Proceeding of the 2013
RSI/ISMInternational Conference on Robotics and Mechatronics
February 13-15, 2013, Tehran, Iran

A New Method for Mobile Robot Navigation in Dynamic Environment: Escaping Algorithm

F. Adib Yaghmaie *Member, IEEE,* A. Mobarhani *Member, IEEE,* and H. D. Taghirad *Senior Member, IEEE,*

***Abstract*—This paper addresses a new method for navigation in dynamic environment. The proposed method is based on force field method and it is supposed that the robot performs SLAM and autonomous navigation in dynamic environment without any predefined information about dynamic obstacles. The movement of dynamic obstacles is predicted by Kalman filter and is used for collision detection purpose. In the time of collision detection, a modifying force is added to repulsive and attractive forces corresponding to the static environment and leads robot to avoid collision. Moreover, a safe turning angle is defined to assure safe navigation of the robot. The performance of proposed method, named Escaping Algorithm, is verified through different simulation and experimental tests. The results show the proper performance of Escaping Algorithm in term of dynamic obstacle avoidance as a practical method for autonomous navigation.**

I. Introduction

Path planning is needed for all mobile robots to perform a predefined task. Usually robots need a collision free path toward the target. It is important to note that the environment is unknown and as a result the robot plans a path while it is moving and gathering data. One of the most common methods for path planning is potential field. In this method, obstacles exert repulsive force to the robot while the target provides attractive force to it. The robot moves toward target avoiding obstacles based on the resultant of attractive and repulsive forces. This method is introduced by [1] and then improved for real time implementations [2]. Potential field is a popular method for mobile robot navigation due to its simplicity and implementation.

First, potential Field approach was suggested for navigation in static environment [3], [4], [5], [6]. However, the real world is not stationary, and the robot moves in dynamic environment and encounters dynamic obstacles like moving humans. Hence it should plan to reach the target without colliding with any static or dynamic objects. Researchers started to develop potential field methods for navigation in dynamic environment. In [7], the velocity of dynamic obstacles are included in the definition of potential function. The basic problem is that the collision depends on the velocity of both robot and obstacle; however, [7], considers only the speed of robot. In [8] relative positions and velocities of mobile robot with respect to obstacles are considered in definition of potential function. However, this method needs exact knowledge of velocity of dynamic objects, which is not available in practice. Potential function to reach a moving target is defined in [9], but the velocities of robot, obstacles and target are assumed to be known.

In this paper, a different approach is suggested for velocity planning of a mobile robot to navigate in dynamic environments based on potential field approach. The proposed method is called Escaping Algorithm (EA), and does not need any pre-knowledge information about dynamic obstacles. This method uses Kalman filter to predict the motion of dynamic objects and combines them with potential field approach to navigate safely in dynamic environment. EA has been simulated several times and implemented on a mobile robot platform. The reported results in this paper show the effectiveness of EA in term of dynamic obstacle avoidance and target tracking without need of expensive computations.

This paper is organized as follows: Section 2 provides environment representation and formulates the problem. The definitions of attractive and repulsive force in static environment are mentioned in this part. In section 3, escaping algorithm and its components are enlightened and the required steps to obtain repulsive force of dynamic obstacles are discussed. Simulation and experimental results of EA in dynamic environment are represented in section 4, and finally, concluding remarks are given.

Authors are with Advanced Robotics and Automated Systems (ARAS), Industrial Control Center of Excellence (ICCE), Faculty of Electrical and Computer Engineering, K. N. Toosi University of Technology, E-mail: farnaz_adibyaghmaie@ieee.org, amir.mobarhani@ieee.org, taghirad@kntu.ac.ir.

II. The Environment Modeling and Problem Formulation

A. The Environment Modeling

In this paper, the popular occupancy grid map is used for environment modeling [10]. The occupancy grid map uses a matrix to represent obstacles. Each entity in matrix is a symbol of one square part of environment and its quantity shows the confidence of the obstacle lying at this location. In this context, the size of square sides is set to $10cm$. Certainty values range from $-\infty$ to $+\infty$ in occupancy grid maps. As the possibility of existence of an obstacle in specific square increases, the certainty values goes to $+\infty$; while for a free cell, this value approaches to $-\infty$. The robot is equipped by one laser range finder and two encoders. Since laser is used to obtain information from environment, this map is suitably adapted to our system. In each range reading, the values of certainties are updated and used for navigation purposes.

B. Problem Formulation

The motion planning problem of a mobile robot is to plan and control it such that it reaches the target while avoiding obstacles. In Force Field Method (FFM) the obstacles exert repulsive force to the robot while the target attract the robot to itself. The total force determines the direction of movement for the robot. The definitions of attractive and repulsive forces are not unique, and different definitions can be found in [11] and [9].

To use FFM for the mobile robot navigation, it is enough to suitably define the repulsive force of static objects, the attractive force of target and the repulsive force of dynamic objects and use their total direction for mobile robot steering. The last component will be explained in next section with more details.

1) The Repulsive Force of Static Objects:

Since gird map is used for environment representation, the repulsive forces of static objects are defined for each occupied cell in grid map. To accomplish this task, it is needed to define a window on the robot coordinates and calculate the repulsive forces through it. This window is called active window and is used to avoid local obstacles while moving toward the target. For each cell in active window the repulsive force of static cell is calculated as follows.

$$f_r(i,j) = -\frac{F_{cr}C(i,j)}{d_{i,j}^2}\left[\frac{x_{i,j}-x_r}{d_{i,j}}\hat{x} + \frac{y_{i,j}-y_r}{d_{i,j}}\hat{y}\right] \quad (1)$$

in which, F_{cr} is the repulsive force constant, $C(i,j)$ denotes the certainty of cell(i,j), $d_{i,j}$ denotes the distance between the robot and cell(i,j), $(x_{i,j}, y_{i,j})$ is the position of cell(i,j), and (x_r, y_r) is the position of the robot. Total repulsive force is calculated by adding all repulsive forces in active window:

$$F_r = \sum_{i,j\in \text{win}} f_r(i,j) \quad (2)$$

2) The Attractive Force of the Target:

The robot moves toward the target while avoiding obstacle. As a result, it is not important that whether the target is located in the active window or not, it always exerts its attractive force to the robot by the following relation:

$$F_a = F_{ca}\left[\frac{x_t-x_r}{d_t}\hat{x} + \frac{y_t-y_r}{d_t}\hat{y}\right] \quad (3)$$

in which, F_{ca} denotes attractive force constant, (x_t, y_t) is the target position, and d_t denotes the distance between the robot and the target.

3) Total Force:

Definition of repulsive force of dynamic objects denoted by F_m and required steps are explained in details in the next section.

The total force is determined as the sum of the repulsive and attractive forces. The total force in static environment is derived from following equation:

$$F_t = F_r + F_a \quad (4)$$

The above force is used for avoiding local static objects while moving toward the target. If a collision is predicted, the repulsive force of dynamic object is added to the total force and the resultant force is used for navigation.

$$F_f = F_t + F_m \quad (5)$$

The direction of F_f is used as steering rate command as follows. In static environment, F_m is equal to zero; in the other words F_f and F_t are the same. Let δ shows direction of F_f. If robot direction is denoted by θ, angular velocity can be given by:

$$\omega = K_s(\delta - \theta) \quad (6)$$

in which, K_s is the steering constant whose dimension is (s^{-1}). This constant is set as the inverse of the sampling time.

III. Escaping Algorithm: A Strategy for Navigation in Dynamic Environment

In static environment, the mobile robots can reach the target by using repulsive forces of static objects and attractive force of the target. However, it needs to perform four steps sequentially, to move in dynamic environment safely. These steps are as follows: moving objects detection, motion prediction, collision detection and velocity planning for obstacle avoidance. These sub programs are executed within the Simultaneous Localization And Mapping (SLAM) algorithm and use the grid map obtained from SLAM in their calculations. Each sub program is explained in following section with more details.

A. Moving Objects Detection

Moving object detection is one of the most important parts of planning in dynamic environment. The objective is to classify observations as static or dynamic. Researchers develops several methods for this classification. One common method is Expectation Maximization Algorithm (EMA) [12], [13]. EMA is a two steps maximization process which solves incomplete data optimization problem [14]. Another method is sample-based variant of probabilistic data association filter. This method filters dynamic observations like human and results robust scan matching [15]. Besides, non-probabilistic methods are also developed. For example, [16] suggested a simple rule for classification. This method is extended for grid map and is used for dynamic observation mapping in this paper.

In this paper a three-state map is generated and used for dynamic object detection. The three-state map has similar structure to grid map and represents environment by set of cells. Each cell in this map can be labeled as free, occupied or unknown. A reading is associated to dynamic object if it locates in a free cell. The three-state map is generated by following formula at each SLAM loop:

$$c_d(i,j) = \begin{cases} \text{free} & \text{if } c(i,j) < c_{min} \ \ (c_{min} < 0) \\ \text{occupied} & \text{if } c(i,j) > c_{max} \ \ (c_{max} > 0) \\ \text{unknown} & \text{otherwise} \end{cases} \quad (7)$$

In above, $c_d(i,j)$ shows the cell (i,j) in three-state map. The values of c_{min} and c_{max} are tuned practically. Using three-state map, observations are divided to dynamic and static. Static observations are used in gird map, while dynamic observations are predicted and special strategy (EA) is used to avoid them. Please note that several sequential dynamic observations refer to of one moving objects because moving objects like human reflects several beams of laser to the robot. As a result, it is required to group sequential dynamic readings. The center of each group is used as pose of dynamic object and the distance between center and dynamic border is considered as obstacle radii, R_{obs}.

B. Motion Prediction

In this context, a set of Kalman filters is defined. Each Kalman filter predicts next poses and velocities of one moving object. The state vector of each Kalman filter is defined as $X = [x, \dot{x}, y, \dot{y}]$. The initial guess of state vector is set to $X = [0, 0, 0, 0]$ for all moving objects. As it can be seen, the constant velocity model is used to represent moving object movement. It is important to note that movement of moving obstacles especially human is unpredictable; however, the constant velocity model with noisy acceleration may be suitably used to predict this behavior [12]. The discrete space state equation is shown as follows:

$$X_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{k-1} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} n_{w_1} \\ n_{w_2} \end{bmatrix} \tag{8}$$

In which T, denotes the sampling time. Dynamic readings which obtained from previous routine are considered as new positions of moving objects. Therefore, the observation equation is as follows.

$$Z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} X_k + \begin{bmatrix} n_{v_1} \\ n_{v_2} \end{bmatrix} \tag{9}$$

In (8) and (9), n_w and n_v are process and observation noises. The variance of process noises have to be carefully tuned in practice to provide desire performance in tracking. The variance of observation noises is related to sensor properties and should be determined based on them. In each sensor range readings, observations are classified into static and dynamic. Dynamic observations are used for updating Kalman filters. Since there are several dynamic observations and moving object, nearest neighbor algorithm is used to match dynamic observations and obstacles. If there is not any observation for one moving obstacle, our algorithm only performs the prediction step. If this happens several times, it means that the obstacle moves out of robot vision and it is not necessary to predict its motion anymore. As a result the corresponding filter is eliminated.

A Kalman filter is also used for the robot position and velocity prediction. The prediction of this filter is used for collision detection between robot and obstacles. Therefore,

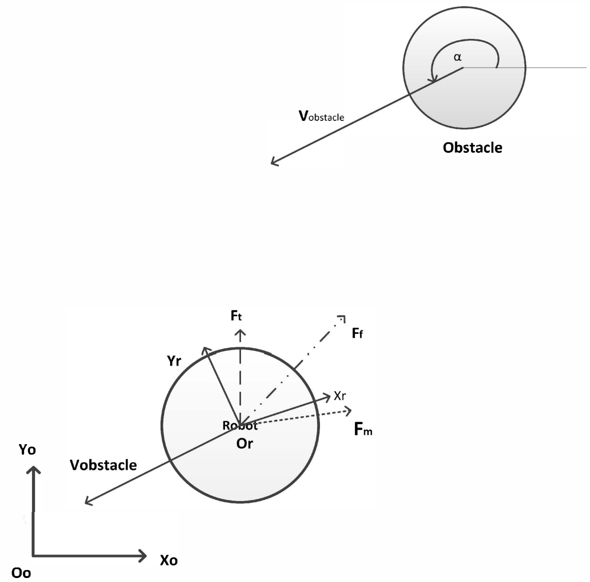

Fig. 1: Escaping algorithm

in each SLAM loop, the following set of Kalman filters is updated.

$$\{(X_k, Z_k)\}_i \qquad i = 1, ..., N+1 \tag{10}$$

In this equation, N denotes the number of predictable moving objects, while the last Kalman filter is for the estimation of robot pose in future. By this means, N may differ at each iteration, according to the number of visible dynamic obstacles.

C. Collision Detection

The set of Kalman filters and their predictions are used to find possible collisions. To perform that, each Kalman filter predicts the next poses and velocities of one dynamic obstacle up to max predictions Time, T_{max}. Similar prediction is done for the robot. Moreover, suppose that dynamic obstacles and the robot can be modeled by circles with radius R_{obs} and R_{rob} respectively. In kth prediction step the distance between the robot pose and the obstacle is calculated. If this distance, denoted by d, is less than the summation of obstacle and robot radii, then the collision will probably happen. The distance is calculated between the robot and all moving obstacles in order to find all possible collisions. It is important to note that as prediction step increases, the uncertainty grows, as well. As a result, a confidence factor, denoted by λ_{conf}, is defined and used in collision detection. The following condition holds if a collision is possibly happening:

$$d_k < \lambda_{conf}(R_{rob} + R_{obs}) \qquad \lambda_{conf} > 1 \tag{11}$$

D. Velocity Planning for Obstacle Avoidance

Navigation in dynamic environment using potential field method is widely studied in literature. In this method, the target exerts attractive force to the robot, while static and dynamic objects apply repulsive forces. There are different definitions for the repulsive and attractive forces. For example in [9] a moving target is considered and the repulsive and attractive forces are derived by supposing full knowledge of the target and obstacle's positions and velocities. The final output of method suggested in [9] is a function of the target velocity and the relative positions between the obstacles, the target and the robot. Similarly, by considering full knowledge about moving objects and the target, in [8] relative distance and velocity between the robot and the obstacle are used in repulsive force definition. The attractive force is defined based on relative distance and velocity between the robot and the target. One key problem of this method is that they need exact knowledge of position and velocity of dynamic obstacle and the target. However, this is an unrealistic assumption as none of them is known in practice.

In this paper Escaping Algorithm (EA) is suggest for obstacle avoidance. This algorithm is originated from a common behavior of human. A person usually intends to move in opposite direction in order to avoid colliding with a moving person. The same strategy can be used for mobile robot navigation. In our approach, the robot tries to moves in opposite direction if possibility of collision is detected. Consider Fig. 1. In this figure, obstacle direction in global frame is denoted by angle α.The projection of this direction in robot frame is a suitable direction to align a new repulsive force. In order to perform that, the velocity of obstacle is expressed in the robot frame by the following rotation.

$$V_{obs}^{r} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} V_{obs}\cos(\alpha) \\ V_{obs}\sin(\alpha) \end{bmatrix} = \begin{bmatrix} V_{obs}\cos(\alpha-\theta) \\ V_{obs}\sin(\alpha-\theta) \end{bmatrix} \quad (12)$$

In Fig. 1, F_t denotes the total force considering static environment. The obstacle velocity which is projected in the robot frame is used to define modifying force F_m. This new force has the same norm as F_t and its direction is opposite to the obstacle direction in x, and parallel to it in y direction. Any time that possibility of a collision is detected, the modifying force F_m is added to the total force and the final force F_f, is used for steering the robot toward the target.

To cosider the safety of motion, consider Fig. 2. As it is mentioned before, it is supposed that the robot and the obstacle can be modeled by circles with radius R_{obs} and R_{rob} respectively. It is similar to consider the robot as a point and enlarge the radius of obstacle to $R_{obs} + R_{rob}$. As a result the minimum turning angle for the robot is φ to avoid collision. This angle is shown in Fig. 2 and may be derived by the following equation.

$$\varphi = arcsin(\frac{R_{rob} + R_{obs}}{d}) \quad (13)$$

In the above equation, R_{rob} is a known parameter and can be measured before test. R_{obs} is calculated as it is explained in section 3-A. To assure safety, if the turning angle $(\delta - \theta)$, is less than φ, the the turning angle φ is considered for robot steering.

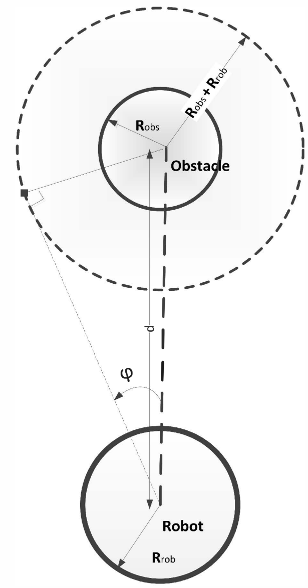

Fig. 2: Definition of safe turning angle

IV. Result

In this part the simulation and experimental results of using EA in dynamic environments are presented. It is important to note that navigation by using force field family suffers from trapping in local minima. This problem occurs when the robot direction differs more than 90 degrees off target or F_f is equal to zero.

$$|\delta - \theta| > \frac{\pi}{2} \quad (14)$$

$$|F_f| = 0 \quad (15)$$

Researchers develop several recovery methods to encounter local minimum problem. For example recovery methods based on electromagnetic field and modification of repulsive potential functions are suggested in [17] and [8] respectively. One of the most popular recovery methods is Wall Following Method (WFM) which is suggested in [1]. In this context, WFM is used for mobile robot navigation to avoid trapping in local minima.

A. Simulation Result

In this section, simulation result of using EA in a dynamic environment is described. For simulation, seven dynamic objects are considered whose positions and motion directions are randomly selected. The robot does not have any prior knowledge about moving objects and their paths are estimated by Kalman filter. The robot obtains its observation from global map shown in Fig. 3. In this figure, the start point, the target point and seven dynamic objects are shown. The robot navigates from the start point to the target point successfully, provided that it does not collide with static and dynamic objects.

It is possible that several collisions are detected at some occasions by collision detection algorithm. In thess situations, only the nearest threat is considered and others are neglected. After eliminating the nearest danger, the next hazardous collision is considered. In Fig. 4 the performance of EA in dynamic environment is depicted. As it is seen in this figure, the robot starts moving to the target in (a). The wall w_0 bans the robot path. As a result, the robot follows it. The robot faces a moving object in (b) and the modifying force is executed and causes the robot to move in opposite direction of the moving object. When the threat of collision is removed, the robot turns to the target in (c). However, appearance of another moving object causes the robot to turn off the target again to avoid it. This movement is depicted in (d). After using EA for dynamic obstacle avoidance, the robot turns to the target and moves toward it in (e). Finally, the robot reaches the target in (f).

Fig. 4 shows only one of several simulation tests. Due to the fact that moving objects are located and directed randomly, almost all scenarios have been tested on the robot. In all of them, the robot was successful in its mission to reach the target and does not collide with any object trough its movement. These set of simulations provides us enough assurance to implement this algorithm on the following real time experiments.

B. Experimental Results

EA for navigation in dynamic environment is implemented on KNTU Mellon mobile robot. The mobile robot perceives environment through a laser range finder whose maximum range reading is 8 meter. The laser scan data is used for map building in a SLAM environment. The ego motion estimation is done by well-known ICP algorithm [18]. To achieve more accurate result in robot localization the odometry information by encoders mounted on two wheels, serves as the initial guess for ICP algorithm. A computer with core i5 processor is used for online execution of the algorithm. The algorithm contains a loop for simultaneous localization and mapping (SLAM) in addition to robot navigation routine.

For the control of Mellon mobile robot two commands are prompted, namely the velocity of the right and left wheels,

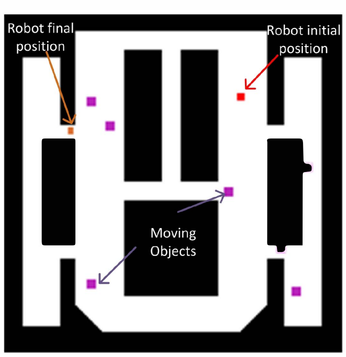

Fig. 3: The global map in which the robot reads observations

denoted by ω_r, and ω_l, respectively. These velocities are easily obtained by linear transformation of linear and angular velocites as bellow:

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & \frac{b}{2} \\ 1 & \frac{-b}{2} \end{bmatrix} \begin{bmatrix} V \\ \omega \end{bmatrix} \tag{16}$$

where, r and b denote radii of wheel and wheel base, respectively. The angular velocity (6) is calculated by considering (5) in EA. Different methods are suggested by researchers [9], [2] for linear velocity definition. In this context, since fast performance of the mobile robot is desired, it is good that the mobile robot moves as fast as possible. However, it should start deceleration before an obstacle appears in its path in order to avoid collision. Hence, it is suitable that the mobile robot moves with its maximum linear velocity until it reaches the region that should decelerate to stop completely near an obstacle. Hence,

$$V = \begin{cases} V_{max} & \text{if } d_{min} > d_{eff} \\ \frac{d_{min}}{d_{eff}} V_{max} & \text{otherwise} \end{cases} \tag{17}$$

In (17), d_{min} is the minimum range reading in range scanning and d_{eff} is the distance passed by the robot when it decelerates from maximum velocity to zero.

In what follows, experimental result of using EA in dynamic environment is given. In this experiment, the target is located in front of the robot with a relative distance of $4m$. While the robot tries to reach target with a distance less than $30cm$. In the terms of static environment, the robot goes straight to the target until it approaches static obstacle and tries to avoid it. However, the robot motion in dynamic environment is different. The dynamic scenario is run twenty two times and the result of one of them is shown in Fig. 5

To examine the effect of dynamic environment, two moving objects approach the robot from the left. The robot tries to find a collision free path for safe navigation. As indicated in Fig. 5 and in part (a), the robot starts moving toward the target, while in (b) a moving obstacle appears and in (c) a collision possibility is detected. In (d) the robot turns left to avoid collision and again in (e) it moves back toward the target. Moreover, in (f) another collision possibility is detected and the modifying force is added to the total force, and hence, the robot turns to the left to avoid collision in (g). Finally, in (h) it reaches the target. Robot path using EA in this experiment can be observed in more detail in Fig. 5b. From the promising results observed in the set of twenty two experiments, we may conclude that EA algorithm is suitable to be used in further development of autonomous robots.

V. Conclusion

In this paper, Escaping Algorithm is suggested for navigation in dynamic environments. This method introduces modifying force in addition to attractive and repulsive forces in a force field environment. EA does not need exact information such as velocity and position of dynamic obstacles. These properties are estimated by Kalman filter and the obtained

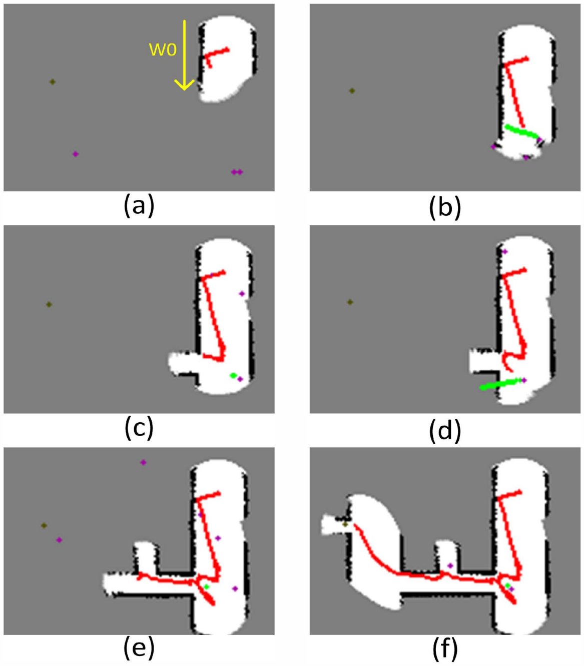

Fig. 4: Robot path in dynamic environment using EA

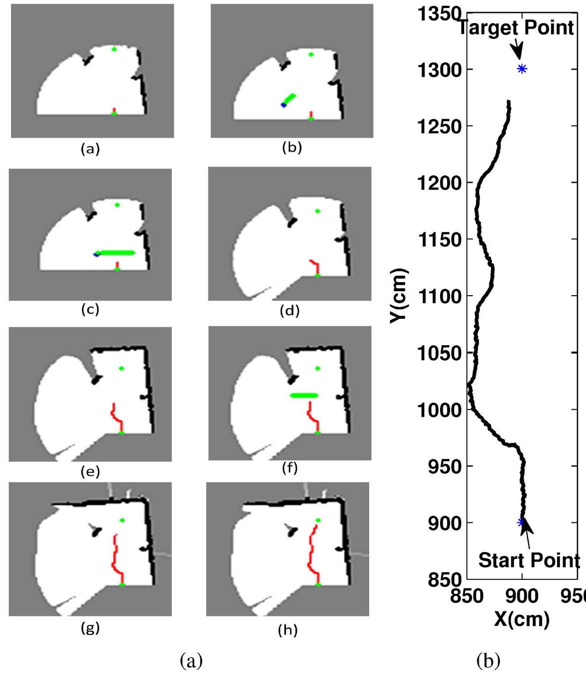

Fig. 5: EA experimental result in a real dynamic environment

results are used in defining the modifying force. The basic idea of EA is obtained from a common behavior of human in obstacle avoidance and it is turning object from behind. The general performance of proposed method is checked through several simulation in U shape environment with a different number of dynamic obstacles. Furthermore, the experimental results show the effectiveness of proposed method in practical implementation. Finally, it can be concluded form these observations that this method is successful in term of dynamic obstacle avoidance. Comparison between EA and other similar methods will be presented in future works.

References

[1] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[2] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fact mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, pp. 1179–1187, 1989.

[3] J. Chuang and N. Ahuja, "An analytically tractable potential field model of free space and its application in obstacle avoidance," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 28, no. 5, pp. 729–736, 1998.

[4] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[5] J. Kim and P. Khosla, "Real-time obstacle avoidance using harmonic potential functions," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 338–349, 1992.

[6] P. Veelaert and W. Bogaerts, "Ultrasonic potential field sensor for obstacle avoidance," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 4, pp. 774–779, 1999.

[7] N. Ko and B. Lee, "Avoidability measure in moving obstacle avoidance problem and its use for robot motion planning," in *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1996, pp. 1296–1303.

[8] S. Ge and Y. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

[9] L. Huang, "Velocity planning for a mobile robot to track a moving target–a potential field approach," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 55–63, 2009.

[10] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 34–46, 2007.

[11] S. Ge and Y. Cui, "New potential functions for mobile robot path planning," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 5, pp. 615–620, 2000.

[12] L. Montesano, J. Minguez, and L. Montano, "Modeling dynamic scenarios for local sensor-based motion planning," *Autonomous Robots*, vol. 25, no. 3, pp. 231–251, 2008.

[13] S. Thrun, C. Martin, Y. Liu, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard, "A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 433–443, 2004.

[14] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. Wiley New York, 1997, vol. 274.

[15] I. Cox, "A review of statistical data association techniques for motion correspondence," *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[16] M. Lindstrom and J. Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2001, pp. 1364–1369.

[17] V. Petridis and T. Tsiboukis, "An optimal solution to the robot navigation planning problem based on an electromagnetic analogue," *Robotic systems: advanced techniques and applications*, vol. 10, pp. 297–303, 1992.

[18] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 2, pp. 239–256, 1992.