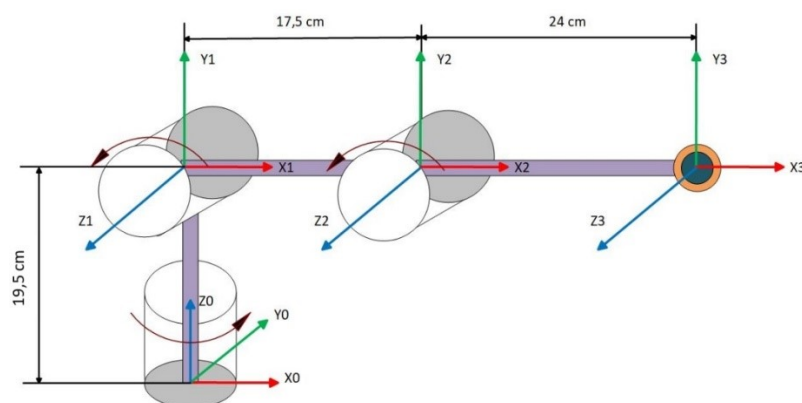


Figure 4. Frame declaration on 3-DoF AX-12A robotic arm

By the frame declaration shown in Figure 4, then the DH-Parameters are defined which include θ_i (joint angle), a_i (link length), d_i (link offset), and α_i (link twist). Then these parameters are described in Table 3.

Tabel 3. The DH-Parameter result of 3 DOF AX-12A robotic arm

Links	a_i	α_i	d_i	θ_i
1	0	90	19.5 cm	θ_1
2	17.5 cm	0	0	θ_2
3	24 cm	0	0	θ_3

The parameter values that have been obtained according to Table 3 are then put into the homogeneous transformation matrix as in equation (1). Each part of the robot arm in the form of a homogeneous transformation matrix is as follows.

$${}^0_1T = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$${}^1_2T = \begin{bmatrix} c\theta_2 & s\theta_2 & 0 & L_2 * c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & L_2 * s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$${}^2_3T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & L_3 * c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & L_3 * s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$${}^0_2T = \begin{bmatrix} c\theta_1 * c\theta_2 & -c\theta_1 * s\theta_2 & s\theta_1 & c\theta_1 * L_2 * c\theta_2 \\ s\theta_1 * c\theta_2 & -s\theta_1 * s\theta_2 & -c\theta_1 & s\theta_1 * L_2 * c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & L_2 * s\theta_2 + d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

After each matrix has been obtained, then to obtain the forward kinematic 3-DoF modeling, the matrix multiplication is ${}^0_3T = {}^0_2T \cdot {}^2_3T$ with the following description.

$${}^0_3T = {}^0_2T \cdot {}^2_3T = \begin{bmatrix} ux & vx & wx & px \\ uy & vy & wy & py \\ uz & vz & wz & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

$$px = (L_3 * c_3 * c_1 * c_2) - (L_3 * s_3 * c_1 * s_2) + (c_1 * L_2 * c_2) \quad (18)$$

$$py = (L_3 * c_3 * s_1 * c_2) - (L_3 * s_3 * s_1 * s_2) + (s_1 * L_2 * c_2) \quad (19)$$

$$pz = (L_3 * c_3 * s_2) + (L_3 * s_3 * c_2) + (L_2 * s_2 + d_i) \quad (20)$$

From the modeling results above, to get the desired end-effector coordinate position values, equation (18) is used for the x-axis, equation (19) for the y-axis, and equation (20) for the z-axis. From these equations, the theta or angle is input for each joint on the robot arm. However, from the modeling made and based on Figure 4, the 3-DoF AX-12A robotic arm has offsets in each theta, they are $\theta_1 = \theta_1 + 30^\circ$, $\theta_2 = \theta_2 - 60^\circ$, and $\theta_3 = \theta_3 - 150^\circ$.

3.2 Inverse Kinematic Modeling Result

From Figure 2, it can be described the value of the arm length on the 3-DoF AX-12A robot arm that has been identified, $L_1 = 19.5$ cm, $L_2 = 17.5$ cm, and $L_3 = 24$ cm. The position values of the coordinates X, Y, and Z are entered based on Table 2. Then these values are put into equation (3) to equation (12) angle/theta (θ_1 , θ_2 , and θ_3) the obtained value is added to the offset value due to the different rotation conditions of the 3-DoF AX-12A robot arm, which is 300° , so the actual theta value is $\theta_1 = \theta_1 - 30^\circ$, $\theta_2 = \theta_2 + 60^\circ$, and $\theta_3 = \theta_3 + 150^\circ$.

3.3 Testing Result of MATLAB Simulation

In this test, the results of the forward kinematic and inverse kinematic modeling were simulated using the MATLAB Graphical User Interface (GUI) and using a toolbox based on the Peter I. Corke model, named Robotic, Vision, and Control (RVC) tools. The data entered is according to the position codes in Table 2. The test results are as follows.

1) Simulation Test of Forward Kinematic Using Denavit-Haterberg Method

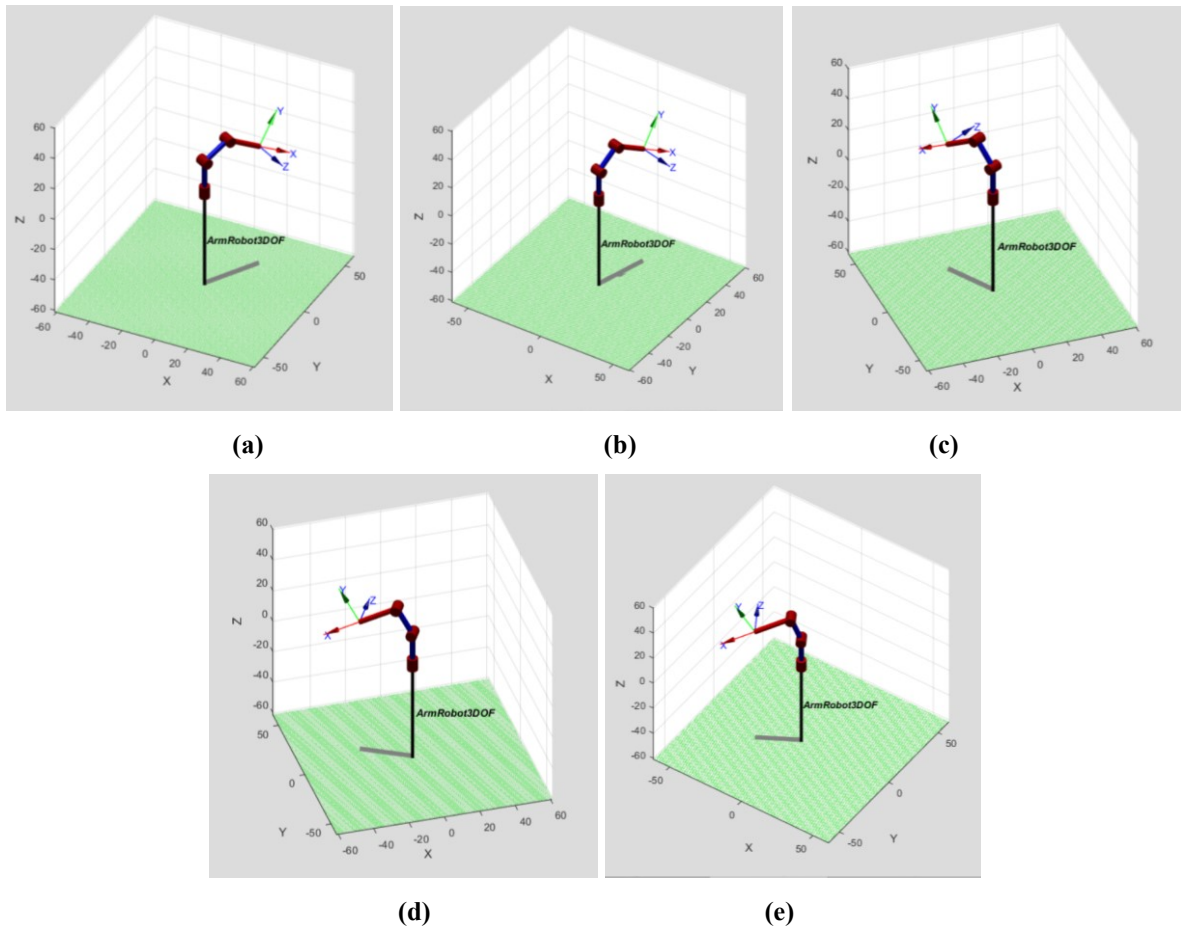


Figure 5. Simulation test of forward kinematic on MATLAB: (a) position A, (b) position B, (c) position C, (d) position D, (e) position E

Based on the simulation test results from Figure 5, data obtained from the position values of the X, Y, and Z coordinates achieved by the end-effector, are listed in Table 4.

Table 4. The simulation test result of forward kinematic

Position code	Input angle (degree)			Output coordinate (cm)		
	θ_1	θ_2	θ_3	X	Y	Z
A	30	90	90	17	31	16
B	45	100	80	8	33	18
C	90	110	70	-17	27	20
D	120	120	60	-26	14	22
E	180	130	50	-24	-14	23

2) Simulation Test of Inverse Kinematic Using Geometric Approach Method

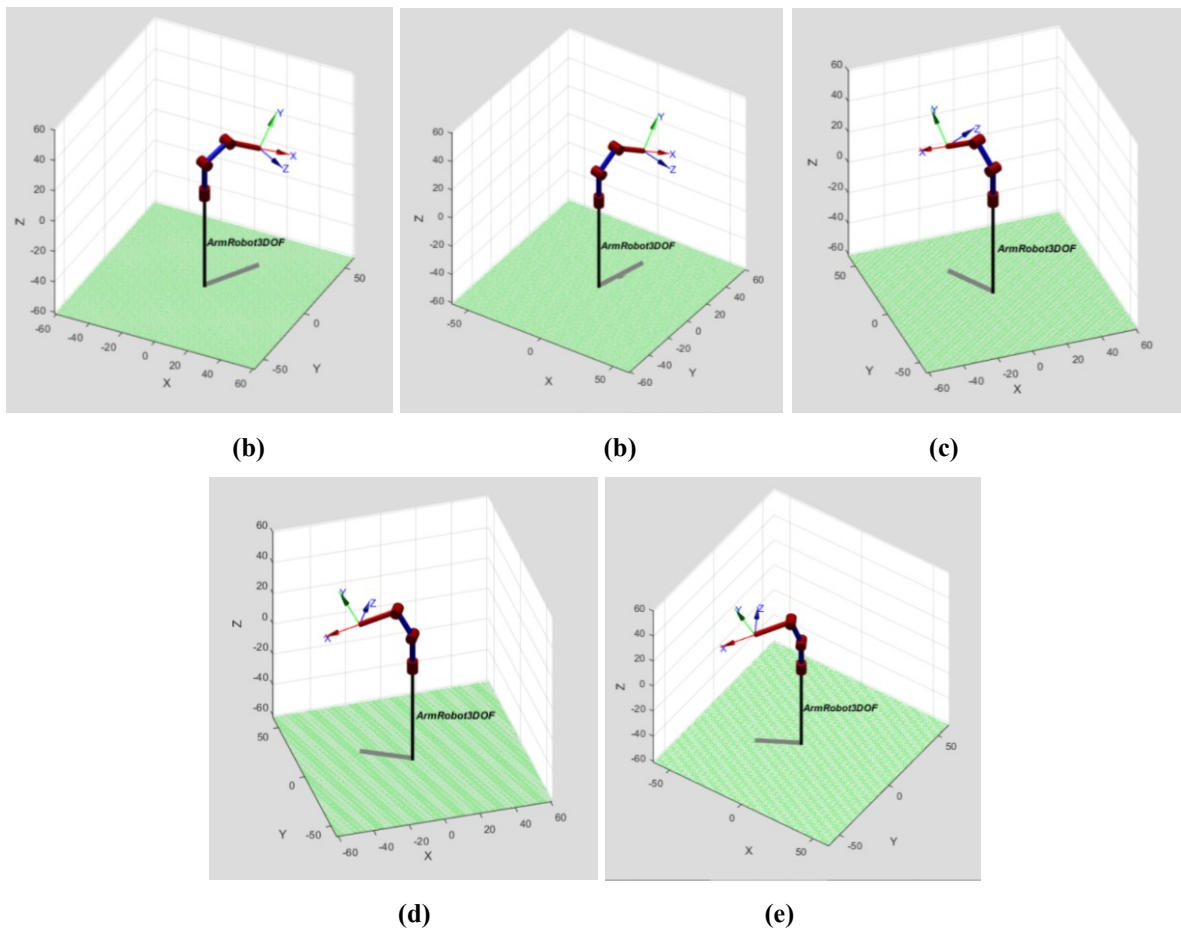


Figure 6. Simulation test of inverse kinematic on MATLAB: (a) position A, (b) position B, (c) position C, (d) position D, (e) position E

Based on the simulation test results from Figure 6, the data obtained from the angle values θ_1 , θ_2 , and θ_3 at each joint, are listed in Table 5.

Table 5. The simulation test result of inverse kinematic

Position Code	Input coordinate (cm)			Output angle (degree)		
	X	Y	Z	θ_1	θ_2	θ_3
A	17	31	16	31.26	91.39	86.89
B	8	33	18	46.37	99.36	78.96
C	-16	28	21	89.74	109.53	70.98
D	-26	14	22	121.69	118.91	59.72
E	-23	-13	24	179.47	131.10	48.74

3.4 Testing Result of Hardware

Testing on this hardware was carried out as a validation that the results of the forward kinematic and inverse kinematic modeling that were made are appropriate to the movement of the 3-DoF AX-12 arm robot. The test is carried out by placing the robot arm on the coordinate board then the robot arm is inputted with the program from the Arduino IDE software with the initial position $\theta_1 = 0^\circ$, $\theta_2 = 150^\circ$ dan $\theta_3 = 150^\circ$. The hardware testing result is shown in Figure 7.

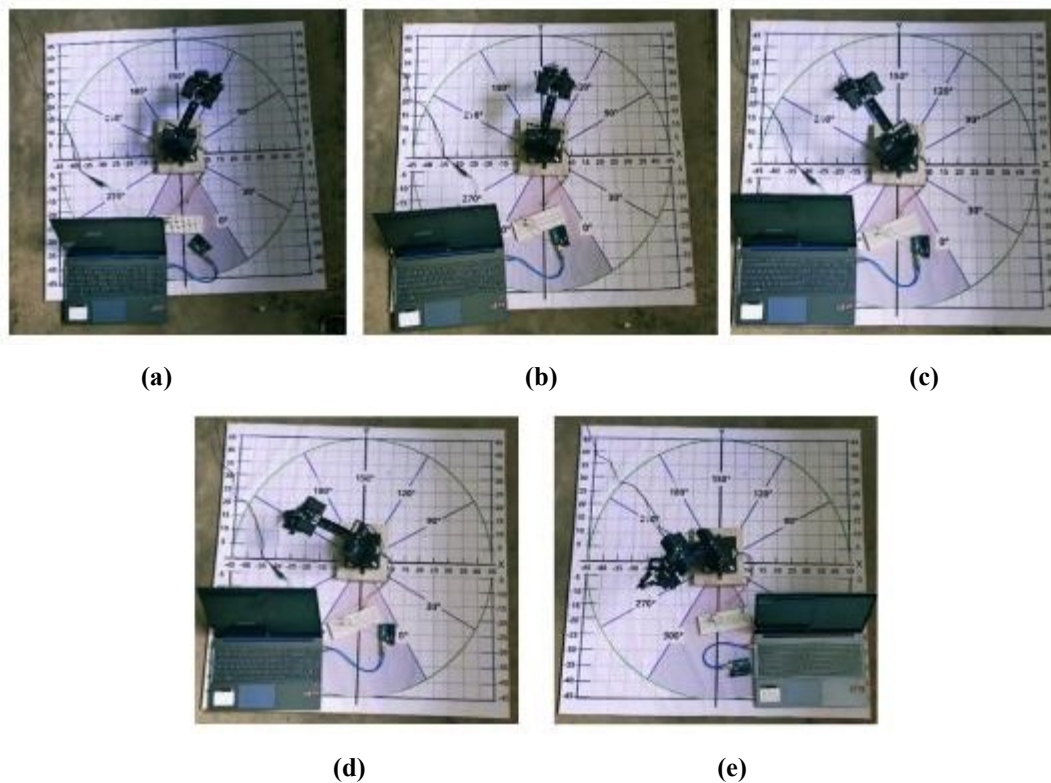


Figure 7. Hardware test: (a) position A, (b) position B, (c) position C, (d) position D, (e) position E

After testing both simulation and hardware, the errors will then be compared and the errors that occur are calculated to find out whether the kinematics modeling made with simulations in MATLAB has the minimum possible error and according to the initial target. As for calculating the error use equation (21).

$$\text{Error (\%)} = \frac{\text{Abs}(\Sigma_{\text{hardware}} - \Sigma_{\text{simulation}})}{\Sigma_{\text{hardware}}} \times 100\% \quad (21)$$

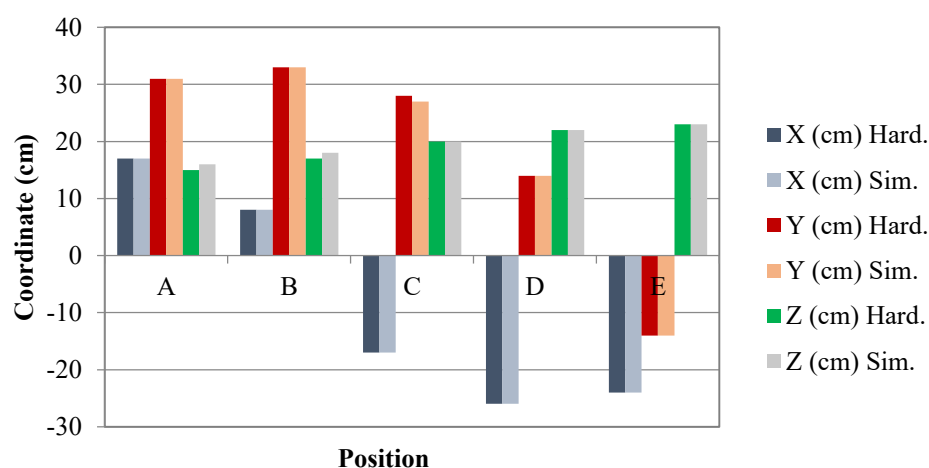


Figure 8. Comparison of hardware and simulation results for forward kinematic

From the comparison results of the forward kinematic simulation model testing with the test results on hardware (Figure 8), the results show that the average error percentage for X coordinate is 0%, for Y coordinate is 1.08%, and for Z coordinate is 2.06%. From the results of the three results of the average percentage of the coordinates, it is found that the forward kinematic model error is 1.04%.

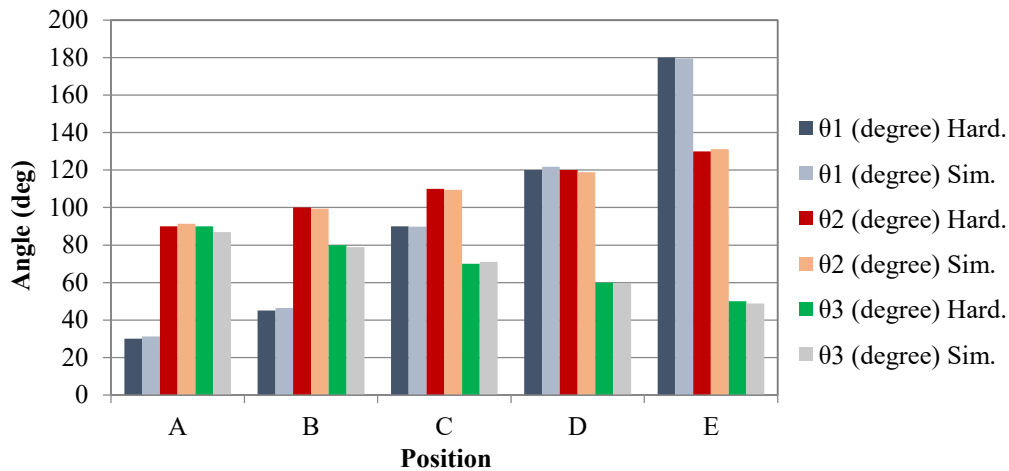


Figure 9. Comparison of hardware and simulation results for inverse kinematic

Meanwhile, from the comparison results of the inverse kinematic simulation model testing with the test results on the hardware (Figure 9), the results show that the average error percentage at angle 1 is 0.76%, at angle 2 is 0.06%, and finally at angle 3 is 0.06%. 1.34%. From the results of the three percentages of the average angle, it is found that the inverse kinematic model error is 0.76%. As for the error in the coordinates and angles during hardware testing, it can be due to the ID2 servo on the hardware that is not functioning properly so when measurements are made, they are less accurate.

4. CONCLUSION

Forward kinematic modeling using the Denavit-Haternberg method and inverse kinematic modeling using a geometrical approach have been successfully developed for the 3-DoF AX-12A robot arm and implemented into the MATLAB program. Meanwhile, from the comparison results of simulation testing with hardware testing, the results obtained in the forward kinematic model are that the average error percentage in the X coordinate is 0%, the Y coordinate is 1.08%, and the Z coordinate is 2.06% so that from the three results of the average percentage of the coordinates, it is found that the forward kinematic model error is 1.04%. Meanwhile, the results obtained in the inverse kinematic model show that the average error percentage at angle 1 is 0.76%, angle 2 is 0.06%, and angle 3 is 1.34%, so from the three results, the average percentage from this angle, the inverse kinematic model error, is 0.76%. Then the two models, forward kinematic and inverse kinematic models, have been made according to the target error percentage, which is a maximum error of 2%. The future research is to generate the robot in the dynamical model using Lagrange-Euler or Newton-Euler method for control application purposes.

REFERENCES

- [1] A. F. Choiri and B. Widiawan, "Penerapan IC 74LS241 Untuk Multi Aktuator Dynamixel AX-12A Pada Biped Robot," *Semnaskit 2015*, 2015, pp. 1-5.
- [2] H. S. Nu'man and Y. Sofyan, "Pengendalian Robot Lengan Pemilah Benda Berdasarkan Bentuk Menggunakan Teknologi," *Seminar Nasional Teknologi dan Riset Terapan 2020*, Sukabumi, 2020, p. 1-7.
- [3] D. Montreano and S. Pradana, "Pengendalian Tangan Robot Perakit Pada Disain Computer Integrated Manufacturing Berbiaya Rendah (CIM) Menggunakan Visual Basic Dan Arduino," *Bina Tek.*, vol. 14, no. 2, pp. 199, 2018.
- [4] Z. Yao, L. Lin, and T. Tang, "Light-Weight Topological Optimization for Upper Arm of an Industrial Welding Robot," *Metals*, vol. 9, no. 9, pp. 1020, 2019.
- [5] R. Oktama, R. Maulana, and G. E. Setyawan, "Implementasi Robot Lengan Pemindah Barang 3 DOF Menggunakan Metode Inverse Kinematics," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 8, pp. 2810-2816, 2018.
- [6] J. L. Aroca Trujillo, R. R. Serrezuela, V. Azhmyakov, and R. S. Zamora, "Kinematic model of the scorbot 4PC manipulator implemented in Matlab's Guide," *Contemp. Eng. Sci.*, vol. 11, no. 4, pp. 183-199, 2018.

-
- [7] X. Zheng, Y. Zheng, Y. Shuai, J. Yang, S. Yang, and Y. Tian, "Kinematics analysis and trajectory planning of 6-DOF robot," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, Mar. 2019, pp. 1749–1754.
 - [8] J. Yu, D. You, and J. Liu, "Analysis of inverse kinematics method for six degrees of freedom manipulator based on MATLAB," in *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)*, Beijing, China, Aug. 2017, pp. 211–215.
 - [9] A. Patil, M. Kulkarni, and A. Aswale, "Analysis of the inverse kinematics for 5 DOF robot arm using D-H parameters," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, Okinawa, Jul. 2017, pp. 688–693.
 - [10] A. Z. Rahman, K. Jauhari, D. Sumantri, T. Hendro, I. D. Nugraha, and S. Amrullah, "Inverse Kinematics dan Pengukuran Akurasi Pergerakan pada Model Robot Manipulator Lengan," *J. Tek. MESIN*, vol. 3, no. 2, pp. 53-58, 2019.
 - [11] A. Uchrowi, L. Lasmadi, and S. Soekarno, "Pemodelan dan Simulasi Robot Lengan 3 DOF Menggunakan V-REP," *AVITEC*, vol. 1, no. 1, pp. 87-98, 2019.
 - [12] A. Surriani, M. Arrofiq, and F. Fahmizal, "Pemodelan Forward Kinematic dan Inverse Kinematic Robot Berlengan PUMA 560," *J. Ilm. Tek. Elektro Komput. Dan Inform.*, vol. 4, no. 2, pp. 66-75, 2019.
 - [13] R. R. Serrezuela, A. F. C. Chavarro, M. A. T. Cardozo, A. L. Toquica, and L. F. O. Martinez, "KINEMATIC MODELLING OF A ROBOTIC ARM MANIPULATOR USING MATLAB," *ARPJN Journal of Engineering and Applied Sciences*, vol. 12, no. 7, pp. 2037-2045, 2017.
 - [14] A. R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale Robot Arm Design with Pick and Place Mission Based on Inverse Kinematics," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 469-475, 2021.

