Reactive Algorithm for Mobile Robot Path Planning Among Moving Target/Obstacles by Means of Dynamic Virtual Obstacle Concept

Hussein Hamdy Shehata*, Josef Schlattmann

Systems Technology and Design Methodology
Hamburg University of Technology
21073 Hamburg, Germany

Abstract

It is stubborn to handle with dynamic environments, which are totally or partially unknown or even dynamically changing. Furthermore, most researchers have focused on solving the path planning problem in a stationary environment. Several algorithms have been proposed and among such algorithms, potential fields are the core of a class of effective navigation schemes for autonomous robots. This paper proposes an improved algorithm for a mobile robot path planning in a dynamic environment where the target and obstacles are moving. This algorithm relies on the integration between a dynamic virtual obstacle concept and a robot size factor, which represents a threat factor to any possible collisions and enables the robot to plan its motion not only with right positions, but also with reasonable velocities. The relative position of the dynamic virtual obstacle along with the proper choice of the robot size factor has the role of adjusting the robot's velocity. The convergence of this proposal is discussed to guarantee its reliability, and the simulation results have proven its feasibility and validity in partially dynamic as well as dynamic environments.

1. Introduction

Autonomous navigation and path planning of a mobile robot in dynamic environments still represent a challenge for real applications. The motion planning problem in dynamic environments is to plan the robot motion to follow a moving target in a desired behavior while avoiding moving obstacles [1]. Autonomous navigation, in general, assumes an environment with known and unknown obstacles, and it includes global path planning algorithms to plan the robot's path among the known obstacles, as well as local path planning for real-time obstacle avoidance [2]. However, the environment is dynamic in many real applications. This means not only the obstacles are moving, so does the target. Therefore, this article concentrates on the local obstacle avoidance aspect, as well as ensures the reachability to the moving target.

In the last decades, the literature focused on path planning in static environment where the target and the obstacles are stationary. Many methods were proposed, however these methods are no longer suited to applications in dynamic environments. As a result of the workspace is dynamic in many real-life applications, the latter studies focused on path planning in dynamic environments. In general, there are two main categories in solving the problem of path planning, which are Artificial Intelligence (AI) algorithms and Artificial Potential Field (APF) algorithms. AI techniques can be used when the available information on the environment is hazy. These techniques are based on tools like neural networks [3], which in turn are interconnected with optimization algorithms resulting in high computational complexity and limit the real-time implementations. On the contrary, APF approaches are much convenient than AI approaches due to its high efficiency in autonomous navigation, elegant mathematical analysis and simplicity [2]. In addition, APF was originally developed as an *on-line* collision avoidance approach, applicable when the robot does not have a-priori model of the obstacle, but senses them during motion execution. Regarding the conventional APF in the literature [4-6], a robot is usually treated as a point and moves in a two-dimensional workspace under the influence of an APF whose local variations are expected to represent the structure of the free space. The negative gradient of the potential field generates appropriate attractive and repulsive forces to facilitate target reaching and guarantee obstacle avoidance. However, APF has some limitations, such as Local Minimum Problem (LMP) and Goal Nonreachable with Obstacles Nearby (GNRON) [2, 6]. Varieties of potential functions

* Corresponding author: Tel.: +49(0)40-42878 4430; Fax: +49(0)40 42878 4076; E-mail: shehata@tu-harburg.de

have been proposed in the literature to deal with these problems. In comparison with the conventional APF method, these functions afford robust and better performance, but still inapplicable for fast and real-time path planning in dynamic environments. It is stubborn to handle with such environments, which are totally or partially unknown or even dynamically changing. In an attempt to solve the problem of path planning in a dynamic environment, Ge and Cui [1] and Huang [7] incorporated the velocities of the target and the obstacle in their APF. Poty et al. [8] merged the proposed approach in [1] and the fractional potential for dynamic motion planning of a mobile robot. Munasinghe et al. [9] presented the velocity dipole field and its integration with the conventional APF to form a new real-time obstacle avoidance algorithm. Prassler et al. [10] developed a velocity obstacle method for obstacles moving on arbitrary known trajectories and applied for *on-line* navigation of the robotic wheelchair. Sugiyama et al. [11] proposed a hydrodynamic potential function to guide the mobile robot towards the target while avoiding moving obstacles. Conn and Kam [12] produced another approach that includes the time as one of the dimensions of the model and thus the moving obstacles can be regarded as stationary in the extended world. Yin and Yin [13] incorporated the velocity and the acceleration of both the target and the obstacles in the potential functions to deal with the dynamic environments. Qixin et al. [14] merged a threat coefficient of the moving obstacles in the APF. Luh and Liu [15] presented a potential field immune network for dynamic navigation in an unknown environment, which was adopted on a velocity obstacle method. Other interesting issues related to tracking multiple moving targets were presented by Schulz et al. [16] and Grundel [17].

Despite the fact that significant results on the path planning problems have been acquired in the above literature, the problem of path planning in uncertain dynamic environments has not been fully investigated. Therefore, this paper proposes an improved potential function for a mobile robot path planning in a dynamic environment where the target and obstacles are moving. As well as the simulated results obtained in our former research [2] encourage us to extend this algorithm in a dynamic environment.

2. PROBLEM FORMULATION

In most real applications, the environment is dynamic. This means not only the obstacles are moving, so does the target. In such situations, it is intuitive to think about the relative velocity of the robot with respect to the obstacles when dealing with collision avoidance. Though the velocity of the obstacle was considered by Ko and Lee [18], and Hussien [19], the velocity of the robot was not taken into account in [18], while [19] assumed that the relative velocity of the robot is invariant regardless of the position of the robot. Both works [18, 19] dealt with the obstacle avoidance problem with a stationary target. Another contribution assumed that the trajectories of the moving obstacles are known a priori [12]. Although Ge and Cui [1] took the velocities of the robot, the target, and the obstacles into account, the robot still has a trend to go apart with the target if their accelerations are different in the final position. Furthermore, from our point of view, it is obviously clear that the robot has a certain size and should not be treated as a point. All these assumptions, which are impracticable, have restricted the use of the potential field in real applications and encouraged us to present this proposal.

3. DYNAMIC VIRTUAL OBSTACLE CONCEPT (DVOC)

Numerous researchers who dealt with the dynamic environments took the velocity and acceleration of both the target and the obstacles in their considerations. While we are in total agreement with their points of view, we disagree with the incorporation of the absolute velocity and acceleration of both the target and the obstacles, which resulting in high computational complexity and limit the real-time implementations, especially in unknown and uncertain environments. To tackle all these unrealistic problems, we present a new function entitled as **D**ynamic **V**irtual **O**bstacle **C**oncept (DVOC), which incorporates not only the size of the robot as a main part of the planning problem but also the relative position of a virtual obstacle in the workspace, which is the responsible for adjusting the robot's velocity.

3.1 CONSTRAINTS, PRECONDITIONS AND ASSUMPTIONS

Generally, there are no constraints or preconditions on the velocity of either the target or the obstacles except that the maximum robot's velocity should be higher than the velocity of both the target and the obstacle. This proposal does not require that the position and the velocity of the robot, the target, and the obstacles are predefined, but senses them during motion execution. No more than accurate *on-line* measurements, which can be acquired by using a laser range finder. In order to clarify the analysis, this proposal assumes some assumptions as follows:

1. The obstacles are convex polygons whose shapes and positions can be accurately measured *on-line*.
2. The mobile robot can move in any direction smoothly, i.e., there are no constraints on the steering angle.

3.2 The Proposed DVOC

The proposed DVOC basically relies on a dynamic virtual obstacle, which has the role of adjusting not only the speed of the robot but also the steering angle. In our algorithm, we imagine that there is an additional force associated with an additional virtual obstacle in the workspace. The positions of the robot and the target are denoted by $q_r = [x_r\ y_r]^T$ and $q_t = [x_t\ y_t]^T$, respectively. While the positions of the obstacles and the virtual obstacle are denoted by $q_{o_i} = [x_{o_i}\ y_{o_i}]^T$ and $q_{vo} = [x_{vo}\ \ y_{vo}]^T$, respectively, where *i* = 1, 2,….*n*, and *n* is number of obstacles. The position of the virtual obstacle is located along the extension line from the target to the robot and in the opposite side of the target, i.e., it enhances the movement toward the target. This virtual obstacle exerts a virtual repulsive force, $f_{vo}(q_r)$, toward the robot. The reasonable virtual repulsive force should be configured as follows: a) it should increase as the robot gets closer to the virtual obstacle in order to enhance the fast maneuvering, b) it should increase quickly in some range of distance so that the robot can avoid collision successfully, c) it should increase slowly outside this range, d) it should decrease as the robot gets closer to the target in order to allow a soft landing on the target, and e) it should have no influence on the robot's path if either the obstacle or the target is far enough from the robot. Accordingly, the need for the virtual obstacle must be a function of the nature and status of the obstacles and the target in the workspace, i.e., the positions of both the target and the obstacles in the workspace efficiently determine the position of the virtual obstacle in the workspace. Consequently, the magnitude of $f_{vo}(q_r)$ is inversely proportional to $\rho(q_r, q_{vo})$, where $\rho(q_r, q_{vo})$ is the minimal distance from the robot to the virtual obstacle, and should be initiated if and only if $\rho(q_r, q_{vo}) \le \rho_o$, where ρ_o is a positive constant denoting the distance of influence of the obstacle. This means outside the virtual obstacle's region of influence; the virtual repulsive force $f_{vo}(q_r)$ equals zero. When $q_r \neq q_t$, the repulsive force of the *i*th obstacle, $f_{rep_i}(q_r)$, and $f_{vo}(q_r)$ are presented as follows:

$$f_{rep_i}(q_r) = \begin{cases} \eta\left(\dfrac{1}{\rho\left(q_r, q_{o_i}\right)} - \dfrac{1}{\rho_o}\right)\left(\dfrac{1}{\rho^2\left(q_r, q_{o_i}\right)}\right)\nabla\rho\left(q_r, q_{o_i}\right), & \text{if } \rho\left(q_r, q_{o_i}\right) \le \rho_o \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$f_{vo}(q_r) = \begin{cases} \eta\left(\dfrac{1}{\rho(q_r, q_{vo})} - \dfrac{1}{\rho_o}\right)\left(\dfrac{1}{\rho^2(q_r, q_{vo})}\right)\nabla\rho(q_r, q_{vo}), & \text{if } \rho(q_r, q_{vo}) \le \rho_o \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where η is a positive scaling number, $\rho(q_r, q_{o_i})$ is the minimum distance from the robot to the *i*th obstacle, and $\nabla\rho\left(q_r, q_{o_i}\right)$ and $\nabla\rho(q_r, q_{vo})$ are two unit vectors pointing from the *i*th obstacle to the robot and from the virtual obstacle to the robot, respectively.

If $\rho\left(q_r, q_{o_i}\right) > \rho_o$, the repulsive force is not defined, since the collision avoidance mechanism is not necessary. If $\rho(q_r, q_{vo}) > \rho_o$, the virtual force is not defined; this means that at least the target or the obstacle is too far from the robot. Both cases do not represent a problem, since most problems are coming when both the robot and the target are within the influence range of the obstacle. Consider the case where the robot and the obstacle are moving toward each other, it is noted that $f_{vo}(q_r) > f_{rep_i}(q_r)$ in some range of distances, while $f_{vo}(q_r) < f_{rep_i}(q_r)$ if the robot is close enough to the obstacle. The first condition, $f_{vo}(q_r) > f_{rep_i}(q_r)$, enhances the movement toward the obstacle, while the second condition, $f_{vo}(q_r) < f_{rep_i}(q_r)$, enhances the movement away from the obstacle. This may lead to some kind of oscillations or collision with the obstacle's boundaries. To avoid this phenomenon, we should dampen the oscillatory motion. In order to achieve this objective, the virtual force must be optimized to keep a proper safety margin between the robot and the obstacle. This safety margin depends on the size of the robot, so the virtual force must integrate the size of the robot.

3.3 The Significance of The Robot Size Factor

A particularly novel feature of this work is the incorporation of the robot size factor, λ. The value of λ is linearly proportional to the size of the robot, while the magnitude of $f_{vo}(q_r)$ is inversely proportional to λ. It is defined as a positive real number belongs to $\lambda \le 1$. When the obstacle is far enough from the robot, so there is no threat on the robot, λ is set to minimum but not less than the actual size of the robot, i.e., λ = 0.1. If the obstacle is close enough to the robot, then λ = 1. This means; the proper choice of λ will produce not only a proper virtual force but also a reasonable speed. We define a safety margin between the robot and the obstacle, which in turn depends on the difference $(\lambda - \text{maximum normalized robot size})$, which represents a safety index. As λ increases, as the safety margin increases. Therefore, the robot size factor is considered a threat factor to any possible collisions. The reader has to notice that we consider only the size of the existing robots, which are around 30-100 cm diameter and are used in many daily life applications such as a general-purpose mobile humanoid robot, a mobile robot for hospital work, and an assistive mobile robot. Through different complex scenarios, which are extensively tested, we constructed a calibration curve between λ and the robot's diameter. The calibration curve is shown in Figure 1. Apparently, the safety index is always 0.2 for hard landing and 0.4 for soft landing. Figure 2 shows the influence of the robot size factor on the safety margin. The modified virtual force, $f_{vo}(q_r)_{mod}$, is presented in (3).

$$f_{vo}(q_r)_{mod} = \begin{cases} \eta \left(\frac{1}{\lambda} \cdot \frac{1}{\rho(q_r,q_{vo})} - \frac{1}{\rho_o}\right)\left(\frac{1}{\lambda^2} \cdot \frac{1}{\rho^2(q_r,q_{vo})}\right) \nabla\rho(q_r,q_{vo}), & \text{if } \rho(q_r,q_{vo}) \le \rho_o \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Apparently, $f_{vo}(q_r)_{mod}$ synthesizes the effect of the robot size as well as the relative position of the virtual obstacle, which in turn depends on the positions of the target and the obstacles as we will explain in section 3.4.

The introduction of λ has a tri-effect. It defines not only a safety margin of possible physical collision, but also can be used to accommodate any possible measurement uncertainty. Sometimes the applications require a soft landing on the target where the robot's velocity is the same as the target's velocity at landing, other times a hard landing is required where there is no restriction on the robot's velocity, especially in a hazardous area. All these requirements can be acquired by controlling the robot size factor, which plays a significant role in controlling the robot's velocity. One may use a large value of λ for the following cases:

1. When the uncertainty of the measurements is large.
2. When a high safety margin is needed.
3. When a soft landing on the target is needed.

While a small value of λ is desirable for the following cases:

4. When the measurements are accurate.
5. When a small impact collision is allowed.
6. When a hard landing on the target is needed.

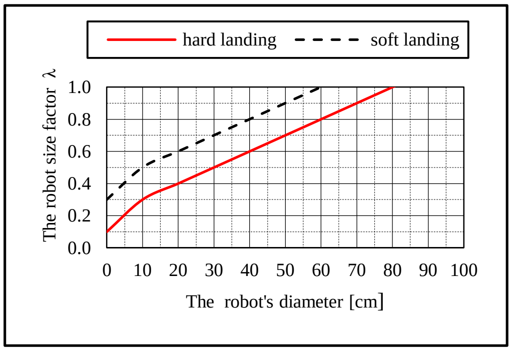

Figure 1: The calibration curve

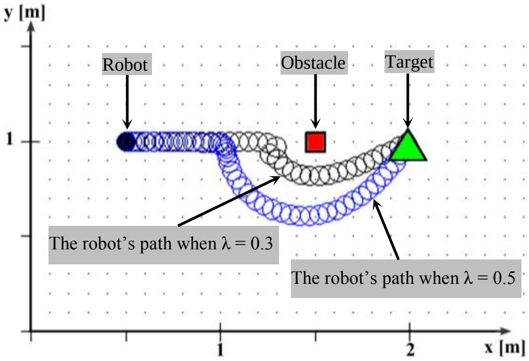

Figure 2: The influence of the robot size factor

Concerning the choice of the attractive force, $F_{att}(q_r)$, the principle is that the robot should rush to the target as quickly as possible. Typically, $F_{att}(q_r)$ is defined as a function of the relative distance between the robot and the target only where the target is stationary. However, when the target moves, it is valuable to incorporate the relative velocity of target in the construction of the attractive force. Conversely, this solution produces additional degree of freedom, which increases the computational complexity. As stated before, the relative placement scheme of the dynamic virtual obstacle is a function of the nature and status of the target in the workspace. So, DVOC can be used to adjust the robot's velocity with respect to the moving obstacles as well as to regulate the robot's velocity with respect to the moving target. In this paper, the attractive force, $F_{att}(q_r)$, is presented as follows:

$$F_{att}(q_r) = \zeta(q_t - q_r) + f_{vo}(q_r)_{mod} \tag{4}$$

where ζ is a positive scaling factor.

The summation of $F_{att}(q_r)$ and all the repulsive forces yields a total force, $F_{total}(q_r)$, which keeps a similar moving trend with the target and a contrary trend with the obstacles. The direction of the total force vector will steer the robot during its journey to the target. The total force $F_{total}(q_r)$ and the steering angle α are given in (5) and (6), respectively.

$$F_{total}(q_r) = F_{att}(q_r) + \sum_{i=1}^{n} f_{rep_i}(q_r) \tag{5}$$

$$\alpha = \tan^{-1} \frac{F_{total,y}}{F_{total,x}} \tag{6}$$

where $F_{total,x}$ and $F_{total,y}$ are the total force in *x-axis* and *y-axis*, respectively.

From (5) and (6), it is apparent that the total force and the steering angle are varying at each instant. This means; the relative distances are interpreted as forces. As a result, it is a reasonable idea to utilize these forces to generate a direct input signal to the controller of the robot in order to enable the robot to plan and control its motion. Consequently, the speed of the robot is proportional to the magnitude of $F_{total}(q_r)$. So, the total force can be interpreted as a velocity or a driving torque. This means; the conjunction of $F_{att}(q_r)$ and $\sum_{i=1}^{n} f_{rep_i}(q_r)$ provides automatically a controller enables the robot to reach its target without any possible collision with the obstacles. When the robot encounters an obstacle in its way, the repulsive force reduces the magnitude of $F_{total}(q_r)$, thereby reduces the robot's velocity, i.e., damping effect. When the obstacle is behind the robot, the robot's velocity will be increased. While the modified virtual force always increases the magnitude of $F_{total}(q_r)$, therefore, increases the robot's velocity. As a result, the conjunction between the modified virtual force and the repulsive forces represents a damping effect, which in turn depends on the robot size factor. It is clear from (1) and (3) that $\rho(q_r, q_{o_i})$ and $\rho(q_r, q_{vo})$ influence the magnitude of $F_{total}(q_r)$, hence they influence the robot's velocity. The speed mechanism should be implemented as follows:

If $\rho(q_r, q_{vo}) < \rho_o$ and $\rho(q_r, q_{o_i}) \geq \rho_o$, the robot will rush with high speed toward the target directly.
If $\rho(q_r, q_{vo}) \leq \rho(q_r, q_{o_i}) < \rho_o$, the robot will move with medium speed to a direction near to the target.
If $\rho(q_r, q_{vo}) > \rho(q_r, q_{o_i})$ and $\rho(q_r, q_{o_i}) > 2R_r$, the robot will deviate with medium speed away from the obstacle.
If $\rho(q_r, q_{vo}) > \rho(q_r, q_{o_i})$ and $\rho(q_r, q_{o_i}) \leq 2R_r$, the robot will move with high speed to the opposite direction of the obstacle, where R_r is the robot's radius.
If $\rho(q_r, q_{vo})$ is not defined, the robot should move with its normal speed.

Therefore, the enhancement of the movement toward the target is coupled with the position of the virtual obstacle as shown in Figure 3, i.e., as $\rho(q_r, q_{vo})$ decreases, as $F_{att}(q_r)$ increases, which in turn increases the robot's velocity. The thick dashed line in Figure 3 represents the direction of movement when the DVOC is not applied.

Generally, the maximum velocity of the robot is subjected to physical constraints, and its magnitude is upper bounded. However, it is important to restrict the maximum travel per each instant no more than the robot's radius in order to acquire a smooth path. To guarantee this, the following condition must be satisfied:

$$T\, \mathbb{V}_{max} \leq R_r \tag{7}$$

where T and $\mathbb{V}_{max}$ denote the sampling period, and the maximum speed of the robot, respectively. $\mathbb{V}_{max}$ should be defined at the beginning of the run.

At each sampling instant, the velocity, v_{robot}, and the angular velocity, ω_{robot}, of the robot can be calculated as follows:

$$v_{robot} = \begin{cases} \mathcal{K}_v F_{total}(q_r), & \text{if } |\mathcal{K}_v F_{total}(q_r)| \leq \mathbb{V}_{max} \\ \mathbb{V}_{max} \dfrac{F_{total}(q_r)}{|F_{total}(q_r)|}, & \text{otherwise} \end{cases} \tag{8}$$

$$\omega_{robot} = \begin{cases} \mathcal{K}_\omega (\alpha - \theta), & \text{if } \omega \leq \omega_{max} \\ \omega_{max}, & \text{otherwise} \end{cases} \tag{9}$$

where $\mathcal{K}_v$ and $\mathcal{K}_\omega$ are the velocity and angular velocity gain coefficient, respectively. θ denotes the direction angle of the current movement, while α is the steering angle. ω_{max} is the maximum allowable steering rate.

Assuming there is no slipping, the time-domain kinematic model of a differential drive mobile robot is governed by the following nonlinear state equation [20]:

$$\begin{bmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \frac{R_w}{4G_r}\big(\omega_r(t)+\omega_l(t)\big)\cos\theta(t) \\ \frac{R_w}{4G_r}\big(\omega_r(t)+\omega_l(t)\big)\sin\theta(t) \\ \frac{R_w}{2LG_r}\big(\omega_r(t)-\omega_l(t)\big) \end{bmatrix} \tag{10}$$

where $\omega_r(t)$ and $\omega_l(t)$ are the angular velocity outputs of the right and left wheel, respectively. R_w, G_r, and L are the diameter of the drive wheels, the gear ratio, and the distance between the driven wheels, respectively. So, the linear and angular velocity at the center of the robot can be rewritten as follows:

$$v_{robot} = \frac{R_w}{4G_r}\big(\omega_r(t)+\omega_l(t)\big) \tag{11}$$

$$\dot{\theta}(t) = \frac{2}{L}v_{robot} - \frac{R_w}{LG_r}\omega_l(t) \tag{12}$$

As shown in (10) and (11), there are two parameters to be controlled, which are $\omega_r(t)$ and $\omega_l(t)$. Therefore, we need to determine only one parameter, $\omega_l(t)$, in the navigation process, while the other parameter, $\omega_r(t)$, can be determined from (11). The schematic drawing of the differential drive mobile robot is shown in Figure 4.

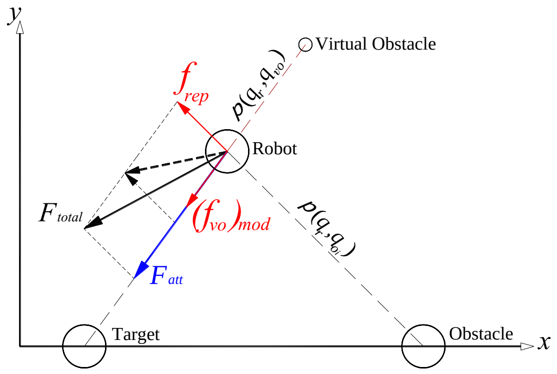

Figure 3: The layout of the dynamic virtual obstacle

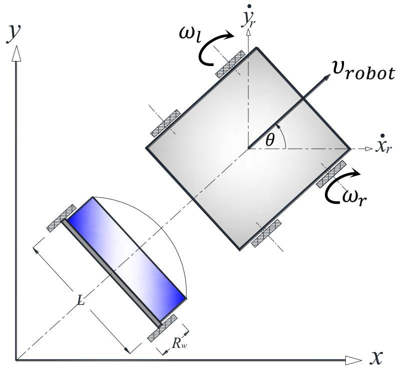

Figure 4: Schematic drawing of a differential drive mobile robot

3.4. *The Configuration of The Virtual Obstacle*

Regarding the dynamic environment, the most common methods ignore the trend of moving target and obstacles. This treatment leads to an inefficient robot path. When the obstacle is moving away from the robot, the robot's path should not be affected by the obstacle as shown by the solid line in Figure 5. When the trend of moving obstacles is not taken into account, the resulting robot's path is not optimal as represented by the dashed line. Frequently, most techniques concerning with obstacle avoidance may result in non-optimal paths, since no prior knowledge about the environment is used. However, with the aid of the DVOC, the optimal *on-line* path can be generated.

The key point of the DVOC is to virtualize the form of the obstacle, and how to set it up in order to obtain an appropriate entire potential field. The dynamic virtual obstacle should be configured as follows: a) it is a dummy obstacle of a point size, b) it should provide an appropriate attractive potential toward the target, c) it should afford a reasonable repulsive potential to keep a safety margin around the robot, d) its position should enhance a fast maneuvering, and e) its position and velocity are varying according to the position and velocity of the robot. The relative position of the virtual obstacle along with the proper choice of the robot size factor is considered a controlling mechanism, which has the role of adjusting the robot's velocity. By using this algorithm, there is no need to additional degree of freedom, since the position of the virtual obstacle gives a warning of any threat, and the robot will adjust its velocity automatically. The relative distance between the robot and the nearest obstacle, and the relative distance to the target, in each step, efficiently determine whether the DVOC should be implemented or not. Therefore, after comparing several placement schemes, the placement of the virtual obstacle in the workspace is chosen as follows:

$$\rho(q_r, q_{vo}) = \frac{\rho(q_r,q_t)+\min\left(\rho\left(q_r,q_{o_i}\right)\right)}{2} \tag{13}$$

where $\rho(q_r, q_t)$, and min $\left(\rho\left(q_r, q_{o_i}\right)\right)$ are the minimum distances from the robot to the target, and to the nearest obstacle, respectively.

This algorithm affords some remarkable merits such as: a) it is suitable for the complex environment, b) it can be used as a protective algorithm that ensures the robot does not become trapped in local minima or GNRON, c) it can eliminate the oscillations, d) less execution time, e) it provide a fast moving maneuvering in unknown environments.

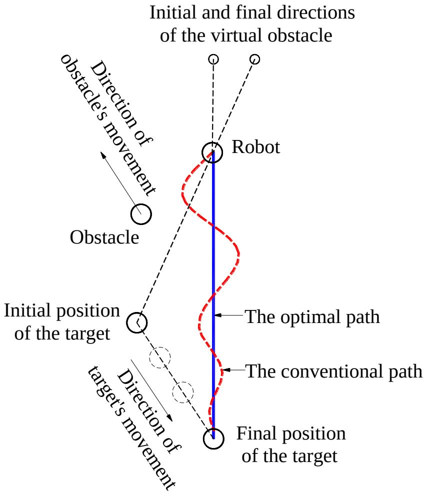
Initial and final directions of the virtual obstacle
Direction of obstacle's movement
Robot
Obstacle
Initial position of the target
The optimal path
The conventional path
Direction of target's movement
Final position of the target

Figure 5: The oscillations of a mobile robot in a dynamic environment

4. Remarks on Practical Implementations

In order to adapt the robot in a dynamic environment without auxiliary human interference, we should provide it with the ability to extract information from the environment. So, a laser range finder, SICK LMS100, is suggested. LMS100 scans the workspace with the resolution of 0.5°, field of view of 270°, and operating range of 20 meters. Hence, 541 readings are observed and denoted by (r_j, γ_j), where $j = 0, 0.5, \ldots, 270$, r_j is the distance to the obstacle and γ_j is the measured angle of the obstacle. Throughout each scan, the environment model is updated continuously, and the desired velocity can be computed directly. In practice, the laser scanner will detect many distances to one obstacle, so our calculations are based on the smallest r_j for each obstacle. The procedure of our algorithm is summarized in Figure 6. If the LMS 100 is mounted at the middle of the robot, then the following algorithm should be applied.

```
For $j$ = 0, 0.5, 1,…, 270
if $|r_j \sin(\alpha - \gamma_j)| > R_r$
then move
if $|r_j \sin(\alpha - \gamma_j)| \leq R_r$
    if $|r_j \cos(\alpha - \gamma_j)| > 2R_r$
    then move
else stop
```

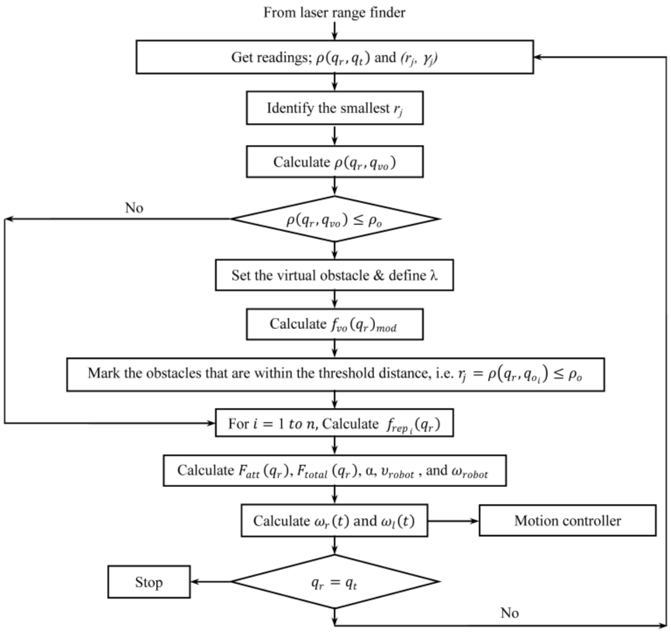
From laser range finder
Get readings; $\rho(q_r, q_t)$ and (r_j, γ_j)
Identify the smallest r_j
Calculate $\rho(q_r, q_{vo})$
$\rho(q_r, q_{vo}) \leq \rho_o$
No
Set the virtual obstacle & define λ
Calculate $f_{vo}(q_r)_{mod}$
Mark the obstacles that are within the threshold distance, i.e. $r_j = \rho(q_r, q_{o_i}) \leq \rho_o$
For $i = 1\ to\ n$, Calculate $f_{rep_i}(q_r)$
Calculate $F_{att}(q_r)$, $F_{total}(q_r)$, α, υ_{robot}, and ω_{robot}
Calculate $\omega_r(t)$ and $\omega_l(t)$
Motion controller
Stop
$q_r = q_t$
No

Figure 6: The flowchart of the proposed DVOC

5. SIMULATION RESULTS

To show the effectiveness of the DVOC, extensive simulation studies are carried out under realistic assumptions. In the operational space, the initial positions of the robot, the target, and the obstacle are represented by a black circle, a green triangle, and a red square, respectively. In order to evaluate our proposal, three categories of the workspace are presented; firstly, the motion behavior in a dynamic environment where no obstacle exists, secondly the motion behavior in a dynamic environment where the obstacles and target are moving, and thirdly, the motion behavior in a partially dynamic environment. Eventually, we shall investigate our proposal in a dynamic environment, which encounters LMP. Simulations were conducted using our own software (ROBINAV-V2) developed by the authors. The values of all parameters are set as $\zeta = \eta = 1, \rho_o = 2,$ and $\mathcal{K}_\upsilon = \mathcal{K}_\omega = 1$. The simulations are based on a sampling period $T = 0.1$ second.

5.1 CASE STUDY 1: DYNAMIC ENVIRONMENT WITHOUT OBSTACLES

We shall evaluate the performance of DVOC in an environment where no obstacle exists. In this case, the motion behavior is influenced by the virtual obstacle, which in turn is influenced by the target's position only. The target is moving linearly from point $[1\ 2.75]^T$ at a constant velocity $[0.028\ -0.028]^T$. When the target reaches to the maximum size of the environment, it changes its movement to the opposite direction. The initial position of the robot is $[0.25\ 2.5]^T$, and its initial and maximum velocities are $[0\ 0]^T$ and $\mathbb{V}_{max} = 0.5$ m/s, respectively. Figure 7 and Figure 8 show the simulation result and the velocity diagram for 10 cm robot's diameter, respectively. When the DVOC is not implemented or the robot size factor is too high, the trajectory suffers from perturbations and unstable oscillations as shown by the thin black path in Figure 7 and the dashed line in Figure 8, as well as the robot cannot reach to the target (we terminated the simulation at $t = 42$ s). The reason for these oscillations is that the robot's velocity is varying according to the relative distance from the robot to the target only. Besides, the target's velocity is higher than the robot's velocity. If λ is too high, the influence degree of the virtual obstacle will be faded. In order to compensate the velocities' difference; we must keep a short distance between the robot and the target and this can be accomplished by adjusting λ to a small value. Conversely, these oscillations are eliminated when our algorithm is implemented and λ is adjusted to 0.3 as shown by the thick blue path in Figure 7. At $t = 7.34$ s, the robot caught the target successfully as shown by the solid line in Figure 8. Since the target's velocity is relatively high, the robot accelerates with its maximum velocity in order to catch the target. Clearly, when the robot size factor is well defined, our algorithm is faster and smoother. Obviously, the simulation result validates the analysis in section 3.3 and verifies that as λ decreases, as oscillations decrease.

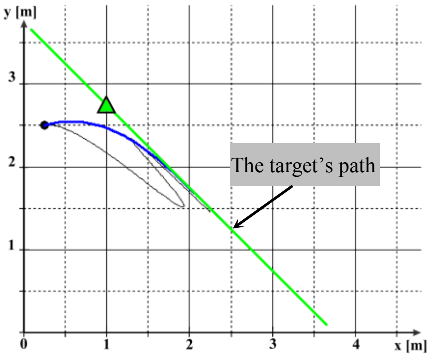

Figure 7: The simulation results of case study 1

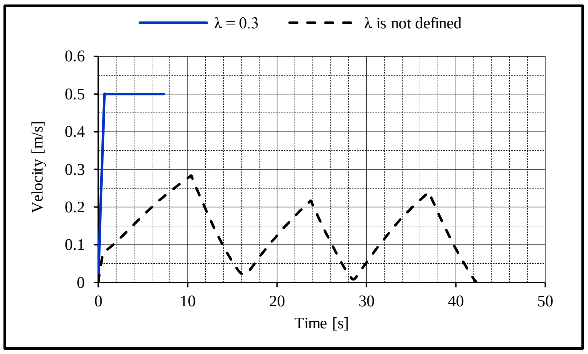

Figure 8: The velocity diagram of case study 1

5.2 Case Study 2: Dynamic Environment with Moving Obstacles

Now, we investigate the validity of DVOC in a dynamic environment where both the target and the obstacles are moving. In this case, the positions of the target and the obstacles concurrently influence the motion behavior. The target is moving linearly at a constant velocity $[0.013\ -0.015]^T$starting from point $[3\ 7]^T$. There are two moving obstacles whose initial positions and velocities are $q_{o_1} = [2.5\ 1.6]^T$, $q_{o_2} = [7\ 7]^T$, $\upsilon_{o_1} = [0\ 0.03]^T$, and $\upsilon_{o_1} = [-0.008\ -0.03]^T$, respectively. The robot starts from its initial position $[0.7\ 0.55]^T$ with zero initial velocity. Figure 9 shows the motion behaviors when λ is adjusted to 0.4 and 0.6, while Figure 10 represents the velocity diagrams for the same situations. This simulation is based on 20 cm robot's diameter and $\mathbb{V}_{max} = 1$ m/s. Consider $\lambda = 0.4$. At first, the robot is far from the obstacles and is influenced only by the target. As a result, the robot accelerates toward the target. At $t = 0.71$ s, the robot enters the influence range of the first obstacle. As the robot moves on, it approaches the first obstacle. Hence, the repulsive force increases gradually and the robot slows down its velocity. At $t = 10.88$ s, the safety margin reaches to its minimum value, which is equal to the robot's diameter. Therefore, the robot decelerates its velocity very quickly and keeps its direction nearly parallel to the obstacle's movement in order to allow the obstacle passes first. At $t = 15.56$ s, the robot succeeded to detour behind the obstacle and avoids collision, however the robot still inside the influence range of the obstacle. Hence, the robot starts to accelerate again. At $t = 19.78$ s, the obstacle moves away from the robot, and its threat decreases, therefore the robot decelerates its velocity to allow soft landing on the target. At $t = 24.17$ s, the obstacle is no longer a threat but the relative distance between the robot and the target increases; therefore, the robot starts to accelerate again. At $t = 26.75$ s, the robot successfully arrived to the target. It is obviously clear that the robot is always outside the influence range of the second obstacle.

Consider $\lambda = 0.6$. In comparison to the previous situation, the two trajectories are identical in the steering angle until $t = 8.66$ s. At $t = 0.36$ s, the robot enters the influence range of the first obstacle. As the robot moves, as it approaches the first obstacle. Hence, the repulsive force increases gradually and the robot slows down its velocity. At $t = 5.47$ s, the robot gets closer to the first obstacle than the virtual obstacle. That means; the first obstacle exerts a repulsive force higher than the modified virtual force coming from the virtual obstacle. Therefore, the robot maintains its deceleration and changes its direction slightly upper left to allow the obstacle passes first. At $t = 12.82$ s, the robot succeeded to detour behind the obstacle and avoids collision, however the robot still inside the influence range of the first obstacle. Hence, the robot starts to accelerate. As the first obstacle moves away, as its threat decreases and the robot starts to decelerate gradually at $t = 17.64$ s. At $t = 22.72$ s, the robot is again outside the influence range of both obstacles, and the target is in front without any threat. However, the robot cannot accelerate toward the dynamic target since the arrangement of the virtual obstacle at this instant makes the robot's velocity less than the target's velocity. As a result, the relative distance between the robot and the target increases, thereby the attractive force increases. Consequently, the robot speeds up again at $t = 30.54$ s. At $t = 35$ s, the second obstacle changed its direction and moves again toward the target; however its effect is still zero. The robot continues to follow the dynamic target but still with low speed. At $t = 40.6$ s, the robot enters into the influence range of the second obstacle, and the repulsive force is generated. At $t = 42.1$ s, the robot detects that the obstacle is nearby and may obstruct its way; in addition, the position of the virtual obstacle at this instant enables the robot to accelerate very quickly and to change its direction slightly left to keep itself away from the obstacle. At $t = 47.22$ s, the robot succeeds to avoid the second obstacle and starts to slow down its velocity. Finally, the robot caught the target successfully at $t = 58.1$ s.

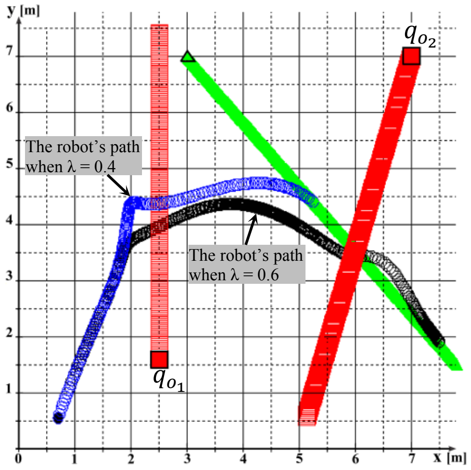

Figure 9: The simulation results of case study 2

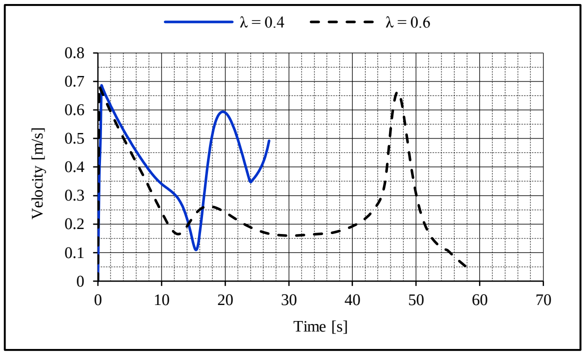

Figure 10: The velocity diagram of case study 2

5.3 *Case Study 3: Partially Dynamic Environment*

This case represents a partially dynamic environment. There are a stationary target located at $[4.5\ 1.8]^T$ and three obstacles located at $[2\ 2.9]^T$, $[3.5\ 1.8]^T$, and $[0.6\ 0.6]^T$, respectively. The first two obstacles are stationary, while the third obstacle is moving at a constant velocity $[0.025\ 0.017]^T$. All obstacles are 30 cm × 30 cm. The robot of 30 cm diameter starts movement from point $[0.6\ 2.9]^T$ with zero initial velocity.

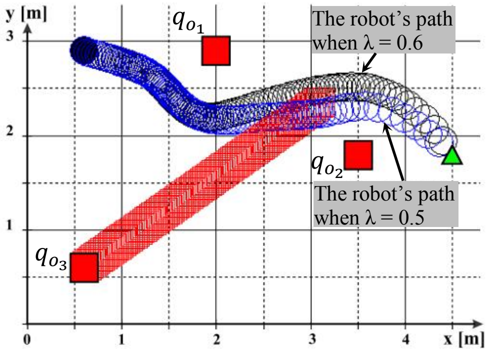

Figure 11: The simulation results of case study 3

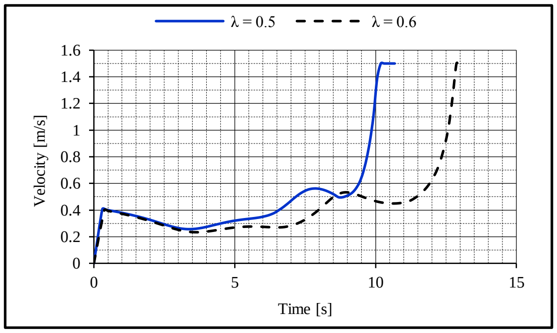

Figure 12: The velocity diagram of case study 3

5.4 *Case Study 4: Dynamic Environment with LMP*

The likelihood of the local minima does exist in the dynamic environment. Regarding the classical LMP, we have solved this phenomenon in our previous research [2]. However, there is a special kind of LMP associated with the dynamic environment. For simplicity, let's consider the case when the robot, the obstacle, and the target are collinear and the obstacle is in between. They all move in the same direction along the same line. Assuming that; the target's velocity is greater than or equals the obstacle's velocity. This assumption ensures that the obstacle is between the robot and the target all the time, and the robot is obstructed by the obstacle. The simplest way to solve this problem is to keep the robot to move according to the total force as usual and wait for the obstacles or the target to change their motion. However, if there is no change after a certain period, the robot will be trapped. Here, we have another solution. Due to the obstacle is not a point size, and rather than the calculations are based on the smallest distance to the obstacle, r_j, we will make slight deviations by considering the distance to one edge of the obstacle. Accordingly, the direction of the total force will slightly deviate upward or downward and the scenario will be changed. Then the position of the virtual obstacle will make the robot to accelerate and detour around the obstacle. This solution is quite realistic since it is consistent with the white noise from the environment. Figure 13 demonstrates that the proposed new methodology is capable of tackle the problem. The target is moving linearly at a constant velocity $[0.03\ 0]^T$ starting from point $[1.8\ 1]^T$, while the obstacle is moving at a constant velocity $[0.02\ 0]^T$

starting from $[1\ 1]^T$. The initial position of the robot is $[0.25\ 1]^T$ and its initial velocity is zero. Figure 13 shows the simulation results for 10 cm robot's diameter and $\mathbb{V}_{max} = 0.5$ m/s. In order to show the influence of the robot size factor, we constructed two trajectories for different λ.

When $\lambda = 0.3$, the virtual obstacle is very close to the robot and the generated virtual force is larger than the repulsive force. As a result, the robot collides with the obstacle. When $\lambda = 0.4$, the virtual obstacle generates a reasonable virtual force, which keeps a safety margin between the robot and the obstacle. The robot recognizes that the obstacle is directly in front, therefore it adjusts its speed and direction to avoid the obstacle. Finally, the robot succeeded to detour over the obstacle and finally it accelerates with its maximum speed toward the target.

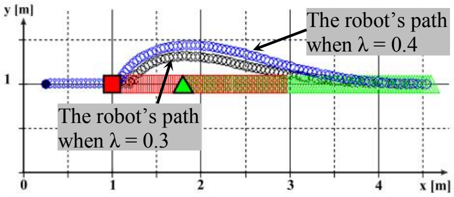

Figure 13: The simulation results of case study 4

6. Conclusions

This paper illustrates an effective algorithm to solve the problem of path planning in a dynamic environment. This algorithm relies on the integration between the dynamic virtual obstacle concept and the robot size factor, which represents a threat factor to any possible collisions. This integration has the role of adjusting not only the speed of the robot but also the steering angle. The conjunction of the new definition of the attractive force and the repulsive force gives the robot the ability to plan and control its motion to follow a moving target in a desired behavior while avoiding moving obstacles. When the robot size factor is well defined, there is always a safety margin between the robot and the obstacle, as well as the robot's path is free from oscillations. The relative position of the virtual obstacle along with the proper choice of the robot size factor has a great influence on the speed mechanism (soft landing and/or hard landing). Furthermore, this algorithm affords high capabilities to deal with local minima and provides a fast moving maneuvering in unknown environment, as well. The convergence of this proposal is discussed to guarantee its reliability and validity in partially dynamic as well as dynamic environments.

References

[1] S. S. Ge and Y. J. Cui: "Dynamic Motion Planning for Mobile Robots using Potential Field Method", International Journal of Autonomous Robots, Vol.13, No.3, pp.207-222, 2002.

[2] H. H. Shehata and J. Schlattmann: "Mobile Robot Path Planning and Obstacle Avoidance Based on a Virtual Obstacle Concept", Proceedings of the 21st International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), Vol. 2, No. 1, pp. 905-914, Taichung, Taiwan, 2011.

[3] S. X. Yang and M. Q. -H. Meng: "Real-Time Collision-Free Motion Planning of a Mobile Robot using a Neural Dynamics-Based Approach", IEEE Transactions on Neural Networks, Vol.14, No.6, pp.1541-1552, 2003.

[4] O. Khatib: "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", International Journal of Robotics Research, Vol.5, No.1, pp.90-98, 1986.

[5] J. Borenstein and Y. Koren: "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", IEEE Transactions on Robotics and Automation", Vol.7, No.3, pp.278-288, 1991.

[6] Y. Koren and J. Borenstein: "Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation", Proceedings of the IEEE International Conference on Robotics and Automation, Vol.2, pp.1398-1404, California, 1991.

[7] L. Huang: "A Potential Field Approach for Controlling a Mobile Robot to Track a Moving Target", Proceedings of the IEEE International Symposium on Intelligent Control, pp.65-70, Singapore, 2007.

[8] A. Poty, P. Melchior, and A. Oustaloup: "Dynamic Path Planning for Mobile Robots using Fractional Potential Field", Proceedings of the 1st International Symposium on Control, Communications and Signal Processing, pp. 557-561, 2004.

[9] S. R. Munasinghe, C. Oh, J. -J. Lee, and O. Khatib: "Obstacle Avoidance using Velocity Dipole Field Method", Proceedings of the International Conference on Control, Automation, and Systems, pp. 1657-1661, Korea, 2005.

[10] E. Prassler, J. Scholz, and P. Fiorini: "A Robotic Wheelchair for Crowded Public Environments", IEEE Robotics and

Automation Magazine, Vol. 8, No. 1, pp. 38-45, 2001.

[11] *S. Sugiyama, J. Yamada, and T. Yoshikawa: "Path Planning of a Mobile Robot for Avoiding Moving Obstacles with Improved Velocity Control by using the Hydrodynamic Potential", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1421-1426, Taipei, Taiwan, 2010.*

[12] *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.*

[13] *L. Yin and Y. Yin: "An Improved Potential Field Method for Mobile Robot Path Planning in Dynamic Environments", Proceedings of the 7th World Congress on Intelligent Control and Automation, pp. 4847-4852, Chongqing, China, 2008.*

[14] *C. Qixin, H. Yanwen, and Z. Jingliang: "An Evolutionary Artificial Potential Field Algorithm for Dynamic Path Planning of Mobile Robot", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3331-3336, Beijing, China, 2006.*

[15] *G. -C. Luh and W. -W. Liu: "Dynamic Mobile Robot Navigation using Potential Field Based Immune Network", Journal of Systemics, Cybernetics and Informatics, Vol.5, No.2, pp.43-50, 2006.*

[16] *D. Schulz, W. Burgard, D. Fox, and A. Cremers: "Tracking Multiple Moving Targets with a Mobile Robot using Particle Filters and Statistical Data Association", Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2, pp. 1665-1670, Seoul, Korea, 2001.*

[17] *D. A. Grundel: "Searching for a Moving Target: Optimal Path Planning", Proceedings of the IEEE International Conference on Networking, Sensing and Control, Vol. 2, pp. 867-872, Arizona, USA, 2005.*

[18] *N. Y. Ko and B. H. Lee: "Avoidability Measure in Moving Obstacle Avoidance Problem and its Use for Robot Motion Planning", Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Vol. 3, pp. 1296-1303, Osaka , Japan, 1996.*

[19] *B. Hussien: "Robot Path Planning and Obstacle Avoidance by Means of Potential Function Method", Ph. D. Dissertation, University of Missouri, Columbia, 1989.*

[20] *C. -C., Tsai: "A Localization System of a Mobile Robot by Fusing Dead-Reckoning and Ultrasonic Measurements", IEEE Transactions on Instrumentation and Measurement, Vol. 47, No. 5, pp. 1399-1404, 1998.*