British Journal of Applied Science & Technology
9(6): 534-550, 2015, Article no.BJAST.2015.292
ISSN: 2231-0843

SCIENCEDOMAIN *international*
www.sciencedomain.org

An Efficient Potential-Function Based Path-Planning Algorithm for Mobile Robots in Dynamic Environments with Moving Targets

Abhinav Rajvanshi[1], Shariful Islam[2], Hasibul Majid[3], Ibrahem Atawi[3], Mohammad Biglerbegian[2*] and Shohel Mahmud[2]

[1]Department of Mechanical Engineering, IIT Roorkee, Roorkee-247667, G-79, Ravindra Bhawan, Uttarakhand, India.
[2]Department of Mechanical Engineering, University of Guelph, 50 Stone Road East, Guelph, Ontario, N1G 2W1, Canada.
[3]Department of Electrical Engineering, University of Tabuk, Saudi Arabia.

Authors' contributions

This work was carried out in collaboration between all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2015/18275
Editor(s):
(1) Mark Vimalan, Department of Physics, Syed Ammal Arts and Science College, India.
Reviewers:
(1) Rajinder Tiwari, Electrical & Electronics Engg, Amity University, Lucknow, India.
(2) Anonymous, Shijiazhuang University, China.
(3) Ravi Narayan, Computer Science & Engineering Department, Thapar University, India.
(4) V. K. Banga, Department of Electronics and Communication Engineering, Amritsar College of Engineering and Technology, India.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=1141&id=5&aid=9599

Original Research Article

Received 13th April 2015
Accepted 5th May 2015
Published 5th June 2015

ABSTRACT

Existing potential-field based path planning methods in the literature often do not take into account environmental constraints and robot dimensions. Moreover, they normally do not provide the shortest path either. In this paper, we develop a new repulsive potential function that incorporates robot dimensions as well as the clearance between the robot and obstacles; using this repulsive function, we mathematically prove that the robot is guaranteed to reach the goal. To avoid obstacle's cavity, we develop our technique "virtual-obstacle", and for local minima we modify the existing artificial goal-technique to ensure robot reaches the goal. Our algorithm renders several

**Corresponding author: E-mail: mbiglarb@uoguelph.ca;*

solutions amongst which we choose the shortest path. We consider both static and dynamic obstacles with static and moving targets and demonstrate the effectiveness of our algorithm in several simulations including narrow passages which is a difficult case. The proposed method, by considering physical and environmental constraints, is an improvement to existing path planning algorithms and is of practical use for implementation in real environments.

Keywords: Mobile robots; potential field; local minima; path planning; dynamic environments; moving targets; virtual obstacles.

1. INTRODUCTION

Mobile robots are increasingly being used in flexible manufacturing, agriculture, land mining, search and rescue especially in hostile/harsh environments where human presence is unsafe. These unique applications of mobile robots have made it an interesting research topic and have caught the attention of numerous researchers over the past few decades. One of the main important aspect in mobile robot research is path planning. Path planning for a mobile robot can be defined as finding a continuous trajectory leading from the initial position of the mobile robot to the target while avoiding obstacles. Developing a collision-free path is important not only for a mobile robot but also for any autonomous system moving through a designated space, such as a robotic manipulator or any autonomous vehicle. In what follows, we first review these methods and state their limitations. We then focus on the potential-field based approaches, which is the basis of our paper. At the end, we propose our method that addresses the drawbacks of the existing methods.

Several approaches have been developed for off-line/on-line mobile robot path planning [1], such as Roadmap [2,3], Cell Decomposition [4], and Potential Fields [5]. All of these are known as classical/deterministic approaches. The roadmap approach may find the shortest path, but it is not sufficiently flexible to apply to the same algorithm for the same environment if the robot's initial and goal positions are changed. In this case, the graph needs to be reconstructed. One can find the near optimal path using the cell decomposition technique by increasing the grid resolution, which in this case will increase computational complexity. The potential field path planning suffers from the problems of goal non-reachable and local minima [6]. Some works have been done to modify the classical approaches, such as Wall Following [7] or using a combination of the approaches [8]. To reduce the computational complexity mentioned above, the probabilistic roadmap algorithm [9] and rapidly-exploring random tree algorithm [10] have been also developed. Although probabilistic algorithms find a feasible path relatively quickly even in high-dimensions, there is no guarantee of finding an optimal path. They also may not work in tiny spaces or narrow hallways. Other approaches that have been attempted include heuristic and meta-heuristic algorithms [11] such as: neural network [3,12], fuzzy logic [6,13-16], genetic algorithms [17-21], ant colony optimization [22,23], Tabu search [24,25], simulated annealing [26], and particle swarm optimization [20,21,27]. These algorithms have been developed to deal with the issues of solution quality. Despite providing a faster and improved solution compared to deterministic methods, these algorithms do not always guarantee an effective solution and sometimes are complex and computationally involved making them difficult to be used in fast applications.

We now focus on the potential field method for path planning, which is the basis of our paper: The potential field method has attracted numerous researchers over the years because of its simplicity, ease of implementation, and low computational cost. This method can be implemented both off-line and on-line. The potential field method was first developed by Khatib [5]. In this approach, the robot experiences an attractive force coming from the goal and a repulsive force from the obstacles. To do so, a scalar function, called potential, is first constructed that has a minimum when the robot is at the goal and high values on obstacles. Everywhere else, the function slopes is downward toward the goal; this will ensure the robot can reach the goal by following negative gradients of the potential. This method has an unavoidable drawback of trapping a robot in a local minimum and goal non-reachable [28]. Local minima will occur when the sum of attractive and repulsive forces on the robot is zero and the goal is non-reachable happens when the goal is close to an obstacle which produces a large repulsive force. Researchers have attempted to solve these issues in various ways. A harmonic potential function is proposed by Kim and Khosla [29] for static obstacle avoidance. Although this method does not have the local minima problem, it does not guarantee to generate the shortest path. Similar work has been done by Daily and Bevly [30]. Ko and Lee [31] applied the artificial potential function to avoid collision with moving obstacles. A virtual distance function was developed in [31] that takes into consideration the distance from the robot to obstacles and the relative motion of the obstacles with the robot; however, the robot might collides with obstacles depending on the ratio of virtual distance to real distance. Ni et al. [32] proposed an improved virtual force field (VFF) approach and added a fuzzy control module to avoid collision with dynamic obstacles. Ni et al. [32] considered relative velocity between the robot and the dynamic obstacles as [31]. In addition [32] considered the angle between the relative velocity vector and the line from the robot to ensure collision avoidance. Ni et al. [32] demonstrated the effectiveness of their proposed approach over the traditional VFF through simulation environment by finding a collision free path, which can be computationally expensive and hard to implement.

Recently, Sgorbissa and Zaccaria [33] proposed a hybrid approach for obstacle avoidance in a partially unknown dynamic environment. They were able to show that their algorithm does not suffer from the local minima problem. This method integrates a prior knowledge of an environment with local perceptions to execute the assigned tasks efficiently and safely. Malakar and Sinha [34] presented an improved artificial potential field-based regression search algorithm for static obstacle avoidance, where the parameters are optimized using the particle swarm optimization technique. They were able to achieve a global sub-optimal/optimal path efficiently without local minima and finally avoid oscillatory movement and unreachable problems. However, this method neglects robot and obstacles dimensions and is not also applicable to a dynamic environment. Similar methods have been proposed in other papers [35-37].

Sheng et al. [38] proposed an improved artificial potential field (AFP) for virtual human path planning and addressed the issue of local minima and goal non-reachable. The local minimum was solved using the virtual goal technique. In the virtual goal technique, when a robot is trapped in a local minimum, a virtual goal is created and the robot starts moving towards the virtual goal instead of the real goal. The goal non-reachable issue was addressed by proposing an improved repulsive force field function. The limitation of their work [38] is the local shock problem (i.e., the robot returns to its original position because of the presence of large obstacles). Yang et al. in [39] also applied the virtual goal technique to avoid the local minimum for an autonomous underwater vehicle. Similar methods have been proposed in other papers [40-42].

Most recently, Guanghui et al. [43] developed an artificial potential field-based regression search method for autonomous mobile robot path planning. They successfully demonstrated finding a collision-free optimal path without local minima for an environment including known, partially-known, or unknown static and dynamic environments. They also addressed the goal non-reachable issue. However, they did not consider the robot diameter and also not defined the clearance space between the robot and obstacle. Moreover, they used a very simplified simulation environment. Montiel et al. [44] developed the parallel evolutionary artificial potential field (PEAPF) algorithm which is an extension of evolutionary artificial potential field (EAPF) algorithm [45]. In EAPF, the artificial potential field (APF) is combined with genetic algorithms to derive optimal potential field functions. The idea of PEAPF is to take the advantage of parallel processing that eventually reduces the computational time. The PEAPF algorithm was tested only for static targets and not for moving ones.

Although several research have been carried out using the APF algorithm [5, 29-45], to the best of our knowledge, no work has been done that considers robot diameters as well as a clearance between robot and obstacle in the potential function, which are necessary to be taken into account in real environments especially when space is limited. In this paper, we present an “enhanced” path planning algorithm using the potential field method that addresses many of the aforementioned concerns: we introduce a new potential function. The proposed potential function can accommodate the robot diameter and a clearance between the robot and obstacles. We also define a region in which obstacles exert a repulsive force to ensure the robot reaches the goal which in effect reduces the computational cost. Moreover, our work does not suffer from the local shock problem [38], it works in dynamic and clustered environments, and we have created a more complicated simulation environment than most recent published works in the literature [38-39,43-44].

Thus, the contributions of this paper are summarized as follows:

(1) Developing an enhanced repulsive force function that guarantees the robot will reach the goal with minimum computational effort
(2) Considering the robot diameter and a clearance between the robot and obstacles in the repulsive force function using an adaptive gain technique
(3) Proposing a new method, called virtual obstacle technique, to avoid obstacle's cavity
(4) Developing a new algorithm to avoid local minima using the artificial goal technique while finding the optimal path
(5) Developing an algorithm for static and dynamic goals (targets) where both static and dynamic obstacles are present.

The remainder of this paper is organized as follows: Section 2 presents our proposed mathematical model; this section includes several subsections to address the techniques to obstacles, avoid local minima, as well as finding the optimal path, Section 3 demonstrates the simulation environment, Section 4 presents the simulation results, and Section 5 concludes the paper.

2. PROPOSED PATH PLANNING ALGORITHM

Our proposed path planning method is potential-field based. A set of obstacles produce a repulsive potential and the goal produces an attractive potential. Forces acting on the robot are thus calculated by taking the negative gradient of the total potential fields. As the attractive force produces negative scalar value and the repulsive force produces a positive scalar, the sum of potential is minimum when the robot is at the goal position and maximum when on obstacles. In this section, we first present a mathematical model of our proposed potential function in 2.1. Subsequently, we develop our algorithm for avoiding obstacle's cavity in 2.2. Subsection 2.3 is dedicated to address the local minima, and Subsection 2.4 presents the strategy for the shortest path.

2.1 Mathematical Model

The Potential field method was first proposed by Khatib [5] and is a popular technique [29-46] for path planning. The mathematical model for the attractive force can be expressed as follows:

$$F_{att}(q_i) = -\zeta(q_i - q_d) \tag{1}$$

where $F_{att}(q_i)$ is the attractive force and a function of gain ζ, q_i and q_d are robot and goal positions, respectively. The value of gain ζ is usually chosen to be less or equal to one, i.e, $\zeta \leq 1$. In this paper, we propose a new repulsive function. The expression for our proposed function is given as follows:

$$F_{rep}(q_i) = \begin{cases} f(r)\eta d \dfrac{(q_i - q_o)}{\rho(q_i)^4} & \rho(q_i) \leq \min\{d, R_l\} \\ 0 & \rho(q_i) > \min\{d, R_l\} \end{cases} \tag{2}$$

where $F_{rep}(q_i)$ is our proposed repulsive force and a function of gain (η), q_i is the robot position, q_o is the position of an obstacle, d is the distance between robot and the goal, $\rho(q_i)$ is the distance between the robot and obstacle, and R_l is a parameter we introduce as the radius of influence. This influence parameter is used to ensure the robot experiences repulsive forces from obstacles that are within the radius of influence. The value of R_l is selected to be greater than the summation of the largest obstacle radius and the robot radius so that the robot can sense the maximum obstacle. The expression for R_l can be written as

$$R_l > \max\{r_i\} + r_b, i = 1,2,3,... \tag{3}$$

where r_i is the radius of the i^{th} obstacle and r_b is the radius of the robot. Inclusion of R_l will reduce the computational efforts in determining the path to the goal. To demonstrate the repulsive potential force, three difference cases have been considered, as illustrated in Fig. 1. Fig. 1 (a) shows the case when the goal lies outside R_l . As can be seen the robot experiences the repulsive force only from the obstacle presents within R_l .

Fig. 1 (b) and Fig. 1(c) show cases where the goal and the obstacle are inside R_l, i.e., $R_l > d$. As shown, the robot receives the repulsive force from the obstacle only if $\rho \le d$ (as shown in Fig. 1 (b)). If $\rho > d$, the robot experiences only attractive force and moves towards the goal, Fig. 1(c), though the obstacle is within the radius of influence R_l. The methodology represented by Fig. 1(b) and Fig.1(c) guarantees reaching the goal even when an obstacle is located very close to the goal. Our proposed algorithm will thus guarantee that the robot in any situation reach the goal.

In Equation (2), we have introduced a new function, $f(r)$, which is expressed as follows:

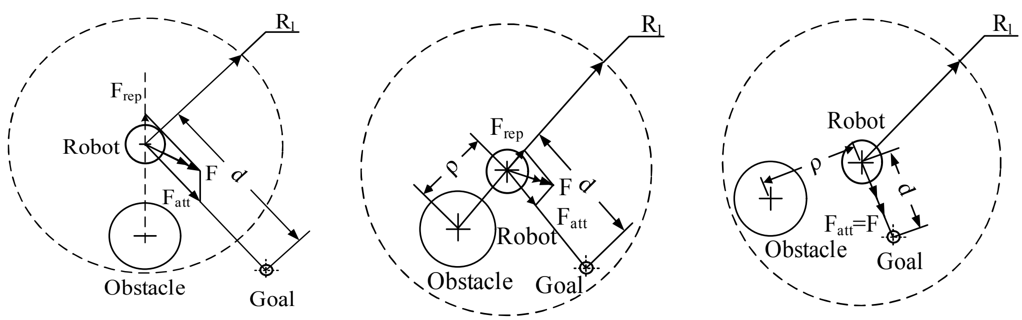
$$f(r) = \frac{\zeta}{\eta}(r + r_b + c)^3 \tag{4}$$

where r is the radius of the obstacle, r_b is the radius of the robot, c is the gap between the robot and obstacle and called clearance and defined as $\beta(r + r_b)$; the clearance can be adjusted by the gain $\beta \ge 0$. The net force causing the robot to move forward towards the goal while avoiding obstacles is thus given by

$$F(q_i) = F_{att}(q_i) + F_{rep}(q_i) \tag{5}$$

2.1.1 Mathematical Justification for $f(r)$

In this section, we provide a mathematical justification for proposing function $f(r)$. Let us consider a robot of radius r_b moving towards the goal under the influence of an attractive force $F_{att}(q_i)$ and faces a circular obstacle of radius r. We assume the line joining the centers of the robot, obstacle, and the goal lie on one line as shown in Fig. 2. There is a point on the line where the net force acting on the robot is zero, i.e., $F_{att} = F_{rep}$, known as a local minimum. It is known that the robot stops moving forward when located at the local minimum. If the local minimum is located inside the obstacle, there is a chance for the robot to collide with it. Therefore, the proposed repulsive function in Equation (4) introduces the term c, as shown in Fig. 2. This parameter is called clearance and meant to adjust the position of the local minima so that the local minimum falls outside the obstacle. At the local minimum: $\rho = r + r_b + c$, i.e., the distance between the center of robot and obstacle is the summation of the radius of obstacle, robot, and a user-defined clearance.



(a) (b) (c)

Fig. 1. Illustration of repulsive force function

By equating Equations (1) and (2), we get

$$F_{att} = F_{rep} \Rightarrow \zeta(q_i - q_d) = f(r)\eta d\frac{(q_i - q_o)}{\rho(q_i)^4} \Rightarrow \xi d = f(r)\frac{\eta d(r + r_b + c)}{(r + r_b + c)^4}$$

$$\Rightarrow f(r) = \frac{\zeta}{\eta}(r + r_b + c)^3 \tag{6}$$

The next subsections present the algorithm for avoiding obstacle's cavity, strategies for addressing the local minimum, and finding the shortest path.

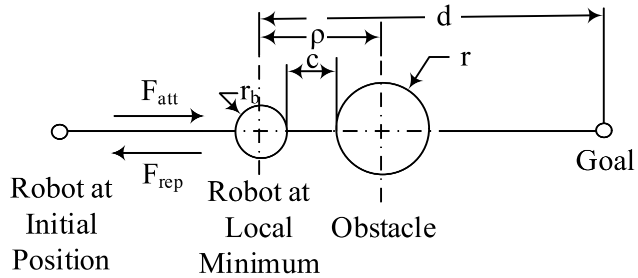


Fig. 2. Robot at the point of local minima

2.2 Avoiding Obstacle's Cavity

The robot can be trapped by an obstacle if there is a cavity in it. To overcome this issue, we propose the following virtual obstacle technique:

Virtual Obstacle Technique: To avoid getting the robot trapped, the cavity of an obstacle is filled out by a virtual obstacle. The outer most two vertex (V_1,V_2) are considered as two end points of the virtual line as shown in Fig. 3. Five virtual circular obstacles are drawn on the virtual line as described in Section 3. Due to the presence of virtual obstacles, the robot cannot enter into the cavity and avoid the local minimum problem. However, local minimum can also be occurred if the robot, obstacle and the goal are on the same line. To address the local minima problem, we propose a strategy which is described in the next section.

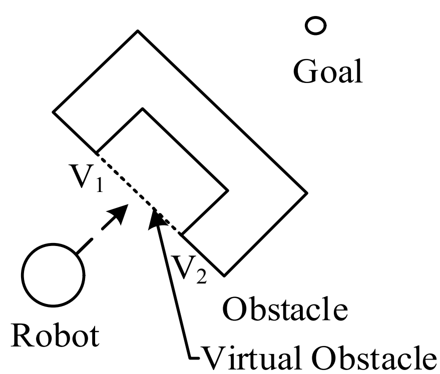


Fig. 3. Virtual obstacle technique

2.3 Addressing Local Minima

The local minimum can occur if the center of the robot, obstacle and goal are on the same line and the total force acting on the robot becomes zero. Under this condition, the robot undergoes very small oscillations around the point of local minimum and cannot move forward. The local minimum is one of the major issues in employing the potential field method. In our proposed algorithm, the difference between the robot's current position and position before 2 iterations is calculated which is key to identify the local minimum location. If this difference is lower than 0.1×step size then it is considered that local minimum has been encountered in the algorithm. A new algorithm is proposed to overcome the local minimum using artificial goals technique.

Artificial Goal: Once the robot is trapped at a local minimum, the algorithm creates artificial goals as shown in Fig. 4.

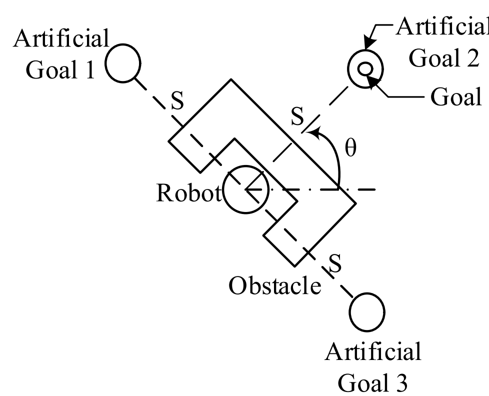


Fig. 4. Artificial goal technique

The artificial goals are created by

$$S = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{7}$$

$$\theta = \tan^{-1} \frac{y_2 - y_1}{x_2 - x_1} \tag{8}$$

Where S is the distance between the robot and the goal, and θ is the heading angle of the robot.

The algorithm generates three virtual goals as shown in Fig. 4: Artificial Goal 1, Artificial Goal 2 and Artificial Goal 3. The co-ordinates of the three goals are as follows:

Goal 1 $\left\{x + S \cdot \cos\left(\theta + \frac{\pi}{2}\right), y + S \cdot \cos\left(\theta + \frac{\pi}{2}\right)\right\}$,
Goal 2 $\{x + S \cdot \cos(\theta), y + S \cdot \cos(\theta)\}$ and
Goal 3 $\left\{x + S \cdot \cos\left(\theta - \frac{\pi}{2}\right), y + S \cdot \cos\left(\theta - \frac{\pi}{2}\right)\right\}$ where x and y are the coordinates of the robot current position The angle between Goal and Artificial Goal 1 is $\frac{\pi}{2}$, and between Goal and Artificial Goal 3 is -$\frac{\pi}{2}$. Artificial Goal 2 lies on the Goal. The robot is pushed out from a local minimum with the inclusion of those three artificial goals. All three artificial goals create repulsive forces according to the following expression:

$$F_{artificial}(q_i) = -k_j \zeta (q_i - q_d) \tag{9}$$

where $j = 1, 2, 3$ represent three artificial goals. One can choose $k_j \leq -1$ in this sub-algorithm (which is dedicated to avoid local minima). Doing so will ensure repulsive forces will be created by Equation (9). The magnitude of each repulsive force of these forces depends on the value of k_j.

Through sequential activation and deactivation of those artificial goals, the robot will get out of the local minimum: when the robot is trapped by an obstacle, there are always two options to get out of the local minimum, i.e., the using either left or right side of the obstacle. In our method, we have calculated both options for each obstacle by creating two imaginary robots. Finally, the minimum path among all feasible paths is selected.

Once the robot is in local minimum ($F_{att} = F_{rep}$), the artificial goals are activated. At the same time, two imaginary robots of same dimensions of original robot are created. In each iteration i , an imaginary robot moves under combined influence of artificial and actual goals and travels by a distance $(3\delta + 0.20i \cdot R_m)$ which increases with the number of iteration. One can also replace 3δ by a suitable positive number depending on the depth of cavity in an obstacle. So, in the first step (i.e. $i = 0$), the robot will move a small distance of 3δ under the influence of artificial and actual goals. For $i > 0$, the robot's travelled distance depends on R_m (i.e., the maximum of the obstacles radii lying within the radius of influence). The idea of defining robot's travelled distance is that the robot will travel the minimum distance to get out of the cavity of an obstacle or local minimum. Once the imaginary robot gets out of the local minimum, the artificial goals are set 'off' and the imaginary robot moves towards actual goal.

The direction of robot movement is depended on particular combination of artificial goals chosen in a step. A combination of artificial and original goals creates the following three scenarios:

a) The imaginary robot may get out of the local minimum or
b) The imaginary robot may return to the local minimum if the distance travelled is not sufficient to get out of the cavity or
c) The imaginary robot may not be able to move by the particular distance because of the presence of obstacle in that direction.

All these three cases have been taken into consideration in the algorithm by using different indices. The proposed artificial goal algorithm is presented in Appendix A. A flowchart for implementing the entire algorithm is shown in Fig. 5.

2.4 Shortest Path

In this paper, we also find the shortest path for the robot. The shortest path is calculated by counting the steps required to reach the goal by the imaginary robot: each time the robot is in a local minimum, two imaginary robots are created in the artificial goal algorithm to find all possible paths. The steps taken by the imaginary robot to reach the goal position from the local minimum are multiplied by δ (step size) to find the arc length. All of the arc lengths are calculated and finally added to determine the length of each path from start to the goal position. The actual robot will follow the shortest path among all feasible paths.

3. SIMULATION ENVIRONMENTS

Obstacle: Two types of obstacles are considered in the simulation: a) static and b) dynamic.

a) Static Obstacles: A static obstacle is considered of an arbitrary polygon shape or a circle. Each edge of a polygon is divided into an odd number (five in this paper) of identical circles as well as two corner circles located at the vertices of the polygon as shown in Fig. 6. The centers of these identical circles divide the edge of the polygon into six equal parts. We consider five circles to reduce the computational efforts in avoiding obstacles and also to reduce the gap between the wall and the robot path. This gap can be further reduced by increasing the number of circles. The diameter of the circles depends on the polygon edge length. The center points of all circles are calculated using the following equation:

$$x = \frac{mx_2 + nx_1}{m + n} \tag{10}$$

$$y = \frac{my_2 + ny_1}{m + n} \tag{11}$$

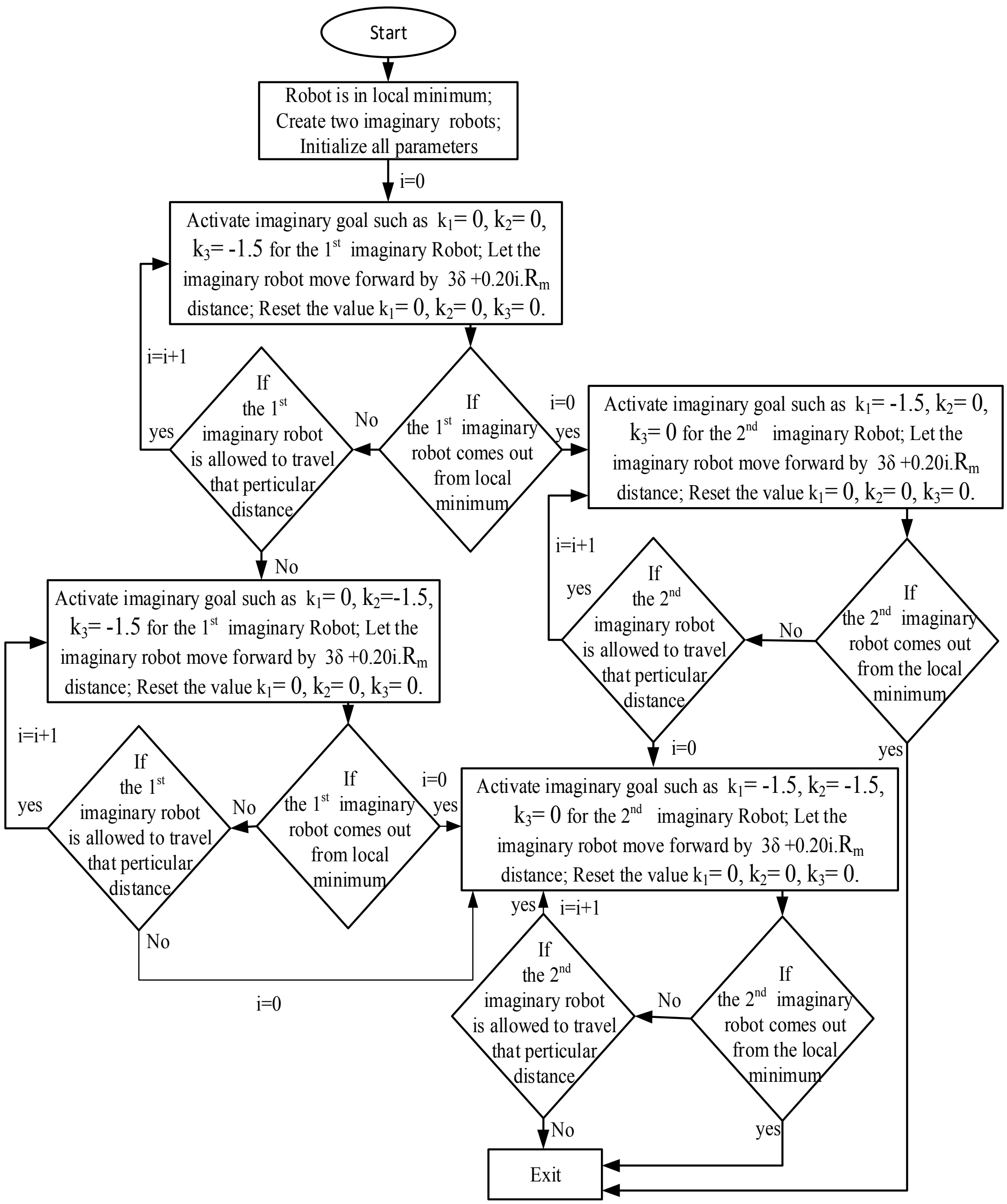


Fig. 5. Proposed flowchart

where (x_1, y_1) and (x_2, y_2) are the two end points of a line, respectively, (x, y) divides the line with the ratio of m to n.

Fig. 6 shows an example of creating two legs of a polygon obstacle. $A(x_1, y_1)$ and $B(x_2, y_2)$ are the coordinates of two vertices with a length AB, and $R_1 = \frac{AB}{12}$. The five centers are obtained from Equations 10 and 11 as:

$\frac{5A+B}{6}$, $\frac{4A+2B}{6}$, $\frac{3A+3B}{6}$, $\frac{2A+4B}{6}$, and $\frac{A+5B}{6}$. The diameter of a corner circle will be the larger than the two circles on the adjacent edges so that there will be no gap among the circles; for example in Fig. 6, $R_1 > R_2$, leads to choose the corner circle with a radius of R_1.

b) Dynamic Obstacles: We also consider dynamic obstacles in developing our path planning algorithm. To do this, a point-mass or a circular obstacle is considered representing dynamic obstacles. Let (x, y) be the coordinate of the center point of an obstacle and assume the obstacle is moving on a user-defined path of $f(x)$ with a velocity v and $\tan\theta$ is the slope of the path at any point (x_t, y_t). The obstacle position is updated after the robot position is updated each time. The position of the obstacle is obtained as follows:

$$x_{t+1} = x_t + t_R v\cos\theta = x_t + t_R \frac{v}{\sqrt{1+\tan\theta^2}} = x_t + t_R \frac{v}{\sqrt{1+[f'(x_t)]^2}} \quad (12)$$

$$y_{t+1} = y_t + t_R v\sin\theta = y_t + t_R \frac{v\tan\theta}{\sqrt{1+\tan\theta^2}} = y_t + t_R \frac{v}{\sqrt{1+[f'(x_t)]^2}} f'(x_t) \quad (13)$$

where (x_t, y_t) is the current position of the obstacle, $f'(x_t)$ is the derivative of a user-defined path for the obstacle at point x_t, t_R is the time takes by the robot to move one step forward and computed by calculating the net force acting on the robot. An obstacle of any shape such as point, circular, or polygon and dimension may have any user-defined path function. For a polygon, defining any arbitrary motion needs to define differential motion on every circle.

4. SIMULATION RESULTS

The mathematical model described in Section 2 along with our proposed algorithm are implemented in MATLAB environment. The parameters used for the simulations are as follows: initial position of the robot = $[0,0]^T$; goal position = $[20,20]^T$; gain $\zeta = 1.5$; $\beta = 0.2$; radius robot, $r_b = 0.3$, and step size, $\delta = 0.1$. Four different simulation environments (Fig. 7-Fig. 10) are developed to demonstrate the effectiveness of our proposed path planning algorithm in various case studies. Fig. 7 shows that the robot can navigate successfully and reach the goal in an environment where 17 different static obstacles of various shapes are present. Furthermore, the effectiveness of the virtual obstacle technique is also shown in Fig. 7, where the cavity of each obstacle is covered by a virtual obstacle. Fig. 7 shows that the robot avoids the cavity of obstacle 1, 3 and 4 due to the presence of virtual obstacles, resulting in reaching the goal without being trapped by any of the obstacles.

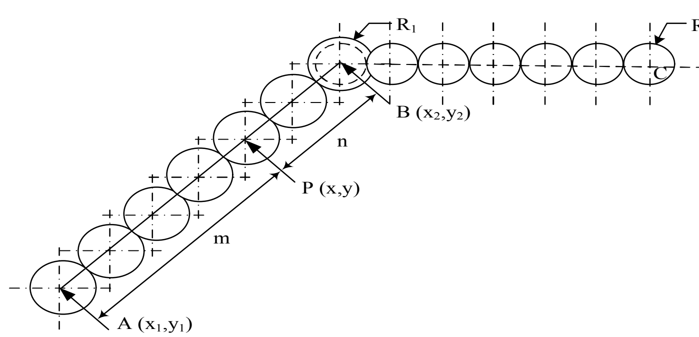


Fig. 6. Construction of polygon edged

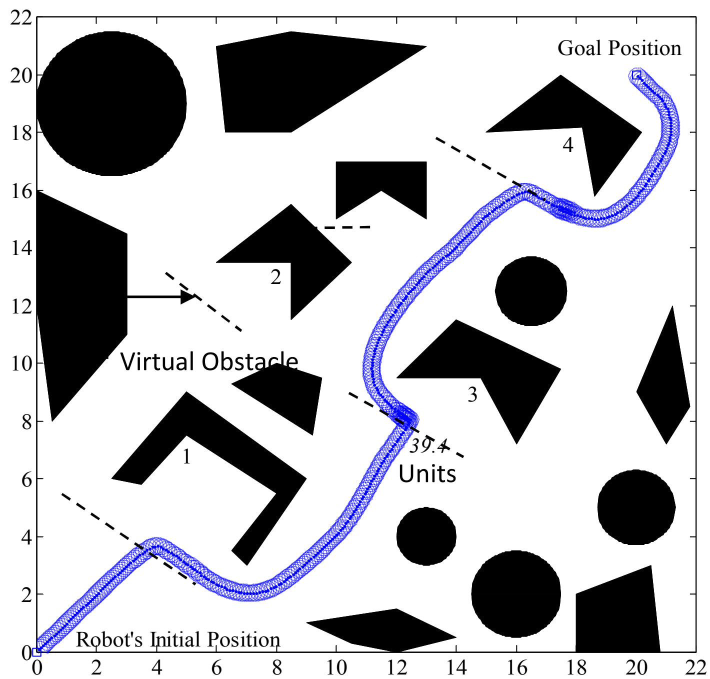


Fig. 7. Virtual obstacles are created to avoid the local minimum

Fig. 8 shows the effectiveness of the artificial goal algorithm. The robot starts and initially gets trapped in obstacle 1. Two imaginary robots of the same dimension as the real robot are thus created at this local minimum. One of the imaginary robots gets out of the cavity of obstacle 1 from the left side and gets further trapped by obstacle 2. On the other hand, the other imaginary robot gets out from the right side of the obstacle 1 and further gets trapped by obstacle 2. Again two imaginary robots of the same dimension as the original one are created at each obstacle (obstacle 2, obstacle 3). Two imaginary robots get out of obstacle 2 and reach the goal. The same phenomenon is seen for the robot trapped by obstacle 3. The imaginary robot getting out from the right side of the obstacle 2 avoids the path between obstacles 4 and 5, because the dimension of the robot is critical this time, although taking the path between obstacles 4 and 5, and reaching the goal can be the shortest path. Same scenario is found when the imaginary robot gets out from the left side of the obstacle 3. This is of practical use in many applications where the space is tight, such as narrow hallways or bathrooms, etc. In summary, every time the robot is at a local minimum or in an obstacle cavity, it will try to find all possible options to get out of that trapped situation. The green lines shown in Fig. 8 are different trials of the robot that eventually led it to reach the goal successfully. All of the possible ways from initial to the goal position are found using the artificial goal technique and finally the real robot follows the shortest path among all possible paths which is shown using the blue line in Fig. 8. Potential-field based methods are often very difficult to find the path when narrow hallways exist [46,47]. In Fig. 9, we created narrow hallways and showed the effectiveness of our algorithm. The green lines in Fig. 9 show different trials for finding the feasible path and the blue line shows that the robot reaches the goal without being trapped by an obstacle.

To demonstrate the algorithm in a dynamic environment, in the next experiment we consider both moving obstacles and target. The result is shown in Fig. 10, where we have assumed that the robot moves in a dynamic environment, where the target moves at a constant velocity of 0.3 units/s along the path, $f_{goal}(x) = 0.1x$, starting from $[20,20]^T$. Sixteen obstacles exist in the environment, among which obstacles 1 and 2 are the moving ones (Fig. 10) according to user-defined paths $f_1(x) = -0.4x$ and $f_2(x) = -x + 3\sin\left(\frac{x}{2}\right)$, respectively.

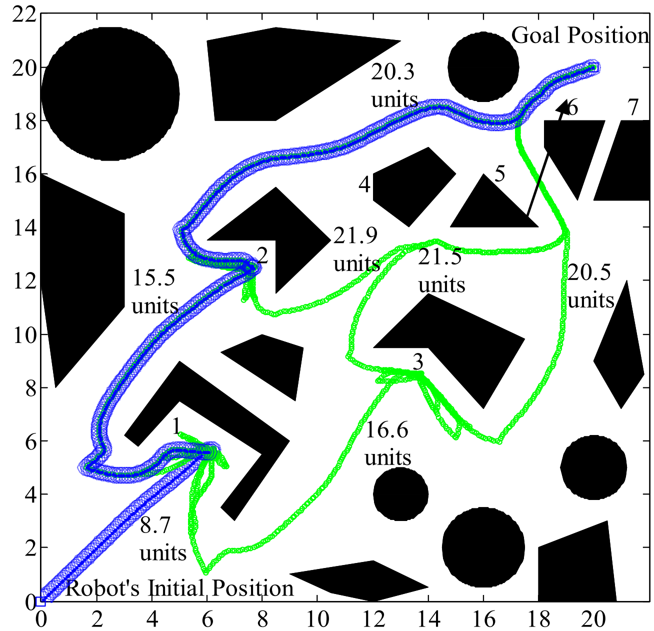


Fig. 8. Demonstration of the algorithm in a cluttered environment

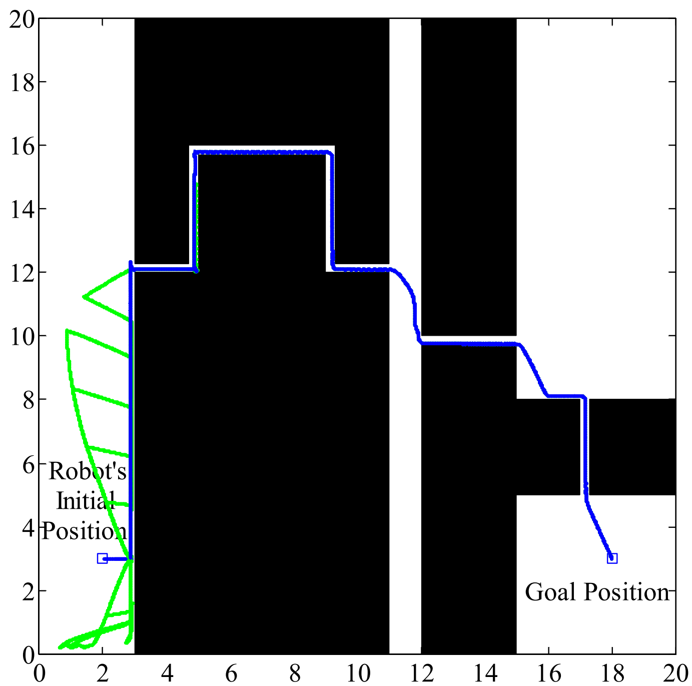


Fig. 9. Demonstration of the algorithm for narrow hallways

The unfilled obstacles shown are the initial positions of the dynamic obstacle. The hexagonal Obstacle 1 moves at a velocity of 0.28 unit/s, and circular Obstacle 2 moves at a velocity of 0.33 unit/s. Fig. 10 demonstrates that the robot reach the dynamic goal without colliding with static or dynamic obstacles in the environment.

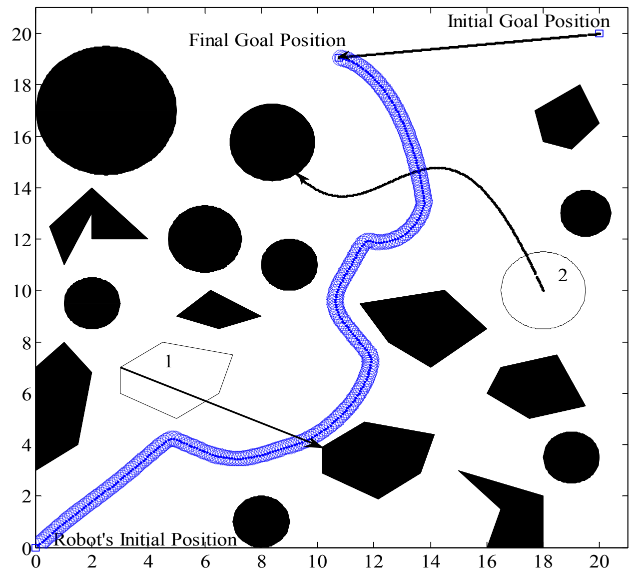


Fig. 10. Combined static and dynamic obstacles with a dynamic target.

5. CONCLUSION

In this paper, a new potential function is proposed for mobile robot motion planning in dynamic environments. The effectiveness of the proposed algorithm is shown through several simulations, where we addressed the issues of local minima, goal non-reachable, shortest path with the presence of both static and dynamic obstacles for static and dynamic goals. We considered robot dimension and clearance between the robot and obstacles in proposed repulsive potential function which adds practicality to our approach. We developed the virtual obstacle technique to avoid getting trapped inside an obstacle and provided a shorter collision-free path. However, the virtual obstacle technique requires accurate obstacle shape. To avoid local minima or cavity of an obstacle, the artificial goal technique is used. In comparing the two methods for avoiding obstacle's cavity (one being the virtual obstacle and other the artificial goal), the virtual obstacle technique provides a shorter collision-free path. At the same time, the virtual obstacle technique does not require finding all feasible paths and therefore, the virtual obstacle technique is computationally less expensive than the artificial goal technique. However, the virtual obstacle technique requires accurate obstacle shape and requires extra care in putting a virtual obstacle in front of the cavity of an obstacle before starting the simulation and also not applicable for all type of environment. In contrast, the artificial goal algorithm needs to be programmed only once and it will guarantee avoidance of the local minimum and any type of obstacle trap. The artificial goal algorithm is also applicable in an environment where narrow hallways are present.

As for future work, smoothing the path, reducing the computational time, and implementing the proposed algorithm in real-time are potential topics for investigation.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Kavraki LE, LaValle SM. Chapter 5: Motion Planning. Springer Handbook of Robotics; 2008.
2. Kavraki LE, Svestka P, Latombe JC, Overmars MH. Probabilis-tic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation. 1996;12(4): 566–580.
3. Sadati N, Taheri J. Hopfield Neural Network in solving the robot motion planning problem. IASTED Intl. Conf. on Applied Informatics, Innsbruck, Austria; 2002.
4. Lozano-Perez T. Spatial planning: A configuration space approach. IEEE

Transactions on Computers. 1983;C-32(2), 108-120.
5. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research; 1986.
6. Zhu Q, Yan Y, Zhuoyi XZ. Robot path planning based on artificial potential field approach with simulated annealing. Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications. 2006;622-627.
7. Yata T, Kleeman L, Yuta S. Wall following using angle information measured by a single ultrasonic transducer. Proceedings of the IEEE International Conference on Robotics & Automation Leuven; 1998.
8. Masehian E, Sedighizadeh D. Classic and heuristic approaches in robot motion planning a chronological review. Proceedings of World Academy of Science, Engineering and Technology. 2007;23:101–106.
9. Park B, Choi J, Chung WK. Sampling-based retraction method for improving the quality of mobile robot path planning. International Journal of Control, Automation, and Systems. 2012;10(5):982-991.
10. LaValle SM, Kuffner JJ. Randomized kinodynamic planning. International Journal of Robotics Research. 2001;20(5): 378–400.
11. Huili W, Weiping F, Zongde F. A path planning method for mobile robot based on improved potential field function. Machine Tool & Hydraulics. 2002;6:67–71.
12. Engedy I, Horvath G. Artificial neural network based mobile robot navigation. In: Proceedings of 2009 the 6th IEEE International Symposium on Intelligent Signal. Budapest, Hungary. 2009;241–246.
13. Chao CH, Hsueh BY, Hsiao MY, Tsai SH, Li THS. Fuzzy target tracking and obstacle avoidance of mobile robots with a stereo vision system. International Journal of Fuzzy Systems. 2009;11(3):183–191.
14. Chen C, Richardson P. Mobile robot obstacle avoidance using short memory: a dynamic recurrent neuro-fuzzy approach. Transactions of the Institute of Measurement and Control. 2012;34(2-3): 148–164.
15. Chen D, Liang Y. Behavior-based fuzzy control of obstacle avoidance for indoor mobile robot. Advanced Materials Research. 2010;34-35,482–486.
16. Meng R, Su WJ, Lian XF. Mobile robot path planning based on dynamic fuzzy artificial potential field method. Computer Engineering and Design. 2010;31(7):1558-1561.
17. Khosla A, Kumar S, Ghosh KR. A comparison of computational efforts between particle swarm optimization and genetic algorithm for identification of fuzzy models. In: Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society. San Diego, USA. 2007;245–250.
18. Kim HS, Han SS, Hong SJ. A comparison and analysis of genetic algorithm and particle swarm optimization using neural network models for high efficiency solar cell fabrication processes. In: Proceedings of the 2009 IEEE International Conference on Fuzzy Systems. Jeju Island, South Korea. 2009;1879–1884.
19. Mahjoubi H, Bahrami F, Lucas C. Path planning in an environment with static and dynamic obstacles using genetic algorithm: A simplified search space approach. IEEE Congress on Evolutionary. 2006;2483-2489.
20. Panduro MA, Brizuela CA, Balderas LI, Acosta DA. A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. Progress In Electromagnetics. 2009;Research B 13:171–186.
21. Yue M, Hu T, Guo B, Guo X. The research of parameters of genetic algorithm and comparison with particle swarm optimization and shuffled frog-leaping algorithm. In: Proceedings of. Shenzhen, China. 2009;77–80.
22. Charabaruk N, Manning M, Nokleby S. Dynamic robot path planning using improved max-min ant colony optimization. Proceedings of the International Conference of Control, Dynamic Systems, and Robotics, Ottawa, Ontario, Canada. 2014;49.
23. Zhao J, Fu X. Improved ant colony optimization algorithm and its application on path planning of mobile robot. Journal of Computers. 2012;7(8):2055–2062.
24. Khaksar W, Hong TS, Khaksar M, Motlagh ORE. Sampling-based tabu search approach for online path planning. Advanced Robotics. 2012;26(8–9):1013–1034.

25. Masehian E, Amin-Naseri MR. Sensor-based robot motion planning – A tabu search approach. IEEE Robotics and Automation Magazine. 2008;15(2):48–57.
26. Miao H, Tian YC. Dynamic robot path planning using an enhanced simulated annealing approach. Applied Mathematics and Computation. 2013;222(1):420–437.
27. Zhang H, Liu Y, Guo Z, Cao Y. The dynamic path planning research for mobile robot based on artificial potential field. Proceedings of International Conference on Consumer Electronics, Communications and Networks (CECNet2011). 2011;2736–2739.
28. Yang SX, Max M. Neural network approaches to dynamic collision-free trajectory generation. IEEE Transaction on Systems, Man, and Cybernetics. 2001; 3:302–318. Part B(S1083-4419)
29. Kim JO, Khosla PK. Real-time obstacle avoidance using harmonic potential functions. IEEE Trans. Robot. Autom. 1992;8(3):338–349.
30. Daily R, Bevly D. Harmonic potential field path planning for high speed vehicles. In: American Control Conference, Washington, USA. 2008;4609-4614.
31. Ko NY, Lee BH. Avoidability measure in moving obstacle avoidance problem and its use for robot motion planning. In Proc. IEEE/RSJ Int. Conf. Intell. Robots and Sys. 1996;3:1296–1303.
32. Ni J, Wu W, Shen J, Fan X. An improved VFF approach for robot path planning in unknown and dynamic environments. Hindawi Publishing Corporation Mathematical Problems in Engineering Volume. Article ID 461237; 2014.
33. Sgorbissa A, Zaccaria R. Planning and obstacle avoidance in mobile robotics. Robotics and Autonomous Systems. 2012; 60(4):628–638.
34. Malakar M, Sinha N. A particle swarm optimized potential field method for obstacle avoidance and path planning of mobile robot. International Journal of Digital Application & Contemporary Research. 2013;2(5).
35. Yu Z, Yan J, Zhao J, Chen Z, Zhu Y. Mobile robot path planning based on improved artificial potential field. Journal of Harbin Institute of Technology. 2011;43(1): 50–55.
36. Zhang J, Zhao Z, Liu D. A path planning method for mobile robot based on artificial potential field. Journal of Harbin Institute of Technology. 2006;38(8):1306–1309.
37. Zhang H, Liu Y, Guo Z, Cao Y. The dynamic path planning research for mobile robot based on artificial potential field. Proceedings of International Conference on Consumer Electronics Communications and Networks (CECNet2011). 2011;2736–2739.
38. Sheng J, He G, Guo W, Li J. An improved artificial potential field algorithm for virtual human path planning. Edutainment, Lecture Notes in Computer Science (LNCS). 2010;6249:592–601.
39. Yang Y, Wang S, Wu Z, Wang Y. Motion planning for multi-HUG formation in an environment with obstacles. Ocean Engineering. 2011;38(8):2262–2269.
40. Hwang YK, Ahuja K. Potential field approach to path planning. IEEE Trans. Robot Automation. 1992;8(1):23–32.
41. Shen YQ, Jiang YM. Footprint planning about virtual human based on artificial potential field method. Journal of System Simulation. 2007;16:3796–3799.
42. Shen KH, Jtan YG. A method solving the local minimization problem of artificial potential field. Journal of Wuhan University of Technology. 2005;8:77–79.
43. Guanghui L, Yusuke T, Atsushi Y, Hajime A. Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning. Int. J. Mechatronics and Automation. 2013; 3(3):41-170.
44. Montiel O, Sepulveda R, Orozco-Rosas U. Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field. Journal of Intelligent & Robotic Systems. Published online; 2014.
45. Vadakkepat P, Tan KC, Wang ML. Evolutionary artificial potential fields and their application in real time robot path planning. In: IEEE - Congress on Evolutionary Computation. 2000;256–263.
46. Aarno D, Kragic D, Christiansen HI. Artificial potential biased probabilistic roadmap method. In Proc. of the IEEE Int. Conf. on Robotics and Automation. 2004; 461–466
47. Kazemi M, Mehrandezh M. Robotic navigation using harmonic function-based probabilistic roadmaps, in Proc. of the IEEE Int. Conf. on Robotics and Automation. 2004;4763–4770.

APPENDIX

This Appendix presents the details of the algorithm for implementing the algorithm described in Section 2.3 to avoid local minima or inside the obstacles. Once the robot is in a cavity of an obstacle or in a local minimum, the artificial goals are activated and execute the following steps as long as all feasible paths are found:

Step 1: Let $F_{att}(q_i) = F_{rep}(q_i)$, h_1= 0 , h_2= 0, a_1= 0, a_2= 0, a_3= 0, a_4= 0, $i = 0$

where, h_1 and h_2 are the success index of getting out of the local minima from left and right side respectively, and i is the number of iterations. If the index $h_1 = 1$, it means that the robot has successfully come out of the local minima from the left side. On the other hand, if the index $h_2 = 1$, it means that the robot has successfully come out of the local minima from the right side. The direction of robot movement is depend on the particular combination of artificial goals chosen in a step. The index a_1, a_2, a_3 or a_4 =1 indicates that the robot was not allowed to travel towards a particular direction. That may occur due to the presence of obstacles in that direction.

Step 2: If h_1= 0 & a_1= 0

Go to Step 3

Else

Go to Step 4

Step 3: $k_1 = 0, \quad k_2 = 0, \quad k_3 = -1.5$ and let the robot travel $(3\delta + i \cdot j \cdot R_m)$ distance and then reset $k_1 = 0, \quad k_2 = 0, \quad k_3 = 0$

If (the robot comes out of local minimum):

h_1= 1;

Else if (the robot was not allowed to travel that distance)

a_1= 1;

Else (the robot returns to the same local minimum)

Continue;

Step 4: If h_2= 0 & a_2= 0

Go to Step 5

Else

Go to Step 6

Step 5: $k_1 = -1.5, \quad k_2 = 0, \quad k_3 = 0$ and let the robot travel $(3\delta + i \cdot j \cdot R_m)$ distance and then reset $k_1 = 0, \quad k_2 = 0, \quad k_3 = 0$

If (the robot comes out of local minimum):

h_2= 1;

Else if (the robot was not allowed to travel that distance)

a_2= 1;

Else (the robot returns to the same local minimum)

Continue;

Step 6: If h_1= 0 & a_3= 0

Go to Step 7

Else

Go to Step 8

Step 7: $k_1 = 0, \quad k_2 = -1.5, \quad k_3 = -1.5$ and let the robot travel $(3\delta + i \cdot j \cdot R_m)$ distance and then reset $k_1 = 0, \quad k_2 = 0, \quad k_3 = 0$

If (the robot comes out of local minimum):

h_1= 1;

Else if (the robot was not allowed to travel that distance)

a_3= 1;

Else (the robot returns to the same local minimum)

Continue;

Step 8: If h_2= 0 & a_4= 0

Go to Step 9

Else

Go to Step 10

Step 9: $k_1 = -1.5, \quad k_2 = -1.5, \quad k_3 = 0$ and let the robot travel $(3\delta + i \cdot j \cdot R_m)$ distance and then reset $k_1 = 0, \quad k_2 = 0, \quad k_3 = 0$

If (the robot comes out of local minimum):

h_2= 1;

Else if (the robot was not allowed to travel that distance)

a_4= 1;

Else (the robot returns to the same local minimum)

Continue;

Step 10: If (none of the steps 3, 5, 7 and 9 can be executed):

Exit;

Else

$i = i + 1$;

Go to Step 2;

The value of k_j is chosen to be $k_j = -1.5$ when one or any combinations of the artificial goals are activated, because repulsive force exerted by the artificial goal on the robot created by Equation (12) must be greater than the attractive force; j is chosen to be $j = 0.2$, R_m is calculated to be the largest dimension of the obstacles inside R_1, i is the number of steps (i.e. $i = 0,1,2,3..$), and δ is the step size. In each of the above steps, the imaginary robot will travel $(3\delta + i \cdot j \cdot R_m)$ distance. If it is not possible for the imaginary robot to move along a particular direction, the motion breaks and returns to the same local minimum and that particular direction is skipped in the next iterations. This is done by introducing indices a1, a2, a3 and a4. The imaginary robot will exit the algorithm once all feasible paths are found.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=1141&id=5&aid=9599