Research Article

Trajectory Planning and Collision Control of a Mobile Robot: A Penalty-Based PSO Approach

Krishna Kant Pandey (ID),[1] **Chandrashekhar Kumbhar,**[2] **Dayal R. Parhi,**[3]
Sandeep Kumar Mathivanan,[4] **Prabhu Jayagopal,**[4] **and Aminul Haque** (ID)[5]

[1]*Department of Mechatronics Engineering, Manipal University Jaipur, Jaipur 303007, Rajasthan, India*
[2]*Learning Facilitator, Educlass Pvt. Ltd., Singapore 408601, Singapore*
[3]*Mechanical Engineering Department, National Institute of Technology, Rourkela, Odisha 769008, India*
[4]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India*
[5]*Department of Computer Science and Engineering, Daffodil International University, Daffodil Smart City, Birulia, Savar, Dhaka 1216, Bangladesh*

Correspondence should be addressed to Aminul Haque; aminul.cse@daffodilvarsity.edu.bd

In this paper, trajectory planning and navigation control problems have been addressed for a mobile robot. To achieve the objective of the research, an adaptive PSO (Particle Swarm Optimization) motion algorithm is developed using a penalty-based methodology. To deliver the best or collision-free position to the robot, fitness values of the all-random-positioned particles are compared at the same time during the target search action. By comparing the fitness values, the robot occupies the best position in the search space till it reaches the target. The new work integrated with conventional PSO is varying a velocity event that plays a vital role during the position acquisition (continuous change in position during the obstacle negotiation with the communication through random-positioned particles). The obstacle-negotiating angle and positional velocity of the robot are considered as input parameters of the controller whereas the robot's best position according to the target position is considered as the output of the controller. Simulation results are presented through the MATLAB environment. To validate simulation results, real-time experiments have been conducted in a similar workspace. The results of the adaptive PSO technique are also compared with the results of the existing navigational techniques. Improvements in results between the proposed navigation technique and existing navigation techniques are found to be 4.66% and 11.30%, respectively.

1. Introduction

In robotics science, navigation and trajectory planning of mobile robots (MRs) using artificial intelligence (AI) approaches are the most common type of research domain. During trajectory planning in the search space, the wheeled mobile robot moves from a source point to the target point by avoiding obstacles present in an arena. Simultaneously, the robot creates a collision-free trajectory in a search space using artificial intelligence (AI) based robot controller. The following control objectives must be followed during the development of any AI-based navigational controller. For the model, the controller should provide good stability (the

signal must be bounded). The controller should have good tracking ability (good understanding) and the controller should be robust (take a self-decision during navigational failures) in nature. In the next section, a detailed review of the literature has been presented addressing strategy, design, and implementation of navigation controllers for wheeled mobile robots in different environments.

Yang et al. [1] developed a layered motion planning scheme for the navigation of a wheeled MR in an unstructured environment using the Fuzzy technique. To produce the intermediate way-point towards the target, the first layer of the planner uses the information about the global end position and the high-range sensory data. Here,

no crucial assumptions have been used regarding the environment. A path-following approach using rule-based fuzzy logic that mimics human behaviour has been proposed by Antonelli et al. [2] for a smart vehicle. Knowledge of the next bend ahead to the vehicle is taken as the fuzzy input data and linear velocity by which the robot can safely move on the path and it is the output of the fuzzy controller. The line following behaviour is executed by the vehicle to validate the proposed algorithm. Gueaieb and Miah [3] developed an intelligent and innovative nonvision-based navigation approach using Radio Frequency Identification (RFID) technology. In their work, only straight path motion has been considered. To mimic human behaviour, they have used single input and single output Mamdani-type fuzzy architecture. Mastrogiovanni et al. [4] developed a new mobile robot navigation approach called "μNav." They said, during path planning in a complicated search space, this technique requires a minimum sensory data, less computational power, and memory. Since it gains robustness intrinsically from the μNav technique, it does not require any self-localization potentiality or inboard geometrical representation. To explore the basic geometrical features of indoor search space and to survive there, a miniaturized triangulation laser scanner-based control architecture has been developed by [5] (using a swarm of the robot). The objectives of the research are to explore the environment using simple, small, low cost, and low power small devices. A fuzzy logic based [6] control algorithm has been trained using an evolutionary-group-based PSO for navigation in an unknown world. The mutation and crossover operation of the fuzzy-based PSO has used group-based framework. The adaptive velocity altered action is presented to enhance the search capability. Takagi–Sugeno–Kang (TSK) type FLC has been used for the navigational analysis. Chou et al. [7] have explored the indoor navigation problems using the dynamic window approach (DWA*) for navigation in an unfamiliar environment. The local reactive method is used to categorize the environment by which the robot achieves smoothness, speed, and local minima-free navigation. A region study technique has been applied to eliminate inadequate commands. To determine the optimal command, the A* algorithm is applied with look-ahead verification. Using the tethered coverage (TC) analysis, Shnaps and Rimon [8] emphasized the strategies of MR motion planning in an unknown scenario. They have deployed a mobile robot of size "D" in the fixed pose "S" connected through the cable of length "L." A novel adaptive localization algorithm [9] performs the estimation of the robot position with great accuracy in an amorphous environment. To estimate the position of the robot, they used image fusion of an omnidirectional visualization system, odometry measurements, and inertial devices (sensors). To decide the robot's orientation and the robot velocity, odometer and inertial sensors obtained the feature points using the sequence of an image. Using an adaptive fusion (AF) based ACO and PSO (AF-ACPSO) [10], cooperative navigation strategies have been developed for two robots in an unspecified environment. During the navigation, AF-ACPSO-based FLC has been executed the boundary following behaviour and the robot

learn this behaviour. The learning behaviours are completed by a single robot and further applied to the second robot that works as a follower. Ramaithitima et al. [11] proposed the work for navigation control of the MR using a swarm of inexpensive portable sensors. In their work, the robot does not require explicitly metric information during the generation of maps. They have used the Voronoi graph to create an approximate map of the covered environment. The ROS (Robot Operating System) is used to demonstrate the algorithm. Almasri et al. [12] developed a technique for the robotic system, by which the mobile robot follows the line and avoid collision with obstacles. They have used low-cost IR sensors. This methodology includes a satisfactory level of calculation. Hence in real-time applications, it has been efficiently implemented. They have used the e-puck robot on the Webots platform to check the simulation results. Golan et al. [13] proposed an online robot navigation system using artificial temperature gradients techniques. The environment with obstacles is taken as a "hot" junction and the target is taken as a "cold" junction. During the navigation, the temperature gradient is solved using a heat conduction partial differentiation equation. Kanezaki et al. [14] developed a reactive neural network approach to tackle the problem of learning. The author's key concept is to crop, resize, and rotate an obstacle map which is based on the target location, and to represent the map better, the agent's current pose is taken rather than the layout of the search space. They have supplied navigation history as input in the robot brain to reduce the rate of failure. Ataka et al. [15] developed an RMF (Reactive Magnetic Field) inspired navigation method in the unspecified 3D convex environment. The robot induced artificial electric current in the obstacle exterior and the MF (magnetic field) guides the robot along with the obstacles surrounding. Hence, the robot does not suffer from the local minima problem in convex obstacles-based 3D environment. Cole and Wickenheiser [16] proposed a path-planning algorithm for multiple mobile robots using a reactive trajectory scheme. The authors stated the route does not disrupt the robot thrust limitation and sensor limitations for the moving obstacles. They used the Sigmoid function for the transition course. Sensor data are updated by matching with sigmoid slopes and curves. Singh et al. [17] developed a vision-based navigational approach for a nonholonomic MR in an unfamiliar known environment. They used switching-based SMC (sliding mode control) methodology for the analysis by which the robot continuously follows the desired route. The robot uses red, green, and blue depth (RGB-D) sensor modules to create angular velocity. In their work, fuzzy logic is used for guidance and the Krasovski method is used to show the asymptotically stable curve. To reference MR applications for the urban search and rescue (USAR), Niroui et al. [18] proposed a deep reinforcement learning strategy for a cluttered environment. The authors combined the Asynchronous Advantage Actor-Critic (A3C) and reinforcement learning to allow the mobile robot to navigate safely. A3C is a technique that accelerates optimization as well as the training process parallelly and maintains the navigational policy.

Due to incomplete information of a mimic and unregular environment, the robot confronts complexity and vagueness during navigation and path planning. It may be that the robot is trapped in the loop due to mimic surrounding inside the environment. Traditional navigational and path-planning approaches such as the Visibility graph [19], Voronoi diagram [20], and Grids [21] are not compatible with navigation and path planning in the mimic and unregular environments. Recently, many researchers have developed various navigation algorithms [22–26] but these methods have still some drawbacks or could not fulfill the desire. These navigational problems are taken as objectives in this analysis. In this research, the mobile robot is using a novel hybrid optimization technique to perform navigation and path-planning tasks. As well as the above problem is expressed as an optimization problem having constraints. The results are compared with other navigational techniques in terms of path length to check the performance of the technique. Using the proposed technique, the robot smartly negotiates obstacles and navigates towards the target. Finally, the mobile robot performs well in the given environment (mimic or unregular environment) and the results are recorded in terms of trajectory length and navigational time.

2. The Proposed Work: Adaptive PSO Concept

It is a population-based randomly determined optimization technique and devoted for artificial life phenomenon. The APSO is inspired by the swarm behaviour of insects, birds, and fishes. It is well suited for continuous variable and global search problems. It is effectively applied to a varied series of problems such as network [27], structural optimization [28], and fuzzy control system [29]. In APSO, the system makes it ready to change with the population of a random value and searches for the best by updating the local generation. The APSO technique updates the population of agents according to the values of the fitness function. The fitness function depends upon the swarm behaviour of local particles.

In this analysis, the PSO has been modified using a local search (LS) ability. The constraints are accommodated to obtain better obstacle-negotiation results using the penalty method. Using local search ability, position as well as control

parameters of agents are tuned perfectly. Consequently, the navigational angle performance has been increasing in terms of position accuracy (best fitness).

2.1. Adaptive Particle Swarm Optimization (APSO) Methodology. Adaptive PSO algorithm always learns from the situations and utilizes it to optimize the problem state. In APSO, all the agents have some fitness values (P_{best}) but global best (G_{best}) is the solution for all agents, which is surrounded by a flock of agents. The solution to the local search phenomenon is the "footprint between agents moving towards the global best position" in the search space. Furthermore, fitness value has been obtained using fitness calculation criteria. A solution with better fitness value during the local search event is selected as the finest result and appointed as the global best. The agent holds some varying properties like velocity, acceleration, and position when trying to finding out the global best in the environment. As a result, the global best of PSO may be changed as the global best of local search PSO. If agents are moving near to the global best, they sequentially follow the updated optimum position of the agent and it may be based on local search criteria. Figure 1 represents the flowchart of the APSO technique for trajectory planning and navigation control.

In Figure 2, the basic operations of APSO have been presented. The "x" is an agent positioned randomly in the search space ("x" personal best is $p_{ij}^n(t)$) and searching for global best with time "t" to "$t+1$" and so on. In each sequence of iteration, every particle is occupying a new position by following two 'best' solutions. The initial best value for all solutions is stored as "P_{best}" (personal best position of an agent). Another value that has been traced by the optimizer is the second-best solution and called a global position (g_{best}). It is stored as a global best (g_{best}) and followed by all agents. "P_{best}" and "g_{best}" are the agent's personal and global positions, respectively. However, agents are following the global best value and updated their position according to that with varying velocity. By updating velocity ($v_{ij}^n(t)$) and position ($x_{ij}^n(t)$) using equations (1) and (2), the agents update their personal best value (P_{best}) and global best value (g_{best}).

$$x_{ij}^n(t+1) = x_{ij}^n(t) + v_{ij}^n(t+1), \tag{1}$$

$$v_{ij}^n(t+1) = wv_{ij}^n(t) + r_{i_1}C_1\left[\left\{p_{ij}^n(t)\right\} - x_{ij}^n(t)\right] + r_{i_2}C_2\left[\left\{g^n(t)\right\} - x_{ij}^n(t)\right]. \tag{2}$$

In equations (1) and (2), "w" is inertia weight and C_1 and C_2 are the positive acceleration coefficients [30]. The uniformly distributed random variables (UDRV) are given as 'r_{i_1}' and 'r_{i_2}.' The costs of UDRV are varied between '0' and '1.' The velocity and position of the "ith" particle are denoted as $x_i = (x_i^1, x_i^2, \ldots\ldots x_i^n)$ and $v_i = (v_i^1, v_i^2, \ldots\ldots v_i^n)$. From equations (3) and (4), the best earlier location '$P_{ij}{}^n(t)$' and

the global best earlier location '$g^n(t)$' of particles in the swarm are chosen.

$$P_{ij}^n(t) = \left\{P_{ij}^1(t), P_{ij}^2(t), P_{ij}^3(t)\ldots\ldots P_{ij}^n(t)\right\}, \tag{3}$$

$$g^n(t) = \left\{g^1(t), g^2(t), g^3(t)\ldots\ldots g^n(t)\right\}, \tag{4}$$

Figure 1: Flowchart of the APSO technique.

Figure 2: Principle diagram of APSO.

Sl. no.	Properties	Values
1	Population size (i)	100
2	"+" acceleration coefficient (C_1)	1.496
3	"+" acceleration coefficient (C_2)	1.496
4	Alpha	0.1000
5	Iteration	100
6	Max iteration	100
7	η_{pop}	100
8	η_{var}	3
9	Inertia weight (w)	0.7298
10	Penalty	1000

universal least of the cost function. Each searcher is expected to know its individual best earlier location and the global best earlier location. Also, each seeker communicates with the agent whose position is overlapped with obstacles but is near to the goal position. If this condition is rising, then the penalty function (using equation (5)) added is with the overlapped agents to separate from the obstacles. The penalty function is taken into account to reduce the navigation time as well as to obtain a smooth path. Table 1 shows the different properties, which are used by the adaptive controller for tuning the navigational path. Furthermore, particles have been accommodated to create a smooth path from the start to the goal position. The adaptive PSO has been analyzed to create optimal and collision-free navigation in an environment using local search methodology.

The negotiation within a threshold range has been performed between obstacles and the robot to obtain the collision-free navigational path. In the next section, the architecture of the controller (APSO) has been demonstrated to tune the navigation path by adding a penalty with colliding agents.

2.2. The Architecture of APSO Controller. In Figure 3, a flowchart of the APSO algorithm is shown. By adding a penalty function with the PSO algorithm, the advanced algorithm has been proposed as APSO for trajectory planning, robot navigation, and path optimization. The disorder environments have been taken for the navigation and path-planning analysis in which obstacles shapes are different in size as well as positioned with different orientations. Using APSO "G_{best}" has been calculated and compared with the fitness value of local search "G_{best}" (penalty-based G_{best}). Then, the preferred "G_{best}" from both events (PSO and penalty-based PSO) is appointed as the next possible location and the best location is updated in the robot brain. Hence, the best position vector and velocity vector of the robot have been updated from point to point during the optimized trajectory formation.

Using the adaptive PSO controller, the mobile robot also minimizes decision response time on the operational platform. It is due to local search criteria in APSO. The social and cognitive behaviour of the PSO method has been embedded with the concept of adaptive (local) search to optimize the response time, navigation time, and trajectory length. The robot can perform tasks such as avoiding obstacles, learn

$$^{P}A_{P} = P_{ij}^{n}(t) + \rho(f). \tag{5}$$

Equations (3) and (4) can be construed in the following way. Let us assume "n" searchers act as "n" elements stirring in the search environment "X," the locus of the "n^{th}" searcher in the "i^{th}" reiteration is signified as "$x_{ij}^{n} \in X_{search-space} \subset \mathbb{R}^2$." The cost function "$f: \mathbb{R}^2 \longrightarrow \mathbb{R}$" sustained by each searcher in the adverse of the signal strength established at its existing position [31]. The aim of the searchers is to connect between "n" numbers of agents and interchange in a manner to verify and occupy the
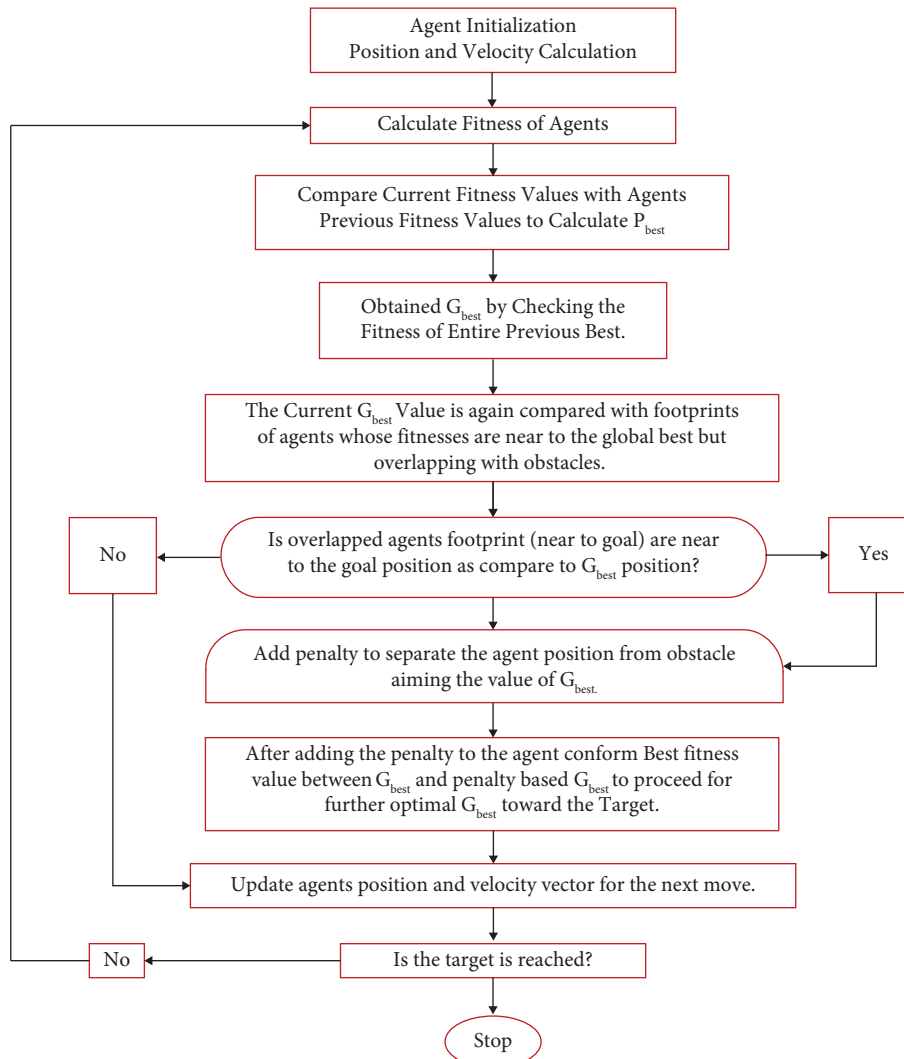
FIGURE 3: Flowchart of the adaptive PSO algorithm.

trajectory, comparing sensors results, and position optimized (aiming the goal) online. The results are analyzed in terms of path cost and path deviation for different environments. The robot reaction time, navigation time, and path length are minimized during the goal search inside the search space. Finally, the robot reaches the goal point smoothly. In the upcoming section, results and discussion using the adaptive PSO algorithm have been presented.

2.3. Stability Analysis in the Mimic and Maze Environment.
In this section, results from simulation and real-time experiment have been discussed for the adaptive PSO approach. The results are shown in terms of trajectory length and navigational time. The height and width of the simulation and experimental environments are taken as "10 × 10." For that, "100 cm" is equal to "10" for simulation and experimental analysis. Two types of simulation environments (Figures 4 and 5) have been created in the MATLAB platform to check the navigational results using the APSO technique. For the real-time experiments, a similar type of

environment (Figures 6 and 7) is taken as compared to the simulation environment. The Khepera II mobile robot has been used in the real-time experiment to validate the simulation results as well as the developed approach. The navigational axis of rotation (obstacle-negotiation angle) has been optimized in a way the robot creates a smooth collision-free trajectory and minimizes navigational time. The robot simultaneously updates its steering angle and positional map using the APSO technique. As a result, the robot has performed an effective exploration work in the given environment.

2.4. Experimental Analysis in a Simulation Platform.
In Figures 4 and 5, trajectory planning and navigational results are shown using the APSO technique. The first simulation environment (Figure 4) contains only wall-type obstacles whereas, in the second simulation environment (Figure 5), wall type and irregular obstacles shape are considered.

The robot start position is taken as 0.7 m in the X-direction and 0.05 m in the Y-direction for both of the
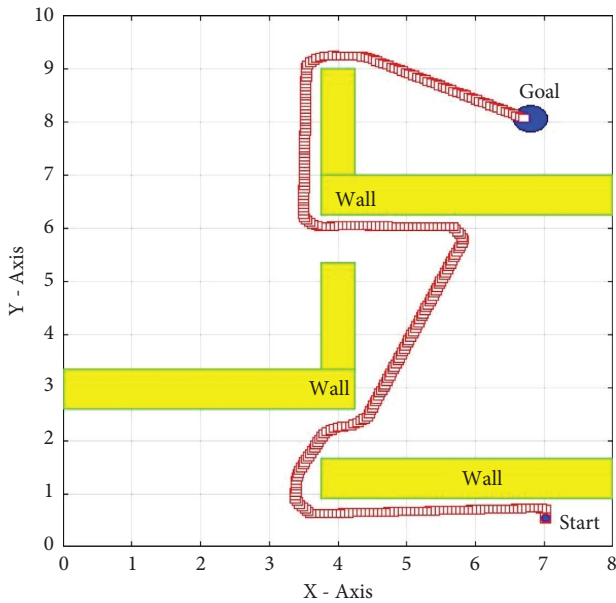
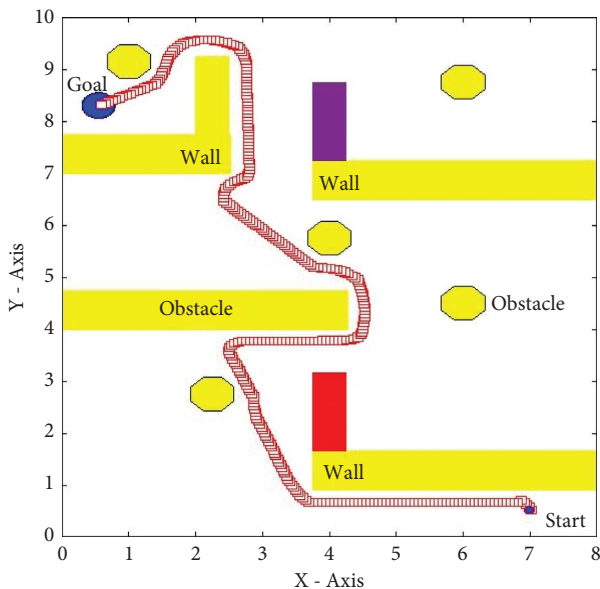FIGURE 4: The mobile robot control analysis in a mimic environment.

FIGURE 5: The mobile robot control analysis in a maze environment.

environments. The target position is 0.675 m in the X-direction, and 0.8 m in the Y-direction has been taken for the first simulation experiment. Similarly, the target position is 0.05 m in the X-direction, and 0.825 m in the Y-direction has been taken for the second simulation experiment. In these environments, the robot navigates in the narrow corridor as well as avoids mimic type wall configurations smoothly. Finally, the robot reaches the target efficiently in both of the environments by negotiating the obstacles using APSO. The time taken and path length obtained during the navigation are tabulated in Table 2.

2.5. Experimental Analysis in a Real-Time Platform Using Khepera-II—A Mobile Robot. The real-time experiments (Figures 6 and 7) are conducted using a Khepera II mobile robot, and the results are presented in terms of trajectory length (TL) and navigational time (NT) in Table 3. In Figures 6 and 7, the robot starts to point and obstacles position, and the target position are plotted similarly to the simulation environment. The robot successfully reached the target by negotiating the obstacles in the real-time environment.

The deviation in results has been taken in terms of trajectory length and navigational time that are represented in Table 4. Performance and effectiveness of the adaptive PSO algorithm are confirmed by considering the minimum deviation in results between both experiments (Simulation and Real-time). The percentage of deviation in both cases (Time and Trajectory length) is recorded as less than 8.5%.

Due to the use of the best fitness value from the APSO algorithm and the prediction of the next possible position, the PSO algorithm gets smarter. Hence, a smooth and collision-free navigational path has been created from the start to the target position. The robot detects its possible position accurately inside the environment due to penalty-based methodology and local search. The APSO AI technique reduced the trajectory map errors as well as position error (in "X," "Y," and "θ" directions) due to local search and penalty criteria. The wheel slippage is reduced up to 80% in the narrow corridor due to varying velocity events during the obstacle negotiation.

The proposed controller takes minimum reaction time to read the next obstacle-negotiation angle subjecting to the target path. Using the proposed APSO algorithm, the robot does not trap in the loop, or the robot smartly avoids the loop of the search space. Finally, the robot achieved the target without any trouble in an irregular environment. The operational performance and effectiveness of the proposed navigation controller are confirmed by comparing it with other navigational techniques in the next section.

3. Simulation Analysis in a Dynamic Environment

3.1. A Comparative Study between Proposed Algorithm and Existing Algorithm [32] and [33]. APSO navigational results (Table 5) have been compared with ACO with Fuzzy (Garcia [32]) and Fuzzy Logic (Yahmedi and Fatmi [33]) which are already developed. To check the effectiveness of the proposed algorithm, the trajectory length has been considered and compared with other developed techniques. The trajectory length using Simple Ant Colony Optimization distance memory (SACOdm) [32] is compared with the trajectory length obtained using the APSO algorithm (Figure 8). It is found that the APSO technique holds good results as compared to SACOdm in terms of trajectory length (Table 5). The improvement in trajectory length using the current approach is recorded by 5% as compared to the 'SACOdm' trajectory length.
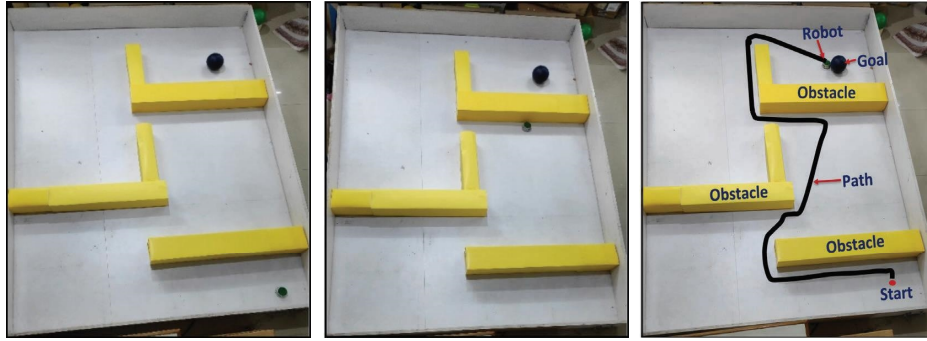
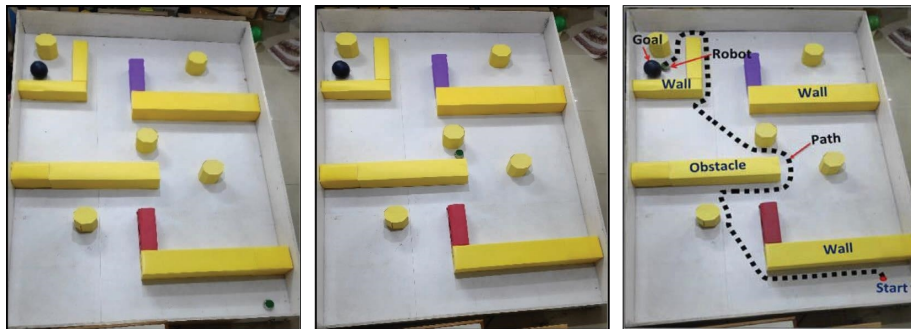FIGURE 6: Real-time experiment using the APSO algorithm to validate the simulation results (Figure 4).

FIGURE 7: Real-time experiment using the APSO algorithm to validate the simulation results (Figure 5).

TABLE 2: Time taken and path length data for the simulation experiment (Figures 4 and 5).

Sl. no.	Scenario	Time ("s")	Trajectory length ("m")
1.	Figure 4	21.36	1.627
2.	Figure 5	27.48	1.763

TABLE 3: Time taken and trajectory length data for the real-time experiment (Figures 6 and 7).

Sl. no.	Scenario	NT ("s")	TL ("m")
1.	Figure 6	23	1.76
2.	Figure 7	30	1.92

TABLE 4: Results comparison between simulation and experiment data using APSO (Figures 4–7).

| Sl. no.	Scenario	NT (in second "±0.5")	"%" deviation	TL ("m" "±0.02")	"%" deviation
1.	Figure 4	21.36	7.13	1.627	7.55
	Figure 5	23		1.76	
2.	Figure 6	27.48	8.4	1.763	8.17
	Figure 7	30		1.92	

Similarly, the second comparison has been made between 'Fuzzy logic' (Yahmedi and Fatmi [33]) and the APSO algorithm in terms of trajectory length (Figure 9). Yahmedi and Fatmi used individual behaviour and action coordination technology with two-layered control. As compared to Yahmedi and Fatmi [33] results in terms of path length, the proposed technique holds improved results by 11% (Table 5).

The simulation and experimental study in various environments are conducted using the proposed methodology (APSO) and results are tabulated in Table 6. To confirm the robustness and effectiveness of the proposed methodology during the trajectory planning, the average percentage deviation study between simulation and experimental results are depicted in Table 6. The average deviation in results is confirmed as 5.53 in terms of navigational time (NT) and 5.28 in terms of trajectory length (TL). In this analysis, the average deviation is less than 6% (Table 6) in perspective studies (NT and TL) and improvement in navigational results are greater than 4.5 percent (Table 5) is found. Hence, based upon depicted data in this analysis, the proposed methodology is robust as compared to existing techniques.

TABLE 5: Trajectory length (TL) comparison between APSO and existing techniques [32, 33].

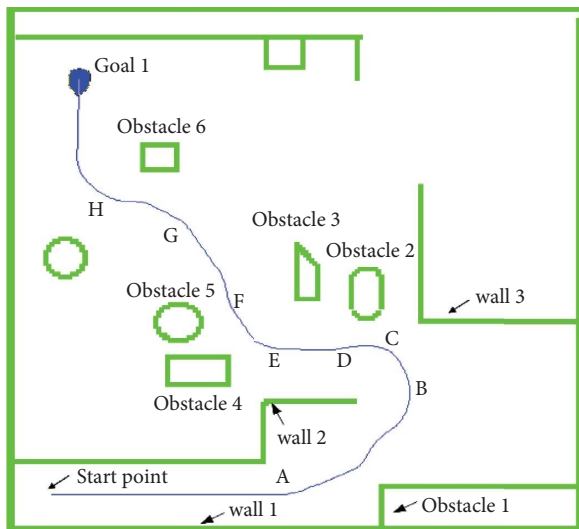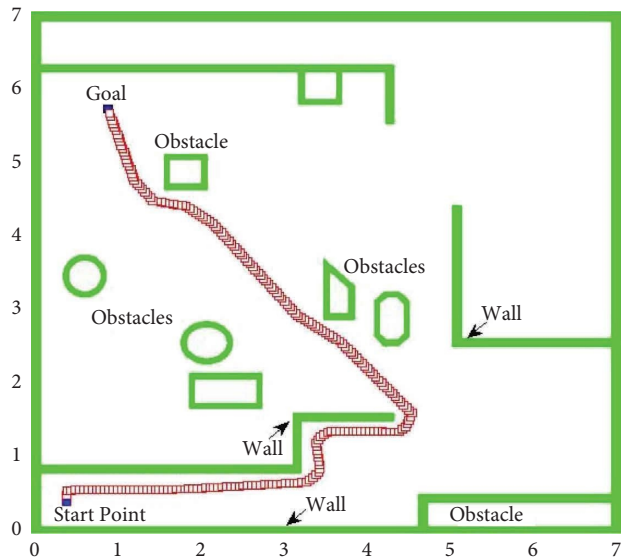| Sl. no.	Tasks	TL	Improvement (%)
01.	Figure 8(a), Garcia, [32]	03.43 m	4.66
	Figure 8(b), proposed technique	03.27 m	
02.	Figure 9(a), Yahmedi and Fatmi, [33]	24.75%	10.30
	Figure 9(b), proposed technique	22.20%	
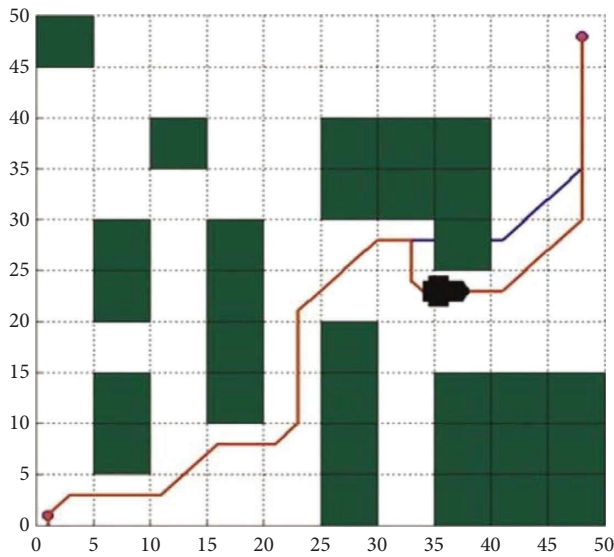
(a)
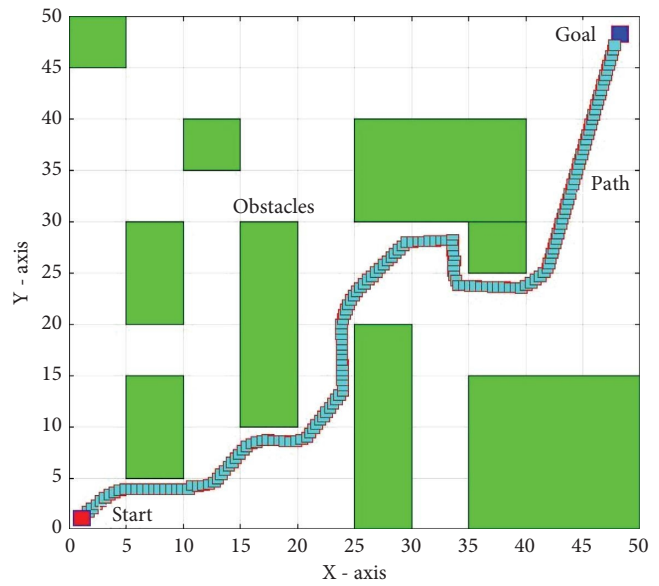
(b)

FIGURE 8: Results comparisons. (a) SACOdm [32]. (b) APSO technique (current analysis).

(a)

(b)

FIGURE 9: Results comparisons. (a) Fuzzy Logic [33]. (b) APSO technique (current analysis).

TABLE 6: Study of simulation and experimental data in the various maze environment.

Sl. No.	Scenario	NT (in second "±0.5")	"%" deviation	TL ("m" "±0.02")	"%" deviation
1.	Simulation-1	22.13	5.43	1.78	5.32
	Experiment-1	23.40		1.88	
2.	Simulation-2	28.18	5.87	1.82	5.70
	Experiment-2	29.94		1.93	
3.	Simulation-3	21.09	5.76	1.59	4.40
	Experiment-3	22.38		1.66	
4.	Simulation-4	26.38	4.70	1.71	4.09
	Experiment-4	27.62		1.78	
5.	Simulation-5	24.56	6.14	1.67	5.38
	Experiment-5	26.07		1.76	
6.	Simulation-6	20.28	6.75	1.53	5.22
	Experiment-6	21.65		1.61	
7.	Simulation-7	20.71	5.98	1.57	5.09
	Experiment-7	21.95		1.65	
8.	Simulation-8	21.13	5.20	1.62	4.94
	Experiment-8	22.23		1.70	
9.	Simulation-9	27.53	4.54	1.79	5.58
	Experiment-9	28.78		1.89	
10.	Simulation-10	28.43	4.92	1.81	6.07
	Experiment-10	29.83		1.90	
	Average deviation in results		5.53		5.28

4. Conclusions

In this paper, a penalty-based adaptive PSO approach has been developed for the trajectory planning of mobile robot in the wall-type maze environment. The agent overlapped with obstacles is taken under consideration to tune the trajectory path using local search and penalty-based method. To improve the robot position, navigational time, trajectory lengths, and response time penalty-based APSO analysis has been proposed. The agents near to the goal but overlapped with obstacles are added with a penalty to separate from a position of obstacles. Using APSO methodology, the advancement in trajectory planning (10% improvement) has been achieved in this analysis. The performance and effectiveness of the algorithm are validated by comparing the simulation and real-time experimental results. The deviation in results between simulation and real-time experiment has been recorded within 9% in terms of path length and navigational time. To check the authenticity of the proposed technique, its results in terms of the trajectory length are compared with existing techniques. The improvements in the results are recorded by 4% and 10.5%, as compared to existing techniques. The applications of the proposed methodology can be used for AI techniques based on decision-making applications. In future work, we will try to implement in real-time unstructured dynamics environment.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] X. Yang, M. Moallem, and R. V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1214–1224, 2005.

[2] G. Antonelli, S. Chiaverini, and G. Fusco, "A fuzzy-logic-based approach for mobile robot path tracking," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 2, pp. 211–221, 2007.

[3] W. Gueaieb and M. S. Miah, "An intelligent mobile robot navigation technique using RFID technology," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 9, pp. 1908–1917, 2008.

[4] F. Mastrogiovanni, A. Sgorbissa, and R. Zaccaria, "Robust navigation in an unknown environment with minimal sensing and representation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 212–229, 2009.

[5] G. Fu, P. Corradi, A. Menciassi, and P. Dario, "An integrated triangulation laser scanner for obstacle detection of miniature mobile robots in indoor environment," *IEEE/ASME Transactions On Mechatronics*, vol. 16, no. 4, pp. 778–783, 2011.

[6] C. F. Juang and Y. C. Chang, "Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 379–392, 2011.

[7] C. C. Chou, F. L. Lian, and C. C. Wang, "Characterizing indoor environment for robot navigation using velocity space

approach with region analysis and look-ahead verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 442–451, 2011.

[8] I. Shnaps and E. Rimon, "Online coverage by a tethered autonomous mobile robot in planar unknown environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 966–974, 2014.

[9] L. Li, Y. H. Liu, K. Wang, and M. Fang, "Estimating position of mobile robots from omnidirectional vision using an adaptive algorithm," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1633–1646, 2015.

[10] C. F. Juang, M. G. Lai, and W. T. Zeng, "Evolutionary fuzzy control and navigation for two wheeled robots cooperatively carrying an object in unknown environments," *IEEE Transactions on Cybernetics*, vol. 45, no. 9, pp. 1731–1743, 2015.

[11] R. Ramaithitima, M. Whitzer, S. Bhattacharya, and V. Kumar, "Automated creation of topological maps in unknown environments using a swarm of resource-constrained robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 746–753, 2016.

[12] M. M. Almasri, A. M. Alajlan, and K. M. Elleithy, "Trajectory planning and collision avoidance algorithm for mobile robotics system," *IEEE Sensors Journal*, vol. 16, no. 12, pp. 5021–5028, 2016.

[13] Y. Golan, S. Edelman, A. Shapiro, and E. Rimon, "Online robot navigation using continuously updated artificial temperature gradients," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1280–1287, 2017.

[14] A. Kanezaki, J. Nitta, and Y. Sasaki, "Goselo: goal-directed obstacle and self-location map for robot navigation using reactive neural networks," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 696–703, 2018.

[15] A. Ataka, H. K. Lam, and K. Althoefer, "Reactive magnetic-field-inspired navigation method for robots in unknown convex 3-D environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3583–3590, 2018.

[16] K. Cole and A. M. Wickenheiser, "Reactive trajectory generation for multiple vehicles in unknown environments with wind disturbances," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1333–1348, 2018.

[17] P. Singh, P. Agrawal, H. Karki, A. Shukla, N. K. Verma, and L. Behera, "Vision based guidance and switching based sliding mode controller for a mobile robot in the cyber physical framework," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 1985–1997, 2019.

[18] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[19] J. S. Mitchell, "An algorithmic approach to some problems in terrain navigation," *Artificial Intelligence*, vol. 37, no. 1-3, pp. 171–201, 1988.

[20] N. Kalra, D. Ferguson, and A. Stentz, "Incremental reconstruction of generalized Voronoi diagrams on grids," *Robotics and Autonomous Systems*, vol. 57, no. 2, pp. 123–128, 2009.

[21] M. Weigl, B. Siemiåatkowska, K. A. Sikorski, and A. Borkowski, "Grid-based mapping for autonomous mobile robot," *Robotics and Autonomous Systems*, vol. 11, no. 1, pp. 13–21, 1993.

[22] P. Y. Zhang, T. S. Lü, and L. B. Song, "Soccer robot path planning based on the artificial potential field approach with simulated annealing," *Robotica*, vol. 22, no. 5, pp. 563–566, 2004.

[23] Y. C. Chang and Y. Yamamoto, "On-line path planning strategy integrated with collision and dead-lock avoidance schemes for wheeled mobile robot in indoor environments," *Industrial Robot: International Journal*, vol. 35, no. 5, pp. 421–434, 2008.

[24] P. K. Mohanty and D. R. Parhi, "Navigation of autonomous mobile robot using adaptive network based fuzzy inference system," *Journal of Mechanical Science and Technology*, vol. 28, no. 7, pp. 2861–2868, 2014.

[25] D. R. Parhi, B. B. V. L. Deepak, J. Mohana, R. Ruppa, and M. Nayak, "Immunised navigational controller for mobile robot navigation," in *Soft Computing Techniques in Vision Science*, pp. 171–182, Springer, Heidelberg, Germany, 2012.

[26] A. Pandey, S. Pandey, and D. R. Parhi, "Mobile robot navigation and obstacle avoidance techniques: a review," *International Robotics & Automation Journal*, vol. 2, no. 3, Article ID 00022, pp. 1–12, 2017.

[27] S. Goudarzi, W. H. Hassan, M. H. Anisi et al., "ABC-PSO for vertical handover in heterogeneous wireless networks," *Neurocomputing*, vol. 256, pp. 63–81, 2017.

[28] M. Zawidzki and J. Szklarski, "Transformations of arm-Z modular manipulator with particle swarm optimization," *Advances in Engineering Software*, vol. 126, pp. 147–160, 2018.

[29] R. Martinez-Soto, O. Castillo, and L. T. Aguilar, "Type-1 and Type-2 fuzzy logic controller design using a Hybrid PSO–GA optimization method," *Information Sciences*, vol. 285, pp. 35–49, 2014.

[30] B. Song, Z. Wang, and L. Zou, "On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm," *Cognitive Computation*, vol. 9, no. 1, pp. 5–17, 2017.

[31] R. Zou, V. Kalivarapu, E. Winer, J. Oliver, and S. Bhattacharya, "Particle swarm optimization-based source seeking," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 865–875, 2015.

[32] M. P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing*, vol. 9, no. 3, pp. 1102–1110, 2009.

[33] S. Amur, A. Yahmedi, and M. A. Fatmi, "Fuzzy logic based navigation of mobile robots," *Recent Advances in Mobile Robotics*, pp. 1–12, Intech Open, London, UK, 2011.