Received: March 29, 2022. Revised: April 26, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Static and Dynamic Path Planning Algorithms Design for a Wheeled Mobile Robot Based on a Hybrid Technique

Ammar Abdul Ameer Rasheed[1]* **Ahmed Sabah Al-Araji[1]** **Mohammed Najm Abdullah[1]**

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

Abstract: During the previous years, the number of researchers who discussed the field of path planning algorithms has increased. In this paper, a proposed hybrid technique for path planning based on the benefits of sampling-based algorithms and heuristic path planning algorithms was enhanced. Rapidly-exploring Random Tree star (RRT*) was used to generate the candidate nodes that represent a collision-free path. These candidate nodes were applied to the Particle Swarm Optimization (PSO) algorithm to generate the shortest and the smoothest path planning (the optimal path planning) for the mobile robot. The RRT*PSO algorithm was applied in two scenarios (static and dynamic environments) with a workspace of [500×500] cm using MATLAB 2021a. In the static environment, a workspace full of obstacles was chosen to find the shortest collision-free path. In addition, a reference path equation was found to calculate the reference linear and angular velocities of the mobile robot as well as the linear and angular velocities of the right and left wheels. In this work, the suggested hybrid RRT*PSO algorithm improves path length 34.27% compared to the A* algorithm and the fuzzy analytic hierarchy process (A*–FAHP hybrid) algorithm and 0.35% compared to the Self-adaptive evolutionary game-based particle swarm optimization (SAEGBPSO) algorithm. In the dynamic environment, an algorithm named (contour path down and contour path up) was suggested to avoid collisions with dynamic obstacles. By using this suggested algorithm, the reference linear and angular velocities of the mobile robot as well as the linear and angular velocities of the right and left wheels were calculated to obtain a smooth path followed and free-navigation with dynamic obstacles in an environment.

Keywords: Mobile robot, Path planning, Static and dynamic environments, Sampling-based algorithms, Heuristic algorithm.

1. Introduction

Path planning is considered an essential step in autonomous mobile robots and an important topic in the field of mobile robot navigation [1]. Path planning is the process of determining an obstacle-free path from the initial state (start) to the final state (goal) to keep the path safe without collisions with other objects. At the same time, execution models such as energy, time, or distance should stay at the optimal points. From all the above, the distance represents the most important factor that should be considered [2]. In particular, designing a path to be feasible and practicable for a mobile robot is an intractable task. There are many applications related to mobile robots in different fields of our life, such as robotics, medicine, virtual reality, search and rescue operations, and bioinformatics. Generally, algorithms of motion planning determine a valid configuration to create a collision-free path for the mobile robot. This becomes difficult as the extent of the configuration space increases [3]. The increase in applications related to intelligent mobile robots reflects the importance of path planning in many fields of robotics. As a result, much research is related to path optimization for mobile robots. However, achieving optimal paths faces many challenges in some complex environments. Path optimization finds the best collision-free path from the initial state (start) to the final state (goal) while respecting specific performance metrics [4]. In the mobile robot, autonomous navigation is considered the main technology of intelligent robot motion systems. The high ability of a mobile robot to execute tasks can be determined by many factors such as the reliable location and the effective path planning. To this end, planning an accurate position and efficient paths remains a challenge. Despite that, advanced works have been done in path planning and goal positioning of mobile robots.

However, many problems still face some issues such as the location of the robot, the environment map generation (which should be more credible and accurate), the algorithm of path planning (which should satisfy the characteristics and restrictions of robot motion), and the paths (which should be designed more effectively and easily) [5].

Several path-planning algorithms were proposed in terms of various theories that were divided into three types [6]. It has been noticed that the main issue of path planning is how to find the optimal (the shortest) path between the start and the goal points while avoiding collision with obstacles. For that, many studies have been conducted to study and solve this issue. For example, Wahhab provided an enhancement of the performance of a mobile robot in terms of the actual output trajectory tracking by using a convolutional neural network controller with off-line and on-line tuning Back-Propagation techniques [7]. However, this study does not consider dynamic obstacles. Rapidly-exploring Random Tree start (RRT*) was used by [8], where the method used a single objective function to achieve global coverage and maximize the utility of a path in a global context, but the problem of this method is required many paths to be continuously adapted to the current map. In addition, the authors in [9] proposed a combined model for robots navigating in confined workplaces that automatically synthesizes local communication and decision-making rules but the system still face the problem of time delay of communication among robots. Additionally, dynamic artificial potential field (DAPF) was used to reduce the local minimum that typically occurs in conventional PF while satisfying the path-planning requirements. However, the limitation of this method is related to generate a sub-optimal path instead of optimal path [10]. Hybrid algorithms were used by many researchers to enhance the path-planning algorithm used hybrid algorithms. For instance, a hybrid Particle Swarm Optimization-Modified Frequency Bat (PSO-MFB) algorithm integrated with local search and obstacle detection and avoidance strategies was used to propose a path-planning algorithm for mobile robots. The algorithm was tested in static and dynamic environments [11]. However, the drawback of this system related to a complex chaotic dynamic workspace with moving goal. Furthermore, a novel neuro-fuzzy technique handled the problem of mobile robot autonomous navigation in an unstructured environment [12]. Moreover, a hybrid velocity obstacle-based modified headed social force model (HVO-based Modified HSFM) was proposed to avoid disturbing crowds of pedestrians while navigating in complex and dense environments. Random paths of mobile robot, single pedestrian, and group of pedestrians were created by used probabilistic roadmap (PRM). The disadvantage of using (PRM) is not giving the optimal solution every time [13].

In our paper, the problem definition considers in two points. The first point is achieving collision-free path planning to avoid obstacles in static and dynamic environments. The second point is finding the shortest and the smoothest path from the start to the goal points.

The main contribution of this paper is the development of an optimal or near-optimal short path with no collisions using the suggested hybrid algorithm, namely the (RRT*PSO). There are three types of environments with regards to autonomous mobile robot path planning research field: a static environment with fixed obstacles, a dynamic environment with movable obstacles, and a dynamic environment with movable obstacles and a movable target point simultaneously. In this context, a dynamic environment is an unknown environment including multiple movable obstacles that move around the space at the same time with the robot with random direction, and velocity. The environment may also incorporate some static obstacles (e.g. walls, doors, furniture, etc.) [14, 15]. This paper is organized as follows: Section 2 addresses kinematics models of differential wheeled mobile robots. Section 3 includes research methodologies for path-planning algorithms, while Section 4 describes the proposed hybrid algorithm. Section 5 shows the simulation results, and the conclusions are given in Section 6.

2. Kinematics models of differential wheeled mobile robots

The platform of the Wheeled Mobility Robot (WMR) consists of two wheels connected to a parabolic shaft and one multidirectional wheel that is fixed to the front or rear of the platform. This wheel keeps the body stable during the motion of the mobile robot in different directions. This platform is a rigid body on wheels, operating on a horizontal plane, as shown in Fig. 1 [16, 17].

The platform of the mobile robot consists of two DC motors that drive the right and left wheels, and the multidirectional wheel, ($\boldsymbol{m}_c$) is the mobile robot center mass, $\boldsymbol{L}$ is the distance between the right and

Received: March 29, 2022. Revised: April 26, 2022.

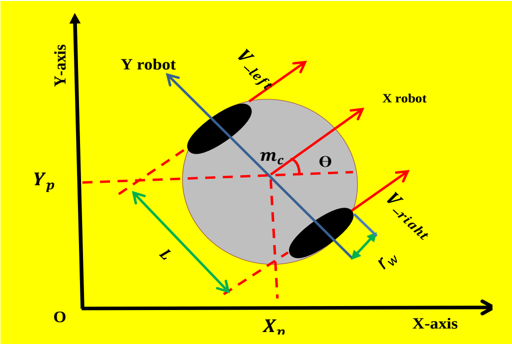


Figure. 1 Sketch of the mobile robot's platform

Table 1. Definitions of the equations' parameters

Parameter	Definition
$V_{\_left}$	Velocity of the left wheel
$V_{\_right}$	Velocity of the right wheel
L	The distance between the two wheels
Ts	The sampling rate time

the left wheels, and r_w is the radius of these wheels. The global coordinate frame is represented by [O, X-axis, and Y- axis]. X_p and Y_p are the coordinates of the point $\boldsymbol{m}_c$ and (0) is the direction angle of the mobile robot measured from the X-axis. The global coordinates are used to define the mobile robot's configuration. As a conclusion, the computer simulation equations are as follows [18, 19]:

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

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

$$\theta(k) = \left[0.5 \times L\left(V_{\ left} - V_{\ right}\right) \times Ts\right] + \theta(k-1) \quad (3)$$

3. Research methodologies for path planning algorithms

This section explains the research methodologies for the path planning algorithms used in our proposed algorithm, namely the RRT*PSO.

3.1 Rapidly-exploring random tree (RRT and RRT*)

Sampling-based algorithms for path planning had attracted a lot of attention during the last decade. The RRT*, a type (family) of RRT, is a source of concern for researchers because of its asymptotic optimality. LaValle and Kuffner, professors at the University of Tokyo, Japan, proposed the RRT., which is an algorithm designed by randomly building a space-filling tree to cover a large space in a short time. The RRT can be applied in a path-planning algorithm to avoid the modeling of space by the detection of any collision of sampling points in the space. In addition, the complexity of constraining and the path-planning problem of a high-dimensional space can be solved by applying the RRT. It is distinguished by its ability to search in a high-dimensional space fast and efficiently, and the search results are directed to the blank area to be used in continuous path planning problems. The RRT is one of the most popular techniques that propose a great degree of flexibility and reliability. Particularly, the RRT is useful for solving problems of multi-degree-of-freedom robots in complex path planning in static and dynamic environments [20]. Fig. 2 shows the algorithm's basic operation [21].

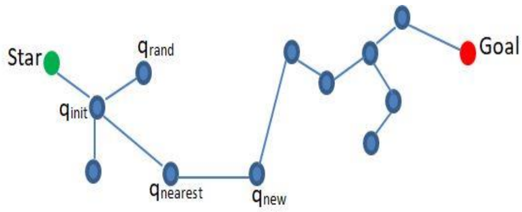


Figure. 2 Plan diagram of the RRT algorithm

At initializing state, only one node is contained in the random tree T; the root node q_{init}. In the beginning, the sample function randomly chooses a sample point q_{rand} from the state space, after that the nearest function chooses a node $q_{nearest}$ from the random tree ending with the extend function, which spreads a distance from $q_{nearest}$ to q_{rand} to catch new nodes q_{new}. When a collision has occurred in q_{new} with the obstacle, the extend function comes back null and discards the growth. All the above steps are repeated until the distance between $q_{nearest}$ and the target point q_{goal} becomes less than a threshold. At this point, the target point is reached by the random tree. In addition, the upper limit of the running time or the upper limit of the searching times can be set [21]. The RRT* is an RRT extension that uses triangle equality to find the best path from the start to the goal nodes. Lower-cost (more optimum) paths are created as the number of nodes grows. RRT-connect is especially beneficial for robotic arms, such as PUMA (Programmable Universal Machine for Assembly) robots. When the majority of planning questions are expected to be in relatively open locations, the link heuristic performs best. The connect heuristic was developed specifically for this type of situations [22].

Rapidly random-exploring tree (RRT) and its variants (family) are very common because of their ability to fast and efficiently discover the state space.

Received: March 29, 2022. Revised: April 26, 2022.

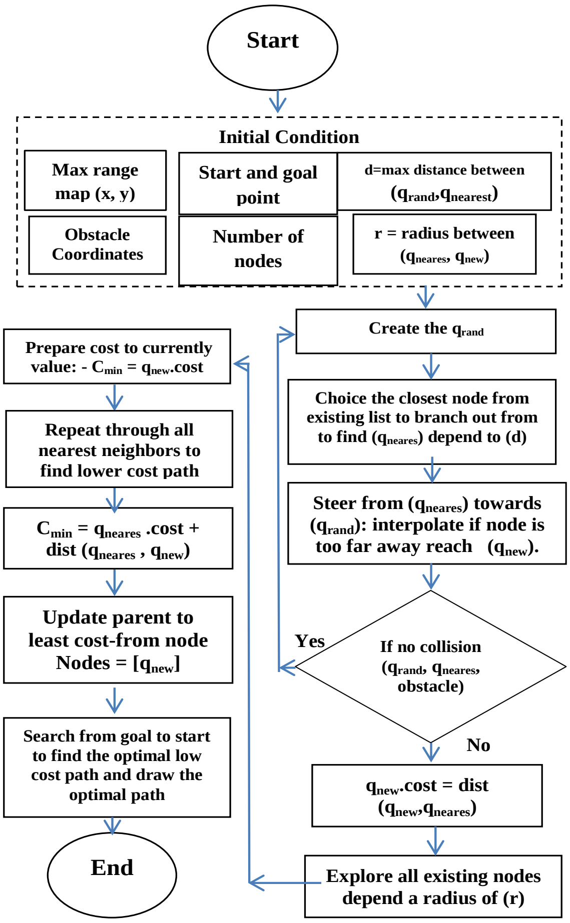


Figure. 3 Flowchart of the RRT algorithm

However, the disadvantage is related to their sensitivity to the initial solution and the slow convergence to the optimal solution. In this regard, they consume much memory and time to reach the optimal path. To this end, finding a short path in many applications such as the autonomous vehicle with limited power/fuel is still a critical issue [23]. The operation of RRT* is similar to that of the RRT, but two different keys add to the algorithms shown in Fig. 3, which shoes the flowchart of RRT* procedure from the start to the goal.

3.2 Particle swarm optimization (PSO) algorithm

It is a population-based stochastic optimization algorithm and a famous heuristic path-planning algorithm. In the PSO framework, the algorithm begins from a random initialization with nominee solutions. Then, a global optimal solution via iteration based on position and velocity updating can be found. For each particle, the velocity characterizes the searching direction and is dynamically updated using its previous value, the personal best position, and the global best position. Mainly, the new position is a dependent value determined by its previous value and the current velocity. This means that when the new position of a particle is better than the previous value related to the fitness function, then the restoration will occur to the new one as the personal best position. If the value of the new position is better than that of other particles, at this point it will be restored as the global best position [24, 25]. The social behavior of some animals in terms of the group's ability to locate a desirable position in the given area was the inspiration to propose the PSO. In the PSO algorithm, each particle has a memory enabling it to track the best position of the previous iteration. The particle's optimal position is P_{best} and the particle's global optimal position is G_{best}, each with a velocity V_i and position X_i. The speed and position update methods for the i^{th} particle at the $(k+1)$ iteration are represented in Eqs. (4) and (5), respectively [26-30].

Table 2. The parameters' definition of the PSO

Parameter	Definition
$V_i(k)$	i^{th} particle's velocity in iteration k
$X_i(k)$	i^{th} position vectors in iteration k
$P_{ibest}(k)$	The best fitness values for the i^{th} particle
$G_{best}(k)$	The best global fitness value
w	Inertia weight of the velocity
c_1 *and* c_2	The acceleration coefficients $c_1 + c_2 < 4$
r_1 *and* r_2	Random numbers between [0, 1].

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

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

The definitions of the PSO parameters of Eqs. (4) and (5) are shown in Table 2.

The following steps show the PSO algorithm:-

Step 1: Start to set particles with random position and velocity vectors.
Step 2: Calculate the fitness value of the particles using the fitness equation
Step 3: Find and update P_{ibest} and G_{best}
Step 4: Calculating and updating the velocity and position for each particle
Step 5: Stop if the termination requirements are met. Otherwise, go back to **Step 2**.
Step 6: Display the G_{best} optimal solution.

4. The proposed algorithm

In this section, the proposed hybrid path planning algorithm based on RRT* and PSO is described. This paper combines these two algorithms to produce a hybrid algorithm called the (RRT*PSO). Fig. 4 shows the flowchart of the RRT*PSO for robot path planning.

Received: March 29, 2022. Revised: April 26, 2022.

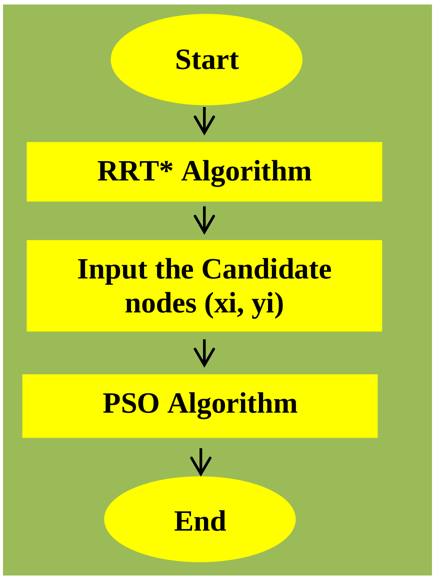


Figure. 4 Flowchart of the RRT*PSO hybrid algorithm

First, the RRT* was applied to find a collision-free path. The RRT* algorithm will pledge a path from the start point to the goal point by moving the mobile robot from node to node with obstacle avoidance until it reaches the goal node. However, the optimal path cannot be reached by using the RRT* because this method produces a winding path that leads to consume more power and produce an additional distance in the path. Second, the candidate nodes, which were extracted using the RRT*, were applied to the PSO algorithm. At this state, the PSO algorithm produces the smoothest and the shortest path. The hybrid algorithm of the RRT*PSO was applied in two scenarios; a static environment with static obstacles and a dynamic environment with dynamic obstacles.

4.1 Static environment with static obstacles

Herein, two maps were planned using one static obstacle, as shown in Fig. 5, and two static obstacles, as shown in Fig. 6.

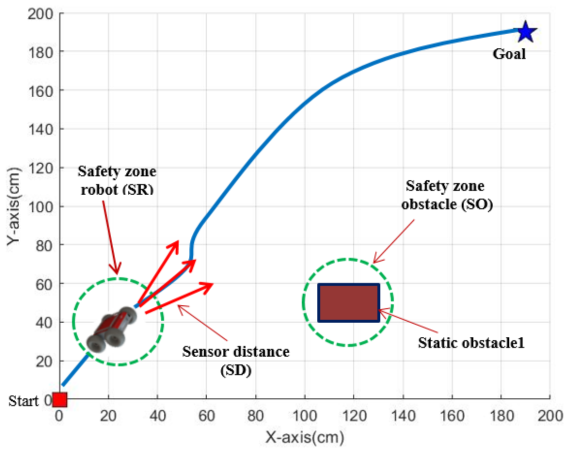


Figure. 5 Static environment with one static obstacle

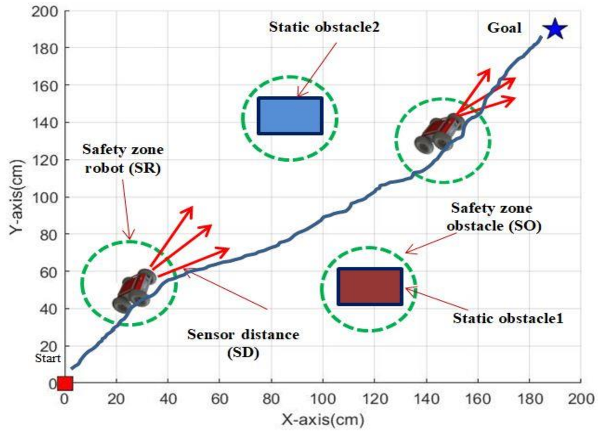


Figure. 6 Static environment with two static obstacles

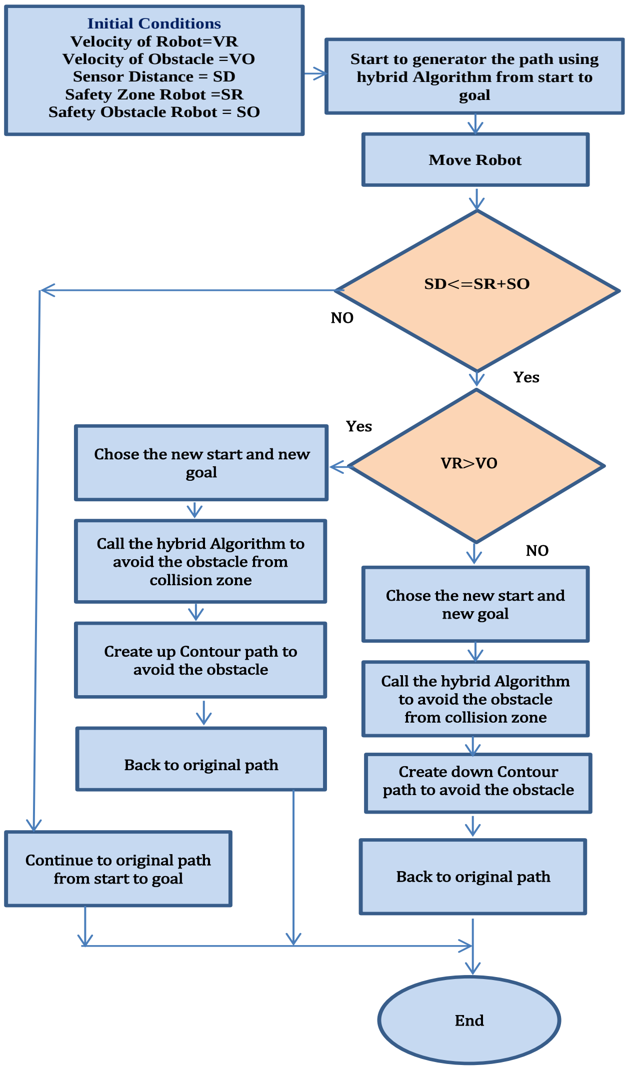


Figure.7 Proposed algorithm to make the robot avoid collision with the dynamic obstacle

In Fig. 5, the map shows the robot moves at a constant velocity from the start node to the goal node in a way that the robot has a sensor distance (SD). The mobile robot has a safety zone (SR), which is calculated as the distance between the driving wheels of the robot along its Y-axes plus half of this distance

Received: March 29, 2022. Revised: April 26, 2022.

to create a circle area representing the safety zone and the distance above represents its diameter. At the same time, the obstacle has a safety zone (SO) which is calculated in the same way as that used for the mobile robot. In Fig. 6, the above scenario was repeated using two obstacles.

4.2 Dynamic environment with dynamic obstacles

Fig. 7 shows the flowchart of the proposed algorithm to make the robot avoid collision with the dynamic obstacle.

Fig. 8 shows our hypothesis about the movement of the robot and obstacles in constant velocity. A point where the robot and the obstacle will be merged is named the cross point collusion, as shown Fig. 9. In addition, Fig. 10 and 11 represent the path that the mobile robot will follow to avoid the movable obstacle. When the robot checks the collision event with the obstacle depending on the sensor distance, the robot selected the new start and the new goal and we call the hybrid algorithm the RRT*PSO to create the (contour path up or down) to avoid the dynamic obstacle.

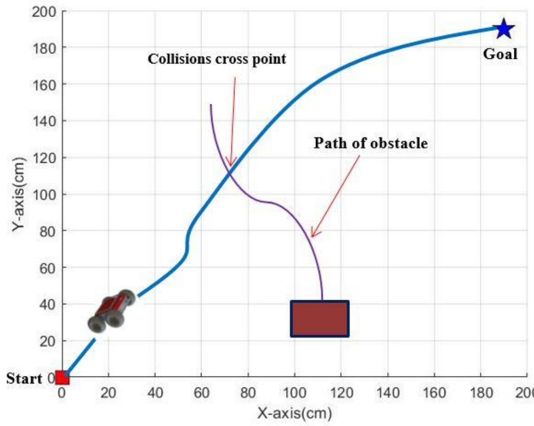


Figure. 8 The path for the robot and the moving obstacle

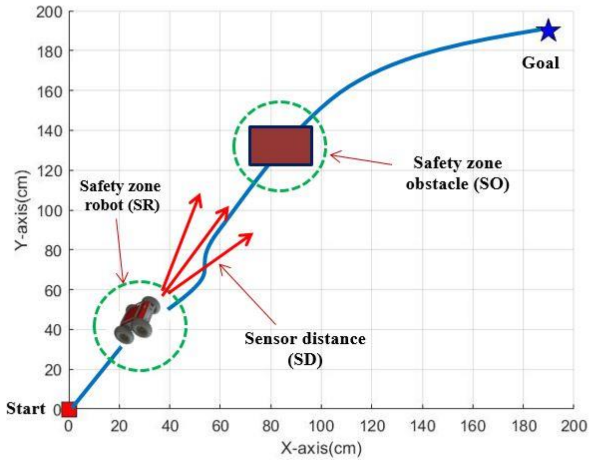


Figure. 9 Obstacle moves and reaches the collision cross point

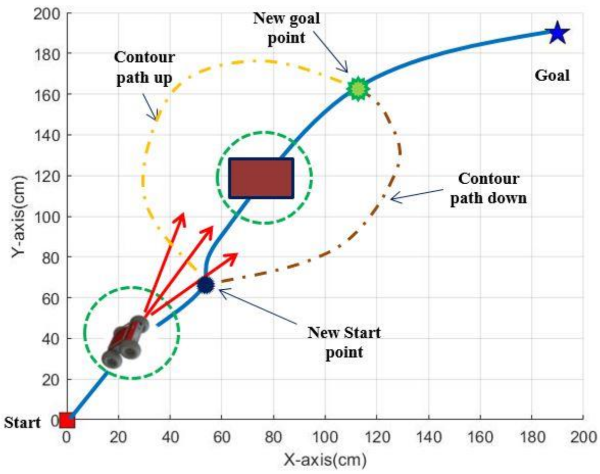


Figure. 10 Up and down contour path

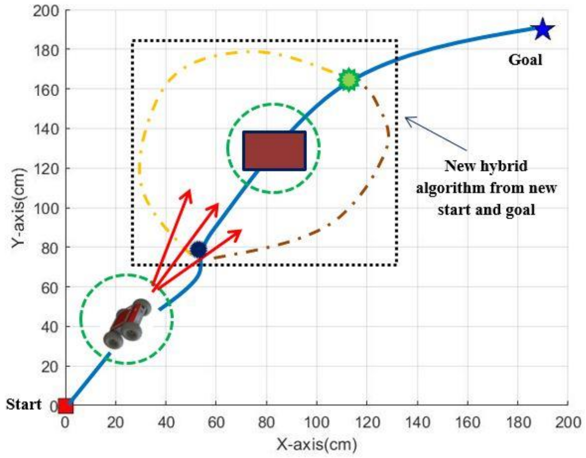


Figure. 11 Hybrid algorithm to avoid the obstacle from the collision zone

5. Simulation results

Static and dynamic obstacle environments were used with a workspace of [500×500] cm, which is shown in Fig. 12. The MATLAB 2020a package with computer hardware specifications of Intel Core i5-1035G7 with 8.00 GB of RAM, and CPU of 1.20GHz were used.

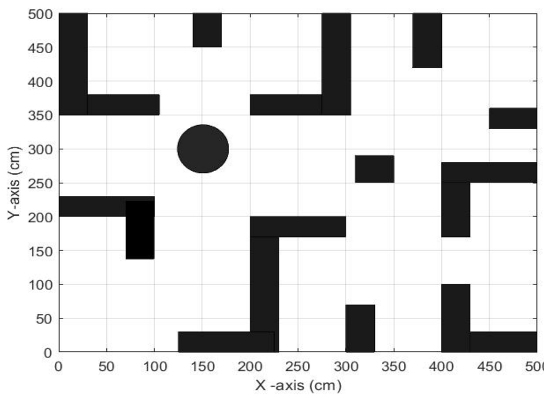
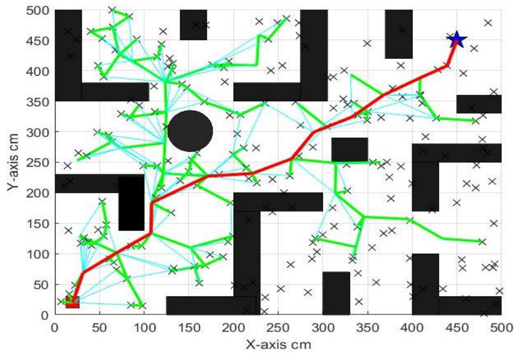


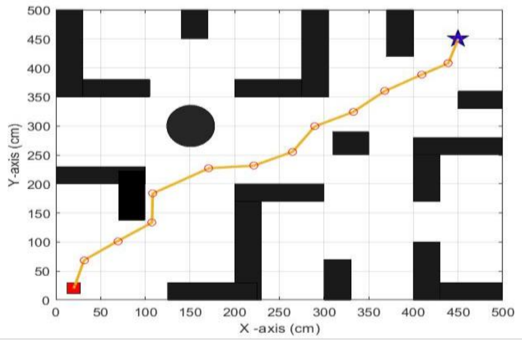
Figure. 12 The proposed environment with obstacles

Received: March 29, 2022. Revised: April 26, 2022.

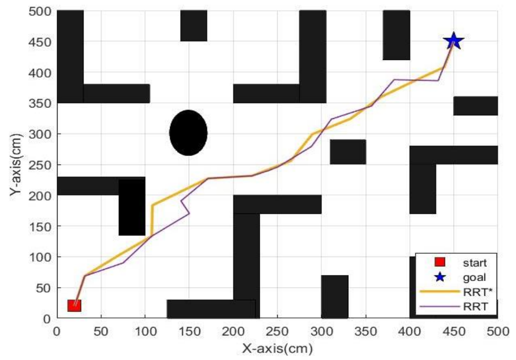
5.1 The first scenario (static environment)



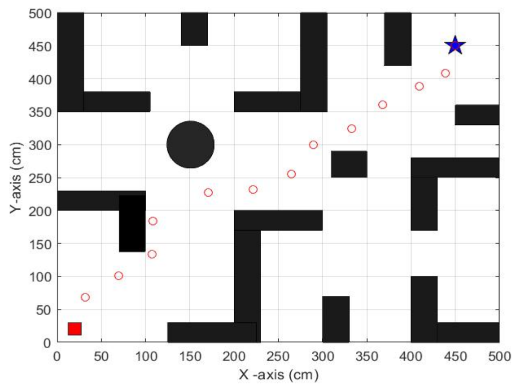
(a)



(b)



(c)

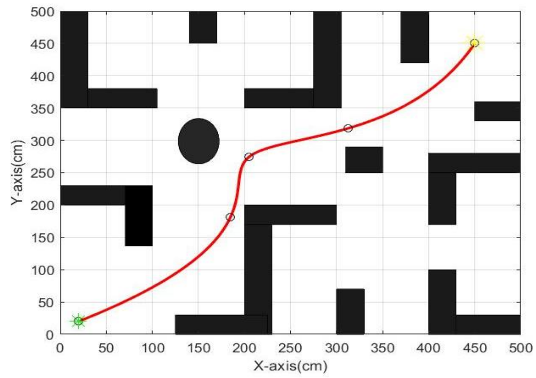


(d)

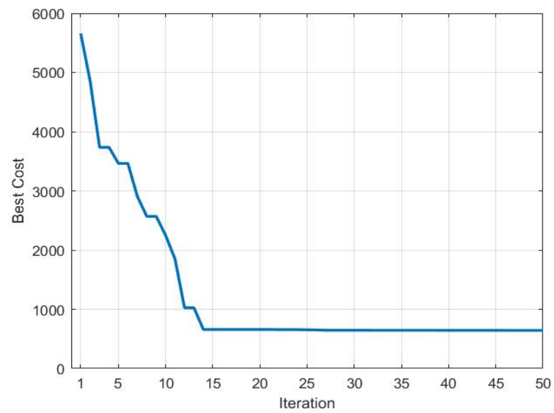
Figure. 13 Simulation result of the RRT* and the RTT, (a, and b) the optimal path of the RRT* algorithm, (c) the optimal path of the RRT&RRT* algorithm and (d) candidate nodes for the optimal path of the RRT* algorithm

The first scenario is a static environment, which is filled with static obstacles, and all information about the locations of the objects in the workspace is provided. Three algorithms including the RRT* algorithm, the RRT algorithm, the PSO algorithm, and the proposed hybrid RRT*PSO algorithm are used to find a collision-free path, and the results are compared among them to find the shortest distance path, considering that a safety zone between the robot and the obstacles must be preserved. The start node position of the mobile robot is at (20, 20) cm, and the goal position node is at (450, 450) cm. After applying the RRT*, the shortest distance of the optimal path was equal to 649.6 cm, as shown in Fig. 13(a) and (b). In the same workspace environment when the RRT was applied, the distance of the optimal path was equal to 656.4 cm, as shown in Fig. 13 (c). On the other hand, Fig. 13(d) shows the candidate nodes for the optimal path of RRT*.

Fig. 14(a) shows the optimal path distance when the PSO algorithm was used in the proposed environment. The best cost function was found in iteration number 15, as shown in Fig. 14(b) with maximum iterations' number equals to 50 iterations. The value of the PSO optimal path distance is equal to 660 cm.



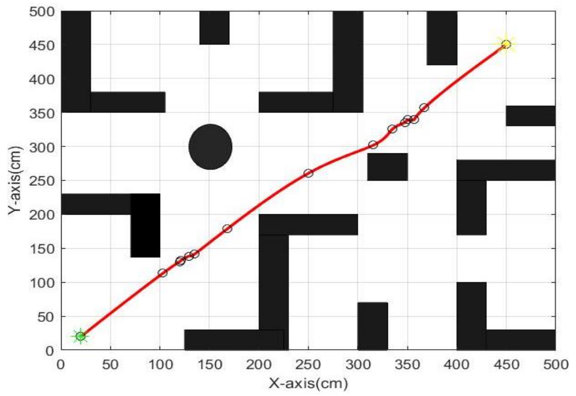
(a)



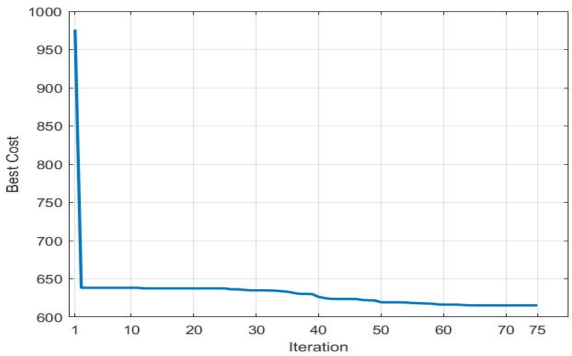
(b)

Figure. 14 The PSO algorithm: (a) the optimal path and (b) the best cost function

Received: March 29, 2022. Revised: April 26, 2022.



(a)



(b)

Figure.15 The proposed hybrid algorithm (RRT*PSO): (a) the optimal path and (b) the best cost function

Table 3. Comparison of the optimal path distance

Algorithm	**Path Length (cm)**	**Iteration**	**No. of input node**	**No. of output node**
RRT	665.4	-	150	18
RRT*	649.6	-	150	14
PSO	660	50	-	-
The proposed algorithm	615.3	50	14	-

In Fig. 15 (a), the proposed hybrid algorithm of RRT*PSO was applied to find the optimal path in the same proposed environment. The best cost function was found in iteration number 12, as shown in Fig. 15 (b) with a maximum iterations' number equals to 50 iterations. The value of the proposed hybrid algorithm optimal path distances is equal to 615.3 cm.

By comparing the RRT, the RRT*, and the PSO algorithms with the proposed hybrid algorithm, it has been found that the proposed algorithm generated the shortest and smoothest path from the starting node to the goal node, as shown in Table 3.

The reference path equation for the optimal path of the RRT*PSO hybrid algorithm was calculated from the optimal path, as shown in Eq. (6):

$$y_{ref}\left(x_{ref}\right) = 3.352e-06 \times x^3_{ref} - 0.00284x^{2}_{-ref} + 1.472 \times x^{1}_{-ref} - 15.67 \quad (6)$$

Depending on the reference path equation (Eq. 6), the reference linear velocity ($v_{\_ref}$.) and the reference angular velocity ($w_{\_ref}$) of the platform of the mobile robot can be calculated as shown in Eqs. (7) and (8), respectively [19].

$$v_{\_ref} = \sqrt[2]{\left(\dot{x}_{\_ref}\right)^2 + \left(\dot{y}_{\_ref}\right)^2} \quad (7)$$

$$w_{\_ref} = \frac{\left(\ddot{y}_{\_ref} \times \dot{x}_{ref} - \ddot{x}_{\_ref} \times \dot{y}_{\_ref}\right)}{\left(\left(\dot{x}_{\_ref}\right)^2 + \left(\dot{y}_{\_ref}\right)^2\right)} \quad (8)$$

Fig. 16 shows the optimal path of the proposed hybrid (RRT*PSO) algorithm, while Fig. 17 shows the reference linear and angular velocities of the mobile robot that were used in order to track the desired optimal path.

The linear velocities of the right wheel $v_{\_R}$ and the linear velocities of the left wheel $v_{\_L}$, as well as the angular velocities of the right and the left wheels $w_{\_R}$ and $w_{\_L}$, respectively, can be calculated using the equations below [20]:

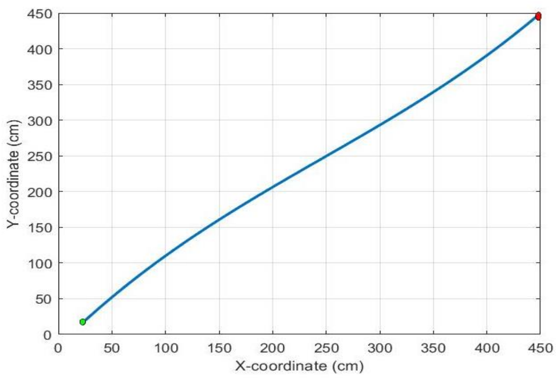


Figure. 16 Optimal path of the proposed RRT*PSO hybrid algorithm

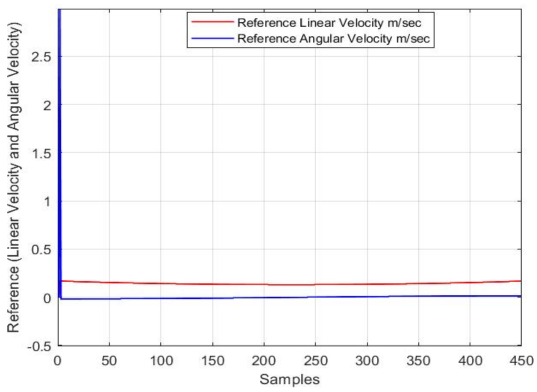


Figure. 17 The reference linear and angular velocities of the mobile robot

 DOI: 10.22266/ijies2022.0831.16

Received: March 29, 2022. Revised: April 26, 2022.

$$v_{\_R} = \frac{(2v_{\_ref}+Lw_{\_ref})}{2} \quad (9)$$

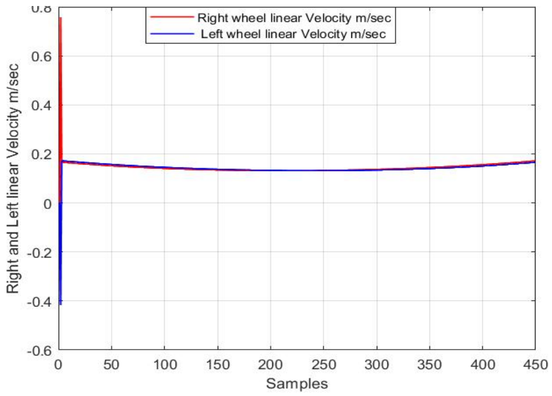
$$v_{\_L} = \frac{(2v_{\_ref}-Lw_{\_ref})}{2} \quad (10)$$

$$w_{\_R} = \frac{(2v_{\_ref}+Lw_{\_ref})}{2r} \quad (11)$$

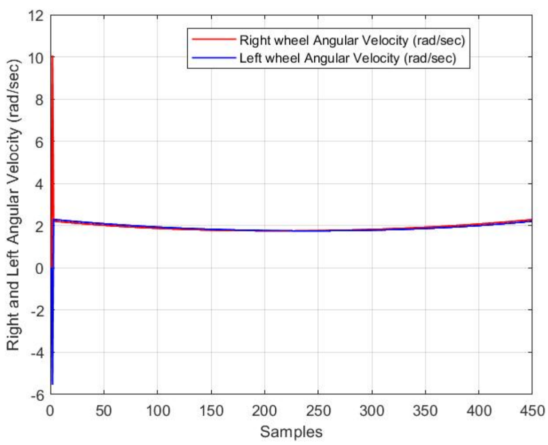
$$w_{\_L} = \frac{(2v_{\_ref}-Lw_{\_ref})}{2r} \quad (12)$$

In the equations above, r is equal to 0.076 m, which indicates the radius of the wheel of the mobile robot, while L is equal to 0.38 m, which indicates the distance between the two wheels of the mobile robot with a sample time of 0.15 sec.

In order to demonstrate the wheels velocities of the mobile robot, Fig. 18(a) shows the linear velocity of the right and the left wheels of the mobile robot. While Fig. 18(b) shows the angular velocity of the right and the left wheels of the mobile robot.



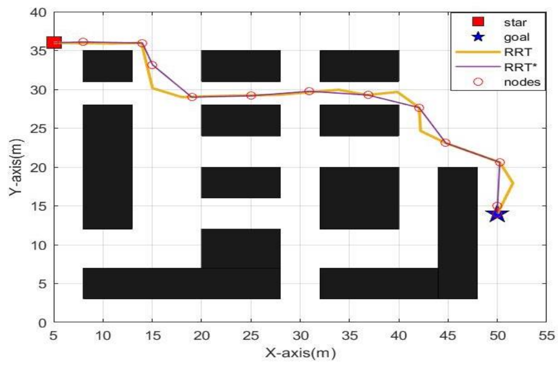
(a)



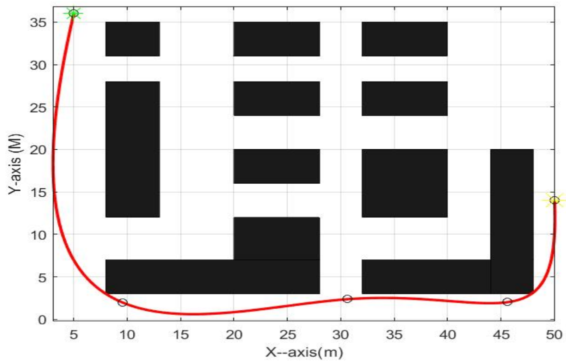
(b)

Figure. 18 (a) The linear velocity of the right and left wheels of the mobile robot and (b) The angular velocity of the right and left wheels of the mobile robot

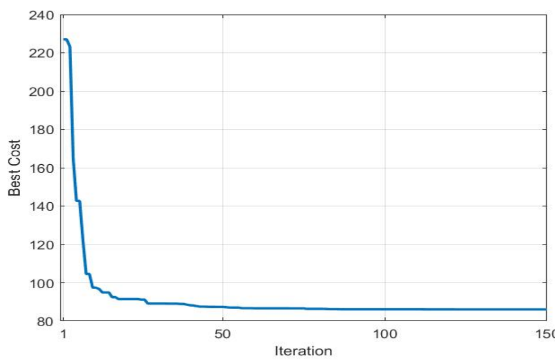
A comparison with other algorithms was conducted to verify our suggested hybrid algorithm. RRT, RRT*, PSO, and our proposed algorithm were compared with the fuzzy analytic hierarchy process and A* algorithm (A*–FAHP) hybrid path planning algorithm [31]. The A*–FAHP used a stationary warehouse setting with no change in the working space, where the environment workspace is [60×40] m, the starting point is at (5, 36), and the goal point is at (50, 14). This environment was applied in RRT and RRT* algorithms to find the best path for the mobile robot, as shown in Fig. 19. The PSO algorithm was also used to obtain the best path, as shown in Fig. 19(b) and (c).



(a)



(b)

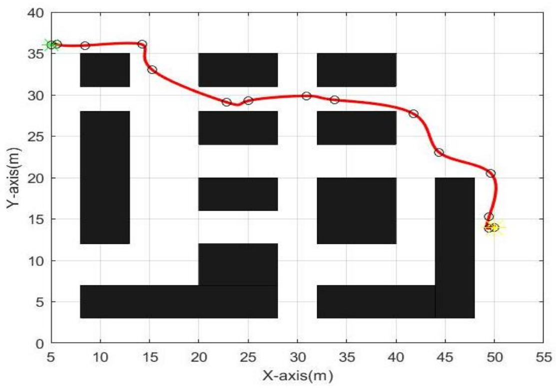


(c)

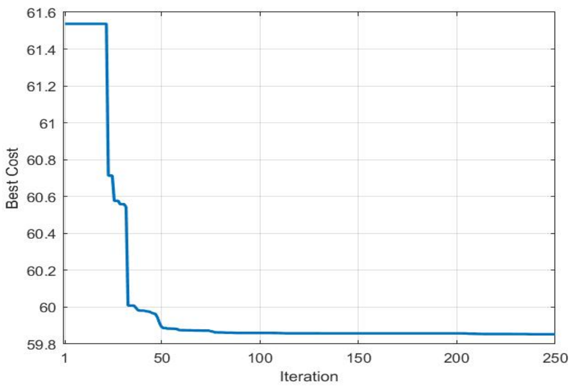
Figure. 19 The environment of [31] using (a) (RRT&RRT*) algorithm, (b) (PSO) algorithm, and (c) the best cost function of the PSO algorithm

 DOI: 10.22266/ijies2022.0831.16

Received: March 29, 2022. Revised: April 26, 2022.



(a)



(b)

Figure. 20 The environment of [31] using (a) the proposed RRT*PSO algorithm and (b) the best cost function

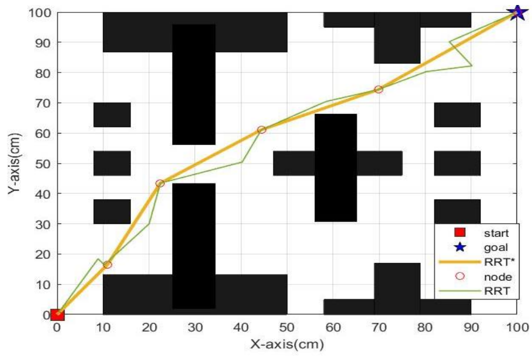
While Fig. 20(a) and (b) show the proposed hybrid technique of the RRT*PSO algorithm, which is used to find the smooth and optimal path for the mobile robot.

The second comparison was performed with the of coevolution-based particle swarm optimization method (SAEGBPSO) [32], where the environment workspace is [100×100] m, the starting point is at (0,0), and the goal point is at (100,100).

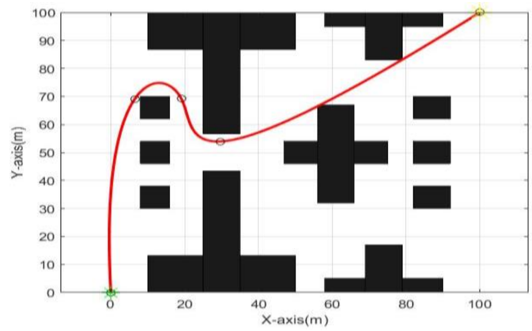
This environment was applied in RRT and RRT* algorithm, as shown in Fig. 21(a), and in the PSO algorithm, as shown in Fig. 21(b) and (c). While, Fig. 22(a) and (b) show the proposed RRT*PSO algorithm to find the optimal path.

All the results of the comparison with the two methods are presented in Table 4.

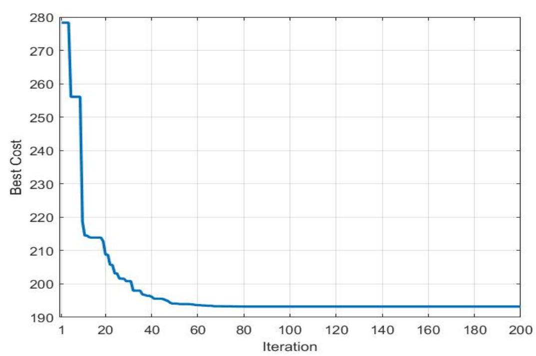
Related to Ref. [31], the results in Table 4 show that the RRT* produced a shorter and collision-free path compared to A*, FAHP, and A*-FAHP. At the same manner, the proposed hybrid algorithm (RRT*PSO) produced a shorter and collision-free path compared to the RRT*. This is due to that the (RRT*PSO) creates a short path between the start and the goal nodes because RRT* used fewer nodes in the workspace. At the same time, RRT* has steering function from node to node made the reaching to the goal node became easier. Later, the PSO will adjust the path to be smoother. While, the A* used a large number of nodes to generate a short path with no collision and high computation time moreover FAHP followed the goal node via a reactive way. The FAHP fail to create the optimal path to the goal node unless that has all information about the workspace. Related to Ref. [32], (RRT*PSO) produce a shorter path than (SAEGBPSO). This method depends self-adaptive strategy in which updating the main control parameters (position and velocity information) of particles based on the EGT and the iteration number of the algorithm. Conversely, (RRT*PSO) depends on the limited number of particles (nodes) which is created by RRT*.



(a)



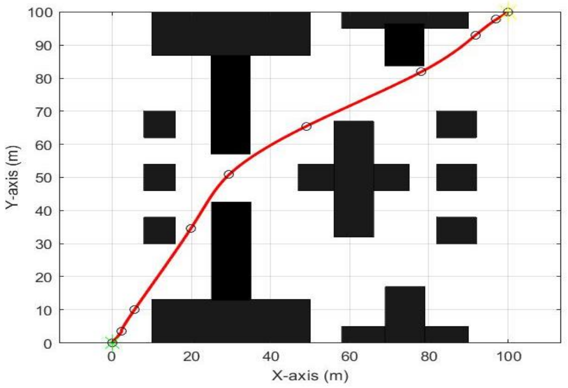
(b)



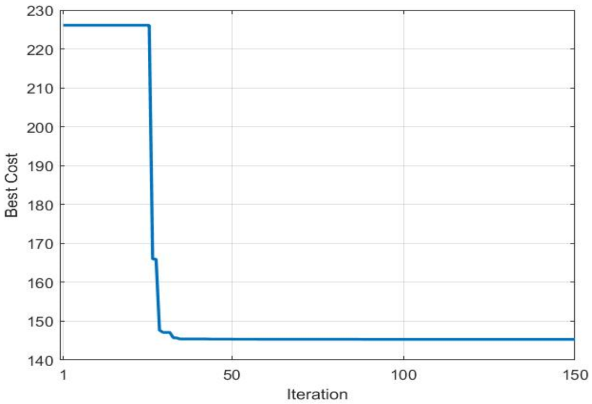
(c)

Figure. 21 The environment of [32] using (a) (RRT&RRT*) algorithm, (b) (PSO) algorithm, and (c) the best cost function of the PSO algorithm

Received: March 29, 2022. Revised: April 26, 2022.



(a)



(b)

Figure. 22 The environment of [32] using (a) the proposed RRT*PSO algorithm and (b) the best cost function

Table. 4 Comparisons of the optimal path distance with [31 and 32]

Environment	Algorithm	Path distance (m)	Minimum iteration
Stationary environment [31]	[31] A*	87.94	-
	[31] FAHP	106.36	
	[31]A*–FAHP	87.54	
	RRT	62.2	-
	RRT*	59.03	-
	PSO	88.1	117
	our proposed hybrid RRT*PSO	57.8	75
First numerical simulation on single-robot path planning [32]	[32]FACPSO	146.50	130
	[32]CIGA	150.95	60
	[32]SAEGBPSO	145.51	150
	[32]SPSO 2011	181.54	175
	RRT	151.23	-
	RRT*	145.5	-
	PSO	193	127
	our proposed hybrid RRT*PSO	145	50

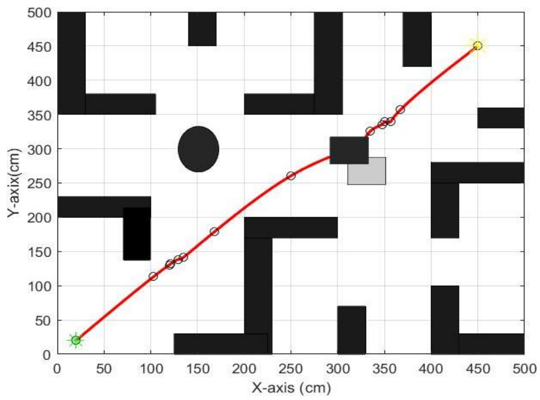


Figure. 23 Movement of the obstacle in the environment

5.2 The second scenario (dynamic environment)

The second scenario in the dynamic environment used moveable obstacles in the workspace depending on the proposed equations as shown below:

$$x_{p(new)} = x_{p(old)} + Vobs \times cos\Theta \times Ts \quad (14)$$

$$y_{p(new)} = y_{p(old)} + Vobs \times sin\Theta \times Ts \quad (15)$$

Where p is the position, $Vobs$ is the velocity of the obstacle, Θ is the angle of movement of the obstacle, and Ts is the sampling time. Fig. 23 shows the movement of the obstacle from the position (310,250) (gray color) to the position (290,280) (black color).

To avoid the collision between the wheeled mobile robot and the dynamic obstacle, the proposed algorithm (flowchart) represented in Fig.11 was used. The simulation result produced a new path started from (20, 20) to (230,232.4), then started from (231,227.7.) to (295,224.8) and (396,386.5) to (450,450). The optimal path distance is 658.5 cm, as shown in Fig. 24(a). While Fig. 24(b) shows the cost function in iteration number 60 with a maxi iterations' number of 100 iterations.

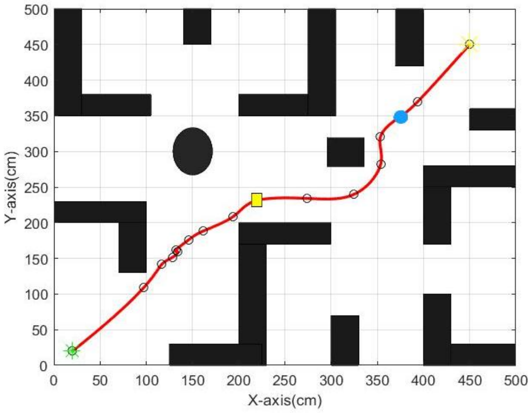
The reference path equation for the optimal path of the hybrid RRT*PSO in a dynamic environment can be dived into three parts as follows:

Part 1: from positions (20, 20) to (230,232.4) represented in Eq. (6):

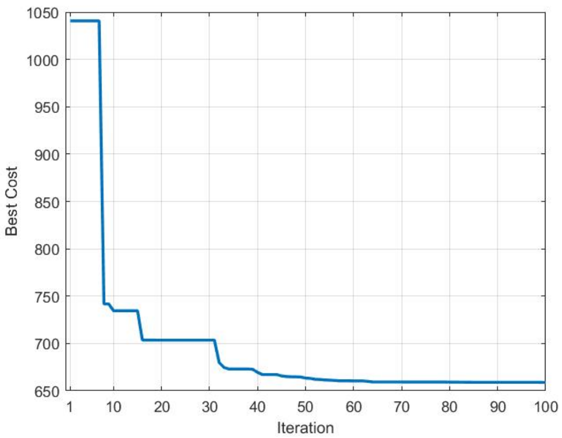
Part 2: from positions (231,227.7.) to (295,224.8) represented in Eq. (16).

$$y_{ref}\left(x_{ref}\right) = 5e-05 \times x^3_{ref} -0.037 \times x^2_{ref} + 8.9902 \times x^1_{-ref} - 490.99 \quad (16)$$

Part 3: from positions (396,386.5) to (450,450) represented in Eq. (6).



(a)



(b)

Figure. 24 The optimal path simulation using the hybrid (RRT*PSO) (a) path planning and (b) the best cost function.

Fig. 25 shows the optimal path generated based on the proposed RRT*PSO hybrid algorithm, while Fig. 26 demonstrates the cart mobile robot's reference linear and angular velocities.

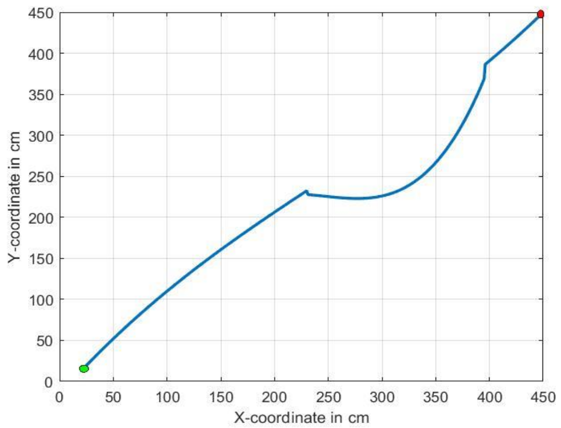


Figure. 25 Optimal path generated based on the proposed RRT*PSO hybrid algorithm

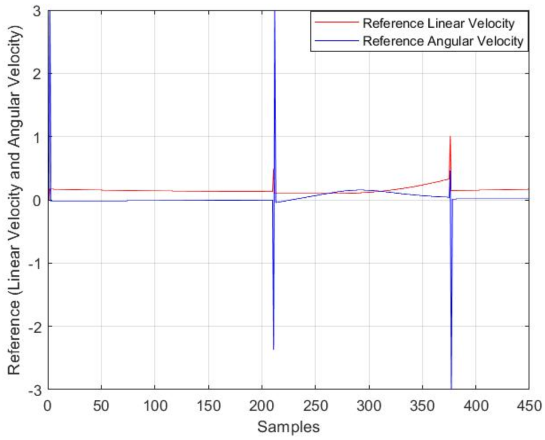
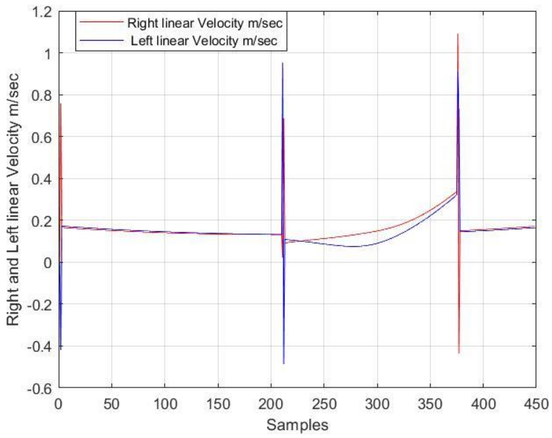
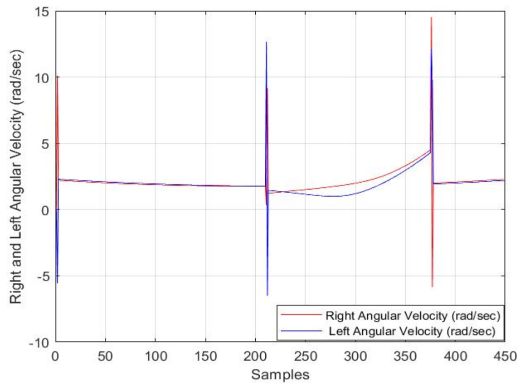


Figure. 26 The reference linear and angular velocities of the cart mobile robot

Furthermore, Fig. 27(a) represents the linear velocity of the right and left wheels of the cart mobile robot, whereas Fig. 27(b) represents the angular velocity of the cart mobile robot is right and left wheels.



(a)



(b)

Figure. 27 (a) linear velocity of the right and left wheels of the mobile robot and (b) the angular velocity of the right and left wheels of the mobile robot

Received: March 29, 2022. Revised: April 26, 2022.

6. Conclusions

Utilizing the sampling-based (RRT*) algorithm and the (PSO) heuristic algorithm, a hybrid (RRT*PSO) algorithm was proposed in this work. This hybrid method was applied in static and dynamic environments to achieve the smoothest and the shortest path for the mobile robot and to avoid obstacles by decreasing the number of iterations, the function of evaluation, and the execution time of the processor unit during producing the desired path. In a static environment, a comparison study with other works was made. It has been found that the RRT*PSO algorithm provides enhancement on the path distance equals 34.27% compared to the A* algorithm and the fuzzy analytic hierarchy process (A*–FAHP hybrid) algorithm and 0.35% compared to the self-adaptive evolutionary game-based particle swarm optimization (SAEGBPSO). The dynamic environment is considered as complex as the real-world environment because obstacles are not static in the moving-obstacle path planning. The main problem of implementing the RRT*PSO algorithm is related to the determination of the collisions of particles with all moving obstacles, which leads to difficulties in handling constraints of the moving-obstacle path planning problem. To handle this difficulty in this work, a new algorithm named (contour path down and contour path up) was suggested to avoid a collision with the dynamic obstacles and to create a collision-free path. In this regard, a great achievement was obtained with respect to the optimal path distance and the cost function. For the future work, we will implement experimental works of the proposed path-planning algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Ammar Abdul Ameer Rasheed, Ahmed Sabah Al-Araji and Mohammed Najm Abdullah improved and developed an intelligent hybrid path-planning technique for the mobile robot system. Ammar Abdul Ameer Rasheed explained the proposed algorithm for path planning. Ahmed Sabah Al-Araji described the kinematics of the mobile robot. Ammar Abdul Ameer Rasheed and Mohammed Najm Abdullah discussed the simulation results of this work.

References

[1] X. Zhong, J. Tian, H. Hu, and X. Peng, "Hybrid Path Planning Based on Safe A* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment", *Journal of Intelligent & Robotic Systems*, Vol. 99, No. 1, pp. 65-77, 2020.

[2] I. Sung, B. Choi, and P. Nielsen, "On the training of a neural network for online path planning with offline path planning algorithms", *International Journal of Information Management*, Vol. 57, pp. 102-142, 2021.

[3] 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.

[4] A. Maoudj and A. Hentout, "Optimal Path Planning Approach based on Q-Learning Algorithm for Mobile Robots", *Applied Soft Computing*, Vol. 97, pp. 106-796, 2020.

[5] L. Bai and C. Du, "Design and Simulation of a Collision-free Path Planning Algorithm for Mobile Robots Based on Improved Ant Colony Optimization", *Ingénierie des Systèmes d'Information*, Vol. 24, No. 3, pp. 331-336, 2019.

[6] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized Voronoi diagram based efficient heuristic path planning method for RRTs in mobile robots", *IEEE Transactions on Industrial Electronics*, DOI 10.1109/TIE.2021.3078390, 2021.

[7] 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.

[8] L. Schmid, M. Pantic, R. Khanna, L. Ott, R. Siegwart, and J. Nieto, "An Efficient Sampling-based Method for Online Informative Path Planning in Unknown Environments", *IEEE Robotics and Automation*, Vol. 5, No. 2, pp. 500-1507, 2020.

[9] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph Neural Networks for Decentralized Multi-Robot Path Planning", In: *Proc. of International Conference on Intelligent Robots and Systems (IROS)*, pp. 1785-1792, 2020.

[10] E. N. Sabudin, R. Omar, S. K. Debnath, and M. S. Sulong, "Efficient Robotic Path Planning Algorithm based on Artificial Potential Field", *International Journal of Electrical & Computer Engineering*, Vol. 11, No. 6, pp. 4840-4849, 2021.

[11] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-Objective Path Planning of an Autonomous Mobile Robot Using Hybrid

PSO-MFB Optimization Algorithm", *Applied Soft Computing*, Vol. 89, p. 106076, 2020.

[12] A. AbuBaker and Y. Ghadi, "Mobile robot Controller Using Novel Hybrid System", *International Journal of Electrical & Computer Engineering*, Vol. 10, No. 1, pp. 1027-1034, 2020.

[13] 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, doi: 10.22266/ijies2021.0630.20.

[14] S. K. Das, A. K. Dutta, and S. K. Debnath, "Operative Critical Point Bug Algorithm-Local Path Planning of Mobile Robot Avoiding Obstacles", In: *Proc. of International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, pp. 1-8, 2019.

[15] D. Li, W. Yin, W. E. Wong, M. Jian, and M. Chau, "Quality-Oriented Hybrid Path Planning based on A* and Q-Learning for Unmanned Aerial Vehicle", *IEEE Access*, Vol. 10, pp. 7664-7674, 2022.

[16] 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, doi: 10.22266/ijies2021.0831.24.

[17] I. Rustam, N. M. Tahir, A. I. M. Yassin, N. Wahid, and A. H. Kassim, "Linear Differential Driven Wheel Mobile Robot Based on MPU9250 and Optical Encoder", *TEM Journal*, Vol. 11, No. 1, pp. 30-36, 2022.

[18] 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 third Scientific Conference of Electrical Engineering (SCEE)*, pp. 84-89, 2018.

[19] 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.

[20] L. Zhang, Y. Zhang, and Y. Li, "Path Planning for Indoor Mobile Robot Based on Deep Learning", *International Journal for Light and Electron Optics*, Vol. 219, p. 165096, 2020.

[21] S. Li, D. Zhao, Y. Sun, J. Yang, and S. Wang, "Path Planning Algorithm Based on the Improved RRT-Connect for Home Service Robot Arms", In: *Proc. of International Conference on Intelligence and Safety for Robotics (ISR)*, pp. 403-407, 2021.

[22] Y. Gao, T. Hu, Y. Wang, and Y. Zhang, "Research on the Path Planning Algorithm of Mobile Robot", In: *Proc. of 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 447-450, 2021.

[23] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A Survey of Path Planning Algorithms for Mobile Robots", *Vehicles*, Vol. 3, No. 3, pp. 448-468, 2021.

[24] J. Wang, W. Chi, C. Li, C. Wang, and M. Meng, "Neural RRT*: Learning-Based Optimal Path Planning", *IEEE Transactions on Automation Science and Engineering*, Vol. 17, No. 4, pp. 1748-1758, 2020.

[25] R. H. A. Rubayi, M. K. Abd, and F. M. F. Flaih, "A New Enhancement on PSO Algorithm for Combined Economic-Emission Load Dispatch Issues", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 1, pp. 77-85, 2020, doi: 10.22266/ijies2020.0229.08.

[26] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient Path Planning for UAV Formation via Comprehensively Improved Particle Swarm Optimization", *ISA Transactions*, Vol. 97, pp. 415-430, 2020.

[27] J. Lian, W. Yu, K. Xiao, and W. Liu, "Cubic Spline Interpolation-based Robot Path Planning Using a Chaotic Adaptive Particle Swarm Optimization Algorithm", *Mathematical Problems in Engineering*, Vol. 2020, pp. 1-20, 2020.

[28] X. Liu, D. Zhang, J. Zhang, T. Zhang, and H. Zhu, "A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm", *Cluster Computing*, Vol. 24, No. 3, pp. 1901-1915, 2021.

[29] S. Q. G. Haddad and A. R. Akkar, "Intelligent Swarm Algorithms for Optimizing Nonlinear sliding mode controller for robot manipulator", *International Journal of Electrical & Computer Engineering*, Vol. 11, No. 5, pp. 3943-3955, 2021.

[30] M. N. Abdullah and K. E. Dagher, "Airborne Computer System Path-Tracking Based Multi-PID-PSO Controller Design", *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 3, pp. 403-411, 2021, doi: 10.22266/ijies2021.1231.22.

[31] C. Kim, J. Suh, and J. Han, "Development of A Hybrid Path Planning Algorithm and A bio-

Received: March 29, 2022. Revised: April 26, 2022.

inspired Control for an Omni-wheel Mobile Robot", *Sensors*, Vol. 20, No. 15, pp. 4258-4278, 2020.

[32] B. Tang, K. Xiang, M. Pang, and Z. Zhanxia, "Multi-robot path planning using an improved self-adaptive particle swarm optimization", *International Journal of Advanced Robotic Systems*, Vol. 17, No. 5, pp. 1-19, 2020.