

Improved Trajectory Planning of Mobile Robot Based on Pelican Optimization Algorithm



Rand Zuhair Khaleel¹, Hind Zuhair Khaleel², Ahmed Abdulhussein Abdullah Al-Hareeri³,
Abdulkareem Sh. Mahdi Al-Obaidi⁴, Amjad J. Humaidi^{2*}

¹ Central Library, University of Technology, Baghdad 10066, Iraq

² Control and Systems Engineering Department, University of Technology, Baghdad 10066, Iraq

³ EL-Nukhba Engineering Company, Baghdad 10066, Iraq

⁴ School of Engineering, Taylor's University, Taylor's Lakeside Campus, Subang Jaya 47500, Malaysia

Corresponding Author Email: amjad.j.humaidi@uotechnology.edu.iq

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.570408>

ABSTRACT

Received: 19 April 2024

Revised: 24 July 2024

Accepted: 9 August 2024

Available online: 27 August 2024

Keywords:

mobile robot, POA, PSO, trajectory, obstacles

The problem of trajectory for mobile robots included motion mobile robot from beginning point to end-point without collision with the obstacles. This article proposed two optimization methods, represented by Pelican Optimization Algorithm (POA) and Particle Optimization Algorithm (PSO) to have optimal trajectory for the mobile wheeled robot without collision in presence of obstacles. A minimization of the Mean Square Error (MSE) of positions (x,y) and orientation (θ) as fitness function for the proposed techniques. The mathematical representation of a mobile wheeled robot has been developed and experimental results have been conducted to verify the numerical results, which have been simulated using MATLAB software environment. The results showed that both POA and PSO could successfully avoid static obstacles. In addition, the results showed that less trajectory error can be obtained with POA as compared to PSO. The experimental validation is done using a BOE-Bot wheeled mobile robot. This robot attached with WiFi camera used to detect the obstacle coordinates using suggested POA algorithm.

1. INTRODUCTION

The mobile robot movement defined as a motion of a mobile wheeled robot beginning from the beginning point to the endpoint. The trajectory becomes more important as mobile robot application scenarios become more complicated and unpredictable [1]. One of the most common problems in mobile robots that have been addressed by several researchers over the past 20 years is obstacle avoidance. Trajectory with obstacle avoidance aimed to determine the best path for a mobile robot between two points [2]. For mobile robots, autonomous motion planning is a useful feature. It has been needed human intervention on their part. But it also involved a lot of work and issues to be resolved, such as Artificial Intelligence (AI) methods. The task at hand involved determining the optimal path to take a wheeled robot from its existing location to final one. For instance, both states could be the starting point and the objective, respectively [3]. Ibrahim et al. [4] employed a fuzzy logic intelligent technique to regulate and control mobile robot mobility in stationary obstacles. The software utilized for the robot's creation and simulation was Webot Pro and MATLAB. The mobile robot's objective was to move while avoiding immovable objects. Lyu et al. [5] enhanced the beetle swarm optimization (IBSO) technique which effectively resolved various trajectory issues in static environments. The IBSO method has demonstrated the mobile robot's ability to move in a stationary obstacle. Shao [6] investigated the Genetic Algorithm called (GA)

which depended on the method of tracking a moving robot in a stationary setting. The researcher simulated the environment, and the robot trajectory using MATLAB.

Raheem et al. [7] presented the path of the robot using the Adaptive Neuro-Fuzzy Inference System which was summarized as (ANFIS) algorithm. It depended on training the positions of the robot to get better locations. Particle's swarm method (PSO) was dependent on the swarm's behavior such as bird flocks. For highly complicated and nonlinear models, the PSO helped resolve difficult issues [8-17]. Alam et al. [18] suggested a trajectory approach based on the PSO method to determine a good path which was avoided collision in static obstacles for a mobile robot. The suggested method simulated and randomly selected grid lines that were created between the beginning and end places of a robot. The Pelican Optimization Algorithm (POA) is among the most advanced optimization algorithms available today. In 2022, the POA was proposed. Based on how pelicans approach their chases and the tactics they were used. It was cunning adaption that allowed the birds to become expert searchers [19-21]. A clearer link between the above background information and the specific problem of this paper is being addressed in all researchers worked on mobile robot with AI algorithms but in this work, the mobile robot linked to WiFi camera based AI modern algorithm (POA) to detect the obstacles. It can be summarized the main contributions as:

- The BOE-Bot wheeled mobile robot with a WiFi camera can avoid two static obstacles.

- The accurate trajectory from beginning point to end-point is implemented by the proposed POA, and it avoids two static obstacles.
- A comparison between two proposed algorithms of PSO and POA implemented.

Comparative study between two proposed methods PSO and POA in terms of MSE measured. This article is structured into eight parts. The Section 2 offered the wheeled mobile robot mathematical model. The Mean Square Error (MSE) equation is illustrated in Section 3. The Section 4 explained the optimization trajectory of a mobile robot using PSO and POA. The proposed Flowchart existed in Section 5. Experimental work hardware connection is illustrated in Section six. The results are illustrated in Section seven. Lastly, conclusion of this work is represented in Section 8.

2. WHEELED MOBILE ROBOT MATHEMATICAL MODEL

The two independently powered wheels installed on side for a mobile body's robot help it to move. The center of rotation could be located anywhere along the line formed by the two tire contact points. A differential drive is called a mobile robot. It has 2 driving wheels that move in the forward direction or backward direction individualistically. They were positioned on the same axis. Even though the velocity was adjusted for every wheel, the mobile robot rotates around a location that deceit along its left and right robot wheels which is a common axis for moving in a rolling status motion. Figure 1 shows an Instantaneous Center of Curvature (ICC), which is the point around which it spins [22, 23].

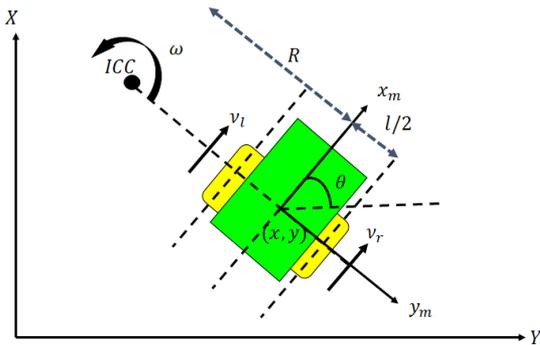


Figure 1. The structure of the differential mobile robot

Kinematics was a mathematical study of location and velocity motion without taking into consideration forces. In Figure 1, the global (world) coordinates $\{X, Y\}$ and the local coordinates $\{x_m, y_m\}$ were utilized to illustrate the differential motion of the mobile robot. A centroid of the moving robot was located at (x, y) . The path that the robot follows causes a shift in the two-wheel velocities. A mobile robot has two speeds for each wheel. Two speeds are combined to form one known as linear velocity: the wheel's left linear velocity (v_l) and the wheel's right linear velocity (v_r). The second velocity (ω) is an angular velocity, which must rotate along the (ICC). ICC was a point which represented the center for circular trajectory which a mobile robot was following at any given moment. In differential mobile robot, this point helpful for describing the rotation and translation of mobile robot [22, 23]. Linear velocities of right and left sides v_r, v_l Eq. (1) and

Eq. (2) for wheeled mobile robot are [22, 23]:

$$\omega \left(R + \frac{l}{2} \right) = v_r \quad (1)$$

$$\omega \left(R - \frac{l}{2} \right) = v_l \quad (2)$$

where, the distance between the robot and wheel centers is marked as (l) . Furthermore, The R presented as the distance from the robot's center (x, y) to ICC. The Eq. (3) and Eq. (4) illustrated as below [22, 23]:

$$R = \frac{l}{2} \left(\frac{v_l + v_r}{v_l - v_r} \right) \quad (3)$$

$$\omega = \frac{v_l + v_r}{l} \quad (4)$$

$$ICC = (ICC_x, ICC_y) = (x - R \sin(\theta), y + R \cos(\theta)) \quad (5)$$

ICC_x and ICC_y are Instantaneous Center of Curvature around two axes: x and y respectively. The mobile robot rotated along ICC with angular velocity ω at the δt time. θ' is the new orientation for mobile robots presented in Eq. (6) [22, 23]:

$$\theta' = \omega \delta t + \theta \quad (6)$$

where, θ' is the mobile robot's new orientation, $\omega \delta t$ is the change of θ called $d\theta$ or $\omega \delta t$ is angular velocity at time δt . A wheeled mobile robot's former orientation is denoted by θ . The general mobile robot's location relative to the time (t) derived as in Eq. (7), Eq. (8) and Eq. (9) [22, 23]:

$$x(t) = \int_0^t V(t) * \cos(\theta(t)) dt \quad (7)$$

$$y(t) = \int_0^t V(t) * \sin(\theta(t)) dt \quad (8)$$

$$\theta(t) = \int_0^t \omega(t) dt \quad (9)$$

A final formula of computer simulation derived as in Eq. (10), Eq. (11), and Eq. (12) of the differential mobile robot [22-24]:

$$x(kt) = x((k-1)t) + \frac{((v_r(kt) + v_l(kt)) \cos(\theta(kt)))}{2} t \quad (10)$$

$$y(kt) = y((k-1)t) + \frac{((v_r(kt) + v_l(kt)) \sin(\theta(kt)))}{2} t \quad (11)$$

$$\theta(kt) = \theta((k-1)t) + \frac{((v_r(kt) - v_l(kt)))}{l} t \quad (12)$$

where, $x((k-1)t)$, $y((k-1)t)$, and $\theta((k-1)t)$ are the position in x-axis, y-axis, and orientation respectively of the reference point. $x(kt)$, $y(kt)$, and $\theta(kt)$ are centroid for a position mobile robot in x-axis, y-axis, and orientation respectively. The left linear velocity of the wheel is marked as

($v_l(kt)$) and the right wheel's linear velocity is presented as ($v_r(kt)$). The dimension of the mobile robot is illustrated in Table 1 [25].

Table 1. The dimension for wheeled mobile robot

Parameter	Dimension (m)
Length	0.125
Width	0.085
Height	0.2
Width of wheel	0.005
Length of wheel	0.00625

The mean square error is named (MSE). It was defined as the variance between centroid mobile robot point ($x(kt), y(kt), \theta(kt)$) and reference mobile robot point ($x((k-1)t), y((k-1)t), \theta((k-1)t)$) and predicted values as presented in Eq. (13) [26, 27]:

$$MSE = \frac{1}{n} \left(\sum_{k=1}^n (x(kt) - x((k-1)t))^2 + (y(kt) - y((k-1)t))^2 + (\theta(kt) - \theta((k-1)t))^2 \right) \quad (13)$$

where, n is the maximum sample number, the k is a current sample, t is the simulation time.

3. OPTIMIZATION TRAJECTORY OF MOBILE ROBOT USING PSO AND POA

In this work, the two optimization algorithms are used for optimizing the mobile robot trajectory.

3.1 Particle swarm approach

PSO represented by way of a kind of intelligence swarm method that has benefits for simple program, less parameters. Every iteration found the optimal particle position, which is noted and given the name $pbest$. The optimal place identified by all particles is known as $gbest$ refers to the best place that all particles have discovered. The new position and velocity of each particle have been calculated using Eq. (14) and Eq. (15) taking into account $pbest$, $gbest$ [28]. The position of mobile robot trajectory (x, y) is described in Eq. (10) and orientation θ in Eq. (11). Within the optimization process, there are 2 measuring positions ($x_{i,j}^k$) and velocity ($v_{i,j}^k$). They were developed for the new variables $x_{i,j}^{k+1}$ and $v_{i,j}^{k+1}$ depending on the two formulas [28]:

$$v_{i,j}^{k+1} = v_{i,j}^k + c_1 r_1 (pbest_{i,j}^{k+1} - x_{i,j}^k) + c_2 r_2 (gbest_j^{k+1} - x_{i,j}^k) \quad (14)$$

$$x_{i,j}^{k+1} = x_{i,j}^k + v_{i,j}^{k+1} \quad (15)$$

There are two components of joints (i and j) which are represented as i^{th} and j^{th} . The r_1, r_2 were two random parameters varied from 0 to 1. The two symbols c_1, c_2 were illustrated as the cognition and social coefficients which they were varied from 0 to 2, correspondingly. k was an iteration number. The next stage in this algorithm depends on the particle individual which was the best ($pbest_{i,j}$) and also

global particle which was the best ($gbest_j$). This technique enhanced the performance by minimizing the problem of the MSE objective function as in Eq. (14). In this work, the PSO algorithm enhanced the performance of mobile robot trajectory by suggesting the following cost function:

$$Cost\ function = \min (MSE) \quad (16)$$

In this work, the data of (x, y, θ) are generated from Eq. (10), Eq. (11), Eq. (12) and from a case study of linear velocities where $v_r > v_l$. A block diagram for an enhanced PSO algorithm was shown as in Figure 2. It contains 3 blocks. The 1st block is the inputs (x, y, θ) which are generated from a mobile robot model represented mathematically as mentioned in section 2. In addition, the right and left linear velocities (v_r, v_l) are initialized. PSO algorithm has received these 3 inputs for optimization as in the 2nd block. The optimal mobile robot's trajectory output is based on optimal values of (x, y, θ) calculated as in 3rd one. Due to an Eq. (16), PSO worked in order to diminish a cost formula depending on MSE of a mobile robot. The PSO parameters are illustrated in Table 2.

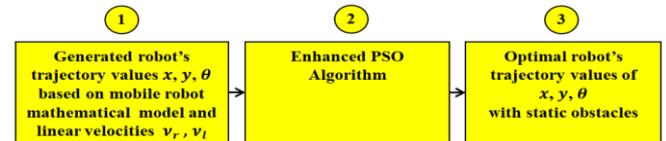


Figure 2. The block diagram for enhanced PSO algorithm

Table 2. PSO parameters

Parameter Description	Value
Number of iteration	25
Number of PSO population, n	100
The cognitive parameter c_1 and social parameter c_2	1.9
Constant factors (r_1, r_2)	0.95
Number of variables	3 (x, y, θ)
Lower & upper variables	$0 < x < -1.1653$ (m)
	$0 < y < 0.5482$ (m)
	$0 < \theta < 90$ (degree)
The linear right velocity's mobile robot (v_r)	7 m/s
The linear left velocity's mobile robot (v_l)	5 m/s
Angular velocity of mobile robot (ω)	40 rad./s

3.2 Pelican optimization algorithm (POA)

Pelican Pelicans introduced the population, which serves as the basis for the POA. Any community of pelicans can be a probable solution in populace depends on algorithms. Due to their positions in searching operation of a planetary, every member's population supplied the values for the variables in the optimization problem. Eq. (17) is used to obtain the lower and upper boundaries for a task, and the population members are started at random. Before discussing the POA procedures, the population is set up. Since pelicans typically seek within a specific radius for prey, the starting locations of each member of the population may be indicative of the group as a whole [19-21].

$$x_{i,j} = lb_{i,j} + random(ub_j - lb_j) \quad (17)$$

where, i was ranged $(1...n)$, j was ranged $(1...m)$, $x_{i,j}$ represented as the first site's pelican. A symbol n presented as number of populace, the symbol m is denoted variables number, $random$ has been varied from 0 & 1. The ub and lb are the minor and higher boundary of m respectively [19-21]. The pelican population matrix X can be expressed as Eq. (2) and $X_i = i^{th}$ pelican [19-21].

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,j} & \cdots & x_{n,m} \end{bmatrix} \quad (18)$$

The fitness function matrix is as below [20]:

$$f = \begin{bmatrix} f_1 \\ \vdots \\ f_i \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ \vdots \\ f_1(x_i) \\ \vdots \\ f_1(x_n) \end{bmatrix} \quad (19)$$

The inspiration behind the POA algorithm and how it related to the pelican's hunting behavior can be explained briefly. The POA has been inspired by the collaborative, effective, and adjustable hunting behaviors for pelicans. These natural strategies were mimicked and the POA accomplished the robust performance for solving compound optimization problems. Agents cooperated and adjusted the two phases: exploration and exploitation.

I. Movement to rapacious phase

Pelicans find their quarry and fly to its direction in the chief step. It has enhanced the POA's ability to precisely assess a problem-solving domain. The pelican's method of achieving its goal and the ideas that were previously covered are mathematically reproduced in Eq. (20).

$$x_{i,j}^{p_1} = \begin{cases} x_{i,j} + random(p_j - l.x_{i,j}), & \text{if } f_1 < f_i \\ x_{i,j} + random(x_{i,j} - p_j), & \text{else} \end{cases} \quad (20)$$

where, $x_{i,j}^{p_1}$ = modernized pelican's position of this phase. p_j = quarry spot, l is randomly adjusted between one or two. It was changed in every repetition. When, it was reached to 2; the supporter has more transposition; it should be yield them into next regions for the interplanetary. Thus, a constraint l determines the extent to which a POA can investigate the space search. A pelican may relocate as long as it enhances its fitness function. It is referred to as operative apprising. This process is simulated using Eq. (22).

$$X_i = \begin{cases} X_i^{p_1}, & \text{if } f_1 < f_i \\ X_i, & \text{else} \end{cases} \quad (21)$$

where, $X_i^{p_1}$ is the updated position of i^{th} pelican in the exploration phase and $f_i^{p_1}$ is the fitness function value of an exploration phase.

II. Exploitation phase

In their second phase, pelicans forced fish upward by expanding their wings and ensnaring them in their neck pouches, which allowed them to rise to the water's surface. Using this strategy, pelicans can capture many fish in the attacked area. The pelicans converged to gain better sites within the hunting zone. This procedure improved the POA's local search and exploitation capabilities. To get the algorithm to converge to an optimal solution, the areas surrounding the pelican site must be mathematically explored. The pelican's hunting habit is mathematically modeled by Eq. (22).

$$x_{i,j}^{p_2} = x_{i,j} + R \cdot \left(1 - \frac{t}{T}\right) (2 \cdot random - 1) x_{i,j} \quad (22)$$

where, $x_{i,j}^{p_2}$ is an updated position pelican of the exploitation phase and $f_i^{p_1}$ presented as the cost function of the exploitation phase. A factor $R = 0.2$. The coefficient $R \cdot \left(1 - \frac{t}{T}\right)$ is a pelican of populace for neighborhood's radius $x_{i,j}$; t which is represented as a counter of iteration, The symbol T expressed by maximum iteration. To find the optimal overall solution, the POA's exploitation power technique is aided by the $R \cdot (1-t/T)$ coefficient. In the beginning iterations, it has been given the high value of this coefficient, a greater area around each member is considered. The approach diminishes with more replications, bringing each neighborhood member's radii down. toward this reason, POA may adapt toward almost globally optimum—that is, nearly absolutely global optimal—solutions, contingent upon the utilization strategy. As a result, it is feasible to scan the region around every person in the population in fewer, more precise steps. The pelican position is now updated and takes on the following form:

$$X_i = \begin{cases} X_i^{p_2}, & \text{if } f_i^{p_2} < f_i \\ X_i, & \text{else} \end{cases} \quad (23)$$

where, $X_i^{p_2}$ denoted as an updated position pelican of the exploitation phase and $f_i^{p_2}$ represented as an objective function of the exploitation phase [19-21]. The POA algorithm improved a robot's trajectory performance in our research, by putting forth the following cost function minimization issue, which is described as:

$$Cost\ function = \min (MSE) \quad (24)$$

Figure 3 shows the enhanced POA algorithm block diagram. It contains of three blocks. The 1st block is the inputs (x, y, θ) which are generated from a mathematical expression of a wheeled robot as mentioned in section 2. In addition, the right and left linear velocities (v_r, v_l) are initialized. POA has received these 3 inputs for optimization as in 2nd block. The optimal mobile robot's trajectory output is based on optimal values of (x, y, θ) calculated as in 3rd block. Due to Eq. (16), the POA method worked to minimize cost function depending on MSE of the mobile robot. The POA parameters seen in Table 3.

For the case study of linear velocities if $v_r > v_l$ then the wheeled robot moved from beginning to endpoint within static obstacles. In this paper, the k equals 2 while the final time of trajectory equals 4 seconds. In real time response of mobile robot Obstacle Avoidance using WiFi camera, the two algorithms POA and PSO are getting a best solution for a given

problem. A modern algorithm called POA is capable to detect obstacle and avoid it faster than PSO. POA is also faster convergence as compared to PSO.

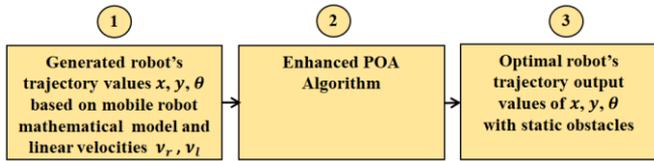


Figure 3. The block diagram for enhanced POA algorithm

Table 3. POA parameters

Parameter Description	Value
Number of iteration	25
n (number of POA population)	100
Values of constant (r_1, r_2)	0.95
Number of Variables	3 (x, y, θ)
limits of variable (m)	$0 < x < -1.1653$ (m)
	$0 < y < 0.5482$ (m)
	$0 < \theta < 90$ (degree)
The linear right velocity's mobile robot (v_r)	7 m/s
The linear left velocity's mobile robot (v_l)	5 m/s
Angular robot's velocity (ω)	40 rad/s

4. THE PROPOSED FLOWCHART

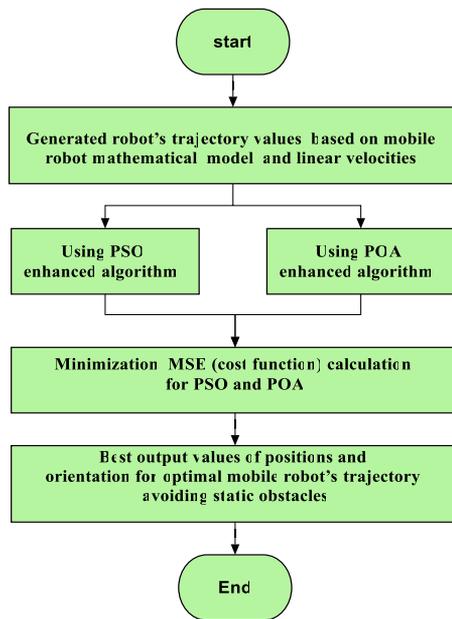


Figure 4. The flowchart for proposed work

The suggested flowchart of this work is seen in Figure 4. It starts with the trajectory the creation positions and directions (x, y, θ) for the wheeled mobile robot arm depending on a mathematical mobile robot's model. The points and orientations selected to be in the robot's trajectory range. In addition, A linear velocities have been initialized where $v_r > v_l$. These two values of velocities have been chosen to suitable the robot's trajectory to avoid static obstacles. Then the suggested algorithms: PSO and POA are implemented. Additionally, the values of output from every two algorithms are the best

positions and orientations (x, y, θ) individually. The PSO and POA depending on generated inputs (x, y, θ) from Eqs. (10)-(12). The position point and orientation (x, y, θ) ranged as seen in the Table 2 and the Table 3. The optimization of suggested algorithms is based on two cost functions (minimization of MSE) as in Eq. (16) for PSO and as in Eq. (24) for POA. The suggested trajectory has been simulated based on these best outputs (positions and orientations) using enhanced PSO and POA.

5. EXPERIMENTAL WORK HARDWARE CONNECTIN

The experimental BOE-Bot wheeled mobile robot platform has been manufactured by (Parallax) as shown in Fig. 5. BOE-Bot wheeled mobile consists of the following hardware components:

- Black rotation crystal
- Wireless WiFi camera monitoring webcam type MD81 mini Wi-Fi P2P sends the Wi-Fi signal to the iPhone and all specifications of this camera are explained in the study [29]. Figure 5 shows the connection between the WiFi camera and iPhone as follows:
 - (A) After fully charging the WiFi camera, Open it on.
 - (B) Install the p2pCamVewier application on an iPhone and then open it.
 - (C) From Wifi iPhone, search for the name of the Wifi camera then choose it.
 - (D) The image has been shown in the figure below:

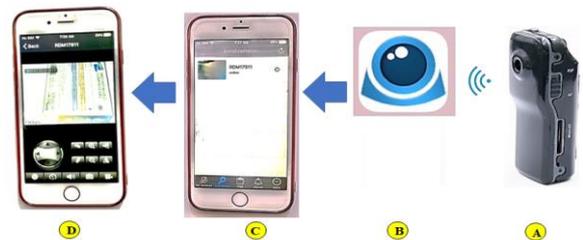


Figure 5. Connection steps between WiFi camera and WiFi iPhone

Figure 6 shows the wheeled mobile robot parts are explained as below:

- 1) The mobile robot chassis is made from aluminum material.
- 2) Wifi-camera.
- 3) Aluminum structure of BOE-Bot wheeled mobile robot.
- 4) Two motor wires, each wire connected between servo motor and mobile robot.
- 5) Switch on/off button.
- 6) BASIC Stamp microcontroller used for programming the code.
- 7) Left robot wheel.
- 8) Right robot wheel.
- 9) Left motor type parallax continuous servo rotating with 360 degrees.
- 10) Right motor type parallax continuous servo rotating with 360 degrees.
- 11) Battery holder attached with 6 volts.

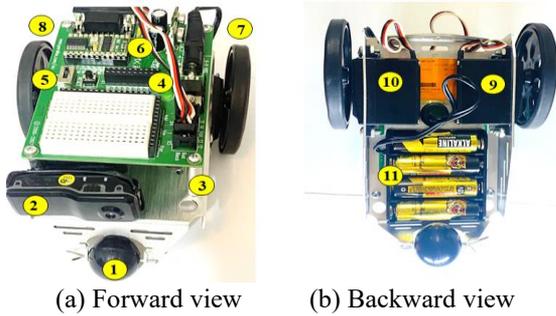


Figure 6. Two views of an experimental BOE-Bot wheeled mobile robot

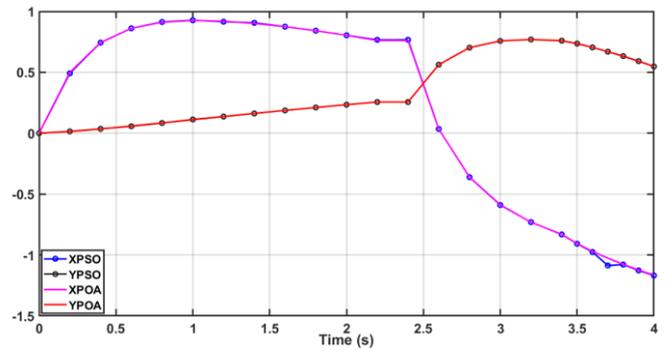


Figure 8. The (x, y) points for mobile robot trajectories of two proposed algorithms POA and PSO VS time

6. SIMULATION RESULTS

The simulation was programmed by the MATLAB R2018b software. Table 4 shows the comparison between the POA and PSO approaches in terms of performance for the mobile robot's trajectory. This is because of the minimization of the cost function (MSE) as in Eq. (16) and Eq. (24). The performance of the objective function values during the optimization method based on proposed optimization algorithms is presented in Table 4.

Table 4. POA parameters

Proposed Algorithm	Cost Function (MSE) Minimization
PSO	0.0145
POA	1×10^{-11}

Figure 7 shows the results of two objective functions as described in Table 4 above for proposed PSO VS proposed POA algorithms. The POA trajectory is simulated in red color while the PSO algorithm is simulated in blue color with the 25 iterations.

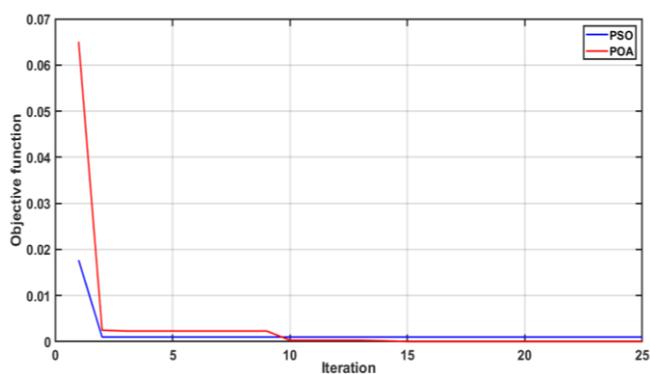


Figure 7. The evaluation of the objective functions VS the iteration

Figure 8 shows the points (x, y) trajectories of PSO and POA of proposed algorithms VS time. These points for the trajectory response of the proposed POA algorithm are better than the points for the trajectory response of the POA algorithm.

Figure 9 shows the orientation(θ) named theta of mobile robot trajectories of PSO and POA of proposed algorithms VS time. The point (x, y) and direction (θ) of a mobile robot trajectory of Figures 8 and 9 started from $(0, 0, 0)$ and ended with $(-1.1653, 0.5482, 90)$ degrees.

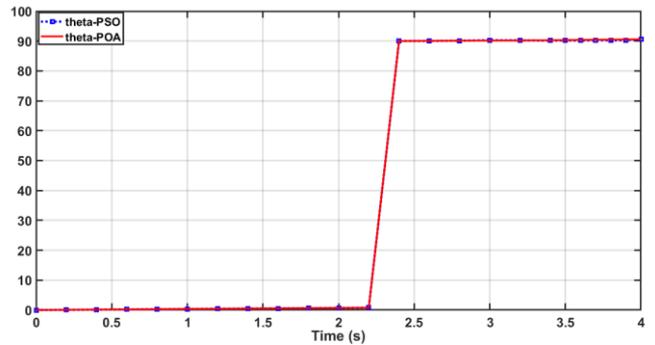


Figure 9. The (θ) orientation for mobile robot trajectories of two proposed algorithms POA and PSO VS time

Figure 10 shows an optimal trajectory of the proposed POA when the center wheeled mobile robot moving from the start point started from $(0, 0, 0)$ and ended with $(-1.1653, 0.5482, 90)$ degrees. The first obstacle is the pink square, while the second obstacle is the red square. This optimal trajectory of the proposed POA trajectory is presented in red color. The wheeled mobile robot dimension is taken from a real mobile robot as mentioned in Table 1. The linear velocities have been chosen ($v_r > v_l$) as mentioned in Table 2 and Table 3.

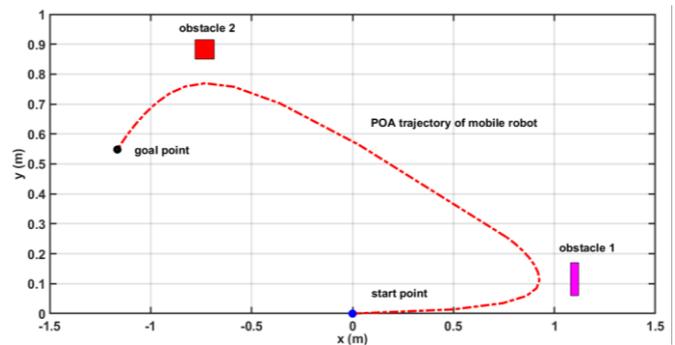


Figure 10. The optimal mobile robot trajectory of the proposed POA algorithm avoiding 2 static obstacles

7. EXPERIMENTAL RESULTS

Figure 11 shows an experiment mobile wheeled robot trajectory moving from start to goal point based on the proposed POA algorithm. At time zero, the mobile robot stands and the WiFi camera detects the object. A center mobile robot (x, y, θ) point computed from the camera and proposed

POA algorithm. Then, at the time of one second the mobile robot faced the first obstacle. At two seconds, A wheeled BOE–Bot robot is moved to the left side and avoids the first obstacle. At three seconds, A wheeled BOE–Bot robot detected the second obstacle. At four seconds, the mobile robot rotated the left side avoiding obstacle 2 then moved to the end point. A wheeled mobile robot can bypass the obstacles without contact collision and move according to the goal point successfully as compared with the simulated results in Figure 9.

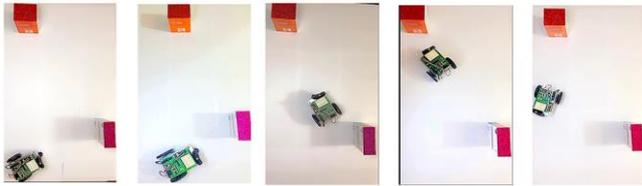


Figure 11. The experiment mobile robot of the optimal trajectory of the proposed POA algorithm avoiding 2 static obstacles

To cover this work's study to future works. One suggestion is to use modern artificial intelligence techniques to improve the path planning [30-44]. In order to make real path planning, the dynamic of mobile robot may be considered and hence a recent control schemes are required [45-54]. Finally, one may use other optimization techniques to replace and compare the proposed method [55-58].

8. CONCLUSION

In this work, two optimization techniques, PSO and POA, have been proposed and to conduct a mobile robot through an optimal path to avoid static obstacles. The algorithms of both methods have been developed and implemented in MATLAB environment. The mathematical model of the wheeled robot has been derived in terms of Cartesian coordinates (x, y) and orientation θ . The robot has been guided from $(0, 0, 0^\circ)$ to $(-1.1653, 0.5482, 90^\circ)$ without collision of any static obstacles based on both techniques. The results showed that the POA gives less MSE than that based on PSO, where the POA yields 10^{-11} , while the PSO results in 0.0145. This indicates that the POA gives is more optimal path than PSO method. In order to validate the numerical simulation, experimental results have been conducted using BOE–Bot wheeled mobile robot. It has been shown that the whole scenario is completed in 4 s.

REFERENCES

[1] Khaleel, H.Z., Raheem, F.A. (2024). Design and implementation accurate three distance sensors device using neural network. *Tikrit Journal of Engineering Sciences*, 31(2): 138-147. <https://doi.org/10.25130/tjes.31.2.13>

[2] Pandey, A., Pandey, S., Parhi, D.R. (2017). Mobile robot navigation and obstacle avoidance techniques: A review. *International Robotics & Automation Journal*, 2(3): 00022.

[3] Sanchez-Ibanez, J.R., Pérez-del-Pulgar, C.J., García-Cerezo, A. (2021). Path planning for autonomous mobile

robots: A review. *Sensors*, 21(23): 7898. <https://doi.org/10.3390/s21237898>

[4] Ibrahim, M.I., Sariff, N., Johari, J., Buniyamin, N. (2014). Mobile robot obstacle avoidance in various type of static environments using fuzzy logic approach. In 2014 2nd International Conference on Electrical, Electronics and System Engineering (ICEESE), Kuala Lumpur, Malaysia, pp. 83-88. <https://doi.org/10.1109/ICEESE.2014.7154600>

[5] Lyu, Y., Mo, Y., Yue, S., Hong, L. (2023). A mobile robot path planning using improved beetle swarm optimization algorithm in static environment. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 45(6): 11453-11479. <https://doi.org/10.3233/JIFS-224163>

[6] Shao, J. (2021). Robot path planning method based on genetic algorithm. In *Journal of Physics: Conference Series*, Stanford, United States, p. 022046. <https://doi.org/10.1088/1742-6596/1881/2/022046>

[7] Raheem, F.A., Khaleel, H.Z., Kashan, M.K. (2018). Robot arm design for children writing ability enhancement using cartesian equations based on ANFIS. In 2018 Third Scientific Conference of Electrical Engineering (SCEE), Baghdad, Iraq, pp. 150-155. <https://doi.org/10.1109/SCEE.2018.8684038>

[8] Humaidi, A., Hameed, M. (2019). Development of a new adaptive backstepping control design for a non-strict and under-actuated system based on a PSO tuner. *Information*, 10(2): 38. <https://doi.org/10.3390/info10020038>

[9] Humaidi, A.J., Abdulkareem, A.I. (2019). Design of augmented nonlinear PD controller of Delta/Par4-like robot. *Journal of Control Science and Engineering*, 7689673. <https://doi.org/10.1155/2019/7689673>

[10] Hassan, R.F., Yasin, N.M. (2018). Model predictive control based stacked asymmetric multilevel inverter driver. *International Journal of Engineering & Technology*, 7(4): 3924-3929.

[11] Humaidi, A.J., Kadhim, S.K., Gataa, A.S. (2022). Optimal adaptive magnetic suspension control of rotary impeller for artificial heart pump. *Cybernetics and Systems*, 53(1): 141-167. <https://doi.org/10.1080/01969722.2021.2008686>

[12] Rasheed, L.T. (2023). An optimal modified Elman - PID neural controller design for DC/DC boost converter model. *Journal of Engineering Science and Technology*, 18(2): 880-901.

[13] Al-Jodah, A., Abbas, S.J., Hasan, A.F., Humaidi, A.J., Mahdi Al-Obaidi, A.S., Al-Qassar, A.A., Hassan, R.F. (2023). PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot. *International Review of Applied Sciences and Engineering*, 14(1): 58-67. <https://doi.org/10.1556/1848.2022.00420>

[14] Korial, A.E., Gorial, I.I. (2022). System analysis and controllers performance comparison for D.C. motor. *International Journal of Mechanical Engineering and Robotics Research*, 11(7): 520-526.

[15] Hamoudi, A.K., Rasheed, L.T. (2024). Design of an adaptive integral sliding mode controller for position control of electronic throttle valve. *Journal Européen des Systèmes Automatisés*, 57(3): 729-735. <https://doi.org/10.18280/jesa.570310>

[16] Dong, L., He, Z., Song, C., Sun, C. (2023). A review of mobile robot motion planning methods: From classical

- motion planning workflows to reinforcement learning-based architectures. *Journal of Systems Engineering and Electronics*, 34(2): 439-459. <https://doi.org/10.23919/JSEE.2023.000051>
- [17] Khaleel, H.Z., Humaidi, A.J. (2024). Towards accuracy improvement in solution of inverse kinematic problem in redundant robot: A comparative analysis. *International Review of Applied Sciences and Engineering*, 15(2): 242-251. <https://doi.org/10.1556/1848.2023.00722>
- [18] Alam, M.S., Rafique, M.U., Khan, M.U. (2020). Mobile robot path planning in static environments using particle swarm optimization. *arXiv preprint arXiv:2008.10000*. <https://doi.org/10.48550/arXiv.2008.10000>
- [19] Trojovský, P., Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, 22(3): 855. <https://doi.org/10.3390/s22030855>
- [20] Algamal, Z., Firas, A.T., Qasim, O. (2023). Kernel semi-parametric model improvement based on quasi-oppositional learning pelican optimization algorithm. *Iraqi Journal for Computer Science and Mathematics*, 4(2): 156-165. <https://doi.org/10.52866/ijcsm.2023.02.02.013>
- [21] Khaleel, H.Z., Ahmed, A.K., Al-Obaidi, A.S.M., Luckyardi, S., Al Husaeni, D.F., Mahmud, R.A., Humaidi, A.J. (2024). Measurement enhancement of ultrasonic sensor using pelican optimization algorithm for robotic application. *Indonesian Journal of Science and Technology*, 9(1): 145-162. <https://doi.org/10.17509/ijost.v9i1.64843>
- [22] Khaleel, H.Z., Oleiwi, B.K. (2024). Ultrasonic sensor decision-making algorithm for mobile robot motion in maze environment. *Bulletin of Electrical Engineering and Informatics*, 13(1): 109-116. <https://doi.org/10.11591/eei.v13i1.6560>
- [23] Klancar, G., Zdesar, A., Blazic, S., Skrijanc, I. (2017). *Wheeled mobile robotics: From fundamentals towards autonomous systems*. Butterworth-Heinemann.
- [24] Mishra, D., Yadav, R.S., Agrawal, K.K. (2020). Kinematic modelling and emulation of robot for traversing over the pipeline in the refinery. *Microsystem Technologies*, 26(3): 1011-1020. <https://doi.org/10.1007/s00542-019-04615-9>
- [25] Aljuboury, A.S., Hameed, A.H., Ajel, A.R., Humaidi, A.J., Alkhayyat, A., Mhdawi, A.K.A. (2022). Robust adaptive control of knee exoskeleton-assistant system based on nonlinear disturbance observer. *Actuators*, 11(78):1-18. <https://doi.org/10.3390/act11030078>
- [26] Abed, A.M., Rashid, Z. N., Abedi, F., Zeebaree, S.R., Sahib, M.A., Mohamad Jawad, A.J.A., Al-khaykan, A. (2022). Trajectory tracking of differential drive mobile robots using fractional-order proportional-integral-derivative controller design tuned by an enhanced fruit fly optimization. *Measurement and Control*, 55(3-4): 209-226. <https://doi.org/10.1177/00202940221092134>
- [27] Hamoudi, A.K., Rasheed, L.T. (2023). Design and study of adaptive backstepping control scheme for position control of the propeller driven pendulum system. *Journal Européen des Systèmes Automatisés*, 56(2): 281-289. <https://doi.org/10.18280/jesa.560213>
- [28] Shero, F.F., Al-Ani, G.S., Khadim, E.J., Khaleel, H.Z. (2020). Assessment of linear parameters of electrohysterograph (EHG) in diagnosis of true labor. *Ann Trop Med & Public Health*, 23(4): 139-147.
- [29] DHgate. Wholesale 4k Ultra Camera. <https://www.dhgate.com/product/universal-brand-new-car-mount-holder-for/97336894.html>.
- [30] Abbas, I.M., Hassan, R.F. (2023). Fuzzy logic based performance enhancement of direct torque control using T-type inverter. *AIP Conference Proceedings*, 2804(1): 050031
- [31] Al-Khazraji, H., Nasser, A.R., Hasan, A.M., Al Mhdawi, A.K., Al-Raweshidy, H., Humaidi, A.J. (2022). Aircraft engines remaining useful life prediction based on a hybrid model of autoencoder and deep belief network. *IEEE Access*, 10: 82156-82163. <https://doi.org/10.1109/ACCESS.2022.3188681>
- [32] Kasim, M.Q., Hassan, R.F. (2021). Reduced computational burden model predictive current control of asymmetric stacked multi-level inverter based STATCOM. In *2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS)*, Shah Alam, Malaysia.
- [33] Korial, A.E., Gorial, I.I., Humaidi, A.J. (2024). An improved ensemble-based cardiovascular disease detection system with chi-square feature selection. *Computers*, 13(6): 126. <https://doi.org/10.3390/computers13060126>
- [34] Xu, X., Chen, H., Scheutzow, M., Simsarian, J.E., Ryf, R., Qua, G., Neilson, D.T. (2024). Automated fiber switch with path verification enabled by an AI-powered multi-task mobile robot. *Journal of Optical Communications and Networking*, 16(6): 624-630. <https://doi.org/10.1364/JOCN.517147>
- [35] Liu, H., Shen, Y., Yu, S., Gao, Z., Wu, T. (2024). Deep reinforcement learning for mobile robot path planning. *arXiv preprint arXiv:2404.06974*. <https://doi.org/10.48550/arXiv.2404.06974>
- [36] Borkar, K.K., Singh, M.K., Dasari, R.K., Babbar, A., Pandey, A., Jain, U., Mishra, P. (2024). Path planning design for a wheeled robot: A generative artificial intelligence approach. *International Journal on Interactive Design and Manufacturing*, 2024: 1-12. <https://doi.org/10.1007/s12008-023-01721-x>
- [37] Gorial, I.I., Korial, A.E. (2022). Modeling and control of ball and beam problem with controllers' performance comparison. *International Journal on Engineering Applications*, 10(6): 453-459.
- [38] Han, C., Li, B. (2023). Mobile robot path planning based on improved A* algorithm. In *2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 672-676. <https://doi.org/10.1109/ITAIC58329.2023.10408799>
- [39] Puriyanto, R.D., Mustofa, A.K. (2024). Design and implementation of fuzzy logic for obstacle avoidance in differential drive mobile robot. *Journal of Robotics and Control*, 5(1): 132-141. <https://doi.org/10.18196/jrc.v5i1.20524>
- [40] Williams, R., Ali, S., Alcantara, R., Burghleh, T., Alghowinem, S., Breazeal, C. (2024). Doodlebot: An educational robot for creativity and AI literacy. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, Boulder, CO, USA, pp. 772-780. <https://doi.org/10.1145/3610977.3634950>
- [41] Hernangómez, R., Palaios, A., Watermann, C., Schäufele, D., Geuer, P., Ismayilov, R., Stańczak, S. (2024). Toward an AI-enabled connected industry: AGV communication

- and sensor measurement datasets. *IEEE Communications Magazine*, 62(4): 90-95. <https://doi.org/10.1109/MCOM.001.2300494>
- [42] Yang, C., Kim, J., Kang, D., Eom, D.S. (2024). Vision AI system development for improved productivity in challenging industrial environments: A sustainable and efficient approach. *Applied Sciences*, 14(7): 2750. <https://doi.org/10.3390/app14072750>
- [43] Kristanto, F.H., Iklima, Z. (2024). Collision avoidance of mobile robot using Alexnet and NVIDIA Jetson Nano B01. *Journal of Integrated and Advanced Engineering*, 4(1): 9-20. <https://doi.org/10.51662/jiae.v4i1.118>
- [44] Mikołajczyk, T., Mikołajewski, D., Kłodowski, A., Łukaszewicz, A., Mikołajewska, E., Paczkowski, T., Skornia, M. (2023). Energy sources of mobile robot power systems: A systematic review and comparison of efficiency. *Applied Sciences*, 13(13): 7547. <https://doi.org/10.3390/app13137547>
- [45] Humaidi, A.J., Tala'at, E.N., Hameed, M.R., Hameed, A.H. (2019). Design of adaptive observer-based backstepping control of cart-pole pendulum system. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, pp. 1-5. <https://doi.org/10.1109/ICECCT.2019.8869179>
- [46] Mahdi, S.M., Yousif, N.Q., Oglah, A.A., Sadiq, M.E., Humaidi, A.J., Azar, A.T. (2022). Adaptive synergetic motion control for wearable knee-assistive system: A rehabilitation of disabled patients. *Actuators*, 11(7): 176. <https://doi.org/10.3390/act11070176>
- [47] Humaidi, A.J., Kadhim, S.K., Gataa, A.S. (2020). Development of a novel optimal backstepping control algorithm of magnetic impeller-bearing system for artificial heart ventricle pump. *Cybernetics and Systems*, 51(4): 521-541. <https://doi.org/10.1080/01969722.2020.1758467>
- [48] Humaidi, A.J., Hameed, A.H. (2017). Robustness enhancement of MRAC using modification techniques for speed control of three phase induction motor. *Journal of Electrical Systems*, 13(4): 723-741.
- [49] Humaidi, A.J., Hasan, S., Al-Jodah, A.A. (2018). Design of second order sliding mode for glucose regulation systems with disturbance. *International Journal of Engineering & Technology*, 7(2.28): 243-247.
- [50] Mahmood, Z.N., Al-Khazraji, H., Mahdi, S.M. (2023). Adaptive control and enhanced algorithm for efficient drilling in composite materials. *Journal Européen des Systèmes Automatisés*, 56(3): 507-512. <https://doi.org/10.18280/jesa.560319>
- [51] Ahmed, A.K., Al-Khazraji, H., Raafat, S.M. (2024). Optimized PI-PD control for varying time delay systems based on modified smith predictor. *International Journal of Intelligent Engineering & Systems*, 17(1): 331-342. <https://doi.org/10.22266/ijies2024.0229.30>
- [52] Al-Khazraji, H., Naji, R.M., Khashan, M.K. (2024). Optimization of sliding mode and back-stepping controllers for AMB systems using gorilla troops algorithm. *Journal Européen des Systèmes Automatisés*, 57(2): 417-424. <https://doi.org/10.18280/jesa.570211>
- [53] Al-Khazraji, H., Cole, C., Guo, W. (2018). Analysing the impact of different classical controller strategies on the dynamics performance of production-inventory systems using state space approach. *Journal of Modelling in Management*, 13(1): 211-235. <https://doi.org/10.1108/JM2-08-2016-0071>
- [54] Korial, A.E. 2022. Brain tumor detection from MRI images using artificial intelligence. *International Journal on Engineering Applications*, 10 (3): 185-192. <https://doi.org/10.15866/irea.v10i3.21213>
- [55] Mahmood, Z.N., Al-Khazraji, H., Mahdi, S.M. (2023). PID-based enhanced flower pollination algorithm controller for drilling process in a composite material. *Annales de Chimie Science des Matériaux*, 47(2): 91-96. <https://doi.org/10.18280/acsm.470205>
- [56] Ahmed, A.K., Al-Khazraji, H. (2023). Optimal control design for propeller pendulum systems using gorilla troops optimization. *Journal Européen des Systèmes Automatisés*, 56(4): 575-582. <https://doi.org/10.18280/jesa.560407>
- [57] Naji, R.M., Dulaimi, H., Al-Khazraji, H. (2024). An optimized PID controller using enhanced bat algorithm in drilling processes. *Journal Européen des Systèmes Automatisés*, 57(3): 767-772. <https://doi.org/10.18280/jesa.570314>
- [58] Al-Khazraji, H. (2022). Comparative study of whale optimization algorithm and flower pollination algorithm to solve workers assignment problem. *International Journal of Production Management and Engineering*, 10(1): 91-98. <https://doi.org/10.4995/ijpme.2022.16736>