

## Research Article

# A New Online Random Particles Optimization Algorithm for Mobile Robot Path Planning in Dynamic Environments

Behrang Mohajer,<sup>1</sup> Kouros Kiani,<sup>2</sup> Ehsan Samiei,<sup>3</sup> and Mostafa Sharifi<sup>2</sup>

<sup>1</sup> Department of Mechanical Engineering, Sharif University of Technology, Tehran 3567-11365, Iran

<sup>2</sup> Department of Electrical and Computer Engineering, Semnan University, Semnan 35131-19111, Iran

<sup>3</sup> Department of Electrical Engineering, Amirkabir University of Technology (AUT), Tehran 15875-4413, Iran

Correspondence should be addressed to Kouros Kiani; [kouros.kiani@semnan.ac.ir](mailto:kouros.kiani@semnan.ac.ir) and Mostafa Sharifi; [mostafa.sharifi88@gmail.com](mailto:mostafa.sharifi88@gmail.com)

Received 24 September 2012; Revised 30 January 2013; Accepted 30 January 2013

Academic Editor: Alexander P. Seyranian

Copyright © 2013 Behrang Mohajer et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new algorithm named random particle optimization algorithm (RPOA) for local path planning problem of mobile robots in dynamic and unknown environments is proposed. The new algorithm inspired from bacterial foraging technique is based on particles which are randomly distributed around a robot. These particles search the optimal path toward the target position while avoiding the moving obstacles by getting help from the robot's sensors. The criterion of optimal path selection relies on the particles distance to target and Gaussian cost function assign to detected obstacles. Then, a high level decision making strategy will decide to select best mobile robot path among the proceeded particles, and finally a low level decision control provides a control signal for control of considered holonomic mobile robot. This process is implemented without requirement to tuning algorithm or complex calculation, and furthermore, it is independent from gradient base methods such as heuristic (artificial potential field) methods. Therefore, in this paper, the problem of local mobile path planning is free from getting stuck in local minima and is easy computed. To evaluate the proposed algorithm, some simulations in three various scenarios are performed and results are compared by the artificial potential field.

## 1. Introduction

Application of mobile robots in industries, military activities, and indoor usages has been remarkably growing recently. In addition, in some missions such as hazardous or hardly accessible territorial, where humans could be in danger, mobile robots can carry out tasks such as nuclear plants, chemical handling, and rescue operations.

One of the most challenging subjects in mobile robot's research is path planning. The path planning problem is usually defined as follows. Given a robot and a description of an environment, plan a path between two specific locations. The path must be collision-free (feasible) and satisfy certain optimization criteria minimum traveling distance [1]. Researchers have developed and used different concepts to solve the robot path planning problem. These concepts have been categorized into the following subclasses: the

graphical approaches, the classical techniques and the heuristic approaches as well as the use of the potential field methods.

Graphical methods are more often than not developed on the principles of geometry and they are dominantly known for their limitation to static environment [2]. Graphical techniques such as voronoi diagram, spatial planning, transformed space, cell decomposition, and configuration space have been proposed and extensively used in the past few decades. One of the well-known graphical approaches is the Configuration Space method used by [3–10].

A second class of approaches which has also been widely studied is the use of classical algorithm such as gradient descent, steepest descent, dynamic programming, iterative technique, randomized methods, and Kino dynamic [11–20].

Another class of path planning approaches is the heuristic-based technique. Path planning in this context consists of every state as a possible position for a robot to

move to and the transitions between these states are the actions the robot takes from the known information of the surrounding environment. The information that is valued the most by the algorithm is the cost of transitions from the current state to the goal state. The path taken is optimal if the sum of transition costs which are labeled global costs, also known as edge costs, is a global minimal. Global costs consist of all known paths for the robot at a specific time. The algorithm also has to be complete. This technique is based on inferential and evolutionary computation. There has been several heuristic techniques in literatures such as potential field, virtual force,  $A^*$  algorithm,  $E^*$  algorithm, genetic algorithm, memetic algorithm, and simulated annealing.

Artificial potential fields (APFs) for collision avoidance have attracted many researchers in the field of robotics. The idea of virtual forces acting on a robot has been proposed by Andrews and Hogan [21] and Khatib [22]. This method is based on the construction of a potential function around the obstacles, deriving repulsive forces to the robot to avoid them; on the contrary, an attractive potential function was defined with its center at the end position. In these approaches the target applies an attractive force to the robot while obstacles exert repulsive forces on the robot. The resultant force (sum of all forces) determines the direction and speed of movement. A generalized potential field method that combines global and local path planning has been introduced by Krogh and Thorpe [23].

Path planning on other hand can be divided into two groups according to the assumptions about the information available for planning. In path planning with complete information perfect knowledge about the robot and the environment are considered, shapes of obstacles are computed algebraically, and path planning is implemented offline. In path planning with incomplete information or sensor-based planning, the obstacles can be of arbitrary shape and the input information is, in general, local information from a range detector or a vision sensor [24]. Path planning could be either local or global. Local path planning means that path planning is done while the robot is moving; in other words, the algorithm is capable of producing a new path in response to environmental changes [25]. On the other, global path planning requires that environment to be completely known and all the features present within the terrain remain static and the algorithm generates a complete path from the start point to the destination point before the robot starts its motion.

In this paper, a local path planning problem in an unknown environment for a holonomic mobile robot in the presence of mobile obstacles is addressed. In order to face this problem several approaches have been proposed in literatures. One of the most general techniques is based on the utilization of artificial potential fields (APFs). As mentioned above, this concept was introduced by Khatib [22], as a real-time obstacle avoidance method. The major drawback of this approach is that the robot may get stuck in local minima. To cope with this problem, Vadakkepat et al. introduced a new technique named evolutionary artificial potential field (EAPF) to address the path planning scenario with mobile obstacles. Genetic algorithm was employed to derive

optimal potential field functions. In addition, an escape-force algorithm was performed to avoid being trapped in the local minima [26]. Ge and Cui [27] proposed a new potential field method for motion planning of a mobile robot in a dynamic environment. The new potential functions take into account not only the relative positions of the robot with respect to the target and obstacles, but also the relative velocities of the robot with respect to the target and obstacles. Then, Poty et al. [28] merged the approach proposed in [27] and the fractional potential for dynamic motion planning of mobile robot. The fractional potential was utilized to characterize danger zone and risk coefficient of each obstacle. Computer simulations demonstrated that mobile robot avoided obstacles and reached the target successfully. Recently, Munasinghe et al. [29] proposed the velocity dipole field and its integration with the conventional potential field to form a new real-time obstacle avoidance algorithm. Unlike the radial field lines of conventional potential field, the velocity dipole field has elliptical field lines that navigate a robot more skillfully. It is useful to skillfully guide the robot around obstacles, quite similar to the way humans avoid moving obstacles. Conn and Kam in [30] proposed a method where the mobile obstacles can be considered as stationary in the extended world by assuming time as one of the dimensions. However, this method required the mobile obstacle trajectory to be known which is not applicable in real world.

The organization of this paper is as follows. In Section 2, properties and assumptions of path planning in this proposal are described. Section 3 introduces the proposed RPO algorithm for local path planning of mobile robots. Simulation and numerical results are provided in Section 4. Finally, in Section 5 we draw some conclusions and discuss about the proposed path planning method.

## 2. System Modeling and Problem Formulation

In path planning problem of mobile robots, obstacle avoidance and target seeking are two fundamental behaviors that must be controlled. On the other hand, path planning of mobile robots in the unknown environments with fixed/moving target is one of the difficulties in this area. As shown in Figure 1, the robot is controlled to track fixed/moving target while avoiding the moving obstacle. The obstacles and the target are moving arbitrarily during the mission. In addition, moving obstacles positions are also unknown unless they are within the robot's sensor detection range. Sensor detection range is defined according to allowed sensory distance to obstacles. A Gaussian function is assigned to each detected obstacle. Furthermore, in mobile robotics, the most common approach is to assume that the robot is holonomic (omnidirectional) and by further assumption it is simply a point, so the configuration space is reduced to a 2D representation with  $x, y$  axis. Since we have reduced the robot to a point, we must inflate each obstacle by the size of the robot's radius to compensate.

Differential equation of robot's motion is described by  $\dot{q}(t) = u(t)$ , where  $u \in R^2$  is control input and  $q(t) = [x(t), y(t)]^T \in R^2$  is the robot's position at time  $t$  and

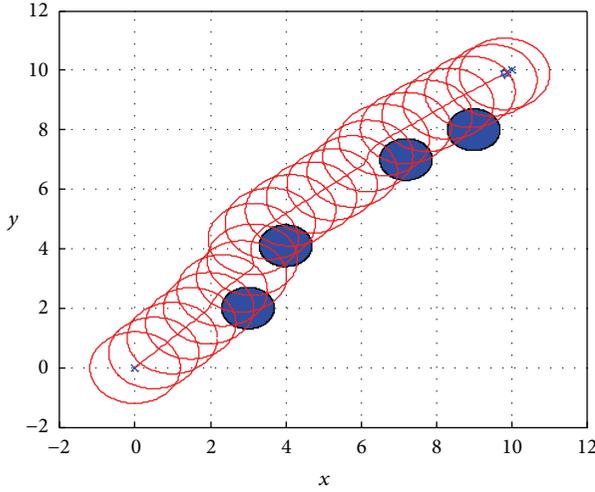


FIGURE 1: The proposed RPO algorithm in fixed obstacle and target.

assumed to be known all along the mission, and  $q_0(t_0) = [x_0(t), y_0(t)]^T \in R^2$  is the robot's initial position. The target position is also defined by  $p_G(t) = [x_G(t), y_G(t)]^T \in R^2$ .

### 3. Robot Path Planning in Dynamic Environments

In this section, an algorithm inspired from bacterial foraging mechanism is designed for local path planning of a mobile robot in dynamic environment. First, we briefly review a bacterial foraging technique and then the random particle algorithm is proposed.

**3.1. Bacterial Foraging Optimization Algorithm.** Bacterial foraging theory is based on the assumption that animals search for and obtain nutrients in a way that maximizes their energy intake  $E$  per unit time  $T$  spent foraging [31]. Hence, they attempt to maximize their long-term average rate of energy intake.

The main target in this algorithm for each bacterium is to find the minimum  $J(\theta)$ ,  $\theta \in R^p$  without considering the gradient,  $\nabla J(\theta)$ , where  $\theta$  is the position of a bacterium, and  $J(\theta)$  is an attractant-repellent profile from the environment.  $J(\theta) < 0$ ,  $J(\theta) = 0$ , and  $J(\theta) > 0$  represent that the bacterium at location  $\theta$  is in nutrient rich, neutral, and noxious environments, respectively. In fact in this algorithm we deal with a nongradient optimization algorithm. Chemotaxis is a foraging behavior that implements a type of optimization where bacteria try to climb up the nutrient concentration (i.e., lower and lower values of  $J(\theta)$ ) and avoids being at positions  $\theta$  where  $J(\theta) \geq 0$  [1].

According to this algorithm,  $i$ th bacterium at the position  $\theta$  takes a chemotaxis step  $j$  with the step size  $c(i)$  in the random direction and calculate the cost function  $J(\theta)$  at each step. If the cost function of the new position  $\theta(i, j+1)$  that is,  $J(\theta(i, j+1))$ , is smaller than the  $J(\theta(i, j))$ , then another step size  $c(i)$  in the same previous direction will be taken. This process in the direction of lower cost function will

be continued until the maximum number of steps ( $N_s$ ) is reached. Then after each  $N_c$  chemotaxis step, the least healthy bacteria as stated by the cost function  $J(\theta(i, j))$  are replaced by the copies of healthy ones. This procedure is called reproduction step, and it is followed by the elimination-dispersal ( $N_{ed}$ ) event. For each elimination-dispersal event each bacterium in the population is subjected to elimination-dispersal with probability  $P_{ed}$ .

**3.2. The Random Particle Path Optimization Algorithm (RPOA).** The proposed random particle optimization (RPO) algorithm is based on particles randomly distributed around a robot. In initial position of mobile robot ( $q_0(t_0)$ ), ( $s = 1, 2, \dots, n$ ) virtual particles are generated and distributed randomly on a circle with radius of  $C(t)$  around the robot position. Each particle position ( $s$ ) in time  $t$  could be defined as  $(\theta_s(t) \in R^2)$  and the next position is calculated as

$$\theta_s(t + dt) = \theta_s(t) + C(t) \frac{\Delta(t)}{\|\Delta(t)\|}, \quad (1)$$

where  $\Delta(t) \in R^2$  is a unit length random vector which is used to define the direction particle in each time. These particles in time  $t + dt$  search the optimal path toward the target position while avoiding the moving obstacles by helping the robot's sensors. Then the best particle which has found the most optimal path among other ones is chosen and the mobile robot goes to the position of selected particle and it will continue again and again until the target is reached by the MR. For choosing the best particle, two different strategies are combined. Distance norm to target position and cost function  $J(\theta_s(t))$ ,  $\theta \in R^2$  which is an attractant-repellent profile from the environment and according to section two is defined as Gaussian function.

**3.2.1. Decision Making Strategy of Finding the Optimal Particle.** When the mobile robot arrives at the moving obstacles, its sensors detect obstacles. Then, the algorithm virtually assigns a repellent Gaussian cost function to the obstacle by the following formula:

$$J^{\text{obstc}} = \alpha^{\text{obstc}} \exp(-\mu^{\text{obstc}} \|\theta_i(t) - P_O(t)\|^2), \quad (2)$$

where  $\alpha^{\text{obstc}}$ ,  $\mu^{\text{obstc}}$  are constant values defining the height and the width of the repellent, respectively.  $P_O \in R^2$  denotes the obstacle position detected by the robot's sensor. Note that, during the algorithm, whenever an obstacle is within the robot's sensor range,  $J^{\text{obstc}}$  has a value unless otherwise its value is zero. In other words, repellent cost function all through the mission is defined as

$$J^{\text{obstc}} = \begin{cases} \alpha^{\text{obstc}} \exp(-\mu^{\text{obstc}} \|\theta_i(t) - P_O(t)\|^2) & \text{if } \|P_O(t) - q(t)\|^2 \leq \beta, \\ 0 & \text{if } \|P_O(t) - q(t)\|^2 > \beta. \end{cases} \quad (3)$$

In which,  $\beta$  is a robot's sensor range. Since the target position is always known, we also assign an attractant Gaussian

TABLE 1: Fixed algorithm parameter.

Number of bacteria	$s = 100$
Sensor detection range	$d^{\text{sensor}} = 1.2 \text{ m}$
Adaptive parameter	
Step size length which is considered fixed all through the mission	$C(t) = (1/12)d^{\text{sensor}}$

TABLE 2: Cost function parameter.

Height of repellent	$\alpha^{\text{obstc}} = 1$
Width of repellent	$\mu^{\text{obstc}} = 4$
Depth of attractant	$\alpha^{\text{goal}} = 1$
Width of attractant	$\mu^{\text{goal}} = 4$

TABLE 3: Obstacle, target, and initial positions.

First obstacle position	$[x, y]^T = [3, 2]^T$
Second obstacle position	$[x, y]^T = [9, 8]^T$
Third obstacle position	$[x, y]^T = [7.2, 7]^T$
Forth obstacle position	$[x, y]^T = [4, 4.1]^T$
Target position	$[x, y]^T = [10, 10]^T$
Initial position	$[x, y]^T = [0, 0]^T$

cost function to fixed/moving target position throughout the mission the same as what we did for the obstacles

$$J^{\text{goal}} = -\alpha^{\text{goal}} \exp(-\mu^{\text{goal}} \|\theta_i(t) - P_G\|^2), \quad (4)$$

where  $\alpha^{\text{goal}}, \mu^{\text{goal}}$  are constant values defining the height and the width of the attractant, respectively. Total cost function is defined as sum of repellent and attractant cost functions as

$$J = J^{\text{obstc}} + J^{\text{goal}}. \quad (5)$$

The objectives of path planning are to simultaneously (1) move the robot on the path that under the minimum time and cost arrive to the target while seeking it and (2) avoid robot to collide with unknown obstacles moving in the environment. These objectives have to be achieved in an unknown environment with online computation. In other words, this paper presents an online algorithm for path planning mobile robots in unknown environments without having prior information of obstacles and domain.

**3.2.2. High Level Decision Control.** High level decision control designed for path planning mobile robot is based on artificial random particles in which at time  $t \geq 0$  artificial particles are placed at the position of robot and at time  $t+dt \geq 0$  where  $dt$  is a small time, they are randomly located over the circle with radius  $C(t)$  around the mobile robot. In this one-step-ahead algorithm, the particle which could find the best path at time  $t+dt$  (whereas our objectives mentioned in Section 3.2 are satisfied) is selected as a best particle and robot with a low level decision control signal (it will be discussed at Section 3.2.3) will put on the point found by the best particle, and again from the new mobile position, a group of artificial

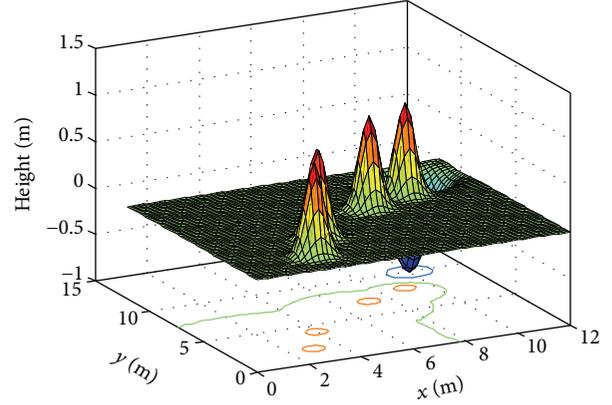


FIGURE 2: Gaussian cost function used by the algorithm for obstacles and target in RPO.

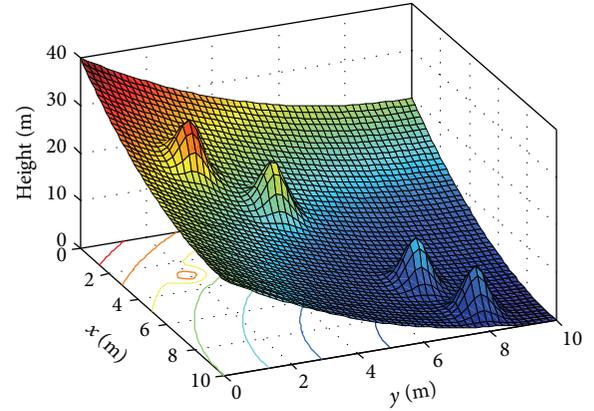


FIGURE 3: Cost function used by the algorithm for obstacles and target in APE.

particles are distributed and the process goes on again and again. Hence, artificial particles generated in the main CPU of mobile robot along with the information given from the robot sensors could search mobile surroundings and determine the robot's optimal path for one step ahead.

A decision making strategy designed for finding the best particle among the others is based on both distance error to target ( $e_s^d(t+dt)$ ) and cost function error  $e_s^J(t)$ . If  $d_s(t)$ ,  $d_s(t+dt)$  represent the distance from particle  $s$  to the target at time  $t$  and  $t+dt$ , and is defined as  $d_s(t) = \|\theta_s(t) - P_G(t)\|^2$ ,  $d_s(t+dt) = \|\theta_s(t+dt) - P_G(t)\|^2$ , respectively, each particle distance error and cost function error will be described as

$$\begin{aligned} e_s^d(t+dt) &= d_s(t+dt) - d_s(t), \quad s = 1, 2, \dots, n, \\ e_s^J(t) &= J(\theta_s(t+dt)) - J(\theta_s(t)), \quad s = 1, 2, \dots, n. \end{aligned} \quad (6)$$

In time  $t+dt$ , all the  $e_s^d(t)$ , for  $s = 1, 2, \dots, n$  are computed and then particles are sorted in order of ascending error  $e_s^d(t)$  in a vector  $S_{\text{sort}}$  such that the particle with the lowest distance error is at top. The first priority in this technique is to glide the robot to the target. Hence, particles are considered as fit ones that put on the higher part of vector  $S_{\text{sort}}$ . Furthermore,

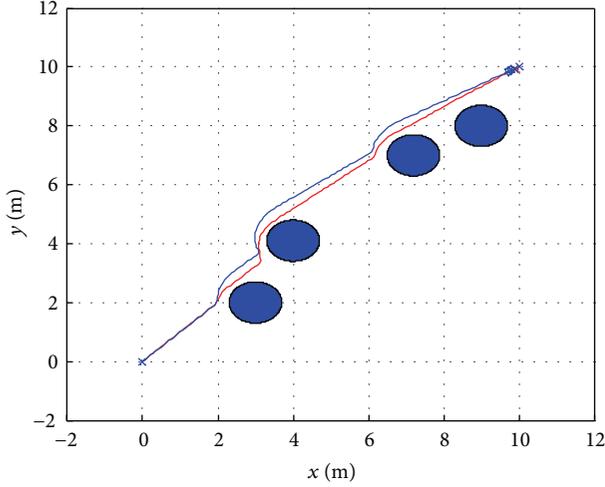


FIGURE 4: Fixed obstacle and goal position. (red) The proposed RPO algorithm (11.0321s), (blue) APF algorithm (11.2337 s).

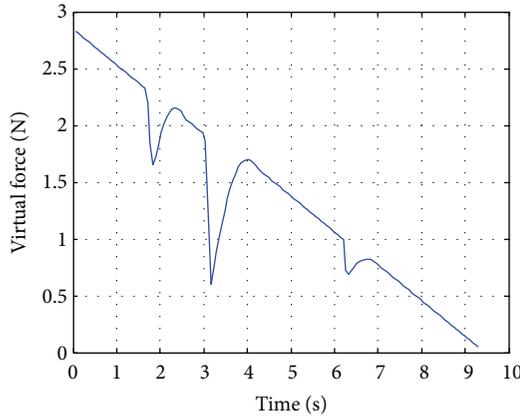


FIGURE 5: Exerted virtual force to the robot in APF method.

among these fit particles, one is chosen that has negative error cost function  $e_s^J(t)$  with priority to the lowest distance error, because they must be kept away from obstacles, and having  $e_s^J(t) > 0$  according to (3) represents that particles are in the region of obstacles. In fact, in this decision making strategy for finding the best particle, a search on distance error  $e_s^d(t)$  from top to bottom is performed and the first particle with  $e_s^J(t) < 0$  is selected as the best particle. By using these two techniques together, objectives are also satisfied.

Note that, according to (3)–(5) when an obstacle is not in the robot's sensor range,  $J^{\text{obstc}}$  is zero and as a result  $-2\alpha^{\text{goal}} < e_s^J(t) < 0$ . In this situation, robot freely moves toward target by choosing the first element of vector  $S_{\text{sort}}$  since all of them have  $e_s^J(t) < 0$ . But, when robot's sensor detect an obstacle,  $J^{\text{obstc}}$  gets a value and consequently,  $-2\alpha^{\text{goal}} < e_s^J(t) < 2\alpha^{\text{obstc}}$ . In this case, the first particle, in the ascended ordered of  $e_s^d(t)$ , which has the negative error  $e_s^J(t)$  is selected as the best one.

A modification could be performed on this strategy. In the above strategy, we have omitted particles with  $e_s^J(t) > 0$

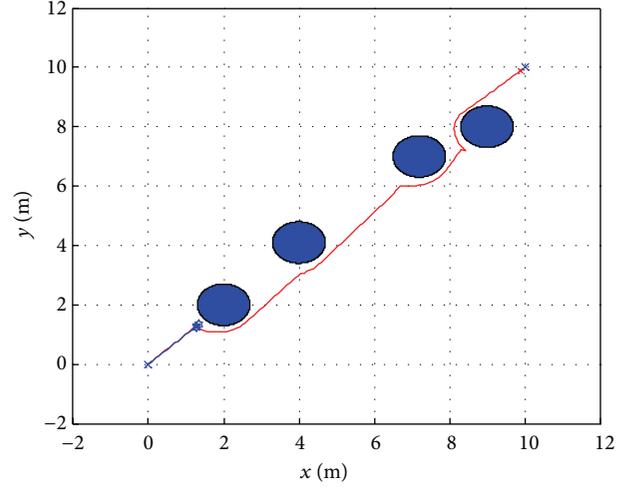


FIGURE 6: Fixed obstacle and goal position. (red) The proposed RPO algorithm, (blue) APF algorithm got stuck in local minimum.

because they are in the region of detected obstacle. But to find the optimal path, we could let particles to somewhat approach to the obstacle in order to find that optimal path. Actually if a particle fall into the region of a detected obstacle, it will not be omitted, but we allow that particle to approach nearby the obstacle to find the new optimal path. For this aim, we introduce a new parameter  $\eta$  which is a coefficient of  $\alpha^{\text{obstc}}$  in decision equation  $e_s^J(t) < \eta$  and let the particle located in the obstacle's region not to be omitted.

**3.2.3. Low Level Decision Control.** Low level decision control of mobile robot is based on each time step. Control law is chosen as follows.

$$u(t) = C(t) \frac{(\theta_{s_{\text{best}}}(t+dt) - q^{\text{first}})}{\|(\theta_{s_{\text{best}}}(t+dt) - q^{\text{first}})\|}, \quad (7)$$

where  $q^{\text{first}}(t)$  is the mobile robot position at time  $t$ , and  $\theta_{s_{\text{best}}}(t+dt)$  is a certain particle position at time  $t+dt$  which has the optimal position and is supposed to be selected by the algorithm and the robot will go through this particle position. To explicit this control law and to compare with classical control law such as PI or PD, it seems it operates as a P controller.

## 4. Simulation Results

In this section, we apply the proposed RPO algorithm on a mobile robot to find an optimal path for the robot to reach its target in the presence of obstacles. This is an online algorithm. In fact, we did not need any information and prior computation from the environment in which the robot performs its task. The only parameters that are obtained from the environment are the target position, robot position, and data from obstacle detection sensors which practically can be obtained from the sensors installed on the robot. All of the other parameters are selected by the operator. Then, the

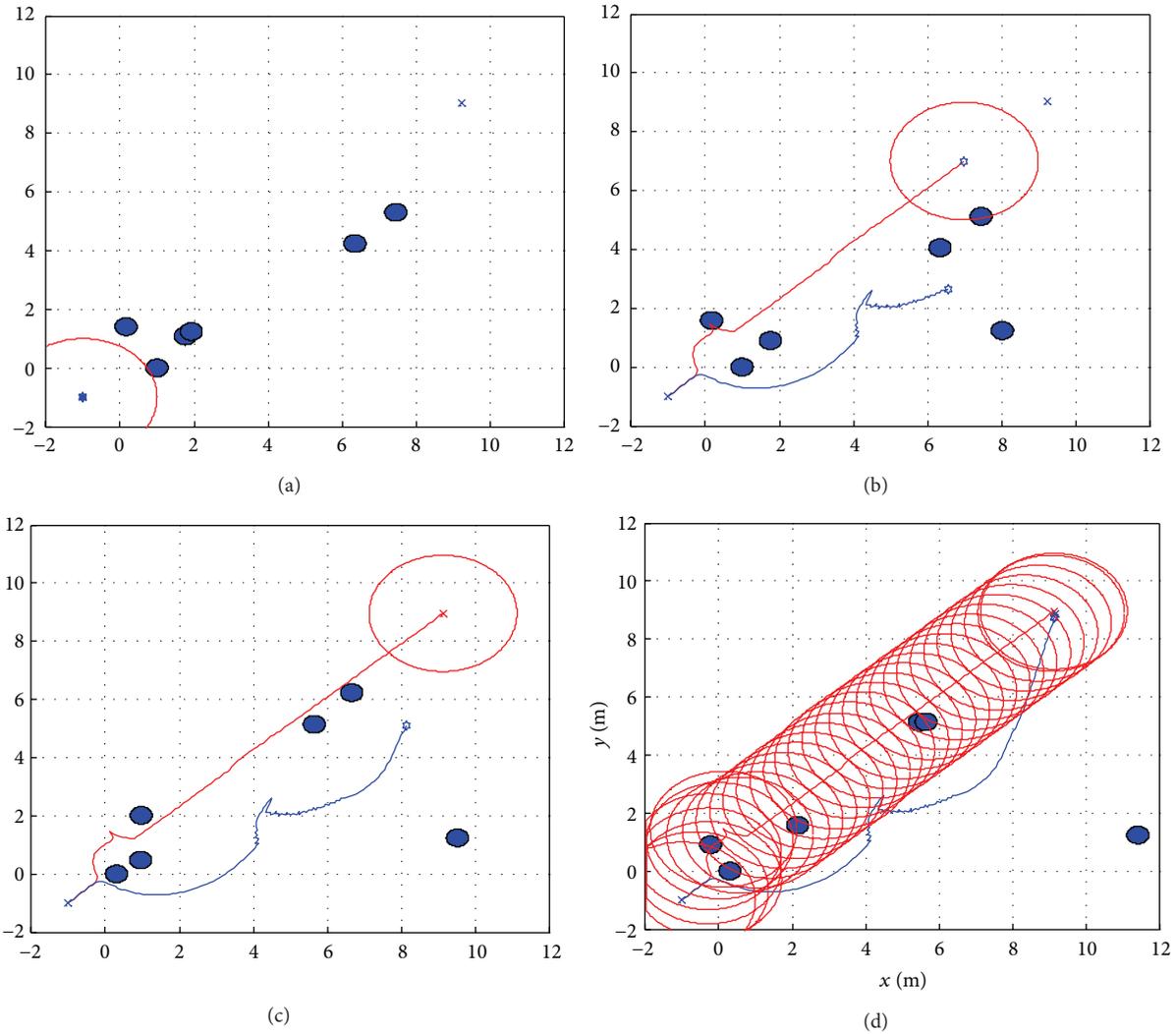


FIGURE 7: Moving obstacle and fixed goal position.

proposed algorithm is computed in the main CPU of the mobile robot, and it helps the robot to find optimal path. The number of obstacles and particles can effect on the real-time performances, but with high speed CPUs there is no problem in real-time performance.

To evaluate our results, in this section three different scenarios have been set up. In the first scenario, obstacles and target are assumed to be fixed. In the second one, the algorithm has been run on a dynamic environment with fixed target and unknown moving obstacles, and in the third one, both moving target and obstacles are evaluated. Furthermore, to assess the performance of the algorithm, a comparison to artificial potential field (APF) method is presented in both scenarios.

The algorithm parameters in simulation are set as shown in Table 1.

As mentioned in Section 3.2.2, an attractant artificial Gaussian function was donated to the target all through the mission. On the other hand, when an obstacle is detected by mobile robot sensors, a repellent artificial Gaussian function

is assigned to the obstacle. Then, the total cost function is computed as a sum of attractant and repellents Gaussian functions. The parameters selected for the Gaussian cost functions are as in Table 2. The parameters for simulation are defined arbitrarily, but in real situation the dimension of obstacles can be estimated by some sophisticated devices like radars and so forth.

**4.1. Fixed Obstacles and Target.** In the first scenario, all the obstacles and the target are assumed to be fixed by the position given in Table 3. Note that obstacles positions are unknown and are detected by the robot's sensors.

The result has been shown in Figure 1. The red circles around the mobile robot show the robot sensor range. Whenever, an obstacle is detected by robot's sensor, the algorithm dedicates a Gaussian function to it and makes the robot change its path to avoid it while it remains on its optimal path. Figure 2 shows these Gaussian functions used by the algorithms. To compare the proposed RPO algorithm by the conventional artificial potential field technique, a simulation

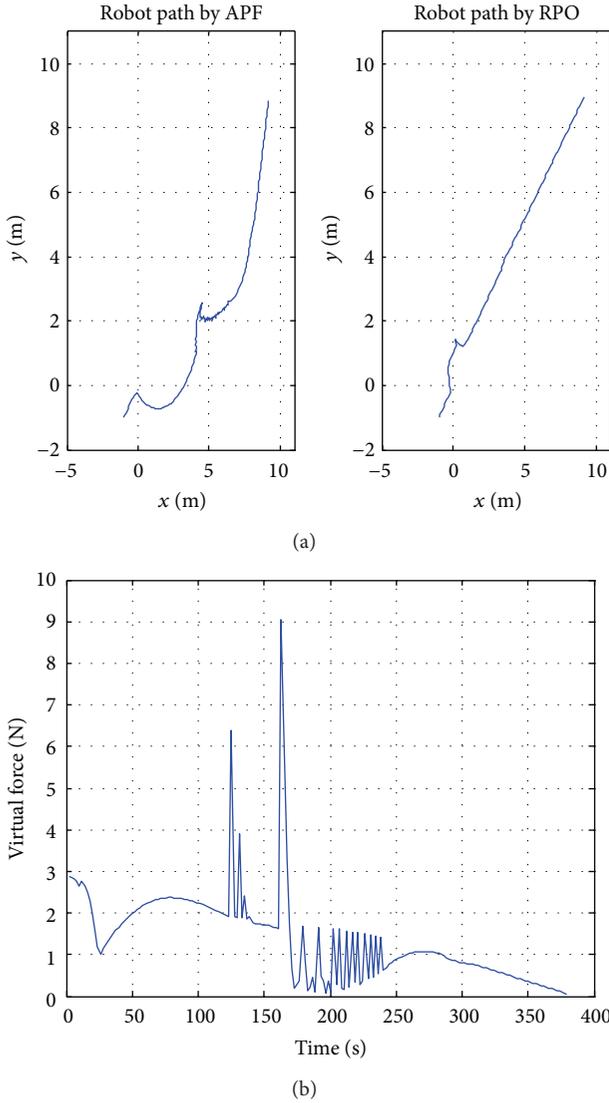


FIGURE 8: (a) Spanned paths by the robot with two algorithms. (b) Exerted virtual force to the robot in APF method.

has been performed on the same condition. Equations (8) and (9) are the most used potential field formula. In the APF method, we suppose that all of the obstacles are unknown until they are within the robot's sensor range, and target positions are known, and attractive potential function is defined as

$$U_{\text{att}}(q) = \frac{1}{2} \zeta \left( \|q(t) - P_{\text{goal}}(t)\|^2 \right), \quad (8)$$

where  $\zeta = 0.2$  is some positive constant scaling factor from [22], and the repulsive potential function is defined as

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2} \eta \left( \frac{1}{\rho_o(q)} - \frac{1}{\rho_o} \right)^2 & \text{if } \rho_o(q) \leq \rho_o, \\ 0 & \text{if } \rho_o(q) > \rho_o, \end{cases} \quad (9)$$

where  $\rho_o$  is a positive constant, and it represents allowance distance to obstacle. In this simulation,  $\rho_o$  is selected same as

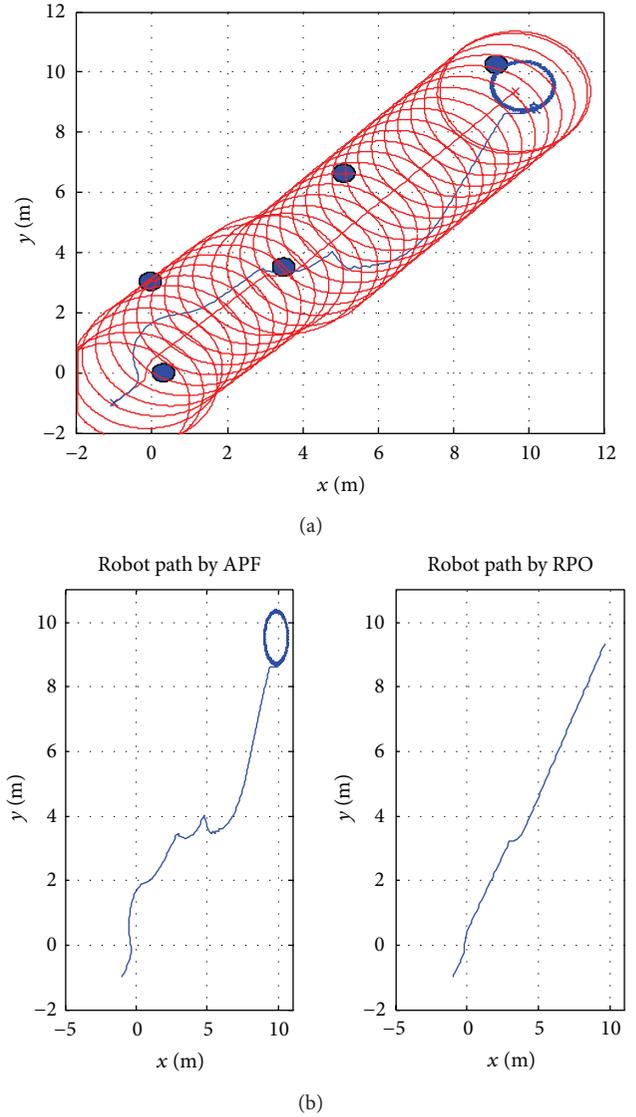


FIGURE 9: (a) Moving obstacle and fixed goal position. (b) Spanned paths by the robot with two algorithms.

$d^{\text{sensor}}$  to make both algorithms the same.  $\eta = 10$  is a positive scaling factor.  $\rho_o(q)$  is defined as  $\rho_o(q) = (\|q(t) - P_{\text{obstc}}(t)\|)$ . Figure 3 shows potential field considered for this simulation. Figure 4 shows paths ran by the both algorithm to reach to the target. The red line shows the path that robot has followed due to the presented algorithm and the blue one refers to the path generated by using the artificial potential field (APF) method. Total virtual force exerted to the robot is also shown in Figure 5. As we expected, in APF target total force is zero.

An important drawback of APF method is getting stuck in local minima. In some certain conditions, the potential field method results in minima other than the target in the plane which absorb the robot, so the robot may get stuck in these minima and could not reach the target. Producing attracting points other than the targets is an inherent drawback of potential field based methods. In this simulation, if we change forth obstacle position to  $[x, y]^T = [4, 4]^T$ , in APF algorithm,

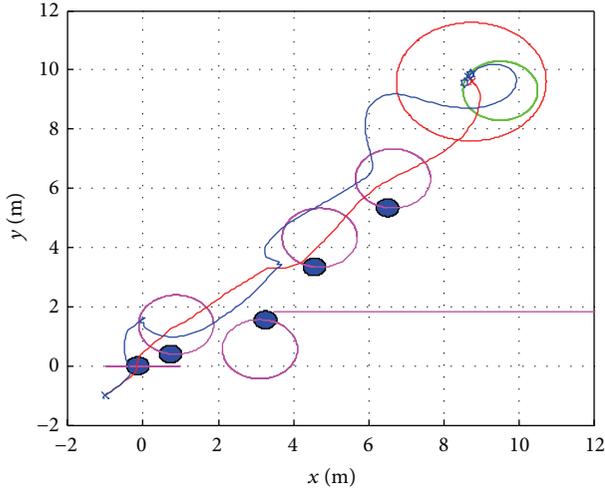


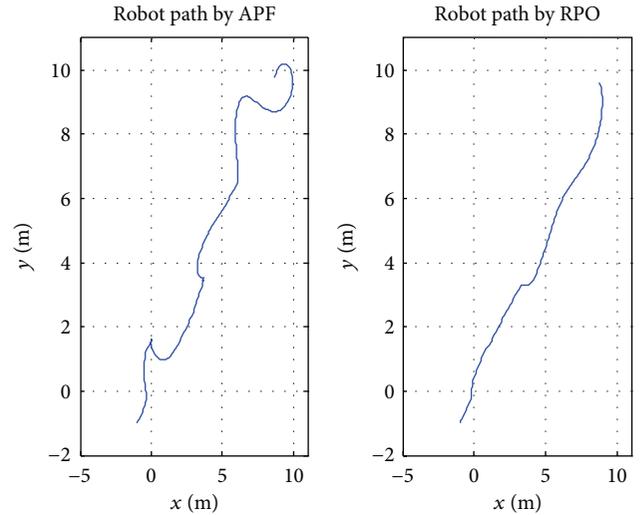
FIGURE 10: Moving obstacle and target.

mobile robot will stuck in local minimum as shown in Figure 6 while mobile robot can find its optimal path to target by the RPO algorithm.

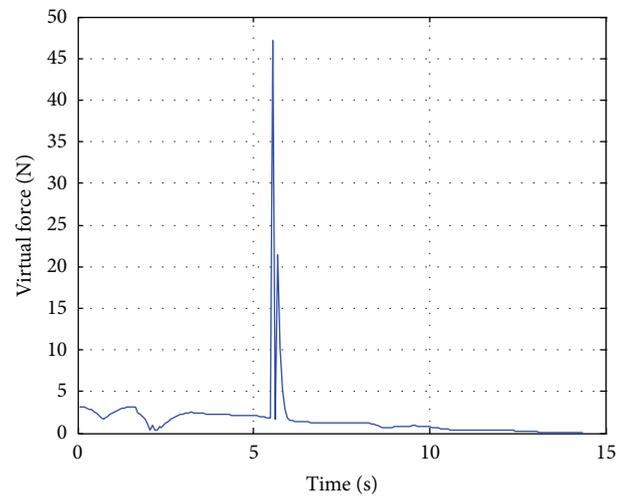
**4.2. Random Moving Obstacle and Fixed Target.** In the second scenario the robot should reach to a fixed target on a dynamic environment with unknown moving obstacles. There are six unknown obstacles in the environment which move randomly. The red circles around the mobile robot show the robot sensor range. Since in this case we deal with moving obstacles, we increase sensor detection range to  $d^{\text{sensor}} = 2 \times 1.2$  m so that the collision avoidance of mobile robot with moving obstacles is guaranteed.  $\rho_o$  is selected same as  $d^{\text{sensor}}$  to make both algorithms the same, and other parameters are the same as the previous one. Figure 7 shows the obtained results. Although the proposed algorithm is online and one step ahead but it generates a near optimal path while the APF generated path is much longer. Generated paths by both algorithms as well as virtual force exerted on robots by the APF are also shown in Figure 8.

In this section, if one of the obstacles positions rotates around the goal, in APF algorithm, mobile robot will stuck in local minimum and with the constant rate rotate around the target and never gets it while the proposed RPO algorithm can find its optimal path to target by the RPO algorithm. These results are shown in Figure 9.

**4.3. Random Moving Obstacle and Target.** In the third scenario the robot should reach to a moving target on a dynamic environment with unknown moving obstacles and a moving target. There are six unknown obstacles in the environment which move randomly. Sensor detection range is  $d^{\text{sensor}} = 2$  m, and  $\rho_o$  is selected same as  $d^{\text{sensor}}$  to make both algorithms the same. Other parameters are the same as previous simulations. Figure 10 shows the obtained results. The red circles around the mobile robot at the target show the robot sensor range. The green and violet circles represent paths ran by moving target and obstacles, respectively. Exerted virtual



(a)



(b)

FIGURE 11: (a) Spanned paths by the robot with two algorithms. (b) Exerted virtual force to the robot in APF method.

force by the APF and spanned paths by both algorithms are shown in Figure 11.

## 5. Conclusion

Local path planning of mobile robots was addressed in this paper. In comparison to artificial potential field method, the RPO gives much optimal solution in shorter time. Furthermore, it is free from some drawbacks such as trapped states or gradient of potential functions which could provide impulsive force to the robot.

## References

- [1] K. Sugihara and J. Smith, "Genetic algorithms for adaptive motion planning of an autonomous mobile robot," in *Proceedings of the IEEE International Symposium on Computational*

- Intelligence in Robotics and Automation (CIRA '97)*, pp. 138–143, Monterey, Calif, USA, July 1997.
- [2] T. Lozano-Perez, “A simple motion planning algorithm for general robot manipulators,” *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 224–238, 1987.
  - [3] J. Laumond, “Feasible trajectories for mobile robots with kinematic and environment constraints,” in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS '86)*, pp. 346–354, Amsterdam, The Netherlands, 1986.
  - [4] E. Palma-Villalon and P. Dauchez, “World representation and path planning for a mobile robot,” *Robotica*, vol. 6, no. 1, pp. 35–40, 1988.
  - [5] C. S. Tseng, C. Crane, and J. Duffy, “Generating collision-free paths for robot manipulators,” *Computers in Mechanical Engineering*, vol. 7, no. 2, pp. 58–64, 1988.
  - [6] O. Takahashi and R. J. Schilling, “Motion planning in a plane using generalized Voronoi diagrams,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143–150, 1989.
  - [7] G. Song, S. Miller, and N. M. Amato, “Customizing PRM roadmaps at query time,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 1500–1505, Seoul, Republic of Korea, May 2001.
  - [8] F. Lingelbach, “Path planning using probabilistic cell decomposition,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 467–472, New Orleans, La, USA, 2004.
  - [9] F. Lingelbach, “Path planning for mobile manipulation using probabilistic cell decomposition,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, pp. 2807–2812, Sendai, Japan, October 2004.
  - [10] R. Gayle, A. Sud, M. C. Lin, and D. Manocha, “Reactive deformation roadmaps: motion planning of multiple robots in dynamic environments,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 3777–3783, San Diego, Calif, USA, November 2007.
  - [11] O. S. Asaolu, *An intelligent path planner for autonomous mobile robots [Ph.D. thesis]*, Department of Systems Engineering, University of Lagos, Lagos, Nigeria, 2001.
  - [12] O. Ibidapo-Obe and O. S. Asaolu, “Optimization problems in applied sciences: from classical through stochastic to intelligent metaheuristic approaches,” in *Handbook of Industrial and Systems Engineering*, A. B. Badiru, Ed., chapter 22, pp. 1–18, CRC Press, Taylor and Francis Group, New York, NY, USA, 2006.
  - [13] O. Ibidapo-Obe, O. S. Asaolu, and A. B. Badiru, “Generalized solutions of the pursuit problem in three-dimensional Euclidean space,” *Applied Mathematics and Computation*, vol. 119, no. 1, pp. 35–45, 2001.
  - [14] O. Ibidapo-Obe, O. S. Asaolu, and A. B. Badiru, “A new method for the numerical solution of simultaneous nonlinear equations,” *Applied Mathematics and Computation*, vol. 125, no. 1, pp. 133–140, 2002.
  - [15] K. Heero, J. Willemson, A. Aabloo, and M. Kruusmaa, “Robots find a better way: a learning method for mobile robot navigation in partially unknown environments,” in *Proceedings of the 8th Conference on Intelligent Autonomous Systems (IAS '04)*, pp. 559–566, Amsterdam, The Netherlands, 2004.
  - [16] J. Fasola, P. E. Rybski, and M. M. Veloso, “Fast goal navigation with obstacle avoidance using a dynamic local visual model,” in *Proceedings of the 7th Brazilian Symposium on Artificial Intelligence*, pp. 1–6, São Luís, Brazil, September 2005.
  - [17] M. Bikdash, S. Karagol, and M. Charifa, “Mesh analysis with applications in reduced-order modeling and collision avoidance,” in *Proceedings of the COMSOL Users Conference*, pp. 1–7, Boston, Mass, USA, 2006.
  - [18] M. Becker, C. M. Dantas, and W. P. Macedo, “Obstacle avoidance procedure for mobile robots,” in *ABCMS Symposium Series in Mechatronics*, vol. 2, pp. 250–257, ABCM, 2006.
  - [19] R. Philippsen, B. Jensen, and R. Siegwart, “Towards real-time sensor-based path planning in highly dynamic environments,” *Springer Tracts in Advanced Robotics*, vol. 35, pp. 135–148, 2007.
  - [20] Y. P. Li, T. Zielinska, M. H. Ang Jr., and W. Lin, “Vehicle dynamics of redundant mobile robots with powered caster wheels,” in *Proceedings of the 16th CISM-IFTOMM Symposium, Romansy 16, Robot Design, Dynamics and Control*, pp. 221–228, Warsaw, Poland, 2006.
  - [21] J. R. Andrews and N. Hogan, “Impedance control as a framework for implementing obstacle avoidance in a manipulator,” in *Control of Manufacturing Processes and Robotic Systems*, D. E. Hardt and W. Book, Eds., pp. 243–251, ASME, Boston, Mass, USA, 1983.
  - [22] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 500–505, St. Louis, Mo, USA, March 1990.
  - [23] B. H. Krogh and C. E. Thorpe, “Integrated path planning and dynamic steering control for autonomous vehicles,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1664–1669, San Francisco, Calif, USA, April 1986.
  - [24] J. Nygard, P. Skoglar, J. Karlholm, R. Bjorstrom, and M. Ulvklö, “Towards concurrent sensor and path planning,” Tech. Rep., FOI, 2005.
  - [25] V. O. S. Olunloyo and M. K. O. Ayomoh, “Autonomous mobile robot navigation using hybrid virtual force field concept,” *European Journal of Scientific Research*, vol. 31, no. 2, pp. 204–228, 2009.
  - [26] P. Vadakkepat, K. C. Tan, and M. L. Wang, “Evolutionary artificial potential fields and their application in real time robot path planning,” in *Proceedings of the Congress on Evolutionary Computation (CEC '00)*, pp. 256–263, July 2000.
  - [27] S. S. Ge and Y. J. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.
  - [28] A. Poty, P. Melchior, and A. Oustaloup, “Dynamic path planning for mobile robots using fractional potential field,” in *Proceedings of the 1st International Symposium on Control, Communications and Signal Processing (ISCCSP '04)*, pp. 557–561, March 2004.
  - [29] S. R. Munasinghe, C. Oh, J. J. Lee, and O. Khatib, “Obstacle avoidance using velocity dipole field method,” in *Proceedings of the International Conference on Control, Automation, and Systems (ICCAS '05)*, Goyang, Republic of Korea, 2005.
  - [30] R. A. Conn and M. Kam, “Robot motion planning on N-dimensional star worlds among moving obstacles,” *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 320–325, 1998.
  - [31] K. M. Passino, “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems*, vol. 22, no. 3, pp. 52–67, 2002.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

