A HYBRID ARTIFICIAL POTENTIAL FIELD-GENETIC ALGORITHM APPROACH TO MOBILE ROBOT PATH PLANNING IN UNKNOWN ENVIRONMENT

Dissertation submitted to Jawaharlal Nehru University, in partial fulfillment of the requirement for the award of the Degree of

MASTER OF TECHNOLOGY
In
COMPUTER SCIENCE AND TECHNOLOGY

By
LIU YANPING

Under the Guidance of
PROF. K. K. BHARADWAJ

SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI -110067
JULY 2011

JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI - 110067

CERTIFICATE

This is to certify that the dissertation entitled "**A Hybrid Artificial Potential Field - Genetic Algorithm Approach to Mobile Robot Path Planning in Unknown Environment**", being submitted by **Liu Yanping** to the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, in partial fulfillment of the requirement for the award of the Degree of **Master of Technology in Computer Science and Technology**, is a bona fide work carried out by him under the guidance and supervision of **Prof. K. K. Bharadwaj**.

The matter embodied in the dissertation has not been submitted for the award of any other Degree or Diploma.

Liu Yanping 22-7-2011

Liu Yanping
(Student)

22-07-2011

Prof. K. K. Bharadwaj
(Supervisor)

Dean,SC&SS
Jawaharlal Nehru University
New Delhi-67
India

प्रोफेसर कर्मेशु/Professor Karmeshu
डीन / Dean
स्कूल ऑफ कम्प्यूटर और पद्धति विज्ञान संस्थान
School of Computer and Systems Sciences
जवाहरलाल नेहरू विश्वविद्यालय
Jawaharlal Nehru University
नई दिल्ली / New Delhi-110067

ACKNOWLEDGEMENTS

First of all, I would like to thank and express my sincere gratitude to my supervisor, Prof. K.K. Bharadwaj for his expertise, support and constructive guidance throughout my research work at SC&SS. It was his confidence in me that encouraged me to pursue my research on Robotics, and I am indebted for his priceless supervision. I would also like to thank him for helping me gain experience on research, which will be very helpful for my future study.

Next, I would like to convey my thanks to Prof. Sonajharia Minz, the Dean of the School of Computer & Systems Sciences for providing the necessary facilities and the faculty members for helping me in my studies.

I would also like to thank, and to praise the staff and my fellow graduate students for the guidance and help throughout my master program. Specially thank to my labmates, Mr. Vibhor Kant, Ms. Pragya Dwivedi and Ms. Vinti Agarwal as they accompany and suggest me a lot during my research work. It has been an honor working with them.

Finally, I am highly thankful to my parents and my sisters for always encouraging and supporting financially for my natural interest in research. And also I acknowledge my friends for their support throughout the years.

LIU YANPING

ABSTRACT

Mobile robots are widely used in many fields and have great applicable potential in human society in the future. Research on path planning for mobile robots is one of the most important aspects in mobile robots research. Path planning for a mobile robot is to find a collision-free route, through the robot's environment with obstacles, from a specified start location to a desired destination while satisfying certain optimization criteria. Many algorithms for path planning have been studied and developed over the past few years. The algorithms focusing on static environments have reached mature stage, such as the Visibility Graph and the Cell Decomposition. Some main AI-based approaches for robot path planning are Genetic Algorithm (GA), Fuzzy Logic Control (FLC) and Artificial Neural Network (ANN). However, because of the complexity of the dynamic environments, research on path planning in the environments with dynamic obstacles is limited, which is a big challenge for the robotics researchers.

The objective of this research is to establish a method of path planning for mobile robots which could be applied to an unknown environment with the static and dynamic obstacles. A hybrid Artificial Potential Field – Genetic Algorithm approach is developed and implemented for accomplishing path planning on a mobile robot in unknown environment. The hybrid approach first uses Grid method where the mobile robot environment is represented by orderly numbered grids, each of which represents a location in the environment. The boundary of obstacles is formed by their actual boundary plus minimum safety distance considering the size of the robot. The grids adopted here do not limit movement of the path, but simplify the chromosome structure and genetic operation by discretizing the environment. Then, it applies Genetic Algorithm, a global planner, to find an approximate optimal path according to the currently known environment. Compared with the traditional GA approaches for path planning, the GA proposed here uses variable length chromosomes. The evaluation not only considers the length of the path, but also takes into consideration Smoothness and Path Security. Smooth operation is used on the feasible path to get a more suitable path for the robot's motion. At last, Artificial Potential Field method, a local planner, is applied to follow

the path obtained by GA from one intermediate node to next intermediate node avoiding the obstacles. Compared with the traditional APF approach which only considers the position of obstacles and targets, the APF method proposed in this dissertation uses novel force functions which consider both position and velocity of the obstacles. The position and velocity of the obstacles are vectors, including the information of magnitude and direction.

The dissertation shows through theoretical analysis, simulation, and experiment that the developed approach can be effectively used to plan an optimal path and avoid collisions with obstacles in the environment.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Science fiction movies, cartoons and novels have long depicted our future with robots. Popular examples include Rosie from the Jetsons and C3P0 from Star Wars that assist, replace and extend human capabilities. But today, these visions of robots do not seem so far from reality. The number of robots being adopted is growing exponentially each year and the range of tasks they can perform continues to expand into new domains such as nursing, entertaining and cleaning.

Mobile robots have great applicable potential in human society in the future. The function of robotics will no longer be restricted to accomplish tasks in assembly and manufacturing at a fixed position. In order to accomplish practical tasks, a mobile robot has to be navigated smoothly in the real world wherein unexpected changes take place.

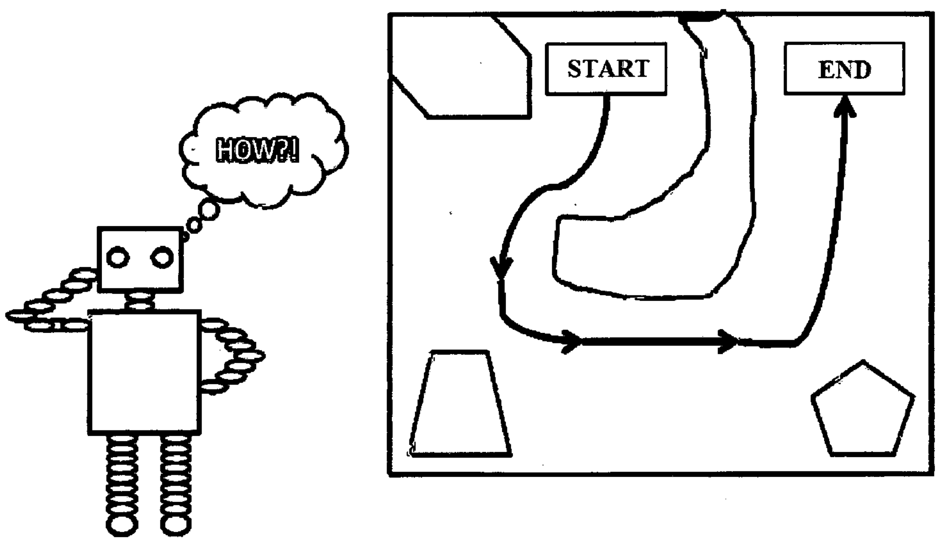

FIGURE 1.1 Path Planning Problem

In the control of mobile robot, path planning is one of the key issues. The robot path planning problem can be typically described as follows: given a robot and a description of its working environment, plan a collision free path between two specified locations that satisfies certain optimization criteria [Prahlad 2001]. Path planning research of mobile robot in known environment has reached mature stage, but the path planning in unknown environment is a big challenge. In practice, it is often no complete knowledge about the environment. Having a detailed map with all the obstacles marked seems to be unrealistic for most of the situations.

Whether the robot path planning belongs to which category, the path should meet the following conditions:

i) The path should be collision-free;
ii) The path should be as short as possible and search time should be as little as possible;
iii) The path should be as smooth as possible.

Many algorithms for path planning have been studied and developed over the past few years. The main methods of path planning for mobile robot can be divided into two categories - Artificial Potential Field (APF) approaches and Artificial Intelligence (AI) approaches. The main AI-based approaches for robot path planning are Genetic Algorithm (GA), Fuzzy Logic Control (FLC) and Artificial Neural Network (ANN) [Cao Qixin 2006]. Recently, many researchers have applied the hybrid approaches, such as Fuzzy-Genetic approach, on path planning.

This dissertation develops and implements a hybrid Artificial Potential Field – Genetic Algorithm approach to mobile robot path planning in unknown environment. The approach first uses Grid method to represent the environment. Then, it applies Genetic Algorithm, a global planner, to find an approximate optimal path according to the currently known environment. At last, Artificial Potential Field method, a local planner, is applied to follow the path from one intermediate node to next intermediate node avoiding the obstacles. This chapter introduces some background information about robot path planning, the motivation of the project, and the organization of this dissertation.

1.1 Background

At present, research on various algorithms for mobile robot path planning is a hot topic. Mobile robots are widely used in many hazardous industrial fields where there may be dangerous for people, such as aerospace research, the nuclear industry, and the mining industry.

To find a safe path in a dangerous environment for the mobile robot is an essential requirement for the success of any mobile robotic systems. Therefore, research on path planning algorithms to make the robot move from the start point to the target point without collision with obstacles is a fundamental requirement for the mobile robot safety in such environments. Moreover, to reduce the processing time, communication delay and energy consumption, the planned path is naturally required to be optimal with the shortest length.

At the initial stage of the robot industry, a robot was simple constituted by mechanical arms controlled by motor engines. Path planning for the robot was often in stationary obstacle environment. As an example of the robot, path planning in static environment were discussed in [J. Cook 1992]. However, with the development of the robot technology, robots have been used in many industrial fields such as aerospace research, marine research, and mining industry, to just mention a few. A lobster-like underwater walking robot [J. Ayers 2004] is one of these new types of robots. Recently, Australian researchers have developed an unmanned underwater vehicle robot for reef surveying [Williams and Mahon 2004]. The robot is equipped with sonar and vision systems, and works at the platform of the sea. Thus, how to respond quickly to the changing environment to avoid the stationary rocks in the seabed and big moving fish is a primary issue in the design and operation of the robot.

1.1.1 The Need for Path Planning

There is an increasing need for proficient path planning systems. Robots have been used for several years in industrial assembly plants. These robots move components into place, weld and bolt them together, and perform many functions which would previously have required large manual work forces. These robots are controlled by programs defining the specific movements which must be performed in order to achieve a goal. A change in the goal, for instance the introduction of a design change for a new model of car, would require expensive reprogramming using a robot control language. It is suggested that a better approach is to build intelligent robot systems which when provided with a goal, can establish for themselves the set of positions which must be followed to achieve the goal.

It would be of great benefit to have independent robots which do not rely on constant human intervention, but are able to continue performing their job even if the goals of that job change slightly, or if the environment in which they are working changes. As well as the possible financial advantages such robots might bring to commercial operations, they could also be used for hazardous work such as in nuclear reactors or in underwater situations, thus removing the need for humans to be placed at risk performing these dangerous jobs [Simon Kent 1999].

1.1.2 The Path Planning Problem

The general requirement of path planning is the ability to move a robot between two points along a collision free course within a given environment. Two techniques often followed in achieving this goal are "path planning" and "obstacle avoidance".

Ideally the route should be such that the robot avoids collisions with other objects in the environment, whether they are stationary or moving (as in the case of other robots working in the same space). Certain robots, such as complex robot manipulators (robot arms), may even be able to collide with themselves — this should be avoided.

Furthermore, the route which is computed should be optimized so that it should minimize some dependent variables, such as the distance covered or the energy used, in executing the path. Typically there will be many possible paths between two points but only the most efficient is sought.

If there exists only one robot in an environment, with all obstacles remaining stationary, path planning alone may be sufficient for the robot to complete its task. The route which is pre-planned can be followed without any chance of collision. However, since the pre-planned route is generated on the basis of information available at a single instance, path planning alone may not be sufficient if the state of the workspace is continually changing as other objects or people move around within the environment. This need for obstacle avoidance also exists when the robot is working within an unknown or partially known environment, where it is not possible to rely on pre-planned routes because of the limited information available to the planner.

In this case, the robot must have the ability to detect what is occurring in its immediate environment by means of some kind of sensors. When an unexpected obstacle is sensed, evasive action must be taken to avoid a collision with that obstacle. A new path can then be planned on the basis of the most up-to-date information available.

In order to develop fully independent robots, planning systems need to be developed which can generalize. Given previous knowledge of situations, humans are able to generalize to new, but similar situations. For a robot, it would be desirable to have a planner which does not have to return to first principles and generate the plan from scratch every time. Instead, the knowledge of previous experiences should be somehow stored and used to achieve future goals faster. There should not be a reliance on humans to instruct the robot what actions to execute precisely.

1.2 Motivation

Path planning and obstacle avoidance can be considered in the static or dynamic environments. The information of the environment can be completely known or partially known through sensors.

Generally, the approaches for mobile robot path planning can be divided into global path planning and local path planning. Global path planning, which is also called the deliberative approach, tries to use global information of the environment to create a smooth destination-directed path and then let the robot reach the goal along the planned path. It is composed of pre-detecting by vision, planning, and acting. In this approach, a global map is usually created by incorporating the sensed information of the environment and the motion decision of the robot is based on the map. This approach normally can find an optimized path for the robot to reach the selected position under some objective function. These objective functions can be minimizing time, energy, variation of the velocity or distance, etc. Global path planning is generally destination-directed. However, the approach of global path planning lacks robustness due to environment uncertainty. As the map only represents a static environment, this method needs repeat cycling of detect-plan-act loop frequently for a robot to accomplish the task in a changing environment, which leads high computational requirements. A robot often cannot make quick accurate decisions when the deliberate method is used in a non-pre-known changing environment, and so global path planning is generally unsuitable in a changing environment. On the other hand, local path planning, which is also called the reactive approach, moves the robot in an unknown environment to avoid collisions with obstacles. Usually, only a feasible path might be found in the reactive method. Local path planning is more flexible in an unknown time-varying environment. However, due to its limited reasoning and representational ability, a complete reactive approach is not destination-directed, a path to the destination position for a robot is not guaranteed.

The reasons of choosing the hybrid Artificial Potential Field – Genetic Algorithm approach to plan the optimal path for robots are listed below:

i) Path planning methods based on Genetic Algorithm don't require derivative information of the solution due to it is stochastic in nature. It is robust, being capable of searching the entire solution space to get global optimization. It is easy to parallelize. However, it requires many function evaluations. As GA searches a global optimal path in a very large workspace, the time complexity is very high. Therefore, more efficient path planning methods in dynamic environments need to be developed for adapting the development of robotics research.

ii) Artificial Potential Field (APF) for path planning was first developed by Khatib in 1985. The advantages of obstacle avoidance algorithm based on Artificial Potential Field are low computational complexity, good real-time characteristics and smooth path. However, the disadvantages are existing trap situations, oscillations in the presence of obstacles and goals nonreachable with obstacles nearby (GNRON) problem caused by local minima. Another problem of APF is that through the path is collision free, it may be not the optimal path.

iii) Combining the Artificial Potential Field and Genetic Algorithm can solve the local minima problem caused by Artificial Potential Field and improve the searching efficiency when using Genetic Algorithm.

1.3 Organization of Dissertation

The following are the descriptions of the organization of this dissertation. This dissertation includes five chapters. Chapter 1 states the main issues of the dissertation, including a general introduction to the path planning problem and motivation and organization of this dissertation. Chapter 2 provides detailed background and literature review information for Artificial Potential Field (APF) and Genetic Algorithm (GA). As one of the main contributions of this dissertation, chapter 3 describes the proposed hybrid Artificial Potential Field – Genetic Algorithm approach to mobile robot path planning. It illustrates the methodology of the hybrid approach to calculate the optimal path for a robot in dynamic environment. In chapter 4, the implementation and experimental results are given. Finally, the conclusion and future work are discussed in chapter 5.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

The path planning problem is known to be PSPACE hard. This means that the complexity of the path planning problem increases exponentially with the dimension of the configuration space. The configuration space is the space of all complete specifications of the position of every point of a robot system [Salvatore Candido 2005].

Path planning for mobile robots is one of the most important aspects in robot navigation research. The mobile robot path planning task is to find a collision-free route, through an environment with obstacles, from a specified start point to a target point while satisfying certain optimization criteria. This chapter classifies various robot path planning methods in different ways and gives some general information about traditional path planning methods in different environments such as the Visibility Graph method, Grid method, then discusses Artificial Potential Field (APF) and Genetic Algorithm (GA) in details.

The robot path planning methods could be classified into different kinds based on different situations. Depending on the environment where the robot is located in, the path planning methods can be classified into the following two types:

i) Robot path planning in a static environment which only contains the static obstacles;

ii) Robot path planning in a dynamic environment which contains static and dynamic obstacles.

2.1 Path Planning Methods in a Static Environment

In a static environment, the obstacles are stationary. If the information of obstacles is known, the optimal path could be computed offline prior to the movement of the robot. The path planning techniques for a static environment are relatively mature. Representative path planning methods for a static environment include the Visibility Graph method, Voronoi

Diagrams method, the Grids method and the Cell Decomposition method [Hui Miao 2009].

Moreover, the Genetic Algorithm, the Simulated Annealing Algorithm, and some other optimization methods have also been used to obtain the optimal path for the robot. Davidor [Davidor 1991] developed a tailored Genetic Algorithm with a modified crossover operator to optimize robot path. Nearchou [Nearchou 1998] used the number of vertices produced in Visibility Graphs to build fixed length chromosomes in which the presence of a vertex within the path is indicated by setting of a bit at the appropriate locus. The method applied a reordering operator for performance enhancement, and the algorithm was capable of determining a near-optimal solution. Cai and Peng [Cai et al. 2006] developed a fixed-length decimal encoding mechanism to replace the variable-length encoding mechanism and other fixed-length binary encoding mechanism used in the genetic approach for robot path planning.

2.1.1 The Visibility Graph Method

In this method, a Visibility Graph is used in robot path planning when the geometry of the environment is known. The main idea of the Visibility Graph method is that if there is a collision-free path between two points, then there is a polygonal path that bends only at the obstacles vertices. As FIGURE 2.1 shows, collision-free path (in curves) could be transformed into line segments (straight line).

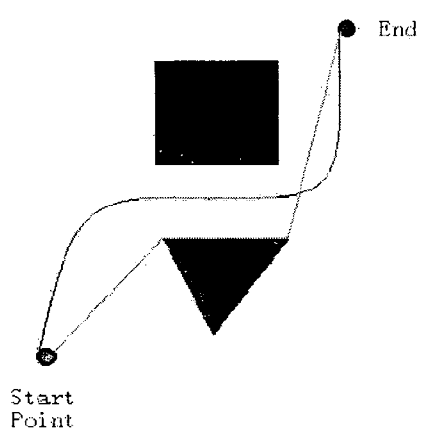

FIGURE 2.1 Visibility Graph

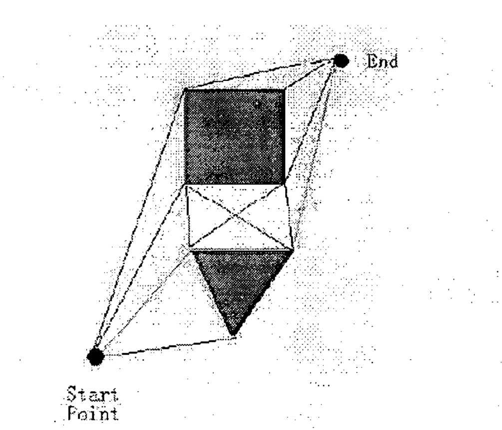

FIGURE 2.2 Complete Visibility Graph

A Visibility Graph is constituted by nodes and edges. Nodes are the start point, destination point and the vertices of all obstacles. Edges are strait-line segment between two nodes which do not pass through obstacles. FIGURE 2.2 shows the complete Visibility Graph based on FIGURE 2.1.

FIGURE 2.2 shows that there are multiplex paths that could lead the robot from the start point to the destination. Then, any optimization algorithms such as the Genetic Algorithm [M. Scott 2004] and the Simulated Annealing Algorithm [C. Edelman et al. 1988] could be used to calculate the optimal path for the robot. The defect of the Visibility Graph method is that the efficiency of the algorithm is low. Furthermore, the obtained path is often very close to obstacles and thus, may lead to crash of the robot. However, this problem can be fixed by enlarging the obstacles by a value according to the dimension of the robot. In this way, the robot can approach obstacles without collision.

2.1.2 The Cell Decomposition Method

The Cell Decomposition method is another algorithm for searching the collision free path for a robot. It uses small non-overlapping grid cells to represent the entire environment. The cells usually are simple squares. There are three types of cells: empty cell, mixed cell and full cell. An empty cell is a free space, where the robot could go through in the environment. A mixed cell contains obstacles and free space. A full cell is the block of the obstacles. In a two-dimensional map, a Quadtree is used to decompose the map, as shown in FIGURE 2.3.

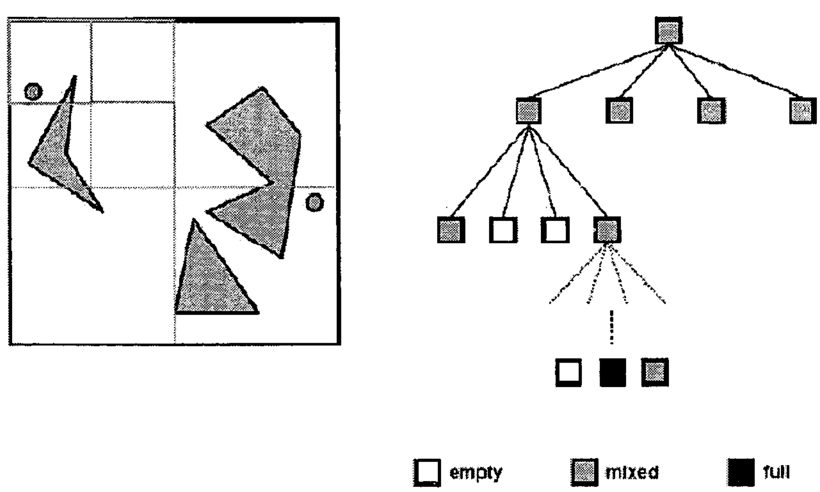

FIGURE 2.3 Quadtree Decomposition

The Cell Decomposition method is briefly outlined below [Hui Miao 2009]:

i) Decompose the map into cells.

ii) Search for a sequence of mixed or free cells that connect the start point and goal point.

iii) Further decompose the mixed cells.

iv) Repeat (ii) and (iii) until a sequence of free cells is found.

Then, the method uses an optimization algorithm, such as A* algorithm, to find the optimal path for the robot. The A* algorithm is first described early in 1968 by Hart, Nilsson and Raphael. The algorithm is a best-first, tree search algorithm, and could find the shortest path from the start point to the target point. Lingelbach in [Lingelbach 2004] used the Cell Decomposition method combined with probabilistic sampling to plan path for a robot in high-dimensional static configuration spaces. The potential field approach based on harmonic function is computed over a non-regular grid decomposition of a high-dimensional space obtained with the probabilistic sampling of cells.

2.2 Path Planning Methods in a Dynamic Environment

Research on methods that deal with the static environment path planning has been introduced in previous sections. Currently, the path planning methods to find paths in a static environment have been well developed with hundreds of published papers. Give the entire information of the environment, the global optimal or near-optimal path could be found by these algorithms.

However, in practical applications, robots often face obstacles that are not all static in the environment, the status and the movement of the obstacles change continuously in the map. Moving obstacles in a dynamic environment increases the difficulty of path planning for the robot in the map.

Unlike the situation for path planning in a static environment, limited reports have been found in the open literature to discuss optimal path planning in dynamic environments. Complexity and uncertainty increase with the number of the dynamic obstacles. Therefore, traditional path planning algorithms, such as the Visibility Graph, the Voronoi Diagrams and the Grids method, do not perform well in dynamic environments. It is also difficult to gain the optimal path for the robot using these methods. Robot path planning in a dynamic environment is thereby an issue for further research. In a dynamic environment, how to manipulate the robot so that it can move to the goal safely and optimally without collision is an important issue of concern.

In 2005, Chestnutt, Lau and Cheung [J. Chestnutt et al. 2005] used a modified A* algorithm to calculate path for a Honda ASIMO humanoid robot. The path planning method is applied to real robots rather than simulation on software. A grid of cells is employed to represent the environment. Colour cells represent the obstacles. The cells create a bitmap representing the free spaces and obstacles in the map. The algorithm plans a sequence of footstep positions to navigate toward a goal location based on known static and moving obstacles with predictable trajectories.

Wang and Sillitoe [Y. Wang et al. 2007] proposed a vertices Genetic Algorithm planner in 2007. The planner is able to rapidly determine optimal or near-optimal solutions for a mobile robot in an environment with moving obstacles. The method uses the vertices of the obstacles as search space and produces off-line path planning through the environment with dynamic obstacles.

It first incorporates the robot speed into the genetic genes, which could optimize both the travel time and distance of the robot. Before the robot starts movement, the complete motion knowledge of the moving obstacles in the observed region is available for the robot. The robot uses the Genetic Algorithm based planner to calculate the time or distance optimized solution and then starts to travel.

A hybrid navigation method [S. Lbszlo et al. 2003] was proposed in 2003. The method consists of two modules: global and local modules, which could combine two robot path methods to deal with the global map information and local sensor information.

i) The global module specifies the global route positions. It uses prior information on the navigation environment and chooses critical points to pass through before actual navigation takes place. This module uses the A* algorithm to determine the optimal route to the specified goal position. Using the algorithm, it is possible to find the optimal route to the goal.

ii) The local module carries out the navigation itself, relying on current sensor data, thus making it easier to avoid static or dynamic obstacles. It uses a fuzzy neural representation of the potential field based navigation method.

Firstly, the global planning module finds the optimal route to the goal and proposes the positions to pass through as intermediate points. These intermediate points are then passed one by one to the local navigator, which makes the robot reach them while reactively avoiding the obstacles present in the environment, according to the potential function previously supplied to the local navigator.

A dynamic environment is more complicated than a static environment in the robot path planning issue. Several methods were proposed to solve the problem. Because the moving information and the obstacle information can be known in advance of movement, the optimal solution can still be obtained.

2.3 Artificial Potential Field (APF) for Path Planning

The Artificial Potential Field method, an idea of adding an imaginary force on the robot, was first suggested by Andrew and Hogan (1983) and Khatib (1985) for obstacle avoidance of manipulators and mobile robots. In their method, the obstacle exerts a virtual repulsive force on the robot, while the goal position applies a virtual attractive force to the robot. The sum of the attractive and repulsive forces is then used to determine the direction and the speed of the robot. The approach was first implemented on a robot by Arkin (1989), but in his experiment, the speed of the robot was very slow (0.12m/s).

2.3.1 Traditional Artificial Potential Field

The application of APF for obstacle avoidance was first developed by Khatib in 1985.

The basic idea of the APF approach is to fill the robot's workspace with an artificial potential field in which the robot is attracted to its target position and is repulsed away from the obstacles [Cao Qixin 2006].

The APF uses two types of potential field, namely a repulsive potential field to force a robot away from obstacles or forbidden regions and an attractive potential field to drive the robot to its goal. An obstacle is considered as point of highest potential, and a goal as a point of lowest potential.

The robot moves under the action of a force that is equal to the negative gradient of that potential, and it is driven towards the target position with the lower potential. FIGURE 2.4 and FIGURE 2.5 indicate the potential field created by a goal and an obstacle, respectively.

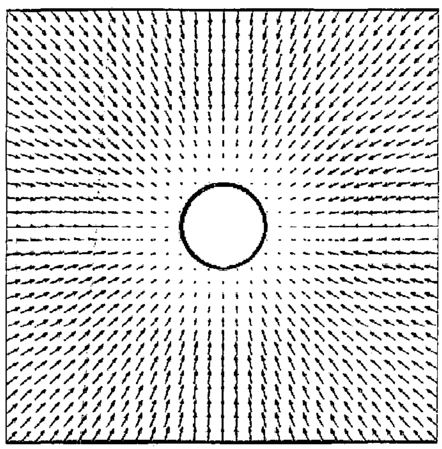

FIGURE 2.4 Attractive potential field created by a goal

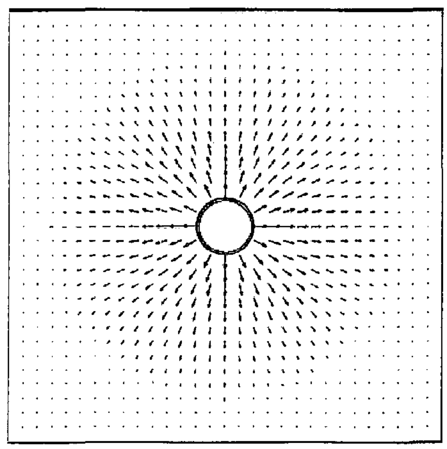

FIGURE 2.5 Repulsive potential field created by an obstacle

i) The Attractive Potential Function

For simplification, the robot is often considered as a mass point. The position of robot is expressed as a vector of $X_r = [x, y]^T$ in two-dimensional workspace. An expression for the conventional attractive potential function $U_{att}(X_r)$ is described by

$$U_{att}(X_r) = \frac{1}{2}\xi \ell^m (X_r, X_g) \tag{2.1}$$

Where ξ is a positive scaling factor, $\ell(X_r, X_g) = \|X_g - X_r\|$ is the distance between the robot and the goal. And $m = 1 \; or \; 2$. For $m = 1$, the attractive potential is conic in shape and the resulting attractive force has constant amplitude except at the goal, which U_{att} is singular. For $m = 2$, the attractive potential is parabolic in shape; the corresponding attractive force is then given by the negative gradient of this attractive potential:

$$F_{att}(X_r) = -\nabla U_{att}(X_r) = \xi(X_g - X_r) \tag{2.2}$$

It converges linearly towards zero as the robot approaches the goal. The force will drive mobile robot to reach the goal.

ii) The Repulsive Potential Function

The conventional repulsive potential function can be represented as

$$U_{rep}(X_r) = \begin{cases} \frac{1}{2}\eta(\frac{1}{\ell(X_r, X_o)} - \frac{1}{\ell_0})^2, & \text{if } \ell(X_r, X_o) \le \ell_0 \\ 0, & \text{if } \ell(X_r, X_o) > \ell_0 \end{cases} \tag{2.3}$$

where η is a positive scaling factor, $\ell(X_r, X_o)$ denotes the minimum distance between the robot and the obstacle, X_o denotes the point on the obstacle such that the distance between this point and the robot is minimal between the obstacle and the robot, ℓ_0 is a positive constant and represents the influence scope of the obstacle.

The corresponding repulsive force is given by:

$$\begin{aligned} F_{rep}(X_r) &= -\nabla U_{rep}(X_r) \\ &= \begin{cases} \eta \cdot (\frac{1}{\ell(X_r, X_o)} - \frac{1}{\ell_0}) \cdot \frac{1}{\ell^2(X_r, X_o)} \cdot \nabla \ell(X_r, X_o), & \text{if } \ell(X_r, X_o) \le \ell_0 \\ 0, & \text{if } \ell(X_r, X_o) > \ell_0 \end{cases} \end{aligned} \tag{2.4}$$

Therefore, the total force applied to the robot is expressed as

$$F_{total} = F_{att} + F_{rep} \tag{2.5}$$

The robot moves under the action of the composition of forces F_{total}, which is the summation of the goal's attractive force F_{att} and the obstacle's repulsive force F_{rep}, as shown in FIGURE 2.6. One path of the robot in the environment of an obstacle and a target is shown in FIGURE 2.7.

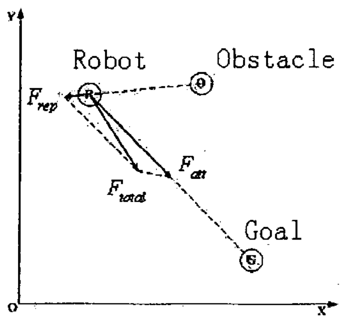

FIGURE 2.6 Visual attractive force of robot in APF

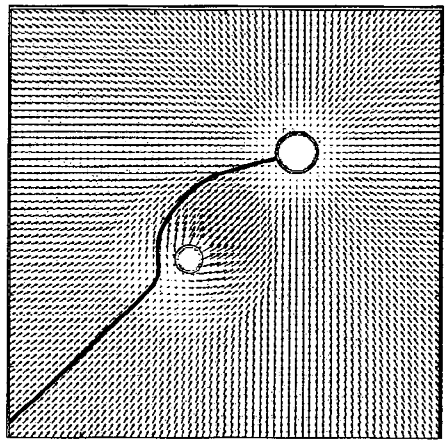

FIGURE 2.7 One path of the robot in the environment of an obstacle and a target

This method is particularly attractive because of its elegant mathematical analysis and simplicity. However, it has some inherent limitations. A systematic criticism of the inherent problems based on mathematical analysis was presented in [Y. Koren and J. Borenstein 1991], which includes the following:

i) Trap situations due to local minima;

ii) No passage between closely spaced obstacles;

iii) Oscillations in the presence of obstacles;

iv) Oscillations in narrow passages.

Besides the four problems mentioned above, there exists an additional problem, goals nonreachable with obstacles nearby (GNRON) [S. S. Ge and Y. J. Cui 2000]. It happens when the goal is very close to an obstacle. When the robot approaches its goal, it approaches the obstacle as well. As a consequence, the attractive force decreases, while the repulsive force increases. Thus, the robot will be repulsed away rather than reaching the goal. A local minimum is the case which occurs when the total force acting on the mobile robot is summed up to zero although robot has not reached its goal position yet. When the mobile robot falls into the local minima of potential function, the final state will never be reached. This situation frequently happens as obstacle is in the vicinity of the target point. FIGURE 2.9 shows the situation of local minima.

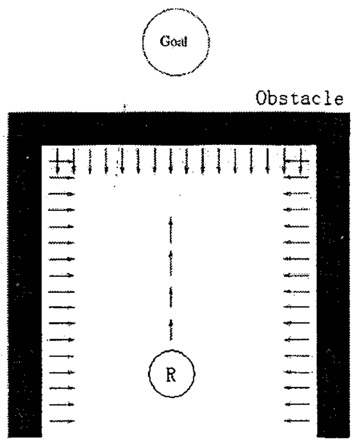

FIGURE 2.8 Local minima

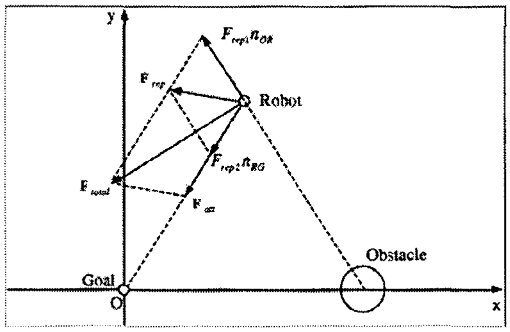

FIGURE 2.9 Total force derived by the new potential function

There are many methods to resolve the local minima problem of traditional APF. The following section will discuss some evolutionary approaches based on APF.

2.3.2 Some Evolutionary Artificial Potential Field Approaches

i) Adding "escape force" [Prahlad et al. 2001]

In the evolutionary artificial potential field, local minima exist within the areas around the null-potential points. The null-potential point condition is:

$$F_{att} + \sum F_{rep} = 0 \tag{2.6}$$

Where F_{att} represents the attractive force, $\sum F_{rep}$ represents the total repulsive forces.

A local minimum is identified when the following two conditions are satisfied.

$$\begin{cases} \dfrac{F_{att} - \sum F_{rep}}{\sum F_{rep}} < b \\ \cos(\angle F_{att} - \angle \sum F_{rep}) < -\cos(c) \end{cases} \tag{2.7}$$

When these two conditions are satisfied, an additional escape force can be applied on the mobile robot to escape.

$$F_e = \left(\frac{1}{dD_{ro}^m} \right) \left(\left| \cos(\angle F_{att} - \angle \sum F_{rep}) - \cos(c) \right| \right) \tag{2.8}$$

Where F_e represents the escape force, D_{ro} is the distance between the robot and an obstacle, the parameters b, c, d and m need to be optimized.

ii) New Repulsive Potential Function [S. S. Ge and Y. J. Cui 2000]

The goals nonreachable with obstacle nearby (GNRON) problem arises because the global minimum of the total potential field is not at the goal position when the goal is within the influence distance of the obstacle. This problem is due to the fact that as the robot approaches the goal, the repulsive potential increases as well. If the repulsive potential approaches zero as the robot approaches the goal, then the total potential will take the global minimum at the goal. Therefore, we can take the relative distance between the robot and the goal into consideration as

$$U_{rep}(X_r) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\ell(X_r, X_o)} - \frac{1}{\ell_0}\right)^2 \ell^n(X_r, X_g), & \text{if } \ell(X_r, X_o) \le \ell_0 \\ 0, & \text{if } \ell(X_r, X_o) > \ell_0 \end{cases} \tag{2.9}$$

Where $\ell(X_r, X_o)$ is the minimal distance between the robot and the obstacle, $\ell(X_r, X_g)$ is the distance between the robot and the goal, ℓ_0 is the distance of influence of the obstacle, and n is a positive constant.

The introduction of $\ell(X_r, X_g)$ ensures that the total potential $U_{total}(X_r) = U_{att}(X_r) + U_{rep}(X_r)$ arrives at its global minimum, 0, if and only if $X_r = X_g$.

The potential function $U_{total}(X_r)$ should have the property that the total force, the sum of the attractive force and the repulsive force, pushes the robot away from the obstacles and pulls toward the goal.

When the robot is not at the goal, $X_r \ne X_g$, the repulsive force is given by

$$\begin{aligned} F_{rep}(X_r) &= -\nabla U_{rep}(X_r) \\ &= \begin{cases} F_{rep1}\mathbf{n}_{\mathbf{RO}} + F_{rep2}\mathbf{n}_{\mathbf{RG}}, & \text{if } \ell(X_r, X_o) \le \ell_0 \\ 0, & \text{if } \ell(X_r, X_o) > \ell_0 \end{cases} \end{aligned} \tag{2.10}$$

Where

$$
\begin{cases} F_{rep1} = \eta\left(\dfrac{1}{\ell(X_r, X_o)} - \dfrac{1}{\ell_0}\right)\dfrac{\ell^n(X_r, X_g)}{\ell^2(X_r, X_o)} \\ F_{rep2} = \dfrac{n}{2}\eta\left(\dfrac{1}{\ell(X_r, X_o)} - \dfrac{1}{\ell_0}\right)^2 \ell^{n-1}(X_r, X_g) \end{cases} \tag{2.11}
$$

$\mathbf{n}_{\mathbf{RO}} = \nabla\ell(X_r, X_o)$ and $\mathbf{n}_{\mathbf{RG}} = -\nabla\ell(X_r, X_g)$ are two unit vectors pointing from the obstacle to the robot and from the robot to the goal, respectively. FIGURE 2.9 shows the total force derived by the new potential function.

There are three forms of repulsive force functions as n varies: $1)\, 0 < n < 1$; $2)\, n = 1$; $3)\, n > 1$. The detailed repulsive force functions are represented in [S. S. Ge and Y. J. Cui 2000]. No matter n belongs to which category, the repulsive potential function ensures that the goal position is the global minimum of the total potential.

2.4 Genetic Algorithms (GAs) for Path Planning

Genetic algorithms (GAs) are search algorithms and optimization techniques using the principles of natural selection inspired by Darwin's theory about evolution (the survival of the fittest). In GA based approaches, the variables are represented as genes on a chromosome. Genetic Algorithms feature a group of candidate solutions (population) on the response surface. Through natural selections and the genetic operations, recombination and mutation, chromosomes with better fitness are found [Jianping Tu et al. 2003].

Recently, it has been widespread interest using genetic and evolutionary algorithms. Compared to traditional search and optimization methods, the evolutionary algorithms are robust, global and generally more straightforward to apply in situations where there is little or no priori knowledge about the problem to solve. As evolutionary algorithms require no derivative information or formed initial estimates of the solution, and because they are stochastic in nature, evolutionary algorithms are capable of searching the entire solution space with more likelihood of finding the global optimum. The Genetic Algorithms are powerful search algorithms based on the mechanism of natural selection and use operations of reproduction, crossover, and mutation on a population of strings.

The popularity of GAs is motivated by a number of factors including: [Tom 1997]

i) Evolution is known to be a successful, robust method for adaptation within biological systems.

ii) GAs can search spaces of hypotheses containing complex interacting parts, where the impact of each part on overall hypothesis fitness may be difficult to model.

iii) Genetic Algorithms are easily parallelized and can take advantage of the decreasing costs of powerful computer hardware.

A typical Genetic Algorithm requires:

i) A genetic representation of the solution domain;

ii) A fitness function to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. It is always problem dependent. Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, inversion and selection operators.

In path planning based on GA, any path from the start point to the goal is a solution, which is generation. First generation is selected by Roulette Wheel Selection, i.e. at the beginning a large random population of strings is generated. Strings representing unacceptable solutions are eliminated and strings representing acceptable solutions get multiplied. Unacceptable solutions are strings that can not reach the target. Acceptable solutions are strings that can reach the target. The decision whether the string is acceptable or unacceptable is decided by the fact whether the string solution would lead the mobile robot into obstacles and whether the mobile robot is progressing towards the goal. Based on such decisions each string in the population is assigned a fitness value. Acceptable solutions will have higher fitness value and unacceptable solution would have lower fitness value [Jianping Tu 2003].

The challengeable problems will be met when applying GA on mobile robot path planning:

i) Encoding of the problem in a binary string, which maps a path from the start point to the goal. Each of the individuals (or chromosomes) of a genetic population is encoded by a genotype like in biology. A chromosome corresponds to a possible solution of the optimization problem.

There are two types of chromosome. One is a fixed length string that the numbers of genes are the same for all the solutions. Another one is that different solutions have different numbers of genes.

ii) Designing a fitness function

The key issue of GA is to determine an appropriate fitness evaluation function according to the pending problem. When design a fitness function, we should take into account the security of the path, the length of the path, and the smoothness of the path. The security of the path is primary factor. A suitable fitness function can be expressed as:

$$\mathrm{Fit} = w_1 \cdot \mathrm{Fit1} + w_2 \cdot \mathrm{Fit2} + w_3 \cdot \mathrm{Fit3} \tag{2.12}$$

Where $\mathrm{Fit1}, \mathrm{Fit2}, \mathrm{Fit3}$ denotes the sub-function of path length, the sub-function of path security, the sub-function of smoothness, respectively, w_1, w_2, w_3 denotes the weighted value of length, security, and smoothness degree in the fitness function.

iii) Choosing the configuration parameters is important as well. These include population size, crossover type, crossover probability, mutation probability, number of generation, etc. The population size dictates the number of chromosomes in the population. Larger population sizes increase the amount of variation present in the initial population at the expense of requiring more fitness evaluations. It has been found that the best population size is both applications dependent and related to the length of the chromosome. A good population of chromosomes contains a diverse selection of potential building blocks resulting in better exploration. For longer chromosomes and challenging optimization problems, larger population sizes are needed to maintain diversity and hence better exploration. Many researchers suggest population size between 25 and 100. Some researchers suggest that the optimal mutation rate is approximately as

$$P_m = 1/(M\sqrt{L}) \tag{2.13}$$

Where M is the size of population and L is the length of the chromosome.

2.4.1 Initialization of the Population

Initialization is the first stage of the iterative GA process, as represented in the flowchart in FIGURE 2.10. An initial population is randomly generated at the start of the run. If desired, the initial population could be interspersed with non-randomly individuals to give the GA process a head start. This might be a previously known good solution to a problem which is the best-to-date solution available. Hopefully, GA should either use the components of these non-random solutions to contribute to a better solution, or discard them if they have nothing to offer. Care must be taken not to unduly pollute the population with such previous knowledge, as the GA search may be inadvertently directed away from better solutions, and instead converge early on sub-optimal solutions because diversity in the population is lost.

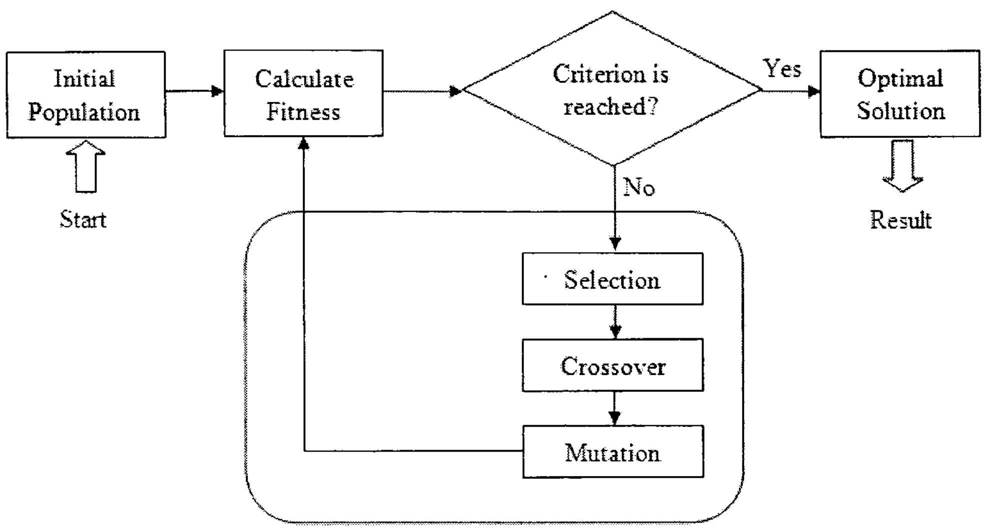

FIGURE 2.10 The flowchart of Genetic Algorithm

The population size dictates the number of chromosomes in the population. Larger population sizes increase the amount of variation present in the initial population at the expense of requiring more fitness evaluations. It has been found that the best population size is both applications dependent and related to the length of the chromosome. A good population of chromosomes contains a diverse selection of potential building blocks resulting in better exploration. For longer chromosomes and challenging optimization problems, larger population sizes are needed to maintain diversity and hence better exploration. Many researchers suggest population size between 25 and 100.

2.4.2 Evaluation

Each individual in the population is evaluated against a function (fitness function), to measure how well it performs against the problem it is addressing. The result of the evaluation is known as the fitness of the individual.

2.4.3 Selection

The next step is the evolutionary stage where a new population is created from the old population. Having measured the fitness, this information is used as the means of comparing the relative ability of individuals to solve the problem. During the evolutionary phase, those individuals with a higher fitness are more likely to survive in the new population. Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. This process of fitness evaluation and evolution is repeated as the GA process efficiently searches for an optimum or near optimum solution to the problem. Popular and well-known selection methods are fitness-proportionate selection using Roulette Wheel and Tournament Selection.

Roulette Wheel Selection

- Add up the fitness of all chromosomes
- Generate a random number R in that range

- Select the first chromosome in the population that gives you at least the value R (when all previous fitness are added)

Example:

Chromosome No.	**Fitness**
1	1
2	2
3	3
4	1
5	3
6	5
7	1
Total fitness	**16**

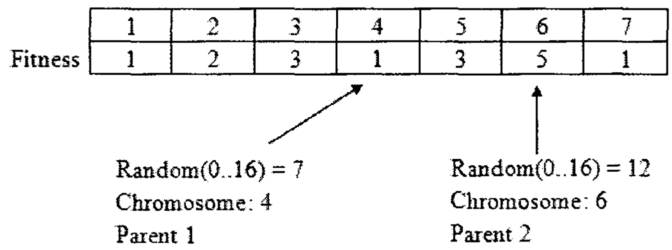

FIGURE 2.11 Roulette Wheel Selection

Tournament Selection

Tournament Selection is a method of selecting an individual from a population of individuals in a Genetic Algorithm. Tournament Selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected.

Tournament Selection pseudo code:

Choose k (the tournament size) individuals from the population at random
Choose the best individual from pool/tournament with probability p
Choose the second best individual with probability p*(1-p)
Choose the third best individual with probability $p*(1-p)^2$
and so on...

Deterministic Tournament Selection selects the best individual (when p = 1) in any tournament. A 1-way tournament (k = 1) selection is equivalent to random selection. The chosen individual can be removed from the population that the selection is made from if desired, otherwise individuals can be selected more than once for the next generation.

Tournament Selection has several benefits: it is efficient to code; works on parallel architectures and allows the selection pressure to be easily adjusted.

2.4.4 The Genetic Operators

i) Crossover

To produce new offspring, parts of individuals from the previous generation are exchanged using a GA operation called crossover which is similar to sexual reproduction. It is hoped that over next generations, all the useful sub-components, which are initially spread throughout the population, will combine in a single individual which will offer a 'good' or even perfect solution. FIGURE 2.12 illustrates the crossover operation.

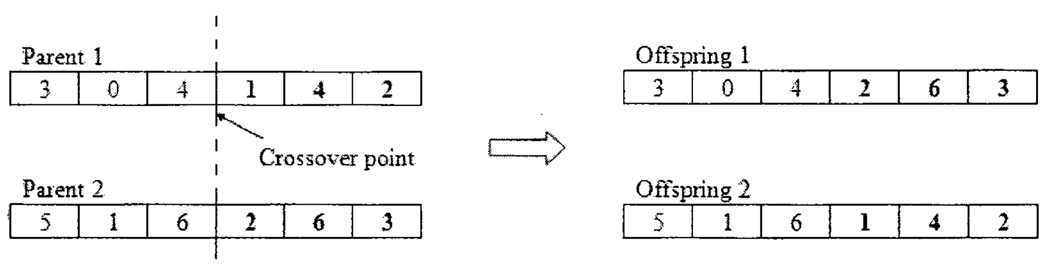

FIGURE 2.12 Single Crossover Operation

ii) Reproduction

To avoid the loss of good individuals from the population, and to improve the speed of convergence of GA, reproduction is also used to copy some better individuals to next generations.

iii) Mutation

Crossover and reproduction are the main operators used, typically to generate 90% and 10% of a new population respectively. Sometimes, it may be useful to introduce a mutation operator applied to individuals with a much lower probability [Simon Kent 1999]. Mutation rate determines the probability that a mutation will occur. Mutation is employed to give new information to the population and also prevents the population from becoming saturated with similar chromosomes. Large mutation rates increase the probability that good schemata will be destroyed, but increase population diversity. The best mutation rate is application dependent but for most applications is between 0.001 and 0.1. The mutation operation involves the random selection of an individual's component and the replacement of this component by another randomly created component. This operator can make sure the diversity of the population. It is useful to prevent convergence to a solution which is sub-optimal, and can be seen as the addition of new genes to the population. FIGURE 2.13 shows the mutation operation.

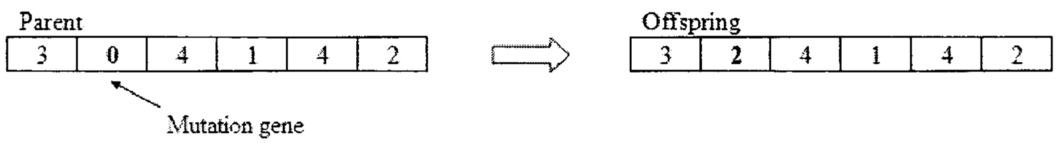

FIGURE 2.13 Mutation Operation

2.4.5 The Problem of Genetic Algorithm for Path Planning

Path planning methods based on Genetic Algorithm don't require derivative information of the solution due to it is stochastic in nature. It is robust, being capable of searching the entire solution space to get global optimization. It is easy to parallelize.

However, GA searches a global optimum path in a very large workspace, so the time complexity is very high. The time cost of GA increases exponentially when the dimension of the configuration space increases. GA is not working properly in dynamic environments.

2.5 The Need for Hybrid Approach

As discussed previously, the single approach such as Artificial Potential Field or Genetic Algorithm has its own advantages. However, there are existing trap situations, oscillations in the presence of obstacles and goals nonreachable with obstacles nearby (GNRON) problem caused by local minima in APF. Another problem is that through the path is collision free, it may be not the optimal path. The time cost of GA increases exponentially when the dimension of the configuration space increases. GA is not working properly in dynamic environments.

There are many benefits of applying hybrid Artificial Potential Field - Genetic Algorithm approach on mobile robot path planning. Combining the Artificial Potential Field and Genetic Algorithm can solve the local minima problem caused by Artificial Potential Field and improve the searching efficiency when using Genetic Algorithm. The next chapter will discuss a hybrid approach based on Artificial Potential Field and Genetic Algorithm in details.

CHAPTER 3

HYBRID APPROACH BASED ON ARTIFICIAL POTENTIAL FIELD AND GENETIC ALGORITHM

3.1 Overview

The mobile robot environment is represented by orderly numbered grids, each of which represents a location in the environment. The boundary of obstacles is formed by their actual boundary plus minimum safety distance considering the size of the mobile robot, which makes it possible to treat the mobile robot a point in the environment. In a large scale environment, suppose the environment is 10×10 meters with a robot size 10×10 centimeters. We can model the environment into 100×100 grids, and treat the robot as a point in the environment. FIGURE 3.1 shows an environment representation with 100×100 grids.

9900	9901	9902	9903	…	…	…	9997	9998	9999
9800	9801	9802	9803	…	…	…	9897	9898	9899
9700	9701	9702	9703	…	…	…	9797	9798	9799
:	:	:	:	…	…	…	:	:	:
:	:	:	:	…	…	…	:	:	:
:	:	:	:	…	…	…	:	:	:
300	301	302	303	…	…	…	397	398	399
200	201	202	203	…	…	…	297	298	299
100	101	102	103	…	…	…	197	198	199
0	1	2	3	…	…	…	97	98	99

FIGURE 3.1 Environment Representation

In this hybrid approach, a Genetic Algorithm is proposed first. It uses a simple effective path representation that combines grids and coordinates representations. Unlike other Grid methods, the grids adopted here do not limit movement of the path, but simplify the chromosome structure and genetic operation by discretizing the environment. This approach makes it possible to have one number for each gene and to use integer numbers instead of real numbers in chromosomes. A potential path is encoded as a sequence of grid numbers starting from the start point and ending at the target with a various number of intermediate nodes (FIGURE 3.2).

0 – 20 – 235 – 1011 – 3212 – 4032 – 5152 – 8472 – 9431 – 9563 – 9767 – 9866 – 9999

Start Point Intermediate Node Target Point

FIGURE 3.2 A sample chromosome

Such a grid representation is different from the one that usually uses grids to limit the movement of a path to be one of its eight adjacent cells and uses relative directions to represent a path. The proposed path representation is more like a coordinate representation, but differs by discretization and using integer numbers instead of coordinates (x, y).

As we know, the time cost of Genetic Algorithm increases exponentially when the dimension of configuration space increases. So the Genetic Algorithm will not work properly on the environments with large scale dimension. The suggested path length should be constrained to be less than 20.

After applying Genetic Algorithm, a global optimal or near-optimal path has been got according to the currently known environment. However, as the environment is dynamic, some obstacles are moving, we can not follow the fixed path obtained by GA simply. How to follow the path? How to avoid the obstacles? It is not efficient to apply GA again when the environment is changed. Therefore, a local planner is proposed. We need an effective local path planner that follows the path obtained by global planner and avoid the moving obstacles properly.

Since the predefined path is obtained before the movement of robot according to the currently known environment. Artificial Potential Field, a local planner, is proposed to follow the path from one intermediate node to next intermediate node. FIGURE 3.3 shows the main idea of the hybrid approach.

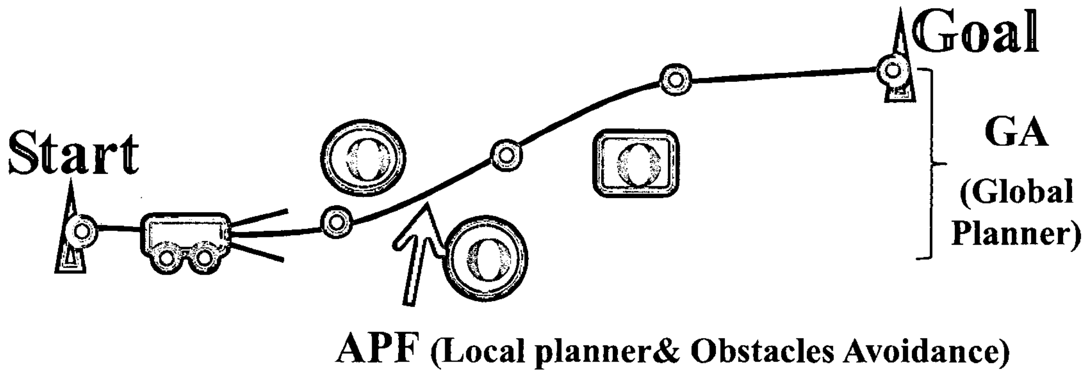

FIGURE 3.3 A hybrid approach

3.2 The Application of GA

GA first searches the optimal path or near-optimal path based on the currently known environment. In this dissertation, GA uses a chromosome with variable length. As the dimension of the environment is very large, the chromosome size (the number of nodes included in the path) should not be as long as the environment size, otherwise GA will not work efficiently. Therefore, the proposed chromosome size is less than 20. The following will discuss the chromosome, fitness function and genetic operators in details.

3.2.1 Representation and Initial Population

A chromosome represents a path as a sequence of nodes, where each node indicates a grid number representing a location in the environment. The first node is the start point and the last node represents the target point. A feasible path is a collision free path, i.e. no nodes fall on any obstacles. The length of a chromosome is variable and between 2 and maximum length 20. FIGURE 3.4 shows a sample chromosome.

0 – 20 – 235 – 1011 – 3212 – 4032 – 5152 – 8472 – 9431 – 9563 – 9767 – 9866 – 9999

Start Point   Intermediate Node   Target Point

FIGURE 3.4 A sample chromosome

The initial population is generated randomly, where each path has a random number of nodes.

3.2.2 Evaluation

Many traditional approaches to path planning consider only the length of the path to compute costs. This evaluation has many drawbacks, since shorter paths may not be safe and smooth and therefore do not represent optimal paths. The same evaluation methodology introduced by Xiao [Xiao et al. 1997] is utilized here with some modifications.

The fitness function is composed of three sub-functions:

i) Sub-function of Path Length

$$\text{Fit}1 = \sum_{i=1}^{N} d_i \tag{3.1}$$

Where N is the number of line segment of a path, d_i is the Euclidean distance of the two nodes forming the line segment.

ii) Sub-function of Smoothness

$$\text{Fit}2 = \sum_{i=1}^{N-1} g(l_i, l_{i+1}) \tag{3.2}$$

Where $g(l_i, l_{i+1})$ is the slope of the adjacent line segments l_i and l_{i+1}. To make sure the value is non-negative, we can define $g(l_i, l_{i+1}) = \cos(l_i, l_{i+1}) + 1$.

iii) Sub-function of Path Security

$$\text{Fit}\,3 = C\bullet\sum_{i=1}^{N}\beta_i \tag{3.3}$$

Where C is the coefficient, β_i is the coefficient denoting depth of collision, its definition is given as

$$\beta_i = \begin{cases} 0, & \text{if the i}^{\text{th}}\text{ line segment is feasible} \\ \sum_{j=1}^{M}\alpha_j, & \text{if the i}^{\text{th}}\text{ line segment intersects obstacles} \end{cases} \tag{3.4}$$

Where M is the number of obstacles the line segment intersects, the value of α_j is 0 or 1, which is determined by considering whether a line segment intersects an obstacle j. α_j is equal to 1 when the i$^{\text{th}}$ line segment intersects obstacle j. This evaluation gives penalty to infeasible path, but still keeps them in the population because they might become good feasible solutions after certain genetic transformations.

Finally, the fitness function can be given as

$$\text{Fit} = w_1\cdot\text{Fit}\,1 + w_2\cdot\text{Fit}\,2 + w_3\cdot\text{Fit}\,3 \tag{3.5}$$

Where $w_i\,(i=1,2,3)$ respectively stands for the weighted values of length, smoothness, and security degree in the fitness function.

3.2.3 Reproduction and Genetic Operators

In this approach a Tournament Selection method is used as a selection strategy (selection determines the pair of individuals chosen for recombination). Two paths are selected randomly and the fitter path will be selected as the first parent. The same process is repeated to select the second parent. After the selection process, four operators are used to evolve the selected paths. Each operator application is controlled by its probability. These operators are:

i) Crossover: This operator combines two selected paths (parents) to generate two offspring as follows: a random mating intermediate node is selected on each parent. This node split the path into two parts. The first offspring is generated by combining the first part of the first parent with the second part of the second part of the second parent, and the second offspring is generated by combing the first part of the second parent with the second part of the first parent. FIGURE 3.5 illustrates the crossover operation.

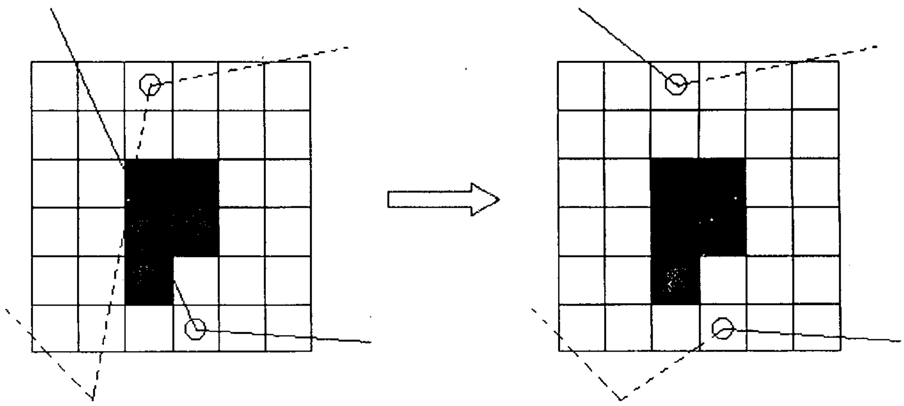

FIGURE 3.5 Crossover Operation

ii) Mutation: This mutation changes the intermediate node in the path. It randomly selects one intermediate node and changes it into another grid number. FIGURE 3.6 illustrates the mutation operation.

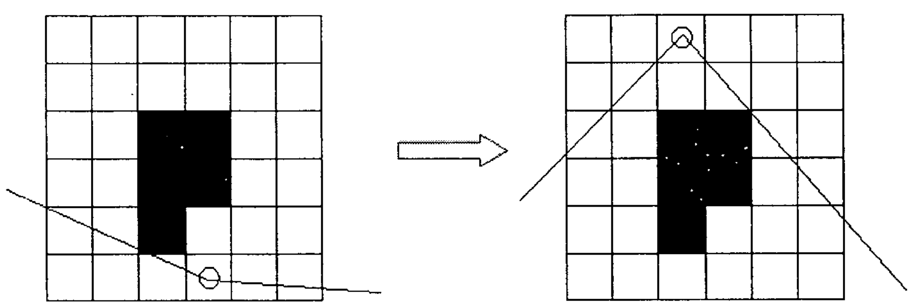

FIGURE 3.6 Mutation Operation

iii) Repair: This operator is applied to infeasible line segments. Random intermediate nodes around the intersecting obstacle are generated and the operator connects these nodes to pull the segment around the obstacle. FIGURE 3.7 illustrates the repair operation.

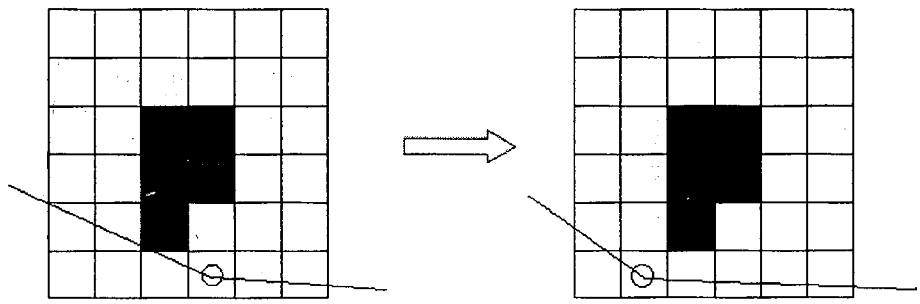

FIGURE 3.7 Repair Operation

iv) Smooth: This operator is applied to feasible paths. A path node is selected and a new node is inserted on each segment such that the segment connecting the new inserted nodes is feasible and the selected node is deleted as illustrated in FIGURE 3.8.

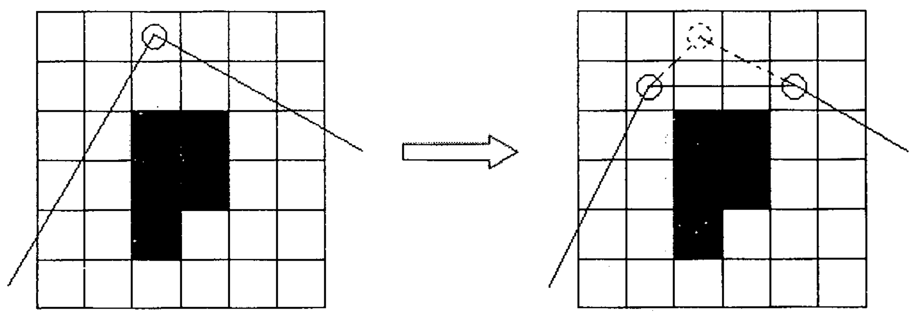

FIGURE 3.8 Smooth Operation

3.3 The Application of APF

In this dissertation, a new potential field method is proposed for path planning of a mobile robot in a dynamic environment where the target and obstacles are moving. The attractive potential is defined as a function of the relative position and velocity of the target with respect to the robot. The repulsive potential is also defined as the relative position and velocity of the robot with respect to the obstacles. Accordingly, the virtual force is defined as the negative gradient of the potential in terms of both position and velocity rather than position only.

3.3.1 Attractive Potential Function

The attractive potential field functions are presented as follows

$$U_{att}(\mathbf{p},\mathbf{v}) = \alpha_p \left\| \mathbf{p}_g(t) - \mathbf{p}_r(t) \right\|^m + \alpha_v \left\| \mathbf{v}_g(t) - \mathbf{v}_r(t) \right\|^n \tag{3.6}$$

Where $\mathbf{p}_r(t)$ and $\mathbf{p}_g(t)$ denote the positions of the robot and the goal at time t, respectively; $\mathbf{p}_r = [x\, y\, z]^T$ in a 3-dimentional space or $\mathbf{p}_r = [x\, y]^T$ in a 2-dimentional space; $\mathbf{v}_r(t)$ and $\mathbf{v}_g(t)$ denote the velocities of the robot and the goal at time t, respectively; $\left\| \mathbf{p}_g(t) - \mathbf{p}_r(t) \right\|$ is the Euclidean distance between the robot and the goal at time t; $\left\| \mathbf{v}_g(t) - \mathbf{v}_r(t) \right\|$ is the magnitude of the relative velocity between the goal and the robot at time t; α_p and α_v are scalar positive parameters; and m and n are positive constants which satisfy $m, n > 1$.

The new attractive potential $U_{att}(\mathbf{p},\mathbf{v})$ is a function of both the position $\mathbf{p}$ and velocity $\mathbf{v}$ of the robot. Therefore, we shall define the corresponding virtual attractive force as the negative gradient of the attractive potential in terms of both position and velocity,

$$\begin{aligned} \mathbf{F}_{att}(\mathbf{p},\mathbf{v}) &= -\nabla U_{att}(\mathbf{p},\mathbf{v}) \\ &= -\nabla_p U_{att}(\mathbf{p},\mathbf{v}) - \nabla_v U_{att}(\mathbf{p},\mathbf{v}) \end{aligned} \tag{3.7}$$

Where

$$\nabla_p U_{att}(\mathbf{p},\mathbf{v}) = \frac{\partial U_{att}(\mathbf{p},\mathbf{v})}{\partial \mathbf{p}} \tag{3.8}$$

$$\nabla_v U_{att}(\mathbf{p},\mathbf{v}) = \frac{\partial U_{att}(\mathbf{p},\mathbf{v})}{\partial \mathbf{v}} \tag{3.9}$$

with the subscripts p and v denoting the gradient in terms of position and velocity, respectively.

When $\mathbf{p}_r \neq \mathbf{p}_g$ and $\mathbf{v}_r \neq \mathbf{v}_g$, substituting (3.6) into (3.7), we will have

$$\mathbf{F}_{att}(\mathbf{p},\mathbf{v}) = \mathbf{F}_{att1}(\mathbf{p}) + \mathbf{F}_{att2}(\mathbf{v}) \tag{3.10}$$

Where

$$\mathbf{F}_{att1}(\mathbf{p}) = m\alpha_p \left\| \mathbf{p}_g(t) - \mathbf{p}_r(t) \right\|^{m-1} \mathbf{n}_{P\,RG} \tag{3.11}$$

$$\mathbf{F}_{att2}(\mathbf{v}) = n\alpha_v \left\| \mathbf{v}_g(t) - \mathbf{v}_r(t) \right\|^{n-1} \mathbf{n}_{V\,RG} \tag{3.12}$$

With $\mathbf{n}_{P\,RG}$ being the unit vector pointing from the robot to the goal and $\mathbf{n}_{V\,RG}$ being the unit vector denoting the relative velocity direction of the goal with respect to the robot.

The attractive force $\mathbf{F}_{att}$ consists of two components: the first component, $\mathbf{F}_{att1}(\mathbf{p})$, pull the robot to the goal and shortens the distance between them, the second component, $\mathbf{F}_{att2}(\mathbf{v})$, drives the robot to move at the same velocity of the target.

From (3.11) and (3.12), for $m > 1$ and $n > 1$, when the robot approaches the goal, $\left\| \mathbf{p}_g(t) - \mathbf{p}_r(t) \right\|$ approaches zero, $\mathbf{F}_{att1}$ approaches zero; when the velocity of the robot approaches that of the goal, $\mathbf{F}_{att2}$ approaches zero. Thus, when both of the position and velocity of the robot approach those the goal, the attractive force $\mathbf{F}_{att}$ approaches zero.

3.3.2 Repulsive Potential Function

Assume that the position $\mathbf{p}_o(t)$ and velocity $\mathbf{v}_o(t)$ of the nearest point on the obstacle to the robot can be obtained online. The relative velocity between the robot and the obstacle is given by

$$v_{RO}(t) = [\mathbf{v}_r(t) - \mathbf{v}_o(t)]^T \mathbf{n}_{RO} \tag{3.13}$$

Where $\mathbf{n}_{RO}$ is a unit vector pointing from the robot to the obstacle. If $v_{RO}(t) \le 0$, the robot is moving away from the obstacle, no avoidance motion is needed. If $v_{RO}(t) > 0$, the robot is moving close to the obstacle, avoidance motion needs to be implemented.

Assume that at time t, the robot is moving toward the obstacle. The shortest distance between the robot and the body of the obstacle is denoted by $D_s(\mathbf{p}_r(t) - \mathbf{p}_o(t))$. Accordingly, the repulsive potential can be defined as follows:

$$U_{rep}(\mathbf{p}, \mathbf{v}) = \begin{cases} 0, & \text{if } D_s(\mathbf{p}_r, \mathbf{p}_o) \ge D_0 \ or \ v_{RO} \le 0 \\ \eta\left(\dfrac{1}{D_s(\mathbf{p}_r, \mathbf{p}_o)} - \dfrac{1}{D_0}\right), & \text{if } D_s(\mathbf{p}_r, \mathbf{p}_o) < D_0 \text{ and } v_{RO} > 0 \end{cases} \tag{3.14}$$

Where U_{rep} denotes the repulsive potential generated by the obstacle; D_0 is a positive constant describing the influence range of the obstacle; and η is a positive constant.

From (3.14), we can see that when the robot is far away from the obstacle, i.e. $D_s(\mathbf{p}_r, \mathbf{p}_o) \ge D_0$, the robot is not influenced by the obstacle, and therefore no avoidance motion is implemented. When the robot is within the influence range of the obstacle and $D_s(\mathbf{p}_r, \mathbf{p}_o)$ approaches zero, the repulsive potential approaches infinity and as the projection of relative velocity of the robot v_{RO} increases, the repulsive potential also increases. Even if the distance between the robot and the obstacle does not approach zero, the repulsive potential approaches infinity if the relative velocity v_{RO} is large enough.

Similar to the definition of the new attractive force, the corresponding new repulsive force is defined as the negative gradient of the repulsive potential in terms of both position and velocity

$$\begin{aligned} F_{rep}(\mathbf{p}, \mathbf{v}) &= -\nabla U_{rep}(\mathbf{p}, \mathbf{v}) \\ &= -\nabla_p U_{rep}(\mathbf{p}, \mathbf{v}) - \nabla_v U_{rep}(\mathbf{p}, \mathbf{v}) \end{aligned} \tag{3.15}$$

To derive the virtual repulsive force, we need to derive the gradient of $v_{RO}(t)$ with respect to position and velocity, respectively. The relative velocity of the robot with respect to the obstacle in the direction from the obstacle to the robot, $v_{RO}(t)$, can be written as

$$\begin{aligned} v_{RO}(t) &= [\mathbf{v}_r(t) - \mathbf{v}_o(t)]^T \mathbf{n}_{RO} \\ &= [\mathbf{v}_r(t) - \mathbf{v}_o(t)]^T \frac{\mathbf{p}_o(t) - \mathbf{p}_r(t)}{\|\mathbf{p}_o(t) - \mathbf{p}_r(t)\|} \end{aligned} \tag{3.16}$$

The gradients of $v_{RO}(t)$ with respect to both velocity and position are given respectively as

$$\begin{aligned} \nabla_v v_{RO}(t) &= \mathbf{n}_{RO} \\ \nabla_p v_{RO}(t) &= \frac{1}{\|\mathbf{p}_o(t) - \mathbf{p}_r(t)\|} \left[v_{RO}(t)\mathbf{n}_{RO} - (\mathbf{v}_r(t) - \mathbf{v}_o(t)) \right] \end{aligned} \tag{3.17}$$

Where $v_{RO}(t)\mathbf{n}_{RO}$ gives the velocity component of $\mathbf{v}_r(t) - \mathbf{v}_o(t)$ in the direction from the robot to the obstacle.

For clarity, let $v'_{RO}(t)\mathbf{n}'_{RO}$ denote the velocity component perpendicular to $v_{RO}(t)\mathbf{n}_{RO}$ as given in the following equation

$$v'_{RO}(t)\mathbf{n}'_{RO} = \mathbf{v}_r(t) - \mathbf{v}_o(t) - v_{RO}(t)\mathbf{n}_{RO} \tag{3.18}$$

Where

$$v'_{RO}(t) = \sqrt{(\mathbf{v}_r(t) - \mathbf{v}_o(t))^2 - v^2_{RO}(t)} \tag{3.19}$$

Therefore, Equation (3.17) can be simply expressed as

$$\nabla_p v_{RO}(t) = -\frac{1}{\|\mathbf{p}_o(t) - \mathbf{p}_r(t)\|} v'_{RO}(t)\mathbf{n}'_{RO} \tag{3.20}$$

FIGURE 3.9 shows the detailed relationships among the vectors.

The virtual repulsive force generated by the obstacle is then given by

$$\mathbf{F}_{rep}(\mathbf{p},\mathbf{v})=\begin{cases}0, & \text{if } D_s(\mathbf{p}_r,\mathbf{p}_o)\geq D_0 \ or\ v_{RO}\leq 0\\ F_{rep1}+F_{rep2}, & \text{if } D_s(\mathbf{p}_r,\mathbf{p}_o)< D_0 \text{ and } v_{RO}>0\end{cases} \tag{3.21}$$

Where

$$\mathbf{F}_{rep1}=\frac{-\eta}{D_s(\mathbf{p}_r,\mathbf{p}_o)^2}\mathbf{n}_{RO} \tag{3.22}$$

And

$$\mathbf{F}_{rep2}=\frac{\eta v_{RO} v'_{RO}}{D_s(\mathbf{p}_r,\mathbf{p}_o)^2}\mathbf{n}'_{RO} \tag{3.23}$$

The relationship between the repulsive force components in a 2D space is shown in FIGURE. 3.10.

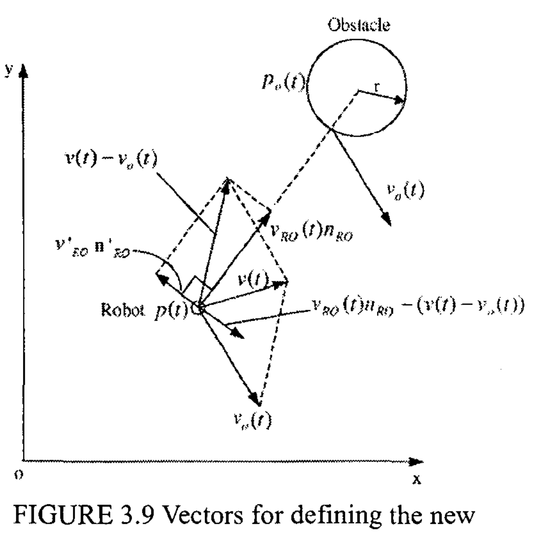

FIGURE 3.9 Vectors for defining the new repulsive potential

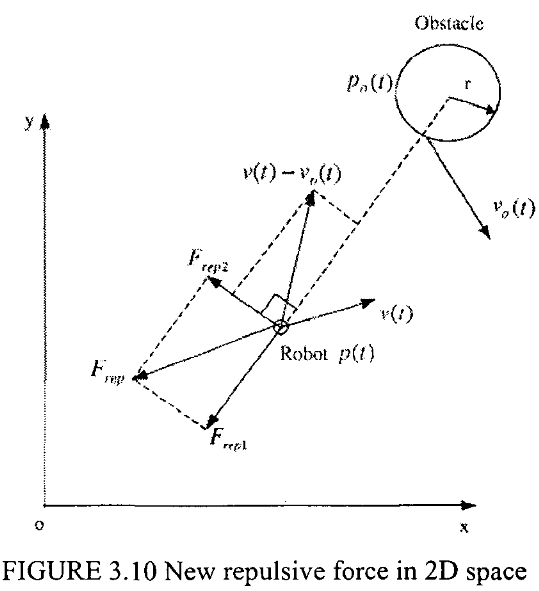

FIGURE 3.10 New repulsive force in 2D space

The repulsive force component $\mathbf{F}_{rep1}$ is in the opposite direction of $v_{RO}(t)\mathbf{n}_{RO}$ which will keep the robot away from the obstacle. The repulsive force component $\mathbf{F}_{rep2}$ is in the same direction of $v'_{RO}(t)\mathbf{n}'_{RO}$, and will drive the robot bypassing/detouring the obstacle.

After the calculation of the attractive and repulsive forces, the total virtual force can be obtained by

$$\mathbf{F}_{total} = \mathbf{F}_{att} + \mathbf{F}_{rep} \tag{3.24}$$

Where $\mathbf{F}_{att}$ and $\mathbf{F}_{rep}$ can be calculated through Equations (3.10) and (3.21). For the case where there are multiple obstacles, the repulsive force is given by

$$\mathbf{F}_{rep} = \sum_{i=1}^{n_o} \mathbf{F}_{repi} \tag{3.25}$$

Where n_o is the number of obstacles and $\mathbf{F}_{repi}$ is the repulsive force generated by the i^{th} obstacle. The total virtual force $\mathbf{F}_{total}$ will be used for local path planning.

When employing the potential functions for dynamic path planning, local minimum problems do exist and should be taken care of. To solve the problem, the simplest method is to keep the robot move according to the total virtual force as usual and wait for the obstacle or the goal to change their positions. Since the environment is highly dynamic where both the goal and the obstacles are moving, the situations where the configuration of the obstacles and goal keeps static are rare. Therefore, the waiting method is often adopted.

CHAPTER 4

IMPLEMENTATION AND EXPERIMENTAL RESULTS

Building an actual physical robot, with its complex motors, sensory system is very difficult, cost, and time-consuming, hence the attraction to bring the search for intelligent behavior inside the more-manageable world of computer. There are so many advantages of simulation:

i) Prototyping new robots in the real world is a task full of soldering irons and nuts and bolts. Moving from one iteration of a design to the next can take weeks or months. However, in the simulation, you can easily make several changes and refinements to a design in just a day or two.

ii) When the robot is running in the simulation environment, it is easy to set breakpoints on the services that control it and to find the bugs in your code. It is often difficult to have a debugger connected to a robot under motion.

iii) The simulation can be used when there is only limited hardware available. The researchers can focus on writing software and debugging the behavior of a simulated robot before testing the final version on the actual hardware.

iv) Another advantage to simulation is personal safety. It provides a way to debug the robot without risking bodily injury.

This dissertation provides a hybrid Artificial Potential Field – Gnetic Algorithm approach to mobile robot path planning in unknown environment. The environment is partially unknown, including the static obstacles and dynamic obstacles. As some obstacles are moving, we can not know their exact location, hence, the sensors installed on the robot are used to sense the moving obstacles and pass the information to the robot for obstacles avoidance.

4.1 Introduction of the Simulator - Microsoft Robotics Development Studio

Microsoft Robotics Developer Studio (MRDS) is a Windows-based environment for robot control and simulation. It is aimed at academic, hobbyist, and commercial developers and handles a wide variety of robot hardware.

MRDS is based on CCR (Concurrency and Coordination Runtime): a .NET-based concurrent library implementation for managing asynchronous parallel tasks. This technique involves using message-passing and a lightweight services-oriented runtime, DSS (Decentralized Software Services), which allows the orchestration of multiple services to achieve complex behaviors.

The features included in MRDS are a visual programming tool, Microsoft Visual Programming Language for creating and debugging robot applications, web-based and windows-based interfaces, 3D simulation (including hardware acceleration), easy access to a robot's sensors and actuators and support for a number of languages including C# and Visual Basic .NET, JScript and IronPython.

Microsoft Robotics Developer Studio includes support for packages to add other services to the suite. Those currently available include Soccer Simulation and Sumo Competition by Microsoft, and a community-developed Maze Simulator, a program to create worlds with walls that can be explored by a virtual robot.

There are four main components in MRDS:

- CCR (Concurrency and Coordination Runtime)
- DSS (Decentralized Software Services)
- VPL (Visual Programming Language)
- VSE (Visual Simulation Environment)

FIGURE 4.1 Sample apartment VSE simulation environment

FIGURE 4.2 Sample outdoor VSE simulation environment

MRDS in only available in one language, U.S. English and requires NET Framework 3.5 SP1. The Visual Simulation Environment requires a DirectX 9 graphics card that supports Pixel Shader 3.0.

4.2 Simulation System

The simulation system includes hardware and software. The hardware used in this simulation is HP laptop 6520s. And the software used in this simulation is Microsoft Robotics Development Studio (MRDS) 2008 R3. The robot selected for the simulation is Pioneer 3DX. The Pioneer 3DX Robot is manufactured by Mobile Robots, Inc. It is interesting because it has an onboard laser range finder and an onboard computer, so it is capable of autonomous movement. It also has bumpers, and the version in the simulator has a mounted webcam. FIGURE 4.3 shows the Pioneer 3DX robot.

The environment where the Pioneer 3DX moves is an indoor environment, the modern apartment, which is shown in FIGURE 4.4. At the first step, we have to extract the information such as obstacles' location from the environment and then model it into a 2D space. According to the size of Pioneer 3DX, we can model the environment into 50×33 grids. Therefore, the mobile robot can be treated as a point in the environment. FIGURE 4.5 shows the space map after modeling the environment.

FIGURE 4.3 The Pioneer 3DX

FIGURE 4.4 The 3D Simulation Environment
Modern Apartment

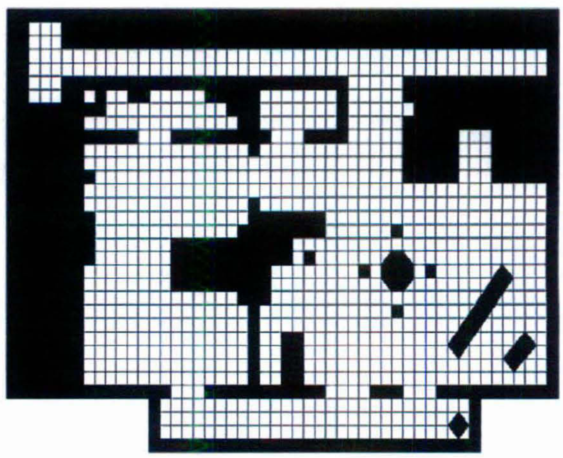

FIGURE 4.5 Space Map

4.3 Simulation Result

In this simulation, the start point is grid number 257 and the goal point is grid number 1498. The population size is 100. The crossover rate is 0.8. The mutation rate is 0.03. The maximum generation is 200. In the evaluation function discussed in Chapter 3, w_1, w_2, w_3 is set as 0.1, 0.3 and 0.6, respectively. C is set as 1000 because the penalty for infeasible path should be high. The proposed genetic algorithm can easily deal with the obstacles and obtain a near-optimal path. FIGURE 4.6 shows the evolution process. The best solution in the initial population in FIGURE 4.6(a) is not feasible. FIGURE 4.6(b) shows the best solution after 10 generation's evolution. The genetic algorithm continues to evolve better solution (FIGURE. 4.6(c)) until the maximum generation is reached. A near-optimal path is obtained after 192 generations, shown on FIGURE 4.6(d)).

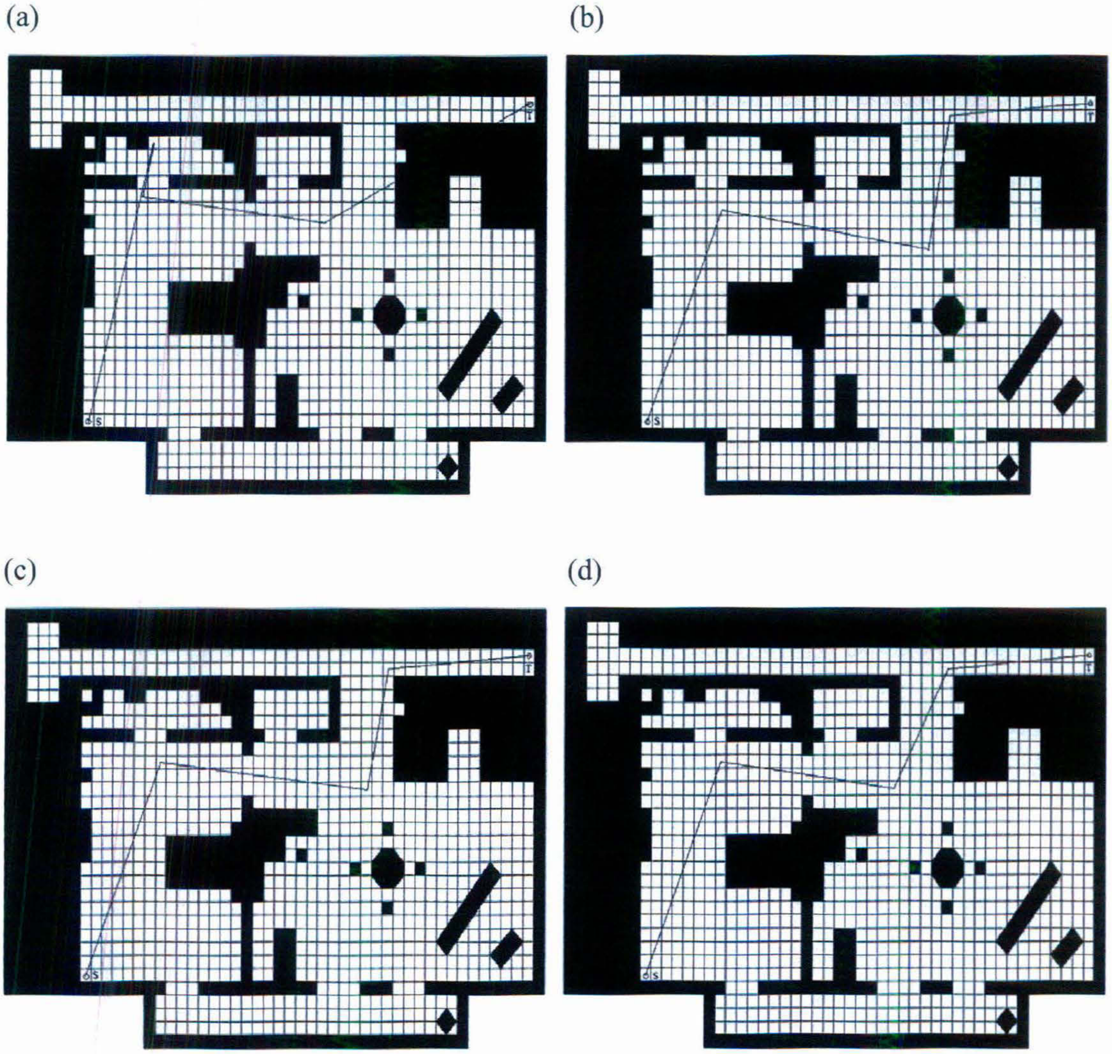

FIGURE 4.6 One typical run of path planning in the simulation environment: (a) The best initial path (Infeasible); (b) The best solution in generation 10 (feasible path); (c) The best solution in generation 164 (Improved solution); (d) The near-optimal path is obtained in generation 192.

FIGURE 4.7 The robot stops moving when it has sensed the obstacles

The near-optimal path is obtained by GA after 192 generations. Then the Artificial Potential Field approach is proposed to follow the optimal path and avoid the obstacles. As a local planning, every time the robot chooses one intermediate node as the start point and its next intermediate point as the target point. It moves from one intermediate node to next intermediate node repeatedly till the robot reaches the final point, which is the global target point. In this simulation, the APF based planner works properly. The robot tries to reach the desired target which is attractive to it. When it moves towards to the obstacles, the repulsive force prevents it colliding with the obstacles. FIGURE 4.7 shows the robot stops moving when it has sensed the obstacle.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This chapter summarizes the dissertation, and discusses the research limitations. Future research on the Hybrid Artificial Potential Field and Genetic Algorithm approach will also be discussed.

In this work we have developed and implemented a Hybrid Artificial Potential Field and Genetic Algorithm approach (chapter 3) to deal with path planning problem for mobile robots in unknown environment with static and dynamic obstacles. Compared with the genetic based approach, the hybrid approach provides a better performance in processing time, and is crucial for a robot to quickly respond to avoid collision with the dynamic obstacles.

The novelty of this work is the combination of a global planner and a local planner for robot path planning in dynamic environments with moving obstacles. The global planner is used to obtain a global optimization based on the currently known environment while the local planner is good at obstacles avoidance when following the optimal path. The hybrid approach is able to quickly determine the optimal feasible path for robot in dynamic environment.

However, there are still some research limitations in this dissertation. The limitations of this work are described below:

i) The shape of the robot is ignored in this work though the dimensions of the obstacles are considered for calculation. Therefore, taking the dimension of the robot into consideration is a future work.

ii) In this simulation, the dynamic obstacles are moving in a fixed speed and the direction is changed based on some predefined algorithms. For a more realistic modeling, making the dynamic obstacles roam with arbitrary speed and arbitrary direction should be considered.

Though the Hybrid Artificial Potential Field and Genetic Algorithm approach solves the issue of obtaining the optimal path without collision with obstacles in dynamic environments, some further extensions to be considered as future work:

i) For a more realistic modeling, a more complex and realistic environment shall be used and tested using the hybrid approach. The environment in this simulation is described in a two-dimensional surface. For modeling a more realistic environment for robot, a three-dimensional modeling needs to be considered.

ii) The hybrid approach is applied on a single robot currently. How to apply it on the multi-robot system is a more interesting future research work. And the target in this simulation is static, for further research, we can consider the target in a moving state, which is more realistic.

REFERENCES

[1] Jianping Tu & Simon X. Yang (2003), "Genetic Algorithm Based Path Planning for a Mobile Robot", *Proceeding of IEEE International Conference on Robotics & Automation*, pp.1221-1226.

[2] Yong Zhang et al. (2008), "Mobile Robot Path Planning base on the Hybrid Genetic Algorithm in Unknown Environment", *Eighth International Conference on Intelligent Systems Design and Application,* pp.661-665.

[3] S. S. Ge & Y. J. Cui (2000), "New Potential Functions for Mobile Robot Path Planning", *IEEE Transactions on Robotics and Automation*, pp.615-620.

[4] Cao Qixin et al. (2006), "An Evolutionary Artificial Potential Field Algorithm for Dynamic Path Planning of Mobile Robot", *Proceeding of IEEE International Conference on Intelligent Robots and Systems,* pp.3331–3336.

[5] Prahlad Vadakkepat et al. (2000), "Evolutionary Artificial Potential Fields and Their Application in Real Time Robot Path Planning", *Proceeding of the IEEE Congress on Evolutionary Computation,* pp.256-263.

[6] O. Khatib (1985), "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", *Proceeding of IEEE international Conference on Robotics and Automation*, pp. 500-505.

[7] Y. Koren & J. Borenstein (1991), "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", *Proceeding of IEEE international Conference on Robotics and Automation*, pp. 1398-1404.

[8] Pu Shi & Yujie Cui (2010), "Dynamic Path Planning for Mobile Robot Based on Genetic Algorithm in Unknown Environment", *2010 Chinese Control and Decision Conference*, pp. 4325-4329.

[9] Salvatore Candido (2005), "Autonomous Robot Path Planning using a Genetic Algorithm".

[10] Hui Miao (2009), "Robot Path Planning in Dynamic Environments using a Simulated Annealing Based Approach".

[11] K.S.Senthil Kumar (2006), "Hybrid Genetic-Fuzzy Approach to Autonomous Mobile Robot Navigation in Unknown Environments".

[12] M. Scott (2004), "An introduction to genetic algorithms", *Journal of Computer Sciences in Colleges*, vol. 20, pp. 115-123.

[13] C. Edelman, M. Franklin, and M. White (1988), "Simulated Annealing on a Multiprocessor", *Proceedings of the 1988 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 540-544.

[14] F. Lingelbach (2004), "Path planning for mobile manipulation using probabilistic cell decomposition", *Intelligent Robots and Systems*, vol 3, pp. 2807-2812.

[15] J. Chestnutt and M. Lau (2005), "Footstep Planning for the Honda ASIMO Humanoid", *IEEE International Conference on Robotics and Automation*, pp. 629-634.

[16] Y. Wang, P. Sillitoe, J. Mulvaney (2007), "Mobile Robot Path Planning in Dynamic Environments", *IEEE International Conference on Robotics and Automation*, pp. 71-76.

[17] S. Lbszlo and K. Annamiria (2003), "Autonomous navigation in a known dynamic environment", *The 12th IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 266-271.

[18] Simon Kent (1999), "Evolutionary Approaches to Robot Path Planning".

[19] J. Cook (1992), "Adding Intelligence to Robot Arm Path Planning Using a Graph-Match Analogical Reasoning System", *IEEE International Conference on Intelligent Robots and Systems*, vol. 1, pp. 657-663.

[20] J. Ayers (2004), "Underwater Walking", *Arthropod Structure & Development*, vol. 33, pp. 347-360.

[21] Tom M. Mitchell (1997), "Machine Learning", *McGraw-Hill Companies Inc.*.

[22] B. Williams and I. Mahon (2004), "Design of an Unmanned Underwater Vehicle for Reef Surveying", *Proceeding of IFAC 3rd Sysmposium on Mechatronic Systems*.

[23] J. Xiao, Z. Michalewicz, L. Zhang and K. Trojanowski (1997), "Adaptive Evolutionary Planner/Navigator for Mobile Robots", *IEEE Transactions on Evolutionary Computation*, vol. 1, pp.18-28.

[24] Yanrong Hu and Simon X. Yang (2004), "A Knowledge Based Genetic Algorithm for Path Planning of a Mobile Robot", *Proceeding of the 2004 IEEE International Conference on Robotics & Automation*, pp. 4350-4355.

[25] Ahmed Elshamli, Hussein A.Abdullah and Shawki Areibi (2004), "Genetic Algorithm for Dynamic Path Planning", *CCECE 2004 – CCGEI 2004, Niagara Falls.*

[26] Y. Davidor (1991), "Genetic algorithm and robotics: a heuristic strategy for optimization", *Singapore: World Scientific Publishing*, vol. 1, pp.220-225.

[27] A. Nearchou (1998), "Path planning of a mobile robot using genetic heuristics", *Robotica*, vol.16, pp.575-588.

[28] J. Cai, M. Peng and S. Ma (2006), "RL-ART2 Neural Network Based Mobile Robot Path Planning", Proceeding of the IEEE Sixth International Conference on Intelligent System Design and Applications, vol.2, pp.581-585.