Received: February 7, 2022. Revised: February 25, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhancement of Cell Decomposition Path-Planning Algorithm for Autonomous Mobile Robot Based on an Intelligent Hybrid Optimization Method

Zainab E. Kanoon[1]* **Ahmed Sabah Al-Araji[1]** **Mohammed Najm Abdullah[1]**

[1] *Computer Engineering Department, University of Technology –Iraq, Baghdad, Iraq*
* Corresponding author's Email: 120105@uotechnology.edu.iq

Abstract: Due to the widespread use of autonomous mobile robots, the path planning of mobile robots take great importance in the research field. This paper aims at developing a path planning method that provides the shortest path with collision avoidance between the starting and the target points in a static robot environment. This paper proposes an intelligent hybrid optimization method that combines two algorithms; the first algorithm is called the Quarter Orbits (QO) algorithm, which tries to enhance the behavior of the cell decomposition path planning algorithm in terms of reducing the path length required to reach the target by steering the mobile robot toward the target using the quarter orbits cells instead of the grid cells and the vertical lines' cells. The second algorithm uses the Particle Swarm Optimization (PSO) algorithm to get a more efficient, smoother, and shorter path for the mobile robot to reach the target in a complex environment. The generated hybrid method is called the Quarter Orbits Particle Swarm Optimization (QOPSO). All simulation results clearly show that the proposed hybrid algorithm provided the shortest path with collision avoidance compared to the results of the Vertical Cell Decomposition (VCD) algorithm and the Radial Cell Decomposition (RCD) algorithm. The proposed hybrid QOPSO algorithm achieves enhancement on the path length equals 29.42% compared to the VCD algorithm and 24.25% compared to the RCD algorithm for different maps using MATLAB 2021a.

Keywords: Mobile robot, Path planning, Static environment, Quarter orbits algorithm, Particle swarm optimization.

1. Introduction

Autonomous Mobile Robots are utilized in a variety of applications such as danger seeker, target finder, exploration, security patrol, medical care, mining, space missions, and education [1]. The breakout of the coronavirus disease (COVID-19) in 2020 has accelerated the process and brought up new opportunities for autonomous robots and automation in a variety of industries [2]. Manufacturing, cleaning, logistics, and healthcare are among the domains where robots can help tackle this global disease. Robots can play an important role in daily activities by acting as a reliable logistics fleet in the event of supply chain disruptions [3]. The mobile robot navigation process includes a set of requirements such as defining the robot environment, localizing the robot position, and path planning [4]. The path planning of mobile robots means finding a feasible path between the start point and the target point taking into consideration many objectives such as avoiding obstacles, finding the shortest path, taking minimum time to reach the target, consuming less energy, and achieving smoothness of the path [5]. From the late 1960's and early 1970's, a lot of research works focus on the mobile robot path planning. In this regard, there are two types of path planning. The first type is global path planning [1, 6], which includes static obstacles, where all the information about the robot environment is predefined and the path is also well-known to the autonomous mobile robot. The second type is the local path planning [1, 7], which includes a dynamic environment with moving obstacles. In this environment, the mobile robot has unknown information or partially known information, and hence, the mobile robot needs to sense the environment before moving. The subject of path

Received: February 7, 2022. Revised: February 25, 2022.

planning for mobile robots has been widely studied by many authors who explore a set of solutions. However, there are some significant gaps and restrictions that must be addressed. In this paper, a hybrid method is used to solve two problems of mobile robot path planning. The first problem is collision avoidance while the second one is providing the smoothest and shortest path from the start point to the target point. This proposed intelligent hybrid method combines a method called the Quarter Orbits (QO) algorithm, which was derived from the cell decomposition algorithm, and it also uses the particle swarm optimization method to produce the Quarter Orbits Particle Swarm Optimization (QOPSO) method, which works to find the smoothest and the shortest path for the mobile robot. This paper is organized as follows: Section 2 represents the non-holonomic wheeled mobile robot system. Section 3 discusses the path planning algorithms' types, and Section 4 describes our hybrid methods, while Section 5 shows the simulation result. Finally, the conclusion of this paper is represented in Section 6.

2. Non-holonomic wheeled mobile robot system

Drive systems can be classified based on how the robot moves into the holonomic and non-holonomic drives [8]. The link between the controlled degree of freedom and the total degrees of freedom of a robot is referred to as holonomic drive and non-holonomic drive [9]. The degree of freedom that can be controlled is specified. The holonomic drive refers to a robot whose controllable degrees of freedom are equivalent to its total degrees of freedom. Similarly, if the robot's controlled degrees of freedom are smaller than its total degrees of freedom, then it is referred to as a non-holonomic drive. Fig. 1 demonstrates a non-holonomic wheeled mobile robot system. It includes two driving wheels positioned within the same axis, as well as one caster wheel at the front or back of the platform for the stability of the mobile robot [10]. The right and left wheels' actuators of the wheeled robot are actuated by two independent analog Direct Current (DC) motors for motion and platform steering. The mobile robot center mass is located at point (m_p), and the two drive wheels are coupled to the axis center [9]. Because the mobile robot model is a multi-input multi-output system, the number of input states is two (left and right wheels' velocities), but the output states are three based on its position in the global coordinate frame [O, X-axis, and Y-axis], and the pose surface is X_p and Y_p representing the coordinates of the point m_p. The kinematics equation of the mobile robot

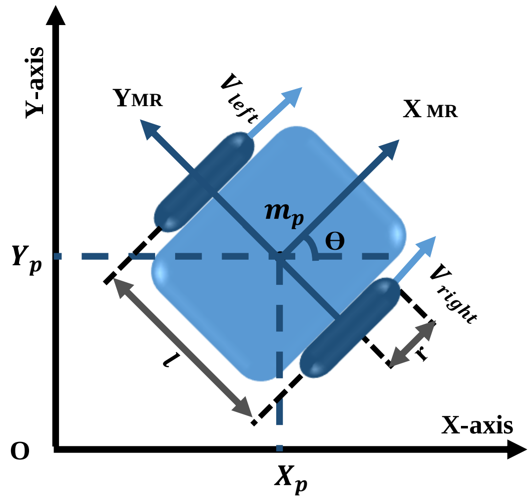


Figure. 1 A non-holonomic wheeled mobile robot

Table 1. The equations' parameters with their definitions

Parameter	Definition
V_{left}	Left wheel velocity
V_{right}	Right wheel velocity
l	The distance between the two wheels
T	The sampling time of the mathematical calculation

platform has highly nonlinear and time variant output states and it also has an under-actuated model. As a result, these three generalized coordinates can be used to define the mobile robot's configuration. As a conclusion, the computer simulation equations are as follows [11]:

$$X_p(k) = \left[\frac{1}{2}\left(V_{left} + V_{right}\right) \times \cos\left(\theta(k)\right) \times T\right] + X_p(k-1) \quad (1)$$

$$Y_p\,(k) = \left[\frac{1}{2}\left(V_{left} + V_{right}\right) \times \sin\left(\theta(k)\right) \times T\right] + Y_p(k-1) \quad (2)$$

$$\theta(k) = \left[\frac{1}{l}\left(V_{left} + V_{right}\right)T\right] + \theta(k-1) \quad (3)$$

3. Path planning algorithms types

The navigational process of the mobile robot highly depends on path planning. The path planning means that the mobile robot must reach the target taking into account the path length, the time to reach the target, obstacle avoidance, and many other criteria. Path planning methods can be classified into traditional methods, graph search methods, and heuristic methods, as shown in Fig. 2. The basic goal of these methods is to identify a solution, or a path

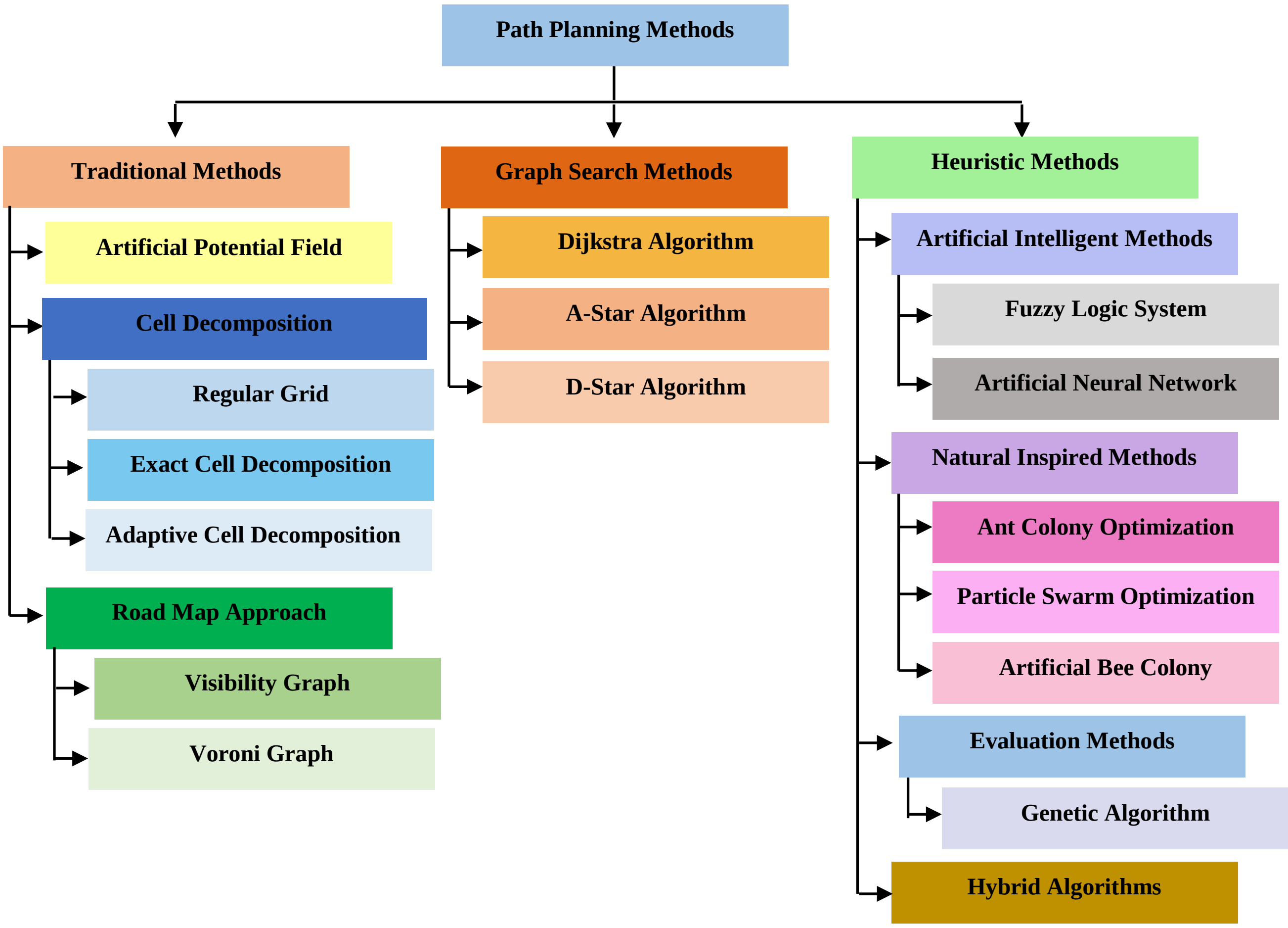


Figure. 2 Classification of the path planning algorithms

that is feasible. These methods have been extended with time by optimization strategies that focused on reducing the distance traveled by the mobile robot [12].

3.1 Traditional methods

In the past, traditional methods were popular for solving the path-planning problem for the mobile robot because artificial intelligence techniques had not been developed yet. By using the traditional methods to perform the task, it has been noted that they may or may not solve the path planning problem. There are different types of traditional methods and the most popular of which is the cell decomposition method [13]. In particular, this study compares different types of the cell decomposition methods such as the Vertical Cell Decomposition (VCD) algorithm and the Radial Cell Decomposition (RCD) algorithm. These algorithms utilize grid cells and vertical-line cells with long distances generated among the cells according to the obstacles in the environment.

The second most popular traditional method is the artificial potential field [14] and this research focuses on real-time obstacle avoidance using an artificial potential field. According to this method, the goal and obstacles behave likecharged surfaces, with the overall potential creating an imagined force on the robot. However, the drawback of this method is that the best path is determined by the attractive potential field function, which has the limitation of getting stuck at the local minima. Another method is the road map method [15], which is a method of getting from one location to another, and it is represented by a series of one-dimensional curves that connect the free spaces. The main disadvantage of traditional methods is their high processing times, especially in complex environments, since they are unable to respond well in dynamic environments. As a result, they are less common.

3.2 Graph search methods

For energy-efficient path planning, graph search algorithms have been employed extensively in previous studies. It checks some nodes/states to determine a path from the starting to the goal places. The problem with this method is that it could fail if there is no existing path [16]. The graph search method is a simple method for discovering the path for a mobile robot. It is a well-defined, very efficient method for identifying a path with less complexity. After the environment is formed, this method will try to find the optimal solution from the start node to the target node. There are different types of graph search methods. The first type is the Dijkstra algorithm [17]. In this research, the Dijkstra algorithm is utilized to identify an optimal path between the start and all other points in the graph depending on traversal costs. However, this method has taken a long execution time. The second type is the A-star algorithm [18], which was proposed by Hart et al. suggesting it as a development of the Dijkstra algorithm. The third type is the D-star algorithm [19], which was developed by Stentz in 1994. Because it is employed in dynamic environments, it might be called the dynamic A* algorithm.

3.3 Heuristic methods

These methods have been shown to be effective in solving difficult navigation problems. In particular, they are used to solve the drawbacks of traditional approaches. Heuristic methods have a strong ability to deal with the instability that exists in their environment, and they include different algorithms, including:

Artificial Intelligence Methods: such as the fuzzy logic system in [20], where Zavlangas et al. used the fuzzy logic for navigation of an omnidirectional mobile robot. Another artificial intelligence method is the artificial neural network in [21], in which Janglova demonstrated the use of a neural network for the navigation of a wheeled mobile robot in a partially unknown environment. He developed a collision-free path using two NN-based techniques.

Nature-Inspired Methods: these algorithms are inspired by nature and they include the particle swarm optimization method in [22], where the authors proposed the MAPSO algorithm, which improves particle properties, speeds up convergence, and increases particle efficiency. In another work, the use of Ant Colony Optimization (ACO) for real-time path planning of mobile robots was given in [23]. On the other hand, Artificial Bee Colony (ABC) was used in [24], where the authors presented the use of the ABC algorithm in the mobile robot navigation process in a static environment.

Evaluation Methods: such as the genetic algorithm used by Shibata et al. [25], who provided a solution to solve the mobile robot navigation problem using the genetic algorithm in a static environment. Hybrid methods: these algorithms consist of a combination of two or more algorithms mentioned previously.

In heuristic methods, the algorithms are supposed to find high-quality solutions in a short amount of time. They also have the ability to solve optimization problems.

4. Hybrid path planning algorithm design

This paper focuses on solving two issues of the path planning problem; the first issue is to avoid collision with obstacles, while the second one is finding the shortest path for a mobile robot to reach the target in a complex environment. To solve these two problems, a hybrid method was proposed, which was developed by combining a proposed Quarter Orbits algorithm with the Particle Swarm Optimization method.

4.1 Quarter orbits algorithm

This algorithm is inspired by the cell decomposition algorithm, which is classified into three types, including the regular grid, the adaptive cell decomposition, and the exact cell decomposition [13]. The regular grid technique [26] is used to discover a collision-free path for an object travelling through crowded obstacles. It can be created by placing a regular grid over the configuration space in general. Regular grid is simple to use because the shape and size of the grid cells are predetermined. It works by sampling the domain and then marking up the graph to determine if the space is occupied, unoccupied, or partially occupied. One important drawback with the regular grid is the cell size. Particularly, when the cell size is made very large the planner may not be complete. Conversely, when it is made very small, then additional computational time will be required. The second type of cell decomposition is adaptive cell decomposition [27], which is created with a quad-tree. Quad-cells trees are classified as either free cells with no obstacles, obstacles' cells with obstacles, or mixed cells with both free and obstacles spaces. The process is repeated by dividing the mixed cells into four identical sub-cells until no barriers' region remains or the smallest cells are created. The main drawback of the adaptive cell decomposition is that it cannot easily update when a new position of an obstacle is encountered. The last type of cell decomposition is the exact cell decomposition [28]. Cells in the exact cell decomposition do not have a fixed shape or size, but they can be determined by the map of the environment, the shape of obstacles inside the environment, and the locations of these obstacles. The available free space in the environment is divided into small elements (trapezoidal and triangular), each of which is assigned a number. Every element in the environment serves as

Received: February 7, 2022. Revised: February 25, 2022.

a node in a graph of connectedness. In the configuration space, nearby nodes are then allowed to join to make the path. However, this method is not suitable when the environment includes irregular obstacle shapes.

The quarter orbits algorithms stem from the cell decomposition algorithm with some differences in the path search process and the shape of the cells. In the quarter orbits algorithm, as the name indicates, the robot static environment is divided into quarter orbits cells and the navigation process is shown in Fig. 3.

This algorithm is summarized as follows:

- Initialize the robot environment with the start and the target points.
- Then, the mobile robot will steer toward the target.
- If the start point and the target point are on the same orbit, then the mobile robot will slide with that orbit until it reaches the destination, otherwise the mobile robot will start to transfer from the closest orbit to the next orbit until it reaches the target.
- When the mobile robot faces an obstacle, it will transfer to the next closest orbit with the safe space and when the mobile robot exceeds the obstacle it will steer toward the target again and start to transfer from one orbit to another until it reaches the target.

The quarter orbits algorithm tries to enhance the cell decomposition algorithm. It reduces the path length required to reach the target by steering the robot toward the target using the quarter orbits cells instead of the grid cells and the vertical lines cells. This method produces a collision-free path when there is one that exists. At the same time, using this method cannot guarantee finding the shortest distance path.

4.2 Particle swarm optimization method

Particle Swarm Optimization (PSO) is a multi-point research technology-based experimental community that simulates the social behaviour of a flock of birds or a group of fish [29, 30]. The PSO tries to emulate the behaviour of a social animal, it does not require a group leader to fulfil the goals. When the flock of birds travels in search of food, they don't need any leaders, but they simply follow one of the individuals that is closest to the food. In this way, the flock of birds achieves its desired goal through effective communication with the rest of the particles. With PSO, the particle of the swarm acts as a potential optimal solution to a certain optimization problem, which specifies the dimension of the particle's velocity and position vectors. An ideal solution can be found by exploring the solution space of the optimization problem.

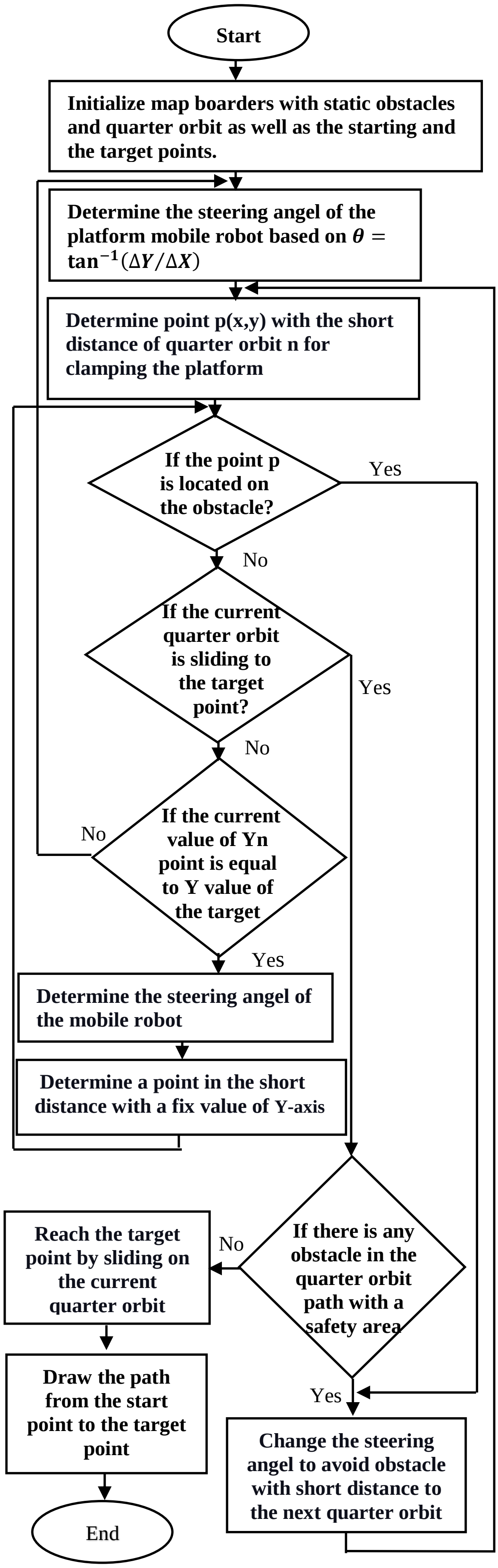


Figure. 3 Flowchart of the quarter orbits algorithm

Table 2 Parameters of PSO with the definition of each parameter

Parameter	Definition
$V_i(j)$	i^{th} particle's velocity in iteration j
$X_i(j)$	i^{th} position vectors in iteration j
$P_{ibest}(j)$	The best fitness values for the i^{th} particle
$G_{best}(j)$	The best global fitness value for the whole swarm
w	Inertia weight of the velocity 0.751
c_1 and c_2	The acceleration coefficients (1.75, 1.75)
r_1 and r_2	Random numbers with a uniform distribution of the range [0, 1].

Step 1: Determine the maximum number of iterations.
Step 2: Initialize each particle.
Step 3: For each particle**,** check the fitness value, if it is greater than the best fitness value (P_{ibest}), then set the current value as (new P_{ibest})
Step 4: For each particle
- Find the particle in the particle neighbourhood with the best global fitness (G_{best}).
- Calculate particle velocity $V_i(j)$, according to Equation (4).
- Apply the new value of velocity.
- Calculate the new particle position $X_i(j)$, according to Equation (5).
- Apply the new value of position.

Step 5: Repeat from **Step 3** until reaching the maximum number of iterations.

Figure. 4 The pseudocode of the PSO algorithm

Fig. 4 represents the pseudocode of the PSO algorithm, and the update functions of the velocity and position vectors at the j^{th} iteration [31] can be represented as follows:

$$V_i(j+1) = wV_i(j) + c_1 r_1\big(P_{ibest}(j) - X_i(j)\big) + c_2 r_2\big(G_{best}(j) - X_i(j)\big) \quad (4)$$

$$X_i(j+1) = X_i(j) + V_i(j+1) \quad (5)$$

c_1, and c_2 must be selected to satisfy the condition in Eq. 6 [32].

$$(c_1 + c_2) < 4 \quad (6)$$

While the PSO algorithm can solve the path planning problem effectively and produce a smooth path, in many optimization problems, it can quickly fall into local optima. Furthermore, there is no generic convergence concept applicable for PSO in practice, and its convergence interval for multidimensional problems is mainly unclear [13]. Moreover, this algorithm cannot guarantee providing the optimum solution in a complex environment.

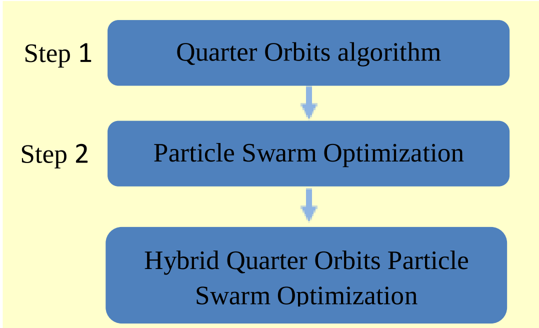


Figure. 5 The proposed path planning methodology

4.3 Hybrid quarter orbits particle swarm optimization (QOPSO) algorithm

The quarter orbits algorithm can find a collision-free path but there are no guarantees to find the optimum path because this algorithm transfers the mobile robot from one orbit to another, which will consume more power, provide an unsmooth path as well as produce an additional distance in the path. On the other hand, the particle swarm optimization algorithm is easy to fall into local optimum producing an ineffective solution in the narrow paths. This paper combines these two methods to generate a hybrid algorithm called QOPSO, as shown in Fig. 5.

This proposed methodology takes the advantages of both the quarter orbits algorithm and the PSO algorithm to generate the shortest path with collision avoidance. The quarter orbits algorithm will guarantee finding a path from the start point to the target point by steering the mobile robot toward the target and moving it from orbits to orbits with obstacle avoidance until it reaches the destination. Then the boundaries of this path will be sent to the PSO algorithm to generate the smoothest and the shortest path.

5. Simulation results

To test the proposed strategy's effectiveness, a fixed-obstacles environment was used with a workspace of [700 by 700] cm, as shown in Fig. 6.

The environment is filled with static obstacles and complete information about the positions of the objects in the workspace is available. The MATLAB 2021a package with computer hardware specifications of Intel Core i5-1035G7 with 8.00 GB of RAM, and CPU of 1.20GHz were used. To find a collision-free path, three algorithms will be used (the proposed Quarter Orbits algorithm, the Particle Swarm Optimization algorithm, and the proposed Hybrid QOPSO algorithm) and the results will be compared to find the shortest distance path, taking into consideration that between the robot and the obstacles, a safe distance must be maintained.

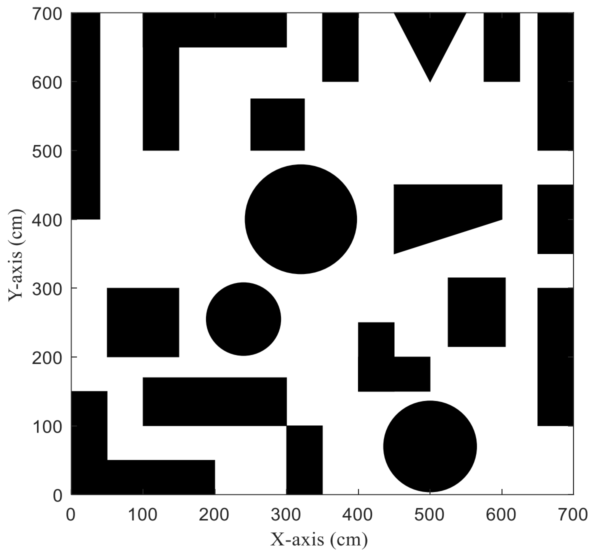


Figure. 6 The suggested environment with static obstacles

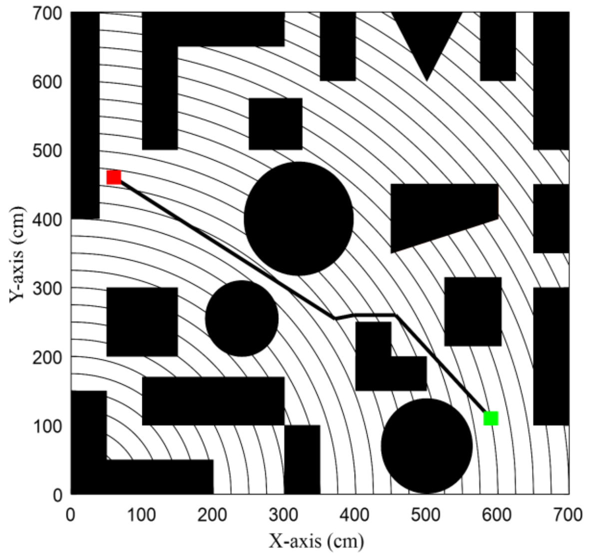


Figure. 7 Simulation result of the quarter orbits algorithm

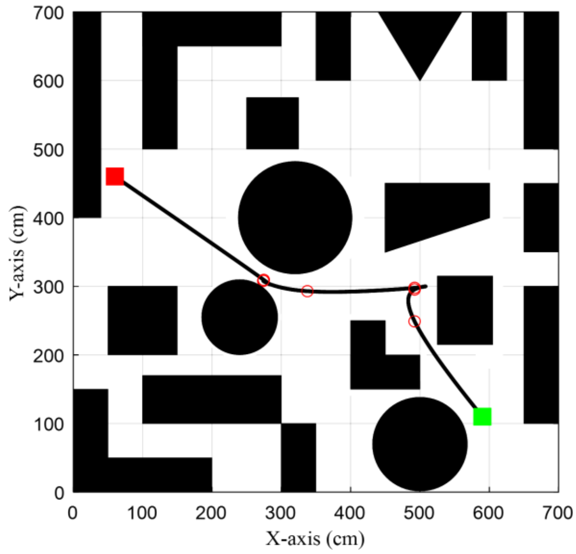
5.1 Case study:

The initial position of the mobile robot is (60, 460) cm and the destination point is at (590, 110) cm. When applying the Quarter Orbits algorithm, the shortest distance path was equal to 659.420cm, as shown in Fig. 7.

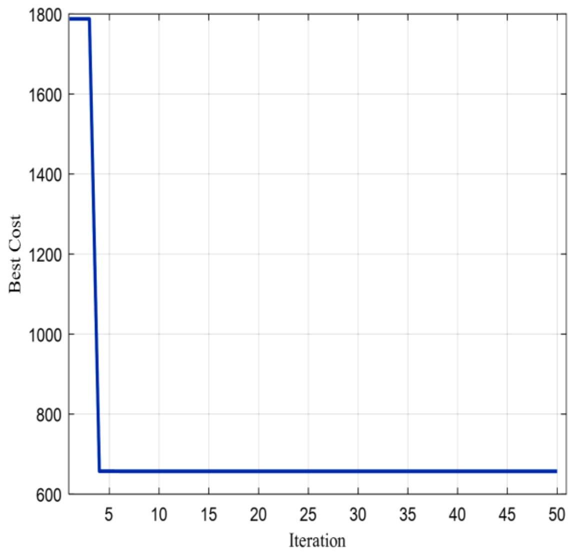
The PSO algorithm was also used to find the shortest distance path by using the suggested environment, as shown in Fig. 8(a). The best cost solution was found at iteration number 20 as shown in Fig. 8(b) with a maximum iteration number equals to 50 iterations. The value of the PSO distance function is equal to 757.1048cm. Finally, the proposed hybrid algorithm was applied to find the shortest distance path using the suggested environment as shown in Fig. 9(a). The best cost solution was found at iteration number 12 as shown in Fig. 9(b) with a maximum iteration number equals to 50 iterations. The value of the proposed hybrid algorithm distance function is equal to 657.1271 cm. When compared to the quarter orbits and PSO algorithms, the path generated by the proposed hybrid algorithm was smooth and it was the shortest path from the starting point to the goal point as shown in Table 3.

Table 3. Comparison of the shortest paths

Algorithm	Path Length	Iteration
The proposed QO	659.420 cm	-
The PSO	757.1048 cm	20
The proposed hybrid QOPSO	657.1271 cm	12



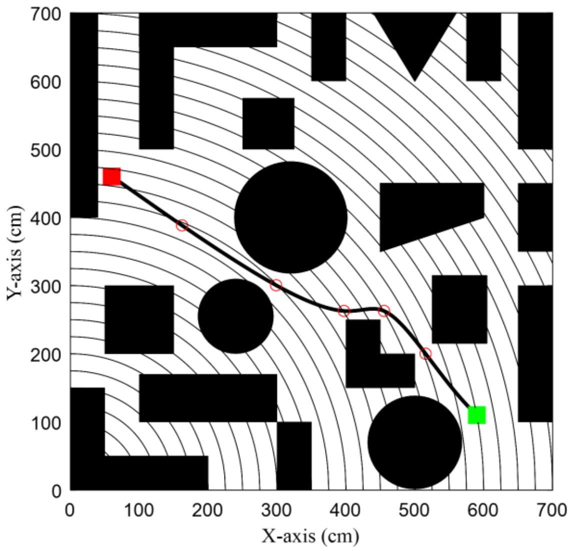
(a)



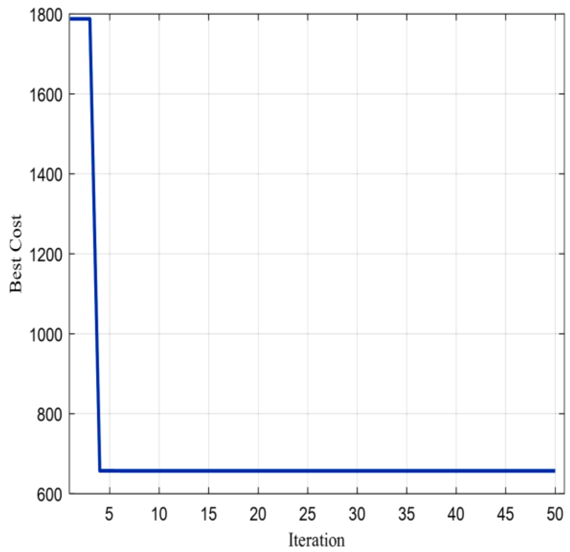
(b)

Figure. 8 The PSO algorithm: (a) path planning and (b) the best cost function

Received: February 7, 2022. Revised: February 25, 2022.



(a)



(b)

Figure. 9 The proposed hybrid algorithm (QOPSO): (a) path planning and (b) the best cost function

The reference path equation for the optimal path of the hybrid QOPSO is represented in Eq. (7):

$$y_{ref}\left(x_{ref}\right) = 7.371e - 13 \times x_{ref}^{6} - 1.448e - 9 \times x_{ref}^{5} + 1.087e - 6 \times x_{ref}^{4} - 0.0003921 \times x_{ref}^{3} + 0.07047 \times x_{ref}^{2} - 6.52x_{ref} + 676 \quad (7)$$

To find the reference linear velocity v_{ref} and the reference angular velocity w_{ref} of the platform of the mobile robot depending on the previous reference path equation, Eqs. (7) and (8) will be used [33].

$$v_{ref} = \sqrt[2]{\left(\dot{x}_{ref}\right)^{2} + \left(\dot{y}_{ref}\right)^{2}} \quad (8)$$

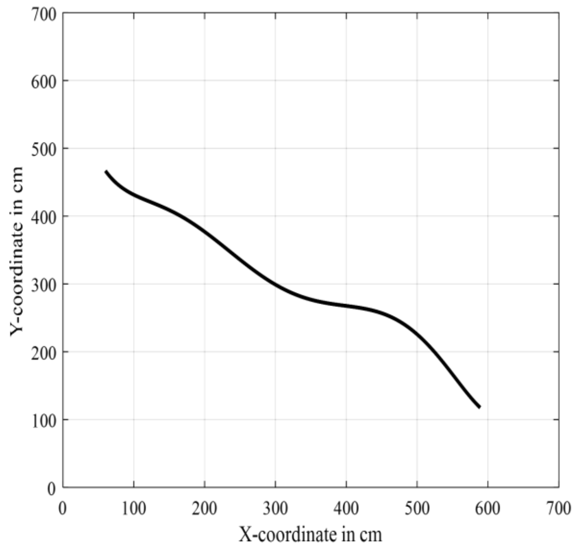


Figure. 10 The optimal path of the proposed hybrid QOPSO algorithm

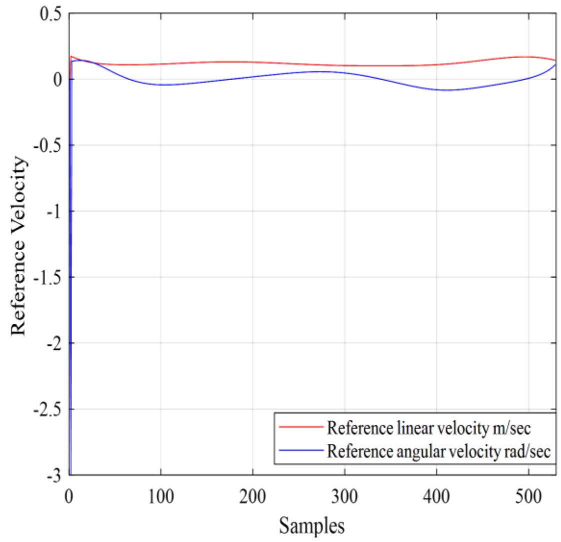


Figure. 11 The referance linear and angular velocities

$$w_{ref} = \frac{\left(\ddot{y}_{ref} \times \dot{x}_{ref} - \ddot{x}_{ref} \times \dot{y}_{ref}\right)}{\left(\left(\dot{x}_{ref}\right)^{2} + \left(\dot{y}_{ref}\right)^{2}\right)} \quad (9)$$

As a result, the linear velocities of the right and left wheels $V_R$and V_L, and the angular velocities of the right and left wheels W_R and W_L can be calculated as follows [34]:

$$V_R = \frac{\left(2v_{ref} + Lw_{ref}\right)}{2} \quad (10)$$

$$V_L = \frac{\left(2v_{ref} - Lw_{ref}\right)}{2} \quad (11)$$

$$W_R = \frac{\left(2v_{ref} + Lw_{ref}\right)}{2r} \quad (12)$$

$$W_L = \frac{\left(2v_{ref} - Lw_{ref}\right)}{2r} \quad (13)$$

 DOI: 10.22266/ijies2022.0630.14

Received: February 7, 2022. Revised: February 25, 2022.

where r represents the radius of the wheel of the mobile robot which is equal to 0.075m and L represents the distance between the two wheels, which is equal to 0.39m with a sampling time equals to 0.1sec. Fig. 10 shows the optimal path of the proposed hybrid QOPSO algorithm, and Fig. 11 represents the reference linear and angular velocities of the mobile robot.

In addition, Fig. 12(a) represents the linear velocity of the right and the left wheels of the mobile robot, while Fig. 12(b) represents the angular velocity of the right and the left wheels of the mobile robot.

To prove that the proposed hybrid algorithm, namely the QOPSO, provides the shortest path, a comparison study has been conducted with other researches who use different path planning algorithms with a static environment. Firstly, the proposed QOPSO was compared with the Vertical Cell Decomposition (VCD) algorithm, and the Radial Cell Decomposition (RCD) algorithm that were suggested in [35], using a static mobile robot environment with a workspace of [11 by 8] m, where the starting point was at (0, 0) and the goal point was at (10, 7.5). This research considers the mobile robot as a point in the 2D workspace. The results of the simulation process with the first environment using the Quarter Orbits algorithm is shown in Fig. 13, which produces a path of length equals to 12.6187 m. Moreover, when using the same environment with the proposed Hybrid QOPSO algorithm, as shown in Fig. 14, the simulation process produces a path length equals to 12.563 m. Therefore, the simulation results indicate that the Quarter Orbits (QO) algorithm provides a shorter path compared to both VCD and RCD.

While the results of the simulation process with the second environment using the Quarter Orbits algorithm is shown in Fig. 15, producing a path of length equals to 14.7627 m.

In addition, when using the same environment with the proposed hybrid QOPSO algorithm, as shown in Fig. 16, the simulation process produces a path length equals to 14.1367 m. The simulation results in this case also indicate that the Quarter Orbits (QO) algorithm provides a shorter path compared to both VCD and RCD, and the hybrid algorithm of the QOPSO can successfully generate the shortest path compared to VCD, RCD, and QO algorithms, as shown in Table 4. Because the hybrid algorithm can generate a short path between each quarter orbit cell with smooth steering, it can obtain a smooth and a

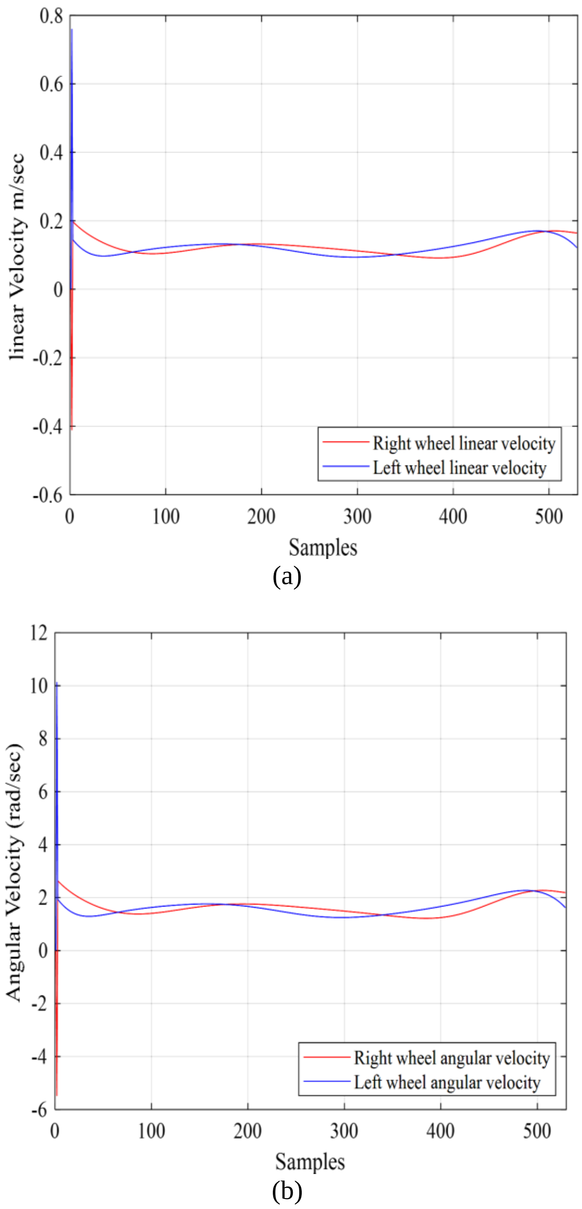


Figure. 12: (a) The linear velocities of the right and the left wheels and (b) The angular velocities of the right and the left wheels

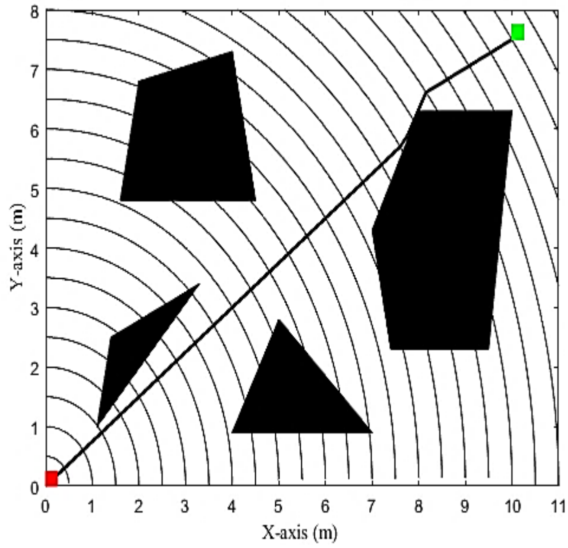


Figure. 13 Results of the shortest-path simulation using the Quarter Orbits algorithm

Received: February 7, 2022. Revised: February 25, 2022.

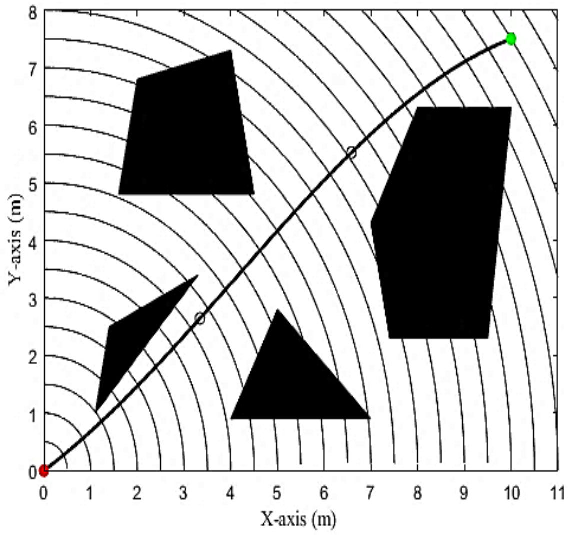


Figure. 14 Results of the shortest-path simulation using the hybrid QOPSO algorithm

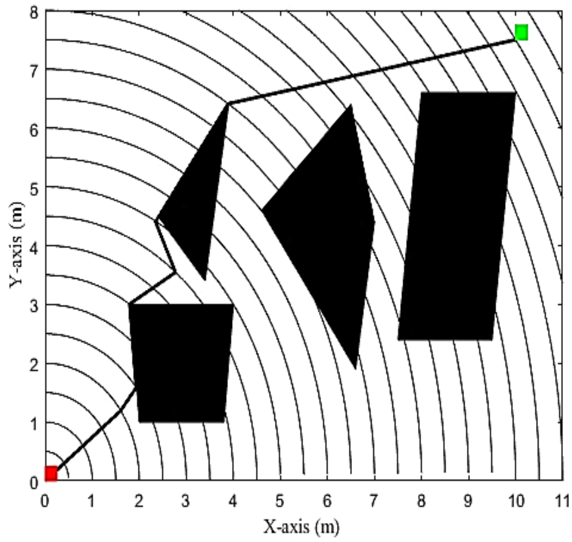


Figure. 15 Results of the shortest-path simulation using the quarter orbits algorithm

short path. On the other hand, the VCD and RCD algorithms use grid cells and vertical-line cells with a long distance between each cell based on the obstacles in the environment.

Secondly, the proposed QOPSO was compared with the Self-Adaptive Ant Colony Optimization (SAACO) and the fuzzy Ant Colony Optimization (FACO), as proposed in [36, 37], using a complex environment map with a workspace of [20 by 20] m. The results of the simulation process using the quarter orbits algorithm are shown in Fig. 17. It produces a path length equals to 30.0413 m.

When the PSO algorithm was applied on the same environment, as shown in Fig. 18(a), it produces a path length equals to 44.9569 m with an iteration number equals to 58, as shown in Fig. 18(b) with a maximum iteration number equals to 100 iterations.

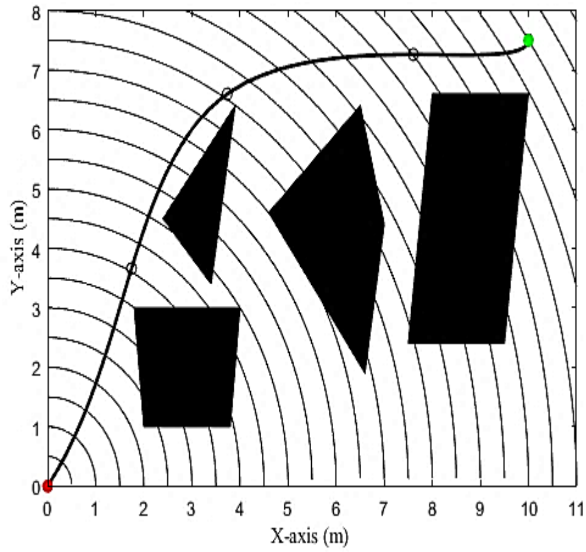


Figure. 16 Results of the shortest-path simulation using the hybrid QOPSO algorithm

Table 4. Comparison of the shortest path with the literature [35]

Environment no.	Algorithm	Path length
Environment 1 [35]	VCD [35]	17.88 m
	RCD [35]	16.66 m
	The proposed QO	12.618 m
	The Proposed Hybrid QOPSO	12.563 m
Environment 2 [35]	VCD [35]	19.55 m
	RCD [35]	15.88 m
	The proposed QO	14.762 m
	The Proposed hybrid QOPSO	14.136 m

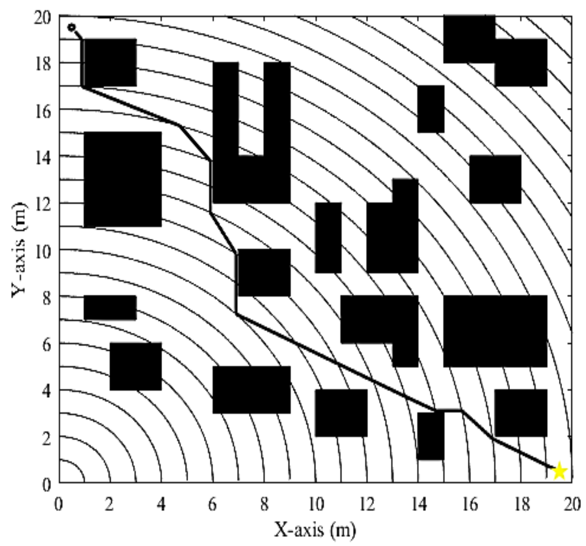
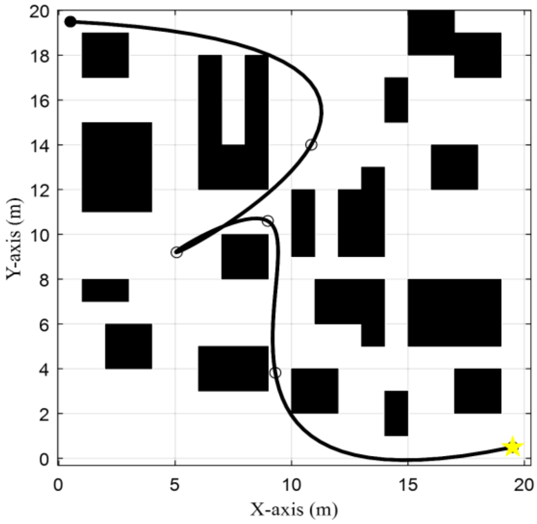
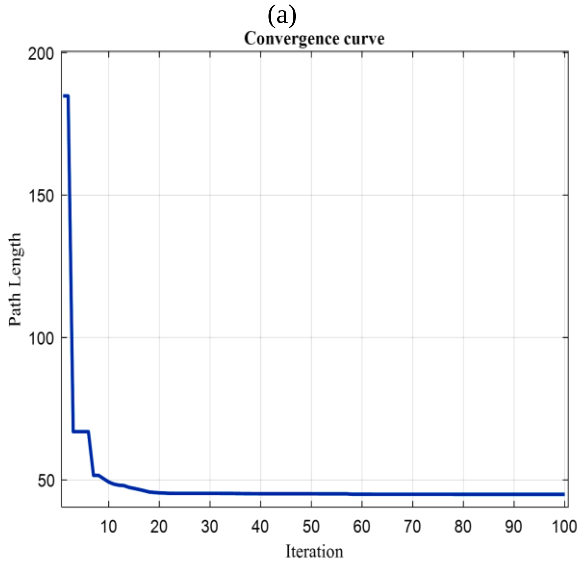


Figure. 17 Results of the shortest-path simulation using the quarter orbits algorithm

Received: February 7, 2022.   Revised: February 25, 2022.



(a)

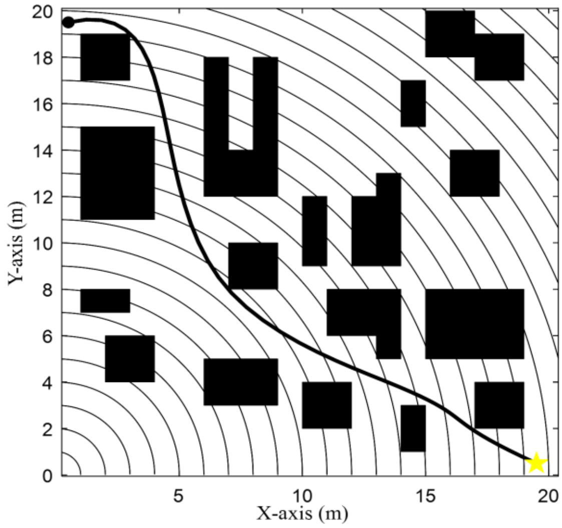


(b)

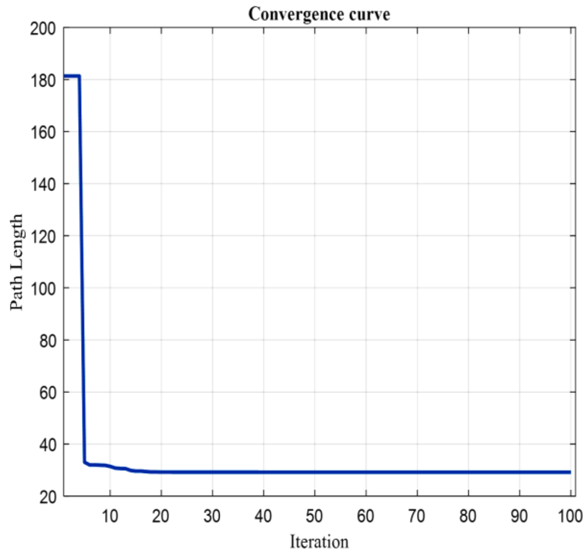
Figure. 18 (a) Results of the shortest-path simulation using the PSO algorithm and (b) the PSO algorithm convergence curve

Moreover, the simulation process using the hybrid QOPSO algorithm shown in Fig. 19(a) produces a path length equals to 29.2049 m with iteration number equals to 20, as shown in Fig. 19(b) with a maximum iteration number equals to 100 iterations.

To confirm the effectiveness of the proposed hybrid algorithm, the results of the comparison process are shown in Table 5, which shows that the hybrid algorithm can generate a short distance between each quarter orbit cell with a smooth steering function, leading to obtaining the smooth and short path. On the other hand, the SAACO algorithm utilizes distances of grid cells and generates a saw tooth path between each cell to avoid obstacles in complex environments. In contrast, the FACO algorithm depends on the distance of the sequence node for the best path generated in the grid map to avoid obstacles in the complex environment.



(a)



(b)

Figure. 19 (a) Results of the shortest-path simulation using the hybrid QOPSO Algorithm and (b) the hybrid QOPSO algorithm convergence curve

Table 5. Comparison of the shortest path with the literature [36, 37]

Algorithm	Path Length	Convergence iteration times
SAACO [36]	29.796 m	25
FACO [37]	29.3848 m	23
The proposed QO	30.0413 m	-
PSO	44.9569 m	58
The proposed hybrid QOPSO Algorithm	29.2049 m	20

Finally, the QOPSO was compared with Fast Marching Method (FMM), Fast Marching Method Hybridized with Regression Search (FMMHRS) methodology, and Artificial Potential Field method

Received: February 7, 2022. Revised: February 25, 2022.

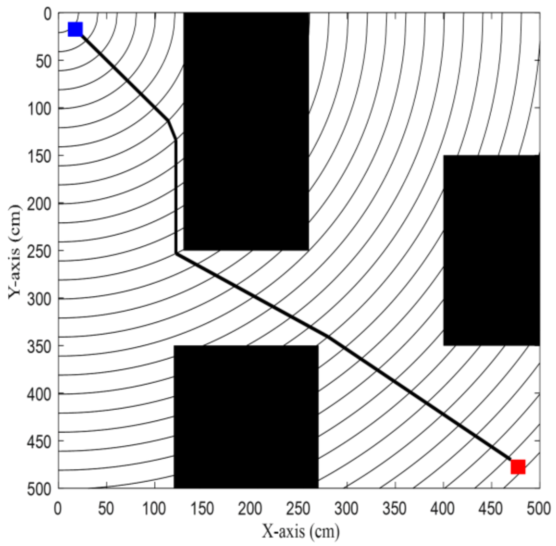


Figure. 20 Results of the shortest-path simulation using the quarter orbits algorithm

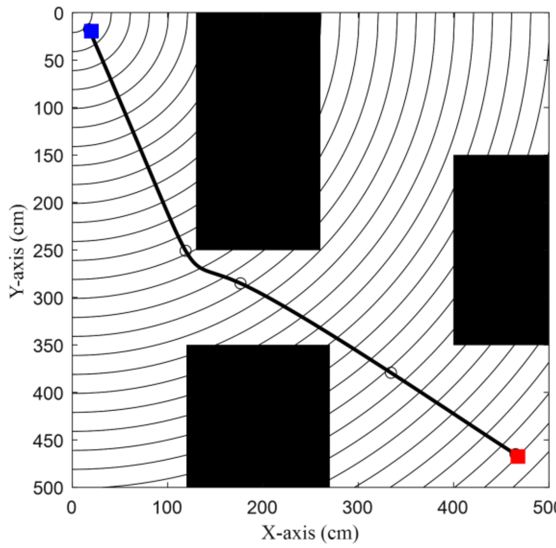


Figure. 22 Results of the shortest-path simulation using the hybrid QOPSO algorithm

(APF) combined with Particle Swarm Optimization (PSO) with a three-point smoothing method that was developed in [38, 39], using a static environment with a workspace of [500 by 500] cm. The results of the simulation process using the Quarter Orbits algorithm is shown in Fig. 20, producing a path of length equals to 684.1660 cm.

In addition, when the same environment was used with the PSO algorithm, as shown in Fig. 21, the simulation process produces a path length equals to 676.6052 cm. While the hybrid QOPSO algorithm, as shown in Fig. 22, produces a path of length equals to 663.856 cm.

The simulation results indicate that the hybrid QOPSO algorithm can successfully generate the shortest path compared with the FMM and FMMHRS algorithms because these methods try to draw a straight line between intermediate points and the goal point. Then, these algorithms will try to determine the location and provide certain permissions around obstacles, leading to free robot navigation without collisions. Therefore, these algorithms generated a longer path between the starting point and the goal. On the other hand, the APF algorithm depends on the attractive potential field function to determine the best path, and to overcome its limitation of getting stuck at local minima, the PSO algorithm was used to improve the generated path. Therefore, the best path generated by these algorithms is still a long one. Moreover, it generates the shortest path compared to the PSO and the Quarter Orbits algorithms. The results of the comparison process are shown in Table 6.

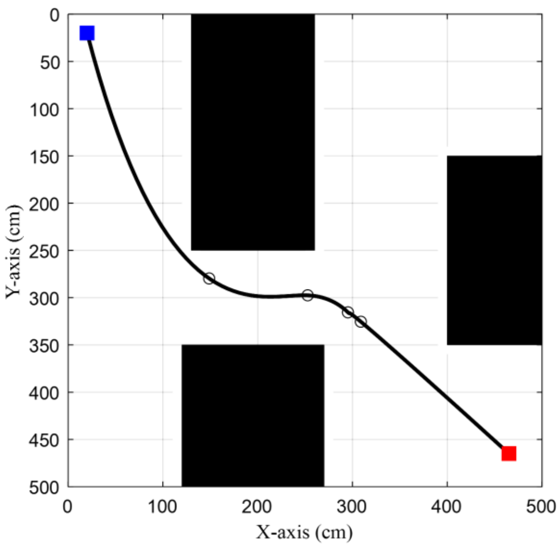


Figure. 21 Result of the shortest-path simulation using the PSO algorithm

Table 6. Comparison of the shortest path with the literature [38, 39]

Algorithm	Path length
FMM [38]	676.08 cm
FMMHRS [38]	664.26 cm
APF+PSO [39]	819. 87 cm
APF+ PSO+3-point [39]	753.26 cm
The proposed QO	684.166 cm
The PSO	676.605 cm
The proposed hybrid QOPSO algorithm	663.856 cm

6. Conclusion

This paper proposed a hybrid algorithm for solving two issues, including (1) generating the smoothest and the shortest path for the mobile robot and (2) avoiding obstacles. The hybrid algorithm takes the advantages of both the quarter orbits algorithm and the PSO algorithm. The simulation

Received: February 7, 2022. Revised: February 25, 2022.

results show that the proposed quarter orbits algorithm provides enhancement on the path length of the cell decomposition algorithm by steering the mobile robot towards the target point and also by using the quarter orbits cells. The quarter orbits (QO) algorithm was compared with the vertical cell decomposition (VCD) and the radial cell decomposition (RCD) using two different environments. Comparison results of the first environment show that the QO algorithm provides enhancement on the path length equals to 29.42% compared to the VCD algorithm and 24.25% compared to the RCD algorithm. Moreover, comparison results of the second environment show that the QO algorithm provides enhancement on the path length equals to 24.48% compared to the VCD algorithm and 7% compared to the RCD algorithm. The proposed hybrid QOPSO algorithm provides a smooth path with the shortest distance compared to that of the proposed quarter orbits algorithm. When comparing the hybrid QOPSO algorithm with the QO algorithm, the QOPSO provides an enhancement on the path length equals to 0.34% compared to the QO algorithm, while the proposed hybrid QOPSO algorithm provided an effective solution in the complex environment compared to the PSO algorithm. When comparing the hybrid QOPSO algorithm with the PSO algorithm using a complex map, the QOPSO provides an enhancement on the path length equals to 13.2% compared to the PSO algorithm. The comparison of the mobile robot path length with different research works shows that the proposed hybrid QOPSO algorithm provided the shortest path with collision avoidance.

As a future work, we will suggest that the proposed hybrid algorithm is adjusted to work with a dynamic environment.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Zainab E. Kanoon, Ahmed Sabah Al-Araji and Mohammed Najm Abdullah enhanced and developed an intelligent path-planning algorithm for mobile robot model. Zainab E. Kanoon described the proposed algorithm for path planning. Ahmed Sabah Al-Araji explained the kinematics mobile model. Zainab E. Kanoon and Mohammed Najm Abdullah discussed the proposed simulation results of this work.

References

[1] S. Sedhumadhavan and E. Niranjana, "An Analysis of Path Planning for Autonomous Motorized Robots", *International Journal of Advance Research*, Vol. 3, No. 6, pp. 1234-1257, 2017.

[2] N. A. K. Zghair and A. S. A. Araji, "A One Decade Survey of Autonomous Mobile Robot Systems", *International Journal of Electrical and Computer Engineering*, Vol. 11, No. 6, pp. 4891-4906, 2021.

[3] M. Fuad, T. Agustinah, and D. Purwanto, "Modified Headed Social Force Model based on Hybrid Velocity Obstacles for Mobile Robot to Avoid Disturbed Groups of Pedestrians", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 222-241, 2021.

[4] B. K. Patle, G. B. L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On Path Planning Strategies for Navigation of Mobile Robot", *Defence Technology*, Vol. 15, No. 4, pp. 582-606, 2019.

[5] M. N. Zafar and J. C. Mohanta, "Methodology for Path Planning and Optimization of Mobile Robots: A Review", *Procedia Computer Science*, Vol. 133, pp. 141-152, 2018.

[6] M. Alajlan, A. Koubâa, I. Châari, H. Bennaceur, and A. Ammar, "Global Path Planning for Mobile Robots in Large-Scale Grid Environments Using Genetic Algorithms", In: *Proc. of International Conf. on Individual and Collective Behaviors in Robotics Collect*, pp. 1-8, 2013.

[7] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H. M. Tai, "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm", In: *Proc. of The Congress on Evolutionary Computation*, pp. 1338-1345, 2004.

[8] S. Prongnuch and S. Sitjongsataporn, "Differential Drive Analysis of Spherical Magnetic Robot Using Multi-Single Board Computer", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 4, pp. 264-275, 2021.

[9] M. Fuad, T. Agustinah, and D. Purwanto, "Collision Avoidance of Multi Modal Moving Objects for Mobile Robot Using Hybrid Velocity Obstacles", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 407-421, 2020.

[10] A. S. A. Araji, A. K. Ahmed, and M. K. Hamzah, "Development of a Path Planning Algorithms and Controller Design for Mobile Robot", In:

Proc. of Scientific Conference of Electrical Engineering, pp. 72-77, 2018.

[11] A. S. A. Araji, K. E. Dagher, and B. A. Ibraheem, "An Intelligent Cognitive System Design for Mobile Robot based on Optimization Algorithm", In: *Proc. of The Scientific Conference of Electrical Engineering*, pp. 84-89, 2018.

[12] F. Rubio, F. Valero, and C. L. Albert, "A Review of Mobile Robots: Concepts, Methods, Theoretical Framework, and Applications", *International Journal of Advanced Robotic Systems*, Vol. 16, No. 2, pp. 1-22, 2019.

[13] S. K. Debnath, R. Omar, S. Bagchi, S. E. N, M. H. A. S. Kandar, K. Foyso, and T. K. Chakraborty, "Different Cell Decomposition Path Planning Methods for Unmanned Air Vehicles - A Review", In: *Proc. of The 11th National Technical Seminar on Unmanned System Technology*, pp. 99-111, 2021.

[14] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mquiee Robots", In: *Proc. of the Autonomous Robot Vehicles*, pp. 396-404, 1985.

[15] H. Choset, S. Walker, K. E. Ard, and J. Burdick, "Sensor-Based Exploration: Incremental Construction of the Hierarchical Generalized Voronoi Graph", *International Journal of Robotics Research*, Vol. 19, No. 2, pp. 96-125, 2000.

[16] S. K. Debnath, R. Omar, N. B. A. Latip, S. Shelyna, E. Nadira, C. K. N. C. K. Melor, T. K. Chakraborty, and E. Natarajan, "A Review on Graph Search Algorithms for Optimal Energy Efficient Path Planning for an Unmanned Air Vehicle", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 15, No. 2, pp. 743-749, 2019.

[17] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, Vol. 1, No. 1, pp. 269-271, 1959.

[18] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100-107, 1968.

[19] A. Stentz, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments", *Carnegie-Mellon University Pittsburgh Pa Robotics*, 1993.

[20] P. G. Zavlangas and S. G. Tzafestas, "Motion Control for Mobile Robot Obstacle Avoidance and Navigation: A Fuzzy Logic-based Approach", *Systems Analysis Modelling Simulation*, Vol. 43, No. 12, pp. 1625-1637, 2003.

[21] D. Janglová, "Neural Networks in Mobile Robot Motion", *International Journal of Advanced Robotic Systems*, Vol. 1, No. 1, pp. 15-22, 2004.

[22] X. L. Tang, L. Li, and B. Jiang, "Mobile Robot SLAM Method based on Multi-Agent Particle Swarm Optimized Particle Filter", *The Journal of China Universities of Posts and Telecommunications*, Vol. 21, No. 6, pp. 78-86, 2014.

[23] T. A. N. G. Zheng, H. Huan, and A. Sloman, "Ant Colony System Algorithm for Real-Time Globally Optimal Path Planning of Mobile Robots", *Acta Automatica Sinica*, Vol. 33, No. 3, pp. 279-285, 2007.

[24] M. A. C. Cruz, V. A. Ramirez, and U. H. H. Belmonte, "Mobile Robot Path Planning Using Artificial Bee Colony and Evolutionary Programming", *Applied Soft Computing*, Vol. 30, pp. 319-328, 2015.

[25] T. Shibata and T. Fukuda, "Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic", In: *Proc. of the 8th IEEE International Symposium on Intelligent Control*, pp. 565-570, 1993.

[26] R. A. Brooks and T. L. Perez, "A Subdivision Algorithm in Configuration Space for Findpath with Rotation", *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 224-233, 1985.

[27] D. Z. Chen, R. J. Szczerba, and J. J. Uhran, "Planning Conditional Shortest Paths Through an Unknown Environment: A Framed-Quadtree Approach", In: *Proc. of International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots Pittsburgh*, pp. 33-38, 1995.

[28] J. T. Schwartz and M. Sharir, "On The 'Piano Movers' Problem I. The Case of a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers", *Communications on Pure and Applied Mathematics*, Vol. 36, No. 3, pp. 345-398, 1983.

[29] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", In: *Proc. of International Conference on Neural Networks*, pp. 1942-1948, 1995.

[30] O. A. R. A. Wahhab and A. S. A. Araji, "Path Planning and Control Strategy Design for Mobile Robot Based on Hybrid Swarm Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 565-579, 2021.

[31] R. H. A. Rubayi, M. K. Abd, and F. M. Flaih, "A New Enhancement on PSO Algorithm for Combined Economic-Emission Load Dispatch Issues", *International Journal of Intelligent*

Received: February 7, 2022. Revised: February 25, 2022.

Engineering and Systems, Vol. 13, No. 1, pp. 77-85, 2020.

[32] F. V. D. Bergh and A. P. Engelbrecht, “A Study of Particle Swarm Optimization Particle Trajectories”, *Information Sciences*, Vol. 176, No. 8, pp. 937-971, 2006.

[33] O. A. R. A. Wahhab and A. S. A. Araji, “An Optimal Path Planning Algorithms for a Mobile Robot”, *Iraqi Journal of Computers, Communication, Control & Systems Engineering*, Vol. 21, No. 2, pp. 44-58, 2021.

[34] A. S. A. Araji, M. F. Abbod, and H. S. A. Raweshidy, “Design of a Neural Predictive Controller for Nonholonomic Mobile Robot based on Posture Identifier”, In: *Proc. of the Lasted International Conference Intelligent Systems and Control*, pp. 198-207, 2011.

[35] O. A. Salama, M. E. Eltaib, H. A. Mohamed, and O. Salah, “RCD: Radial Cell Decomposition Algorithm for Mobile Robot Path Planning”, *IEEE Access*, Vol. 9, pp. 149982-149992, 2021.

[36] Y. Peng, Z. Wang, J. Jin, P. Y. Huang, and W. Lu, “The Global Trajectory Programming of Robot based on Adaptive Ant Colony Algorithm”, In: *Proc. of International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol. 1, pp. 108-111, 2013.

[37] C. Yen and M. Cheng, “A Study of Fuzzy Control with Ant Colony Algorithm Used in Mobile Robot for Shortest Path Planning and Obstacle Avoidance”, *Microsystem Technologies*, Vol. 24, No. 1, pp. 125-135, 2018.

[38] R. K. Mandava, K. Mrudul, and P. R. Vundavilli, “Dynamic Motion Planning Algorithm for a Biped Robot Using Fast Marching Method Hybridized with Regression Search”, *Acta Polytechnica Hungarica*, Vol. 16, No. 1, pp. 189-208, 2019.

[39] R. K. Mandava, S. Bondada, and P. R. Vundavilli, “An Optimized Path Planning for the Mobile Robot using Potential Field Method and PSO algorithm”, In: *Proc. of International Conference on Soft Computing and Problem Solving*, pp. 139-150, 2017.