DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

Dynamic Obstacle Avoidance Algorithm for Autonomous Mobile Robots

Noor Abdul Khaleq Zghair[1], Ahmed S. Al-Araji[2]
[1,2]*Computer Engineering Department, University of Technology, Baghdad, Iraq*
[1]*Noor.a.zghair@uotechnology.edu.iq*, [2]*Ahmed.s.alaraji@uotechnology.edu.iq*

Abstract— *A mobile robot's major purpose is to get to its destination by traveling over an optimum path defined by various parameters such as time, distance, and the robot's safety from any impediments in its path. As a result, the backbone of the autonomous mobile robot is path planning and obstacle avoidance. Several algorithms for path planning and obstacle avoidance have been presented by various researchers, each with its own set of benefits and drawbacks. This paper focuses on two parts; the first part finds the short and smooth collision-free path for a mobile robot to navigate in a static environment based on two proposed hybrid algorithms. The first hybrid is between Firefly Algorithm (FA) and Modify Chaotic Particle Swarm Optimization (MCPSO), namely (HFACPSO), while the other hybrid is between Genetic Algorithm (GA) and MCPSO, namely (HGACPSO). The second part suggests an algorithm planner for improving the efficiency of the route-planning algorithm with moving obstacle avoidance by adjusting the velocity or re-planning the path for the mobile robot. To demonstrate the effectiveness of the proposed algorithms in terms of the shortest path length and collision-free, as well as obtaining optimal or near-optimal wheel velocities with the minimum number of iterations. The proposed hybrid (FAMCPSO) algorithm provides enhancement on the path length equal to (0.82%) compared to the firefly algorithm (FA). Moreover, the hybrid (GAMCPSO) algorithm enhancement on the path length equals (0.67%) compared to the genetic algorithm (GA). All methods are simulated in a static and dynamic obstacle environment using MATLAB 2018b.*

Index Terms— *Dynamic environment, Collision avoidance, obstacle avoidance, Path planning, PSO, FA, GA.*

I. INTRODUCTION

Given the potential effect and societal benefit of this technology, research interest in autonomous guided robots has progressively grown in the modern era. Autonomous robot control systems must be capable of making navigation decisions based on past knowledge of the working environment (building maps, sensor models, and robot dynamics) as well as observations about the robot's vicinity. The observations originate from the perception system, which can include odometry, cameras, and sonars, as well as other sensors [1].

On the other hand, most mobile robot application environments, are dynamic, which implies that mobile robots must be able to construct collision-free pathways with both static and dynamic obstacles that are not included in the map and can collide with the robot's intended path. Robots can create a collision-free route in static environments by using the environment's occupancy grid map. However, in dynamic situations, the robot must account for the dynamic nature of barriers not depicted on the map, that can collide with the robot's planned path. the robot must forecast the future trajectories of these obstacles to planning its

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

path accordingly. Because of sensor limitations and inaccurate dynamic models, these forecasts have a significant level of uncertainty [2]. As a result, real-world applications of mobile robot navigation in complex and dynamic situations represents still a challenge. The robot should be able to navigate safely through moving people or vehicles to reach its desired location, despite the implied unpredictability of the environment and the limitations of its vision system [3]. The problem of autonomous path planning and navigation has been studied in literature and many review articles about robot path planning have been published. For instance, A* algorithms [4], D* algorithms [5], fuzzy logic (FL) [6], genetic algorithms (GA) [7], particle swarm algorithms (PSO) [8], ant colony algorithms (ACO) [9], simulated annealing (SA) [10], Artificial Potential Fields [11], Probabilistic Road map (PRM) [12], and so on. Each has its advantages and drawbacks in different environments. The algorithm's optimization concerns the path length and total time consumed while avoiding any collisions with obstacles. Obstacle avoidance, on the other hand, is a critical approach for the design and implementation of mobile robots, as static and even dynamic impediments frequently appear in their pathways. When numerous robots move near one another, they become impediments to one another [13]. As a result, obstacle avoidance research has become a hot issue in the realm of mobile robotics, and several solutions for avoiding static and/or dynamic barriers have been presented. Such as: In [14] proposed a modified potential field approach to tackle the mobile robot motion planning problem in dynamic situations by defining the collision angle with exponential form as the constraint having the repulsive potential field. In addition to the angle and magnitude of the robot's relative velocity and that of the obstacle. In [15] introduced Smoothly RRT (S-RRT), a dynamic path-planning solution for autonomous robotic manipulator obstacle avoidance based on an enhanced Rapidly Exploring Random Tree (RRT) algorithm. This technique revealed that the manipulator can avoid not just a static global impediment, but also a dynamic obstacle that may develop unexpectedly in an unstructured dynamic environment. In [16], the authors suggested a simulated annealing-based method for determining the best path for a mobile robot in dynamic settings containing static and dynamic impediments. This approach employed the vertices of the obstacles to creating the search space, which was computed based on known static obstacles and then re-calculated the path in real-time whenever a moving obstacle was identified. While in [17] proposed approach combining distance computation and discrete collision detection (DCD) is suggested to avoid dynamic obstacles during industrial operations performed by manipulators. The Gilbert–Johnson–Keerthi algorithm calculates the closest distances between the links of a manipulator and the convex hull of an arbitrarily-shaped dynamic obstacle obtained in real-time from the Kinect-V2 camera, and the minimum one is defined as the closest distance between the manipulator and the obstacle. In [18] determined the shortest and smoothest collision-free path under several static and dynamic scenarios based on swarm intelligence optimization and improved two path planning algorithms. The first is a modified frequency bat algorithm (MFB), while the second is a mix of particle swarm optimization and the modified frequency bat algorithm (MFB) (Hybrid PSO-MFB). In this work, we suggest a solution to the problem of mobile robot path planning in a dynamic environment by using two hybrid algorithms, namely (Hybrid FFCPSO) and (Hybrid GACPSO). The hybrid algorithms generate a shorter path, an improved distance cost function, and good obstacle avoidance that may be used in a dynamic environment. The rest of this paper is organized as follows: Section (2) describes the kinematic of the wheeled mobile robot. Section (3) explains the dynamic obstacle avoidance method depending on the proposed hybrid path planning

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

algorithms. While Section (4) displays the numerical results and analysis of the MATLAB simulation in a dynamic environment, and finally the conclusion is discussed in Section (5).

II. KINEMATIC OF THE WHEELED MOBILE ROBOT

In general, the wheeled mobile robot platform depicted in *Fig. 1* comprises left and right wheels placed on a parabolic shaft, with two multi-directional wheels fitted in the front or back for stabilization. The two separate analog Direct Current (DC) motors that provide torque to the right and left wheels of the mobile robot dictate the robot's motion and steering [19]. The center mass of the wheel's mobile robot is denoted by C_m, which is halfway between the right and left wheels.

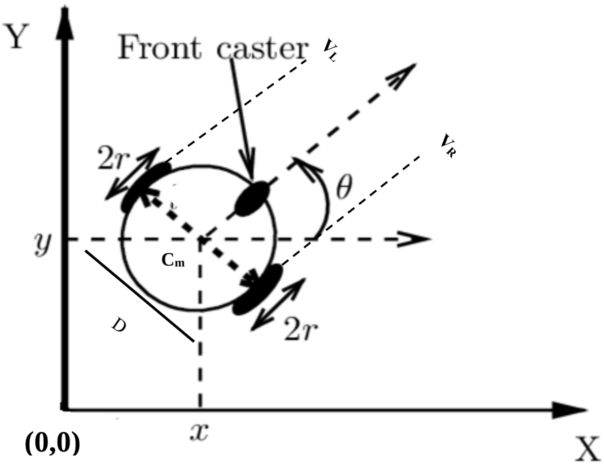


FIG. 1. KINEMATIC MODEL OF A DIFFERENTIAL-DRIVE MOBILE ROBOT [20].

In general, the global reference frame is [X, Y], while the position vector of the local reference frame of the mobile robot is defined as Ɋ in Eq. (1):

$$Ɋ = [x, y, \theta]^T \tag{1}$$

Where (x,y) is the position coordinates at midway C_m, and θ is the nonholonomic constraint operating on the motion's orientation. So, the three kinematic equations for nonholonomic wheels mobile robots can be represented as in Eq. (2), (3), and (4):

$$x(t) = \frac{1}{2}[V_R(t) + V_L(t)]\cos\theta(t)T_S + x(t-1) \tag{2}$$

$$y(t) = \frac{1}{2}[V_R(t) + V_L(t)]\sin\theta(t)T_S + y(t-1) \tag{3}$$

$$\theta(t) = \frac{1}{D}[V_L(t) - V_R(t)]T_S + \theta(t-1) \tag{4}$$

Where $V_R(t)$ and $V_L(t)$ are the right and left wheel velocity of the platform respectively; the distance between the driving wheels of the platform is taken as D; and the sampling time of the numerical calculation is denoted by Ts [19]-[21].

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

III. DYNAMIC OBSTACLE AVOIDANCE ALGORITHM

If we wanted to transport a mobile robot to a certain location, especially in a dynamic environment, the first challenge it would encounter is selecting the best way to take, followed by avoiding obstacles and arriving at the destination with acceptable precision [22], [23]. As a result, the ability to avoid obstacles is crucial. To maintain the path as short as possible, the robot must be trusted to complete its task without hurting itself or others. Many algorithms have been proposed in recent years to develop an ideal path and avoid colliding with obstacles [24]–[27]. The suggested hybrid algorithms and collision avoidance strategies are explained in this section.

A. Collision Avoidance Methods

For mobile robots to achieve collision-free motion planning, they must first detect and identify obstacles. mobile Robot collision avoidance is divided into three parts, according to the collision avoidance process: perception of obstacles, collision decision, and collision avoidance [25]. Furthermore, the robot must decide whether it will collide with the identified obstacle depending on various criteria after establishing the position and size of the obstacle. A safe distance between the robot and all obstacles might be one of these criteria used in this article [28]. The distance (D) between the obstacle and the mobile robot is calculated according to Eq. (5). After that, the user normally determines the safe distance N according to Eq. (6). So, if the distance between a robot and a detected obstacle is smaller than the set safety distance, in which case the collision avoidance system will be engaged to assist the robot in escaping the impediment. Unlike that, the mobile robot completes the road to the goal.

$$\text{Distance} = \sqrt{(\mathrm{x_r} - \mathrm{x_{obs}})^2 + (\mathrm{y_r} - \mathrm{y_{obs}})^2} \tag{5}$$

$$\text{Distance} <= \mathrm{N} \tag{6}$$

Where x_r and y_r indicate the position of the mobile robot in the X and Y axis; while x_{obs}, and y_{obs} are the position of moving obstacles in the X and Y axis; N is the safe distance and proposed equal to 75 cm.

After obstacles are detected and threat analysis is done, the proposed strategy of obstacle avoidance are: to increase or/ decrease the velocity of the mobile robot, or otherwise, replan another path to avoid moving obstacle. A decision is taken by the robot depending on the obstacle velocity, the obstacles change their location continuously at each time step, and, in dynamic environments, the position of the obstacle (x_{obs}, y_{obs}) is updated according to the following relationship as shown in Eqs. (7) and (8), suppose that the obstacles move linearly at speed as V_{obs} and direction θ_{obs}.

$$\mathrm{x_{obs}} = \mathrm{x_{obs}} + \mathrm{V_{obs}} \times \mathrm{Cos}\,\theta_{\mathrm{obs}} \tag{7}$$

$$\mathrm{y_{obs}} = \mathrm{y_{obs}} + \mathrm{V_{obs}} \times \mathrm{Sin}\,\theta_{\mathrm{obs}} \tag{8}$$

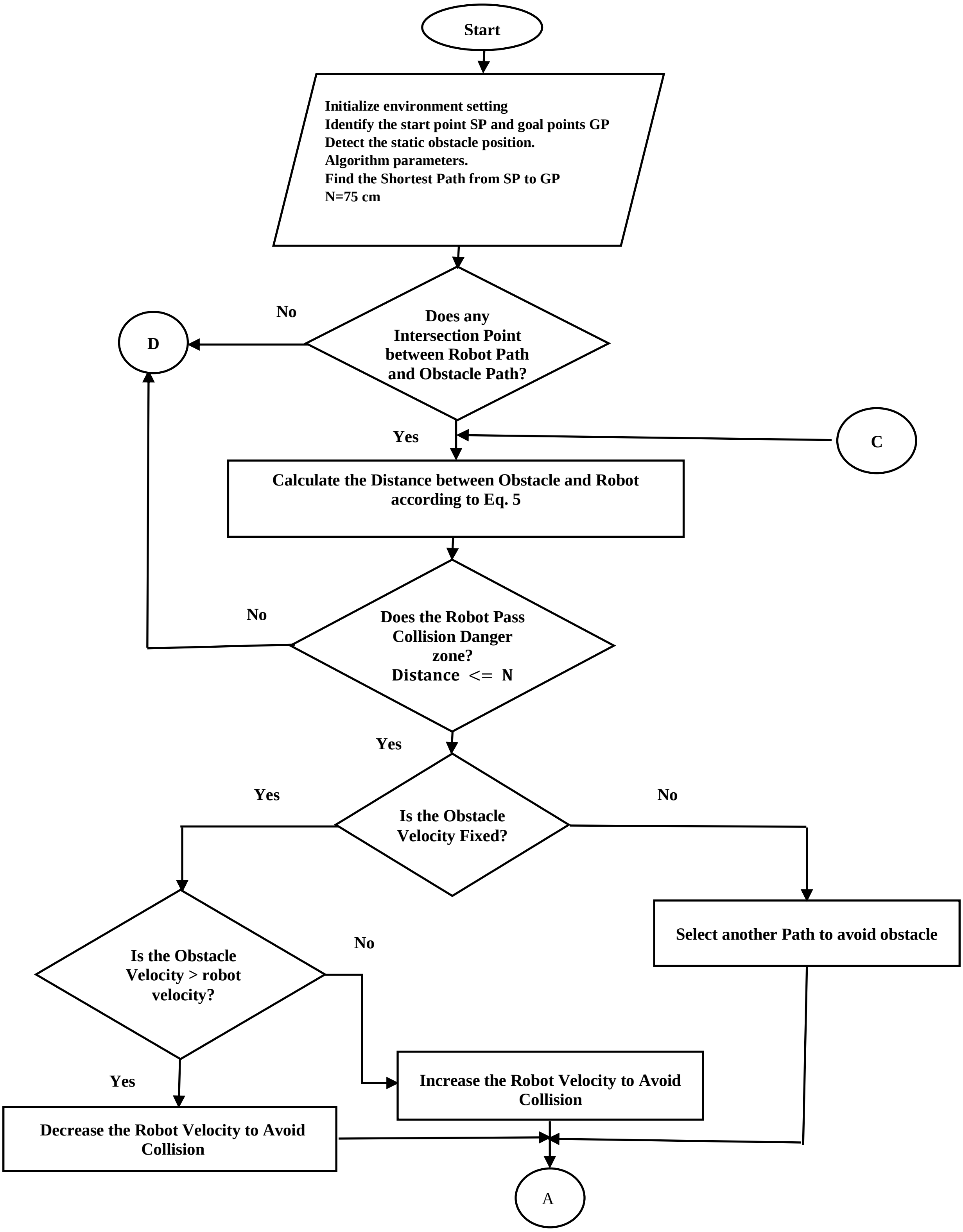
The flow chart of the proposed dynamic obstacle avoidance method is shown in *Fig. 2*. On the other hand, *Fig. 3* presents the robot traveling to the goal and the obstacle moving from the bottom to the top, then the mobile robot chooses the first proposed method to avoid the collision by increasing or decreasing its speed in the X direction.

Received 30/July/2022; Accepted 23/August/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

After that, if the mobile robot passes the obstacle, it returns to the previous velocity and continues down the path until it reaches the goal location. While, *Fig. 4* shows the second proposed method for avoiding collision, which is to choose a different path to avoid a dynamic obstacle in the collision zone. Where the green circle represents the collision region while the red point represented the intersection point between the robot path and the obstacle path.



Received 30/July/2022; Accepted 23/August/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

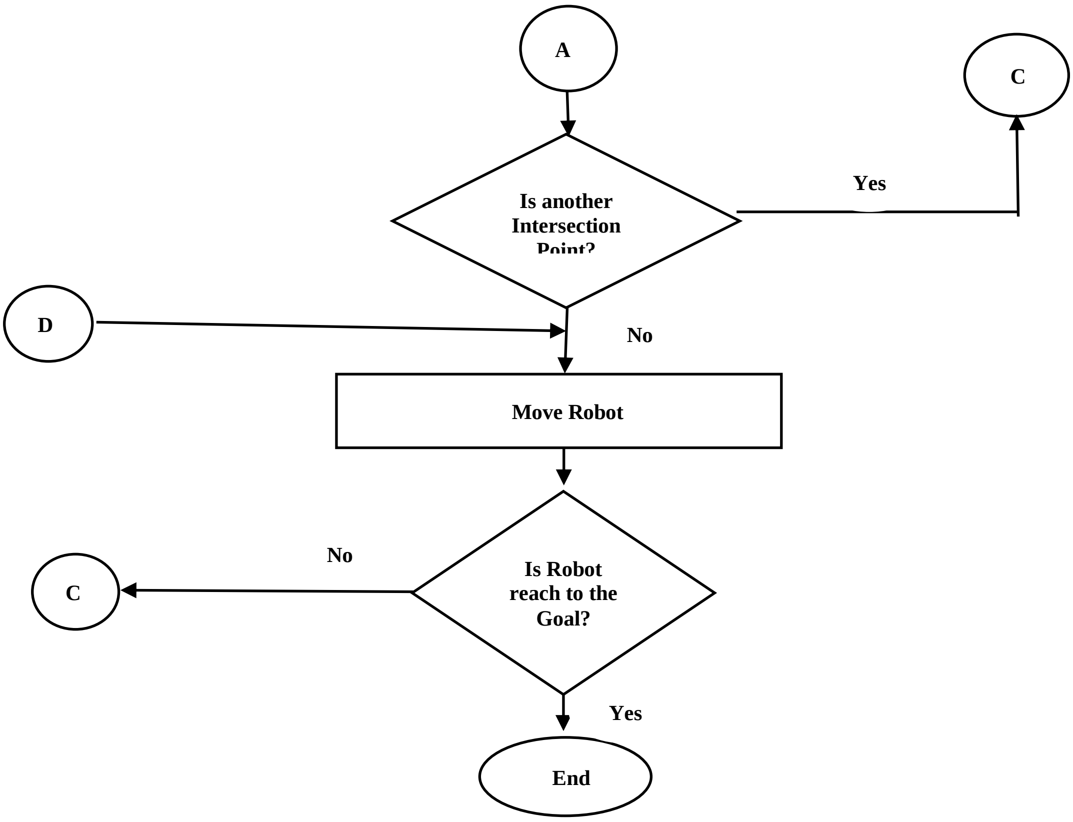


FIG. 2. THE FLOW CHART OF THE SUGGESTED DYNAMIC OBSTACLE AVOIDANCE METHOD.

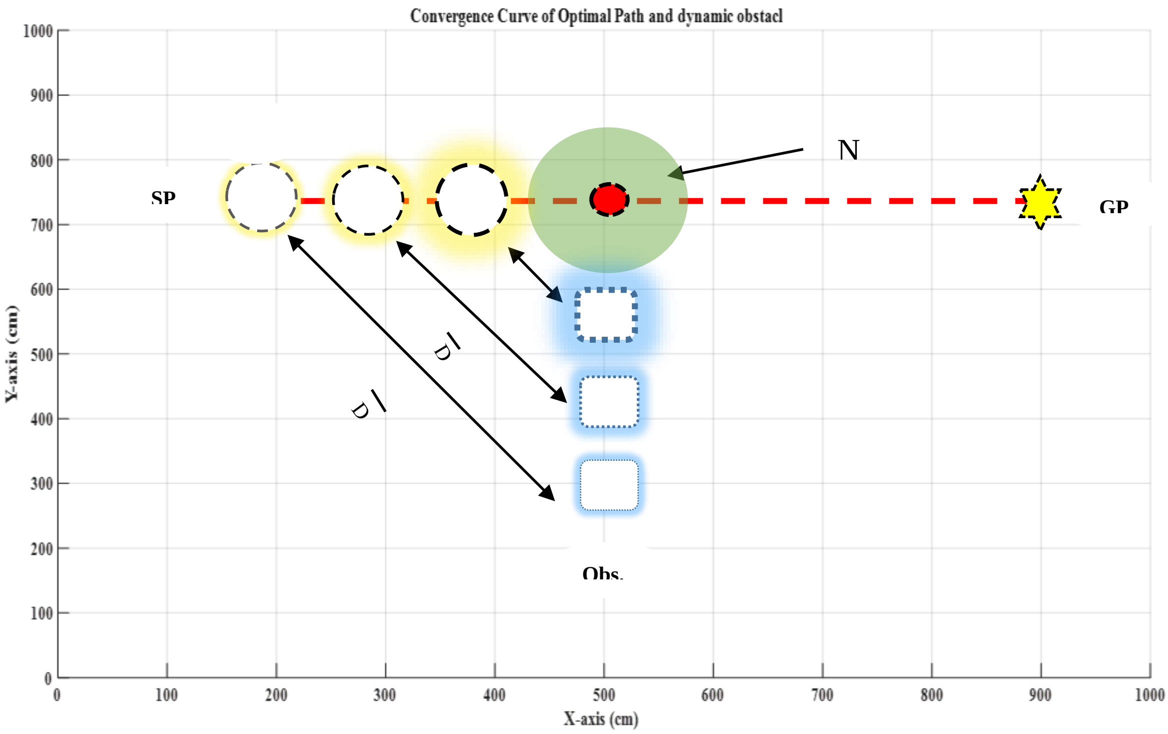


FIG. 3. THE FIRST PROPOSED METHOD TO AVOID THE COLLISION.

Received 30/July/2022; Accepted 23/August/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

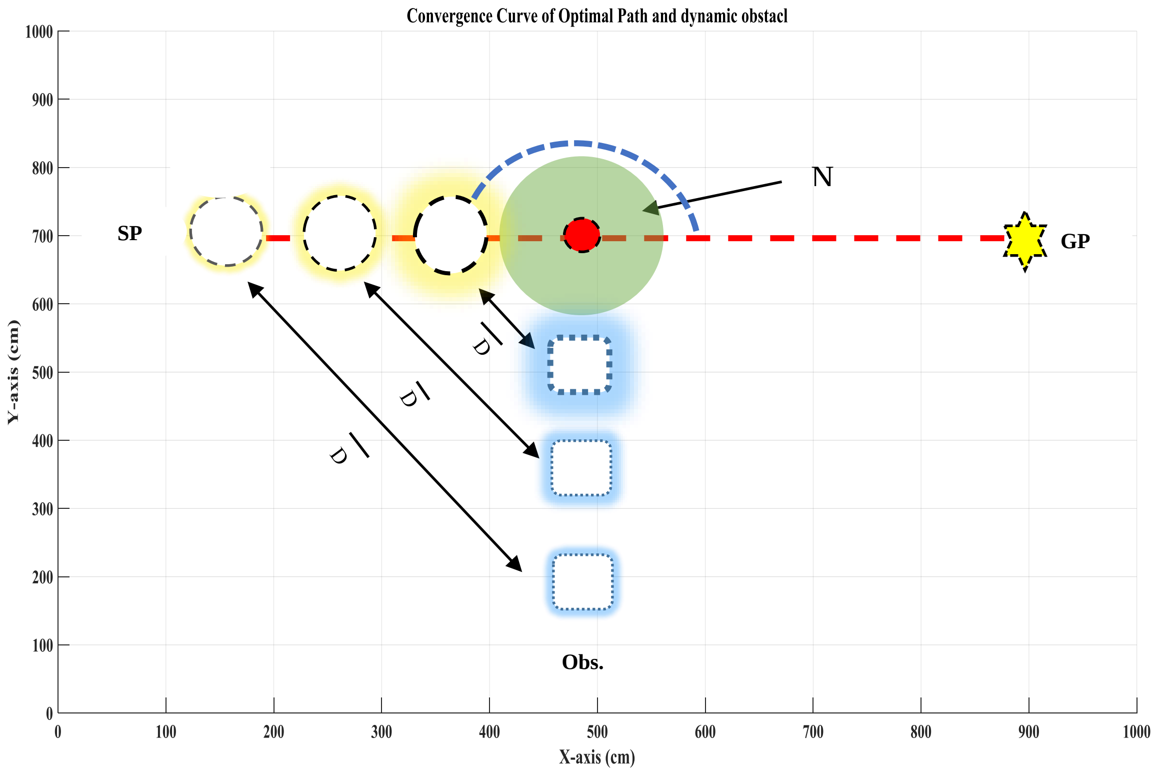


FIG. 4. THE SECOND PROPOSED METHOD TO AVOID THE COLLISION.

B. Hybrid Firefly Algorithm with Modify Chaotic Particle Swarm Optimization (Hybrid FFMCPSO)

By combining the strengths of the two techniques, a hybridization of the firefly and modify chaotic particle swarm optimization (Hybrid FFMCPSO) was suggested. The MCPSO approach is highly successful and performs quick searching due to the velocity parameter's fast convergence. The algorithm occasionally fails to generate the best results due to oscillations in local searches for more details about MCPSO, see [29]. FF, on the other hand, is incapable of maintaining a personal best position and does not have a velocity characteristic. As a result, they will move, regardless of their previous best places. *Fig. 5* depicts the flowchart of the suggested hybrid algorithm.

As a result, based on the local search space conditions, fireflies can select the best option. To increase the convergence of the firefly algorithm and prevent it from slipping into the local minimum, MCPSO characteristics are combined with the FF algorithm to form a hybrid optimization approach known as hybrid FFMCPSO. In this paper, we initially hypothesized that by dynamically altering the acceleration coefficient (C) using an iterative process, we could enhance the optimal performance of the PSO algorithm with chaotic searching, and we got superior optimal performance compared to the chaotic state in the inertia weight (W) [30]. After applying the chaotic equations adopted in this article as shown in Eq.s (9), (10), (11), and (12). Based on this, the secondly proposed hybrid combination of FF and MCPSO will provide a better solution for the local search capabilities of firefly and the global search capabilities of the MCPSO algorithm [31], [32].

The fireflies are estimated using an objective function in the proposed hybrid algorithm-based route planning problem. To see how they are relative to the ideal solution, use the minimal path length (P_a) estimation function, which permits the mobile robot to go

Received 30/July/2022; Accepted 23/August/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

from the start point to the target position in the shortest path feasible, as given in Eq (13). In addition to the above, the proposed method incorporates the concepts of personal best and global best into FF. Except for the firefly movement, which has been updated to integrate the notions of personal and global best [33]-[35], all the methods in FF have remained the same. Consequently, the altered position vector of the FF method may be written as follows in Eq. (14) and (15):

$$Z^{itr+1} = \mu \times Z^{itr} \times (1 - Z^{itr}) \quad (9)$$

$$C = C_{MX} - \frac{(C_{MX} - C_{MN})}{MaxIt} \times itr \quad (10)$$

$$C_{1new} = C \times Z^{itr+1} \quad (11)$$

$$C_{2new} = C_{1new} \quad (12)$$

Where (μ) is the control parameter and when μ= 4 the system enters into a chaotic state, while, the initial value of deterministic is Z_0. while, in Eq. (10) C_{MX} and C_{MN} are the maximum and the minimum acceleration values, respectively, MaxIt is the maximum iterations' number, and itr is the present iteration.

$$P_a = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (13)$$

Where P_a is the distance between two points. x_i and y_i are x and y coordinates of the current waypoints. x_{i+1} and y_{i+1} are x and y coordinates of feasible waypoints in the next iteration.

$$D_{pxy} = \sqrt{\sum_{k=1}^{D} (p_{best(i,itr)} - xy_{(i,itr)})2} \quad (14)$$

$$D_{gxy} = \sqrt{\sum_{k=1}^{D} (g_{best(i,itr)} - xy_{(i,itr)})2} \quad (15)$$

Where D_{pxy} is the distance between the best local fitness values p_{best} for the i^{th} particle's position in the itr^{th} iteration. D_{gxy} is the distance between the best global fitness values g_{best} for all particles and the i^{th} particle's position in the itr^{th} iteration. The new position $x_i^{(itr+1)}$ and $y_i^{(itr+1)}$ values of the particles are calculated according to Eq. (16) denoted coordinates number in the X and Y axis, respectively.

$$xy_i^{itr+1} = W \times xy_i^{itr} + C_{1new} \times e^{-Dpxy2} \left(P_{best,i} - xy_i^{itr}\right) + C_{2new} \times e^{-Dgxy2} \left(g_{best,i} - xy_i^{itr}\right) + \alpha \pounds \quad (16)$$

Where α represents the randomization parameter [0 to 1] and £ represents a vector of random variables (rand - 0.5), making the investigation of the search distance more successful.

Received 30/July/2022; Accepted 23/August/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

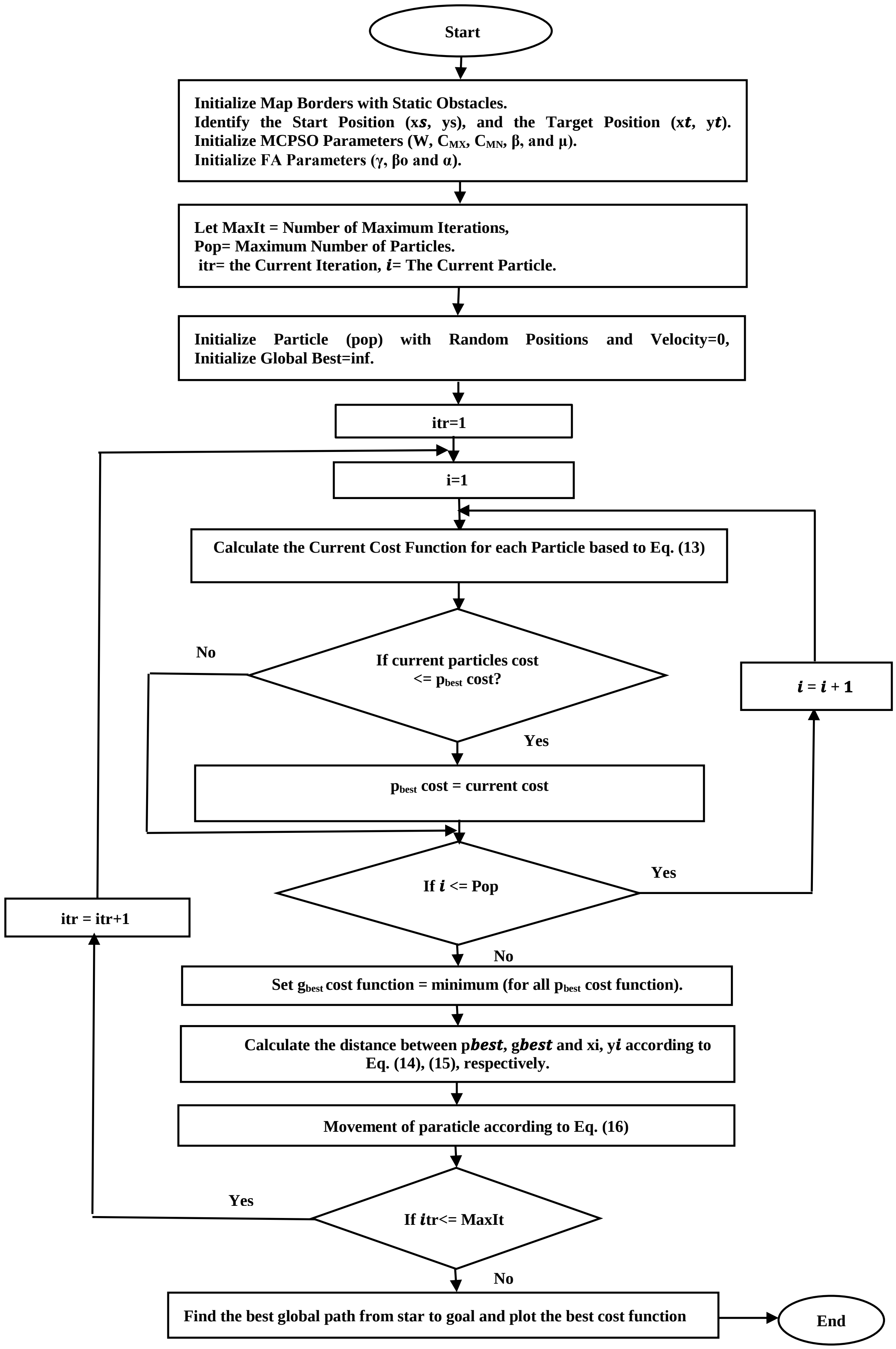


FIG. 5. THE FLOWCHART OF HYBRID FFMCPSO ALGORITHM-BASED PATH PLANNING.

Received 30/July/2022; Accepted 23/August/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

C. Hybrid Genetic Algorithm with Modify Chaotic Particle Swarm Optimization (Hybrid GAMCPSO)

To improve the efficiency of the best path from the starting position to the destination and execute a more efficient local search for the optimal solution, we incorporate the Genetic Algorithm (GA) into the Modify Chaotic Particle Swarm Optimization (MCPSO). Algorithms such as GA and MCPSO are population-based optimization techniques, each having its own set of advantages and disadvantages. In particular, GA is a global optimization method that simulates heredity and the evolution process in the environment. To achieve the survival of the fittest, it employs three operating processes: selection, crossover, and mutation for the generation of solutions, thereby leading to many function evaluations. The strength of the MCPSO, on the other hand, is that it can use swarm intelligence to replicate cooperation amongst individuals in the same group and pass on experiences from generation to generation [36]. This is because MCPSO generates solutions using mathematical operators rather than evolutionary operators. As compared to the GA approach, the PSO algorithm is characterized by its easier coding. The MCPSO approach has some advantages for exploiting and exploring the hyperspace global optimum, particularly the quick convergence. The suggested hybrid algorithm (HGACPSO) combines the advantages of these two algorithms' characteristics and overcomes the drawbacks of both. In other words, combining MCPSO's capacity to search large spaces and the fast convergence associated with GA's global search feature results in avoiding convergence ahead of time and high diversity, as well as the stochastic nature of its mutation capabilities. It is possible to incorporate crossover and mutation from the GA process into the MCPSO [37]-[39] because both algorithms have a similar working methodology of starting by randomly creating the beginning population with the initial position. Then, objective function values (cost functions) are evaluated for every chromosome depending on the position based on Eq. (13). After the evaluation is done, the chromosomes are separated into two parts according to the performance of the fitness value. The first part is the chromosomes with better fitness values (Best Cost) while the second part includes chromosomes with worse fitness values (Worst Cost). The algorithm selects two parents according to the relative fitness ($cost_{rel}$) for carrying out the roulette selection value for every individual in the population. Then the mathematical crossover is then used to create new offspring by recombining information from the two parents and it carries out mutation randomly to ensure population variety by modifying the genetic structure of some individuals according to a mutation rate. Finally, it merges the offspring obtained from the GA and updates the best solution and position to each chromosome. After that, it updates the position and velocity using the MCPSO equations and calculates the corresponding cost to update personal and global best values. The MCPSO population is then returned to the main population, the entire population is ranked, and the procedure is repeated until a convergence criterion or iteration limit is reached [36]-[38]. The equations for selection, crossover, and mutation are shown in Eq. (17) through Eq. (21) [40] while the velocity and the update position are shown in Eq. (22) and (23). The flowchart of the proposed hybrid GAMCPSO is illustrated in *Fig. 6*.

$$\mathrm{pop.cost_{rel}} = e^{\frac{-B \times \mathrm{pop.cost}}{\mathrm{worstcost}}} \tag{17}$$

$$\mathrm{pop.cost_{rel}} = \frac{\mathrm{pop.cost_{rel}}}{\mathrm{sum(pop.cost)}} \tag{18}$$

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

$$\mathrm{Offspring}_1 = \mathrm{alpha} \times p_1 + (1 - \mathrm{alpha}) \times p_2 \tag{19}$$

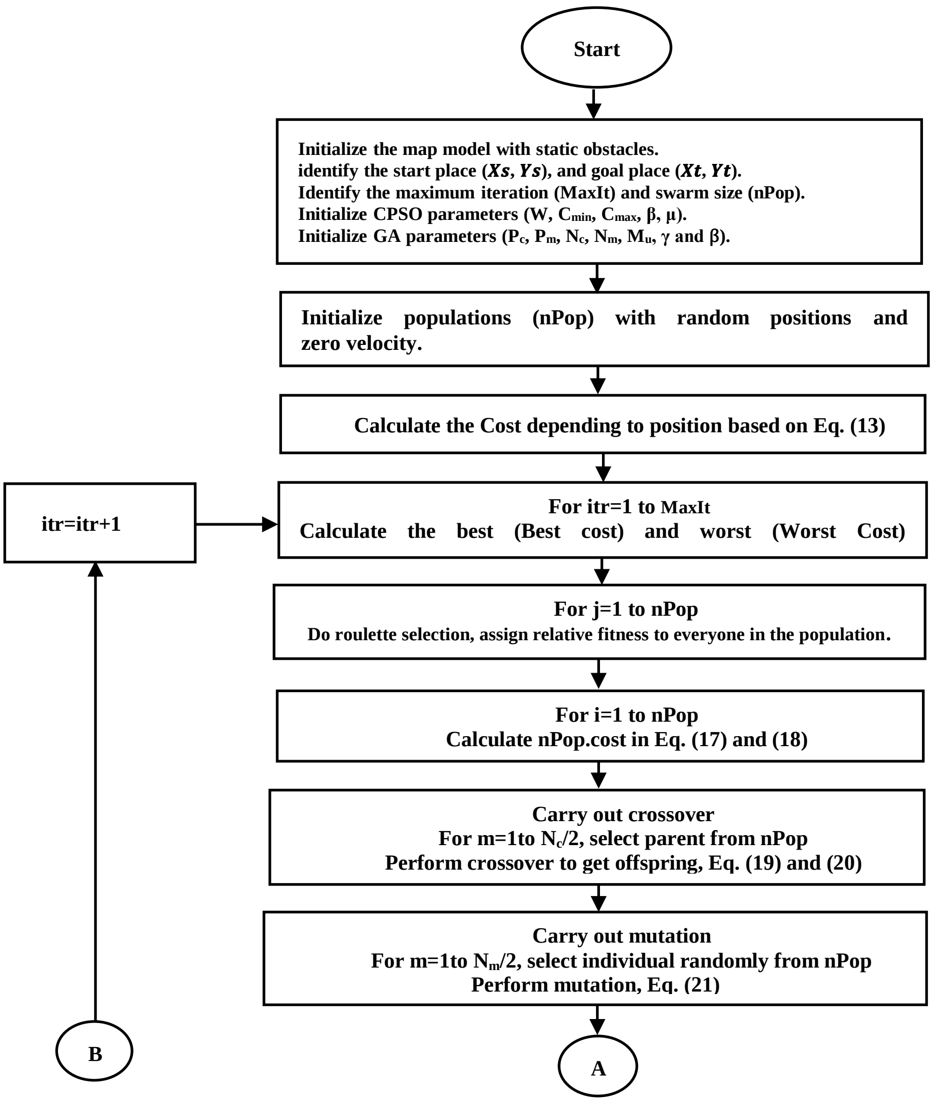
$$\mathrm{Offspring}_2 = \mathrm{alpha} \times p_2 + (1 - \mathrm{alpha}) \times p_1 \tag{20}$$

$$\mathrm{mutation} = p_1 + \alpha \times \mathrm{randn} \tag{21}$$

$$[V(i,j)]_p^{iter+1} = \begin{bmatrix} W \times V(i,j) + C_{1new} \times r_1 \times \left(P_{best}(i,j) - xy(i,j)\right) \ldots \\ +C_{2new} \times r_2 \times (G_{best}(i,j) - xy(i,j)) \end{bmatrix}_p^{iter} \tag{22}$$

$$[xy(i,j)]_p^{iter+1} = [xy(i,j)]_p^{iter} + [V(i,j)]_p^{iter+1} \tag{23}$$

Where the parameter in the GA algorithm refers to pop.cost is the cost value for each chromosome depending on the position; while $\mathrm{pop.cost}_{rel}$ is the relative fitness of each individual in the population; B represented the selection pressure; P_1 and P_2 are the two parents selected; alpha is randomly selected; α is random $[0 - 1]$. In another word, the parameter in the MCPSO algorithm refers to the proposed values of r_1 and r_2 are constant 0.9, $[xy(i,j)]_p^{itr}$ and $[V(i,j)]_p^{itr}$ of the p particle represent the position and velocity at the itr^{th} iteration, respectively, and (i, j) denotes the coordinates' values on the X and Y axis.



Received 30/July/2022; Accepted 23/August/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

Iraqi Journal of Computers, Communications, Control & Systems Engineering (IJCCCE), Vol. 23, No. 2, June 2023

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

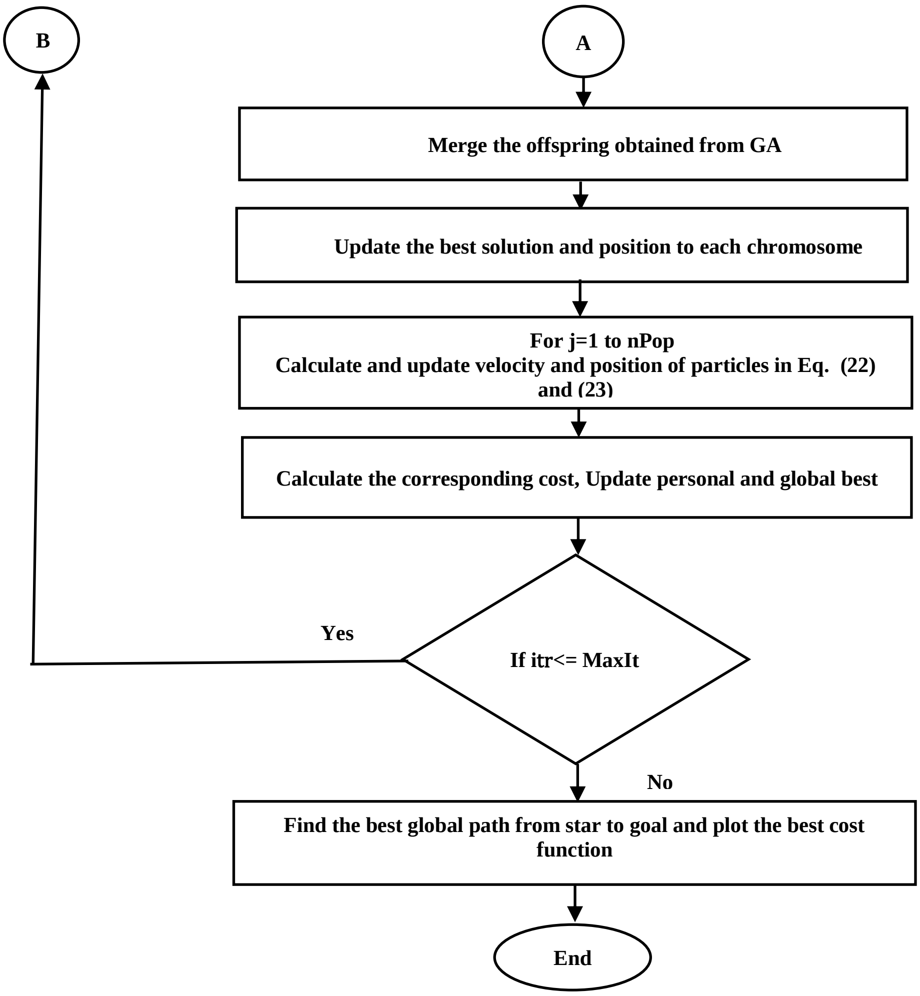


FIG. 6. THE FLOWCHART OF HYBRID GAMCPSO ALGORITHM-BASED PATH PLANNING.

IV. SIMULATION RESULT AND ANALYSIS

Two tests were carried out in the simulations: the first involved simulations in a static-obstacle environment, and the second involved simulation results in a dynamic-obstacle environment. The size of the search space in each simulation was [800 × 600] cm^2 per grid cell. The route planning hybrid algorithm (HFAMCPSO & HGAMCPSO) plans the full path from the start point [75, 750] cm (yellow-point), to the endpoint [470, 300] cm (yellow-point). The efficiency of the suggested hybrid algorithms was investigated in this paper using MATLAB software (2018b) in the computer hardware specifications including Intel Core i7-10750H with 16.0 GB of RAM, and a CPU of 2.60GHz.

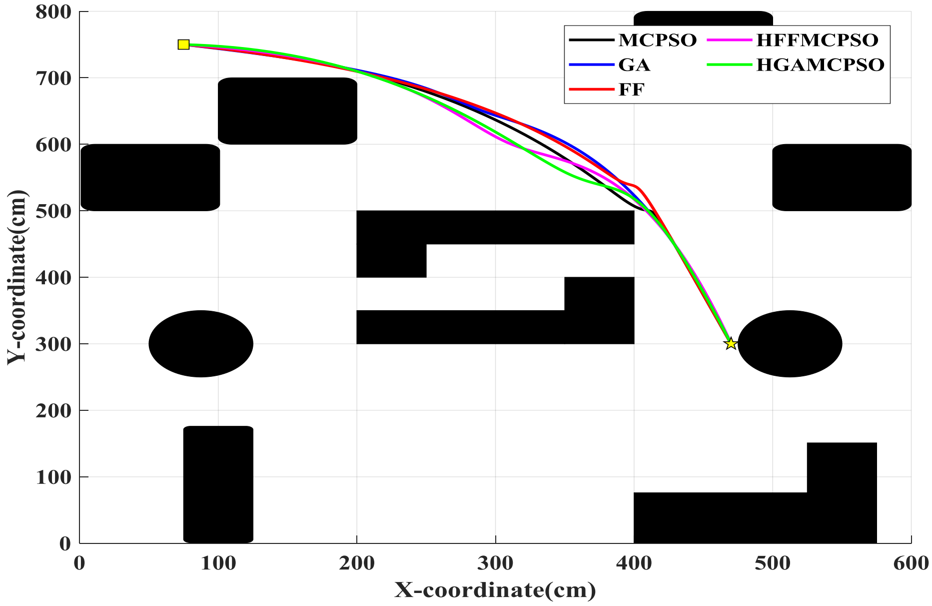
Experiment 1: Static-Obstacle Environment

The static-obstacle environment was used in this experiment to demonstrate the efficacy of the suggested hybrid method for path planning. The mobile robot's static workspace comprises ten static obstacles of various sizes and positions. The suggested hybrid algorithms were applied in the same environment and compared to discover the optimal cost distance function and optimum path several times with various iterations' numbers ranging from 50 to 100 and particles' numbers ranging from 25 to 50.

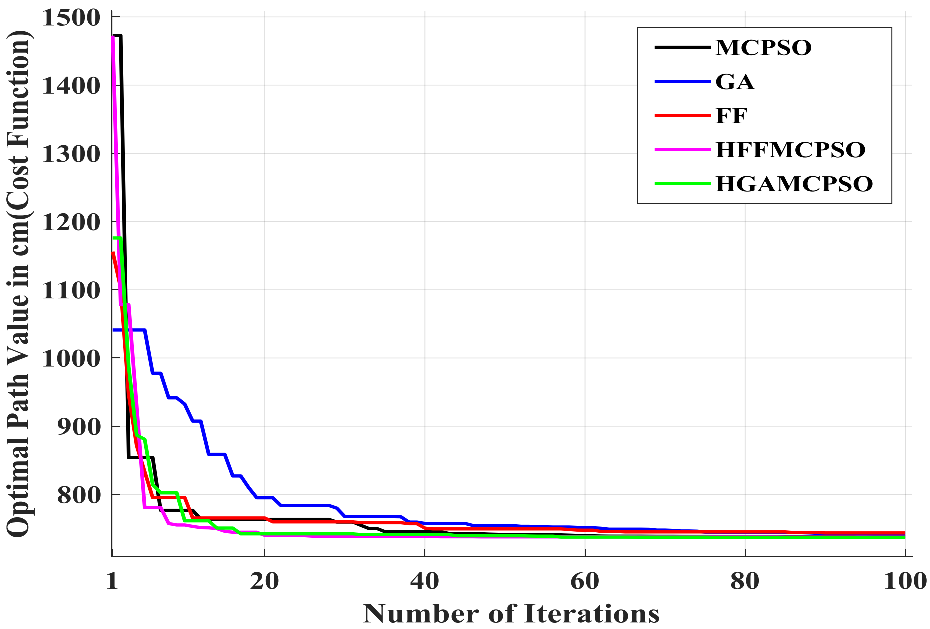
Received 30/July/2022; Accepted 23/August/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

The shortest path length was (737.399 cm and 737.094 cm) to HFFMCPSO and HGAMCPSO respectively, at iterations (50, and 48), with varying execution durations with a maximum iterations number equal to 100 iterations. Finally, the two hybrid algorithms provide a smooth path with the shortest distance and a minimum number of iterations compared to the original algorithms. On the other hand, the comparison results showed that the HGAMCPSO provides an enhancement on the path length of 0.14% compared to the MCPSO algorithm and 0.67% compared to the GA algorithm. The best path and cost function for hybrid and original algorithms obtained are shown in *Fig. 7* -a and -b and Table I shows the best path values.



(a)



(b)

FIG. 7. THE BEST PATH AND COST FUNCTION OBTAINED.

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

TABLE I. COMPARISON RESULTS FOR EXPERIMENT 1

Algorithm Types	Best Path Length	Best No. of Iterations
GA	742.131	72
FA	743.555	65
MCPSO	738.992	60
HFAMCPSO	737.399	50
HGAMCPSO	737.094	48

The reference route equation is obtained in Eq. (24), and (25) based on the fitting function as the best path that depends on the suggested hybrid algorithms: HFFMCPSO and HGAMCPSO respectively.

$$y_r(x_r) = -4.8791 \times 10^{-13} x_r^6 + 5.6773 \times 10^{-10} x_r^5 - 2.4954 \times 10^{-7} x_r^4 + 4.886 \times 10^{-5} x_r^3 - 0.0053179 x_r^2 + 0.14714 x_r + 753.15 \quad (24)$$

$$y_r(x_r) = -9.0952 \times 10^{-13} x_r^6 + 8.3904 \times 10^{-10} x_r^5 - 1.6277 \times 10^{-7} x_r^4 - 5.4952 \times 10^{-5} x_r^3 + 0.022309 x_r^2 - 2.7587 x_r + 861.64 \quad (25)$$

These equations will be employed in Eq. (26) to Eq. (31) based on the reference path equation to get the right (V_R), left (V_L) wheel linear velocities, and right (W_R), left (W_L) wheel angular velocities [41]

$$v_r(t) = \sqrt{(\dot{x}_r(t))^2 + (\dot{y}_r(t))^2} \quad (26)$$

$$w_r(t) = \frac{\ddot{y}_r(t) \times \dot{x}_r(t) - \ddot{x}_r(t) \times \dot{y}_r(t)}{x_r{}^2 + y_r{}^2} \quad (27)$$

$$V_R(t) = 0.5 \times (2v_r(t) + D \times w_r(t)) \quad (28)$$

$$V_L(t) = 0.5 \times (2v_r(t) - D \times w_r(t)) \quad (29)$$

$$W_L(t) = 0.5 \times (2v_r(t) - D \times w_r(t))/r \quad (30)$$

$$W_R(t) = 0.5 \times (2v_r(t) + D \times w_r(t))/r \quad (31)$$

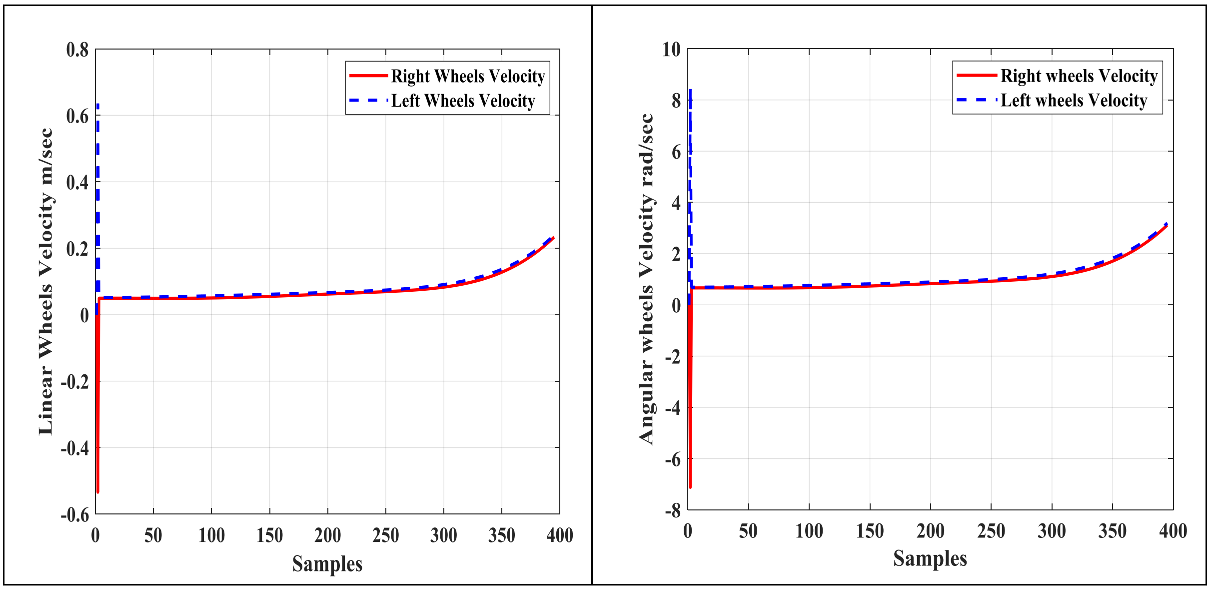
Where, v_r is the reference linear velocity and w_r is the reference angular velocity of the platform mobile robot, while r is denoted by the radius of the wheel of the mobile robot platform. Based on equations (28) to (31) *Fig. 8*-a shows the right and left wheel linear velocity, whereas the right and left wheel angular velocity are shown in *Fig. 8* -b, to one of the hybrid algorithm reference path equations.

Where the parameters of the mobile robot platform are used: r = 0.075 m and D= 0.39 m with a sampling time equal to 0.2 sec.

Received 30/July/2022; Accepted 23/August/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6



(a) (b)

FIG. 8. EXPERIMENT 1: A) THE RIGHT AND LEFT WHEELS LINEAR VELOCITIES, B) THE RIGHT AND LEFT WHEELS ANGULAR VELOCITIES.

Experiment 2: Dynamic-Obstacle Environment

In this experiment, the suggested path planning from experiment 1 (static-obstacles) has been used, based on the reference route equation Eq. (24) under a dynamic workspace, which consists of one dynamic obstacle at position (340, 550), to demonstrate the efficacy of the suggested collision avoidance method. In this workspace, the obstacle moves linearly according to Eq. (7) and (8) with V_{obs}= 0.5 (m/sec) and the direction θ_{obs}= 1.5708 (rad). *Fig. 9* - a and -b shows the moving obstacle-avoidance simulation results with the two proposed method for avoiding collision.

In the first figure, the mobile robot avoids the moving obstacle by accelerating or decelerating its velocity. After the mobile robot passes by the obstacle, it returns to its original speed and continues moving along the path until it reaches its destination point.

While in the second figure, the mobile robot selects another road to avoid collision and then returns to the original path where the obstacle passes.

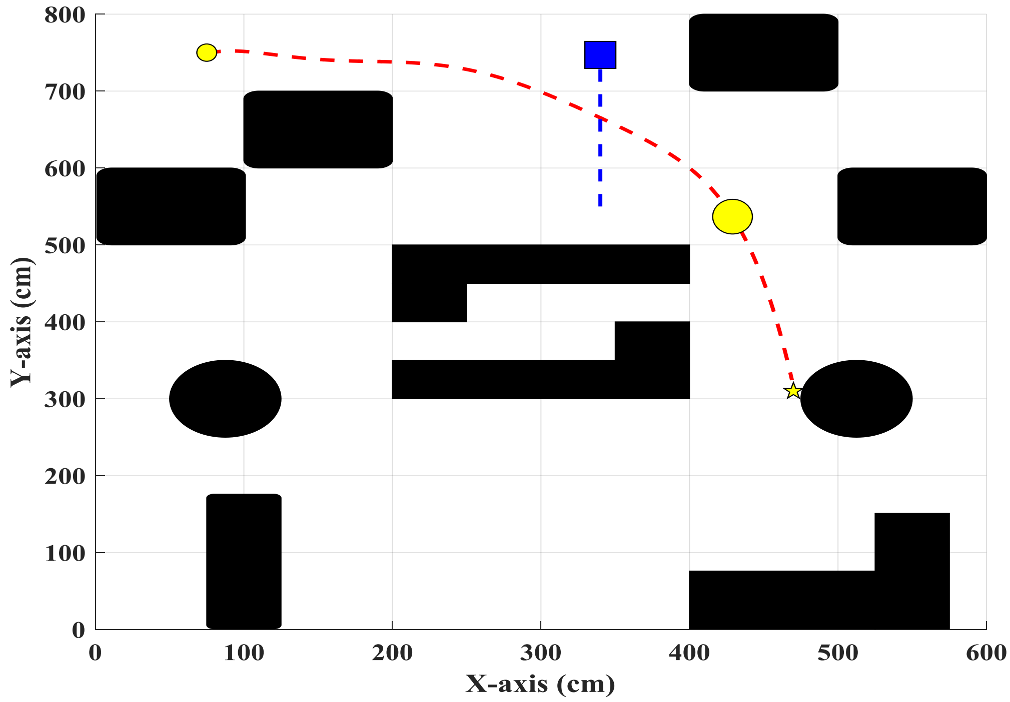
On the other hand, *Fig. 10* -a shows the right and left wheel linear velocity, whereas the right and left wheel angular velocity are shown in *Fig. 10* -b. Where the figure shows how the mobile robot increases its speed to avoid a collision in the expected period of the collision (the danger zone), according to the equation safety distance Eq. (5)– (6).

In the inverse case, *Fig. 11* -a and -b shows how the mobile robot decreases its velocity to avoid the collision. While *Fig. 12* -a and -b show the wheel velocity of the second proposed method to avoid obstacle collision by replanning the path.

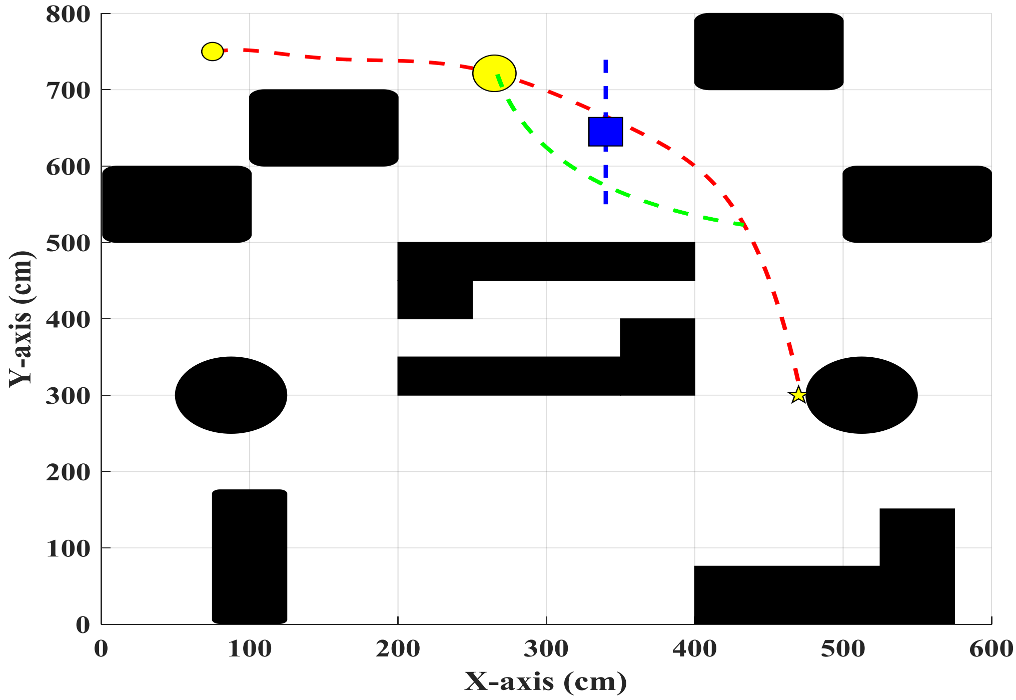
Received 30/July/2022; Accepted 23/August/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6



(a)



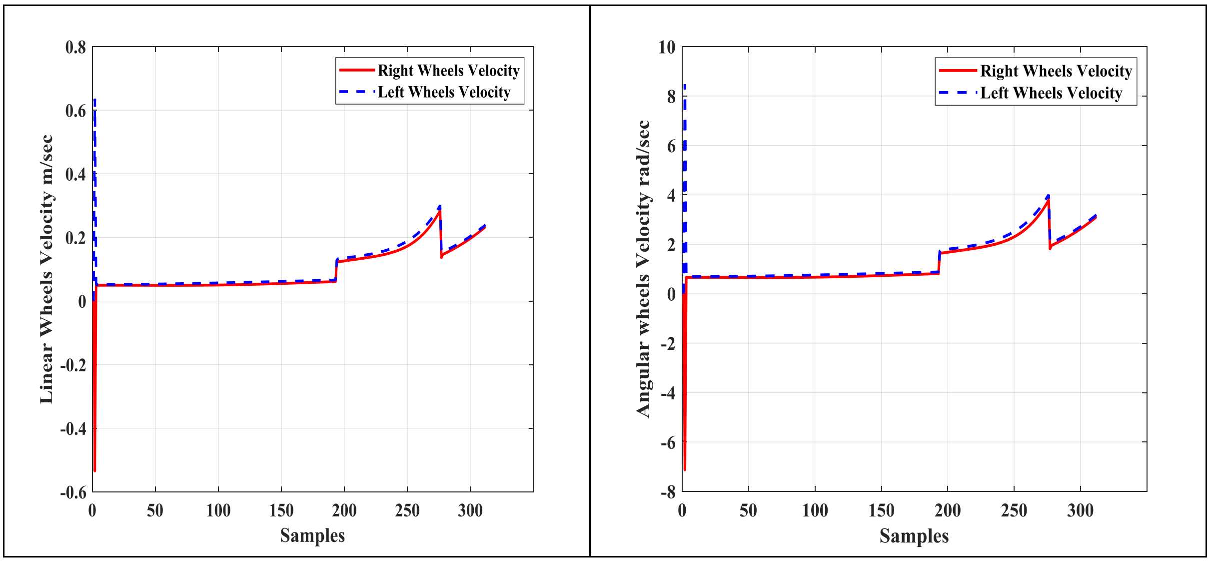
(b)

FIG. 9. PATH PLANNING IN DYNAMIC ENVIRONMENT A) INCREASE OR DECREASE THE ROBOT VELOCITY. B) SELECT ANOTHER PATH TO AVOID OBSTACLE.

Received 30/July/2022; Accepted 23/August/2022

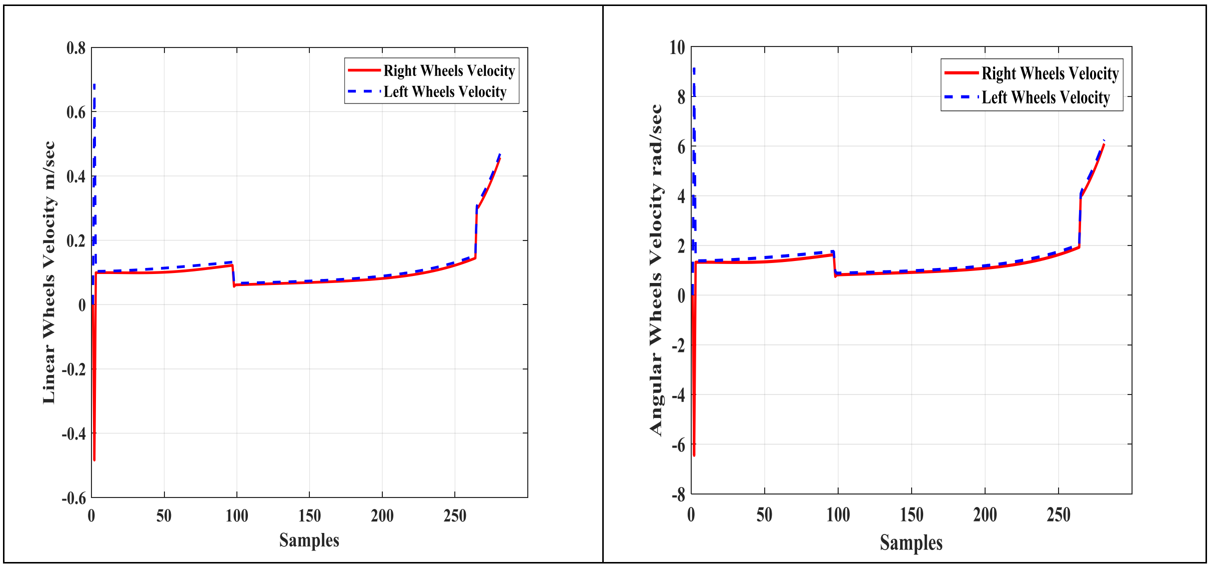
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6



(a) (b)

FIG. 10. EXPERIMENT 2: INCREASE THE VELOCITY OF MOBILE ROBOT A) THE RIGHT AND LEFT WHEELS LINEAR VELOCITIES, B) THE RIGHT AND LEFT WHEELS ANGULAR VELOCITIES.

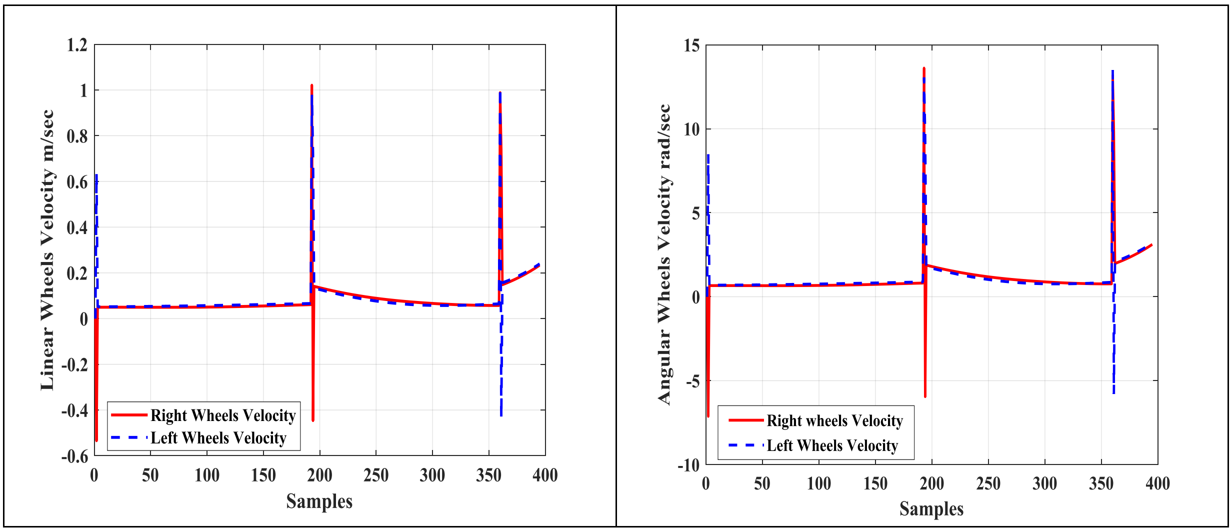


(a) (b)

FIG. 11. EXPERIMENT 2 : DECREASE THE VELOCITY OF MOBILE ROBOT A) THE RIGHT AND LEFT WHEELS LINEAR VELOCITIES, B) THE RIGHT AND LEFT WHEELS ANGULAR VELOCITIES.

Received 30/July/2022; Accepted 23/August/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6



(a) (b)

FIG. 12. EXPERIMENT 2: REPLAN THE PATH OF MOBILE ROBORT A) THE RIGHT AND LEFT WHEELS LINEAR VELOCITIES, B) THE RIGHT AND LEFT WHEELS ANGULAR VELOCITIES.

V. CONCLUSIONS

The mobile robot path planning algorithm is an important aspect of the robotics field that focuses on finding the short and smooth collision-free path in the global environment that has been proposed in this paper. Different types of meta-heuristic algorithms are proposed for solving the path planning problem for the mobile robot, and they are used to find the best path in the environment with dynamic and static obstacles, namely, FA, GA, MCPSO, and hybrid algorithms. According to the MATLAB simulation results, the proposed hybrid algorithms can determine the most ideal path for the mobile robot with the minimum number of iterations under the same conditions of obstacles in the environment as the original methods. Where, the comparison results of the first hybrid algorithm (FAMCPSO) provide enhancement on the path length of 0.82%, up to 0.21% compared to the FA and MCPSO algorithms, respectively. Moreover, the comparison results of the second hybrid algorithm (GAMCPSO) provide enhancement on the path length equals 0.67% and 0.14% compared to the GA and MCPSO algorithms, respectively. Furthermore, a proposed algorithm planner for improving the efficiency of the route planning algorithm with dynamic obstacle avoidance by adjusting the velocity or replanning the path for the mobile robot has been proposed. Simulation results have demonstrated the effectiveness and efficiency of the proposed approach by showing its ability to produce smooth and small values of the angular and linear velocities of the left and right wheels without abrupt spikes leads to small amounts of power being wanted by the mobile robot to move on its path.

In future work, the proposed strategy of obstacle avoidance can be carried out in experimental work by first using an ultrasonic sensor to detect the safe distance between the mobile robot and all moving obstacles and, secondly, using low control on the motor of the mobile robot to control the velocity of the mobile robot wheels.

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

REFERENCES

[1] J. L´opeza , P. Sanchez-Vilari˜no , M. D. Cacho , E. L. Guill´en, ”Obstacle avoidance in dynamic environments based on velocity space optimization,” Robotics and Autonomous Systems, vol.131, no.1, pp.1-60, 2020.

[2] A. Vemula, K. Muelling, J. Oh., "Path planning in dynamic environments with adaptive dimensionality," Ninth Annual Symposium on Combinatorial Search, 2016.

[3] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid," Proceedings in 2007 IEEE International Conference on Robotics and Automation, IEEE, 2007.

[4] P. Sudhakara and V. Ganapathy, "Trajectory Planning of a Mobile Robot Using Enhanced A-Star Algorithm," Indian Journal of Science and Technology, vol. 9, no. 41, pp. 478- 482, 2016.

[5] C. Saranya, M. Unnikrishnan, S. A. Ali, D. S. Sheela, and V. R. Lalithambika, "Terrain Based D∗ Algorithm for Path Planning," IFAC-PapersOnLine, vol. 49, no. 1, pp. 178–182, 2016.

[6] B. K. Patle, A. Jha, A. Pandey, N. Gudadhe and S.K. Kashyap,"The Optimized Path for A Mobile Robot Using Fuzzy Decision Function," in 2019 9th International Conference of Materials Processing and Characterization, vol. 18, part 7, pp. 3575-3581, 2019.

[7] M. Li, C. Wang, Z. Chen, X. Lu, M. Wu, and P. Hou, "Path Planning of Mobile Robot Based on Genetic Algorithm and Gene Rearrangement," Chinese Automation Congress (CAC), pp. 6999-7004, 2017.

[8] P. I. Adamu, H. I. Okagbue, and P. E. Oguntunde, "Fast and Optimal Path Planning Algorithm (FAOPPA) for A Mobile Robot," Wireless Personal Communications, vol. 106, no. 2, pp. 577-592, 2019.

[9] G. Chen and J. Liu, "Mobile Robot Path Planning Using Ant Colony Algorithm and Improved Potential Field Method," Computational Intelligence and Neuroscience, vol. 2019, pp. 1-10, May 2019.

[10] M. S. Tsuzuki, T. C. Martins, and F. K. Takase, "Robot path planning using simulated annealing," IFAC Proceedings, vol. 39.3, no. 3, pp. 175-180, 2006.

[11] Iswanto, I., et al. "Artificial potential field algorithm implementation for quadrotor path planning." Int. J. Adv. Comput. Sci. Appl 10.8 (2019): 575-585.

[12] R. M. C. Santiago, A. L. De Ocampo, A. T. Ubando, A. A. Bandala and E. P. Dadios, "Path planning for mobile robots using genetic algorithm and probabilistic roadmap," Proceedings in 9th international conference on humanoid, nanotechnology, information technology, communication and control, environment and management (HNICEM). IEEE, pp. 1-5, 2017.

[13] R. Wang, M. Wang, Y. Guan, and X. Li, "Modeling and Analysis of the Obstacle-Avoidance Strategies for a Mobile Robot in a Dynamic Environment," Mathematical Problems in Engineering, vol. 2015, pp. 1-11, 2015.

[14] Q. Zhang, Sh. Yue, Q. Yin, and Y. Zha, “Dynamic Obstacle-Avoiding Path Planning for Robots Based on Modified Potential Field Method,” Proceedings in 9th international conference on Intelligent Computing Theories and Technology, vol. 7996, pp. 332–342, 2013.

[15] K. Wei and B. Ren, “A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm,” Sensors, vol. 18, no. 2, pp. 1-15, 2018.

[16] H. Miao and Y. Tian, "Robot path planning in dynamic environments using a simulated annealing-based approach," Proceedings in 10th International Conference on Control, Automation, Robotics and Vision, IEEE, pp. 1253-1258, 2008.

[17] D. Hana, H. Nie , J. Chena , M. Chen, “Dynamic obstacle avoidance for manipulators using distance calculation and discrete detection,” Robotics and Computer-Integrated Manufacturing, vol. 49, pp. 98-104, 2018.

[18] F. H. Ajeil, I. K. Ibraheem, A. T. Azar and A. J Humaidi, “Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 3, pp. 1-15, 2020.

[19] A. S. Al-Araji, “A Cognitive PID Neural Controller Design for Mobile Robot Based on Slice Genetic Algorithm”, Engineering & Technology Journal, vol. 33, no. 1, pp. 208-222, 2015.

[20] A. S. Al-Araji, M. F. Abbod, H. S. Al-Raweshidy, “Neural autopilot predictive controller for nonholonomic wheeled mobile robot based on a pre-assigned posture identifier in the presence of disturbances”, The 2nd International Conference on Control, Instrumentation and Automation (ICCIA), pp. 326-331, 2011.

[21] J. Li, "Research on Robot Motion Control Based on Variable Structure Fuzzy Neural Network Based On T-S Model," Proceedings in IOP Conference Series: Earth and Environmental Science, vol. 440, no. 3, pp. 1-9, 2020.

[22] N. A. Zghair, and A. S. Al-Araji, "A One Decade Survey of Autonomous Mobile Robot Systems," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 6, pp. 4891-4906, 2021.

[23] M. EL KHAILI, “Path Planning in a Dynamic Environment,” International Journal of Advanced Computer Science and Applications(IJACSA), vol. 5, no. 8, pp. 86-92, 2014.

[24] Basavanna. M. and Shivakumar. M., "An Overview of Path Planning and Obstacle Avoidance Algorithms in Mobile Robots," International Journal of Engineering Research and Technology, vol. 8, no. 12, 2019.

[25] Y. Wang, X. Li, J. Zhang, S. Li, Z. Xu and X. Zhou, "Review of wheeled mobile robot collision avoidance under unknown environment," Science Progress, vol. 104, no. 3, 2021.

Received 30/July/2022; Accepted 23/August/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.2.6

[26] F. Kamil, SH. Tang, W. Khaksar, N. Zulkifli and AS. Ahmad, “A review on motion planning and obstacle avoidance approaches in dynamic environments,” Advances in Robotics & Automation, vol. 4, no. 2, pp. 134-142, 2015.

[27] A. Pandey, Sh. Pandey and P. DR., "Mobile robot navigation and obstacle avoidance techniques: A review," *International Robotics & Automation Journal*, vol. 2, no. 3, 2017.

[28] R. L., Williams II, and J. Wu, "Dynamic obstacle avoidance for an omnidirectional mobile robot," *Journal of Robotics*, vol. 2010, pp. 1-14, 2010.

[29] N. A. Zghair, and A. S. Al-Araji, “Intelligent Hybrid Path Planning Algorithms for Autonomous Mobile Robots”, International Journal of Intelligent Engineering and Systems, Vol. 15, No. 5, pp. 309-325, 2022.

[30] W. Yang, X. Zhou, and Y. Luo, "Simultaneously Optimizing Inertia Weight and Acceleration Coefficients Via Introducing New Functions into PSO Algorithm," *Journal of Physics*: Conference Series, vol. 1754, no. 1, pp. 012195, 2021.

[31] M. M. Jawad, and E. A. Hadi, "A Comparative study of various intelligent algorithms-based path planning for Mobile Robots," *Journal of Engineering*, vol. 25, no. 6, pp. 83-100, 2019.

[32] Q. Zhao and S. Yan, "Collision-Free Path Planning for Mobile Robots Using Chaotic Particle Swarm Optimization," in 2013 First international conference on Advances in Natural Computation, vol. 3612, pp. 632-635, 2005, doi:10.1007/11539902_77.

[33] B. Li et al., "Path Planning Based on Firefly Algorithm and Bezier Curve," Proceedings in 2014 International Conference on Information and Automation Hailar, pp. 630-633, 2014.

[34] L. Zhang1, L. Liu, X-S Yang, and Y. Dai, "A Novel Hybrid Firefly Algorithm for Global Optimization," PLoS ONE, vol. 11, no. 9, pp. 1-17, 2016.

[35] A. K. Abdul Hassan and D. J. Fadhil, "Mobile Robot Path Planning Method Using Firefly Algorithm For 3D Sphere Dynamic & Partially Known Environment," Journal of University of Babylon for Pure and Applied Sciences, vol. 26, no. 7, pp. 309-320, 2018.

[36] U. Srilakshmi, M. Bhargavi, B. Mounika, and N. Uttej Kumar, "Optimal Path Selection for Autonomous Robot Navigation Using Hybrid PSO-GA Routing Protocol," European Journal of Molecular & Clinical Medicine, vol. 7, no. 11, pp. 4210-4226, 2021.

[37] R. J. Kuo, and Y. S. Han, "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Solving Bi-Level Linear Programming Problem–A Case Study on Supply Chain Model," Applied Mathematical Modelling, vol. 35, no. 8, pp. 3905-3917, 2011.

[38] A. Pedlow, G. McCullough, A. Goguet, O. Garcia-Afonso, and D. Sellick, "The Development and Application of An Evolutionary Algorithm for The Determination of Kinetic Parameters in Automotive Aftertreatment Models," Emission Control Science and Technology, vol. 4, pp. 113-133, 2018.

[39] D. Sharma, P. Gaur, and A. P. Mittal, "Comparative Analysis of Hybrid GAPSO Optimization Technique with GA and PSO Methods for Cost Optimization of an Off-Grid Hybrid Energy System," Energy Technology & Policy, vol. 1, pp. 106-114, 2014.

[40] B. Abhishek, S. Ranjit, T. Shankar, Geofrey Eappen, P. Sivasankar, and A. Rajesh, "Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs", SN Applied Sciences, vol. 2, no. 11, pp. 1-16, 2020.

[41] O. A. Abdul Wahhab, and A. S. Al-Araji, “An Optimal Path Planning Algorithms for a Mobile Robot,” Iraqi Journal of Computers, Communications, Control & Systems Engineering *(IJCCCE)*, vol. 21, no. 2, pp. 44-58, 2021.

Received 30/July/2022; Accepted 23/August/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*