Jurnal Kejuruteraan 36(3) 2024: 1127–1134
https://doi.org/10.17576/jkukm-2024-36(3)-23

Structural Optimization of 4-DOF Agricultural Robot Arm

Nurul Emylia Natasya Ahmad Zakey, Mohd Hairi Mohd Zaman*, & Mohd Faisal Ibrahim

Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia, Malaysia

**Corresponding author: A174449@siswa.ukm.edu.my*

Received 29 July 2023, Received in revised form 6 February 2023
Accepted 6 March 2023, Available online 30 May 2024

ABSTRACT

The shortage of human labor is increasing; thus, more agricultural machinery and equipment are expected to enter the agricultural sector. One of the agricultural machinery widely studied nowadays involves robot arms. Therefore, developing robot arms is a hot issue in this field. The ideal structure of the robot arm with optimal length is currently gaining popularity and being used in many sectors, such as manufacturing and agriculture. This is closely related to the dynamic structure of agricultural areas. Therefore, this study uses the forward kinematic modeling method to design an optimal robot arm to achieve a specific coordinate in a dynamic environment. The robot in this study arm mimics the boom and arm installed on a tractor. The forward kinematic problem in this study is defined using the Denavit-Hartenberg (DH) convention method. The DH convention is commonly used to solve kinematic analysis problems of a robot arm. Simulation of kinematic modeling is performed using MATLAB software. This study studies various optimization algorithms to compare the performance of algorithms that can achieve the optimal length with minimum errors. The comparison between artificial bee colony (ABC) and particle swarm optimization (PSO) is studied. At the end of the study, the best algorithm was selected for the robot arm design with a four-degree-of-freedom (4-DOF). The best algorithm, i.e., the PSO algorithm, is evaluated by calculating mean square error (MSE of 0.00108527), root mean square error (RMSE of 0.01678), mean absolute error (MAE of 0.004286081), and end-effector position error (error of 0.080557045), where the best algorithm has the lowest value of error.

Keywords: Agricultural; Structural optimization; Robot manipulator; Artificial bee colony, Particle swarm optimization; Forward kinematic

INTRODUCTION

Over the past few decades, the global agricultural industry has been facing an increasing demand for food while grappling with resource constraints and environmental concerns. The main problem is the limited use of agricultural technologies (Roshanianfard et al. 2019). Learning to operate agricultural equipment is sometimes challenging because it requires time and physical effort. According to the Malaysian Palm Oil Board, 69% of agricultural industry laborers were foreign workers, which increased to 76.5% in 2012 (Crowley 2020). However, labor shortage issues cause a negative impact on agricultural industry output. The development of robot arms is one of the big reasons to overcome this problem. The research and production in the 1950s of the robot arm was to increase productivity and improve product quality. They are aimed to replace workers in hazardous professions. Kinematics is the platform to build the robotic manipulator (Nguyen et al. 2021). Therefore, robotic arms have been used rapidly to ensure that the workforce problem can be solved (Suleiman et al. 2018).

Robot arms are commonly used in the manufacturing sector. However, much research is still needed for robotic arms to interact with unstructured and dynamic environments, such as in the agricultural sector. Driving in this environment is more difficult for the robot since it needs to recognize and respond to changes. In other words, this unstructured environment is unpredictable.

Furthermore, an accurate structural design of the robot arm is required to operate in areas with dynamic structures, such as in agricultural areas.

This study aims to develop a methodology using forward kinematics on the dimensional optimization of the four degrees of freedom (DoF) robot arm. 4 DoF robot arm is selected to mimic the boom and arm installed on a tractor used in the agricultural sector, which typically has a 4 DoF structure. The kinematic modeling used is to investigate the optimal length of the robot arm by comparing two algorithms: Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO).

FORWARD KINEMATIC MODELING

Forward kinematics refers to using a robot arm's kinematic equations to determine the end effector's location. A series of links joined together by multiple joints make up the robot arm. The relationship between the position of the end effector and each joint on the robot arm is the focus of the forward kinematic problem.

Corke (2017) has developed a toolbox that uses MATLAB software with a robot arms toolbox that can operate effectively. It has a feature including the functions transformation matrices, reach the trajectories, etc.

The first rule in modeling the robot arm is to assign $\hat{z}_i$ axis tally with joint ii and $z_{i-1}z_{i-1}$ axis and coincide with joint axis $i-1$ (Ibrahim 2018.)The right-hand rule determines the positive rotation's direction. The second rule is finding the line segment orthogonally intersecting both joint axes. z_{i-1} and z_i. Then, extract the Denavit-Hartenberg (DH) parameters by defining the four parameters that exactly specify $T_{i-1,i}$. The four parameters are link length, a_{i-1}, link twist α_{i-1}, link offset $d_i$and joint angle $\emptyset_i$.

OPTIMIZATION METHOD

In 2005, Karaboga put forth ABC, an innovative artificial intelligence algorithm inspired by honey bee behavior. Since its creation, ABC has been applied to a wide range of issues. ABC's recently developed optimization method simulates the honey bees' intelligent foraging behavior (Suresh et al. 2018).

The swarm-based optimization method known as the particle swarm optimization (PSO) algorithm models animal social behavior, including fish, birds, livestock, insects, and other species. The herd forages together, and each animal constantly modifies its search pattern in response to its own and other members' developing experiences (Wang et al. 2018). By selecting these two algorithms, this paper comprehensively compares their performance, convergence speed, and accuracy in finding the optimal arm length.

METHODOLOGY

The method proposed in this paper involves kinematic modeling of 4DoF OpenManipulator-X, as shown in Figure 1. The robot in this study arm mimics the boom and arm installed on a tractor typically used in the agricultural sector, as shown in Figure 2. The OpenManipulator-X is chosen because it is an open-source platform and L_2 and L_4can be modified and customized for the user to change the link length. This robot arm is suitable for this study since the optimal length can be used to change the link length L_2 and L_4. This paper focuses on optimizing those link lengths using MATLAB and modeling them using the Simscape extension from Simulink (Rozaini et al. 2023).

FIGURE 1. Robot arm OpenManipulator-X
Source: Robotis

FIGURE 2. Tractor with boom and arm

FORWARD KINEMATIC MODELING

To start modeling, the first rule is to assign $\hat{z}_i$ axis tally with joint i and z_{i-1} axis and coincide with joint axis $i - 1$ (Ibrahim 2018.). The right-hand rule determines the direction of positive rotation. The second rule is to find the line segment that orthogonally intersects both joint axes, z_{i-1} and z_i. Then, the modeling process extracts the Denavit-Hartenberg (DH) parameters by defining the four parameters that exactly specify $T_{i-1,i}$. The four parameters are link length, a_{i-1}, link twist, α_{i-1}, link offset, d_i and joint angle, $\emptyset_i$.

Figure 3 shows the flowchart that represents the forward kinematic modeling. Forward kinematic modeling was performed by using the Denavit-Hartenberg (DH) convention method. This method is to make the presentation of kinematic robots easier. The performance is carried out by using MATLAB.

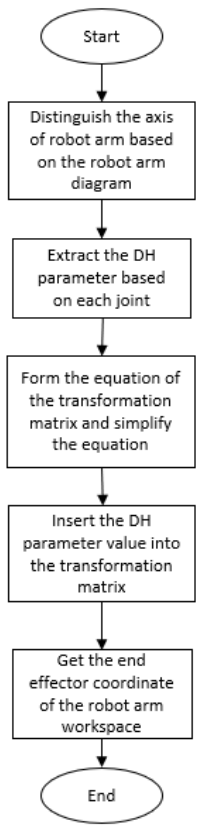


FIGURE 3. Flow chart of the forward kinematic modeling

Firstly, the process distinguishes the axis of the robot arm and analyzes the configuration analysis based on the robot arm diagram of OpenManipulator-X based on its specification, as shown in Figure 4, which is modeled in 2D. Figure 5 shows the kinematic model of the robot arm OpenManipulator-X. Four angles exist in that robot arm.

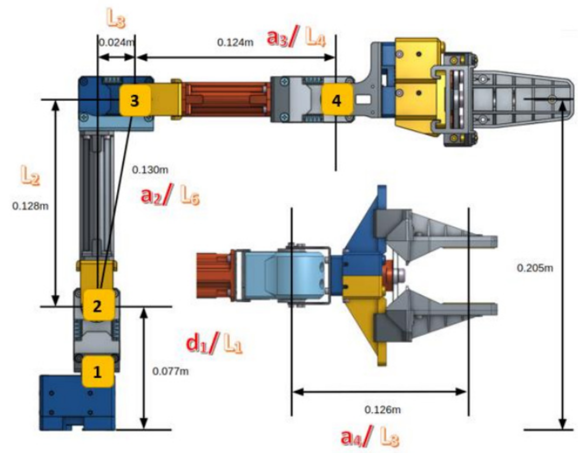


FIGURE 4. Configuration analysis of OpenManipulator-X

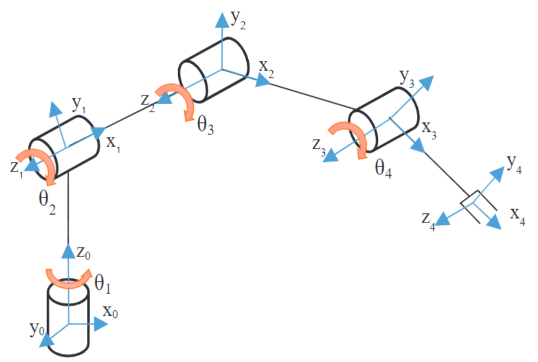
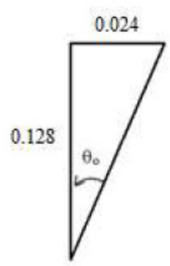


FIGURE 5. Kinematic Diagram of OpenManipulator-X

An offset exists between joint 3 and joint 2 on the robot arm that presents as the offset angle. It is considered in DH parameters. Figure 6 shows the calculation taken to show the offset angle.



$$\theta_o = \tan^{-1}\left(\frac{0.024}{0.128}\right) \approx 11°$$

FIGURE 6. Offset joint 2 and joint 3

After that, the DH parameters from the analysis configuration are extracted, and the result is tabulated in Table 1.

TABLE 1 DH parameter of OpenManipulator-X

Link	θ_i (°)	α_i (°)	$a_i\ (m)$	$d_i\ (m)$
1	θ_1	90	0	0.077
2	$\theta_2 - \theta_0 + 90^\circ$	0	0.130	0
3	$\theta_3 + \theta_0 - 90^\circ$	0	0.124	0
4	θ_4	0	0.126	0

The DH parameters are used for the modeling robot arm. Then, form the equation of the transformation matrix using the DH convention. Equation (1) shows the equation of the transformation matrix. The DH parameter that is being inserted into the *i* in this study is 4 frames, known as 4 DoF, and the matrix is T_4^0. Equation (2) shows the simplified transformation matrix.

$$T_i^{i-1} = \left[cos\ cos\ (\theta_i)\ \ sin\ sin\ (\theta_i)\ \ cos\ cos\ (\alpha_i)\ \ sin\ sin\ (\theta_i)\ \ sin\ sin\ (\alpha_i)\ \ a_i\ cos\ cos\ (\theta_i)\ \ sin\ sin\ (\theta_i)\ \ cos\ cos\ (\theta_i)\ \ cos\ cos\ (\alpha_i)\ \ -cos\ cos\ (\theta_i)\ \ sin\ sin\ (\alpha_i)\ \ a_i\ sin\ sin\ (\theta_i)\ \ 0\ \ sin\ sin\ (\alpha_i)\ \ cos\ cos\ (\alpha_i)\ \ d_i\ \ 0\ 0\ 0\ 1 \right] \tag{1}$$

$$T_4^0 = T_1^0 T_2^1 T_3^2 T_4^3 = \left[n_x\ o_x\ a_x\ p_x\ n_y\ o_y\ a_y\ p_y\ n_z\ o_z\ a_z\ p_z\ 0\ 0\ 0\ 1 \right] \tag{2}$$

The DH parameter that has been obtained is then inserted into equation (2) using MATLAB to get the end effector coordinate.

DIMENSIONAL OPTIMIZATION

Dimensional optimization refers to the process of optimizing a system. This paper tries to identify the optimal set of robot manipulator joint angles (θ_1, θ_2, θ_3, θ_4) and the link parameter L_4 and L_6. L_2 can be found in computed L_6 since length L_3 has a fixed value. L_6 and L_6 dimension is chosen to be optimized because its dimensionality can be described. The lower and upper limit becomes a constraint in the optimization algorithm and is configured as:

$$-180° \leq \theta_1 \leq 180°, -117.465° \leq \theta_2 \leq 90° \tag{3}$$

$$-90° \leq \theta_2 \leq 87.699°, -103.14° \leq \theta_4 \leq 114.6° \tag{4}$$

$$0.115m \leq L_4 \leq 0.135m, 0.120m \leq L_6 \leq 0.140m \tag{5}$$

The objective function is used as mathematical optimization that is described as,

$$obj = \sqrt{(x - x_t)^2 + (y - y_t)^2 + (z - z_t)^2} \tag{6}$$

where (xt , yt , zt) is the position of the target end-effector. To calculate the real *x, y* and *z* coordinates of the end effector, the objective function sends a potential solution to the derived forward kinematic equation for each iteration.

OPTIMIZATION METHOD

Figure 7 shows the optimization process by first selecting the robot arm. The kinematic model was then run using MATLAB, including the ABC and PSO algorithms selected.

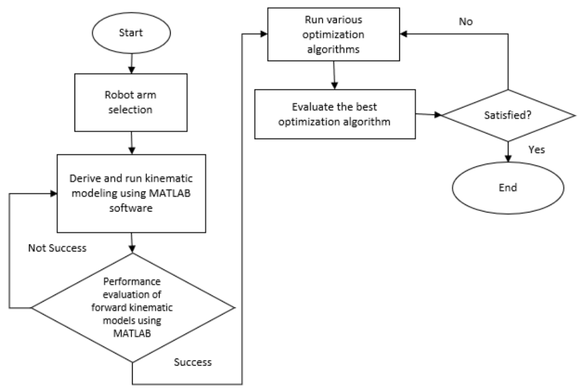


FIGURE 7. General flowchart of the optimization process

The ABC algorithm is a metaheuristic optimization algorithm inspired by the foraging behavior of honeybees (Suresh et al. 2018). It aims to find the optimal solution to a problem by simulating the collaboration between bees in a hive. Figure 8 shows the flowchart of the optimization in ABC. The algorithm creates a population of candidate solutions represented as bees. The bee's position in search spaces correlates to a potential solution. Each bee evaluates the quality of its solution by calculating the objective function. This function represents the fitness that needs to be optimized. During the employed bees phase, the bees have already found a solution and are working to improve it. They accomplish this by performing a local search around their current position to explore the surrounding region in the search space. Next, the bees who had not found the solution yet observed the employed bees' dances and chose a dance (solution)to explore based on the quality of the dances (fitness value). Better performing bees are more likely to be selected as guides for the onlookers. After a certain number of iterations, a scout bee is sent to explore random positions, offering more varied exploration. After settling the three phases, the employed and onlooker bees update their solution based on knowledge gained throughout their exploration. If a better solution is found, it is saved as the current best solution.

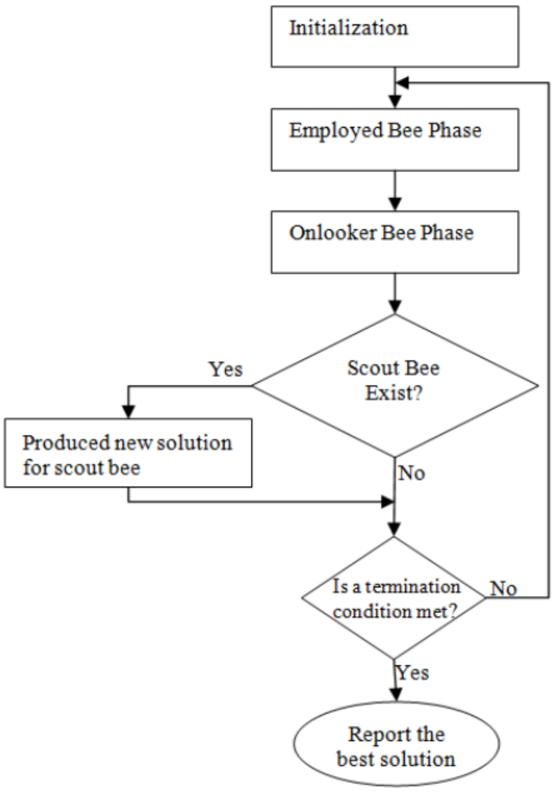


FIGURE 8. Flowchart of ABC optimization process
Source: Kiran & Babalik (2014)

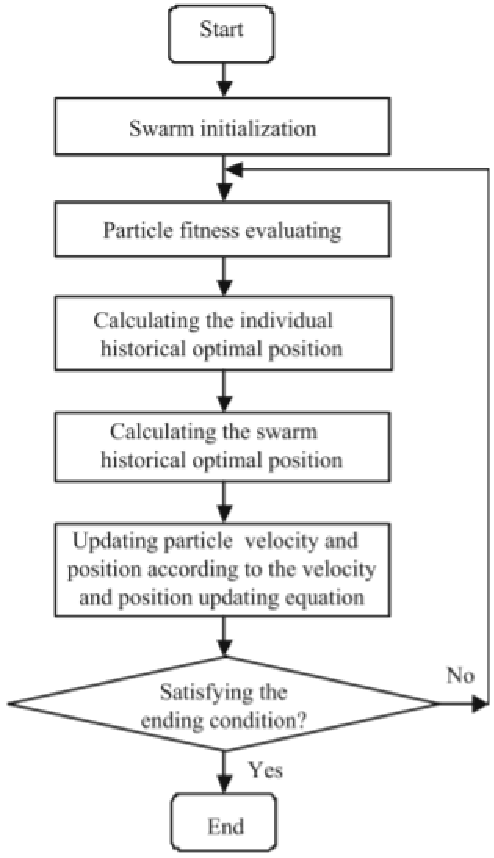


FIGURE 9. Flowchart of PSO optimization process
Source: Wang et al. (2018)

PSO is an algorithm with a population-based optimization technique inspired by the social behavior of birds flocking or fish schooling. (Prayogo et al. 2020). It is commonly used to solve optimization problems by iteratively searching for the optimal solution within a multidimensional search space.

Figure 9 shows the flowchart of process optimization in PSO. Firstly, initialize a population of particles randomly within the search space. Determine each particle's fitness based on its position in the search space. Each particle updates its personal best position based on its current position and fitness value. The personal best position represents the best so far for now. Then, the global best position is updated by choosing the particle with the highest fitness value. This is represented as the best solution found by particles in the population. After that, based on each particle's current position, update each particle's position and speed. The determined new velocity and position guide parts toward the optimal solution.

For both algorithms, the algorithm iterates through 500 iterations. The process stops when this limit is reached, and the best solution found during the process is returned as the final result. After the execution, the OpenManipulator-X links' optimum lengths and optimum set of joint angles were found, and the algorithm's optimal outcome for the dimensional optimization problem was achieved.

The best optimization algorithm is evaluated by using mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and end effector position error, which are the difference between the actual and the target coordinates of the end effector. The errors are described as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |X_i - X| \tag{9}$$

RESULTS AND DISCUSSION

FORWARD KINEMATIC RESULTS

Table 2 shows the result of the coordinate end effector obtained from the DH parameter value OpenManipulator-X. Table 1 shows 10 case results of the forward kinematic with 10 sets of coordinate end effectors. The purpose is for the workspace analysis of the robot arm itself.

Calculating and studying the end effector's coordinates can determine the workspace boundaries of the robot. This allows us to gain insights into the robots' performance, capabilities, and limitations, leading to the development of more efficient in an agricultural environment.

TABLE 2 Result of forward kinematic

Set	Joint angle (°)				Coordinate of end effector		
	θ_1	θ_2	θ_3	θ_4	p_x	p_y	p_z
1	0	0	0	0	0.2748	0	0.2046
2	56	3	-13	79	0.1037	0.1537	0.3018
3	103	15	-5	21	-0.0497	0.2154	0.2931
4	108	92	21	52	0.0923	-0.2839	0.2441
5	-158	56	-79	14	-0.1360	-0.0549	0.1008
6	65	68	-23	-20	-0.0441	0.0946	0.2706
7	52	31	-28	20	0.1203	0.1539	0.2549
8	93	-1	-40	84	-0.0111	0.2125	0.2087
9	118	73	-63	14	-0.0575	0.1081	0.2108
10	166	42	-35	-40	-0.1570	0.0391	0.1349

OPTIMAL LENGTH RESULTS

The optimization of the robot arm structure is carried out for ABC and PSO algorithms. Each of the 10 sets of coordinates obtained from the forward kinematic modeling results underwent an optimization process for both algorithms. Each value for each coordinate's mean, minimum, maximum, and standard deviation is taken. Tables 3 and 4 show the result by using coordinates (0.2748,0,0.2046).

TABLE 3 Optimal length using ABC algorithm using coordinate (0.2748,0,0.2046)

Link	Optimal length			
	Mean	Min	Max	Standard Deviation
L4	0.1239	0.1154	0.1339	0.0063
L6	0.1303	0.1214	0.1391	0.0068

TABLE 4 Optimal length using PSO algorithm using coordinate (0.2748,0,0.2046)

Link	Optimal length			
	Mean	Min	Max	Standard Deviation
L4	0.127604	0.115	0.12	0.009231
L6	0.132778	0.135	0.14	0.009426

Figure 10 and Figure 11 show the visual representations that can help in understanding the distribution of data and insight into the performance and effectiveness of the algorithm in terms of mean optimal length.

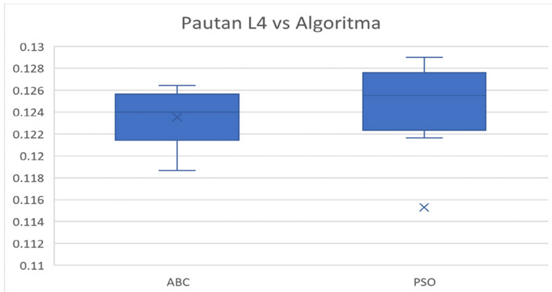


FIGURE 10. Optimal link length 4

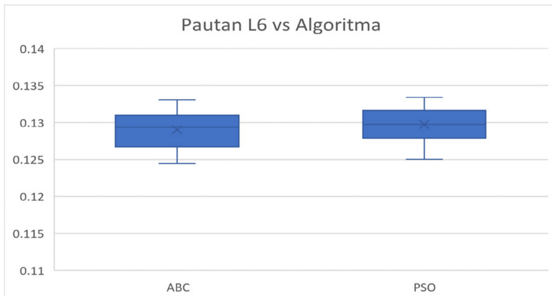


FIGURE 11. Optimal link length 6

The length of the robot arm as a whole is calculated based on the mean value obtained from all 10 target sets of end effector coordinates. Table 5 shows the results for the optimal arm length for the ABC and PSO algorithms for link arm 4 and arm 6, respectively.

TABLE 5 Optimal length after the optimization process

Link	Optimal length	
	ABC	PSO
L4	0.12354	0.12903
L6	0.11459	0.12958

CONVERGENCE

From Figure 12, The ABC algorithm aims to converge towards the optimal solution but takes longer than PSO due to the stochastic nature of the scout bee. For the PSO algorithm, it converges when the particles reach a stable state.

Overall, the PSO convergent graph shows smoother convergence with possible oscillations around the optimum while ABC shows a more stochastic behavior with fluctuations in the objective function value.

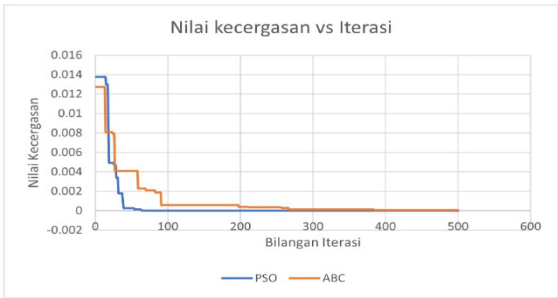


FIGURE 12. Convergence of ABC and PSO

ERROR EVALUATION AND DISCUSSION

Figure 13 shows a comparison of the final position error for the ABC and PSO algorithms. It can be seen roughly that PSO has a lower final position error than ABC. Table 6 shows the calculated errors for ABC and PSO algorithms more precisely.

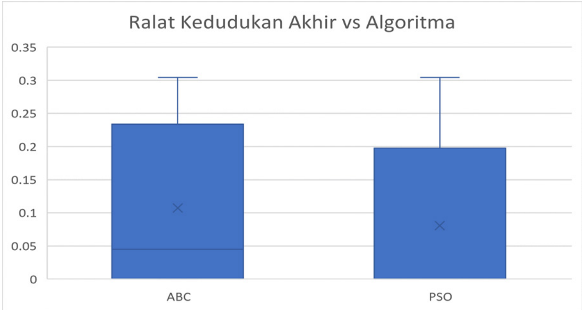


Figure 13. Comparison of error end position

TABLE 6 Error result for ABC and PSO algorithms

Error	Algorithm	
	ABC	PSO
MSE	0.00863	0.00108527
RMSE	0.07755	0.01678
MAE	0.02604	0.004286081
Standard Deviation	3.09989×10^{-5}	1.0361×10^{-3}
Error end position	0.107065654	0.080557045

Based on Table 4.13, the MSE of PSO is lower than ABC, which is 0.00108527 and 0.00863, respectively, proving that the model prediction is closer to the real point and suggests higher accuracy. In other words, a low MSE makes a closer and more accurate prediction. The RMSE for PSO also shows lower values than ABC, suggesting higher accuracy and better fit to the data, which are 0.01678 and 0.007755, respectively. The MAE also shows a lower value for the PSO, indicating a more minor average absolute deviation from the ground truth. For the final position error, the lower value has a higher accuracy dominated by PSO, which is 0.080557045. The results from this evaluation prove that PSO has a lower average error value compared to ABC.

SIMULATION

The simulation is also done to prove the algorithm PSO has the most optimal length. A trajectory that has an optimal arm shows a smoother movement with minimal fluctuations. This proves that the optimization method found a length that produced a more stable and controlled final detector motion.

When the simulation was carried out, it was found that the robot arm that used the optimal length of ABC exhibited a slightly oscillating movement. This is due to the challenges in finding the length to produce a smooth and consistent movement. The smooth movement has been proven by calculating the root mean square error (RMSE) where PSO is 0.01678 while ABC is 0.07755, proving that PSO has a smoother movement because it has a lower error value. Figure 14 and 15 show the robot arm model that uses optimal length by ABC and PSO, respectively.

Figure 14 Model robot arm using optimal length ABC

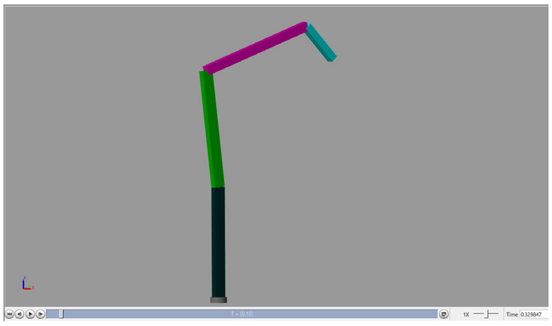


Figure 15 Model robot arm using optimal length PSO

CONCLUSION

In summary, based on the objective of this paper, the first is to publish a forward kinematic model using the Denavit-Hartenberg convention for robot arm modeling. Forward kinematic modeling is performed by using the OpenManipulator-X robot arm by analyzing the configuration of the robot arm and then extracting the DH parameters. The conventional DH method uses the DH parameters to obtain the transformation matrix. The position and target orientation of the end effector coordinates were successfully determined based on the joint angle of the OpenManipulator-X robot arm. The applied forward kinematics successfully allows us to calculate the transformation matrix to represent the position of the end effector in the robot's basic coordinate system. Then, The optimization algorithms compared are ABC and PSO. PSO was chosen as the best method in this study because it has a lower error rate than ABC. Both methods are metaheuristic optimization algorithms used to solve the problem of this study. By choosing PSO as the best method, we can accurately determine the optimal length of the robot arm.

ACKNOWLEDGMENT

This work was financially supported by Universiti Kebangsaan Malaysia (grant no. GUP-2023-016).

DECLARATION OF COMPETING INTEREST

None

REFERENCES

Andhare, P. & Rawat, S. 2017. Pick and place industrial robot controller with computer vision. *Proceedings 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA 2016*.

Corke, P. 2017. *Robotics, Vision, and Control*. Springer International Publishing: Cham. Springer Tracts in Advanced Robotics 118. 2nd Ed. New York: Springer Nature.

Crowley, M.Z. 2020. Foreign labor shortages in the Malaysian palm oil industry: Impacts and recommendations. *Asian Journal of Agriculture and Development* 17(2): 1–19.

Ibrahim Ismael Hamasarah. 2018. Forward Kinematics. *Robotics Systems, Ch. V. Robotics: Rigid Body Motion*, 1-19.

Kiran, M.S & Babalik, A. 2014. Improved artificial bee colony algorithm for continous optimization problems. *Journal of Computer and Communications* 2(4): 108–116.

Nguyen, T.Q., Phan, V.T., Vo, D.T., Trinh, V.H., Nguyen, H.V., Tran, M.S. & Tran, D.T. 2021. Kinematics, dynamics and control design for a 4-DOF robotic manipulator. *Proceedings of 2021 International Conference on System Science and Engineering, ICSSE 2021*, 138–144. Institute of Electrical and Electronics Engineers Inc.

Prayogo, D., Sebastian, Hauwing, W., Wong, F.T. & Tjandra, D. 2020. Structural design optimization using particle swarm optimization and its variants. *IOP Conference Series: Earth and Environmental Science.* Institute of Physics Publishing.

Roshanianfard, A., Noguchi, N. & Kamata, T. 2019. Design and performance of a robotic arm for farm use. *International Journal of Agricultural and Biological Engineering* 12(1): 146–158.

Rozaini, I.A., Ahmad Zakey, N.E.N., Mohd Zaman, M.H., Ibrahim, M.F., Mustaza, S.M. & Mohamed Moubark, A. 2023. Bilateral teleoperation with a shared design of master and slave devices for robotic excavators in agricultural applications. *International Journal of Experimental Research and Review* 35: 119–127.

Suresh & Vaishnavai. 2018. Artificial bee colony algorithm: A survey and recent application. *International Journal of Pure and Applied Mathematics* 120(6).

Suleiman, I., Salam, E. & Tanimu, Y. 2018. Development of a robot arm: A review. In *8th National Engineering Conference.*

Development of a bird pest monitoring and repellent robot system for farm use.

Design and fabrication of a cassava drying machine

Ting, H.Z., Zaman, M.H.M., Ibrahim, M.F. & Moubark, A.M. 2021. Kinematic analysis for trajectory planning of open-source 4-dof robot arm. *International Journal of Advanced Computer Science and Applications* 12(6): 768–775.

Wang, D., Tan, D. & Liu, L. 2018. Particle swarm optimization algorithm: An overview. *Soft Computing* 22(2): 387–408.

Zhao, Y., Gong, L., Huang, Y. & Liu, C. 2016. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture* 127: 311–323.

Zheng, Z., Ma, Y., Zheng, H., Gu, Y. & Lin, M. 2020. Industrial part localization and grasping using a robotic arm guided by 2D monocular vision. *Industrial Robot: An International Journal* 45(6): 794–804.