

Autonomous Mobile Robot Navigation Using Hybrid Virtual Force Field Concept

V.O.S. Olunloyo

Department of Systems Engineering, University of Lagos, Lagos, Nigeria
E-mail: vosolunloyo@hotmail.com

M.K.O. Ayomoh

Department of Systems Engineering, University of Lagos, Lagos, Nigeria
E-mail: mikeayomoh@yahoo.com

Abstract

A path planning and obstacle avoidance scheme for autonomous mobile robot in a partially known 2-D environment is presented in this paper. The navigation environment comprises static obstacles and a robot desired goal state. A new methodology christened, Hybrid Virtual Force Field (HVFF), which is an integration of the Virtual Obstacle Concept (VOC) and the Virtual Goal Concept (VGC) in combination with the traditional Virtual Force Field concept is proposed for an effective robot path planning. The specific problem addressed in this paper is the local minima problem posed by concave shaped obstacles. The validation process for the HVFF scheme was carried out on workspaces developed by an earlier worker Sedighi et al. who used genetic algorithm to solve a similar problem. The simulation experiments which demonstrate the superiority of HVFF over earlier evolutionary schemes were implemented using MobotSim software, a customized mobile robot simulator, on an Intel Pentium®4. system

Keywords: Virtual obstacle concept, virtual goal concept, virtual force field, concave obstacles, partially known environment,

1. Introduction

The last few decades have recorded a paradigm shift from remotely operated to autonomous Robotic vehicles. Research to enhance autonomous navigation has accordingly received considerable attention as seen from recent literature. In this scheme of autonomy, the robot travels between a starting point and a target point without the need for human intervention. Ordinarily, the problem of navigation could be summarized in the following three questions namely: where am I?", where am I going?", and how should I get there?" The first question is one of localization; the second question is on goal state description while the last question comes under the domain of path planning and obstacle avoidance. The path-planning problem is usually defined as follows: "Given a robot and a description of an environment, plan a path between two specific locations. The path must be collision-free (feasible) and satisfy certain optimization criteria" Sugihara and Smith (1997).

Robot Path planning, in a broad sense, refers to the determination of how a Robot will maneuver within an environment (or workspace) to achieve its objectives. The path planning problem involves computing a collision free path between two locations. Path planning for mobile Robots is a

complex problem that does not just guarantee a collision-free path with minimum traveling distance but also requires smoothness and clearances. Path planning could be either local or global. Local path planning means that path planning is done while the robot is moving; in other words, the algorithm is capable of producing a new path in response to environmental changes. On the other hand, Global path planning requires that the environment be completely known and all the features present within the terrain remain static. In this approach the algorithm generates a complete path from the start point to the destination point before the robot starts its motion.

Researchers, over the years, have developed and used different concepts to solve the robot path planning problem. These concepts have been categorized into the following sub-classes namely: the graphical approach, the classical methods, the heuristic approach as well as the use of the potential field methods.

Graphical techniques such as: voronoi diagram, spatial planning, vgraph, transformed space, oct trees, cell decomposition and configuration space have been proposed and extensively used in the past few decades. Graphical methods are more often than not developed on the principles of geometry and they are dominantly known for their limitation to static environment. In his work, Lozano-Perez (1987) discussed a generate and test strategy which is used for path planning. Another well known graphical approach is the Configuration Space method used by Laumond (1986), Palma-Villalon and Dauchez (1988) and Tseng et al. (1988). On the other hand, Takahashi and Schilling (1989); as well as Song et al. (2001) suggested the Voronoi Diagram as an approach for representing space. Lingelbach (2004a;2004b) proposed a novel path planning method called Probabilistic Cell Decomposition (PCD) while Gayle et al. (2007) also proposed a new roadmap representation scheme christened Reactive Deforming Roadmap (RDR), for multiple robots.

A second approach which has also been widely explored over the years is the use of classical algorithms. Some of the classical methods that have been used over time include: gradient descent, steepest descent, steepest ascent, dynamic programming, iterative technique, randomized methods, kinodynamic, fibonacci Improved Network Optimization algorithms etc. Asaolu (2001) applied Dijkstra's algorithm to solve the shortest route problem and furthermore developed a new approach christened the intercept approach which was used to solve the pursuit problem. Also, some classical and geometrical approach were deployed in the obstacle avoidance problem. Ibidapo-Obe et al. (1999;2002;2006) used some classical, numerical and optimization techniques in solving problems related to intelligent Meta-Heuristics. They also developed generalized solutions of the pursuit problem in three dimensional Euclidean space.

Heero et al. (2004) presented a method for mobile robot navigation in environments where obstacles are partially unknown. A single monocular camera for both localization and obstacle detection was integrated into an algorithm developed by Fasola et al. (2005) while Bikdash (2006) used the finite element mesh analysis and COMSOL software for the robot path planning problem. In their article, Becker et al. (2006) used the Obstacle Velocity Approach. Recently, Philippsen et al. (2007) presented a work on sensor-based motion planning in initially unknown dynamic environments while Li et al. (2006) analyzed the vehicle dynamics of redundantly actuated wheeled mobile robots with powered caster wheels.

Another class of dominating navigation technique is the heuristic based technique. This technique is based on inferential and evolutionary theory. Usually they are associated with some degree of inconsistency. Some widely used heuristic techniques include A* algorithm, E* algorithm, genetic algorithm, memetic algorithm and simulated annealing. The A* algorithm was originally proposed by Hart et al. (1968). Literature over the years have shown the effectiveness of integrating this concept with some form of graphical methods such as the configuration space technique which was used by Davis and Camacho (1984).

Geisler and Manikas (2002) improved on the genetic algorithm performance by developing a more efficient genotype structure for a known environment with static obstacles. Motion was constrained to only row-wise navigation. Sedighi et al. (2004) improved on the work of Geisler and Manikas and presented results of a genetic algorithm based path-planning model developed for local

obstacle avoidance (local feasible path) of a mobile robot in a given search space while Simionescu et al. (2006) discussed a new approach to solving constrained nonlinear programming problems using evolutionary computations. In their paper, Valavanis et al. (2006) presented fundamental aspects of a multi layer, hybrid, deliberative and reactive Distributed Field Robot Architecture (DFRA). Also, in the work carried out by Ayari and Chatti (2007), they used the neuro-fuzzy controller in an unknown obstacle environment while Yu et al. (2007) worked on the detection of static and dynamic obstacles. Ordonez et al. (2008) presented a solution named the Virtual Wall Approach (VWA) to the limit cycle problem for robot navigation in a cluttered environment. A new fuzzy method was recently presented by Jaafar and McKenzie (2008) for an action selection method.

We now review some of the potential field models on which the virtual force field concept is hinged. This technique is predicated on the gravitational force field. It is popular because of its simplicity, online adaptive nature and real time prompting. However, it is associated with some basic navigation limitations which results in “cul de sac” also known as the local minima problem. The concept of virtual forces acting on a robot under navigation was introduced long ago by Andrews and Hogan (1983) and later by Khatib (1986). Within this context, and for avoidance of collisions, obstacles are deemed to exert repulsive forces on the robot while the target is presumed to simultaneously apply an attractive force on the robot. In such circumstances the resultant force determines the robot's subsequent direction and speed of motion. Indeed the simplicity and elegance of the technique popularized the use of the potential field method (PFM) in robot navigation over the years.

Notwithstanding its simplicity and elegance, the PFM still suffers from a number of significant problems itemized by Koren and Borenstein (1991). Prominent among these is the local minimum trap occasioned by a concave shaped obstacle which results in continuous re-circulation of the robot within the trap. Whitcomb and Koditschek (1991) have also extended the application of the PFM to automated assembly planning and control. Moravec and Elfes (1985) developed the certainty grid concept which represents an obstacle map of the environment using probabilistic representation of obstacle locations in order to overcome the inaccuracies emanating from sensory data. One limitation with this technique however is that it assumes a static environment.

Moreover, Borenstein and Koren (1990) had developed a new concept for robot navigation. The novelty of their approach, entitled the Virtual Force Field, lies in the integration of two known concepts viz: Certainty Grids for obstacle representation and Potential Fields for navigation. Furthermore, Borenstein and Koren (1991) developed another PFM based algorithm christened the vector field histogram. In their paper, Nourani-Vatani et al. (2006) presented a robotic platform for autonomous coverage tasks. The system architecture adopted for their case integrates laser based localization and mapping using the Atlas Framework with Rapidly-Exploring RandomTrees (RRT) path planning and Virtual Force Field obstacle avoidance. Chengqing et al. (2000) earlier pioneered the virtual obstacle concept as a strategy for local-minimum-recovery in potential-field based navigation. Im and Oh (2000) used an approach christened Extended Virtual Force Field (EVFF) by integrating neural network theory and evolutionary programming. Zou and Zhu (2003) on the other hand were the first set of investigators to work with the virtual local target approach for solving the local minima problem associated with the potential field method.

2. Navigation Environment / Nature of Obstacle

In this paper the navigation environment considered for the robot could be categorized into: completely known environment and partially known environment. A navigation type is said to be in a partially known environment (PKE) when prior to navigation, the robot already has knowledge of some areas within the workspace i.e. areas likely to pose local minima problem. These areas are identified through the process of workspace mapping usually by a human agent.

The nature of an obstacle is described in terms of its configuration which is usually concave or convex shaped. The status of an obstacle is described in terms of its relative position or orientation per unit time as the robot navigates. The status of an obstacle is said to be static when its position or

orientation relative to a known and fixed origin does not change with time. On the other hand, the status of an obstacle is said to be dynamic when either its position or orientation or both relative to a known and fixed origin does change with time. However, in this paper the status of obstacles considered is static. Also, it should be noted that the robot is undergoing a 2-dimensional motion in a 3-dimensional workspace.

3. Method of Solution

A new approach christened the Hybrid Virtual Force Field (HVFF) is being proposed. This concept integrates: the virtual force field method (VFF), the virtual obstacle concept (VOC) and the virtual goal concept (VGC).

3.1. Navigation and Obstacle Avoidance Algorithm

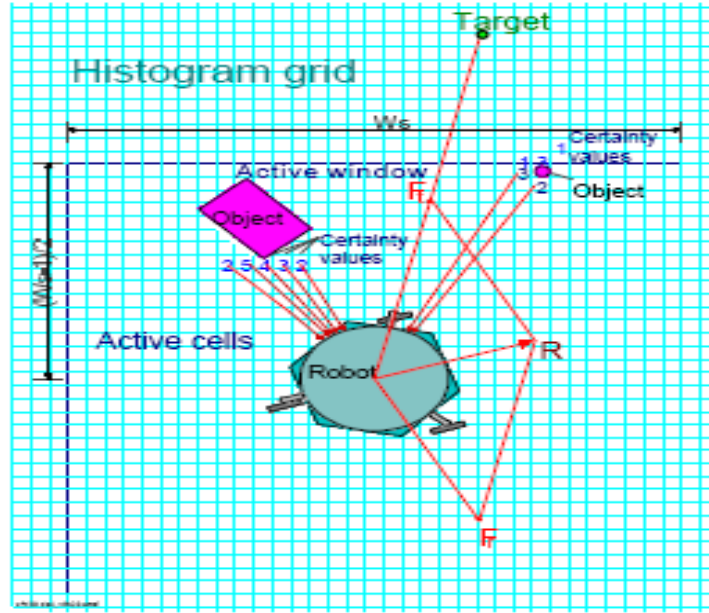
3.1.1 The VFF Scheme

The process of navigation and obstacle avoidance of a robot is carried out by the virtual force field (VFF) algorithm. Like the PFMs, the VFF method hinges on the concept of repulsion and attraction and uses a two dimensional cartesian grid system to represent the workspace or environment within which the robot is confined to navigate. Each cell in the workspace is assigned a certainty value (CV) which represents the confidence of the algorithm as to whether or not an obstacle is located on a given cell within the workspace. Such representation was derived from the certainty grid concept that was originally developed by Moravec and Elfes (1985).

Table 1: Model Notations and Description for the HVFF Scheme

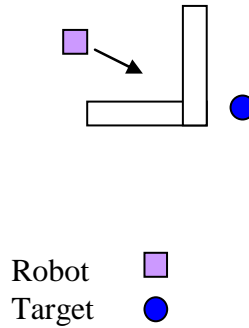
S/N	Symbol	Meaning	S/N	Symbol	Meaning
1	x_r	robot x-coordinate	17	ψ_2	Oritational difference between initial and current orientation when rotation is in the right direction of the differential wheels.
2	y_r	robot x-coordinate	18	∇	Grad operator
3	$c_{i,j}$	dynamic window cells	19	y_{target}	y-coordinate of robot goal state
4	dia_{robot}	dynamic window cells	20	n	arbitrary integer representing the upper limit of a given range
5	$d_{obstacle}$	distance between the robot and an obstacle in the robot's active window	21	θ_{ini}	initial orientation of robot after rotational difference.
6	d_{target}	distance between the robot and the desired goal state	22	θ_{cur}	current orientation of robot.
7	Fcr	repulsive Force variable of sensed obstacle	23	$U_{rep}^x(q_r)$	repulsive potential (x-component)
8	rel_angle	orientation of active sensor relative to robot head	24	$U_{rep}^y(q_r)$	repulsive potential (y-component)
9	x_{target}	x- coordinate of robot goal state	25	$U_{att}^y(q_r)$	attractive potential generated by the goal state (y-component)
10	y_grid	Gridline equivalent on the y-axis of cell currently occupied by the robot within the active window	26	$U_{att}^x(q_r)$	attractive potential generated by the goal state (x-component)
11	x_grid	Gridline equivalent on the x-axis of cell currently occupied by the robot within the active window	27	$U^x(q_r)$	resultant potential in the x axis
12	i	Gridline occupied by robot on x-axis.	28	$U^y(q_r)$	resultant potential in the y axis
13	j	Gridline occupied by robot on y-axis.	29	$U_{att}(q_r)$	total attractive potential from the target point
14	$x_{obstacle}$	obstacle's x-coordinate within robot's active window	30	$U_{rep}(q_r)$	total repulsive potential from obstacles
15	$y_{obstacle}$	obstacle's y-coordinate within robot's active window	31	q_r	Free configuration space describing current position of robot
16	ψ_1	Oritational difference between initial and current orientation when rotation is in the left direction of the differential wheels.	32	q_{target}	Free configuration space describing target point

Thus CV is a dynamic quantity that changes as the robot navigates in the workspace such that it increases with closeness to obstacles and decreases as the robot moves further away from obstacles. Fig. 1 below displays a window illustrating the VFF concept.

Figure 1: The virtual force field concept Koren and Borenstein (1991)

3.1.2 Problems associated with the Virtual Force Field (VFF) Concept

In an attempt to resolve some of the problems associated with PFMs, the VFF method was introduced by Borenstein and Koren (1990). One of the dominating problems associated with the VFF concept is illustrated in Fig. 2 below. The problem illustrates the occurrence of a local minimum trap occasioned by a concave shaped obstacle which results in continuous re-circulation of the robot within the trap.

Figure 2: Local minimum trap by a concave shaped obstacle

3.1.3. The Virtual Force Field (VFF) Algorithm

Assuming q_r describes the configuration space of a robot in a 2D domain such that $q_r = f(x_r, y_r)$ where $x_r \equiv i$ and $y_r \equiv j$ it then follows that $q_r = q_r(i) + q_r(j)$ is the directional representation of q_r .

If the vector operator ∇ defined by $\nabla = i \frac{\partial}{\partial x} + j \frac{\partial}{\partial y}$ is considered, then

$$\nabla(q_r) = \left(i \frac{\partial q_r}{\partial x} + j \frac{\partial q_r}{\partial y} \right) \quad (\text{Model 1})$$

If at every point in time the moving robot R is subjected to both attractive and repulsive potentials resulting from the target position and stationary obstacles respectively, then $U_{att}(q_r)$ and $U_{rep}(q_r)$ would represent the resultant potential acting on the robot where, $U_{att}(q_r)$ is the sum of the total attractive potential from the target point at a given instant while $U_{rep}(q_r)$ the total repulsive

potential from the obstacle(s) in the work space at the same instant such that instantaneous motion of the robot is governed by these resultant potentials.

The gradient of U at q_r is given as:

$$\nabla U(q_r) = \begin{bmatrix} \frac{\partial U}{\partial x} \\ \frac{\partial U}{\partial y} \end{bmatrix} \quad (\text{Model 2})$$

$\nabla U(q_r)$ is a vector that points in the direction of the fastest change of U at configuration q_r . The magnitude of the rate of change is expressed in terms of

$$|\nabla U(q_r)| = \sqrt{\left(\left(\frac{\partial U}{\partial x} \right)^2 + \left(\frac{\partial U}{\partial y} \right)^2 \right)} \quad (\text{Model 3})$$

Moreover each of the potentials can be further decomposed in terms of attraction and repulsion on both x and y axes. As such we have:

$$U_{att}(q_r) = U_{att}^x(q_r) + U_{att}^y(q_r) \quad (\text{Model 4})$$

while

$$U_{rep}(q_r) = U_{rep}^x(q_r) + U_{rep}^y(q_r) \quad (\text{Model 5})$$

Assuming that

$$d_{target} = \sqrt{((x_{target} - x_r)^2 + (y_{target} - y_r)^2)} \quad (\text{Model 6})$$

where,

d_{target} = Euclidean distance between the robot and the target point

Then the configuration space for the robot and the target are respectively defined as

$q_r(x_r, y_r)$ and $q_{target}(x_{target}, y_{target})$

where,

$x_r = 1, \dots, (n-1)$, n and $y_r = 1, \dots, (n-1)$, n

It follows that $d_{target} = \|q_{target} - q_r\|$

Assuming the $q_r(x_r)$ direction is considered then

$$d_{target} = \|q_{target} - q_r\| = \left(\sum_{r=1}^n (x_{target} - x_r)^2 \right)^{1/2} \quad (\text{Model 8})$$

$$\nabla d_{target}(q_r(x_r)) = \nabla \left(\sum_{r=1}^n (x_{target} - x_r)^2 \right)^{1/2} \quad (\text{Model 9})$$

$$= 1/2 \left(\sum_{r=1}^n (x_{target} - x_r)^2 \right)^{-1/2} \nabla \left(\sum_{r=1}^n (x_{target} - x_r)^2 \right) \quad (\text{Model 10})$$

$$= 1/2 \left(\sum_{r=1}^n (x_{target} - x_r)^2 \right)^{-1/2} \left(2(x_{target} - x_1) \dots 2(x_{target} - x_n) \right) \quad (\text{Model 11})$$

$$= \frac{\left(x_{target} \right) - \left(x_1, \dots, x_n \right)}{\sum_{r=1}^n \left(x_{target} - x_r \right)^2}^{1/2} = \frac{\left(x_{target} - x_r \right)}{\sum_{r=1}^n \left(x_{target} - x_r \right)^2}^{1/2} \quad (\text{Model 12})$$

Assuming the $q_r(y_r)$ direction is considered, then

$$d_{target} = \|q_{target} - q_r\| = \left(\sum_{r=1}^n (y_{target} - y_r)^2 \right)^{1/2} \quad (\text{Model 13})$$

$$\nabla d_{target}(q_r(y_r)) = \nabla \left(\sum_{r=1}^n (y_{target} - y_r)^2 \right)^{1/2} \quad (\text{Model 14})$$

$$= 1/2 \left(\sum_{r=1}^n (y_{target} - y_r)^2 \right)^{-1/2} \nabla \left(\sum_{r=1}^n (y_{target} - y_r)^2 \right) \quad (\text{Model 15})$$

$$= 1/2 \left(\sum_{r=1}^n (y_{target} - y_r)^2 \right)^{-1/2} \left(2(y_{target} - y_1) \dots 2(y_{target} - y_n) \right) \quad (\text{Model 16})$$

$$= \frac{\left(y_{target} \right) - \left(y_1, \dots, y_n \right)}{\sum_{r=1}^n \left(y_{target} - y_r \right)^2} = \frac{\left(y_{target} - y_r \right)}{\sum_{r=1}^n \left(y_{target} - y_r \right)^2} \quad (\text{Model 17})$$

From (12) and (17) above, it could be deduced that

$$\nabla d_{target}(q_r(x_r, y_r)) = \frac{\left(q_{target} - q_r \right)}{\sum_{r=1}^n \left(x_{target} - x_r \right)^2 + \sum_{r=1}^n \left(y_{target} - y_r \right)^2} \quad (\text{Model 18})$$

$$= \frac{q_{target} - q_r}{\|q_{target} - q_r\|} \quad (\text{Model 19})$$

Hence for our problem,

$$U_{att}^x(q_r) = \frac{(x_{target} - x_r)}{d_{target}} \quad (\text{Model 20})$$

and

$$U_{att}^y(q_r) = \frac{(y_{target} - y_r)}{d_{target}} \quad (\text{Model 21})$$

Within the same context, we can, by following the work of Koren and Borenstein represent the repulsive potentials as:

$$U_{rep}^x(q_r) = U_{rep}^x(q_r) + \frac{Fcr}{3} * \frac{c_{i,j}}{d_{obstacle}^2} * \frac{(x_{grid} - x_{obstacle})}{d_{obstacle}} \quad (\text{Model 22})$$

$$U_{rep}^y(q_r) = U_{rep}^y(q_r) + \frac{Fcr}{3} * \frac{c_{i,j}}{d_{obstacle}^2} * \frac{(y_{grid} - y_{obstacle})}{d_{obstacle}} \quad (\text{Model 23})$$

where,

The repulsive variable Fcr , contained in the afore-mentioned relations, is in our case given by the expression

$$Fcr = \frac{3 * rel_angle}{d_{obstacle}} \quad (\text{Model 24})$$

Fcr is usually activated when an obstacle is present and sensed in the robot's active window $c_{i,j}$. In the absence of an obstacle, Fcr gives an output of zero.

The variables $U_{rep}^x(q_r)$ and $U_{rep}^y(q_r)$ both have initial values of zero. This is in line with the assumption that the robot ideally takes off from a state of rest where it is absolutely ignorant of its environment. As the vehicle moves towards the goal, it creates for itself a dynamic window. The dynamic window is made up of a set of cells around the robot. The window is usually of a fixed dimension about the robot. The repulsive algorithm triggers immediately an obstacle is sensed in the dynamic window hence causing the robot to repel from the obstacle.

In particular,

$$c_{i,j} = \begin{cases} 0 & \text{If no obstacle is present on any cell in the robot's active window} \\ 1 & \text{If obstacle is present on any cell in the robot's active window} \end{cases}$$

if $c_{i,j} = 1$ and $d_{target} > 0$ then

$$d_{obstacle} = \sqrt{((x_r - x_{obstacle})^2 + (y_r - y_{obstacle})^2)} \quad (\text{Model 25})$$

measuring the distance between the obstacle and the robot.

Here, $x_{obstacle}$ and $y_{obstacle}$ are the coordinates of the obstacle and as the robot navigates towards the target point, it is routinely updated by the control unit as to whether or not an obstacle is present in any cell within its active window.

Thus in general

$$U^x(q_r) = U_{att}^x(q_r) + U_{rep}^x(q_r) \quad (\text{Model 26})$$

and

$$U^y(q_r) = U_{att}^y(q_r) + U_{rep}^y(q_r) \quad (\text{Model 27})$$

Equations (26) and (27) denote the resultant potential in the x and y axes respectively. The actual navigation path of the robot is determined by these two variables as they jointly determine the robot's new position as it navigates from one state space towards the target point.

4. Escape Algorithm from Local Minima Traps

By way of implementation of our new algorithm, we propose a combination of two emerging concepts for solving the inherent problems associated with the traditional VFF technique. Although it has earlier been established in the literature that either of these concepts could be used independently, the dual use of both concepts as an integral part of the solution process is in fact novel. To be sure, it was Chengqing et al. (2000) who first integrated the virtual obstacle technique with the potential field method to maneuver cylindrical mobile robots in unknown environments and christened the method as the virtual obstacle method. In a similar fashion, Zou and Zhu (2003) later introduced the concept of integrating a virtual goal with the potential field principle and they called this approach the Virtual Local Target (VLT) method.

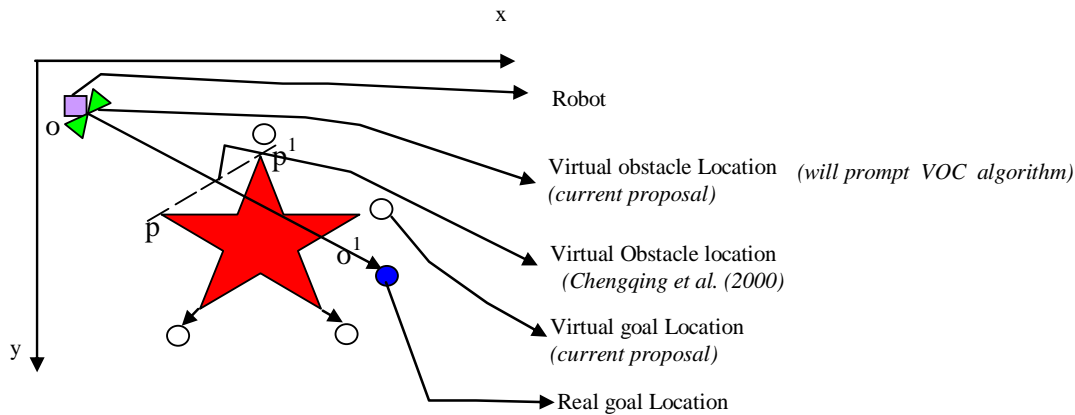
4.1 Virtual Object Placement

The need for virtual objects i.e. (virtual obstacle and virtual goal) and their relative placement scheme is a function of the nature and status of the obstacle along the robot's line of sight. A detailed description of the virtual object placement scheme is given in sections 4.1.1 through 4.1.4.2. It would be seen that for **static obstacles with either convex or concave formation**, once it is established that there is need for virtual objects to aid robot navigation then the virtual obstacle is first introduced at the

initial position of the robot prior to navigation to prevent the robot from travelling along its initial trajectory which is in the direction of a local minima trap. This is immediately followed by the introduction of virtual goals. The placement of virtual goal is always adjacent to the obstacle closest to the goal. The placement of virtual goals is based on ray theory in a backtracking order i.e. backward chaining from the position of the real goal to the robot. More information about the placement scheme is described in section 4.1.1 through 4.1.4.2.

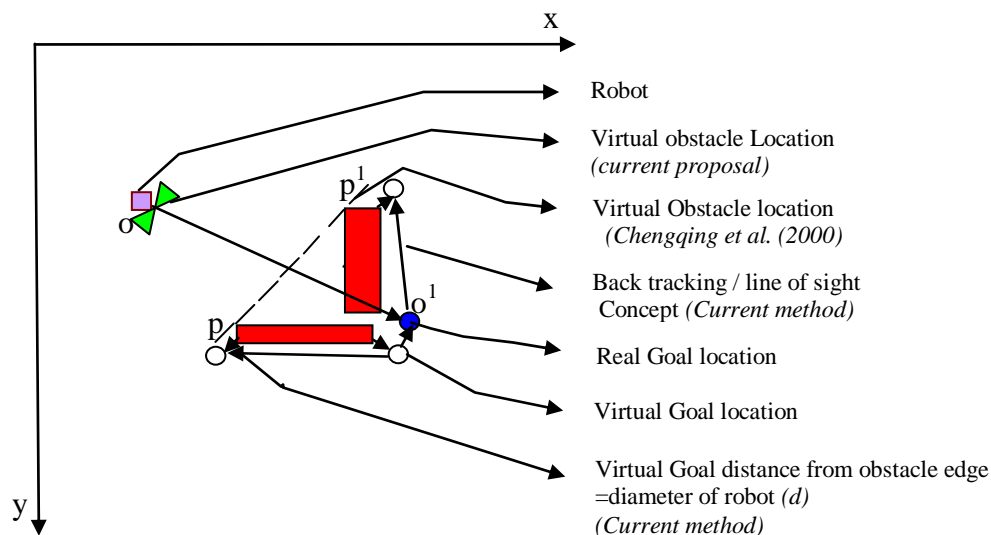
For example figure 3a illustrates the use of VOC and VGC concepts in an environment containing a star shaped obstacle while in Figure 3b the goal is located diametrically

Figure 3a: Case I sample workspace demonstrating the virtual obstacle concept and the virtual goal concept



opposite the robot but with the line of sight grazing two touching obstacles leaving no clearance through which the robot could maneuver.

Figure 3b: Case II sample workspace demonstrating the virtual obstacle concept and the virtual goal concept



For a start we note that in Figures 3a and 3b the HVFF scheme locates a virtual obstacle right next to the robot. This is merely to trigger the VOC scheme described in the subsequent part of this section. This allows the robot to respond faster than the algorithm developed by Chengqing et al. (2000) who place their virtual obstacle farther down the line. At the same time the complementary effect of using back tracking along the line of sight from a hidden goal state around the perimeter of the obstacle facilitates optimal location of virtual goals.

4.1.1 Virtual Obstacle Concept (VOC)

The virtual obstacle concept in our context is basically a proactive algorithm that ensures that the robot is not attracted into a corner region of obstacles that has no outlet. It is hinged on the concept of completely blocking off such passageways that could lead the robot to a local minimum trap. The approach we are proposing for the representation of our VOC is the concept of intersecting vertices. This involves the introduction of a new line that closes the edges of the obstacles that frame the local minimum trap. The intersection of the line of sight of the target from the robot with this new line represents the farthest location the virtual obstacle can be placed from the robot; otherwise it is located right next to the robot along the line of sight of the target from the robot. The VOC representation for some sample configurations is as shown in Figs. 3a and 3b.

The location of the target point relative to the robot's initial position at the start of navigation determines whether or not the virtual obstacle algorithm should be implemented or not. Our concept of virtual obstacle is generalized in principle, and is in no way limited by the shape of the obstacles that define the concave environment. We have tried to illustrate its level of completeness and generalization by creating different scenarios of concave shaped problems using different types of obstacle shapes. In comparison with the earlier work of Chengqing et al (2000) our VOC algorithm is more efficient.

The reader is also invited to note that in the case of Chengqing et al (2000), entrapment in local minima was avoided by replacing the two converging obstacles with a virtual third side of the triangle that blocks the trap. In our case as developed in Ayomoh (2008) we do subscribe to the concept of preventing access into the corner region but instead of introducing a virtual third side we are proposing the intersecting vertices concept as described below:

4.1.2 Itemization of the procedural flow of the VOC

4.1.2.1 Need For a Virtual Obstacle (VO)

The function $VOC=f(I_{LS})$ is an index used to determine whether the VO is needed.

I_{LS} is an index for describing the status of the line of sight of robot from the goal position in respect of interception by obstacles that could cause a local minima problem as earlier stated in section 2.

In particular,

$$I_{LS} = \begin{cases} 0 & \text{Line of sight of robot is not intercepted by obstacle likely to cause a local minima problem} \\ 1 & \text{Line of sight of robot is intercepted by obstacle likely to cause a local minima problem} \end{cases}$$

if $I_{LS} < 0$ then VOC is prompted.

4.1.2.2 Positioning a Virtual Obstacle (VO)

Step 1: Locate the vertices at the corners of the obstacles that form the entry into suspected minima traps e.g. Points p and p¹ in Figures 3a and 3b are typical examples.

Step 2: Project a line $\overline{pp^1}$ as illustrated in Figures 3a and 3b.

Step 3: Draw a straight line $\overline{oo^1}$ from the robot's position to the desired target point. This shows the natural trajectory of the robot to the target point assuming no obstacle was present see figures 3a and 3b.

Step 4: If $\overline{oo^1}$ intersects $\overline{pp^1}$ at any point along $\overline{pp^1}$ it implies that the robot's natural trajectory is in the direction of a trap.

Steps 1-4 are verification steps as to whether or not the VOC should be initiated. If it is verified that the robot's trajectory is in the direction of a trap then Step 5 is initiated.

Step 5: This introduces a virtual obstacle at the current position of the robot. This is done by using some form of production rules as control command to impede robot motion. This prevents the robot from moving in the direction of local minima trap and hence facilitates the eventual minimization of the overall navigation time. Immediately following this step, proceed to implement the VGC as discussed below by setting $f_{voc} = 1$.

4.1.3 Virtual Goal Concept (VGC)

Assuming an object at an arbitrary position say $p_{i,j}$ is viewed from a translated position $p_{i+1,j+1}$ then the visibility of this object at its initial state $p_{i,j}$ from its current state $p_{i+1,j+1}$ forms a significant question posed by the VGC. The VGC in our work hinges on the concept of relative visibility. It operates on the principle of backtracking from an initially known position to a newly navigated position. The primary objective of the VGC in this context is to lead the robot from the point where the virtual obstacle was activated to the final target point through the aid of one or more temporary goals referred to as virtual goals.

The number of virtual goals and their relative position is a function of the geometry of the obstacles intercepting the visibility of the robot from the goal state. The degree of visibility of an object from its initial position $p_{i,j}$ as earlier described, relative to its new position $p_{i+n,j+n}$ determines whether the new position should be considered a virtual position of the object. If the object's visibility is obscured at point $p_{i+n,j+n}$ then the preceding point $p_{i+n-1,j+n-1}$ is considered for the virtual position of the object. For this purpose the intermediate points between $p_{i,j}$ and $p_{i+n,j+n}$ are selected along and close to the vertices of the intervening obstacle boundary. Figs. 3a and 3b above are also case illustrations of the VGC. The required steps for the VGC are itemized as follows:

4.1.4 Itemization of the procedural flow of the VGC

4.1.4.1 Need For a Virtual Goal (VG)

Step 1: Is a VG needed?

If either the function f_{voc} or $T_{nso} = 1$ is satisfied then a VG is needed.

Here

$$f_{voc} = \begin{cases} 0 & \text{virtual obstacle module is not activated} \\ 1 & \text{virtual obstacle module is activated} \end{cases}$$

The index T_{nso} has also been introduced to describe the situation that occurs when the robot and goal are at opposite ends of the space in between two narrowly spaced obstacles. Here although the orifice is wide enough the robot might still find it difficult to maneuver its way unaided through the clearance between the two obstacles to the goal position.

When such a condition exists we set $T_{nso} = 1$; otherwise $T_{nso} = 0$ which is equivalent to the situation that would trigger the introduction of a virtual obstacle sealing off entry into the region.

4.1.4.2 Locating a Virtual Goal (VG)

Step 1: Locate the goal position or the target point. This is as shown in figures 3a and 3b at point o^1 .

Step 2: Locate the robot's position. This could also be seen in figures 3a and 3b at point o .

Step 3: From the goal position, backtrack along a projected line of sight towards the robot position skirting the perimeter of the intervening obstacle. Ensure that the line of sight is not intercepted by any obstacle(s). See figures 3a and 3b.

NOTE: The line of sight is drawn using the concept of ray theory. It is drawn to circumnavigate the obstacle(s) whose boundaries frame the local minima problem.

Step 4: locate and position virtual goals at appropriate points along the line of sight as you back track from the real goal to the robot position. The exact location of each virtual goal is a function of two variables: (V_{rg} , d).

where,

V_{rg} either describes the relative visibility of the most recent virtual goal from the immediate preceding virtual goal or the relative visibility of virtual goal from the real goal (in the case of the first virtual goal).

d =distance between VG and the edge of the obstacle obstructing robot's line of sight from the real goal.

These concepts are also illustrated in figures 3a and 3b.

Furthermore,

$$I_{LS} = \begin{cases} 0 & \text{when relative visibility of virtual goal from immediate} \\ & \text{preceding virtual goal does not exist} \\ 1 & \text{when relative visibility of virtual goal from immediate} \\ & \text{preceding virtual goal does exist} \end{cases}$$

In fact,

$$d \geq 0.5(dia_{robot})$$

where,

$$dia_{robot} = \text{diameter of robot}$$

Another important parameter which is necessary to terminate the process of introducing VG is V_{mr} which represents the relative visibility of the most recent virtual goal from the robot.

In fact

$$V_{mr} = \begin{cases} 0 & \text{Virtual goal is visible from robot position} \\ 1 & \text{Virtual goal is visible from robot position} \end{cases}$$

If $V_{mr}=0$ then no more VG is needed; however if $V_{mr}=1$ then another VG is needed. This cycle is repeat until $V_{mr} = 0$.

5.0. The HVFF Algorithmic Procedure

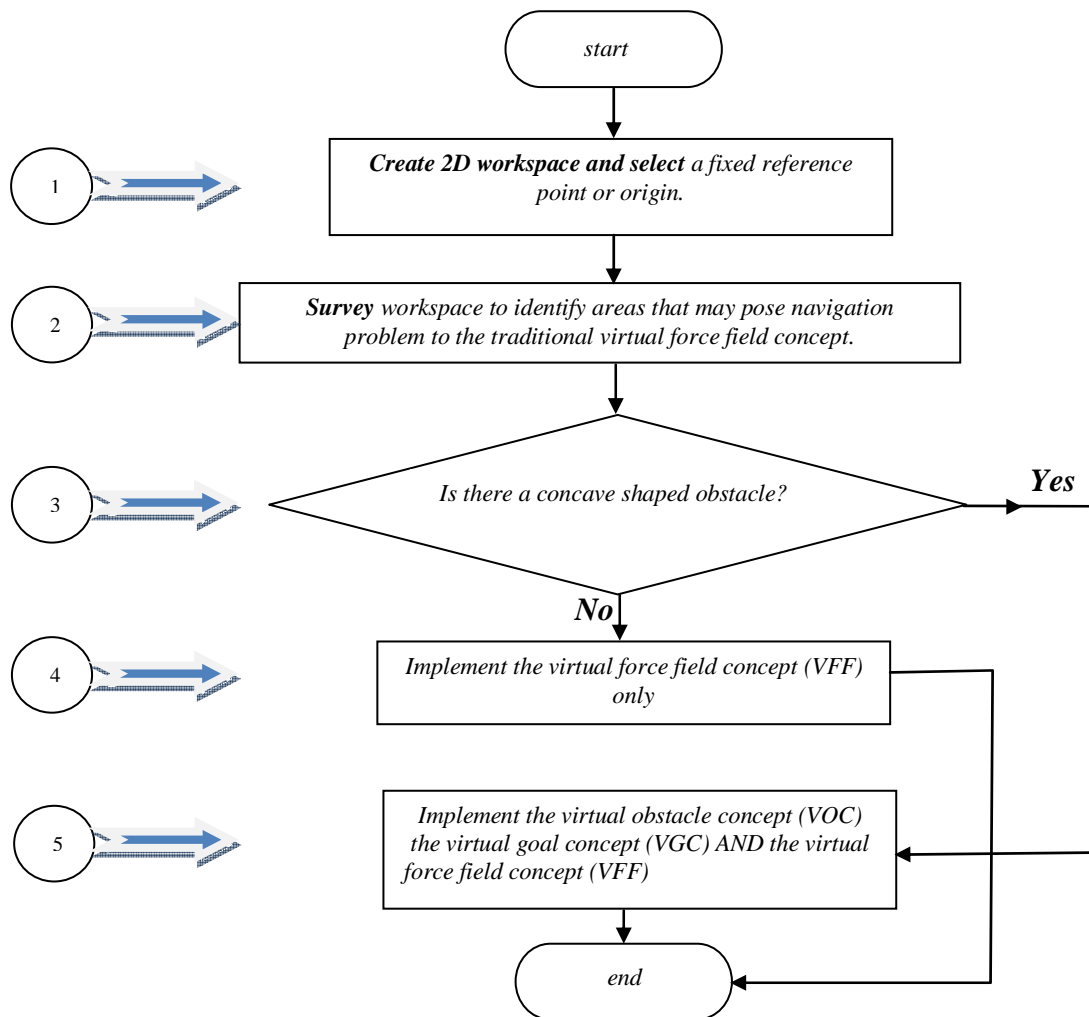
We next itemize how the VFF concept combines with the VOC and VGC to form the proposed hybrid concept by giving a step by step outline of the operational mode of the overall algorithm as illustrated in Fig. 4.

Step I: This module allows us to create a 2D workspace of dimension X by Y and select a fixed reference point for the workspace. The emphasis in this step is the availability of a workspace or navigation environment. Since the current problem is 2D, we have the x and y axes for the purpose of position tracking. A navigation environment may be regular or irregularly shaped; what is important is the availability and delineation of the boundaries of the workspace. This task is next followed by the identification of a point within the workspace which serves as a reference point for the robot navigation. The reference point helps the robot to know its own relative position, as well as those of the goal state or target point and the workspace obstacles.

Step II: This module carries out a mapping of the workspace for identification of areas that may pose navigation problems to the traditional Virtual Force Field concept. Such areas are

characterized by navigation impediments leading to local minima problems. Once the areas prone to local minima occurrence are identified, their relative position vectors from a

Figure 4: A flow chart diagram for the HVFF algorithm



specified reference point is obtained. Such information is encoded in the robot's knowledge base prior to commencement of the navigation exercise. The relevance of this step is hinged on the fact that our proposed algorithm is for a partially known environment where some a priori knowledge of the navigation environment is needed.

Step III: This module basically serves as a sequel to step II above in that the outcome of the survey exercise in step II determines the activation or deactivation of certain algorithms in the control sub-unit. If the environment is such that a local minima trap is not likely to occur then we proceed to step IV. However, if navigation traps are suspected then we skip **step IV** and proceed to either **step V**. Hence, the essential purpose of this module is to link step II with either step IV or step V as the case may be.

Step IV: This module, when triggered implements the traditional Virtual Force Field (VFF) algorithm. If the outcome of the workspace survey in step II is such that local minima trap is not likely to occur, then the flow is from **step III** to **step IV**. At this point the environment of navigation is referred to as a completely unknown environment since the robot's navigation control is completely reactive and no a priori knowledge about the

workspace obstacles is needed. If the outcome of the workspace survey is such that local minima trap is likely to occur then the algorithmic flow is from **step III to step V**.

Step V: This module is prompted when local minima problem results from concave shaped obstacles as sketched in Figs. 3a and 3b. Here, the virtual obstacle concept (VOC), the virtual goal concept (VGC) and the virtual force field (VFF) concepts are co-implemented. At this point the environment of navigation is classified as a partially known environment since the robot's navigation to the target point is dependent on the a priori knowledge of some obstacles in the workspace. The implementation procedure for the VOC and VGC are described in sections 4.1.1 through 4.1.4.2 respectively.

7. Discussion of Simulation Results

As a means of validating our algorithm, a comparative study was carried on the results obtained from simulations using HVFF and those of some earlier workers. More specifically, the simulations were conducted on workspaces developed by Sedighi et al. whose solution technique was based on genetic algorithm. This formed the basis for our algorithmic comparison as shown in table 2 below. We implemented our simulation using MobotSim software, a customized mobile robot simulator, on an Intel Pentium®4, 2.4GHz, 224MB of RAM.

The workspaces were reproduced in terms of the shape, size and position of the obstacles as presented in Sedighi et al. However, it was not always possible to reproduce these selected workspaces to scale in our work. This is mostly due to non-availability of detailed information such as the obstacle dimensions and in some cases, the workspace dimensions. Nevertheless, our priority is to be able to reproduce such workspace environment to a justifiable degree to carry out our validation. These are as shown in Figs. 5a through 11c.

It is clear from Table 2 that for search spaces SPSet 06, 07 and 08 the solutions of both Geisler and Harmani had poor performance in relation to the genetic algorithm of Sedighi et al. for which success rates (i.e. ability to reach the desired goal) of 100 and 73.3% were recorded respectively.

Table 2: Success Rate for typical Sample Workspaces Sedighi et. al. [24]

Search Space	Success Rate (%)			
Geisler's GA	Hermami's	GA	Sedighi'sGA	PROPOSED(HVFF)
SPSet01	100	93.3	93.3	100
SPSet02	0.00	86.7	100	100
SPSet04	53	80	100	100
SPSet05	0.00	100	100	100
SPSet06	0.00	20	100	100
SPSet07	0.00	0.00	86.7	100
SPSet08	0.00	0.00	73.3	100

Also, it could be seen from sample workspaces SPSet01 and SPSet08 shown in Table 2 that these two workspaces were the most difficult cases encountered by Sedighi et al. While their GA gave a success rate of 93.3% for SPSet01, 100% in workspaces SPSet02- SPSet06, the success rate for SPSet07 and SPSet08 were 86.7% and 73.3% respectively. For all the workspaces, Figs. 5a,6a,7a,8a,9a,10a and 11a represent the initial positions of the robot relative to the goal state before navigation while Figs. 5c,6c,7c,8c,9c,10c and 11c are navigation results obtained from the current research. On comparison, the trajectories in Figs. 5b and 5c; 6b and 6c; 7b and 7c; 8b and 8c; 9b and 9c; 10b and 10c; 11b and 11c clearly show that our model is efficient. It is also to be noted that where the solution is hinged on evolutionary algorithm, the evolving set of paths for different trials may not be consistent. Our concept does not vary the path over trials as long as the conditions of the workspace remain constant.

In this paper we have demonstrated that the HVFF scheme is quite versatile and robust. It has the merits of high efficiency and effectiveness in terms of goal state attainment. Irrespective of the degree to which a workspace is clustered with obstacles of different shapes and sizes, a robot controlled by this approach can always reach the goal state without colliding with obstacles provided a feasible path exists. This investigation has however focused attention on partially known environments with static obstacles. The HVFF technique can in fact be extended to cover different cases of changing environment which can also be referred to as the dynamic implementation of HVFF.

We find that our approach of representation for both the VOC and VGC can be generalized for all shapes of obstacles and is in no way limited for example, by the shape of the obstacles creating the concave environment. We have also tried to establish some level of completeness and generalization by comparing its performance with those of earlier workers. The indications are that the HVFF is clearly more efficient than the earlier algorithms it was compared with.

```
"C:\WINNT\Profiles\kamran\Desktop\k\Debug\myThesis1.exe"
```

```
XXXXXXXXX  
      X  
XXXXXX  X  
      XXXX  
X       XXXX  
XXXXX  
XXXXX   XXXX  
      X  
      X  
      X
```

```
Generation:41                               Computational time: 0.13sec
```

```
PathLocation: 0-0-6-4-4-5-7-6-6-9-  
PathDirection: 0-1-0-1-0-1-1-1-1-1-  
PathOrientation: 1-  
PathSwitch: 1-6-  
Press any key to continue.
```


Figure 5c: Solution for Sample Space01 (*Current Scheme*)

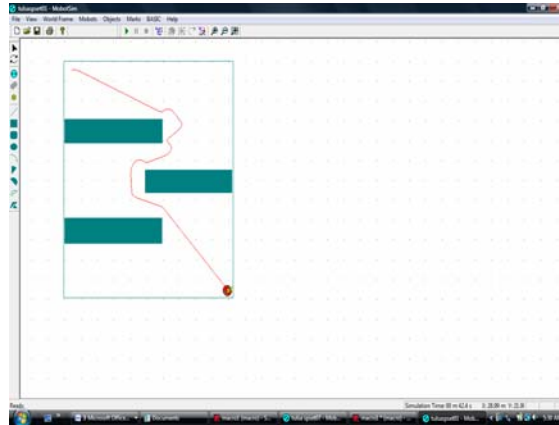


Figure 6a: Problem Sample Space02 (*Sedighi et al. 2004*)

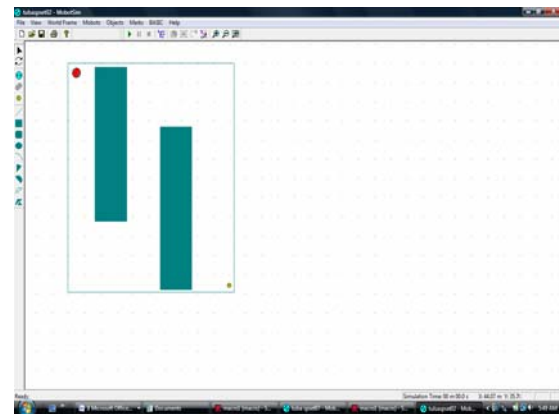


Figure 6b: Solution attempt for Sample Space02 (*Sedighi et al. 2004*)

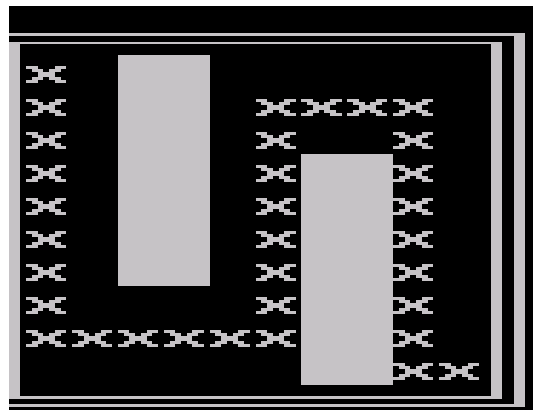


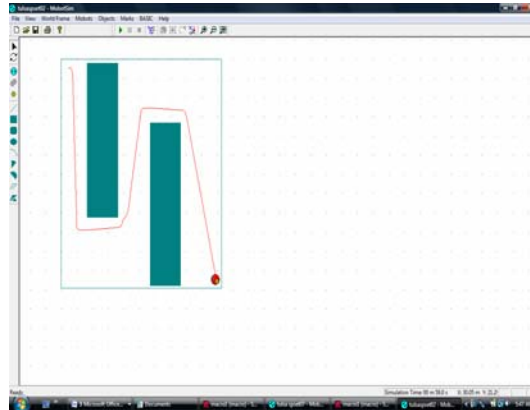
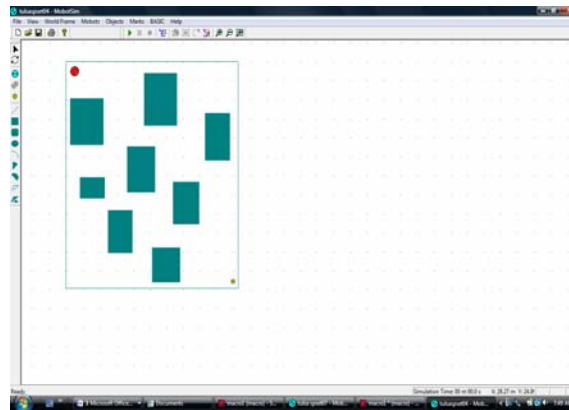
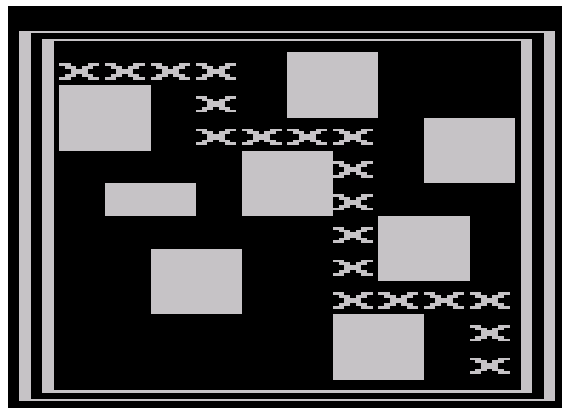
Figure 6c: Solution for Sample Space02 (*Current Scheme*)**Figure 7a:** Problem Sample Space04 (*Sedighi et al. 2004*)**Figure 7b:** Solution attempt for Sample Space04 (*Sedighi et al. 2004*)

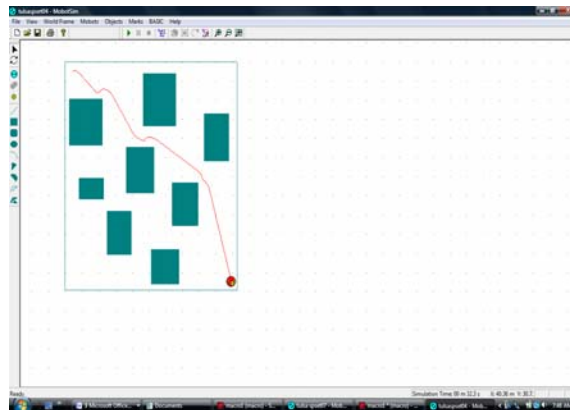
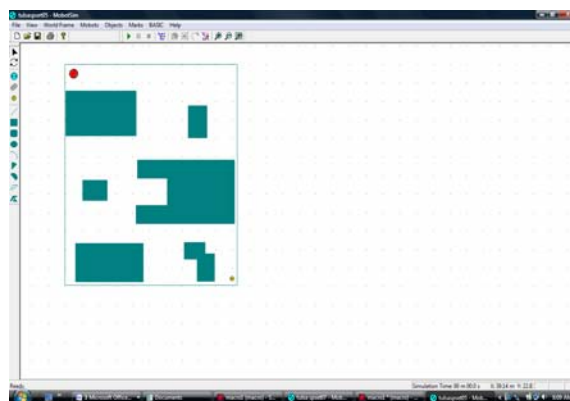
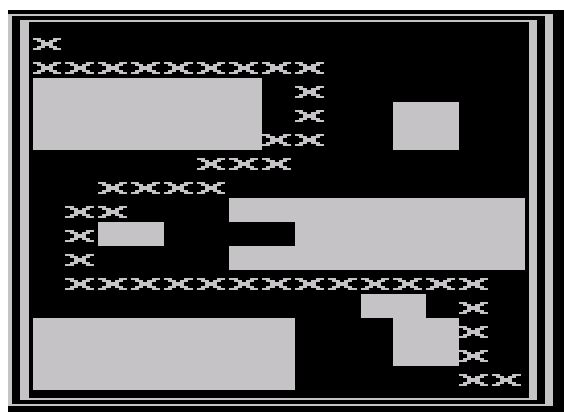
Figure 7c: Solution for Sample Space04 (*Current Scheme*)**Figure 8a:** Problem Sample Space05 (*Sedighi et al. 2004*)**Figure 8b:** Solution attempt for Sample (*Sedighi et al. 2004*)

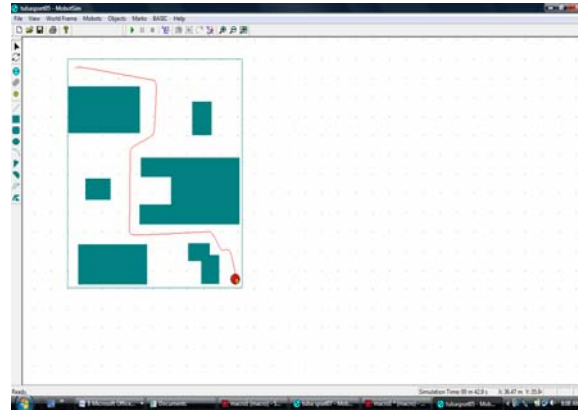
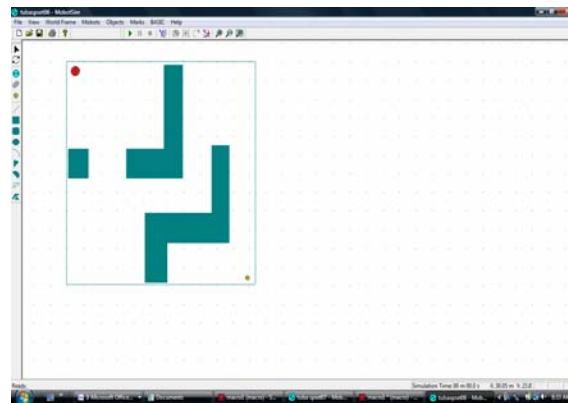
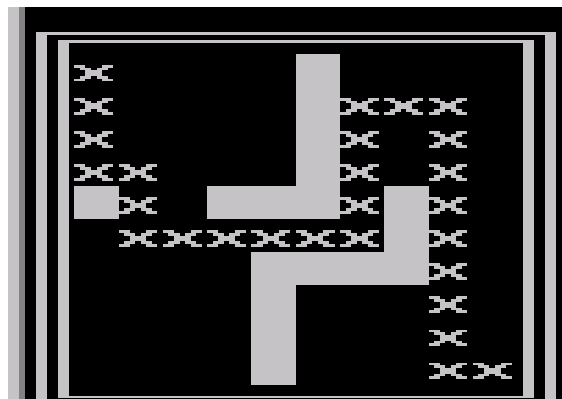
Figure 8c: Solution for Sample Space 05 (*Current Scheme*)**Figure 9a:** Problem Sample Space 06 (*Sedighi et al. 2004*)**Figure 9b:** Solution attempt for Sample Space 06 (*Sedighi et al. 2004*)

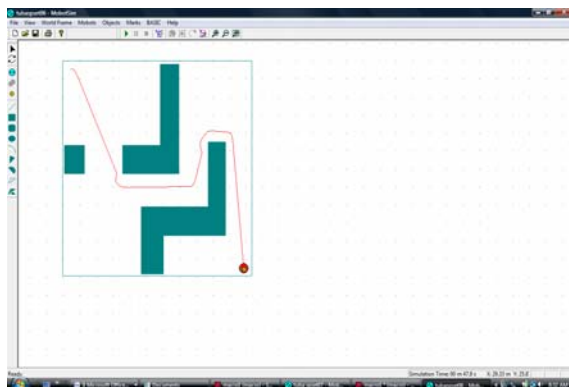
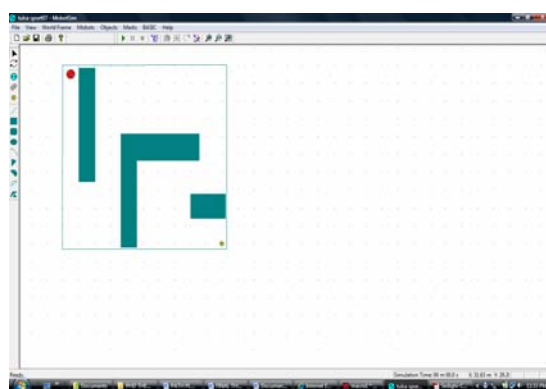
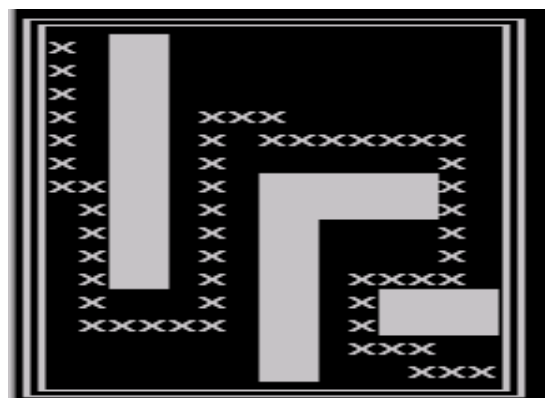
Figure 9c: Solution for Sample Space 06 (*Current Scheme*)**Figure 10a:** Problem Sample Space 07 (*Sedighi et al. 2004*)**Figure 10b:** Solution attempt for Sample Space 07 (*Sedighi et al. 2004*)

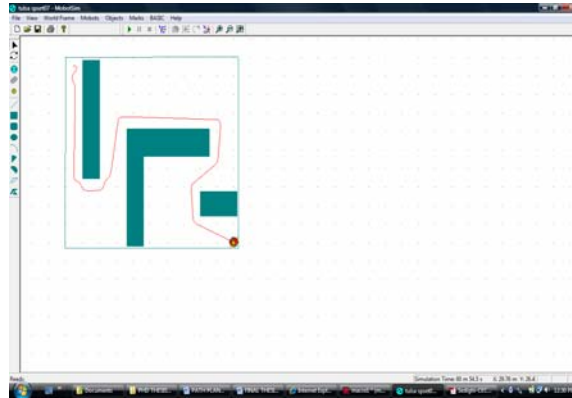
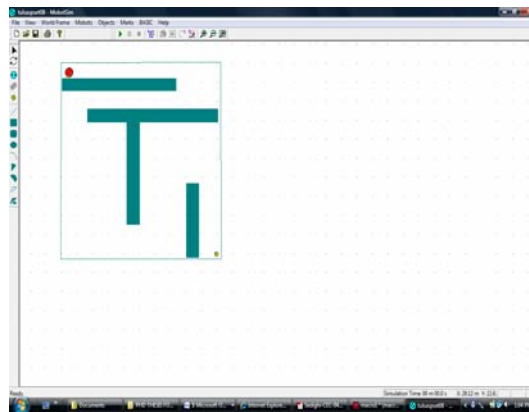
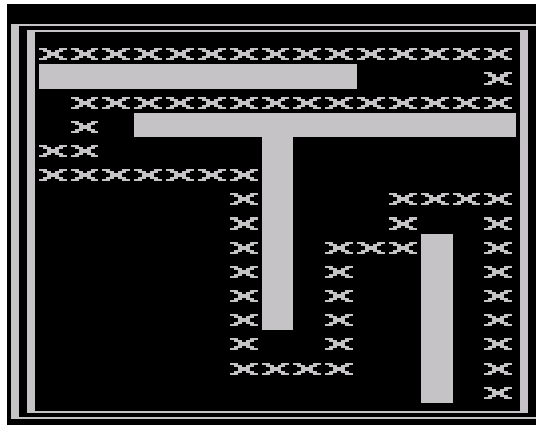
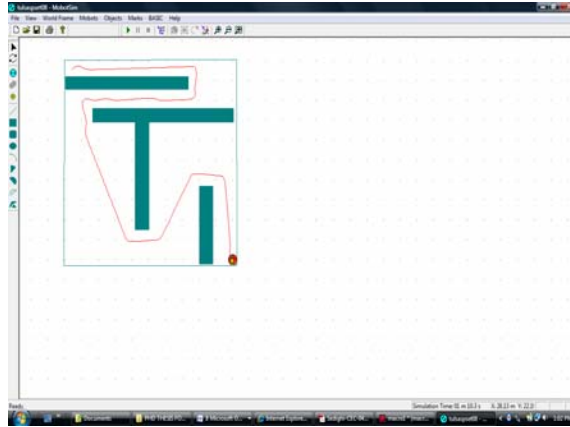
Figure 10c: Solution for Sample Space 07(*Current Scheme*)**Figure 11a:** Problem Sample Space 08 (*Sedighi et al. 2004*)**Figure 11b:** Solution attempt for Sample (*Sedighi et al. 2004*)

Figure 11c: Solution for Sample Space 08 Space 08 (*Current scheme*)

References

- [1] Andrews, J.R. and Hogan, N. (1983) "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator, Control of Manufacturing Processes and Robotic Systems", Eds. Hardt, D.E. and Book, W., ASME, Boston. pp. 243-251.
- [2] Asaolu, O.S. (2001) "An Intelligent Path Planner for Autonomous Mobile Robots", PhD Thesis, Department of Systems Engineering, University of Lagos, Nigeria.
- [3] Ayari, I. and Chatti, A. (2007) "Reactive Control Using Behavior Modelling of a Mobile Robot", International Journal of Computers, Communications & Control Vol. II, No. 3, pp. 217-228.
- [4] Ayomoh, M.K.O. (2008) "A Hybrid Virtual Force Field Model for Autonomous Mobile Robot Navigation", Unpublished Ph.D Thesis, Department of Systems Engineering, University of Lagos.
- [5] Becker, M., Dantas, C. M. and Macedo, W. P. (2006) "Obstacle Avoidance Procedure for Mobile Robots", ABCM Symposium series in Mechatronics vol. 2, pp. 250-257, copyright by ABCM.
- [6] Bikdash, M., Karagol, S. and Charifa, M. (2006) "Mesh Analysis with Applications in Reduced-Order Modeling and Collision Avoidance" Proceedings of the COMSOL Users Conference Boston, pp. 1-7.
- [7] Borenstein, J. and Koren, Y. (1990) "Real-time obstacle avoidance for fast mobile robots in cluttered environments", Proceedings of IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, pp. 572-577.
- [8] Borenstein, J. and Koren, Y. (1991) "The Vector Field Histogram - Fast Obstacle-Avoidance for Mobile Robots", IEEE Journal of Robotics and Automation, Vol.7, No. 3, pp: 278-288.
- [9] Chengqing, L., Ang Jr, M. H., Krishnan, H. and Yong, L.S. (2000) "Virtual Obstacle Concept for Local minimum recovery in Potentialfield Based Navigation", Proc. of IEEE. International Conf. on Robotics and Automation, San Francisco, pp. 983-988.
- [10] Davis, R.H. and Camacho, M. (1984) "The Application of Logic Programming to the Generation of Paths for Robots" Robotica vol.2, pp 93-103.
- [11] Fasola, J., Rybski, P.E. and Veloso, M.M. (2005) "Fast Goal Navigation with Obstacle Avoidance Using a Dynamic Local Visual Model", VII SBAI / II IEEE LARS. São Luís, setembro de, pp.1 6.
- [12] Gayle, R., Sud, A., Lin, M.C and Manocha, D. (2007) "Reactive deformation roadmaps: Motion planning of multiple robots in dynamic environments", In Proc IEEE International Conference on Intelligent Robots and Systems, San Diego, CA, pp. 3777-3783.

- [13] Geisler, T. and Manikas, T. (2002) "Autonomous Robot Navigation System Using a Novel Value Encoded Genetic Algorithm", Proceeding of IEEE Midwest Symposium on Circuits and Systems, Tulsa, OK, pp. 45-48.
- [14] Hart, P.E., Nilsson, N.J. and Raphael, B. (1968) "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Trans. of Systems Science and Cybernetics, 4 (2), pp. 100-107.
- [15] Heero, K., Willemson, J., Aabloo, A. and Kruusmaa, M. (2004) "Robots Find a Better Way: A Learning Method for Mobile Robot Navigation in Partially Unknown Environments", In Proceedings of the 8th Conference on Intelligent Autonomous Systems, (IAS-8), Amsterdam pp.559-566
- [16] Ibidapo-Obe, O. and Asaolu, O.S. (2006) "Optimization Problems in Applied Sciences: From Classical Through Stochastic To Intelligent MetaHeuristic Approaches", 22:1-18; Contributed chapter in Handbook of Industrial and Systems Engineering, Edited by Badiru, A.B, CRC Press, Taylor and Francis Group, New York.
- [17] Ibidapo-Obe, O., Asaolu, O.S. and Badiru, A.B. (1999) "Generalized Solutions of the Pursuit problem in Three Dimensional Euclidean Space", Applied Mathematics and Computation, Elsevier Science Inc. 6580, pp. 1-11.
- [18] Ibidapo-Obe, O., Asaolu, O.S. and Badiru, A.B. (2002) "A New Method for the Numerical Solution of Simultaneous Non-Linear Equations", Applied Mathematics and Computation, Elsevier Science Inc. 125(1), pp. 133-140.
- [19] Im, K.Y. and Oh, S.Y. (2000) "An Extended Virtual Force Field Based Behavioral Fusion with Neural Networks and Evolutionary Programming for Mobile Robot Navigation", Evolutionary Computation, IEEE Congress Volume 2, pp. 1238-1244.
- [20] Irich, I., Borenstein, J. (2000) "VFH*: local obstacle avoidance with look-ahead verification", Proceedings of IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, Vol. 3, pp: 2505 - 2511.
- [21] Jaafar, J. and McKenzie, E. (2008) A Fuzzy Action Selection Method for Virtual Agent Navigation in Unknown Virtual Environments, Journal of Uncertain Systems Vol.2, No.2, pp.144-154, 2008
- [22] Khatib, O. (1986) "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", Int. J. Robotics Res. 5(1). pp 90-98.
- [23] Koren, Y. and Borenstein, J. (1991) "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", Proc. of the IEEE Conference on Robotics and Automation, Sacramento, California, 1991, pp. 1398-1404.
- [24] Laumond, J. (1986) "Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints", Intelligent Autonomous Systems, An International Conference held in Amsterdam, The Netherlands, pp 346-354.
- [25] Li, Y.P., Zielinska, T., Ang Jr. M.H. and Linz, W. (2006) "Vehicle Dynamics of Redundant Mobile Robots with Powered Caster Wheels", Proceedings of the Sixteenth CISM_IFToMM Symposium, Romansy 16, Robot Design, Dynamics and Control, Warsaw, Poland, pp. 221-228.
- [26] Lingelbach, F. (2004a) "Path planning using probabilistic cell decomposition," in Proc. of the IEEE Int. Conf. on Robotics and Automation, New Orleans, LA, USA, pp. 467-472.
- [27] Lingelbach, F. (2004b) "Path Planning for Mobile Manipulation using Probabilistic Cell Decomposition", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan.
- [28] Lozano-Perez, T. (1987). "A simple Motion-Planning Algorithm for General Robot Manipulators", IEEE J. of Robotics and Automation. RA-3(3): 224-238.
- [29] Moravec, H.P. and Elfes, A. (1985) "High Resolution Maps from Wide Angle Sonar, IEEE Conference on Robotics and Automation", Washington D.C. pp 116-121.

- [30] Nourani-Vatani, N., Bosse, M., Roberts, J., and Dunbabin, M. (2006) "Practical Path Planning and Obstacle Avoidance for Autonomous Mowing", In Proc. of the Australasian Conference of Robotics and Automation.
- [31] Ordóñez, C., Collins Jr. E.G., Selekwa, M.F., Dunlap, D.D. (2008) The virtual wall approach to limit cycle avoidance for unmanned ground vehicles, Elsevier, Robotics and Autonomous Systems 56, pp. 645–657.
- [32] Palma-Villalon, E. and Dauchez, P. (1988) "World Representation and Path Planning for a Mobile Robot", *Robotica*, Volume 6, pp 35-40.
- [33] Philippsen, R., Jensen, B. and Siegwart, R. (2007) "Towards Real-Time Sensor-Based Path Planning in Highly Dynamic Environments", *Autonomous Navigation in Dynamic Environments*, Springer Tracts in Advanced Robotics, vol.(35), pp. 135-148.
- [34] Sedighi, K.H., Ashenayi, K., Manikas, T.W., Wainwright, R.L. and Tai, H.M.(2004) "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm", *Proceeding IEEE Congress on Evolutionary Computation (CEC)*,1338-1345.
- [35] Simionescu, P.A., Dozier, G.V. and Wainwright, R.L. (2006) "A Two-Population Evolutionary Algorithm for Constrained Optimization Problems", *IEEE Congress on Evolutionary Computation*, 2006. CEC 2006. Department of Mechanical Engineering, The University of Tulsa, 600 S. College Ave., Tulsa, OK 74104 USA, pp:1647- 1653.
- [36] Song, G., Miller, S. and Amato, N. (2001) "Customizing PRM roadmaps at query time", *IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, pp. 1500-1505.
- [37] Sugihara, K. and Smith, J. (1997) "Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot", *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, pp. 138-146.
- [38] Takahashi, O. and Schilling, R.J. (1989) "Motion Planning in a Plane Using Generalized Voronoi Diagrams", *IEEE Transactions on Robotics and Automation*, Vol.5, No. 2, pp. 143-150.
- [39] Tseng, C., Crane, C. and Duffy, J. (1988) "Generating Collision Free Paths for Robot Manipulators", *Computers in Mechanical Engineering*, pp. 58-64.
- [40] Valavanis, K. P. Nelson, A. L. Doitsidis, L. Long, M. and Murphy, R. R. (2006), Validation of a Distributed Field Robot Architecture Integrated with a MATLAB Based Control Theoretic Environment: A Case Study of Fuzzy Logic Based Robot Navigation", *IEEE Robotics and Automation Magazine*, Vol. 13, No. 3, pp: 93-107.
- [41] Whitcomb, L. L. and Koditschek, D. E. (1991) "Automatic Assembly Planning and Control via Potential Functions", In Proc. IEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka, Japan, vol. (1), pp. 17-23.
- [42] Yu, J. Cai, Z. Duan, Z. (2007) "Detection of Static and Dynamic Obstacles Based on Fuzzy Data Association with Laser Scanner", *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, Haikou, Vol.4, pp.172-176.
- [43] Zou, X.Y. and Zhu, J. (2003) "Virtual Local target method for avoiding local minimum in potential field based robot navigation", *Journal of Zhejiang University of Science*. 4(3), 264-269.