

Mobile Robot Path Planning with Randomly Moving Obstacles and Goal

Vanitha Aenugu Peng-Yung Woo

Department of Electrical Engineering

Northern Illinois University

Dekalb, IL 60115, USA

Abstract- This article presents the dynamic path planning for a mobile robot to track a randomly moving goal with avoidance of multiple randomly moving obstacles. The main feature of the developed scheme is its capability of dealing with the situation that the paths of both the goal and the obstacles are unknown *a priori* to the mobile robot. A new mathematical approach that is based on the concepts of 3-D geometry is proposed to generate the path of the mobile robot. The mobile robot decides its path in real time to avoid the randomly moving obstacles and to track the randomly moving goal. The developed scheme results in faster decision-making for successful goal tracking. 3-D simulations using MATLAB validate the developed scheme.

Index Terms- mobile robot, path planning, randomly moving obstacles, obstacle avoidance, randomly moving goal

I. INTRODUCTION

Mobile robots are automated machines that are capable of movements in a given environment [1]. Usually, these autonomous mobile robots are used in higher, deeper, riskier environments where humans cannot imagine treading [2]. In order to achieve tasks, they are intelligent in deciding their own actions. Especially, in an environment with obstacles, it is necessary to plan a collision-free path for the mobile robot to reduce cost, energy, time, and distance [1].

Mobile robots were first installed in factories around 1968 and were mainly automated guided vehicles (AGVs) and vehicle transporting tools following a predefined trajectory. However, the study in this area deals now with autonomous indoor and outdoor navigation. Autonomous mobile robots execute the task in four stages: perception of the environment, self-localization, motion planning, and motion generation [2].

In structured environments, the perception process allows maps or models of the world to be generated that are used for mobile robot localization and motion planning. In unstructured environments, however, the mobile robot has to learn how to navigate. The localization process allows a mobile robot to know where it is at any moment relative to its environment. For this purpose, sensors are used that will take measurements of the mobile robot's current state and its environment [2]. In motion planning, the path to be taken is decided and in motion generation, the planned motion is implemented and monitored [3].

A great number of different techniques have been and are being developed in order to carry out efficient mobile robot obstacle avoidance, while the obstacle avoidance

algorithm determines a suitable direction of motion based on recent sensor data. Obstacle avoidance is performed locally in order to ensure that real-time constraints are satisfied [4]. A fast update rate of the obstacle avoidance is required to allow the mobile robot to move safely at high speeds.

The potential field method is commonly used for autonomous mobile robot navigations because of its simple and elegant mathematical analysis. The basic concept of the potential field method is to fill the mobile robot's workspace with an artificial potential field where the mobile robot is pulled to its goal and is pushed back from the obstacles [5]. However, substantial shortcomings that are inherent to these principles do exist and are presented in [6].

Borenstein and Koren [7] developed the Virtual Force Field approach to detect unknown obstacles and simultaneously to steer the mobile robot towards the goal. This method combines the potential field method with a certainty grid. One problem inherent to this method is the possibility of the mobile robot getting trapped. Borenstein and Koren [8] developed the Vector Field Histogram (VFH) approach, a real-time obstacle avoidance method, for fast-running vehicles, which has a significant improvement over the Virtual Force Field approach. Brooks [9] achieved collision-free path planning by good representation of free space. 3-D collision-free path planning was achieved by Herman [10]. All these methods were developed by assuming that both the obstacles and the goal are stationary. In many real-life implementations, however, everything is dynamic. Not only the obstacles, but also the goal is moving.

Fujimura and Samet [11] presented a method for path planning among moving obstacles, where time is included as one of the dimensions of the model world. Thus, the moving obstacles are regarded as stationary in the extended world. The major problem in this approach is that the information of the obstacle is provided *a priori* to the mobile robot. Chakravarthy and Ghose [12] presented a collision-cone approach for collision detection and navigation based on kinematic equations. The collision conditions are established based on the range and line-of-sight rates. Ge and Cui [13] presented a dynamic potential field method to avoid randomly moving obstacles and to reach a randomly moving goal. However, there are some problems, for example, local minima and free-path local minima. Berg and Overmars [14] extended a classical probabilistic road map method, which was originally designed for static environments, to the dynamic environment. This method has limited

performance when obstacles are allowed to move in the environment.

Recently, Seder and Petrovic [15] presented a dynamic window method. This approach works well in static environments and do not perform well in dynamic environments due to the large difference in the collision detection and the necessity to control the speed of the mobile robot for the two different environments, dynamic or static. Masehian and Katebi [16] presented a new sensor-based online method for path planning of mobile robots with a moving goal among stationary and moving obstacles. Belkhouche [17] introduced the concept of the virtual plane for path planning in dynamic environments. The virtual plane is an invertible transformation equivalent to the workspace, which is constructed by using a local observer.

In this study, dynamic path planning for a mobile robot that can track and reach a randomly moving goal is generated, while multiple randomly moving obstacles of various shapes and sizes are avoided. Here, "random" means that the trajectories of the moving obstacles and goal are not known *a priori*. The MATLAB 3-D simulations mimic closely the real-life environments.

In Section 2, advantages of the proposed method are presented compared to the dynamic potential field method. The concepts of the new mathematical approach are discussed and the related equations are derived. Special cases are also discussed in this section. In Section 3, simulation results for the multiple randomly moving obstacles and the randomly moving goal as well as the tracking path of the mobile robot are presented. Section 4 gives the conclusion and future work.

II. DEVELOPMENT OF THE SCHEME

The main object of the developed scheme is to propose a new method based on 3-D geometry for a mobile robot to track and reach a randomly moving goal in a 3-D environment with avoidance of multiple randomly moving obstacles, while no knowledge of the movements of the goal as well as the obstacles is known to the mobile robot *a priori*.

A. Superior to the dynamic potential field method

Ge and Cui [13] proposed a new potential field method for path planning of a mobile robot in a dynamic environment where both the obstacle and the goal are moving. According to this method, the attractive potential is a function of the relative position and velocity of the goal with respect to the mobile robot, and the repulsive potential is defined as the relative position and velocity of the mobile robot with respect to the obstacles. Thus, the virtual force is defined as the negative gradient of the potential in terms of the position and velocity rather than the position only. They discussed two problems when this method is applied to dynamic environments.

(1) Local minimum

Consider the case that the mobile robot, the obstacle and the goal are moving in the same direction along a line and the obstacle is in between the mobile robot and the

goal as shown in Figure 2.1. According to the dynamic potential field method, the mobile robot is always repelled from the obstacle and cannot reach the goal. This is called the local minimum problem. To solve this problem, Ge and Cui [13] suggested that the mobile robot needs to wait until either the goal or the obstacle changes its position. After a certain time of waiting, if the mobile robot is still in the same situation, the conventional local minimum recovery approaches such as the wall-following method proposed by Yun and Tan [18] are applied. In their method, the mobile robot follows the current obstacle contour, which is the boundary of the obstacle that pushes the mobile robot away. Once the mobile robot is recovered from the trapping situation, it will follow the original method to reach the goal.

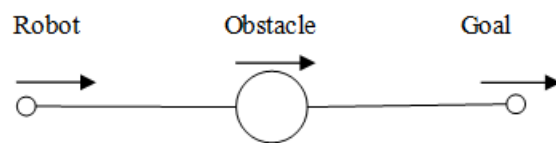


Figure 2.1 Local minimum

(2) Free-path local minimum

Another kind of local minimum problem is generated when the goal is within the influence range of the obstacle. Consider the case that both the mobile robot and the goal are near to the obstacle, with the goal in between the mobile robot and the obstacle, moving along a line as shown in Figure 2.2. In this case, when the mobile robot tries to reach the goal, it is repelled from the obstacle at the same time. The mobile robot can reach the goal only if the velocity of the mobile robot is large enough. This is called the free-path local minimum problem. This problem is solved by setting the repulsive force to zero, i.e., only the attractive force is taken into account [13].

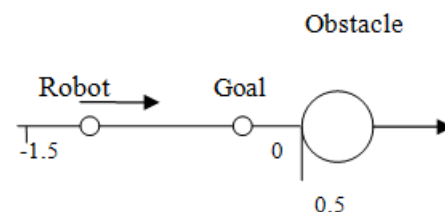


Figure 2.2 Free-path local minimum

The developed scheme eliminates the problems of the dynamic potential field method.

(1) The local minimum problem as well as the free-path local minimum problem is avoided by moving the mobile robot through the via-point to reach the goal, even though all the aforementioned three bodies are moving along a line with the same direction.

(2) The collision with any obstacle is detected long before the collision could take place. Therefore, the chance for any collision is well prevented.

(3) The safety zone of the obstacle, which is to be defined in Subsection B, can be varied depending upon requirements.

B. Collision detection

The convention used in this study relates to analytic geometry and 3-D vectors.

(1) Mobile robot

The mobile robot is defined as a point object $R(R_x, R_y, R_z)$ in the 3-D space. It is represented as a solid dot in MATLAB plots.

(2) Obstacle

Obstacles are unpredictable objects that the mobile robot may encounter during the execution of the task. In general, obstacles can be of any shape and size with a volume characteristic. Each obstacle's position is represented by $C(C_x, C_y, C_z)$. Here, the obstacles are in random movements. The mobile robot has to be alert at all times to avoid randomly moving obstacles and, at the same time, to move towards a randomly moving goal.

(3) Safety zone

The safety zone of the obstacle is the region around the obstacle that the mobile robot must avoid. It is a sphere with radius r around the obstacle as shown in Figure 2.3. The radius r is chosen in accordance with the size of the obstacle. If the obstacle is of an irregular shape, radius r is one half of the longest side of the obstacle. The sphere drawn by using this longest side should "wrap" the entire obstacle.

(4) Goal

The goal is defined as a point object $G(G_x, G_y, G_z)$ in the 3-D space. It is represented as a square in MATLAB plots. The goal moves randomly.

In general, the obstacles are detected by sensors, which are assumed being mounted on the mobile robot. When the mobile robot detects the presence of any obstacle with the help of the sensors, it has to decide whether the obstacle is in its way. The steps below are taken to detect any possible collision with an obstacle.

The equation of a line passing through two points, the mobile robot's current position $R(R_x, R_y, R_z)$ and the goal's current position $G(G_x, G_y, G_z)$, is

$$\frac{x - R_x}{G_x - R_x} = \frac{y - R_y}{G_y - R_y} = \frac{z - R_z}{G_z - R_z} = \lambda \quad (2.1)$$

where λ is a constant.

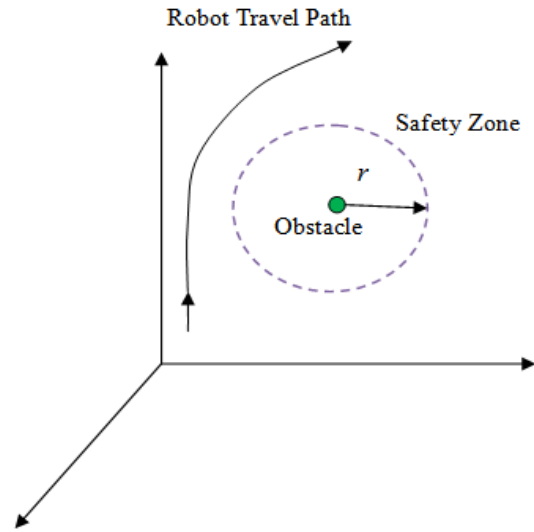


Figure 2.3 Safety zone of the obstacle

The equation of a sphere that has the center $C(C_x, C_y, C_z)$ and radius r representing the safety zone of the obstacle is

$$(x - C_x)^2 + (y - C_y)^2 + (z - C_z)^2 = r^2 \quad (2.2)$$

Combining (2.1) and (2.2) results in

$$[\lambda(G_x - R_x) + (R_x - C_x)]^2 + [\lambda(G_y - R_y) + (R_y - C_y)]^2 + [\lambda(G_z - R_z) + (R_z - C_z)]^2 = r^2 \quad (2.3)$$

The line and the sphere may intersect in three possibilities:

- (1) No intersection
- (2) Intersection at only one point
- (3) Intersection at two points

Mathematically, if the quadratic equation (2.3) has no real solution, then the line and the sphere do not intersect (Case (1)). If this equation has only one real solution, then the line intersects the sphere at only one point, i.e., the line is a tangent line to the sphere (Case (2)). If this equation has two real solutions, then the line intersects the sphere at two points (Case (3)).

For Case (1) and Case (2), the mobile robot does not hit the obstacle, and it moves towards the goal without any collision. For Case (3) as shown in Figure 2.4, there is a collision, since the obstacle is in the way of the mobile robot to its goal. To avoid the collision, a feasible path between the current mobile robot position and the goal position needs to be decided. In this study, the path is generated by locating a via-point with the use of the proposed mathematical approach.

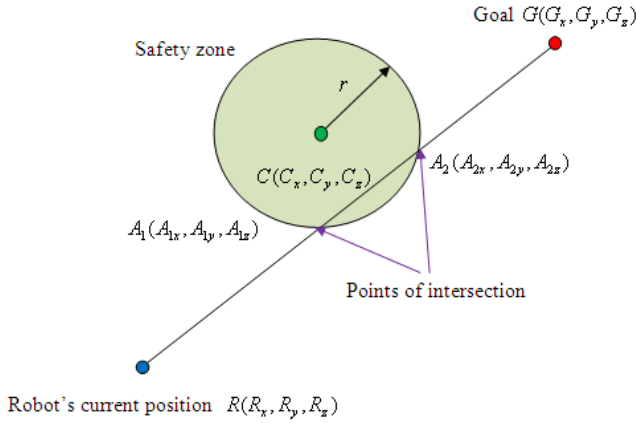


Figure 2.4 Intersection of the line and sphere (Case 3)

C. Mathematical approach to locating the via-point

A novel mathematical approach is proposed to generate the path of the mobile robot in a 3-D environment to reach a randomly moving goal with avoidance of randomly moving obstacles. The key of the mathematical approach involves locating the via-point that is to be aimed at so as to avoid the possible collision with any randomly moving obstacle by using 3-D geometry of spheres and planes. After locating the via-point, the mobile robot moves to the via-point and takes its position as the current position for the next movement.

The via-point $V(V_x, V_y, V_z)$ is a point on the line drawn from the obstacle's center $C(C_x, C_y, C_z)$ to a unique tangent point $T(T_x, T_y, T_z)$ of the sphere and is at a distance of a radius r from the unique tangent point. The unique tangent point $T(T_x, T_y, T_z)$ is one of the intersection points of the tangent lines drawn to the sphere from the mobile robot's current position $R(R_x, R_y, R_z)$. In order to locate the via-point $V(V_x, V_y, V_z)$, the unique tangent point $T(T_x, T_y, T_z)$ needs to be decided first. See Figure 2.5. The steps below are involved to locate the via-point $V(V_x, V_y, V_z)$.

(1) The tangent point $T(T_x, T_y, T_z)$ on the sphere

The tangent point $T(T_x, T_y, T_z)$ is a point on the surface of the sphere. Therefore, the equation of the sphere passing through the tangent point $T(T_x, T_y, T_z)$ with its center at $C(C_x, C_y, C_z)$ and radius r is

$$(T_x - C_x)^2 + (T_y - C_y)^2 + (T_z - C_z)^2 = r^2 \quad (2.4)$$

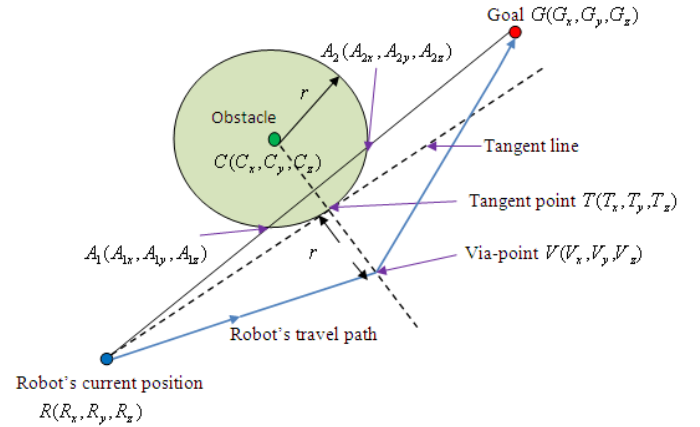


Figure 2.5 Mobile robot's path through the via-point

(2) The tangent plane of the sphere at the tangent point $T(T_x, T_y, T_z)$

The equation of the tangent plane to a surface defined by an implicit equation $f(x, y, z) = 0$ at the point $T(T_x, T_y, T_z)$ is

$$\frac{\partial f}{\partial x}(x - T_x) + \frac{\partial f}{\partial y}(y - T_y) + \frac{\partial f}{\partial z}(z - T_z) = 0 \quad (2.5)$$

Since the surface here is a sphere described by (2.2),

$$f(x, y, z) = (x - C_x)^2 + (y - C_y)^2 + (z - C_z)^2 - r^2 \quad (2.6)$$

Thus,

$$\begin{aligned} \frac{\partial f}{\partial x} &= 2(x - C_x) \\ \frac{\partial f}{\partial y} &= 2(y - C_y) \\ \frac{\partial f}{\partial z} &= 2(z - C_z) \end{aligned} \quad (2.7)$$

Since $T(T_x, T_y, T_z)$ is a point on the surface of the sphere, the equation of the tangent plane becomes

$$\begin{aligned} 2(T_x - C_x)(x - T_x) + 2(T_y - C_y)(y - T_y) \\ + 2(T_z - C_z)(z - T_z) = 0 \end{aligned} \quad (2.8)$$

Simplifying the tangent plane equation (2.8) by using (2.4) leads to

$$\begin{aligned} 2x(T_x - C_x) + 2y(T_y - C_y) + 2z(T_z - C_z) \\ - T_x^2 - T_y^2 - T_z^2 + C_x^2 + C_y^2 + C_z^2 - r^2 = 0 \end{aligned} \quad (2.9)$$

Furthermore, the tangent plane of the sphere at the point $T(T_x, T_y, T_z)$ that also passes through the external point $R(R_x, R_y, R_z)$ is then of the form

$$2R_x(T_x - C_x) + 2R_y(T_y - C_y) + 2R_z(T_z - C_z) - T_x^2 - T_y^2 - T_z^2 + C_x^2 + C_y^2 + C_z^2 - r^2 = 0 \quad (2.10)$$

It should be noted that this tangent plane is drawn at an arbitrary tangent point and is *not* unique. The locus of the tangent planes passing through $R(R_x, R_y, R_z)$ on the sphere is actually a circle along the circumference of the sphere and is defined as circle M . Figure 2.6 shows one of the tangent planes and the circle M .

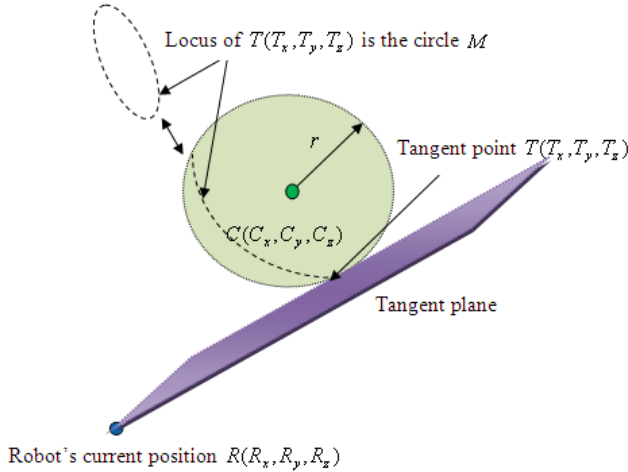


Figure 2.6 One of the tangent planes and the circle M

(3) The plane Π passing through three points $A_1(A_{1x}, A_{1y}, A_{1z})$, $R(R_x, R_y, R_z)$ and $C(C_x, C_y, C_z)$

The plane Π passing through three points $A_1(A_{1x}, A_{1y}, A_{1z})$, $R(R_x, R_y, R_z)$ and $C(C_x, C_y, C_z)$ can be defined as all the points (x, y, z) that satisfy the determinant equations below.

$$\begin{vmatrix} x - A_{1x} & y - A_{1y} & z - A_{1z} \\ R_x - A_{1x} & R_y - A_{1y} & R_z - A_{1z} \\ C_x - A_{1x} & C_y - A_{1y} & C_z - A_{1z} \end{vmatrix} = \begin{vmatrix} x - A_{1x} & y - A_{1y} & z - A_{1z} \\ x - R_x & y - R_y & z - R_z \\ x - C_x & y - C_y & z - C_z \end{vmatrix} = 0 \quad (2.11)$$

The plane Π can also be described as an equation in the form of $Lx + My + Nz - D = 0$ by solving the set of equations below.

$$\begin{aligned} LA_{1x} + MA_{1y} + NA_{1z} - D &= 0 \\ LR_x + MR_y + NR_z - D &= 0 \\ LC_x + MC_y + NC_z - D &= 0 \end{aligned} \quad (2.12)$$

These equations can be solved by using the Cramer's Rule.

$$K = \begin{vmatrix} A_{1x} & A_{1y} & A_{1z} \\ R_x & R_y & R_z \\ C_x & C_y & C_z \end{vmatrix} \quad (2.13)$$

If K is nonzero (for the planes not passing through the origin of the sphere), the coefficients L , M and N are obtained as follows.

$$L = \frac{D}{K} \begin{vmatrix} 1 & A_{1y} & A_{1z} \\ 1 & R_y & R_z \\ 1 & C_y & C_z \end{vmatrix} \quad (2.14)$$

$$M = \frac{D}{K} \begin{vmatrix} A_{1x} & 1 & A_{1z} \\ R_x & 1 & R_z \\ C_x & 1 & C_z \end{vmatrix} \quad (2.15)$$

$$N = \frac{D}{K} \begin{vmatrix} A_{1x} & A_{1y} & 1 \\ R_x & R_y & 1 \\ C_x & C_y & 1 \end{vmatrix} \quad (2.16)$$

These equations are parametric in D . By comparing (2.11) and (2.12), and using (2.13), D is identified as K . Therefore, the equation of the plane Π passing through three points can be obtained as

$$Lx + My + Nz - D = 0 \quad (2.17)$$

Where

$$\begin{aligned} L &= (R_y C_z - R_z C_y) - (A_{1y} C_z - A_{1z} C_y) + (A_{1y} R_z - A_{1z} R_y) \\ M &= (R_z C_x - R_x C_z) + (A_{1x} C_z - A_{1z} C_x) + (A_{1z} R_x - A_{1x} R_z) \\ N &= (R_x C_y - R_y C_x) - (A_{1x} C_y - A_{1y} C_x) + (A_{1x} R_y - A_{1y} R_x) \\ D &= A_{1x} (R_y C_z - R_z C_y) - R_x (A_{1y} C_z - A_{1z} C_y) \\ &\quad + C_x (A_{1y} R_z - A_{1z} R_y) \end{aligned} \quad (2.18)$$

Thus, the equation of this plane Π passing through the tangent point $T(T_x, T_y, T_z)$ is

$$LT_x + MT_y + NT_z - D = 0 \quad (2.19)$$

The intersection of this plane Π and the sphere is also a circle along the circumference of the sphere and is defined as circle N . Figure 2.7 shows the intersection and the circle N . As well known, two circles along the circumference of a sphere have two intersections as long as the two planes where the circles are located are not parallel with each other. Thus, there presents a way to decide the unique tangent point.

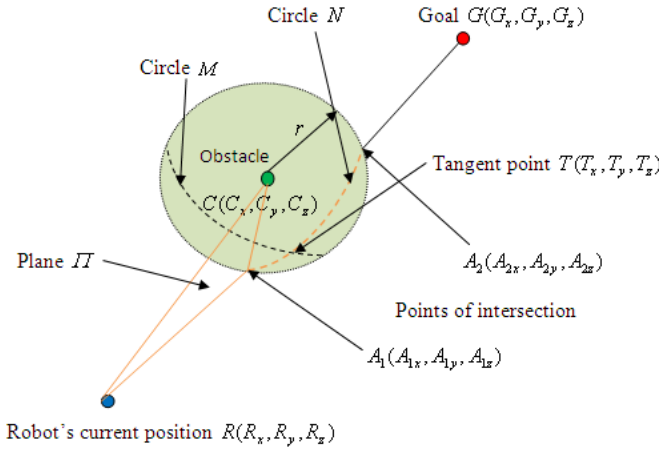


Figure 2.7 Plane II and the circle N

(4) The unique tangent point

The three variables, T_x , T_y and T_z of the unique tangent point can be decided by using three equations, (2.4), (2.10) and (2.19). They are rewritten below for convenience.

$$(T_x - C_x)^2 + (T_y - C_y)^2 + (T_z - C_z)^2 = r^2 \quad (2.4)$$

$$\begin{aligned} 2R_x(T_x - C_x) + 2R_y(T_y - C_y) + 2R_z(T_z - C_z) \\ - T_x^2 - T_y^2 - T_z^2 + C_x^2 + C_y^2 + C_z^2 - r^2 = 0 \end{aligned} \quad (2.10)$$

$$LT_x + MT_y + NT_z - D = 0 \quad (2.19)$$

Expanding (2.4) leads to

$$\begin{aligned} T_x^2 + C_x^2 - 2T_x C_x + T_y^2 + C_y^2 - 2T_y C_y + T_z^2 + C_z^2 - 2T_z C_z = r^2 \\ - T_x^2 - T_y^2 - T_z^2 = C_x^2 + C_y^2 + C_z^2 - r^2 \\ - 2T_x C_x - 2T_y C_y - 2T_z C_z \end{aligned} \quad (2.20)$$

Expanding (2.10) leads to

$$\begin{aligned} 2R_x T_x - 2R_x C_x + 2R_y T_y - 2R_y C_y + 2R_z T_z - 2R_z C_z \\ - T_x^2 - T_y^2 - T_z^2 + C_x^2 + C_y^2 + C_z^2 - r^2 = 0 \end{aligned} \quad (2.21)$$

Substituting (2.20) into (2.21) results in

$$\begin{aligned} 2R_x T_x - 2R_x C_x + 2R_y T_y - 2R_y C_y + 2R_z T_z - 2R_z C_z \\ + C_x^2 + C_y^2 + C_z^2 - r^2 - 2T_x C_x - 2T_y C_y - 2T_z C_z \\ + C_x^2 + C_y^2 + C_z^2 - r^2 = 0 \\ 2T_x(R_x - C_x) + 2T_y(R_y - C_y) + 2T_z(R_z - C_z) \\ - 2R_x C_x - 2R_y C_y - 2R_z C_z + 2C_x^2 + 2C_y^2 + 2C_z^2 - 2r^2 = 0 \end{aligned} \quad (2.22)$$

From (2.22), T_x can be expressed in terms of T_y , T_z and T_z as

$$T_x = \frac{-T_y(R_y - C_y) - T_z(R_z - C_z) + R_x C_x + R_y C_y + R_z C_z - C_x^2 - C_y^2 - C_z^2 + r^2}{(R_x - C_x)}$$

$$\text{i.e., } T_x = \frac{-T_y(R_y - C_y) - T_z(R_z - C_z) + k}{(R_x - C_x)} \quad (2.23)$$

where

$k = R_x C_x + R_y C_y + R_z C_z - C_x^2 - C_y^2 - C_z^2 + r^2$, which is a constant since all the variables at the right side of the expression are known.

Substituting (2.23) into (2.19) results in

$$\begin{aligned} L \left[\frac{-T_y(R_y - C_y) - T_z(R_z - C_z) + R_x C_x + R_y C_y + R_z C_z - C_x^2 - C_y^2 - C_z^2 + r^2}{(R_x - C_x)} \right] \\ + MT_y + NT_z - D = 0 \end{aligned} \quad (2.24)$$

$$\begin{aligned} -LT_y(R_y - C_y) - LT_z(R_z - C_z) \\ + L \left[R_x C_x + R_y C_y + R_z C_z - C_x^2 - C_y^2 - C_z^2 + r^2 \right] \\ + MT_y(R_x - C_x) + NT_z(R_x - C_x) - D(R_x - C_x) = 0 \end{aligned}$$

Thus,

$$\begin{aligned} T_y = T_z \left[\frac{L(R_z - C_z) - N(R_x - C_x)}{M(R_x - C_x) - L(R_y - C_y)} \right] \\ + \left[\frac{D(R_x - C_x) - L(R_x C_x + R_y C_y + R_z C_z - C_x^2 - C_y^2 - C_z^2 + r^2)}{M(R_x - C_x) - L(R_y - C_y)} \right] \end{aligned} \quad (2.25)$$

where

$$k_1 = \left[\frac{L(R_z - C_z) - N(R_x - C_x)}{M(R_x - C_x) - L(R_y - C_y)} \right] \text{ and}$$

$$k_2 = \left[\frac{D(R_x - C_x) - L(R_x C_x + R_y C_y + R_z C_z - C_x^2 - C_y^2 - C_z^2 + r^2)}{M(R_x - C_x) - L(R_y - C_y)} \right]$$

Substituting (2.23) and (2.25) into (2.4) leads to

$$\begin{aligned}
 & \left[\frac{-T_y(R_y - C_y) - T_z(R_z - C_z) + k}{(R_x - C_x)} - C_x \right]^2 \\
 & + (T_y - C_y)^2 + (T_z - C_z)^2 = r^2 \\
 & \left[\begin{aligned} & T_z(-k_1R_y + k_1C_y - R_z + C_z) - k_2R_y + k_2C_y \\ & + k - C_xR_x + C_x^2 \end{aligned} \right]^2 \\
 & + [(T_zk_1 + k_2) - C_y]^2 (R_x - C_x)^2 + (T_z - C_z)^2 (R_x - C_x)^2 \\
 & = r^2 (R_x - C_x)^2 \\
 & (T_zk_3 + k_4)^2 + [(T_zk_1 + k_2) - C_y]^2 (R_x - C_x)^2 \\
 & + (T_z - C_z)^2 (R_x - C_x)^2 \\
 & = r^2 (R_x - C_x)^2
 \end{aligned} \tag{2.26}$$

where $k_3 = -k_1R_y + k_1C_y - R_z + C_z$ and $k_4 = -k_2R_y + k_2C_y + k - C_xR_x + C_x^2$

$$\begin{aligned}
 & T_z^2 k_3^2 + k_4^2 + 2T_z k_3 k_4 \\
 & + [(T_z k_1 + k_2)^2 + C_y^2 - 2(T_z k_1 + k_2)C_y] (R_x - C_x)^2 \\
 & + [T_z^2 + C_z^2 - 2T_z C_z] (R_x - C_x)^2 = r^2 (R_x - C_x)^2 \\
 & T_z^2 k_3^2 + k_4^2 + 2T_z k_3 k_4 \\
 & + [T_z^2 k_1^2 + k_2^2 + 2T_z k_1 k_2 + C_y^2 - 2T_z k_1 C_y - 2k_2 C_y] \\
 & (R_x - C_x)^2 + [T_z^2 + C_z^2 - 2T_z C_z] (R_x - C_x)^2 = r^2 (R_x - C_x)^2 \\
 & T_z^2 [k_3^2 + k_1^2 (R_x - C_x)^2 + (R_x - C_x)^2] \\
 & + T_z \left[\begin{aligned} & 2k_3 k_4 + 2k_1 k_2 (R_x - C_x)^2 - 2k_1 C_y (R_x - C_x)^2 \\ & - 2C_z (R_x - C_x)^2 \end{aligned} \right] \\
 & + k_4^2 + [k_2^2 + C_y^2 - 2k_2 C_y + C_z^2 - r^2] (R_x - C_x)^2 = 0
 \end{aligned} \tag{2.27}$$

Eq. (2.27) can be expressed in the form

$$AT_z^2 + BT_z + C = 0$$

Where

$$\begin{aligned}
 A &= k_3^2 + (k_1^2 + 1)(R_x - C_x)^2, \\
 B &= 2k_3 k_4 + [2k_1 k_2 - 2k_1 C_y - 2C_z] (R_x - C_x)^2 \text{ and } C = k_4^2 + \\
 & [k_2^2 + C_y^2 - 2k_2 C_y + C_z^2 - r^2] (R_x - C_x)^2
 \end{aligned}$$

$$\text{Therefore, } T_z = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{2.28}$$

As noticed, (2.28) gives two solutions of T_z .

Furthermore, as mentioned before, T_y can be obtained from (2.25), and T_x can be obtained by substituting T_y and T_z into (2.23).

Since (2.28) gives two solutions, two tangent points are therefore decided. However, only one of them needs to be selected to serve as the via-point. In this study, to reduce the moving time for the mobile robot, the tangent point nearest to the goal is selected. Suppose the two

tangent points are $T_1(T_{x1}, T_{y1}, T_{z1})$ and $T_2(T_{x2}, T_{y2}, T_{z2})$, and the goal position is $G(G_x, G_y, G_z)$. Then the distance between $T_1(T_{x1}, T_{y1}, T_{z1})$ and $G(G_x, G_y, G_z)$ is

$$D_1 = \sqrt{(T_{x1} - G_x)^2 + (T_{y1} - G_y)^2 + (T_{z1} - G_z)^2}$$

and the distance between $T_2(T_{x2}, T_{y2}, T_{z2})$ and $G(G_x, G_y, G_z)$ is

$$D_2 = \sqrt{(T_{x2} - G_x)^2 + (T_{y2} - G_y)^2 + (T_{z2} - G_z)^2}$$

If D_1 is less than D_2 , $T_1(T_{x1}, T_{y1}, T_{z1})$ is decided as the unique tangent point and otherwise $T_2(T_{x2}, T_{y2}, T_{z2})$ is.

(5) The via-point

The equation of a line passing through two points, the unique tangent point $T(T_x, T_y, T_z)$ and the obstacle's center $C(C_x, C_y, C_z)$, is

$$\frac{x - T_x}{T_x - C_x} = \frac{y - T_y}{T_y - C_y} = \frac{z - T_z}{T_z - C_z} = \beta \tag{2.29}$$

where β is a constant. (2.29) can be written as

$$\begin{aligned}
 x &= T_x + \beta(T_x - C_x) \\
 y &= T_y + \beta(T_y - C_y) \\
 z &= T_z + \beta(T_z - C_z)
 \end{aligned} \tag{2.30}$$

By choosing different values of β , different points can be located on the line. The via-point is a point on this line, which is located at a distance of the radius r from the unique tangent point. That is, the via-point can be located by choosing $\beta = 1$. In other words, the coordinates of the via-point $V(V_x, V_y, V_z)$ are obtained by

$$\begin{aligned}
 V_x &= T_x + (T_x - C_x) \\
 V_y &= T_y + (T_y - C_y) \\
 V_z &= T_z + (T_z - C_z)
 \end{aligned} \tag{2.31}$$

Now, the via-point is located. Whenever the mobile robot detects a possible collision, the mobile robot aims at this via-point on its way to reach the goal.

D. Special cases

The mathematical approach presented in this study detects any possible collision and locates the via-point if any obstacle is in its way. However, there is a special case that the obstacle is coming from any other side towards the mobile robot's current position as the example shown in Figure 2.8. In Figure 2.8, the mobile robot, goal and obstacle 3 are moving in the same direction, and obstacle 1 is moving towards the mobile robot's current position. Obstacle 3 is located on the line created by the mobile robot's current position and the

goal. According to the mathematical approach, the mobile robot detects a possible collision with obstacle 3 and locates a via-point. Unfortunately, when the mobile robot is moving towards the via-point, obstacle 1 could be too close to the mobile robot or may have a collision with it.

To avoid this kind of collision, the mobile robot decides the distance between its current position and each obstacle every time. The distance between the mobile robot's current position $R(R_x, R_y, R_z)$ and the obstacle 1's position $C_1(C_{x1}, C_{y1}, C_{z1})$ is

$$RC_1 = \sqrt{(R_x - C_{x1})^2 + (R_y - C_{y1})^2 + (R_z - C_{z1})^2} \quad (2.32)$$

Similarly, it decides RC_2, RC_3, RC_4 and RC_5 for each of the obstacles, respectively. If any of these distances is less than $r + 0.05$ (where r is the radius of the safety zone), the mobile robot is then going to move in the positive or negative direction of the unit movement vector of the obstacle in concern, which can be obtained from the current position and the previous position of the obstacle. In Figure 2.8, the current and previous position of obstacle 1 are $C_1(C_{x1}, C_{y1}, C_{z1})$ and $C_{p1}(C_{px1}, C_{py1}, C_{pz1})$, respectively. The vector $C_1 - C_{p1}$ is then,

$$C_1 - C_{p1} = (C_{x1} - C_{px1}, C_{y1} - C_{py1}, C_{z1} - C_{pz1}) \quad (2.33)$$

The unit vector of $C_1 - C_{p1}$ is therefore

$$\vec{c_1 - c_{p1}} = \frac{C_1 - C_{p1}}{\|C_1 - C_{p1}\|} \quad (2.34)$$

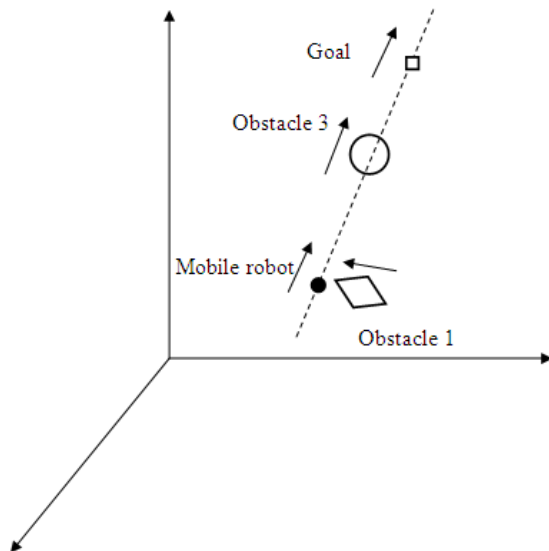


Figure 2.8 Special case 1

To understand the above concepts better, numerical examples are presented.

1. Suppose, at a certain instant, there are $R(1.0, 1.0, 2.0)$, $C_1(3.0, 2.5, 2.0)$ and $C_{p1}(3.1, 2.5, 2.0)$. Thus, the unit vector of $(C_1 - C_{p1}) = (-1, 0, 0)$.

(1) The distance between R and C_1 is $\sqrt{(1.0 - 3.0)^2 + (1.0 - 2.5)^2 + (2.0 - 2.0)^2} = 2.5 < r + 0.05 = 2.5 + 0.05 = 2.55$. Thus, $C_1(3.0, 2.5, 2.0)$ is judged to be too close to the mobile robot or may have a collision with it.

(2) The next position of the mobile robot R_n , if moving in the *positive* direction of the unit movement vector $(-1, 0, 0)$, would be $(0.8, 1.0, 2.0)$. Then, the distance between R_n and C_1 would be $\sqrt{(0.8 - 3.0)^2 + (1.0 - 2.5)^2 + (2.0 - 2.0)^2} = 2.66 > r + 0.05 = 2.5 + 0.05 = 2.55$. Thus, the mobile robot follows the direction $(-1, 0, 0)$ to go to $R_n = (0.8, 1.0, 2.0)$.

2. Suppose, at a certain instant, there are $R(1.0, 1.0, 2.0)$, $C_1(3.0, 2.5, 2.0)$ and $C_{p1}(2.9, 2.5, 2.0)$. Thus, the unit vector of $(C_1 - C_{p1}) = (1, 0, 0)$.

(1) The distance between R and C_1 is $\sqrt{(1.0 - 3.0)^2 + (1.0 - 2.5)^2 + (2.0 - 2.0)^2} = 2.5 < r + 0.05 = 2.5 + 0.05 = 2.55$. Thus, $C_1(3.0, 2.5, 2.0)$ is judged to be too close to the mobile robot or may have a collision with it.

(2) The next position of the mobile robot R_n , if moving in the *positive* direction of the unit movement vector $(1, 0, 0)$, would be $(1.2, 1.0, 2.0)$. Then, the distance between R_n and C_1 would be $\sqrt{(1.2 - 3.0)^2 + (1.0 - 2.5)^2 + (2.0 - 2.0)^2} = 2.34 < r + 0.05 = 2.5 + 0.05 = 2.55$. Thus, the mobile robot *cannot* follow the direction $(1, 0, 0)$ to go to $R_n = (1.2, 1.0, 2.0)$.

(3) The next position of the mobile robot R_n , if moving in the *negative* direction of the unit movement vector $(1, 0, 0)$, would be $(0.8, 1.0, 2.0)$. Then, the distance between R_n and C_1 would be $\sqrt{(0.8 - 3.0)^2 + (1.0 - 2.5)^2 + (2.0 - 2.0)^2} = 2.66 > r + 0.05 = 2.5 + 0.05 = 2.55$. Thus, the mobile robot follows the direction $(1, 0, 0)$ to go to $R_n = (0.8, 1.0, 2.0)$.

From the examples above, it is seen that, in this special case, it is absolutely depending on the current movement direction of the obstacle in concern for the mobile robot to decide whether to move in the positive or the negative direction of the unit movement vector of the obstacle.

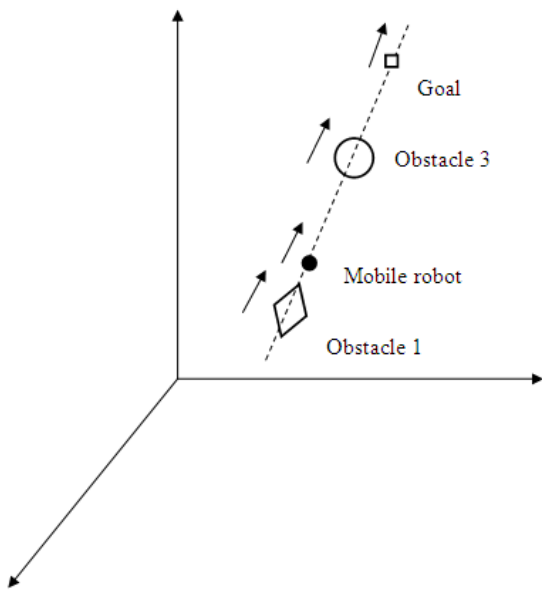


Fig.2.9 Special case 2

Another kind of special case arises when obstacle 1 is on the line created by the mobile robot and the goal and is also moving towards the mobile robot's current position as shown in Figure 2.9.

According to the mathematical approach, the mobile robot locates a via-point only if the obstacle is in between the mobile robot and the goal. For the present case, the mobile robot locates the via-point to avoid obstacle 3 and moves towards that via-point. However, in the future, there is a chance of collision with obstacle 1, if it continuously moves on the same line and towards the mobile robot's current position. This case is essentially the same as the special case before. The only specialty in this case is that the mobile robot detects a possible collision with obstacle 1 that is on the created line but not between the mobile and the goal. In Figure 2.9, if obstacle 1 is far away from the mobile robot's current position though is on the created line, the mobile robot can ignore obstacle 1 and locates the via-point to avoid obstacle 3. If the distance between the mobile robot and obstacle 1 is less than $r + 0.05$, the mobile robot is to move in the positive or the negative direction of the unit movement vector of the obstacle 1 in the same way as shown in the special case before.

E. Algorithm and flowchart

The steps below give an idea of the algorithm used in this study. Each step is also explained in some detail.

- (1) Start.
- (2) Decide the new positions of mobile robot, goal, and obstacles (from sensors).
- (3) Define the mobile robot's direction of movement towards the goal.
- (4) Check for any possible collision with obstacles in the near future. If a possible collision is detected, go to step (5).
- (5) Else: go to step (6).

- (5) Locate the via-point to avoid the closest obstacle in the way of the mobile robot and change the mobile robot's direction of movement towards the via-point.
- (6) Check for special cases. If there are special cases, go to step (7).
- Else: go to step (8).
- (7) Change the mobile robot's direction of movement towards the positive or negative unit vector of the obstacle.
- (8) Move the mobile robot along its direction of movement decided by (3), (5) or (7).
- (9) If the goal is reached, go to (10).
- Else: go to step (1).
- (10) Stop.

The mobile robot finds its own current position and the current positions of the goal and obstacles from sensors. The mobile robot's direction of movement is then defined towards the goal. The mobile robot creates a 3-D line of infinite length between itself and its goal. For the mobile robot, all the obstacles are in sphere shapes because of the defined safety zone for each obstacle. The intersections of the aforementioned line and the sphere are decided to detect the possible collision as explained in Subsection B. For a two point intersection, the mobile robot decides that there is going to be a collision and then locates a via-point. The mobile robot's direction of movement is then changed towards the via-point. The mobile robot also checks for special cases in every iteration as explained in Subsection D. If there is going to be any special case, the mobile robot changes its direction of movement towards the positive or negative unit vector of the obstacle. Then, the mobile robot moves along its direction of movement in any one of the defined directions as shown in the flowchart depending upon the situation. If the mobile robot reaches the goal, the algorithm is stopped; otherwise, iterations will continue.

It is worth emphasizing that each time the goal position changes randomly, the mobile robot's direction of movement also changes accordingly. Similarly, as an obstacle moves randomly, the mobile robot locates a new via-point in the present iteration and changes its direction of movement towards this new via-point. Figure 2.10 presents the flowchart for the mathematical approach to reaching the randomly moving goal while avoiding randomly moving obstacles.

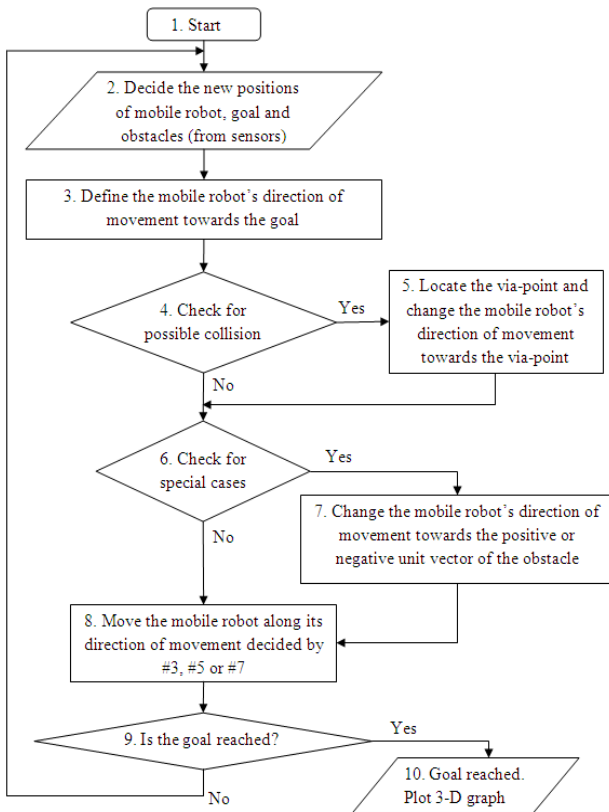


Figure 2.10 Flowchart for the mathematical approach

III. SIMULATION RESULTS

This section presents the simulation results. To represent 3-D graphs in two dimensions, the terms Azimuth (Az) and Elevation (El) are used. Azimuth (Az) is a polar angle in the x-y plane, with positive angles indicating counterclockwise rotation of the viewpoint. Elevation (El) is the angle above (positive angle) or below (negative angle) the x-y plane. Az and El are expressed in degrees throughout. Figure 3.1 illustrates Az and El in the 3-D coordinate system. The Az and El arrows indicate positive directions.

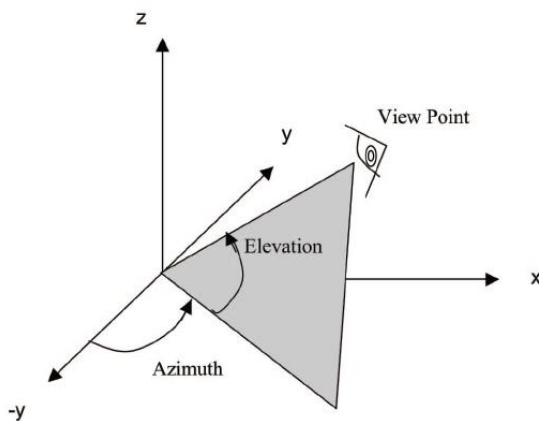


Figure 3.1 The Azimuth and Elevation

(1). Figure 3.2 gives the first set of simulation results, where the black solid dot represents the mobile robot, and the square represents the goal. The small solid dots represent the path traveled by the mobile robot, and the solid line represents the trail of the goal. The obstacles shown in this set of simulations are “wrapped” by invisible safety zones with a radius of 2.5 each. The “diamond”, “downward triangle”, “circle”, “plus” and “six-pointed star” stand for obstacles 1-5, respectively. In this study, the velocities of the goal and obstacles are taken as 0.1/iteration, and the velocity of the mobile robot is taken as 0.2/iteration.

In Figure 3.2(a), the initial positions of the mobile robot $R(1,12,31)$, goal $G(30,26,23)$, and five obstacles are shown. In Figure 3.2(b), the mobile robot is on its way to reach a randomly moving goal while all obstacles are also moving randomly. The mobile robot's path and the trail of the goal can be seen. In Figure 3.2(c)-(d), the paths traveled by the mobile robot at the 110th iteration with different Az and El are seen. Figures 3.2(e)-(g) present the path traveled by the mobile robot at the 233th (final) iteration with different Az and El.

(2). Figure 3.3 gives the second set of simulation results. The obstacles shown in this set of simulations are “wrapped” by invisible safety zones with a radius of 4.5 each. In Figure 3.3(a), the initial positions of the mobile robot $R(11,5,9)$, goal $G(37,32,15)$, and five obstacles are shown. In Figure 3.3(b), the mobile robot is on its way to reach a randomly moving goal while all obstacles are also moving randomly. The mobile robot's path and the trail of the goal can be seen. In Figure 3.3(c)-(d), the paths traveled by the mobile robot at the 140th iteration with different Az and El are seen. Figures 3.3(e)-(g) present the path traveled by the mobile robot at the 212th (final) iteration with different Az and El.

From Figure 3.3(e)-(g), it is clear that the mobile robot has reached the randomly moving goal successfully even though all the obstacles that have enlarged safety zones are moving randomly near the goal. For larger obstacles, safety zones are also enlarged so as to “wrap” the obstacle completely. It is seen then that the developed scheme works well even for large obstacles.

(3). Figure 3.4 gives the third set of simulation results. The obstacles shown in this set of simulations are “wrapped” by visible safety zones with a radius of 4.5 each. In Figure 3.4(a), the initial positions of the mobile robot $R(11,5,9)$, goal $G(37,32,15)$, and five obstacles that are the same as those in Figure 3.3(a) are shown. Also, it is seen that the goal is hidden under obstacle 4. In Figure 3.4(b), the mobile robot is on its way to reach a randomly moving goal while all obstacles are also moving randomly. The mobile robot's path and the trail of the goal can be seen. It is seen again that the goal is still hidden under obstacle 4. In Figure 3.4(c)-(d), the paths traveled by the mobile robot at the 140th iteration with different Az and El are seen. Figures 3.4(e)-(g) present the path traveled by the mobile robot at the 212th (final) iteration with different Az and El.

As seen, in the generation of the path for the mobile robot, if the mobile robot detects any possible collision with any obstacle (say obstacle 5), it will locate the via-point and move towards the via-point to avoid obstacle 5. In the next iteration, the mobile robot will still check for any possible collision with any obstacle again, since it is very possible that obstacle 5 is still in its way. The mobile robot will locate the new via-point at this time and move towards the new via-point to avoid obstacle 5. This is because of the change in obstacle 5's position all the time, i.e., the random movement of obstacle 5. After a certain period of time, obstacle 5 may completely go away from the mobile robot's path. It is very rare for the mobile robot to go to a via-point and then to the goal directly unless there are stationary or slowly moving obstacles. For a stationary or slowly moving obstacle, the path of the mobile robot is described as shown in Figure 2.5.

It should be noted that, in the generation of the path for the mobile robot, since all the bodies, i.e., the mobile robot, the goal as well as the multiple obstacles, are moving, it is difficult to catch by eyes whether any collision is occurring. To solve this problem, an alarm is set in MATLAB. As long as the mobile robot collides with any obstacle by any chance, the alarm will sound and the program will be stopped. Numerous real-time simulations with different settings are performed successively. Due to limitation of space, only three sets of simulation results are presented in this article,

IV. CONCLUSION

In this article, the proposed mathematical approach is implemented in real-time to generate a 3-D path for the mobile robot to track and reach a randomly moving goal while avoiding all five randomly moving obstacles. The developed scheme is effective with the drawbacks of the dynamic potential field method eliminated. The relative velocities of the mobile robot, goal, and obstacles are considered in the generation of the path. This method can be essentially implemented in real life to avoid randomly moving obstacles to reach a randomly moving goal. MATLAB simulation validates the proposed method.

Future work could be the consideration of a larger number of obstacles with different velocities. Moreover, multiple mobile robots can be considered in place of a single mobile robot.

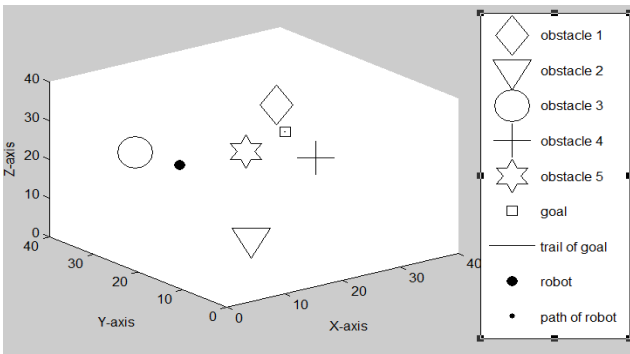


Figure 3.2(a) Initial positions of the mobile robot, goal, and five obstacles ($Az=-38^{\circ}$, $El=30^{\circ}$)

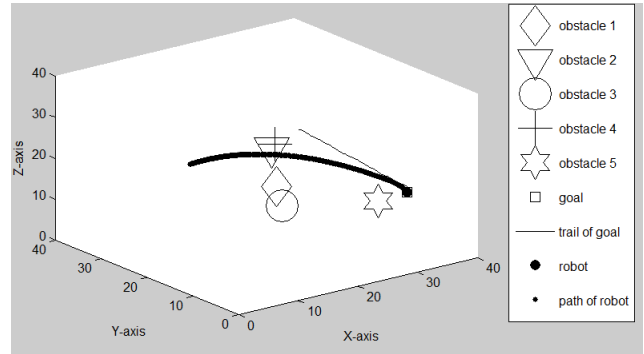


Figure 3.2(e) Path traveled by the mobile robot at the 233th (final) iteration ($Az=-38^{\circ}$, $El=30^{\circ}$)

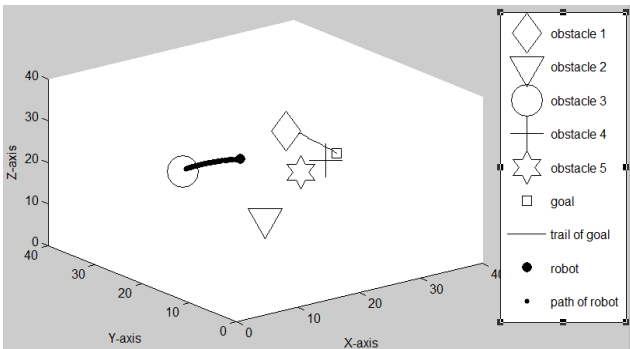


Figure 3.2(b) Path traveled by the mobile robot at the 70th iteration ($Az=-38^{\circ}$, $El=30^{\circ}$)

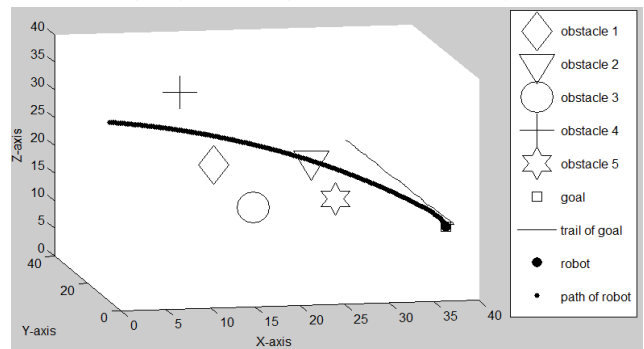


Figure 3.2(f) Path traveled by the mobile robot at the 233th (final) iteration ($Az=-11^{\circ}$, $El=14^{\circ}$)

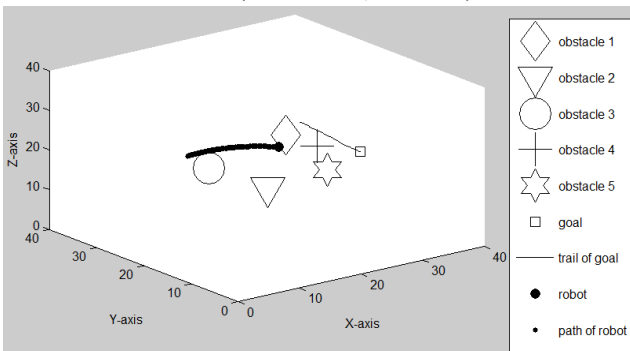


Figure 3.2(c) Path traveled by the mobile robot at the 110th iteration ($Az=-38^{\circ}$, $El=30^{\circ}$)

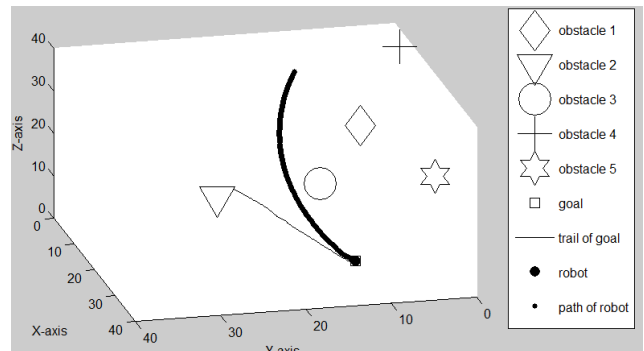


Figure 3.2(g) Path traveled by the mobile robot at the 233th (final) iteration ($Az=-77^{\circ}$, $El=32^{\circ}$)

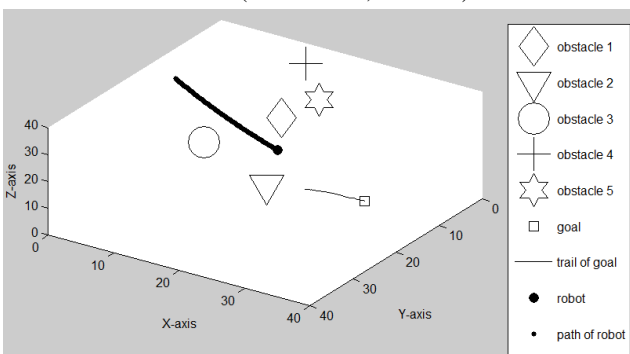


Figure 3.2(d) Path traveled by the mobile robot at the 110th iteration ($Az=-34^{\circ}$, $El=-50^{\circ}$)

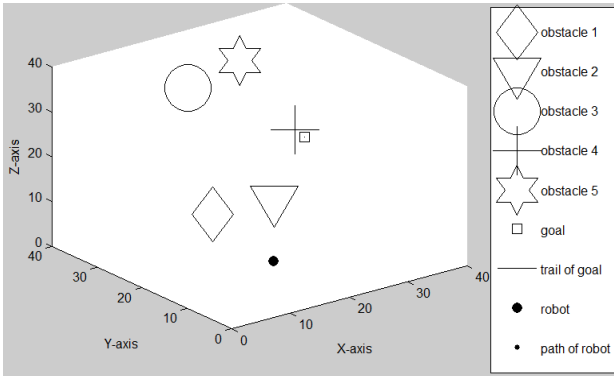


Figure 3.3(a) Initial positions of the mobile robot, goal, and five obstacles ($Az=-38^{\circ}$, $El=30^{\circ}$)

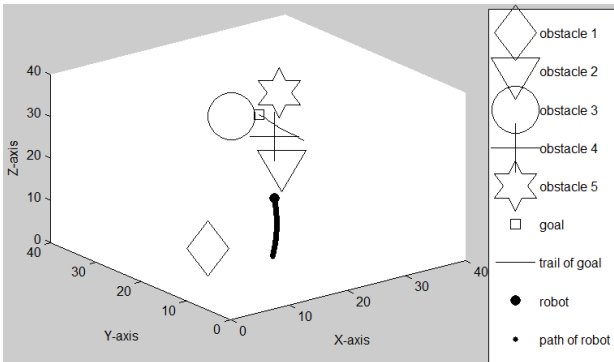


Figure 3.3(b) Path traveled by the mobile robot at the 87th iteration ($Az=-38^{\circ}$, $El=30^{\circ}$)

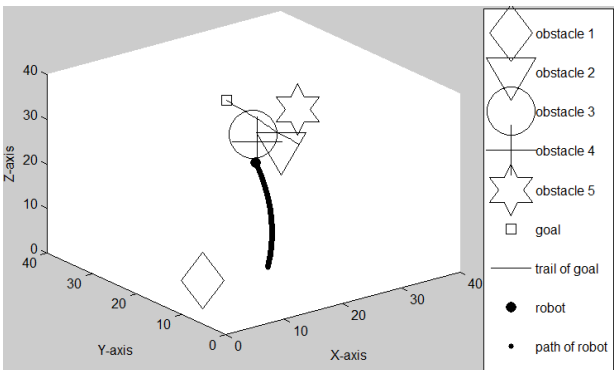


Figure 3.3(c) Path traveled by the mobile robot at the 140th iteration ($Az=-38^{\circ}$, $El=30^{\circ}$)

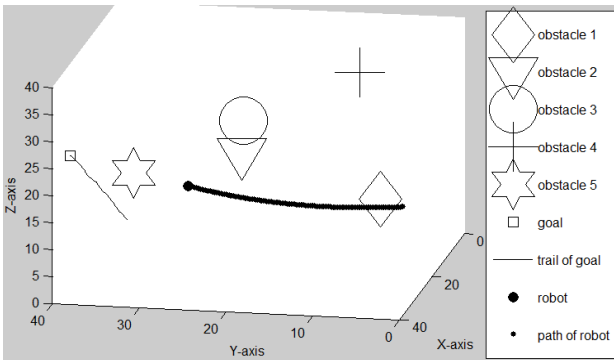


Figure 3.3(d) Path traveled by the mobile robot at the 140th iteration ($Az=-101^{\circ}$, $El=-22^{\circ}$)

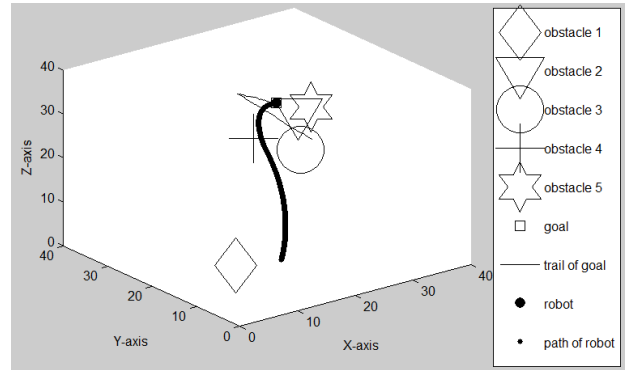


Figure 3.3(e) Path traveled by the mobile robot at the 212th (final) iteration ($Az=-38^{\circ}$, $El=30^{\circ}$)

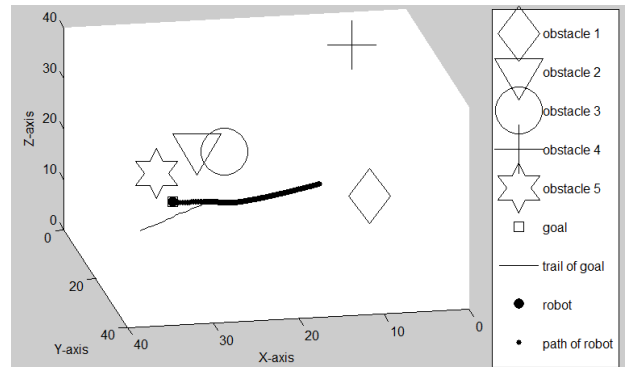


Figure 3.3(f) Path traveled by the mobile robot at the 212th (final) iteration ($Az=-121^{\circ}$, $El=26^{\circ}$)

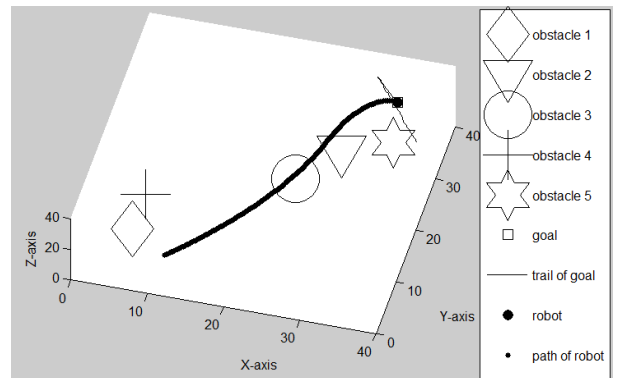


Figure 3.3(g) Path traveled by the mobile robot at the 212th (final) iteration ($Az=15^{\circ}$, $El=74^{\circ}$)

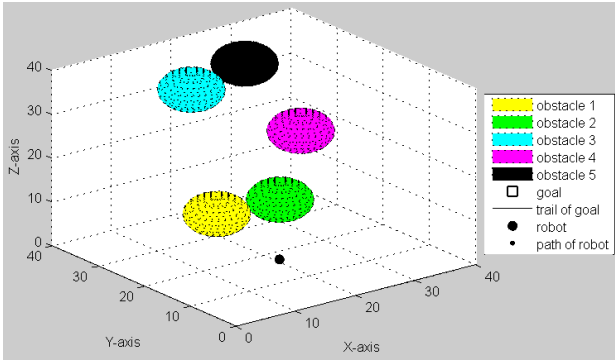


Figure 3.4(a) Initial positions of the mobile robot, goal, and five obstacles ($Az=-38^\circ, El=30^\circ$)

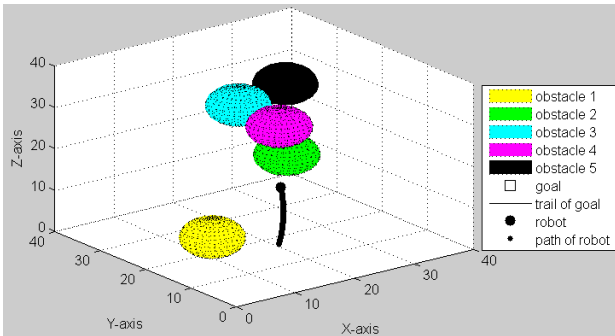


Figure 3.4(b) Path traveled by the mobile robot at the 87th iteration ($Az=-38^\circ, El=30^\circ$)

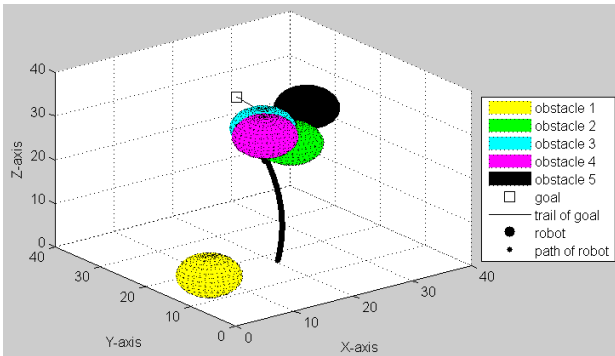


Figure 3.4(c) Path traveled by the mobile robot at the 140th iteration ($Az=-38^\circ, El=30^\circ$)

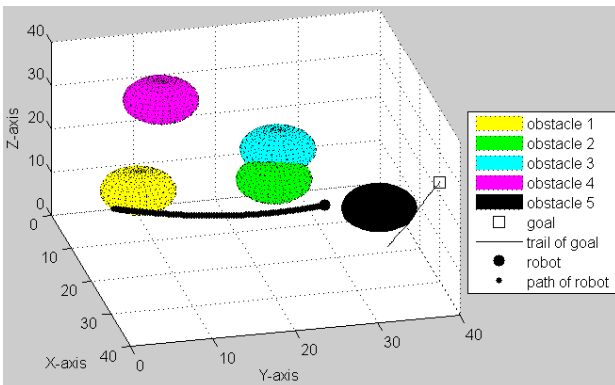


Figure 3.4(d) Path traveled by the mobile robot at the 140th iteration ($Az=77^\circ, El=38^\circ$)

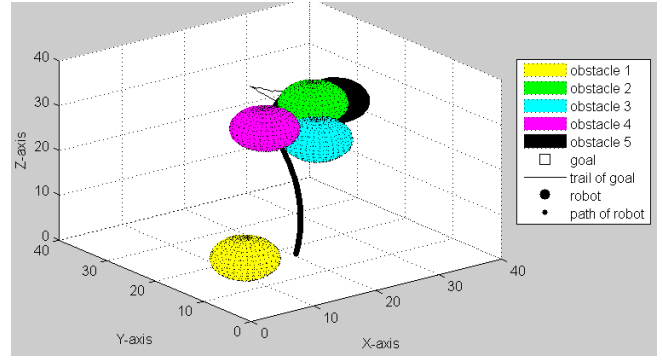


Figure 3.4(e) Path traveled by the mobile robot at the 212th (final) iteration ($Az=-38^\circ, El=30^\circ$)

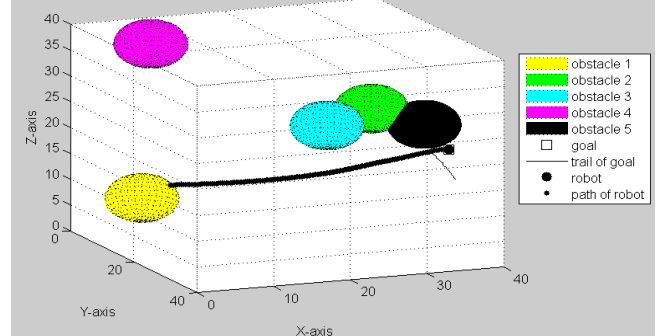


Figure 3.4(f) Path traveled by the mobile robot at the 212th (final) iteration ($Az=23^\circ, El=18^\circ$)

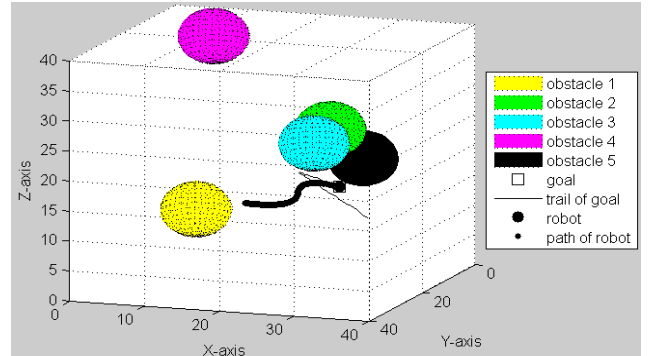


Figure 3.4(g) Path traveled by the mobile robot at the 212th (final) iteration ($Az=-20^\circ, El=-14^\circ$)

REFERENCES

- [1] Hachour Ouarda, "Intelligent Autonomous Path Planning Systems," *International Journal of Systems Applications, Engineering and Development*, Issue 3, Vol. 5, pp. 377-386, 2011
- [2] Elena Garcia, Marcia Antonia Jimenez, and Pablo Gonzalez De Santos, Manuel Armada, "The Evolution of Mobile robotics Research," *Industrial Mobile robotics to Field and Service-IEEE Mobile robotics and Automation Magazine*, pp.90-103, March 2007
- [3] Vamsi Polisetty, "ANFIS Generated Dynamic Path Planning for Mobile robots in a 3-D Environment," Diss. Northern Illinois U, 2009
- [4] John F. Canny, "The Complexity of Mobile robot Motion Planning," *Cambridge, MA: MIT Press*, 1988
- [5] Latombe, "Mobile robot Motion Planning", *Norwell, MA: Kluwer*, 1991
- [6] Y. Koren and J. Borenstein, "Potential Field Methods and Their Inherent Limitations for Mobile robot Navigation," *Proceedings of the IEEE Conference on Mobile robotics and Automation*, pp. 1398-1404, April 1991
- [7] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile robots," *Reprinted, with permission, from IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, pp. 1179-1187, Sept./Oct. 1989
- [8] J. Borenstein and Y. Koren, "Real-Time Obstacle Avoidance for Fast Mobile robots in Cluttered Environments," *Reprint of Proceedings of the 1990 IEEE International Conference on Mobile robotics and Automation*, Cincinnati, Ohio, pp. 572-577, May 13-18, 1990
- [9] R.A. Brooks, "Solving the find-path problem by good representation of free space," *IEEE Transactions on Systems, Man, Cybernetics*, Vol. SMC-13, pp. 190-197, March 1983
- [10] M. Herman, "Fast, three-dimensional, collision free motion planning," *Proceedings of IEEE International Conference on Mobile robotics and Automation*, San Francisco, CA, 1056-1063, April 1986
- [11] Kikuo Fujimura and Hanan Samet, "A Hierarchical Strategy for Path Planning Among Moving Obstacles," *IEEE Transactions on Mobile robotics and Automation*, Vol. 5, No. 1, pp. 61-69, February 1989
- [12] Animesh Chakravarthy and Debaisish Ghose, "Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach," *562 IEEE Transactions on Systems, Man, Cybernetics-Part A: Systems and Humans*, Vol. 28, No. 5, pp. 562-574, September 1998
- [13] S.S. Ge and Y.J. Cui, "Dynamic Motion Planning for Mobile robots Using Potential Field Method," *Autonomous Mobile robots 13*, pp. 207-222, 2002
- [14] Jur P. van den Berg and Mark H. Overmars, "Roadmap-Based Motion Planning in Dynamic Environments," *IEEE Transaction on Mobile robotics*, Vol. 21, No. 5, pp. 885-897, October 2005
- [15] Marija Seder and Ivan Petrovic, "Dynamic Window Based Approach to Mobile robot Motion Control in the Presence of Moving Obstacles," *Proceedings of IEEE International Conference on Mobile robotics and Automation*, Roma, Italy, pp. 1986-1992, April 2007
- [16] Ellips Masehian and Yalda Katebi, "Mobile robot Motion Planning in Dynamic Environments with Moving Obstacles and Target," *World Academy of Science, Engineering and Technology 29*, pp. 107-112, 2007
- [17] Fethi Belkhouche, "Reactive Path Planning in a Dynamic Environment," *IEEE Transaction on Mobile robotics*, Vol. 25, No. 4, pp. 902-911, August 2009
- [18] Xiaoping Yun and Ko-Cheng Tan, "A Wall-Following Method for Escaping Local Minima in Potential Field Based Motion Planning," *Advanced Mobile robotics, ICAR '97. Proceedings., 8th International Conference on*, pp. 421-426, July 7-9, 1997

Vanitha Aenugu received B.E degree in electrical and electronics engineering from Osmania University, India, in 2008. She also received M.S. degree in electrical engineering from Northern Illinois University, IL, USA in 2011. During her Master's education, she did research on path planning of a mobile robot. Her current research area includes robotics and digital signal processing. She is actively looking for employment opportunities in robotics/digital signal processing field.

Peng-Yung Woo received the B.S. degree in physics /electrical engineering from Fudan University, Shanghai, China in 1982 and the M.S. degree in electrical engineering from Drexel University, Philadelphia, PA, in 1983. In 1988, he received the Ph.D. degree in system engineering from the University of Pennsylvania, PA. He joined Northern Illinois University in 1988 and is currently a Professor of Electrical Engineering. His current research interests include intelligent control, robotics, digital signal processing, neural networks, fuzzy systems and digital image processing. He has authored and co-authored over 50 international journal articles and book sections as well as over 60 international conference papers. He is a senior member of the *IEEE* and the member of the *IASTED* Standing Technical Committee. He has been the member of many International Program Committees for various *IEEE* and *IASTED* International Conferences. He is currently also an Advisory Professor of Tongji University, China.

How to cite this paper: Vanitha Aenugu, Peng-Yung Woo, "Mobile Robot Path Planning with Randomly Moving Obstacles and Goal", *International Journal of Intelligent Systems and Applications(IJISA)*, vol.4, no.2, pp.1-15, 2012. DOI: 10.5815/ijisa.2012.02.01