A Generic Force Field Method for Robot Real-time Motion Planning and Coordination

Dalong Wang

A thesis submitted in fulfilment
of the degree of
DOCTOR OF PHILOSOPHY

Faculty of Engineering and Information Technology
University of Technology, Sydney

October 2009

CERTIFICATE OF AUTHORSHIP / ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

_____

(DALONG WANG)

Sydney, October 2009

Abstract

This thesis presents a systematic study on a novel force field method (F^2) for robot motion planning and multi-robot motion coordination. In this F^2 method, a force field is generated for each robot based on its status: location, orientation, travel speed, priority, size, and the robot's environment. A robot with larger volume, travelling at higher speed or with higher task priority than other robots, will have a larger force field, and consequently has priority in collision avoidance. The interaction of a robot's force field with its environment provides a natural way for real-time motion planning and multi-robot coordination.

Four novel F^2 based methods have been investigated for applications in different cases. The Canonical Force Field method (CF^2) is first designed based on the concept of the F^2 method, in which a robot is assumed to be travelling with constant speed and its moving direction is determined by the resultant forces acting on it. This CF^2 method has proved to be very efficient in applications in simple and structured environments. A Variable Speed Force Field method (VSF^2) which takes a robot's kinematic and dynamic constraints into consideration is further investigated. The VSF^2 method allows a robot to change its speed based on environmental information and the status of obstacles and other robots in the same environment. A Subgoal-Guided Force Field method (SGF^2) is developed to enhance the F^2 method by generating subgoals based on updated sensor data. A robot using the SGF^2 method will then move towards a subgoal instead of the global goal, which greatly broadens the applicability of the F^2 method in more complex environments. Finally, a Dynamic Variable Speed Force Field method ($DVSF^2$) is designed for applications in partially known and dynamically changing environments. In this method, subgoals are selected on a pre-planned global path.

In order to investigate the effect of parameters on the performance of the proposed F^2 methods, two optimization algorithms have been proposed in this research for optimal design of the parameters in F^2 methods: the Particle Swarm Optimization-tuned Force Field method (PSO-tuned F^2) for single objective parameter optimization and the Ranked Pareto Particle Swarm Optimization approach for multiobjective parameter optimization.

Extensive simulations and experiments with real robots in an indoor environment have been carried out to verify these methods. The results have demonstrated the feasibility and efficiency of the F^2 methods in real-time robot motion planning and multi-robot coordination in various environments.

Acknowledgements

I would like to express my deepest gratitude to the following people for helping me during my study in the ARC Centre of Excellence for Autonomous Systems (CAS), Faculty of Engineering and Information Technology, University of Technology, Sydney. First of all, I would like to thank my principal supervisor Associate Professor Dikai Liu for his advice and guidance throughout these years. Without his support the completion of this thesis would not have been possible. I would also like to thank Professor Gamini Dissanayake for advising me on my research. His expert knowledge and deep insight provide invaluable help to my research.

I would like to thank all friends and colleagues in CAS, especially Gavin Paul, Nathan Kirchner, Zhengzhi Zhang, Pholchai Chotiprayanakul, Tianran Ren, Tarek Taha, and Stephen Webb for their friendship and help. I would like to thank Mr. Tarek Taha and Dr. Jaime Valls Miro for their great support in experiments with robots. A special thank goes to Dr. Ngai Ming Kwok, who helped a lot in my research on parameter optimization of the force field methods and gave me many helpful suggestions on the writing of my thesis.

Finally, I would like to thank my family who has always supported me and my wife Danna for always being by my side. You are the source of my inspiration, courage and happiness.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Robot applications are increasingly being used in a variety of environments. Examples include home cleaning robots [1, 2], automated wheelchairs to assist the handicapped or elderly people [3, 4], museum-guide robots in museums and exhibitions [5-7], autonomous straddle carriers in container handling [8, 9], driverless cars in the DARPA Grand Challenge [10, 11], and so on. Figure 1-1 depicts five examples of robotic systems. Figure 1-1 (a) shows a home cleaning robot which is designed to vacuum dirt from carpets and hard floors, the robot in (b) is a wheelchair platform developed at UTS to assist people with reduced mobility, (c) shows a mobile robot used as an interactive museum guider, (d) shows a fully autonomous vehicle from Stanford University in the DARPA Grand Challenge 2006, and (e) shows an autonomous straddle carrier for handling containers in an automated container terminal located in Brisbane, Australia. A basic requirement of fulfilling their tasks is that a robot must be able to move from its start point to destination and avoid possible collisions. This raises the problem of motion planning, which can be described as the construction of a collision-free trajectory that connects a robot to its destination [12]. Being a key challenge of robotics and automation engineering, motion planning has been studied extensively in past decades and a variety of approaches has been presented [12, 13].

Currently there is no single motion planning approach which is suitable for all applications. There are many reasons for this. Firstly, robots vary in their physical properties and kinematic and dynamic characteristics, such as size, mass, speed and acceleration abilities. For example, an autonomous straddle carrier in container handling weighs 65 tonnes and is 10 metres high, 3.5 metres wide and 9 metres long [9]. The straddle carrier may travel at a speed up to 10 m/s. An Amigorot robot for education and research weighs 3.6 kilograms and is 33 cms long, 28 cms wide and 15 cms high [14]. The maximum speed for an Amigorot is only 1 m/s. Thus, a robot's physical properties, kinematic characteristics and dynamic characteristics must be taken into account in motion planning.

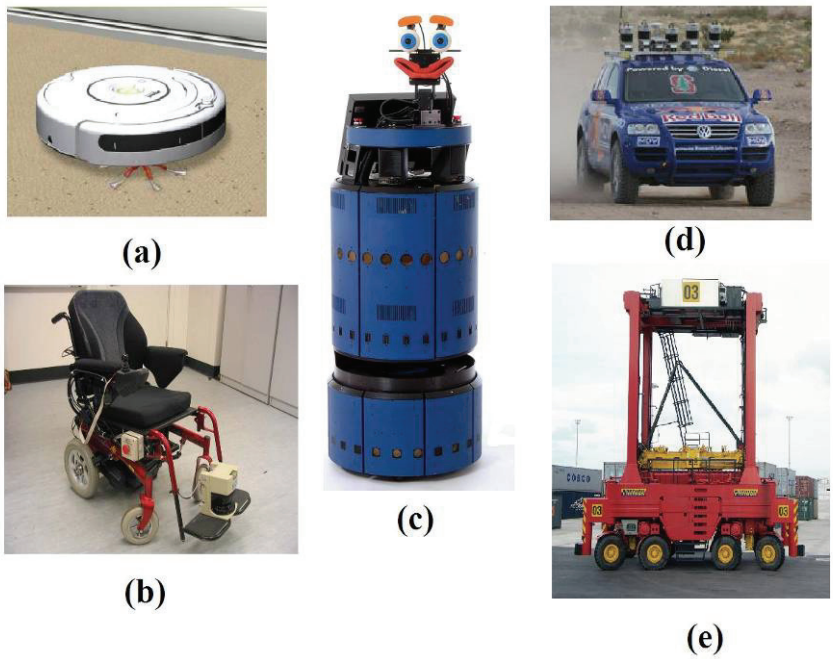


Figure 1-1 Various types of robots: (a) the *irobot* cleaning robot, (b) a wheelchair platform developed in UTS, (c) a museum guide robot, (d) *Stanley* from Stanford University in the DARPA Grand Challenge 2006, (e) an autonomous straddle carrier

Secondly, a robot's knowledge of its working environment varies. In some applications, a robot's working environment can be assumed to be completely known, for example, a robot arm working in an automotive assembly line. In many other cases, robots need to work in partially known or dynamically changing environments. For example, a vacuum cleaning robot needs to work in a room where people are walking around and furniture may be moved frequently. A museum guide robot has to navigate safely in an environment where many people are moving at the same time.

Thirdly, approaches to single robot motion planning may not be transferred directly to multi-robot motion planning and coordination. A team of robots is often utilized to accomplish complex tasks such as coordinated material handling [8, 15], the exploration of unknown terrain [16-18] or robot soccer [19, 20]. When there are many robots operating in the same environment, their motions have to be coordinated to avoid congestion and collisions, which creates the requirement to develop multi-robot cooperation methodologies, including multi-robot task allocation, multi-robot localization and real-time multi-robot motion planning and

collision avoidance. Planning the paths for a team of mobile robots is significantly more complex than single robot motion planning.

1.1 Robot Motion Planning Algorithms

1.1.1 Motion Planning Approaches

Approaches developed for mobile robot motion planning may be broadly divided into three major categories: roadmap-based methods, cell decomposition-based methods and potential field-based methods [12].

A. Roadmap Approaches

A roadmap approach captures the connectivity of a robot's free space in a network of 1-dimensional (1D) curves or lines, which is called a *roadmap*. Once the network is constructed, a path can be extracted by connecting the start and goal positions in the roadmap. A graph search algorithm, such as Dijkstra's algorithm [21] or A* search algorithm [22, 23], is normally used to find the best path to take the robot from the start position to the goal position. Well-known roadmaps methods include Visibility graph [24], Voronoi diagram [25-28], probabilistic roadmap (PRM) [29-34] and rapidly-exploring random trees (RRT) [35-37].

B. Cell Decomposition Approaches

In a cell decomposition approach, a robot's free space is decomposed into a set of non-overlapping cells, and the adjacency relationships among the cells are computed. A collision-free path between the start and the goal of a robot is found by first identifying the two cells containing the start position and the goal position and then connecting them with a sequence of connected cells. Cell decomposition approaches include exact decomposition methods [38] and approximate decomposition methods [39-41]. In exact decomposition methods, the set of cells

covers the free space exactly, which may bring complicated cells with irregular boundaries (contact constraints) which are hard to compute. On the other hand, an approximate cell decomposition method generates a set of cells which covers free space approximately. This leads to simpler cells with regular boundaries and is easier to compute [39, 40].

C. Potential Field-based Approaches

The basic concept of the Artificial Potential Field Method (often referred to as APM or PFM) is to fill a working environment with an artificial potential field in which the robot is attracted by the goal and repulsed by obstacles [42]. Researchers have developed a variety of methods based on the concept of potential field. For example, Connolly presented a method using Laplace's Equation to avoid the existence of local minima [43]. Ge and Cui developed a potential field method which defines attractive and repulsive potentials by taking account of the relative position and velocity of a robot with respect to obstacles and targets [44]. A potential field model using generalized sigmoid functions is proposed in [45] to meet the requirement of accurate representation of objects with complex geometry in applying the artificial potential field in some practical applications. Methods based on the concept of potential field have also been widely used in real-time path planning and collision avoidance for manipulators [46-49] and multi-robot systems [50-54].

1.1.2 Force Field Related Work

Some researchers have investigated methods for constructing a kind of *Safe Zone* or *Free Zone* to protect a robot from possible collisions with obstacles and other robots. Masoud proposed a repulsive field which is strictly localized in a robot's vicinity to protect it from collision, in which the repulsive field is generated as the gradient flow of a spherically symmetric potential field [51]. Seraji and Bon proposed an approach for real-time manipulator collision avoidance, in which a *Safe Zone* is defined for each obstacle [55]. When a manipulator enters this safe zone, it will suffer from virtual intrusion force, which is defined as a virtual spring-dumper model and increases as the manipulator moves towards an obstacle. This virtual force will push the manipulator out of the safe zone.

The *Elastic Band* method tries to combine global path planning with real-time sensor-based collision avoidance [56]. In this approach, a pre-planned global path is deformed in real-time to keep a robot away from obstacles during its movement, while the internal contraction forces will bring the robot back to its original path when the obstacle is out of the sensor range. This method also takes into account the robot geometry and restricts the search space by the concept of a *bubble*, which is defined as the maximum local subset of the free space around a given configuration of the robot which can be safely travelled in any direction without collisions. Given such bubbles, a band or string of bubbles can be used along the trajectory from the robot's initial position to its goal position to show the robot's expected free spaces along the pre-planned path.

Since the Elastic Band method was found to be inefficient for robots with high degrees of freedom, such as 6DOF industrial manipulators, the concepts of *Protective Hull* and *Elastic Tunnel* were proposed in [57]. A *Protective Hull* is a description of workspace volume containing the robot. An *Elastic Tunnel* is a set of overlapping protective hulls placed along a pre-calculated path. Thus the robot is protected by the elastic tunnel during trajectory execution. Like the *Elastic Band*, the *Elastic Tunnel* deforms automatically to adapt to the environment.

A robot is represented with the composition of elastic elements in [58]. The interactions between elastic elements - reaction force - help to avoid self-collision, for example, collision between a robot arm and its body. The direction of this virtual reaction force is on the line through the centres of two elastic elements. The magnitude is determined by the distance between two elements.

Some approaches take a robot's kinematic/dynamic constraints into consideration. One of them is the Curvature-Velocity Method (CVM), in which constraints derived from physical limitations on a robot's velocities and accelerations, and from sensor data that indicate the presence of obstacles, are taken into consideration in motion planning [59]. The robot then chooses velocity commands that satisfy all constraints and maximize an objective function. Another popular method is the Dynamic Window Approach (DWA), in which kinematic

constraints are considered by directly searching the velocity space of a synchro-drive robot [60]. Brock and Khatib extended the Dynamic Window Approach to the Global Dynamic Window Approach (GDWA), which is applicable to both nonholonomic and holonomic mobile robots and is suitable for unknown and changing environments [61]. By taking dynamic constraints into consideration, the Curvature-Velocity Method and Dynamic Window Approach reduce the search space greatly.

The family of Vector Field Histogram (VFH) techniques also addresses some kinematic/dynamic constraints. The VFH method looks for gaps between the obstacles in front of the robot and builds a local map based on the concept of a certainty grid from recent sensor range readings [62]. A variation of the original VFH, the VFH+, first comes up with a simplified model of the moving robot's possible trajectories based on its kinematics constraints. Obstacles which block the robot's allowable trajectories are then properly taken into account in a polar histogram [63]. VFH* introduced the global A* search into the direction determination and has been proved to obtain better solutions than VFH+ in some cases [64].

In this research work, a novel concept of force field (F^2) is presented in detail, which is a generic approach for robot motion planning and coordination. Several approaches are then developed based on the concept of F^2 for various applications. The F^2 method is not only an efficient way for real-time motion planning and collision avoidance for a single robot in a partially known and dynamic changing environment, but is also suitable for multi-robot real-time motion planning and coordination.

1.1.3 Virtual Force Field Method for Real-time Motion Planning and Coordination

In the F^2 method, a virtual force field is generated for each robot in its vicinity and is continuously changing based on its status, including its size, travelling speed, priority with respect to other robots and environment factors. The force field varies with this robot's status when it travels in an environment. If there are obstacles or other robots in the area of a robot's force field, this robot will be acted on by virtual repulsive forces from them and be repelled. A robot with larger volume, travelling at higher speed or with higher priority, will have priority

in collision avoidance. The interaction of a robot's force field with its environment provides a natural way for real-time motion planning and collision avoidance. In the F^2 method, a robot is attracted by a force from a selected target point. This target point can be its final destination or a temporary subgoal which is generated based on local sensor data or by other external global planners. This research focuses on the theoretical developments and experimental studies of the F^2 method on mobile robots in 2-dimensional (2D) environments, but it is also applicable to manipulator motion planning problems in 3-dimensional (3D) environments.

The concept of F^2 resembles the Potential Field to some extent. Both concepts use repulsive potential/force fields to avoid collision with obstacles and an attractive potential/force to guide a robot to its target. However the differences between the F^2 and the Potential Field are distinct. A potential field is generated based on environment information. That is to say, the potential value of a point in potential field is determined by its location in the environment. This potential field remains unchanged if the environment does not change. In the F^2 method, the repulsive force field of a robot covers the robot body, instead of being around an obstacle, as those of potential field-based approaches. This force field is continuously changing during the robot's movement based on its own status. Collision avoidance is achieved by the interaction of a robot's repulsive force field with its environment.

Compared with currently existing approaches, the F^2 method has the following desirable features:

- In the F^2 method, a robot's physical characteristics, such as size and geometry, are used in the construction of its force field. Its dynamic and kinematics characteristics, such as constraints on linear velocity and angular velocity, are taken into consideration when determining a robot's motion. The F^2 method is a generic approach for any kind of mobile robot and suitable for real applications.
- The F^2 method is suitable for applications in partially known or dynamically changing environments. A robot using the F^2 method needs to know its location and destination in its movement but a precise map is not essential. If there are environmental changes or moving obstacles in the work space, a robot reacts immediately based on information

obtained from inter-robot communication and sensor data. No preplanning and replanning is needed.

- The F^2 method is suitable for motion planning and collaboration of multiple robots working in a decentralized manner. A robot plans its path and motion independently according to the surrounding environment and its own status, so the F^2 method will not suffer from the exponentially increasing computation burden, as do some centralized approaches, and can be used online. Another advantage of the F^2 method is that the task priority is taken into account in the construction of the force field.
- In the F^2 method, a robot only reacts to obstacles which are in the coverage of its own force field. A robot using the F^2 method does not therefore need to search the whole work space as many other methods do, which significantly increases the efficiency of motion planning and coordination.

1.2 Scope and Objectives

The problem of mobile robot motion planning and collaboration is addressed in this research. It is assumed that robots move in a 2D space and each robot is aware of its current location and goal position. Robots are equipped with communication devices so that they are aware of the status of other robots, including priorities, velocities, locations, sizes and geometries. Robots are capable of using onboard sensors to sense their vicinities and obstacles. The objectives of this research are:

- To carry out a systematic study of the F^2 method.
- To investigate algorithms for applications of the F^2 method in various scenarios.
- To test the F^2 method with real robot experiments.

The proposed approaches should have the following merits:

- Applicable in real-world path planning scenarios, by taking a robot's dynamic and kinematics characteristics into consideration, so that it is able to react to the environment.
- Capable of reacting to environmental changes based on updated information, so that the proposed approaches are applicable to partially known or dynamically changing environments.
- A robot plans its motion independently based on its own status and received information, so that the proposed approaches are suitable for use in both single robot and multi-robot cases.
- Mathematically simple, computationally efficient and suitable for real-time applications.

1.3 Contributions

The major contributions of this research are

- A systematic investigation on a novel force field (F^2).
- The development of F^2 based algorithms for applications in various scenarios.
- Parameter optimization approaches on the F^2 method for motion planning and coordination.
- Experimental studies and verification of the F^2 method.

1.4 Publications Associated with This Research

Parts of the research work have been published in the following papers [65-76]:

Journal article

1. Jaime Valls Miró, Tarek Taha, Dalong Wang and Gamini Dissanayake (2008), "An adaptive manoeuvring strategy for mobile robots in cluttered dynamic environments", *International Journal of Automation and Control*, vol. 2, Nos. 2/3, 2008, pp. 178-194.

Book chapter

2. D. Wang, N. M. Kwok, D. K. Liu and Q. P. Ha (2009), "Ranked Pareto Particle Swarm Optimization for Mobile Robot Motion Planning", in *Design and Control of Intelligent Robotic Systems*, Berlin Heidelberg: Springer-Verlag, 2009, pp. 97-118.

Peer reviewed conference papers

3. D. Wang, D. K. Liu, N. M. Kwok and K. J. Waldron (2008), "A subgoal-guided force field method for robot navigation", *Proceedings of the 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA08)*, Beijing, China, pp. 488-494.
4. Matthew Clifton, Gavin Paul, Ngai Kwok, Dikai Liu and Dalong Wang (2008), "Evaluating performance of Multiple RRTs", *Proceedings of the 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA08)*, Beijing, China, pp. 564-569.
5. P. Chotiprayanakul, D. Wang, N.M. Kwok, D.K. Liu (2008), "A Haptic Based Human Robot Interaction Approach for Robotic Grit Blasting", *Proceedings of the 25th International Symposium on Automation and Robotics in Construction (ISARC 2008)*, 26-29 June 2008, Vilnius, Lithuania, pp.148-154
6. Jaime Valls Miró, Tarek Taha, Dalong Wang, Gamini Dissanayake and Dikai Liu (2007), "An efficient strategy for robot navigation in cluttered environments in the

presence of dynamic obstacles", *Proceedings of the Eighth International Conference on Intelligent Technologies (InTech 07)*, 12-14 December 2007, Sydney, Australia, pp. 74-81.

7. D. Wang, N. M. Kwok, D. K. Liu and G. Dissanayake (2007), "PSO-tuned F^2 method for multi-robot navigation", *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 07)*, 29 October-2 November 2007, San Diego, California, USA, pp. 3765-3770.

8. P. Chotiprayanakul, D.K. Liu, D. Wang and G. Dissanayake (2007), "Collision-free trajectory planning for manipulators using virtual force based approach", *Proceedings of the International Conference on Engineering, Applied Sciences, and Technology (ICEAST 2007)*, 21-23 November 2007, Swissôtel Le Concorde, Bangkok, Thailand.

9. P. Chotiprayanakul, D. K. Liu, D. Wang and G. Dissanayake (2007), "A 3-dimensional force field method for robot collision avoidance in complex environments", *Proceedings of the 24th International Symposium on Automation and Robotics in Construction (ISARC 2007)*, 19-21 September 2007, Kochi, Kerala, India, pp. 139-145.

10. D. Wang, D. K. Liu and G. Dissanayake (2006), "A variable speed force field method for multi-robot collaboration", *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 06)*, 9-15 October 2006, Beijing, China, pp. 2697-2702.

11. D. K. Liu, D. Wang and G. Dissanayake (2006), "A force field method based multi-robot collaboration", *Proceedings of the IEEE International Conference on Robotics, Automation & Mechatronics (RAM 06)*, 7-9 June 2006, Bangkok, Thailand, pp. 662-667.

12. D. L. Wang, D. K. Liu, X. Wu and K. C. Tan (2005), "A force field method for robot navigation", *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 05)*, 2005.

1.5 Thesis Outline

In Chapter 2, earlier research works related to this research are reviewed in detail. The discussion starts from approaches for single robot motion planning and collision avoidance, including the Potential Field Method (PFM) and its varieties, to the Vector Field Histogram (VFH) and its varieties and the Dynamic Window Approach (DWA). Other works on multi-robot robot motion planning and collision avoidance are then reviewed.

In Chapter 3, the basic concepts of the F^2 method, including the definition of force field, attractive force and repulsive force, are presented in detail. The Canonical Force Field method (CF^2) is then proposed for robot real-time motion planning and collision avoidance. In this method, a robot is assumed to be travelling at a constant speed and its moving direction is determined by the direction of the resultant force. The CF^2 method is especially suitable for robots with limited computation and motion control capabilities.

Chapter 4 describes further developments on the F^2 method. Three F^2 based algorithms are proposed for various applications. The Variable Speed Force Field method (VSF^2) takes a robot's dynamic and kinematic characteristics into consideration and enhances the performance of the F^2 method by changing the robot speed based on environment information. The Subgoal-Guided Force Field method (SGF^2) improves the F^2 method by combining the concept of a subgoal with the F^2 method. In the SGF^2 method, a robot generates subgoals continuously based on its sensor data and a selected subgoal is then used as temporary guidance when the global goal is not in the field of view. The SGF^2 method is especially suitable for real-time motion planning and collision avoidance in partially known and dynamically changing environments. The third algorithm in Chapter 4 is the Dynamic Variable Speed Force Field method ($DVSF^2$), in which a temporary waypoint is selected from a pre-planned global path. The $DVSF^2$ method is suitable for acting as a real-time collision avoidance component in a navigation framework for real-time motion planning and collision avoidance in dynamically changing environments.

Chapter 5 focuses on parameter optimizations for the F^2 method. It has been proved that the setting of parameters in the F^2 method noticeably affects its performance, which creates the optimization problem of finding appropriate parameters for the F^2 method. A Particle Swarm Optimization method (PSO) is utilized to solve single objective optimization problems for the F^2 method. A novel Ranked Pareto Particle Swarm Optimization approach (RPPSO) is then proposed to tackle the multiobjective optimization problem in parameter optimization. This approach has been successfully utilized in multiobjective optimization problems using the F^2 method for motion planning and coordination.

Chapter 6 presents experimental studies. To prove the feasibility of the F^2 method, experiments are carried out with real robots in various environments. This chapter reports the experiments with a Pioneer robot for verifying the Canonical Force Field method (CF^2), the Variable Speed Force Field Method (VSF^2) and the Subgoal-Guided Force Field method (SGF^2). Simulations on multiple robots cases are carried out in the Player/Stage platform.

Chapter 7 concludes this thesis and suggests some directions for future work.

Chapter 2
Literature Review

This chapter provides a literature review of previous works on robot motion planning and collision avoidance. The robot motion planning and collision avoidance problem has been extensively studied in the past three decades and a variety of approaches has been developed [12]. Discussions in this chapter are limited to approaches which are closely related to this research. This chapter is organized as follows: Section 2.1 introduces typical approaches for single robot motion planning and collision avoidance; Section 2.2 reviews important issues on multi-robot motion planning and collaboration.

2.1 Single Robot Motion Planning Approaches

2.1.1 Potential Field Method and Its Varieties

Artificial Potential Field Method (often referred to as APF or PFM) has been a very popular approach in path planning and obstacle avoidance. It was first proposed by Khatib [42], and many researchers subsequently developed a variety of methods based on the concept of potential field [43, 46, 50, 56, 57, 77-86].

2.1.1.1 Potential Field Method

The Potential Field method (PFM) is a popular approach for robot path planning. In the PFM, a robot is treated as a point under the influence of a potential field. This robot is attracted by the goal, which is a global minimum in the field, and repulsed by the environmental obstacles, which are represented by peaks in the potential field. This process can be compared to a ball rolling down a hill [42]. A typical potential field function presented by Khitib [42] is given below:

$$U_o(x)=\begin{cases}\frac{1}{2}\eta\left(\frac{1}{\rho}-\frac{1}{\rho_0}\right)^2 & if\ \rho\le\rho_0\\ 0 & if\ \rho>\rho_0\end{cases} \tag{2-1}$$

where ρ represents the shortest distance between the robot and the obstacle and ρ_0 denotes the influence distance of this potential field. η is a constant which determines the magnitude of repulsive potential.

Figure 2-1 illustrates the potential field in a simple case. In Figure 2-1, (a) denotes a goal position and the locations of two obstacles, (b) shows the attractive potential generated by the goal, (c) shows the repulsive potentials generated by two obstacles, and (d) gives the combined potential field.

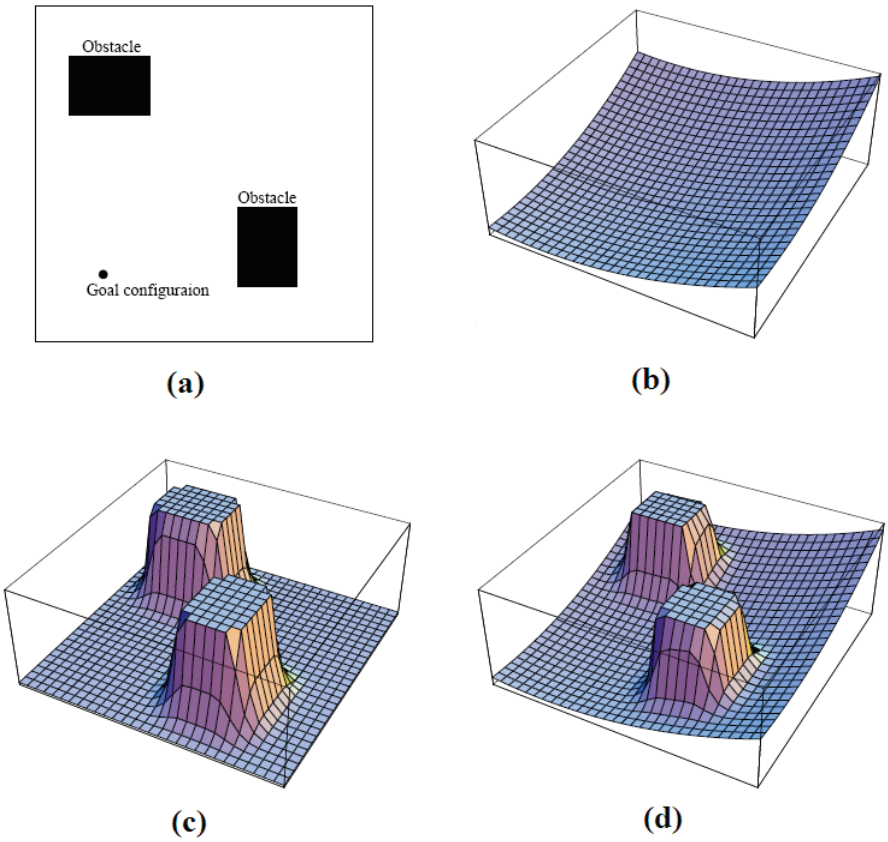


Figure 2-1 An example of potential field [87]

The main advantage of the PFM is its mathematical simplicity and efficiency. The drawback of such methods is that they are usually susceptible to local minima. An example of local minimum is illustrated in Figure 2-2. The repulsive force $\mathbf{f_r}$ from the obstacle and attractive force $\mathbf{f_a}$ from the goal point are opposing each other. Thus, the robot in the potential field cannot escape from this 'potential trap'.

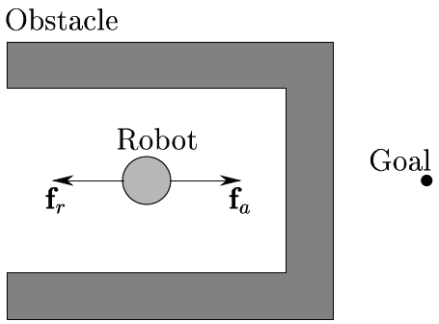


Figure 2-2 An example of local minima [87]

2.1.1.2 Elastic Band

The Elastic Band method was proposed by Quinlan and Khatib in [56]. This method tries to combine global path planning with real-time sensor-based collision avoidance. In this method, a path is first generated by a global planner. This path will be used as the original elastic band in the algorithm. If unexpected obstacles are found by sensors during the movement of a robot, this elastic band will deform to keep the robot away from the obstacles due to repulsive forces from the obstacles. After this robot passes the obstacles, the internal contraction force will bring the robot back to its original path. The Elastic Band method provides a feasible solution to reacting in real-time environment changes while preserving the global nature of the planned path [56]. More research on the Elastic Band approach can be found in [88-92].

Figure 2-3 explains how the Elastic Band method works [56]. First, a path is generated by a global planner as shown in (a). This path is used as the original elastic band. Then in (b) the composition of external repulsive forces from the obstacles and internal contraction force of the elastic band make the path smoother than the former one. When another obstacle is

detected by sensors as in (c), the elastic band continues to deform to keep away from the obstacle.

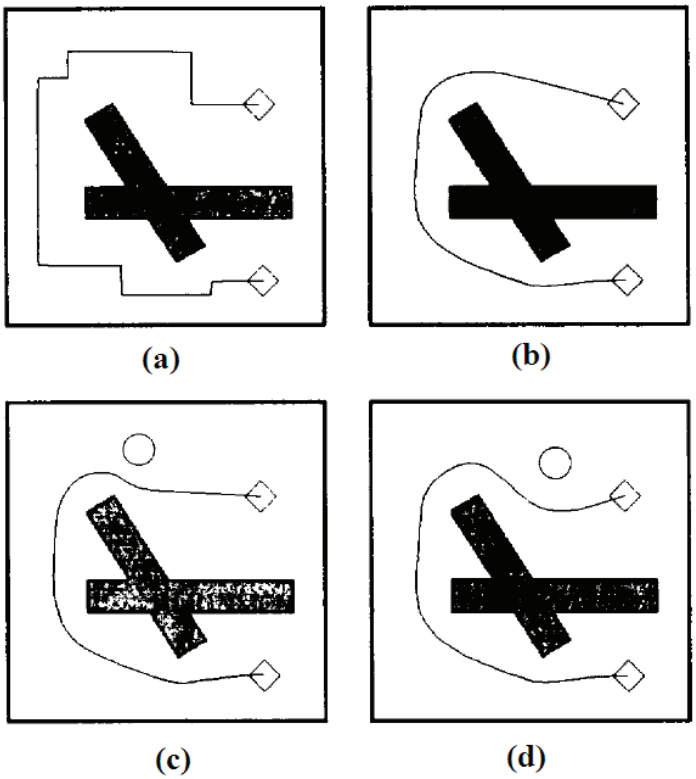


Figure 2-3 Elastic band: (a) a path is pre-planned by a planner, (b) the repulsive forces from obstacles and internal contraction force make the path smoother, (c) when an obstacle is found, the elastic band deforms to avoid collision, (d) the elastic band continues to deform as the obstacle moves [56]

Another important concept in this method is the 'bubble'. A bubble is defined as maximum sublets of the free space around a given configuration of the robot which can travel in any direction without collision [56]. The elastic band can be represented by a series of bubbles. As long as the path remains in these bubbles, it will be collision-free (see Figure 2-4).

One advantage of the bubble representation of the elastic band is that the complexity of representation is related to the complexity of environment. When a robot is travelling in a large

free space, the bubbles will grow bigger. By contrast, when this robot is in an obstacle cluttered environment, bubbles will become smaller.

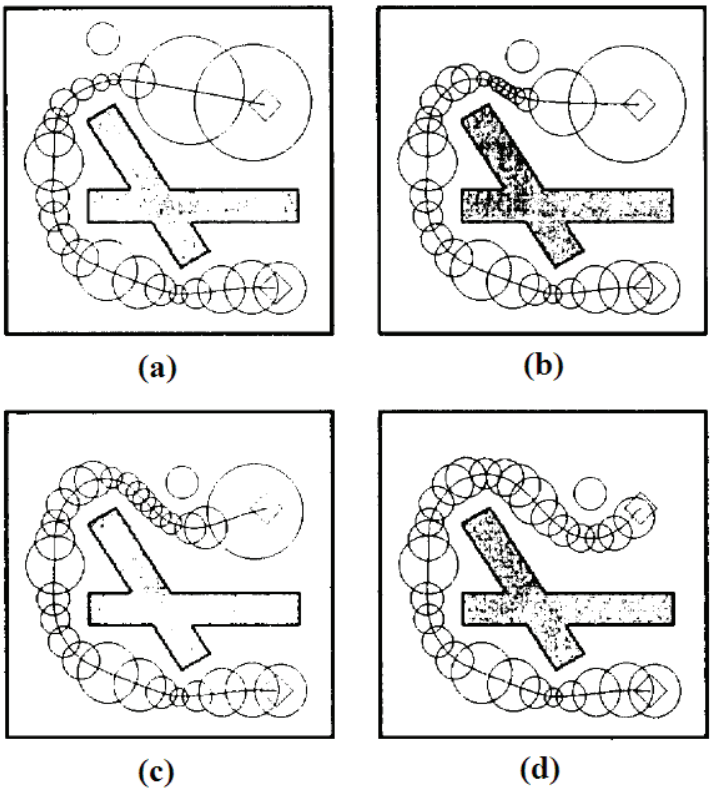


Figure 2-4 Bubbles in elastic band: as long as the path is in the bubble sets, it is collision-free. Bubbles are updated in real-time and their sizes vary with the environment [56].

2.1.1.3 Elastic Strip

The concept of Elastic Strip is similar to that of the Elastic Band method. The Elastic Band was found to be inefficient for robots with high degrees of freedom, such as 6-axis robotic arms [87]. To overcome this problem, the free space in Elastic Strip is represented by a robot's workspace volume. This brings up the concepts of Protective Hull and Elastic Tunnel. The Protective Hull is a volume description of the workspace containing the robot but having no obstacles within (see Figure 2-5). An Elastic Tunnel is a set of overlapping protective hulls placed along a pre-calculated path. Thus the robot is protected by the elastic tunnel during trajectory execution [87]. Like the elastic band, the elastic tunnel deforms automatically to

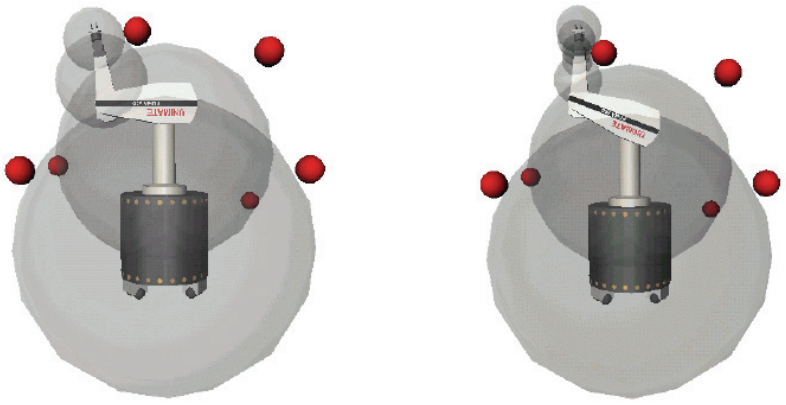
adapt to environment changes (see Figure 2-6). For more information on the Elastic Strip approach, refer to [57, 87, 93, 94].

The disadvantages of the Elastic Band and Elastic Strip approaches are:

a) Both methods rely on a global planner to generate a pre-calculated path, which means that complete environment information is needed before a robot starts to move. This heavily restrains their applicabilities.

b) The Elastic Band and Elastic Strip methods rely on internal forces to bring the robot back to its pre-planned path. There exist some situations in which internal forces fail to work. Figure 2-7 shows such a case, in which a large obstacle stops on the pre-planned path and the internal forces fail to connect the start point and the goal. This causes the disconnection of the elastic band. It can also be called a local minimum from the viewpoint of potential energy.

c) Both methods ignore the kinematic constraints of robots. Therefore, paths found by these two methods may not be feasible for nonholonomic robots.



(a) (b)

Figure 2-5 Protective hull: the bubbles show the free work space around this robot, and the small obstacles represent obstacles nearby. The bubble sizes are limited by obstacles. When this robot approaches an obstacle as shown in b), more bubbles are needed to describe the free space [87].

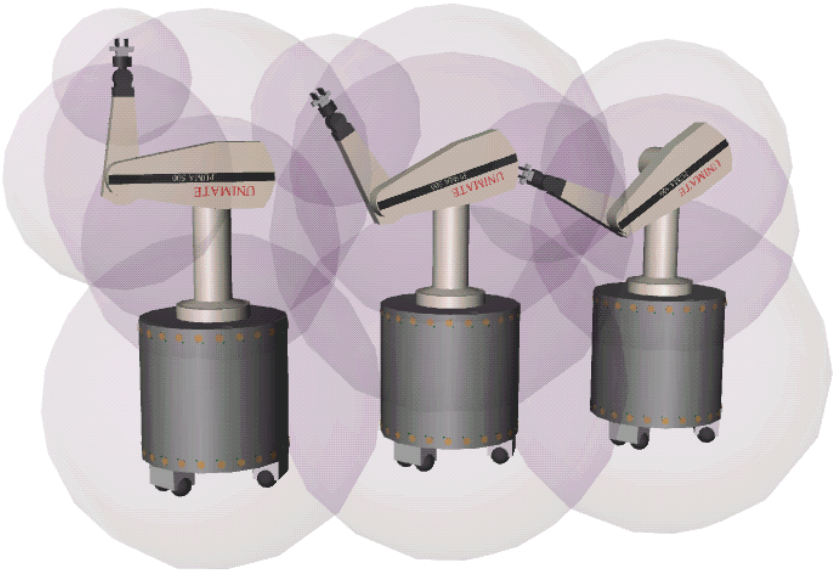


Figure 2-6 Elastic tunnel: some configurations are selected from a pre-planned path. The combination of protective hulls of these configurations forms an elastic tunnel [87].

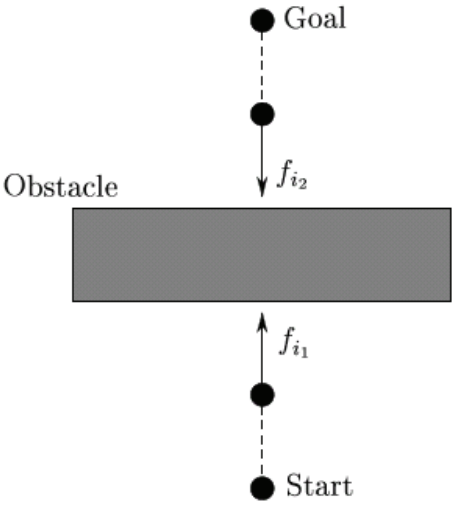


Figure 2-7 Disconnection of elastic band: an obstacle stops on the pre-planned path. The internal forces cannot reconnect the broken elastic strip [87].

2.1.1.4 A Fractional Potential Field Method

Ge and Cui developed a fractional potential method which defines attractive and repulsive potentials by taking into account the relative position and velocity of a robot with respect to

obstacles and targets [44, 95]. Their work was followed by [96], where the attractive potential function is defined based on the relative position, velocity, and acceleration between the robot and the goal, and the repulsive potential function is defined to be a function of the relative position, velocity, and acceleration between the robot and the obstacles.

In Ge and Cui's work, an attractive potential is defined as a function of the relative position and velocity of the target with respect to the robot. By choosing different parameters, the robot can either soft-land on the target, which means the velocity of the robot is the same as that of the target when landing, or hard-land on the target, which means there is no requirement on its velocity when landing. The attractive potential is given by:

$$U_{att}(\mathbf{p},\mathbf{v}) = \alpha_p \left\| \mathbf{p_{tar}}(\mathbf{t}) - \mathbf{p}(\mathbf{t}) \right\|^m + \alpha_v \left\| \mathbf{v_{tar}}(\mathbf{t}) - \mathbf{v}(\mathbf{t}) \right\|^n \tag{2-2}$$

where $\mathbf{p}(\mathbf{t})$ and $\mathbf{p_{tar}}(\mathbf{t})$ denote the positions of the robot and the target at time *t*, respectively; $\mathbf{p} = [x \quad y \quad z]^T$ in a 3D space or $\mathbf{p} = [x \quad y]^T$ in a 2D space; $\mathbf{v}(\mathbf{t})$ and $\mathbf{v_{tar}}(\mathbf{t})$ denote the velocities of the robot and the target at time *t*, respectively; $\left\| \mathbf{p_{tar}}(\mathbf{t}) - \mathbf{p}(\mathbf{t}) \right\|$ is the Euclidean distance between the robot and the target at time *t*; $\left\| \mathbf{v_{tar}}(\mathbf{t}) - \mathbf{v}(\mathbf{t}) \right\|$ is the magnitude of the relative velocity between the target and the robot at time *t*; α_p and α_v are scalar positive parameters; and *m* and *n* are positive constants.

A repulsive potential is also defined as a function of the relative position and velocity of a robot with respect to obstacles. Hence, the virtual force is defined as the negative gradient of the potential in terms of position and velocity. Assume that the position $\mathbf{p_{obs}}(\mathbf{t})$ and velocity $\mathbf{v_{obs}}(\mathbf{t})$ of the nearest point on an obstacle to the robot can be obtained online, the relative velocity between a robot and an obstacle in the direction from the robot to the obstacle is given by

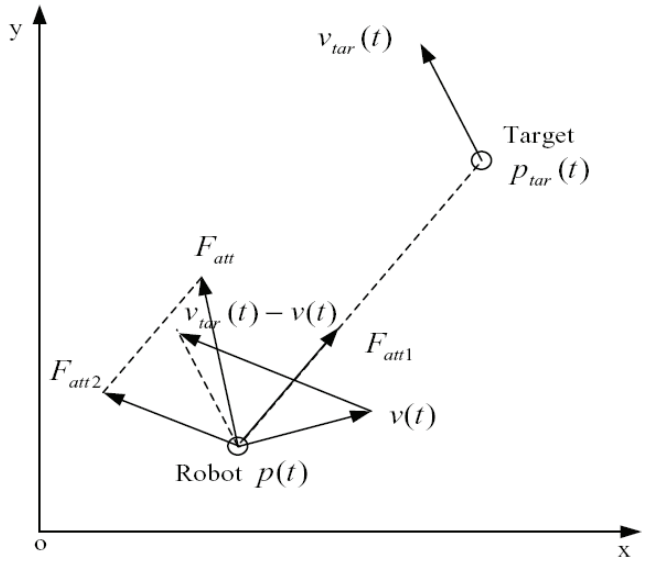


Figure 2-8 Ge & Cui's method: attractive force in 2D space [44]

$$v_{RO}(t)=\left[\mathbf{v}(\mathbf{t})-\mathbf{v}_{\mathbf{obs}}(\mathbf{t})\right]^{T}\mathbf{n}_{\mathbf{RO}} \tag{2-3}$$

where $\mathbf{n_{RO}}$ is a unit vector pointing from the robot to the obstacle. If $v_{RO}(t)\le 0$, i.e. the robot is moving away from the obstacle, no collision avoidance action is needed. If $v_{RO}(t)>0$, i.e. the robot is moving close to the obstacle, collision avoidance action needs to be implemented. Define that $\rho_m(v_{RO})$ is the distance traveled by the robot before $v_{RO}(t)$ reduces to zero, if a maximum deceleration of magnitude a_{max} is applied, $\rho_m(v_{RO})$ is given by

$$\rho_m(v_{RO})=\frac{v_{RO}^2(t)}{2a_{max}} \tag{2-4}$$

The repulsive potential is defined by:

$$U_{rep}(\mathbf{p},\mathbf{v}) =$$

$$\begin{cases} 0 & \text{if } \rho_s(\mathbf{p},\mathbf{p_{obs}}) - \rho_m(v_{RO}) \geq \rho_0 \text{ or } v_{RO} \leq 0 \\ \eta\left(\dfrac{1}{\rho_s(\mathbf{p},\mathbf{p_{obs}}) - \rho_m(\mathbf{v_{RO}})} - \dfrac{1}{\rho_0}\right) & \text{if } 0 < \rho_s(\mathbf{p},\mathbf{p_{obs}}) - \rho_m(v_{RO}) < \rho_0 \text{ and } v_{RO} > 0 \\ \text{not defined} & \text{if } v_{RO} > 0 \text{ and } \rho_s(\mathbf{p},\mathbf{p_{obs}}) < \rho_m(v_{RO}) \end{cases}$$

(2-5)

where the shortest distance between a robot and an obstacle is denoted by $\rho_s(\mathbf{p}(\mathbf{t}),\mathbf{p_{obs}}(\mathbf{t}))$. U_{rep} denotes the repulsive potential generated by this obstacle, ρ_0 is a positive constant describing the influence range of the obstacle, and η is a positive constant.

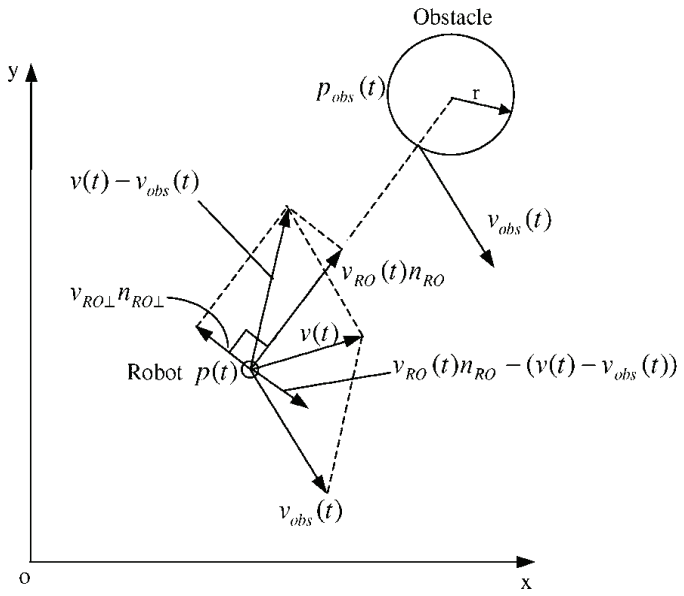


Figure 2-9 Ge & Cui's method: vectors for defining repulsive potential [44]

The advantage of Ge & Cui's fractional potential method is that the robot's relative position and velocity with respect to obstacles and the target are taken into consideration when constructing the potential field. The robot's physical size is also integrated into the control model [44]. Since the relative position and velocity used to define potential functions are related to a robot and its target only, it is clear that this approach cannot be used in multi-robot cases.

2.1.1.5 A Potential Field Model Using Generalized Sigmoid Functions

Ren, et al. noticed that the existing potential field methods often fail to provide accurate representations of objects of arbitrary shapes, so they suggested a potential field model based on generalized sigmoid functions. This approach is capable of providing an accurate representation of objects with complex geometric shapes [45]. The generalized sigmoid function is given by

$$f(\mathbf{q}) = \sum_{i=0}^{M} \prod_{j=0}^{N} f_{sig}\left(\varphi_{ij}(\mathbf{q})\right) \tag{2-6}$$

where

$$f_{sig}(s) = \frac{1}{1+e^{-\gamma f(s)}} \tag{2-7}$$

where $\mathbf{q} = (x, y, z)$ denotes any point in 3D space, $s = \varphi(\mathbf{q})$ is a surface function representing the distance from the object surface. M is the number of line segments to represent the object surfaces. $N = 2$ in 2D case and $N = 3$ in 3D case. The function s is defined as follows: on the boundary surface, $s = 0$; for all points on the boundary, $s = 0.5$; in the obstacle area, all points are associated with high potential values approaching unity; in the clear area, potential values approach zero. By choosing different values on parameter γ, the affected area of this potential field can be adjusted, as shown in Figure 2-10. If a mobile robot travels in an obstacle cluttered environment, a larger γ can be used to restrict the coverage areas of the obstacles' potential fields so that this robot is capable of passing through narrow passages. Conversely, if a robot is travelling in a wide environment, a smaller γ can be applied. Figure 2-11 shows the potential field of a polygon with different γ values ($\gamma = 2,6,10$). When γ is with a larger value, e.g., $\gamma = 6$, its potential field decays more rapidly than when it is smaller ($\gamma = 2$). At the same time, the coverage area of its potential field becomes smaller than when γ is a small value.

The main advantage of this approach is that it is capable of providing an accurate representation for objects with complex geometric shapes. The coverage area of a potential field can be controlled by changing the value of γ, which makes it flexible for both obstacle cluttered environments and empty environments. The disadvantage is that, as in other potential field-based approaches, a robot's dynamic and kinematic characteristics are not taken into consideration in motion planning.

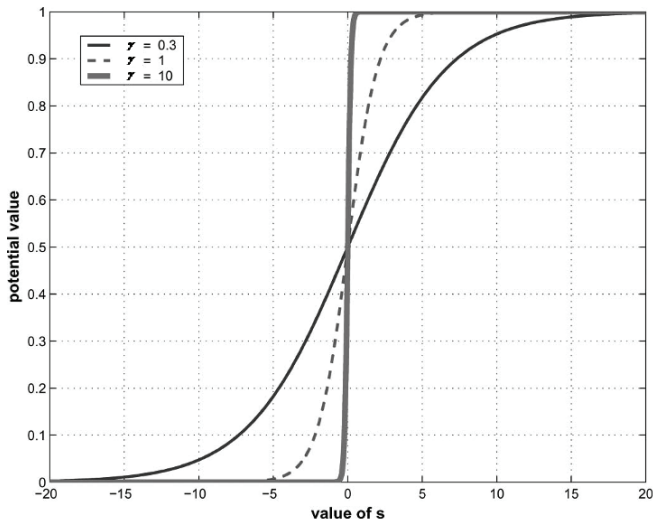


Figure 2-10 Effect of parameter γ [45]

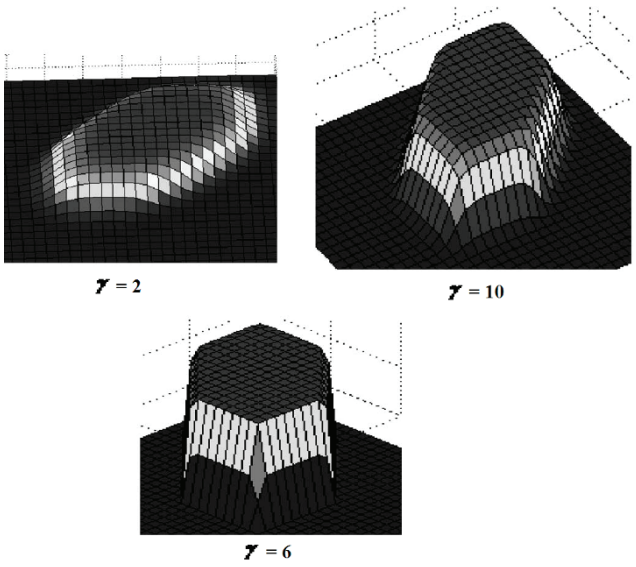


Figure 2-11 The potential field with different γ [45]

2.1.2 Vector Field Histogram and Its Varieties

Borenstein and Koren studied the limitations of potential field methods [97] and developed the Vector Field Histogram (VFH) method [62, 98]. Further research led to VFH+ [63], VFH* [64] and VPH [99].

2.1.2.1 Vector Field Histogram Method

VFH is an algorithm looking for gaps between obstacles which are in front of a robot [62, 98]. VFH builds a local map based on the concept of *certainty grid* [100] from the current sensor range readings. VFH then generates a polar histogram as shown in Figure 2-12. The *x* axis represents the angle at which the obstacles are found and the *y* axis represents the probability that there is an obstacle in that direction based on the occupancy grid's cell values.

From this histogram a steering direction is calculated. In the polar histogram in Figure 2-12, peaks denote sectors with high obstacle density, and valleys denote sectors with low obstacle density. Any valley with obstacle densities below the threshold value is a possible gap for travel. Since there usually exist several openings, VFH simply selects the steering direction dependent on the width of opening.

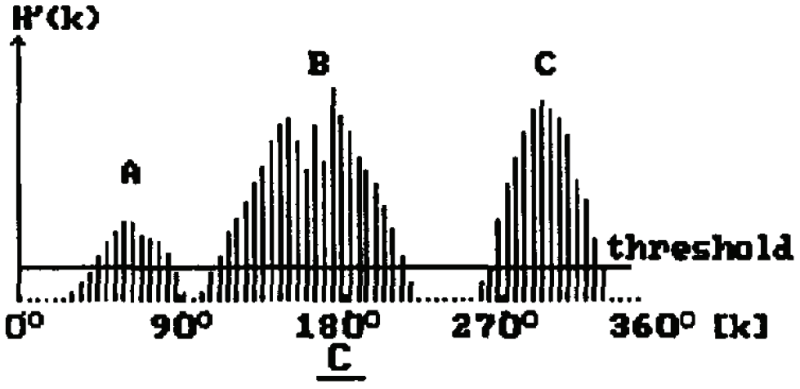


Figure 2-12 Polar histogram in VFH [98]

2.1.2.2 Enhanced Vector Field Histogram Method

The Enhanced Vector Field Histogram (VFH+) offers several improvements on the VFH that result in smoother robot trajectories and greater reliability [63], including

a) Obstacles are enlarged by the robot radius and a security distance.

b) A threshold hysteresis is applied on the polar histogram to reduce oscillations between valleys.

c) Valleys that require control inputs exceeding actuator limits are blocked, thus the kinematics limitations are taken into consideration.

d) A cost function is used to evaluate each candidate and choose an appropriate direction to move:

$$G = a \cdot p_1 + b \cdot p_2 + c \cdot p_3 \tag{2-8}$$

where p_1 is the component of target effect on the cost function, p_2 is the difference between the direction of opening and current wheel orientation, and p_3 is the difference between the previous selected direction and the direction of opening. The selection of parameters *a, b, c* will determine the way the robot reacts to obstacles.

2.1.2.3 VFH* Method

VFH+ sometimes fails to choose the most appropriate direction because of its local nature. VFH* amends this problem by introducing A* search algorithm [22] into the direction determination. Unlike VFH+, VFH* analyses the consequences of heading towards each possible direction before making a final choice for the new direction of motion. For each candidate direction, VFH* computes the new position and orientation that the robot would have after moving for several steps. Using this search process, the robot will find a better solution than using VFH+ only [64].

The performance of VFH* apparently depends on the look-ahead distance. Increasing this distance will slow the obstacle avoidance process. Thus, parameter selection is a trade-off between the quality and speed of the algorithm.

2.1.2.4 Vector Polar Histogram Method

The Vector Polar Histogram (VPH) improves VFH+ in another way [99]. The VPH employs a three-step data disposal process to get the new steering direction. In the first and second step, the group of distances to obstacles is transformed into a polar histogram, and a threshold function is set up. The polar histogram is reduced to a binary histogram (see Figure 2-13) by comparing with the threshold function. In the third step, a set of candidates can be obtained from the binary histogram; a cost function is then brought up to find the best steering direction.

In VFH and VFH+, the threshold is set to be a constant value, as shown in Figure 2-12. In VPH, a robot's dynamic characteristics are taken into consideration when the polar histogram is generated [99]. The threshold is defined by

$$D(i,t)=\begin{cases}\dfrac{v_t^2\,cos^2\left(\theta(t,i)\right)}{2a}+R+D; & \left|\theta(t,i)\right|\le\dfrac{\pi}{2}\\ R+D; & \left|\theta(t,i)\right|>\dfrac{\pi}{2}\end{cases} \tag{2-9}$$

where $D(i,t)$ is the threshold for obstacle i in time t. v_t is the real-time speed of mobile robot, $\theta(t,i)$ is angle between the robot and obstacle i, a is the deceleration of mobile robot, R is radius of mobile robot, D is the safe distance between this robot and the obstacle. In this equation, v_t and $\theta(t,i)$ can be obtained real-time from the robot, a and R are constants. D should be adjusted based on environment information.

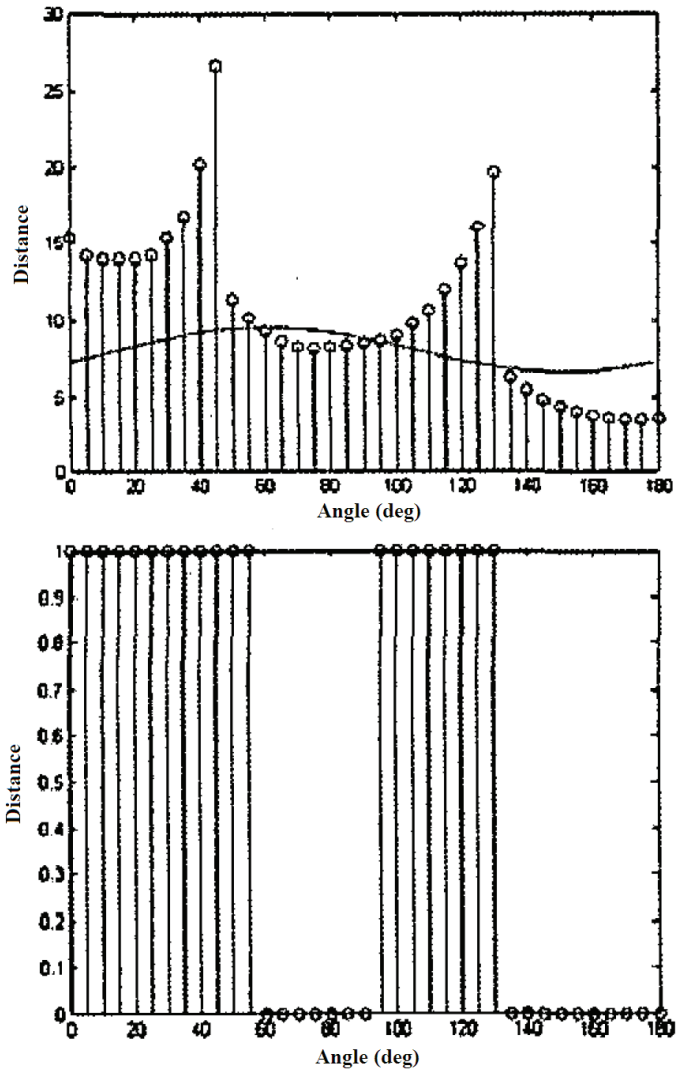


Figure 2-13 Creation of a binary histogram [99]

2.1.3 Dynamic Window-based Approaches

2.1.3.1 Dynamic Window Approach

The Dynamic Window Approach (DWA) only considers circular trajectories determined by translational and rotational velocities (v,ω) [60, 101]. This results in a 2D velocity search space. The restriction to admissible velocities ensures that only safe trajectories are considered. A pair (v,ω) is considered admissible only if the robot is able to stop before it reaches the

closest obstacle on the corresponding curvature. The dynamic window further restricts the admissible velocities to those that can be reached within a short time interval given current velocity and the limited acceleration of a robot [60]. The dynamic window is a rectangle since acceleration capabilities for translation and steering are independent (see Figure 2-14).

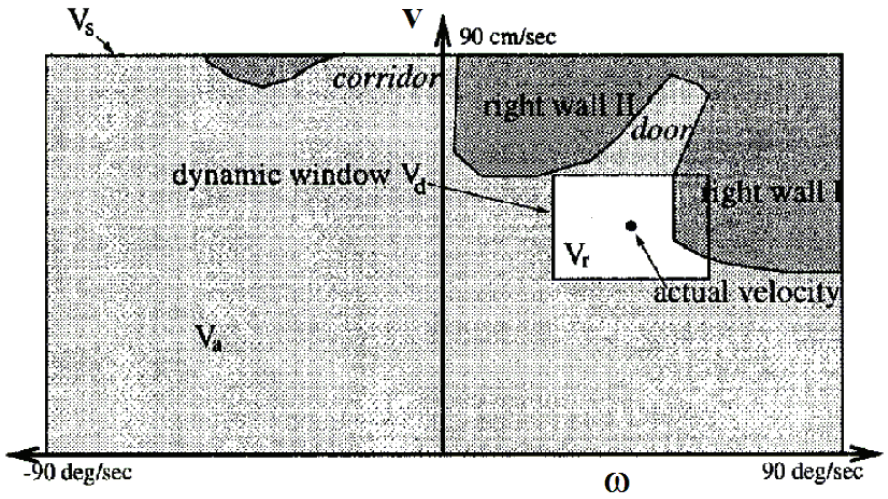


Figure 2-14 Dynamic window [101]

The desired velocity and acceleration are then chosen by applying an objective function in the search space.

$$G(\mathbf{v},\boldsymbol{\omega}) = \sigma(\alpha \cdot p_1(\mathbf{v},\boldsymbol{\omega}) + \beta \cdot p_2(\mathbf{v},\boldsymbol{\omega}) + \gamma \cdot p_3(\mathbf{v},\boldsymbol{\omega})) \tag{2-10}$$

Target heading $p_1(\mathbf{v},\boldsymbol{\omega})$ measures progress towards the goal location and is given by $180-\theta$. θ is the angle of the target point relative to the robot's heading direction. Clearance $p_2(\mathbf{v},\boldsymbol{\omega})$ represents the distance to the closest obstacle in the trajectory. If no obstacle is on the curvature this value is set to a large constant. Velocity $p_3(\mathbf{v},\boldsymbol{\omega})$ is the forward velocity of the robot.

2.1.3.2 Global Dynamic Window Approach

Brock and Khatib extended DWA to the Global Dynamic Window Approach (GDWA) [61, 87]. In DWA, the robot's motion with respect to the goal is only influenced by the target heading so it is susceptible to local minima. GDWA remedies this by incorporating information about the connectivity of the free space into the selection of a motion command. This is achieved by combining with a Navigation Function 1 (NF1) which is which is a simple global motion planning algorithm [12]. The objective function in GDWA is:

$$\Omega_g(\mathbf{p},\mathbf{v},\mathbf{a}) = \alpha \cdot nf1(\mathbf{p},\mathbf{v}) + \beta \cdot vel(\mathbf{v}) + \gamma \cdot g(\mathbf{p},\mathbf{v},\mathbf{a}) + \delta \cdot \Delta nf1(\mathbf{p},\mathbf{v},\mathbf{a}) \tag{2-11}$$

The value of $nf1(\mathbf{p},\mathbf{v})$ can be determined by examining the neighbours of the grid cell that corresponds to the robot's location. Compared with Equation (2-10), the function $p_1(\mathbf{v},\boldsymbol{\omega})$ is replaced by $nf1(\mathbf{p},\mathbf{v})$. This function's value increases if $\mathbf{v}$ is aligned with the gradient or the navigation function at the robot's location, rendering the GDWA immune to local minima, since NF1 is a local minima-free navigation function. The GDWA provides real-time local minima-free obstacle avoidance at high speed. For more research on this method, refer to [102, 103].

Besides GDWA, there are some approaches which improve the DWA approach in other ways. For example, the focussed D* algorithm [104] is integrated with DWA for the purpose of real-time motion planning in a partially known environment in [105, 106]. To enhance the convergency of DWA, the potential field-based navigation function proposed by Rimon and Koditschek in [107] is combined with DWA in [103].

2.1.4 Curvature Velocity Method

Simmons developed the Curvature Velocity Method (CVM), which takes the actual kinematics constraints and some dynamic constraints into account for obstacle avoidance [59]. Further

research led to the Lane Curvature Method (LCM), which demonstrates better performance than the CVM in some cases [108].

The CVM formulates the local obstacle avoidance problem as a constrained optimization problem in the velocity space of the robot. The velocity space of a robot is a set of controllable velocities, which consists of rotational velocity ω and translational velocity v with:

$$-v_{max} < v < v_{max} \tag{2-12}$$

$$-\omega_{max} < v < \omega_{max} \tag{2-13}$$

A robot is assumed to travel along arcs of circles with curvature $c = \omega / v$. Obstacles are also transformed into the velocity space by calculating the distance from a robot's location to the obstacles following some constant curvature trajectory. Only the curvature between C_{min} and C_{max} are considered. The final decision of a new velocity is then made based on an objective function.

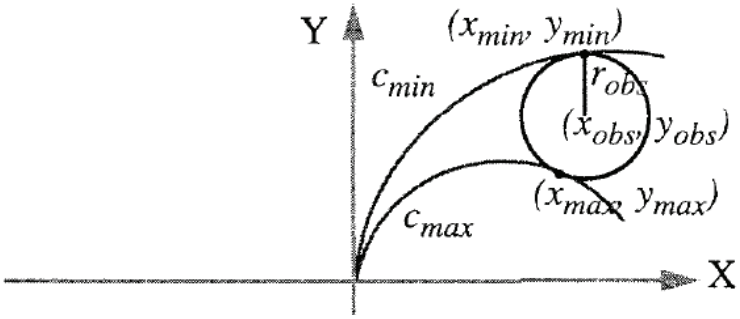


Figure 2-15 Tangent curvatures for an obstacle [59]

The CVM considers the robot's kinematics and dynamics constraints, which makes it an efficient approach for real-time obstacle avoidance. The drawback of this method is its circular simplification of the shape of an obstacle and the existence of local minima.

2.2 Approaches to Multi-Robot Motion Planning

Motion planning methods for multi-robots are different from those for a single robot. Although some algorithms are proved to be feasible and efficient for single robot cases, they cannot be transferred directly to multiple robot cases because of the increased number of requirements and constraints introduced by a multi-robot system. This section does not intend to be a complete review of multi-robot motion planning approaches, and only covers some topics which are closely related to this research.

2.2.1 Centralized and Decentralized Approaches

In general, methods for multi-robot path planning can be categorized into centralized and decentralized techniques [12].

Centralized approaches consider all robots together as if they are forming a high degree of freedom robot [109-111]. The advantage of centralized approaches is that they can provide complete and optimal solutions, but as the number of robots and obstacles increases, the centralized method will suffer from the exponentially increasing computation complexity. Centralized approaches are often slow and require complete system information. This makes them inappropriate for applications in which robots are operating in dynamic or unknown environments and online planning abilities are needed.

Some approaches have been proposed to reduce the size of the search space; for example, Svestka and Overmars created a probabilistic roadmap through the environment and this method was shown to be probabilistically complete [112, 113]. This approach constructs a roadmap which represents a network of feasible motions for the robots. The roadmap is constructed in two steps. First, a simple roadmap, which is referred as a "subgraph", is constructed for each robot. Then all subgraphs are combined into a roadmap for the composite robot, which is a called a "super-graph". This super-graph is then utilized to find approaches for multiple robots. In Figure 2-16, subgraph $A=\{x_1,x_2,x_3\}$ and $B=\{x_4,x_5,x_6,x_7\}$ are

combined into a super-graph. A robot denoted by a white disc is supposed to travel from x_7 to x_2. Another robot represented by a black disc is supposed to travel from x_3 to x_4. Their movements are denoted by snapshots in Figure 2-16. This approach has been proved to be probabilistically complete but is not feasible for dynamic or unknown environments.

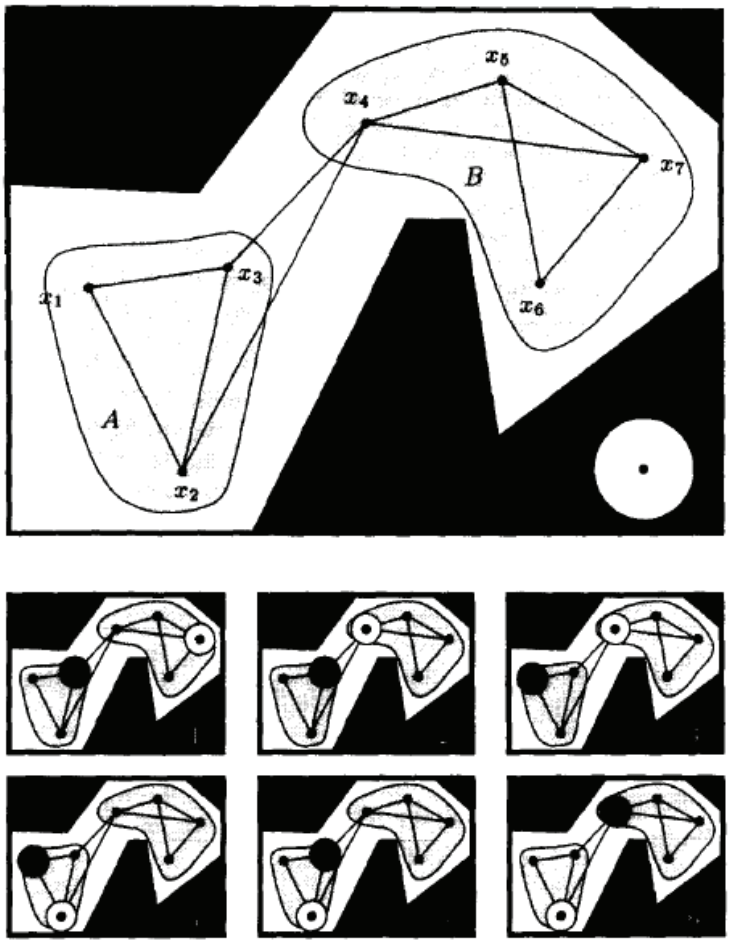


Figure 2-16 Combining subgraphs into a super-graph [113]

Decentralized approaches typically generate a path for an individual robot independently and avoid collision locally [50, 54, 114-122]. The problem of multi-robot motion planning is then converted into several single robot motion planning problems. Decentralized approaches do not require robots to have complete system information and are generally fast enough for real

time planning. However, they are inherently incomplete and cannot find solutions in some situations [123].

Some approaches have been developed to search for near-optimal solutions. In Guo and Parker's work, a cost function is designed for individual robots and D^* is utilized to minimize the cost function to find a optimal solution [124]. In Azarm and Schmidt's work, a dynamic priority assignment approaches has been proposed [122]. In this approach, all combinations of priorities are considered for robots which are about to collide. The priority assignment plan which will result the least cost will be selected.

2.2.2 Priority-based Planning

Priority-based planning is a popular approach for multi-robot motion planning [50, 114, 116, 125-127]. Techniques of this class assign priorities to each robot and compute paths in the order of decreasing priority. A robot with higher priority is treated as an obstacle in the planning of a robot with lower priority. This simplifies a multi-robot motion planning problem into single-robot motion planning problems.

Warren described an artificial potential field method for multiple robot motion planning [50]. In Warren's method, each robot is assigned a unique task priority. A path that avoids collision with stationary obstacles only is first planned for the robot with highest priority. Then, a trajectory for the second highest priority robot is planned so that it avoids both the stationary obstacles and the highest priority robot, which is treated as a moving obstacle. This process is continued until trajectories for all robots have been planned. An example of motion planning for four robots is given in Figure 2-17. The path for Robot 1 is planned first, so this path directly connects the start point and destination for Robot 1. A path is then planned for Robot 2, in which the path of Robot 1 is treated as an obstacle. The path of Robot 3 should avoid collisions with Robot 1 and Robot 2. Robot 4, which has the lowest priority, needs to avoid collisions with the other three robots.

An important issue of prioritized planning is that the way priority is assigned influences the resultant paths significantly. Such an example is given in [7], as shown in Figure 2-18. Robot 1 and Robot 2 are denoted by a blue circle and red circle respectively, and their start and goal positions are denoted by S1, S2, G1, G2. (a) gives the optimal paths of the two robots. If a path is first planned for Robot 1, the path of Robot 2 will have to include a large contour since Robot 1 blocks the corridor (as shown in (b) of Figure 2-18). By contrast, if a path is planned for Robot 2 first and then for Robot 1, the resultant paths are shown in (c) of Figure 2-18, which is much more efficient in terms of the total cost.

Some approaches have been developed to explore the problem of priority assignment [118, 122, 126]. The priorities of robots are set dynamically, based on the current situation, to minimize the total cost in [118, 122]. A randomized hill-climbing search is utilized to find an optimal priority scheme in [126].

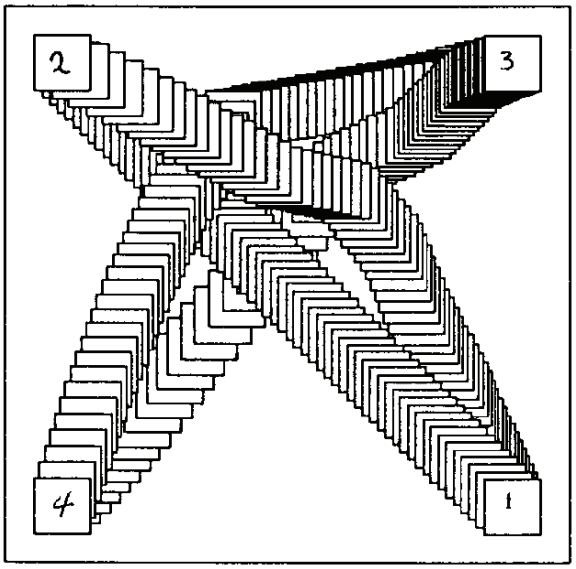


Figure 2-17 Prioritized planning: the path of Robot 1 is planned first. Paths for Robots 2, 3 and 4 are then planned in sequence [50].

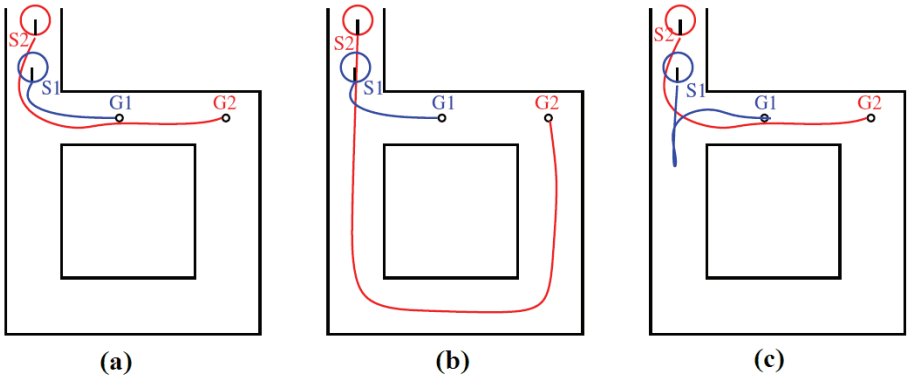


Figure 2-18 The effect of priority assignment: (a) optimal paths for two robots (b) if a path is planned for Robot 1 first, Robot 2 will have to follow a large contour. (c) if a path is planned for Robot 2 first, the total path length is shorter [7].

2.2.3 Path-Velocity Decomposition Approaches

Some approaches decompose the motion planning problem into two sub-problems: path planning and velocity planning [124, 128-131]. In these approaches, robots are kept on their preplanned paths and speed changing strategies are applied to avoid collisions. The robots may need to stop, move forward, or even move backward along their trajectories. For example, idle times are inserted into preplanned trajectories to avoid collisions in [132].

In the work of Ferrari et al, three performance indexes are adopted to evaluate the quality of planned trajectories for multiple robots [131, 133]. The first one is the running time (RT), which is the minimum time for a robot to reach its goal using a pre-planned path. Since a robot moving on its path must avoid potential collisions with other robots, RT depends not only on the robot's path and velocity, but on the paths and velocity schedules of other robots. The second one is the motion error (ME), which measures how far a robot can move away from its pre-planned path without colliding with obstacles or other robots. The ME index is computed in the pre-planning phase and is proportional to the distance of the path from the obstacles. The third performance index is the velocity error (VE), which measures how much a robot can vary its velocity without worsening the global plan. For example, in Figure 2-19 robot R_A is scheduled to stop at point A and wait for robot R_B to pass. In this case, collision may occur if

robot R_B arrives later than scheduled. Hence robot R_B has a value of VE equal to zero. robot R_A can arrive later than scheduled because it will not result in collision with R_B. Moreover, robot R_C is scheduled to stop at point C and wait for R_A. Robot R_A should not arrive at its stop point A later than the time scheduled, otherwise this delay could cause a collision with R_C. The value of VE for R_A has a finite value different from zero. The value of VE for R_C, in this example, is positive infinite, because its delay will not cause collision with other robots. VE is not a property of the single path, but a property of the paths for all robots. The approach will search from a set of possible paths and find a solution with good performance indexes.

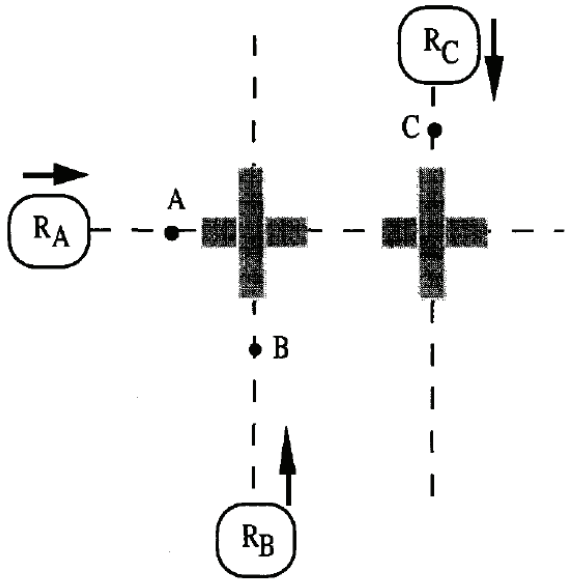


Figure 2-19 VE evaluation for robot path [133]

In the work of Guo and Parker [124], a coordination diagram, which represents an N-dimensional (N is the number of robots) space using path length as the parameter, is constructed based on collision checks among all preplanned paths. The D* search algorithm [104, 134] is then executed on the coordination diagram, and a velocity profile which minimizes a global performance function is chosen.

2.3 Conclusions

This chapter provides a literature review on previous works which are closely related to this research. Some approaches for single robot motion planning and collision avoidance have been introduced, which include the Potential Field Method (PFM) and its varieties, Vector Field Histogram (VFH) and its varieties, Dynamic Window Approach (DWA) and Global Dynamic Window Approach (GDWA), and the Curvature Velocity Method (CVM). These approaches have been proved to be applicable in some cases, but can not be directly applied to multi-robot motion planning and cooperation directly. Some important issues in multi-robot motion planning were subsequently discussed, including centralized approaches and decentralized approaches, priority-based planning and path-velocity decomposition approaches. The approaches mentioned above have been proved to be feasible in certain cases but their applicability is limited. This research work presents a novel force field method, the F^2 method, which is a generic approach for robot motion planning and collaboration.

In the F^2 method, the coverage of a robot's force field is determined by parameters including the robot's size, travelling speed, and priority with respect to other robots. If there are obstacles or other robots in the area of a robot's force field, this robot will be acted on by virtual repulsive forces from other robots/obstacles. A robot only reacts to obstacles that are in the coverage of its own force field and does not need to search the whole work space as many other methods require, which significantly increases the efficiency of motion planning and coordination.

In the F^2 method, a robot's physical characteristics, such as size and geometry, are used in the construction of its force field. Its dynamic and kinematic characteristics, such as linear velocity and angular velocity, are taken into consideration when determining a robot's motion. These make the F^2 method suitable for real applications.

In the F^2 method, a robot needs to know its location and destination in the environment but a precise map is not essential. If there are environmental changes or moving obstacles in the work space, a robot reacts immediately based on information obtained from updated sensor

data and inter-robot communication. The F^2 method is suitable for applications in partially known or dynamically changing environments.

The F^2 method is suitable for use in multi-robot cases. A robot using the F^2 method works in a decentralized manner, that is, a robot plans its path and motion independently according to the surrounding environment and its own status, so the F^2 method will not suffer from an exponentially increasing computation burden as some centralized approaches do. Another advantage of the F^2 method is that the task priority is taken into account in the construction of the force field. A robot with higher task priority will have priority in collision avoidance.

This research work presents a systematic study of the F^2 method. The basic concept of the F^2 method is described in detail in Chapter 3. Chapter 4 presents several approaches which are developed based on the concept of F^2 and are suitable for different applications. Chapter 5 focuses on the parameter optimization problems of the F^2 method. Chapter 6 introduces experimental studies on the F^2 method. Appendix A briefly presents the work extending the F^2 method to 3D case, that is, 3-Dimensional Force Field ($3DF^2$) for manipulator real-time motion planning and collision avoidance.

Chapter 3
Force Field Method

In this chapter, the concept of the proposed force field (F^2) method is described in detail. In the F^2 method, a virtual repulsive force field is created and centred on the robot. This force field varies with the robot's status during its movement, including the robot's speed, task priority, environmental factors and the robot body dimension. If any obstacle exists in the coverage area of this force field, the robot will act on a repulsive force from the obstacle. If a robot's force field overlaps with another robot's force field, both robots will suffer repulsive forces from each other. The interactions among a robot's force field with obstacles and other robots, i.e., the repulsive forces, combining with the attractive force from the goal, are utilized to determine a robot's motion in the F^2 method.

This chapter starts by analysing the drawbacks of the classical Potential Field Method in Section 3.1. A robot model is given in Section 3.2. The development of the force field method is detailed in Section 3.3, including the generation of attractive and repulsive forces. A Canonical Force Field method (CF^2) is described in Section 3.4. Simulation results are given in Section 3.5. Section 3.6 evaluates the performance of the CF^2 method and Section 3.7 concludes this chapter.

3.1 Introduction

The Potential Field Method is the approach which most resembles the proposed F^2 method [42]. This section reveals the drawbacks of the Potential Field Method and highlights the differences between the Potential Field and the F^2 methods. For simplicity of explanation, discussions below are limited to mobile robots moving on a 2-Dimensional (2D) surface.

A. The Effect of a Robot's Velocity on Collision Avoidance

The effect of a robot's velocity on collision avoidance has not been taken into account in most of the potential field methods. In Figure 3-1, obstacles A, B, C, D and E are located the same distance away from a robot. A, C and D are stationary while B and E are moving obstacles. It can be clearly seen that obstacle D will not have any influence on this robot's movement at the current moment because it is located behind the robot and does not move. Obstacle C is stationary and does not block the robot's motion. Obstacle B is moving away from the robot's path, therefore B and C would not affect the robot's movement either. Obstacle E is moving towards the robot and has the potential to block the robot's motion soon. Obstacle A is stationary but located on the robot's front, so obstacles A and E must be taken into consideration in collision avoidance and the other three obstacles can be ignored. However all obstacles will have same influence on the robot's movement in most conventional potential field approaches since their distance from this robot is equal.

The proposed F^2 method works more reasonably than Potential Field-based approaches in such cases. The force field of a robot will vary with its velocity and focus on the directions where collisions may occur. The coverage area of the F^2 increases when the robot is travelling at a higher speed and decreases when the speed is lower.

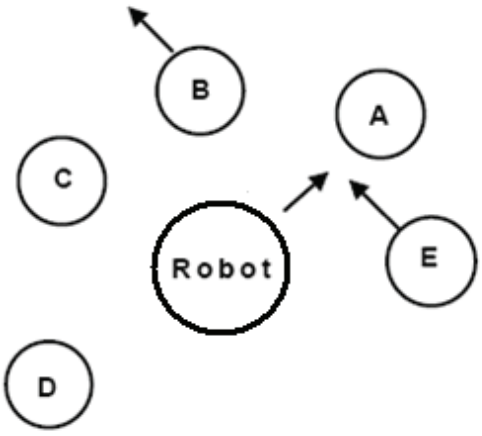


Figure 3-1 The effect of velocity on collision avoidance

B. Complex and Narrow Environments

A reported drawback of Potential Field-based approaches is the 'dead-lock' caused by narrow passage [97, 135]. That is, when obstacles are close to each other, the resultant potential fields may overlap and produce a relatively higher potential field in the overlapped area. A robot using the potential field for collision avoidance will fail to climb this potential hill and stop. This case becomes more crucial when a robot navigates in a corridor-like environment.

A possible solution to this problem is to create a virtual obstacle at the detected dead-lock position, which will repel the robot from this local minimum [80]. Another solution is to merge separate small obstacles into a virtual large obstacle and let the robot take a large detour around the virtual large obstacle [135]. The disadvantage of both approaches is that they require more information about the environment and robots may have to travel longer distances.

According to the force field proposed in this research, a robot travelling in an obstacle-cluttered environment will reduce its speed according to sensor measurements. As a result, its force field will shrink, which allows the robot to pass through narrow passages.

Based on the analyses above, the proposed new force field should have the following merits:

1) The coverage of a force field should vary with a robot's speed. When a robot moves faster, its force field should cover a larger area than when the robot moves with a slower speed. This means that a robot moving faster is given more space for decelerating and steering away from obstacles.
2) The coverage of a robot's force field should vary with a robot's physical properties, such as its size. The force field of a robot with larger size (or volume) should cover more area than that of a robot with smaller size (or volume) under otherwise identical conditions.

3) The proposed force field should be centred at the robot instead of obstacles. The magnitude of the repulsive force should decrease with the increase of the distance to this robot, i.e. the repulsive force should increase as the robot moves closer to obstacles.

4) The proposed force field should focus on a robot's moving direction. As illustrated before, an obstacle which is located in front of a robot will have a significant effect on the robot's movement, while an obstacle behind a robot will pose less threat. Obstacles in a robot's moving direction should be paid more attention in motion planning and collision avoidance.

3.2 Mobile Robot Motion Model

In this research, a robot is modelled as a rigid body moving on a horizontal plane. The dimensionality of the robot in a working space is three, two for position in the plane and one for orientation along the perpendicular axis, which is orthogonal to the plane. Other factors which may affect the robot's movement, such as wheel slippages and additional degrees of freedom due to a robot's internal structure, are ignored.

Two reference frames are defined: a global reference frame $O(X,Y)$ and a local reference frame $o(x,y)$ attached to a robot's body (Figure 3-2). The axes X and Y define an inertial frame of reference on the plane as a global reference frame from an origin $O(X,Y)$. To specify the position of the robot, choose a point o on the robot as its position reference point. $o(x,y)$ defines two axes relative to o on the robot body and is thus the robot's local reference frame. Note that x axis is set to be along this robot's moving direction. The position of o in the global reference frame is specified by (X_r,Y_r) and the angular difference between the global and local reference frames is given by θ_r. The pose of a robot in the global reference frame is described as a vector with three elements.

$$\mathbf{r} = \left[X_r, Y_r, \theta_r\right]^T \tag{3-1}$$

To describe the motion of a robot motion, it is necessary to map the robot motion along the axes of the robot's local reference frame to motion along the axes of the global reference frame. The mapping is a function of the current location of the robot and is accomplished using the orthogonal rotation matrix:

$$R(\theta_r) = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \tag{3-2}$$

The mapping function is therefore defined by:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} X_r \\ Y_r \end{bmatrix} \tag{3-3}$$

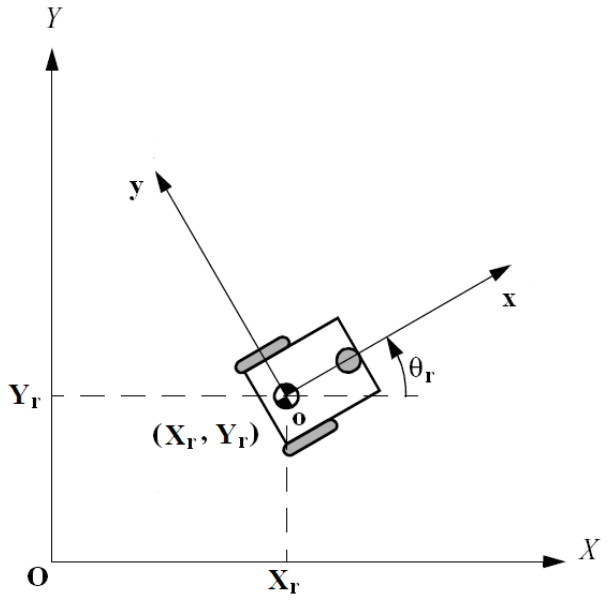


Figure 3-2 Global reference frame and local reference frame

Consider a scenario of N mobile robots navigating in a global coordinate frame. Let robot i, at time $t = s$, locate at (X_i, Y_i) and orient with angle θ_i to the X-axis of the world coordinate. The state of robot i can be described by:

$$\mathbf{r_{i,s}} = \left[X_{i,s}, Y_{i,s}, \theta_{i,s} \right]^T, \quad i = 1,2,\ldots N \tag{3-4}$$

A robot is assigned a task such that it needs to travel from its current or start location to a goal location at $\mathbf{r_g} = (X_g, Y_g)$. The robot then moves, under an appropriate control, towards the goal in a number of time steps Δt. The equation that describes the robot motion from time interval s to $s+1$ can be expressed as a function of the current location, orientation and control:

$$\mathbf{r_{i,s+1}} = h(\mathbf{r_{i,s}}, \mathbf{u_{i,s}}) \tag{3-5}$$

where $\mathbf{u_{i,s}} = (\mathbf{v_{i,s}}, \boldsymbol{\omega}_{\mathbf{i,s}})$ is the control issued to the mobile robot i, in which $\mathbf{v_{i,s}}$ is the translational velocity and $\boldsymbol{\omega}_{\mathbf{i,s}}$ is its angular velocity. The motion equation of robot i at time interval $s+1$ is described by:

$$\begin{bmatrix} x_{i,s+1} \\ y_{i,s+1} \\ \theta_{i,s+1} \end{bmatrix} = \begin{bmatrix} x_{i,s} + \mathbf{v_{i,s}} \cos(\theta_{i,s}) \Delta t \\ y_{i,s} + \mathbf{v_{i,s}} \sin(\theta_{i,s}) \Delta t \\ \theta_{i,s} + \boldsymbol{\omega}_{\mathbf{i,s}} \Delta t \end{bmatrix} \tag{3-6}$$

The control commands, $\mathbf{v_{i,s}}$ and $\boldsymbol{\omega}_{\mathbf{i,s}}$ are determined such that during the travel, the robot is free of collisions onto obstacles or other robots. The force field method is applied to generate the control commands based on the forces exerted on the robot, which is the function of attractive force from the goal and repulsive force from obstacles and other robots:

$$\mathbf{u} = f\left(\mathbf{F}_{\mathbf{rep\_o}}, \mathbf{F}_{\mathbf{rep\_j}}, \mathbf{F}_{\mathbf{att\_g}}\right) \tag{3-7}$$

where $\mathbf{F}_{\mathbf{rep\_o}}$ is the repulsive force from obstacles, $\mathbf{F}_{\mathbf{rep\_j}}$ is the repulsive force from other robots, and $\mathbf{F}_{\mathbf{att\_g}}$ is the attractive force to the goal. The control is further expressed by its components as

$$\mathbf{u} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} f_v\left(\mathbf{F}_{\mathbf{rep\_o}}, \mathbf{F}_{\mathbf{rep\_j}}, \mathbf{F}_{\mathbf{att\_g}}\right) \\ f_\omega\left(\mathbf{F}_{\mathbf{rep\_o}}, \mathbf{F}_{\mathbf{rep\_j}}, \mathbf{F}_{\mathbf{att\_g}}\right) \end{bmatrix} \tag{3-8}$$

3.3 Construction of a Force Field

3.3.1 Definition of a Force Field

In the F^2 method, a force field is defined as a virtual field of repulsive force in the vicinity of a robot when it travels in a working space. The magnitude and orientation of a force field are determined by, and vary with, the robot's status. This virtual repulsive force increases when the distance to the robot decreases. Note that the construction of the force field described below is in the global reference frame. Some parameters in the F^2 method are listed in Table 3-1 and other parameters are defined below:

$$\theta = \theta_0 - \theta_r \tag{3-9}$$

$$E_r = \frac{v_r}{v_{max} C} \tag{3-10}$$

$$D_{max} = \frac{k\ E_r\ R_r}{1 - E_r\ \cos\theta} T_p \tag{3-11}$$

$$D_{min} = \rho_0\ D_{max} \tag{3-12}$$

Parameters	Descriptions
R_r	radius of a robot
v_r	absolute value of a robot's speed
v_{max}	maximum absolute value of a robot's speed
T_p	a robot's task priority
θ_r	angle between a robot's moving direction and the X coordinate

Table 3-1 Parameters in the F^2 method

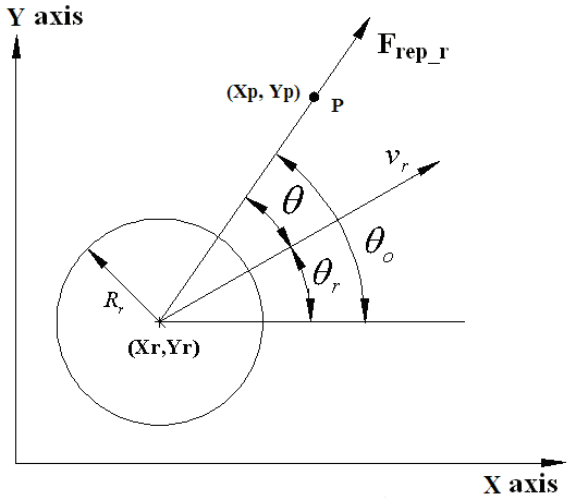


Figure 3-3 Illustration of a robot's parameters

where θ_r denotes a robot's orientation in a global coordinate system. R_r is the radius, from the robot's origin, of the minimum circle embedding the robot entity. For any point $P(X_p, Y_p)$ in the robot's local reference frame, θ denotes the angle of this point in the robot's coordinates (Figure 3-3), which can be determined from θ_r and the angle of this point in the global coordinates (θ_0), as in Equation (3-9). C is a positive number ($C > 1$) which represents the environmental influence to the force field. E_r is a positive decimal fraction with $0 \le E_r < 1$. k is a positive coefficient which determines the size of the area to be covered by the force field. D_{max} is the maximum action distance of a robot's force field and D_{min} is the distance at which this robot has maximum repulsive force. D_{max} shows how far this robot

can affect others in its vicinity. D_{min} provides a safe distance for the robot to prevent other objects from moving into this area. ρ_0 is a positive fractional number with $0 < \rho_0 < 1$ that defines how close the robot can be to obstacles. T_p represents the priority of a robot, which is especially useful for multi-robot coordination. Note that for single robot case, T_p is set to be *1*. Example force fields are shown in Figure 3-5.

Equation (3-11) defines D_{max}, which is the influence area of the force field. Equation (3-10) and (3-11) indicates that the coverage of a force field (D_{max}) in the F^2 method is determined by a robot's size (R_r), its travelling speed (v_r) and maximum speed (v_{max}), a positive multiplier (k), an environment factor (C) and priority (T_p). For a robot with larger volume or travelling at higher speed, its force field will cover a larger area. When a robot travels in an obstacle-cluttered environment, the environment factor (C) can be set larger. Then the coverage of its force field will become smaller and allow the robot to pass through narrow passages.

The magnitude of a repulsive force to be generated by a robot's force field is defined by:

$$\left|\mathbf{F}_{\mathbf{rep\_r}}\right| = \begin{cases} 0 & when\ D > D_{max} \\ \dfrac{D_{max} - D}{D_{max} - D_{min}} P & when\ D_{min} \le D < D_{max} \\ F_{max} & when\ D < D_{min} \end{cases} \tag{3-13}$$

where D is the shortest distance from point $P\left(X_p, Y_p\right)$ in the 2D space to the perimeter of the robot. P is a positive constant scalar which determines the magnitude of the repulsive force. When D changes from D_{min} to D_{max}, the magnitude of the repulsive force changes from P to *0* gradually. Furthermore, F_{max} is the maximum repulsive force which will cause the maximum deceleration on the robot. With $F_{max} \gg P$, P and F_{max} should be selected on the basis of the robot's characteristics, Equation (3-13) shows that the magnitude of

repulsive force varies with the distance to the robot, robot speed, robot size, task priority and several scaling parameters. The force starts from the robot's centre to the given point, whose direction in the global reference frame is given by

$$\angle \mathbf{F}_{\mathbf{rep\_r}} = arctan\left(\frac{Y_p - Y_r}{X_p - X_r}\right) \tag{3-14}$$

To highlight how the force varies with distance and angle with regard to robot, the force field is represented with magnitude contours by defining

$$\rho = D / D_{max} \tag{3-15}$$

Equation (3-13) can then be re-presented in an alternative form as:

$$\left|\mathbf{F}_{\mathbf{rep\_r}}\right| = \begin{cases} 0 & when\ \rho > 1 \\ \dfrac{1-\rho}{1-\rho_0} P & when\ \rho_0 \le \rho < 1 \\ F_{max} & when\ \rho < \rho_0 \end{cases} \tag{3-16}$$

When ρ changes from ρ_0 to *1*, the magnitude of the repulsive force changes from P to *0*. Equations (3-13) and (3-16) explain the concept of force field in two aspects. Equation (3-13) shows that the magnitude of repulsive force varies with distance, and Equation (3-16) emphasizes the concept of contours.

Figure 3-4 illustrates how the magnitude of a repulsive force varies with ρ. In this example, $\rho_0 = 0.2$, $P = 10$, $F_{max} = 20$. When a point $P\left(X_p, Y_p\right)$ is far from this robot and out of the robot's D_{max} , $\rho > 1$, the force is *0*. As ρ decreases from *1* to *0.2*, the force magnitude

increases linearly from *0* to *10*. For a point in the D_{min} of a force field ($\rho < 0.2$), the maximum force F_{max} will be attained.

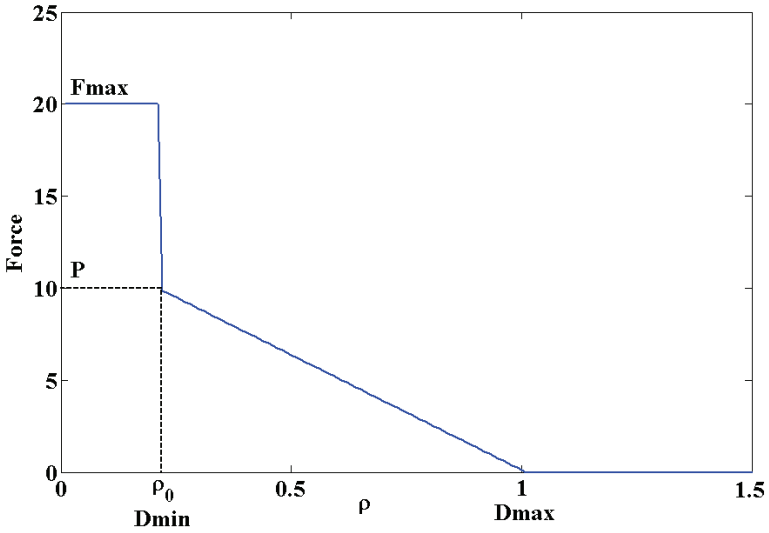


Figure 3-4 The effect of ρ on force magnitude

Figure 3-5 illustrates how the parameters of F^2 determine the coverage of a force field. Some of the parameters of four robots involved are listed in Table 3-2. Other parameters for all robots are: $k = 5$, $\rho_0 = 0.2$, $P = 10$, $C = 1.25$, $v_{max} = 0.08 m/s$. Robot entities are denoted by circulars for simplifying the illustration. The coverage areas of their force fields are shown by the corresponding force contours, in which the outermost force contours and the innermost force contours denote their D_{max} and D_{min}, respectively.

Robot A is stationary with radius $R_r = 1m$ and speed $v_r = 0$. From Equation (3-10) to (3-12), $D_{max} = D_{min} = 0$, which means D_{max} and D_{max} are coincident with its perimeter. Robot B and Robot C are two robots with the same speed $v_r = 0.04 m/s$ but different radii ($R_r = 0.5m$ and $R_r = 1m$ respectively). From Equation (3-12), Robot C has a larger force field than Robot B. Robot D is a robot with $R_r = 0.5m$ and $v_r = 0.07 m/s$. Its force field covers more area than Robot B since Robot D has a higher speed although both robots have the

same size. Note also that a robot's force field covers more area in its moving direction than in other directions.

In summary, the force field of a robot in the F^2 method focuses on its moving direction and varies with its status, including position, speed, volume, environmental factors and so on.

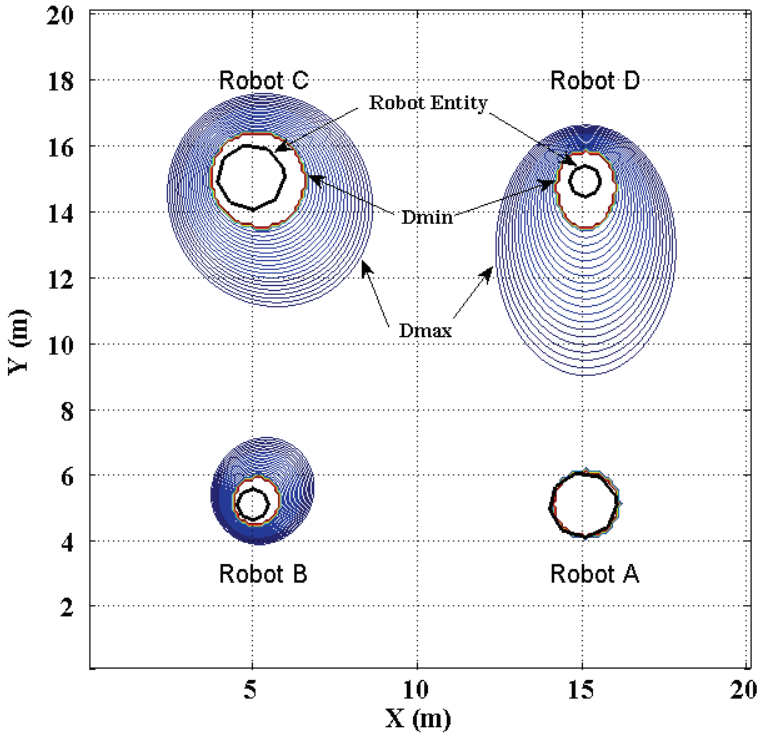


Figure 3-5 Force field: a robot's force field covers more area in its moving direction than in other directions

	Location	R_r (m)	v_r (m/s)	θ_r
A	(15,5)	1	0	0
B	(5,5)	0.5	0.04	π/3
C	(5,15)	1	0.04	-π/3
D	(15,15)	0.5	0.07	-π/2

Table 3-2 Some parameters of four robots

3.3.2 Attractive Force

When a robot is undertaking a particular task, for example, travelling from start point (X_s , Y_s) to goal point (X_g , Y_g), a virtual attractive force which attracts this robot from the start point to the goal point is generated (Figure 3-6). The attractive force, denoted by $\mathbf{F}_{\mathbf{att\_g}}$, directs to the goal point from the centre of the robot and its magnitude can be assumed to be a constant. It drives a robot to its destination (X_g, Y_g). The magnitude of this attractive force is given by:

$$\left|\mathbf{F}_{\mathbf{att\_g}}\right| = Q \tag{3-17}$$

where Q is a positive constant scalar which determines the magnitude of the attractive force. The attractive force directs the robot to the goal point from the centre of the robot and can be assumed to be a constant. Furthermore, the orientation of the attractive force in the global reference frame is given by

$$\angle\mathbf{F}_{\mathbf{att\_g}} = arctan\left(\frac{Y_g - Y_r}{X_g - X_r}\right) \tag{3-18}$$

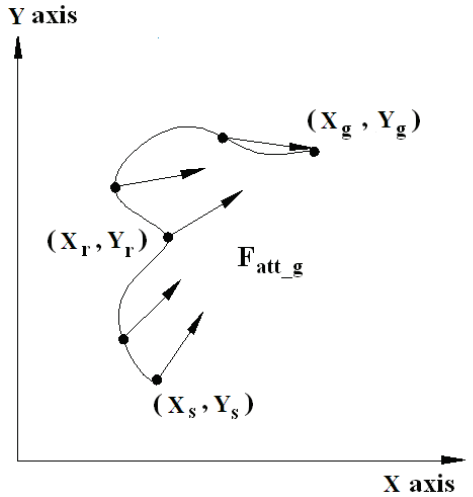


Figure 3-6 Attractive force

3.3.3 Repulsive Force

When a robot approaches an obstacle or other robots' force fields, its force field is suppressed and the robot will be repelled by the virtual reaction forces from the obstacle or other robots. These interactions can be categorized into interactions with obstacles and interactions with other robots. For the interactions between a robot and an obstacle, the repulsive force is expressed as $\mathbf{F}_{\mathbf{rep\_o}}$. For the interactions between two robots, the repulsive force is denoted by $\mathbf{F}_{\mathbf{rep\_j}}$. Calculation of these forces will be described below.

A. Interactions Between a Robot and Obstacles

An obstacle locating in a robot's force field will have influence on this robot's movement. This influence is defined as a virtual repulsive force acting on this robot from the obstacle, denoted by $\mathbf{F}_{\mathbf{rep\_o}}$. The magnitude of this repulsive force can be calculated by:

$$\left|\mathbf{F}_{\mathbf{rep\_o}}\right| = \left|\mathbf{F}_{\mathbf{rep\_r}}\right| \tag{3-19}$$

In Figure 3-7, the dark circle denotes a robot's entity (solid line). The D_{max} and D_{min} of this robot are shown in outer dashed ellipse and inner dashed ellipse, respectively. Point A is a point on the obstacle's surface where the largest repulsive force is attained. The dashdotted ellipse denotes the interaction contour (shown in a dashdotted line) intersecting with the surface of the obstacle on point A. Two possible options in determining the direction of the reaction force are given below:

Option 1: the reaction force directs from the interaction point to the centre of robot, $\mathbf{F}_{\mathbf{rep}}$ as shown in Figure 3-7.

Option 2: the direction of the reaction force is along the normal line of interaction contour at the interaction point A, $\mathbf{F}_{\mathbf{rep}}^{*}$ as shown in Figure 3-7.

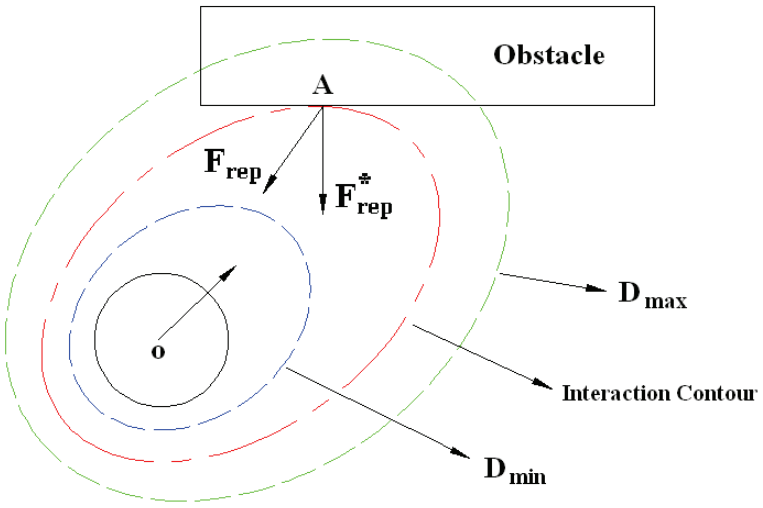


Figure 3-7 Reaction force between a robot and an obstacle

B. Interactions Between Robots

The interaction between two robots can be considered as the interactions between their force fields. In Figure 3-8, the robots' entities are denoted by circles (solid lines), and D_{max} and D_{min} are shown in outer dashed ellipses and inner dashed ellipses, respectively. The dotted ellipses show the interaction contours of the two robots' force fields, where a robot's D_{max} intersects with an inner contour of another robot's force field.

The magnitude of repulsive force which a robot suffers from another robot is defined as equal to the magnitude of its own force field at the interaction point.

$$\left|\mathbf{F}_{\mathbf{rep\_j}}\right| = \left|\mathbf{F}_{\mathbf{rep\_r}}\right| \tag{3-20}$$

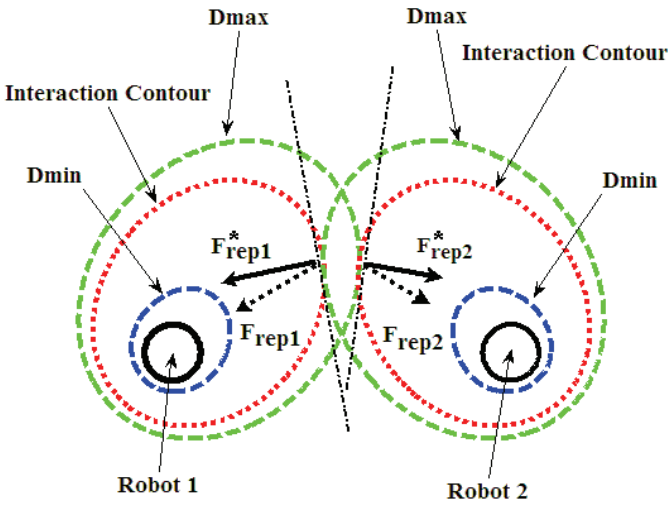


Figure 3-8 Reaction forces between two robots

Similar to the robot-obstacle interaction, there are two possible ways to determine the direction of a reaction force:

Option 1: the reaction force directs from the interaction point to the centre of a robot, $\mathbf{F}_{\mathbf{rep1}}$ and $\mathbf{F}_{\mathbf{rep2}}$ as shown in Figure 3-8.

Option 2: the reaction force is along the normal line of the interaction contour at the interaction point, $\mathbf{F}^{*}_{\mathbf{rep1}}$ and $\mathbf{F}^{*}_{\mathbf{rep2}}$ as shown in Figure 3-8.

The effects of the two directions of reaction force on the performance of collision avoidance and multi-robot motion coordination will be studied and compared in Section 3.5

3.4 Canonical Force Field Method

Based on the definition of force field, attractive force and repulsive force, a Canonical Force Field method (CF^2) is presented in this section.

When a robot moves in an environment, it may be acted on by two kinds of forces, including $\mathbf{F}_{\mathbf{att\_g}}$ which attracts the robot to its goal, and $\mathbf{F}_{\mathbf{rep}}$ which includes the repulsive forces from obstacles ($\mathbf{F}_{\mathbf{rep\_o}}$) and other robots ($\mathbf{F}_{\mathbf{rep\_j}}$). In the CF^2 method, a robot is assumed to travel with a constant speed and its direction of motion is determined by the resultant force (denoted by $\mathbf{F}_{\mathbf{total}}$):

$$\mathbf{F}_{\mathbf{total}} = \mathbf{F}_{\mathbf{att}} + \mathbf{F}_{\mathbf{rep}} = \mathbf{F}_{\mathbf{att\_g}} + \sum_{i=1}^{n} \mathbf{F}_{\mathbf{rep\_o\_i}} + \sum_{l=1}^{m} \mathbf{F}_{\mathbf{rep\_j\_l}} \tag{3-21}$$

where n is the number of obstacles, m is the number of other robots, $\mathbf{F}_{\mathbf{rep\_o\_i}}$ is the repulsive force from static obstacle i, and $\mathbf{F}_{\mathbf{rep\_j\_l}}$ is the reaction repulsive force from robot l. The robot then moves along the direction given by $\mathbf{F}_{\mathbf{total}}$.

Consider a robot located at (X_r, Y_r) with angle θ_r to the X-axis of the global coordinate at time $t = s$, its current moving direction θ_s is given by

$$\theta_s = \theta_r \tag{3-22}$$

In the CF^2 method, this robot's next moving direction θ_{s+1} is the direction determined by $\mathbf{F}_{\mathbf{total}}$.

$$\theta_{s+1} = \angle \mathbf{F}_{\mathbf{total}} \tag{3-23}$$

The robot's angular speed $\boldsymbol{\omega}$ should satisfy:

$$\boldsymbol{\omega} = \frac{\theta_{s+1} - \theta_s}{\Delta t} \tag{3-24}$$

Thus the motion command is given by

$$\mathbf{u} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} v \\ \dfrac{\theta_{s+1} - \theta_s}{\Delta t} \end{bmatrix} \tag{3-25}$$

3.5 Case Studies

The simulation studies in this section are designed for two purposes. Firstly, simulations are carried out to prove the feasibility of the CF^2 method in both single robot cases and multi-robot cases. Secondly, the options in repulsive force direction selection introduced in Section 3.3.3 are evaluated.

3.5.1 Single Robot Cases

For the single robot cases, simulations are carried out in a map shown in Figure 3-9 and Figure 3-10. A robot is supposed to travel from start position $S1(1,2)$ to goal $G1(9.5,9)$. An obstacle locates on its way to the goal (shown by the rectangle). Parameters used in this simulation are $\rho_0 = 0.2$, $k = 5$, $v_{max} = 0.04m/s$, $v_r = 0.03m/s$, $C = 1.25$, $P = 20$, $Q = 20$, $T_p = 1$, $R_r = 0.2m$. In the following simulation snapshots, the green ellipse and the blue ellipse show a robot's D_{max} and D_{min}, respectively. The red ellipse shows the interaction contour of its force field with the obstacle. The black circle denotes a robot's entity.

When the direction of repulsive force is assumed to be the one from the interaction point to the robot centre (Option 1), i.e., as $\mathbf{F_{rep}}$ in Figure 3-7, the robot trajectory obtained by the CF^2

method is shown in Figure 3-9 with the total travel distance of $34.74m$ (*1158* time steps, each time step is $1s$). If the force direction is along the normal line (Option 2), i.e. the direction of $\mathbf{F}_{\mathbf{rep}}^{*}$ in Figure 3-7, the trajectory obtained by the CF^2 method is the one shown in **Figure 3-10**. The total travel distance is $14.01m$ (*467* steps), which is much shorter than that of the previous one.

For the purpose of explanation, this robot's trajectories in a small area which is denoted by small squares in Figure 3-9 and Figure 3-10 (with $x \in [8,8.3]$ and $y \in [3.4,3.7]$) are shown in Figure 3-13 and Figure 3-15, respectively. Each green dot represents one step.

With Option 1, the robot travels *90* steps to pass this area (Figure 3-13). Figure 3-11 illustrates the situation of this robot at its 450th step. Here the forces acting on this robot include an attractive force ($\mathbf{F}_{\mathbf{att\_g}}$) from its destination $G1(9.5,9)$ and a repulsive force ($\mathbf{F}_{\mathbf{rep\_o}}$) from the obstacle. The interaction contour (red ellipse) nearly overlaps with the D_{min} (blue ellipse), so the magnitude of the repulsive force is very large. As a result, the resultant force $\mathbf{F}_{\mathbf{total}}$ has the direction shown in Figure 3-11. In the CF^2 method, a robot's next motion direction is determined by the direction of the resultant force acting on it, so this robot will move along the direction of $\mathbf{F}_{\mathbf{total}}$. Its pose in step 451 is shown in Figure 3-12, in which this robot's orientation changes noticeably from the previous step. As a result, the repulsive force from the obstacle changes and its magnitude reduces significantly. The direction of resultant force $\mathbf{F}_{\mathbf{total}}$ is shown in Figure 3-12. This generates the to-and-fro behaviours shown in Figure 3-13.

Option 2 provides a much better result with *10* steps only in the analysed area shown in Figure 3-15. Figure 3-14 gives a snapshot of its pose in step 251. The repulsive force $\mathbf{F}_{\mathbf{rep\_o}}$ is perpendicular to the surface of the obstacle and pushes the robot away. There is no significant change in the direction of the resultant force in the analysed area, so the robot's path is quite smooth.

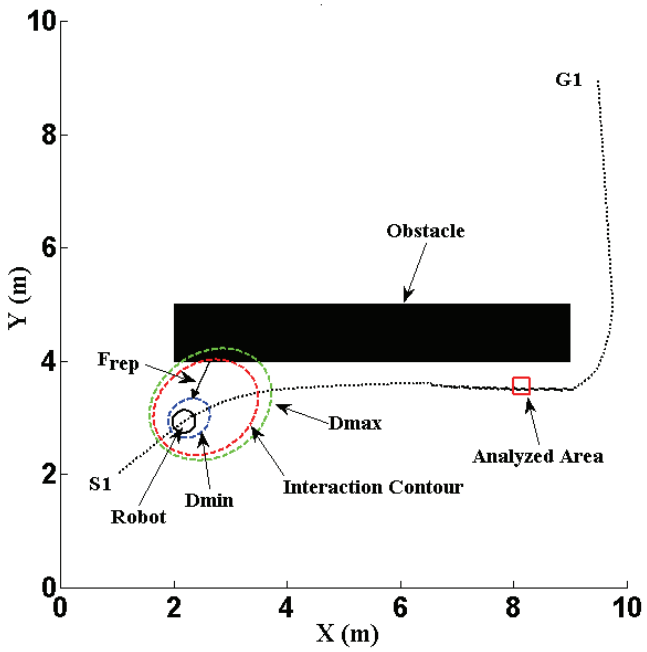


Figure 3-9 CF^2 for single robot Case 1: the direction of a repulsive force is from the interaction point to the robot centre (Option 1)

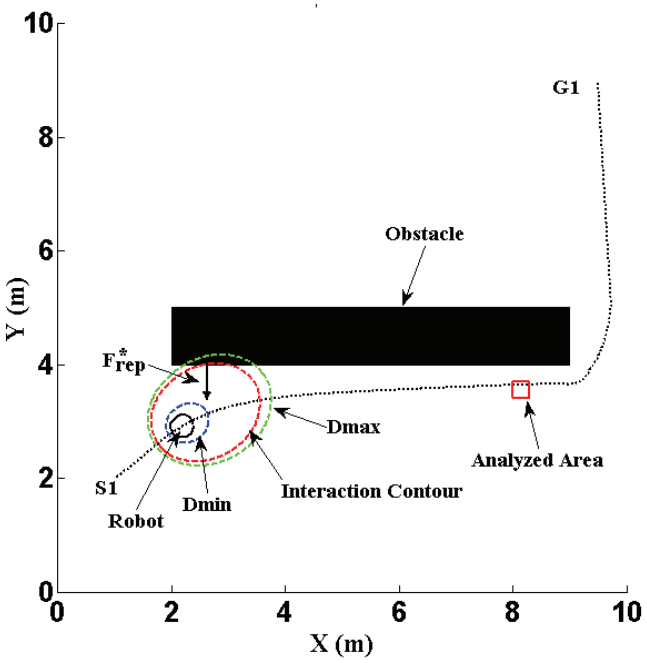


Figure 3-10 CF^2 for single robot Case 2: the repulsive force direction is along the normal line of interaction contour at the interaction point (Option 2)

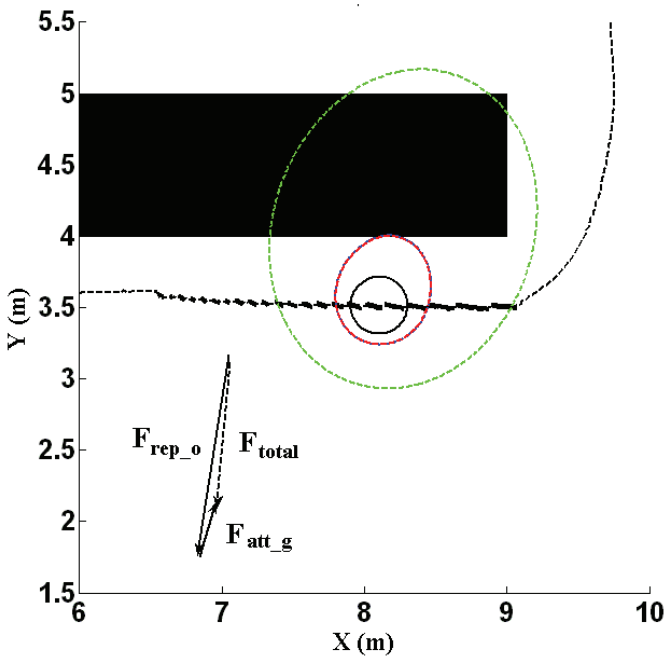


Figure 3-11 CF^2: single robot Case 1 (snapshot 1)

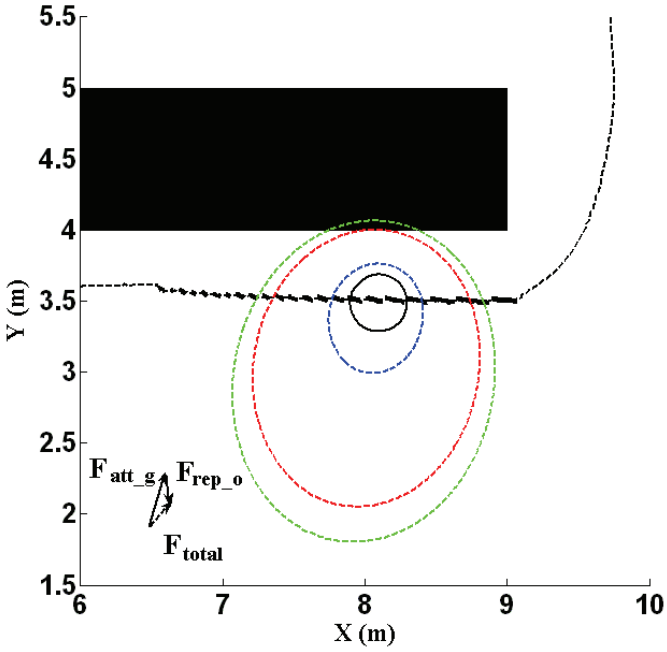


Figure 3-12 CF^2: single robot Case 1 (snapshot 2)

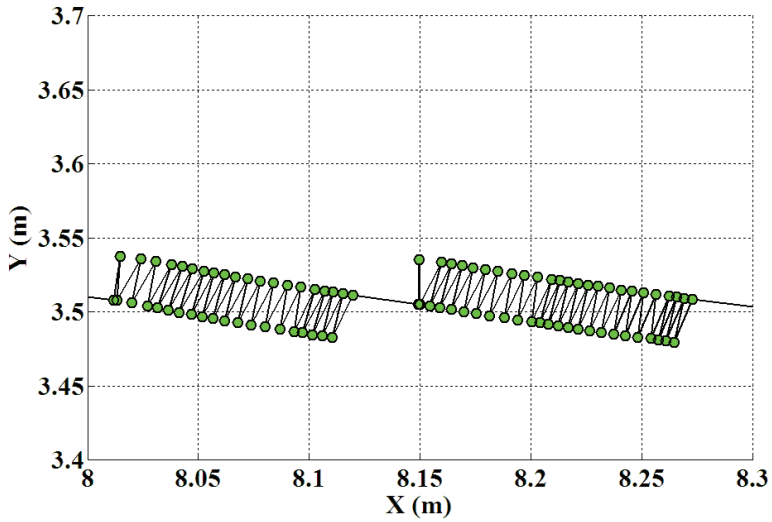


Figure 3-13 CF^2: single robot Case 1 (trajectories in the analysed area)

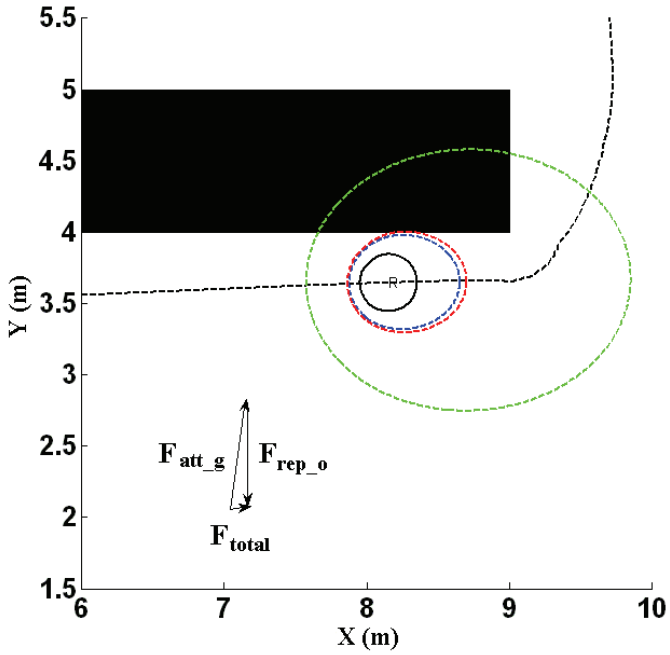


Figure 3-14 CF^2: single robot Case 2 (snapshot 1)

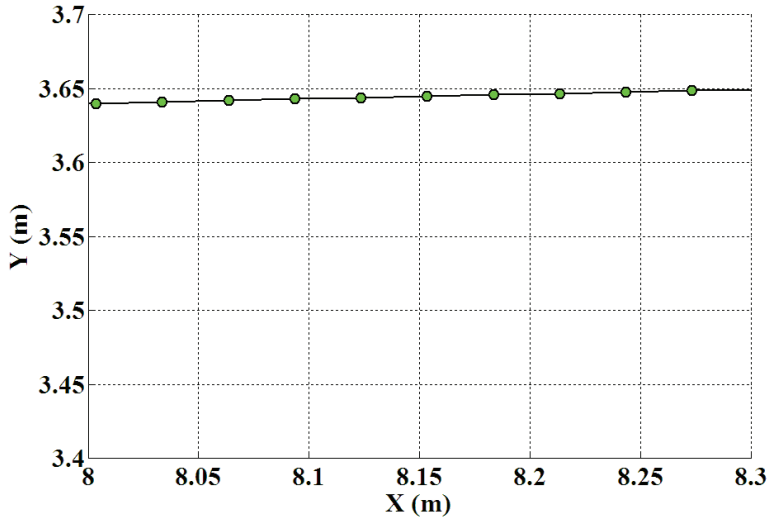


Figure 3-15 CF^2: single robot Case 2 (trajectories in the analysed area)

From the simulation results, it can be seen that Option 2 provides better results than Option 1. Thus, for the interaction between a robot and an obstacle, the direction of a repulsive force is selected to be along the normal line of interaction contour at interaction point (Option 2).

3.5.2 Multiple Robots Cases

The proposed CF^2 methods have also been applied in multi-robot motion planning problems. Simulations presented in this section were carried out in an indoor environment using the CF^2 method. These simulations consist of four robots travelling in an atrium area. Each robot was assigned a task travelling from its start point to its goal position. Simulations have been designed to:

- Test the feasibility of the CF^2 method on multi-robot motion planning and collision avoidance.
- Evaluate the two options on repulsive force direction between two robots proposed in Section 3.3.3.

- Highlight the effect of robot priority in multi-robot motion planning and collision avoidance using the CF^2 method.

Five simulations have been conducted. Simulations 1 and 2 were designed to let each robot travel one by one without other robots interfering. The problem then becomes four single-robot navigation problems. In Simulation 1, the direction of repulsive force between a robot and obstacles was selected as starting from the interaction point to a robot's centre (Option 1). In Simulation 2, the repulsive force direction was selected to be along the normal line of the interaction contour at the interaction point (Option 2). Simulations 3 and 4 consisted of four robots' navigation in the environment. Robots met each other in the centre area of the room and managed to avoid collisions. The two options on the directions of repulsive forces between robots were tested in Simulations 3 (with Option 1) and 4 (with Option 2). In Simulation 5, Robot 2 was assigned a higher priority than other robots. Thus Robot 2 had priority in the collision avoidance. This simulation demonstrates the effect of priority on motion planning and collision avoidance using the CF^2 method.

The dimension of this simulation environment is $20m$ by $20m$. Some common parameters for the robots are: $R_r = 0.2m$, $\rho_0 = 0.2$, $k = 5$, $v_{max} = 0.04m/s$, $C = 1.25$, $P = 20$, $Q = 20$. The start and goal positions are marked as S_i and G_i (i=*1, 2, 3, 4*), respectively, as listed in Table 3-3. The straight-line distances between S_i and G_i are given by D_{real} in the 5th column of Table 3-3. In the figures below, the paths of the four robots obtained by the CF^2 method are shown in solid lines, dotted lines, dashdotted lines and dashed lines, respectively. D_{max} and D_{min} for each robot are shown in dashed ellipses. Solid circles denote the robots' entities. A robot is considered to be in its goal position if the distance from its centre to goal is less than its radius.

A. Simulation 1: single robot navigation with repulsive force direction Option 1

Let each robot travel one after another from its start point to its goal without other robots interfering. With force direction Option 1 in Section 3.3.3 A, the resultant paths are shown in

Figure 3-16 and their corresponding path lengths are denoted by D_1 in the 6th column of Table 3-3. The path lengths for Robots 1, 2, 3 and 4 are *23.7m*, *20.02m*, *18.825m* and *19.6m*, respectively.

B. Simulation 2: single robot navigation with repulsive force direction Option 2

Let each robot travel separately (one after another) with force direction Option 2 in Section 3.3.3 A, the resultant paths are shown in Figure 3-17 and their corresponding path lengths are denoted by D_2 in the 7th column of Table 3-3. The path lengths for Robots 1, 2, 3 and 4 are *21.1m*, *19.84m*, *18.25m* and *19.525m*, respectively, which are shorter than the path lengths obtained from Option 1 (*23.7m*, *20.02m*, *18.825m* and *19.6m*, correspondingly). Note that the straight-line distance from $S_2(18,4)$ to $G_2(2,1)$ is *20m*, which is slightly longer than the path length of Robot 2 (*19.84m*) in this simulation. The reason for this is that the robot is considered to arrive at its goal if the distance from its centre to the goal is less than its radius (*0.2m*).

As has been proved by simulations with a single robot, Option 2, where the repulsive force is set to be along the normal line of interaction contour at interaction point, gives better results than Option 1, where the repulsive force directs from the interaction point to the robot's centre.

C. Simulation 3: multi-robot navigation with repulsive force direction Option 1

The third simulation consists of four robots' navigation using the CF^2 method with reaction force direction Option 1 in Section 3.3.3, i.e. the repulsive force between two robots directs from the interaction point to a robot's centre. The resultant paths are shown in Figure 3-18. When robots approached the centre area and were close to other robots, they managed to avoid collisions with each other. Figure 3-18 shows a snapshot of robots moving in this case. The path lengths for Robots 1, 2, 3 and 4 are *22.075m*, *21.78m*, *19.175m* and *20.25m*, respectively (D_3 in the 8th column of Table 3-3).

D. Simulation 4: multi-robot navigation with repulsive force direction Option 2

The fourth simulation has been carried out using the CF^2 method with reaction force direction Option 2, i.e., the repulsive force is along the normal line of interaction contour at the interaction point. The resultant paths are shown in Figure 3-19. Their path lengths are given by D_4 in the 9th column of Table 3-3. Compared with D_3 in Simulation 3, the path length of Robot 1 decreased from *22.075m* to *20.65m*. The path length of Robot 2 decreased from *21.78m* to *21.66m*. The path length of Robot 3 reduced from *19.175m* to *18.75m*. The path length of Robot 4 increased slightly from *20.35m* to *20.375m*. In general, the force direction Option 2 provides better results than Option 1.

E. Simulation 5: multi-robot navigation with priorities

The priorities of all robots in the above simulations were set to be the same $T_p = 1$. In the fifth simulation, Robot 2 was assigned a higher priority $T_p = 2$, while other robots' task priorities were set to be 1. Their paths are shown in Figure 3-20 and the path lengths are given by D_5 in the 10th column of Table 3-3. Note that in the scene shown in Figure 3-20, the force field of Robot 2 covers more area than other robots because of its higher task priority. So Robot 2 got priority during collision avoidance collaboration with other robots. Robot 3 was being forced to change its orientation and give way to Robot 2. The path length of Robot 2 is reduced from *21.66m* to *21.26m*, but other robots' path lengths, e.g., Robot 1 and 3, are increased correspondingly. This case proves the effect of T_p in multi-robot cases.

i	S_i (x,y)	G_i (x,y)	v_r (m/s)	D_{real} (m)	D_1 (m)	D_2 (m)	D_3 (m)	D_4 (m)	D_5 (m)
1	18,1	11,19	0.025	19.313	23.7	21.1	22.075	20.65	22.075
2	18,4	2,1	0.02	20	20.02	19.84	21.78	21.66	21.26
3	1,4	17,12	0.025	17.889	18.825	18.25	19.175	18.75	25.575
4	18, 18	7,2	0.025	19.416	19.6	19.525	20.25	20.375	19.85

Table 3-3 CF^2: simulations results

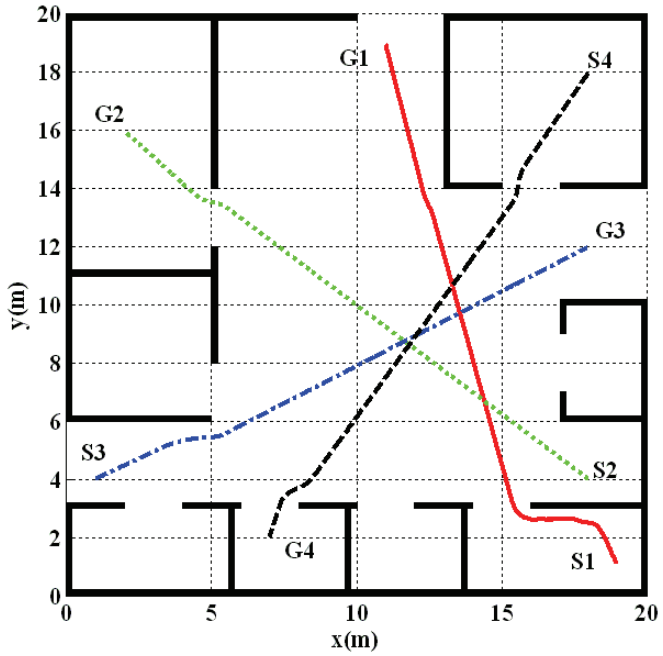


Figure 3-16 CF^2: individual paths for four robots with force direction Option 1 (D_1)

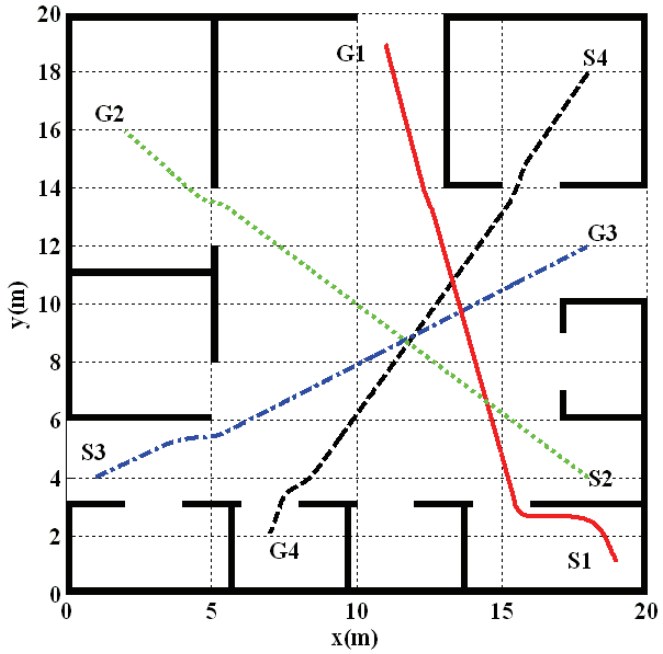


Figure 3-17 CF^2: individual paths for four robots with force direction Option 2 (D_2)

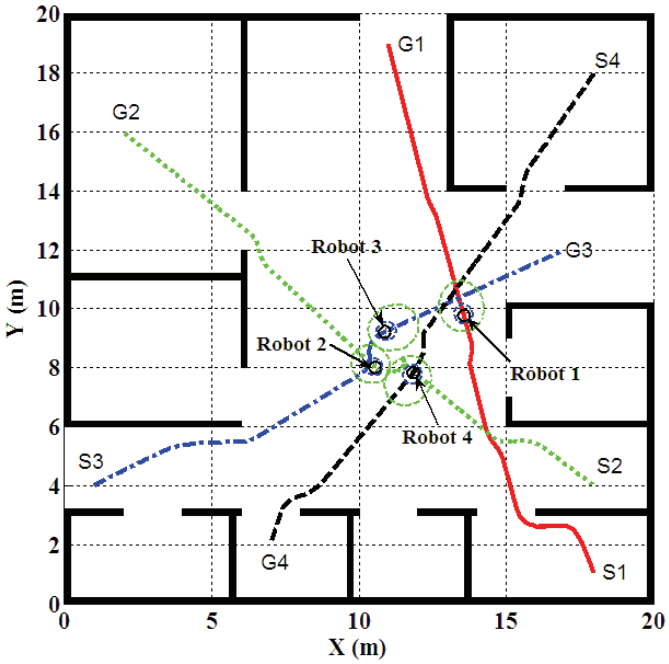


Figure 3-18 CF^2: multi-robot navigation with force direction Option 1 (D_3)

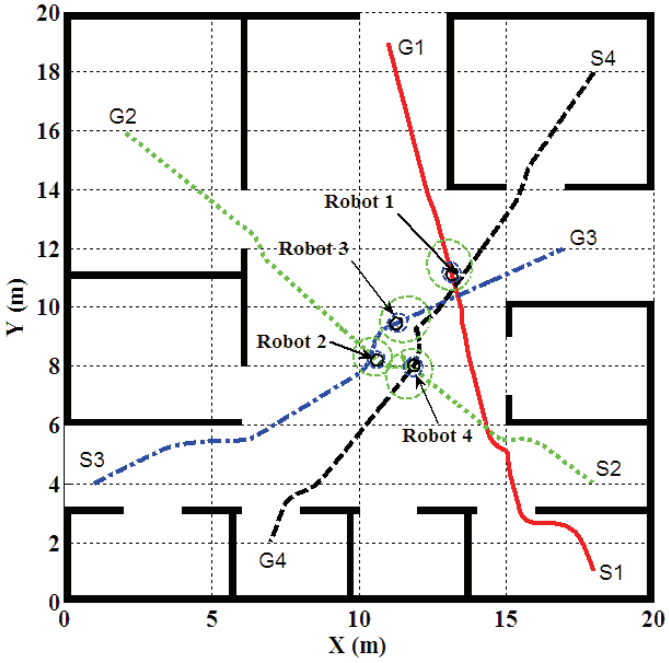


Figure 3-19 CF^2: multi-robot navigation with force direction Option 2 (D_4)

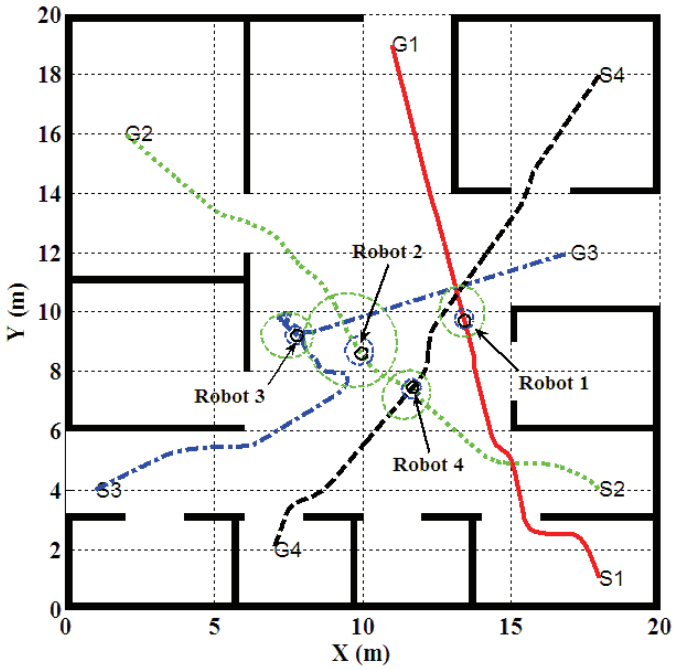


Figure 3-20 CF^2: multi-robot navigation with priorities (D_5)

3.6 Algorithm Efficiency Evaluation

All simulations in the chapter have been carried out on a computer with a CPU of Centrino 1.6GHz and 1G RAM, running Matlab 7 in Window XP.

Table 3-4 lists the computation time of each robot in the simulation presented in Section 3.5.2 D (Simulation 4). For Robot 1, the time required to plan the whole path is 20.93s. The path consists of 816 steps. The path of Robot 2 consists of 1089 steps and was calculated in 28.781s. The path of Robot 3 consists of 760 steps and was calculated in 19.378s. The path of Robot 4 consists of 830 steps and takes 21.821s to calculate.

The time for a robot to plan its motion varies with the complexity of its surrounding environment, that is, it takes a longer time when there are more obstacles and other robots in its vicinity. It is found that in this simulation, the time spent for a single step motion plan is between 0.02s to 0.05s.

A six-robot navigation case has also been carried out and compared with the four-robot case to test the performance of the CF^2 method with increasing number of robots. The dimension of this simulation environment and the parameters for the robots are the same as in Section 3.5.2, case D: $20m$ by $20m$, $R_r = 0.2m$, $\rho_0 = 0.2$, $k = 5$, $v_{max} = 0.04m/s$, $C = 1.25$, $P = 20$, $Q = 20$. The tasks for Robots 1, 2, 3 and 4 are the same as well: Robot 1 is supposed to travel from position (18, 1) to (11, 19), Robot 2 travels from (18, 4) to (2, 1), Robot 3 travels from (1, 4) to (17, 12) and Robot 4 travels from (18, 18) to (7, 2). Robots 5 and 6 have been added to the simulation. Robot 5 is supposed to travel from positions (2, 7) to (18, 8) and Robot 6 travels from (18, 8) to (2, 7). The speed of Robot 5 and 6 is set to be 0.025m/s, which is different from Robot 2 (0.02m/s). The task priorities of all robots are the same (i.e. 1). The paths of Robots 1, 2, 3, 4, 5 and 6 are shown in Figure 3-21 and denoted by a red solid line, green dotted line, blue dash dot line, black dashed line, blue dotted line and black solid line, respectively. Table 3-5 gives the simulation results.

The simulation results of Robot 1 are listed in the second row of Table 3-5. The path length and total steps of the path are almost the same as those of the four-robot case (the second row in Table 3-4). In the four-robot case, the path is 20.65m in length and consists of 816 steps. In the six-robot case, the path is 20.375m in length and consists of 815 steps. However, the computation time to plan the path for Robot 1 has increased from 20.93s in the four-robot case to 30.694s (an increase of 46.65%). The computation time for Robot 2 increases from 28.781s to 42.091s (an increase of 46.25%). The computation time for Robots 3 and 4 increase 72.85% (from 19.378s to 34.289s) and 48.28% (from 21.821s to 32.357s), respectively. The reason for the increases in computation time is that there are more robots in the working environment, so it takes longer time for a robot to plan its motions.

The simulation results show that the CF^2 method is very efficient and suitable for real time motion planning and collision avoidance. In the CF^2 and other F^2-based methods to be presented in this thesis, a robot plans its next motion indepently. It does not need to search the whole space as some conventional approaches do, but only obstacles and other robots' force fields which are in the coverage of its force field are considered in the collision avoidance process. Robots in the F^2 method work in a decentralized way and the increase of the number of robots only results in an almost linear increase to the computational burden.

Robot	Number of Planning Steps	Total Computation Time (s)	Path Length (m)
1	816	20.930	20.65
2	1089	28.781	21.78
3	760	19.378	19.175
4	830	21.821	20.25

Table 3-4 Computation time: a four-robot case (Simulation 4)

Robot	Number of Planning Steps	Total Computation Time (s)	Time Increase	Path Length (m)
1	815	30.694	46.65%	20.375
2	1099	42.091	46.25%	21.98
3	875	34.289	76.85%	21.875
4	841	32.357	48.28%	21.025
5	726	29.723	N/A	18.15
6	766	29.384	N/A	19.15

Table 3-5 Computation time: a six-robot case

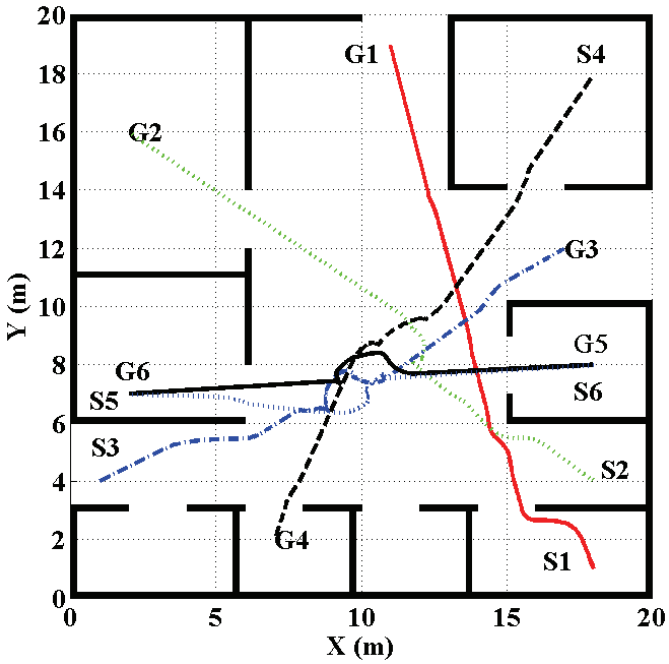


Figure 3-21 CF^2: a six-robot case

3.7 Conclusions

In this chapter, the basic concept of a novel force field (F^2) is detailed. Compared to other existing methods, a robot's force field in the F^2 method is constructed based on its status. The coverage of a force field is determined by the robot's size (R_r), speed (v_r), priority (T_p), environment factor (C) and a scalar constant k. The interaction of this force field with other robots and obstacles within its work space provides a feasible and natural way for motion planning and collision avoidance.

The Canonical Force Field (CF^2) is based on the basic concept of the force field, in which a robot is assumed to travel with a constant speed and its motion direction is determined by the resultant force acting on it. Simulations have been carried out to verify the CF^2 method and to determine appropriate repulsive forces to guide robot motion, that is, the direction of repulsive force is selected to be along the normal line of the interaction contour at the interaction point. Results have shown that the CF^2 method is applicable in motion planning and coordination for

both single robot and multi-robot cases. A robot using the CF^2 method works in a decentralized way: each robot determines its next motion independently. The increase of the number of robots in the work space will not exponentially increase computation burden as some conventional approaches did.

The CF^2 method is especially suitable for

- Real-time applications. Due to the simplicity of the CF^2 method, it is fast and especially suitable for real-time applications.
- Robots with limited motion control and computing capabilities.
- Applications in which there is large free space in the work environment so that a robot's acceleration and deceleration process can be safely ignored.

Chapter 4
Development of Force Field Algorithms

Chapter 3 has defined the basic concept of the proposed Force Field (F^2) and then introduced the Canonical Force Field (CF^2) method for robot motion planning and coordination. The concept of the F^2 method is to generate a force field for each robot that is continuously changing according to its status including travelling speed, dimension, priority, location and environmental factors. The interactions among a robot's force field and the environment provide a natural way for motion planning and collision avoidance while robots are moving in the environment. In the CF^2 method, a robot travels at a constant speed and its moving direction is determined by the resultant force acting on it.

This chapter presents further developments on the F^2 based algorithms. Three algorithms which are based on the concept of F^2 and especially designed for different applications are described in detail. Section 4.1 presents the Variable Speed Force Field method (VSF^2), in which a robot's speed adaptively changes based on environmental information and its own status. The VSF^2 method takes a robot's dynamics and kinematics constraints into consideration in motion planning, which greatly broadens the applications of the F^2 method. Section 4.2 presents the Subgoal-Guided Force Field method (SGF^2) which improves the F^2 method by changing the way to generate attractive force. In the SGF^2 method, an attractive force is generated from a chosen subgoal instead of a robot's global goal as in the CF^2 method and VSF^2 method. The subgoal is updated continuously in a robot's movement based on latest sensor information. The SGF^2 method is suitable for real-time motion planning and collision avoidance in partially known and dynamically changing environments. Section 4.3 describes the Dynamic Variable Speed Force Field method ($DVSF^2$). In the $DVSF^2$ method, an intermediate waypoint is selected on a pre-planned path given by a global planner. A robot using the $DVSF^2$ method is then capable of tracking on the pre-planned optimal path while avoiding collisions. The $DVSF^2$ method is particularly suitable for real-time collision avoidance in a dynamically changing environment with unexpected obstacles. Simulations have been carried out to verify the feasibility and performance of the proposed methods.

In the research scope of this chapter, it is assumed that robots move in a 2D space and each robot is aware of its current location and goal position. Robots are capable of using onboard sensors to sense their vicinities and obstacles. In the multi-robot cases, it is assumed that robots are equipped with communication devices so that they are aware of the status of other robots, including priorities, velocities, locations, sizes and geometries.

4.1 Variable Speed Force Field Method

In the CF^2 method, a robot is assumed to travel with constant speed and its moving direction is straightforwardly determined by the attractive force and the repulsive forces from obstacles and other robots. In many applications, a robot is required to be able to change its travel speed. Robot orientation should also be changed but it has to satisfy the robot's kinematic and dynamic constraints. Dynamic and kinematic characteristics must therefore be taken into account in motion planning.

4.1.1 The Concepts of the Variable Speed Force Field Method

The Variable Speed Force Field (VSF^2) method adopts the definitions of force field, attractive force and repulsive force of the CF^2 method but changes the way to decide a robot's motion in CF^2. In the VSF^2 method, a robot's motion is determined by the attractive force and repulsive forces acting on it. In particular, its translational acceleration is a function of the attractive force and repulsive forces acting on it and its angular acceleration is a function of force moments acting on it, as given by Newton's Second Law of Motion.

Define a fixed global reference frame $O(X,Y)$ and a moving local reference frame $o(x,y)$ attached to a robot's body, as shown in Figure 4-1. Let $\mathbf{v}$ be the robot's translational velocity and $\boldsymbol{\omega}$ its angular velocity. Assume that a robot has mass m and inertia I about its centre of mass. Its translational acceleration $\dot{\mathbf{v}}$ and angular acceleration $\dot{\boldsymbol{\omega}}$ in the moving frame $o(x,y)$ are given by:

$$m\dot{\mathbf{v}} = \mathbf{F}_{att\_g} + \sum \mathbf{F}_{rep\_o} + \sum \mathbf{F}_{rep\_l} \tag{4-1}$$

$$I\dot{\omega} = \mathbf{M}_{att\_g} + \sum \mathbf{M}_{rep\_o} + \sum \mathbf{M}_{rep\_l} \tag{4-2}$$

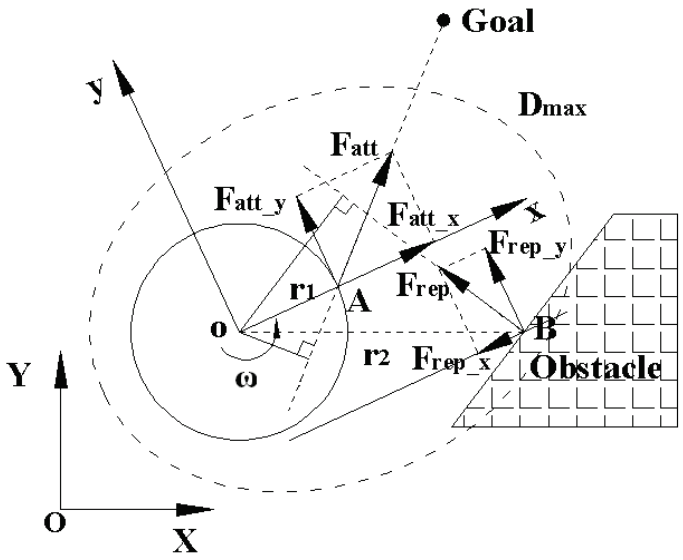


Figure 4-1 VSF^2 method parameters

where $\mathbf{F}_{att\_g}$ is the attractive force from the goal position, $\mathbf{F}_{rep\_o}$ are repulsive forces from obstacles and $\mathbf{F}_{rep\_l}$ the repulsive forces from other robots. $\mathbf{M}_{att\_g}$, $\mathbf{M}_{rep\_o}$ and $\mathbf{M}_{rep\_l}$ are force moments generated by $\mathbf{F}_{att\_g}$, $\mathbf{F}_{rep\_o}$ and $\mathbf{F}_{rep\_l}$, respectively.

Figure 4-1 illustrates a case containing a robot and an obstacle. In this figure, the circle denotes a robot's entity and the D_{max} of its force field is shown in dashed ellipse. $\mathbf{F}_{att\_x}$ and $\mathbf{F}_{att\_y}$ are the components of the virtual attractive force, respectively. $\mathbf{F}_{rep\_x}$ and $\mathbf{F}_{rep\_y}$ are the components of the virtual repulsive forces, respectively. The robot's acceleration in x and y direction, denoted by $\mathbf{a}_x$ and $\mathbf{a}_y$, are given by

$$m\mathbf{a}_x = \mathbf{F}_{att\_x} + \mathbf{F}_{rep\_x} \tag{4-3}$$

$$m\mathbf{a}_y = \mathbf{F}_{att\_y} + \mathbf{F}_{rep\_y} \tag{4-4}$$

$\mathbf{M}_{\mathbf{att}}$ and $\mathbf{M}_{\mathbf{rep}}$ are the moments generated by the virtual attractive force and repulsive force, respectively, which are given by:

$$\mathbf{M}_{\mathbf{att}} = \mathbf{r}_1 \times \mathbf{F}_{\mathbf{att}} \tag{4-5}$$

$$\mathbf{M}_{\mathbf{rep}} = \mathbf{r}_2 \times \mathbf{F}_{\mathbf{rep}} \tag{4-6}$$

where $\mathbf{r}_1$ is the distance from the centre of robot o to the attractive force ($\mathbf{F}_{\mathbf{att}}$) acting point *A*. $\mathbf{r}_2$ is the distance from the centre of robot o to the repulsive force ($\mathbf{F}_{\mathbf{rep}}$) acting point *B*. It is assumed that the attractive force acts on the robot's front edge *A*, which is the intersection point of the x axis and the robot's body (see Figure 4-1).

It should be noted that there are constraints which need to be satisfied, including the maximum translational speed (v_{max}), translational acceleration ($\dot{v}_{max}$), angular speed (ω_{max}) and angular acceleration $\dot{\omega}_{max}$. These parameters should be chosen based on a robot's dynamic and kinematic characteristics.

$$|\mathbf{v}| \le v_{max} \tag{4-7}$$

$$|\dot{\mathbf{v}}| \le a_{max} \tag{4-8}$$

$$|\boldsymbol{\omega}| \le \omega_{max} \tag{4-9}$$

$$|\dot{\boldsymbol{\omega}}| \le \dot{\omega}_{max} \tag{4-10}$$

4.1.2 Simulations on Variable Speed Force Field Method

Simulations have been carried out to verify the feasibility and performance of the VSF^2 method. In the first simulation, two-robots navigate through a narrow passage. More complicated simulation is conducted in Simulation 2 where four-robots navigate in a complex

indoor environment. The simulation results of the VSF^2 method are compared with those of the CF^2 method.

Amigo robots are used in the simulation studies. The parameters of an Amigo robot (see Figure 4-2) are: the dimension is $33cm$ by $28cm$ and the body clearance is $3cm$. The weight of this robot is $3.6kg$. The maximum speed of an Amigo robot is $1m/s$ and the maximum angular speed is *300* degrees per second ($5\pi/3$ in radian) [136].

Figure 4-2 Amigo robot [136]

A. Simulation 1: Two-Robot Motion Planning and Collaboration

Figure 4-3 shows a scene of two robots passing a corridor. Robot 1 is supposed to travel from $S1(1,5.1)$ to $G1(9,5.4)$ and Robot 2 travels from $S2(9,5)$ to $G2(1,5)$. The robots' entities are denoted by cycles and D_{max} and D_{min} are presented by outer dashed ellipses and inner dashed ellipses, respectively.

Let the robots navigate in this environment using the CF^2 method presented in Section 3.4. Parameters used in this simulation are: $R_r = 0.18m$, $\rho_0 = 0.2$, $k = 5$, $|\omega| \le 5\pi/3$, $v = 0.7m/s$, $v_{max} = 0.75m/s$, $C = 2$, $P = 20$, $Q = 5$, $T_p = 1$. The resultant path of Robot 1 is denoted by solid line and the path of Robot 2 is denoted by dotted line. It is found that the robots failed to pass this corridor.

When the robots approach the corridor, part of the walls (obstacles) will gradually enter the robot force fields. The robots then begin to be acted on by the repulsive forces from the obstacles before they enter the corridor. After the robots enter the corridor, their virtual force fields cover obstacles (solid patches) of both sides, as shown in Figure 4-3. Forces acting on Robot 1 include a repulsive force from Obstacle 1 ($\mathbf{F}_{\mathbf{rep\_A1}}$), a repulsive force from Obstacle 2 ($\mathbf{F}_{\mathbf{rep\_B1}}$), a repulsive force from Robot 2 ($\mathbf{F}_{\mathbf{rep\_21}}$) and an attractive force from its goal ($\mathbf{F}_{\mathbf{att\_1}}$). Similarly, forces acting on Robot 2 include a repulsive force from Obstacle 1 ($\mathbf{F}_{\mathbf{rep\_A2}}$), a repulsive force from Obstacle 2 ($\mathbf{F}_{\mathbf{rep\_B2}}$), a repulsive force from Robot 1 ($\mathbf{F}_{\mathbf{rep\_12}}$) and an attractive force from its goal ($\mathbf{F}_{\mathbf{att\_2}}$). For Robot 1, in the direction of Y axis, the main forces acting on it are the repulsive forces from Obstacle 1 and Obstacle 2 ($\mathbf{F}_{\mathbf{rep\_A1}}$ and $\mathbf{F}_{\mathbf{rep\_B1}}$). These two forces have the same magnitude and are collinear when the robot is in the middle of the corridor. If the robot is not in the middle of the corridor, the two reaction forces will finally 'push' the robot, moving it to the middle.

When the two robots meet in the corridor, according to the CF^2 method, the moving direction of each robot is determined by the resultant force acting on it. Any small change in the orientation causes a significant change to the repulsive forces from obstacles and the robot is forced to turn to another side. Thus severe oscillation in travel direction occurs as shown in Figure 4-4. The unit of x coordinate is $0.01s$. As the result of actions of the reaction forces and attractive forces, the two robots can not pass each other to arrive at their goals.

The VSF^2 method is then applied to solve this problem. Robots, obstacle information and task information are the same as those in the case shown in Figure 4-3. Parameters used in this simulation are: $R_r = 0.18m$, $m = 3.6kg$, $\rho_0 = 0.2$, $k = 5$, $|\boldsymbol{\omega}| \le 5\pi/3$, $v_{max} = 0.75m/s$, $C = 2$, $P = 20, Q = 5$, $T_p = 1$. Robots start with speed $v_r = 0.7m/s$. Paths found by the VSF^2 method are shown in Figure 4-5. Robots' moving directions and speeds are showed in Figure 4-6, where the unit of x coordinate is *0.01s*. The gap width between the two obstacles (black patches) is *1.5m*. Robots' entities are denoted by cycles and D_{max} and D_{min} are denoted by outer dashed ellipses and inner dashed ellipses, respectively. The path of Robot 1 is denoted by a dashed line and path of Robot 2 is denoted by a dotted line.

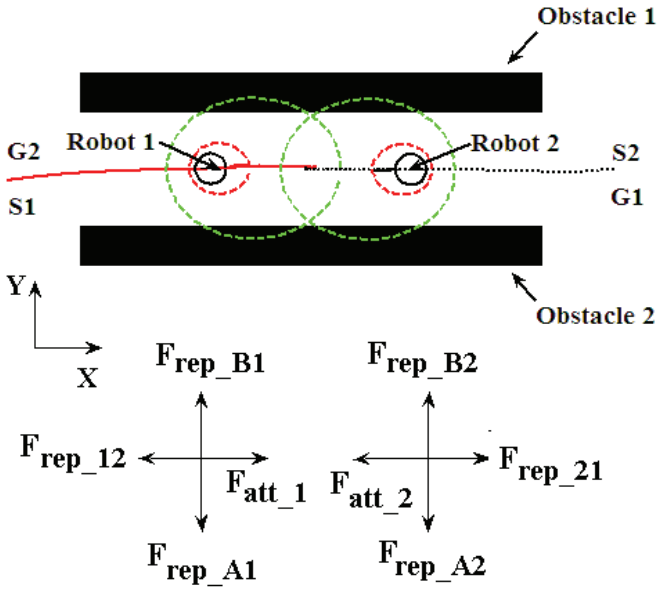


Figure 4-3 A two-robot case with CF^2 method

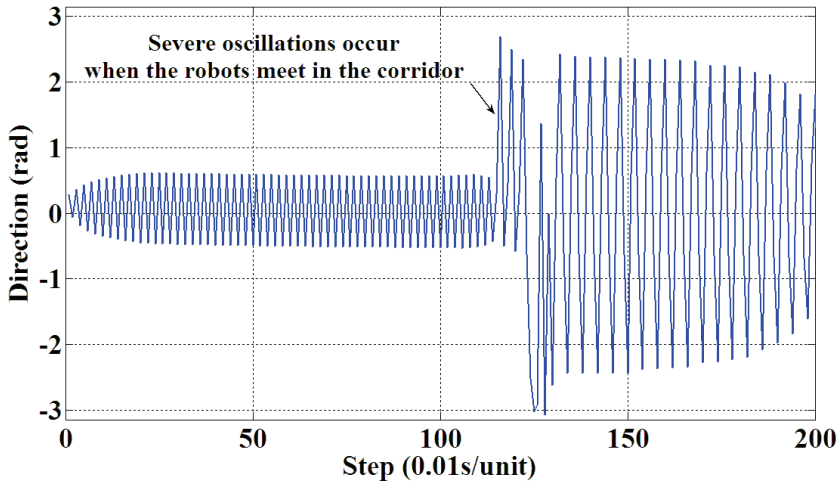


Figure 4-4 Direction oscillation in a two-robot case with CF^2 method

Figure 4-5 shows a snapshot of the two robots' navigation in the narrow corridor at time $4.6s$. Compared with Figure 4-3, the robots speeds are decreased from $0.7m/s$ to $0.6m/s$ (see robots' speed values in Figure 4-6). As a result, the coverage of the robots' force fields becomes smaller and robots can traverse the narrow passages.

After the robots pass each other, the actions of the virtual attractive force from the goal and virtual repulsive forces from the walls accelerate the robots to a higher travel speed while the robots steer away from the obstacles and finally arrive at the goal. Path lengths for Robots 1 and 2 are $8.308m$ and $8.096m$ (the straight distances for Robots 1 and 2 are $8.006m$ and $8m$, respectively). Hence, the VSF^2 method provides a solution which allows the two robots to effectively coordinate their motion.

In this simulation, the two robots change their speeds and orientation gradually (Equation (4-3) to (4-6)). When the two robots move closer in the corridor, they will suffer repulsive force from each other. Both robots begin to decelerate and make a turn to avoid collision. Robot 1's speed reaches its minimum $0.478m/s$ at $5.22s$ (Figure 4-6 (a)). Robot 2's speed reaches its minimum $0.471m/s$ at $5.23s$ (Figure 4-6 (c)). When the robots pass each other, they accelerate to higher speeds. Robot 1 reaches its maximum speed $0.75m/s$ at $7.41s$ (Figure 4-6 (a)) and Robot 2 reaches its maximum speed $0.75m/s$ at $8.28s$ (Figure 4-6 (c)). It seems that Robot 2's moving direction (Figure 4-6 (d)) changes rapidly at about step *400* (at $4s$). Note that in Figure 4-6 the orientation of the robots is represented in radian between $(-\pi,\pi)$, so in fact the path of Robot 2 is quite smooth.

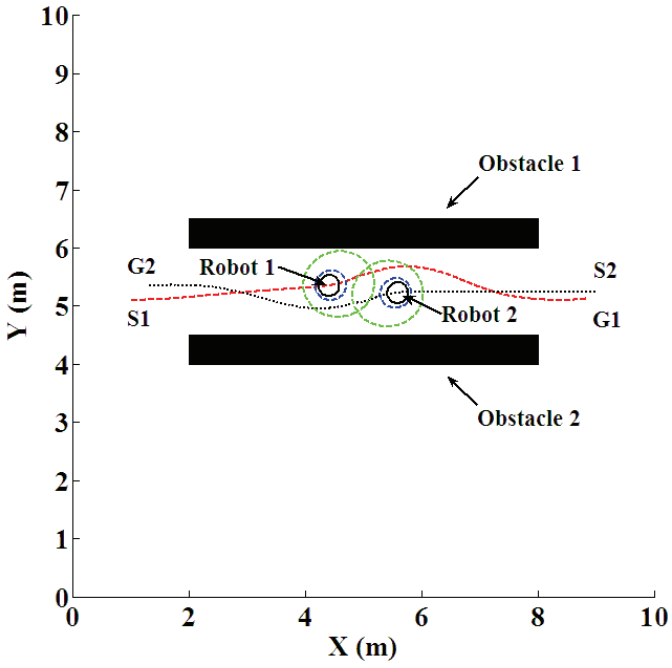


Figure 4-5 VSF^2 method: two-robot case

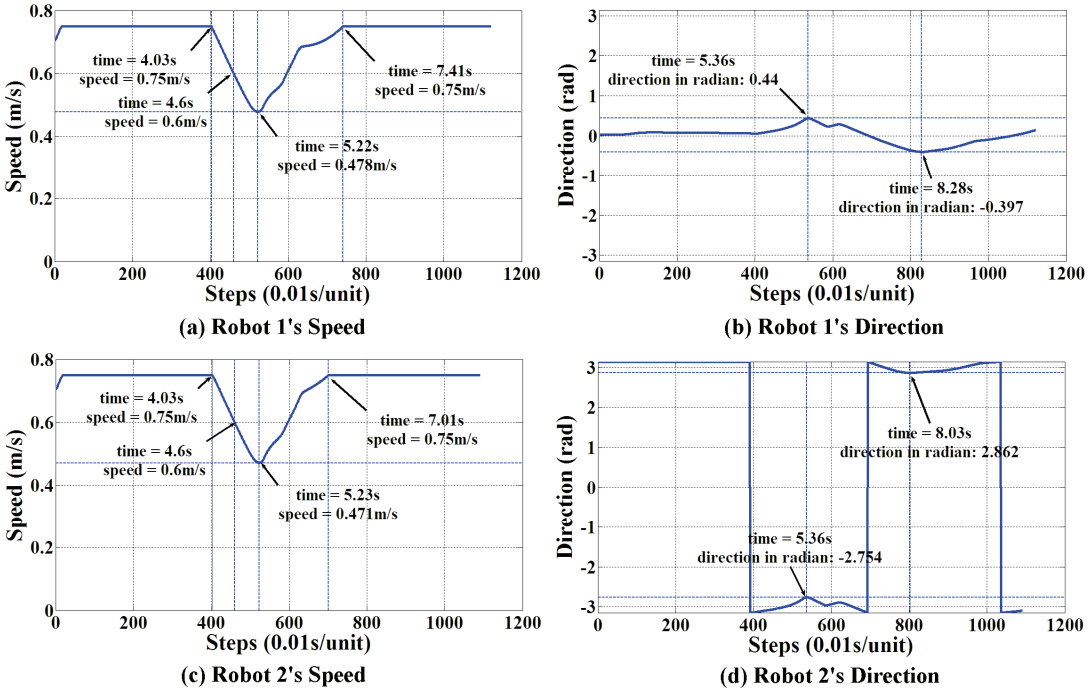


Figure 4-6 VSF^2 method: two-robot case (robots' speeds and moving directions)

B. Simulation 2: Motion Planning and Collaboration of a Team of Four Robots in a Complex Indoor Environment

This simulation is conducted in the same environment as the one in Section 3.5.2 but with more obstacles (see Figure 4-7). The dimension of this area is $20m$ by $20m$. The simulation includes four robots travelling in the indoor environment. Parameters used in this simulation include: $R_r = 0.18m$, $m = 3.6kg$, $\rho_0 = 0.2$, $k = 6$, $|\omega| \le 5\pi/3$, $v_{max} = 0.75m/s$, $C = 2$, $P = 20$, $Q = 5$, $T_p = 1$. Robot 1 is supposed to travel from $S_1(18,1)$ to $G_1(7,18)$. Robot 2 is supposed to travel from $S_2(2,18)$ to $G_2(18,4)$. Robot 3 travels from $S_3(1,4)$ to $G_3(17,12)$. Robot 4 travels from $S_4(18,18)$ to $G_4(7,2)$. In Table 4-1, D shows the straight-line distances from S_i to G_i. D_{real} is the resultant path length found by the VSF^2 method. The straight-line distances from S_i to G_i $(i = 1,2,3,4)$ are $20.248m$, $21.26m$, $17.889m$ and $19.416m$, respectively. The resultant path lengths with for Robots 1, 2, 3 and 4 using the VSF^2 method are $23.844m$, $25.27m$, $18.629m$ and $21.318m$, respectively.

In Figure 4-7, the paths of the four robots obtained by the VSF^2 method are shown in solid line, dotted line, dash dot line and dashed line, respectively. D_{max} of each robot is shown in dashed ellipse. The robots' entities are denoted by circles. The robots' moving directions and speeds during the process are showed in Figure 4-8, where the unit of x coordinate is $0.1s$.

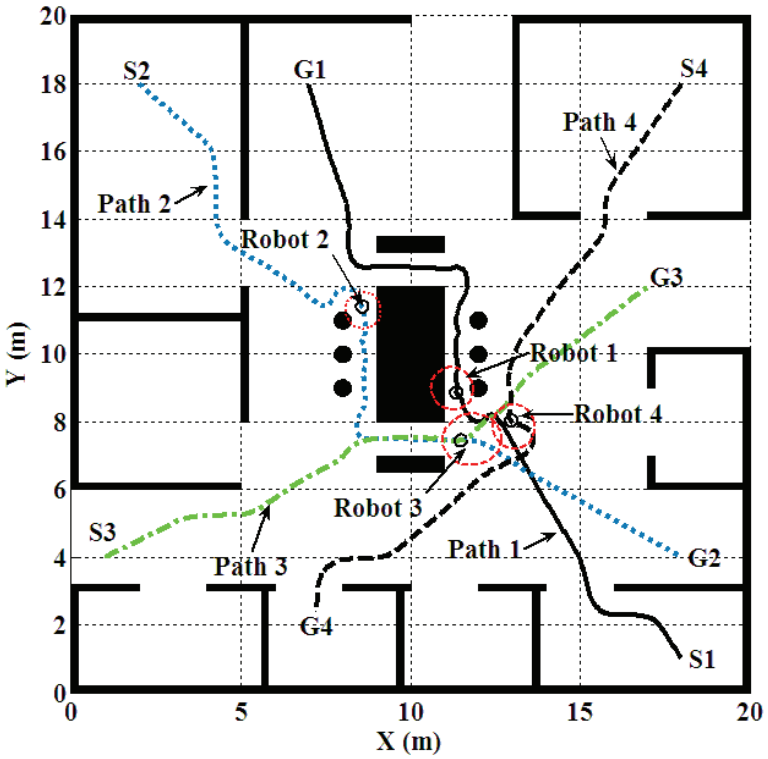


Figure 4-7 VSF^2 method: four-robot case

i	S_i	G_i	D (m)	D_{real} (m)
1	18,1	7,18	20.248	23.844
2	2, 18	18,4	21.26	25.27
3	1,4	17,12	17.889	18.629
4	18, 18	7,2	19.416	21.318

Table 4-1 Four robots simulation results

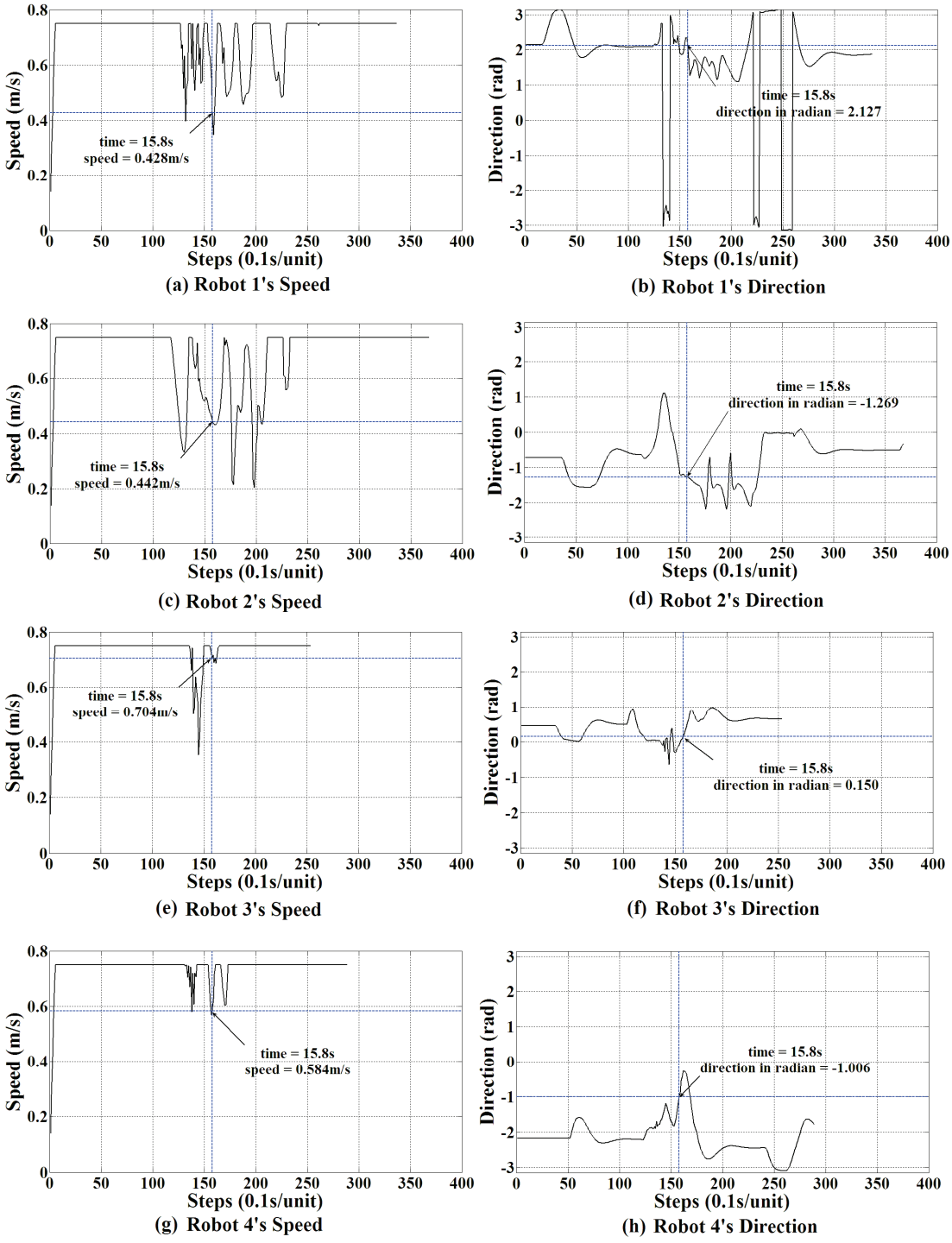


Figure 4-8 VSF^2 method: four-robot case (robots' speeds and orientations)

When a robot moves in free space, it travels with its maximum speed ($0.75m/s$), as shown in Figure 4-8 (a), (c), (e) and (g). When it enters the obstacle-cluttered area, it decelerates and the

coverage of its force field becomes smaller which allows it to enter narrow passages. Figure 4-7 shows a snapshot of the simulation in this environment at time $15.8s$, in which robots are travelling in the obstacle-cluttered area. The speed of Robots 1, 2, 3 and 4 are reduced to $0.428m/s$, $0.442m/s$, $0.704m/s$ and $0.584m/s$, respectively. Robots 1 and 2 are slowing down to enter the interspaces between rectangle obstacle and round small obstacles. Robots 3 and 4 are going to collide and Robot 4 is forced to slow down and steers clear of Robot 3's front. When they leave the obstacle clustered area, the robots accelerate to their maximum speeds (see Figure 4-8).

4.1.3 Conclusions on Variable Speed Force Field Method

In this section, the Variable Speed Force Field method (VSF^2) for multi-robot motion planning and collaboration has been described in detail. Taking a robot's dynamic and kinematic characteristics into consideration, this method provides a much better solution than the CF^2 method. By changing its speed, a robot can pass narrow passages which are not feasible in the CF^2 method. When a robot travels near obstacles or other robots, it slows down to find a possible path. When the robot leaves the obstacle-cluttered area, it will accelerate to its normal speed.

The VSF^2 method inherits the advantages of the CF^2 method. Robots using the VSF^2 method work in a decentralized architecture. A robot plans its path and motion independently according to the surrounding environment and its own status. The VSF^2 method does not suffer from replanning problems and is suitable for real-time motion planning and coordination in complex environments.

In the simulations presented in this section, it is assumed that there are no moving obstacles in the environment. VSF^2 method can be extended for use in environments with moving obstacles. In real applications, robots are usually equipped with sensors to detect their surrounding environments, which provide range and direction information data of obstacles nearby. In each control cycle, moving obstacles can be treated as static obstacles. Section 6.1 details

experimental studies on using a laser sensor to get the information required by the F^2 method for real time motion planning and collision avoidance.

In the VSF^2 method and other F^2 based methods, only obstacles and force fields of other robots in a robot's force field are considered in collision avoidance. Therefore a robot using the VSF^2 method does not need to search for a path as many conventional approaches require. The VSF^2 method is very efficient and is suitable for real time application. Since it is an inherently local approach, the VSF^2 method suffers from the local minima problem. A possible solution is to combine the F^2 method with a global planner to avoid being trapped into local minima, the attractive force is then generated by a subgoal given by a global planner. Such an approach will be detailed in Section 4.3.

4.2 Subgoal-Guided Force Field Method

4.2.1 Introduction

Figure 4-9 shows a problematic case of the CF^2 and VSF^2 methods. Let an obstacle be located between a robot and its goal position. The three patches in the middle show several possible robot locations on its way towards the goal. According to the CF^2 and VSF^2 methods, the robot is attracted by the attractive force from the goal and repelled by repulsive forces from the obstacles. Since Obstacle 1 is in the way of this robot, the robot goes towards Obstacle 1 and is repelled from it when the repulsive force is large enough (from position *A* to position *B*). When this robot is at position *B*, there is no repulsive force from Obstacle 1 and this robot turns back to its goal direction (position *C*) because of the effect of the attractive force. Then it will be pushed away by the obstacle again (position *C*). This 'go towards obstacles' behaviour causes the zigzag movements and oscillations on the speed and direction of the robot.

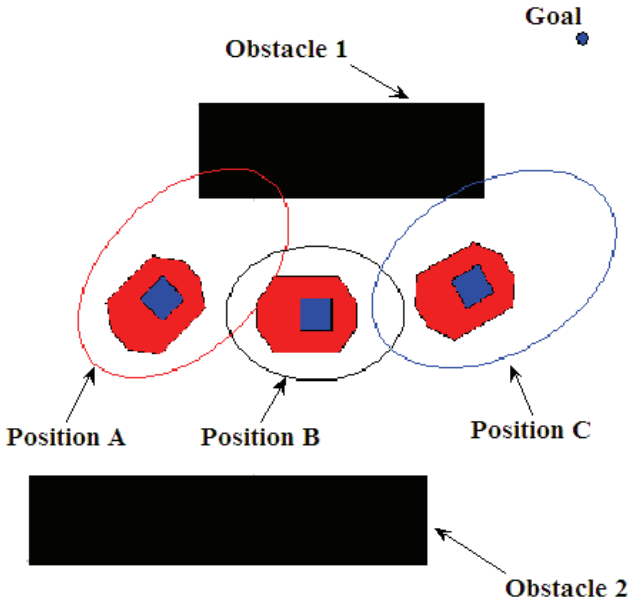


Figure 4-9 SGF^2 method: a problematic case

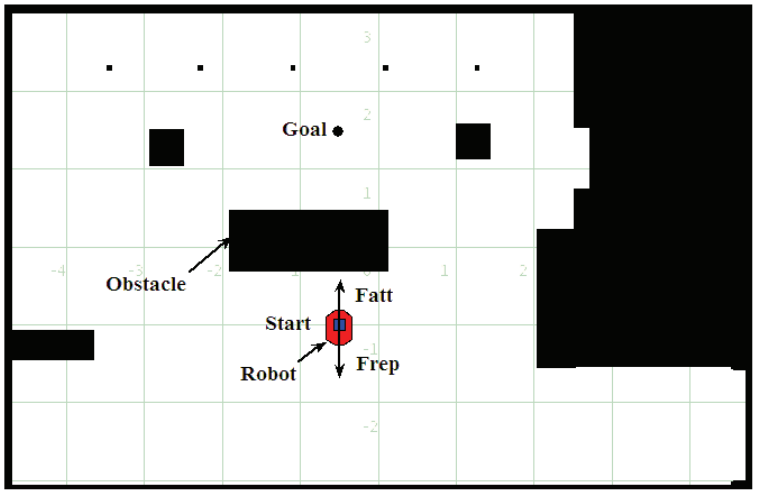


Figure 4-10 SGF^2 method: a local minimum for F^2 method and PFM

Figure 4-10 shows a situation where the F^2 method suffers a local minimum problem like the Potential Field Method. An obstacle is located between its start point and goal point. The attractive force from the goal ($\mathbf{F}_{att}$) and repulsive force from the obstacle ($\mathbf{F}_{rep}$) are opposite and collinear, which causes the robot to stop at the point when the two forces are equal (in equilibrium).

The concept of *subgoal* is used in robot motion planning [137-140]. In [139], the subgoal positions are continuously updated based on sensor data while robots are moving. In a manipulator path planning problem, a global search tries to find a sequence of subgoals and a serial local search conducts a local search between subgoals. This method traces back to the global search when the local search fails to find local paths [58]. In [140], long-range sensor data and global information are used to generate an intermediate goal and the short-range sensors are used to guide a robot to the subgoal.

4.2.2 Subgoal-Guided Force Field Method

Based on the analyses above, a Subgoal-Guided Force Field method (SGF^2) has been developed, in which the concept of *subgoal* is integrated with the VSF^2 method. In the scenarios considered in this section, a robot is equipped with a laser sensor. This robot is capable of identifying openings in front of it based on the returned range data. The midpoints of these openings are selected as subgoal candidates. A heuristic function is then utilized to evaluate these candidates and the subgoal candidate with lowest cost will be selected as the current subgoal. This section focuses on the theoretical study of subgoal-guided motion planning; the approach to identify openings in a real robot's surroundings will be described in detail in Section 6.1.6.

Figure 4-11 and Figure 4-12 illustrate how to determine subgoal candidates in the case of Figure 4-10. Firstly, two openings are found by the laser sensor. Opening 1 is a gap between the front obstacle and the wall on the right and Opening 2 is a gap between the obstacle and the wall on the left. The midpoints of Opening 1 and 2 are denoted by P_1 and P_2, respectively. P_1 and P_2 are selected as subgoal candidates.

The cost function applied by the robot to evaluate subgoals is defined as a sum of two functions. The first function is the distance from the robot position to a subgoal candidate (denoted by S_1). The second function is the estimated distance from a subgoal candidate to the global goal (denoted by S_2). In the scope of this research work, a robot is assumed to have

knowledge of its own position in the global map and the position of the subgoal candidate can be estimated from the sensor reading, so S_1 and S_2 can be calculated. The cost function is defined by:

$$g = k_1 s_1 + k_2 s_2 \tag{4-11}$$

The subgoal candidate with lowest cost g will be selected as the current subgoal. If there is more than one minima existing, the subgoal will be selected randomly from candidates with minimum cost. By tuning weighing factors k_1 and k_2, different subgoals may be selected. For example, a robot may choose to go to an opening which is close to its final destination (with smaller S_2) or tend to go to an opening which is near to its current location (with smaller S_1). From Figure 4-11, it seems that the robot will collide with obstacle edges if it goes directly to subgoals. But this will not occur since the robot is always protected by its repulsive force field while moving.

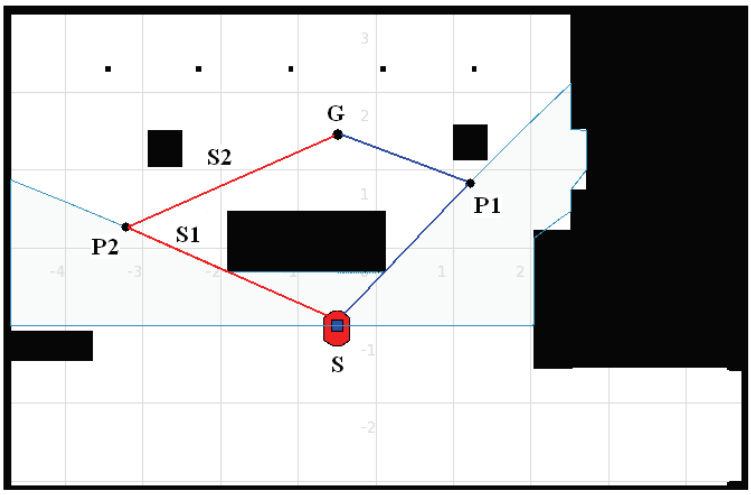


Figure 4-11 SGF^2 method: illustration of subgoals

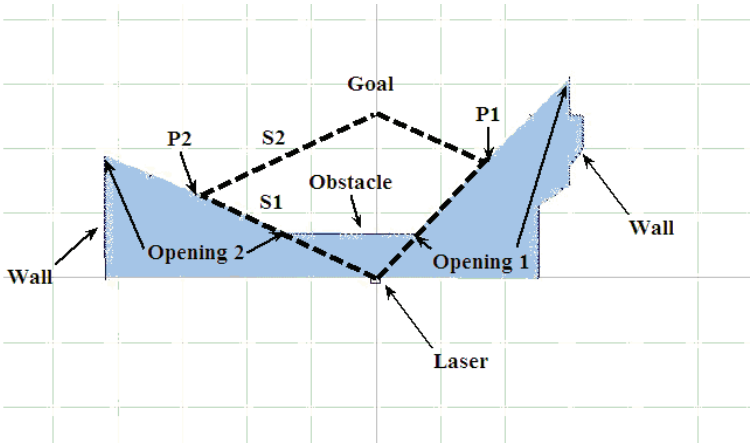


Figure 4-12 SGF^2 method: laser view

When a subgoal is determined and the final destination is not in the sensor's range, an attractive force is generated from this subgoal, instead of from the final destination, and attracts this robot to the subgoal. The subgoal is continuously changed and updated based on sensor data when a robot is moving. When the final destination is found within the sensor's range, the robot will move to it directly.

The SGF^2 method follows the definition of the force field and repulsive force of the F^2 method, but changes the way of defining attractive force. It is a generic approach for all F^2 based methods. For different applications, a robot's motion can be determined by either CF^2 with Equation (3-21) to (3-25) or by VSF^2 with Equation (4-1) to (4-10).

4.2.3 Simulation Studies on Subgoal-Guided Force Field Method

This section presents the simulations carried out using the Player/Stage platform [141], which will be introduced with more details in Section 6.1.1. A robot is supposed to travel from its start point to a goal point in indoor environments. This robot is equipped with a laser sensor and an Adaptive Monte-Carlo Localization driver (*amcl*) [142] is utilized to determine its location. The update threshold of the *amcl* driver is set to be 10 degrees or $5cm$, that is, the *amcl* driver returns with this robot's new pose when there is a displacement of $5cm$ or

angular displacement compared to the *amcl* driver's last output. Robot parameters are selected based on those of a pioneer robot [143]. In all simulations, k_1 and k_2 are set to *1*.

A. Case 1: local minimum problem

In this simulation, the local minimum problem shown in Figure 4-10 is studied. A robot is supposed to travel from $S(-0.5,-1)$ to $G(-0.5,1.5)$. Obstacle 1 is located between the robot and its destination. In the case shown in Figure 4-11, two subgoal candidates are P_1 and P_2. For P_1, the distance from the robot to P_1 is $S_1 = 2.567m$ and the distance from P_1 to global goal is $S_2 = 1.995m$. From Equation (4-11), the cost $g_1 = k_1 S_1 + k_2 S_2 = 1 \times 2.567 + 1 \times 1.995 = 4.562$. For P_2, the distance from the robot to P_2 is $S_1 = 2.875m$, the distance from P_2 to global goal $G(-0.5,1.5)$ is $S_2 = 2.844m$. The cost $g_2 = k_1 S_1 + k_2 S_2 = 1 \times 2.875 + 1 \times 2.844 = 5.719$. Since $g_1 < g_2$, P_1 is then selected as the current subgoal. The motion control is then calculated by Equation (4-1) to (4-10) as in the VSF^2 method.

Simulation snapshots are given in Figure 4-13. In Figure 4-13 (a), P_1 and P_2 are two subgoal candidates. P_1 is selected as the current subgoal according to Equation (4-11). Then an attractive force is generated from P_1, which makes the robot turn right, as shown in (b). In (b), there are three subgoal candidates, P_1, P_2 and P_3. P_2 is selected as the new subgoal, which also makes the robot turn right. The subgoal is continuously updated in the robot's movement, e.g., P_2 in (c), P_3 in (d), P_5 in (e), P_7 in (f) and P_6 in (g). Under the influence of attractive force from these subgoals, the robot steers clear of Obstacle 1. In (h), the robot's global goal G is in the laser's range. An attractive force is then generated from G and no subgoal is needed. The resultant path is shown in Figure 4-14. The robot arrives at its destination successfully and the path is quite smooth.

B. Case 2: motion planning in a changing environment

The map used in this simulation is shown in Figure 4-15. A robot is supposed to travel from the start point at $S(-3,-2)$ to the destination $G(1,1.5)$. Figure 4-15 gives a snapshot when the robot is about to move. Six subgoal candidates are found (denoted by P_1, P_2, P_3, P_4, P_5, P_6) here. P_2 is selected as current subgoal. The resultant path is shown in Figure 4-16.

If the environment changes, for example, the lower corridor is blocked as shown in Figure 4-17, the subgoal candidates found are depicted as P_1 to P_4. From Equation (4-11), P_2 (P_4 in Figure 4-15) is selected as the current subgoal. The resultant path is shown in Figure 4-18. These simulations show that the SGF^2 method is applicable in dynamically changing environments.

The proposed Subgoal-Guided Force Field (SGF^2) method has been tested in simulations. Case 1 shows that the SGF^2 method performs better than the CF^2 and VSF^2 methods in solving local minimum problem. Since the subgoal is continuously updated based on sensor data, the Subgoal-Guided F^2 method is applicable in a changing environment (as in Case 2).

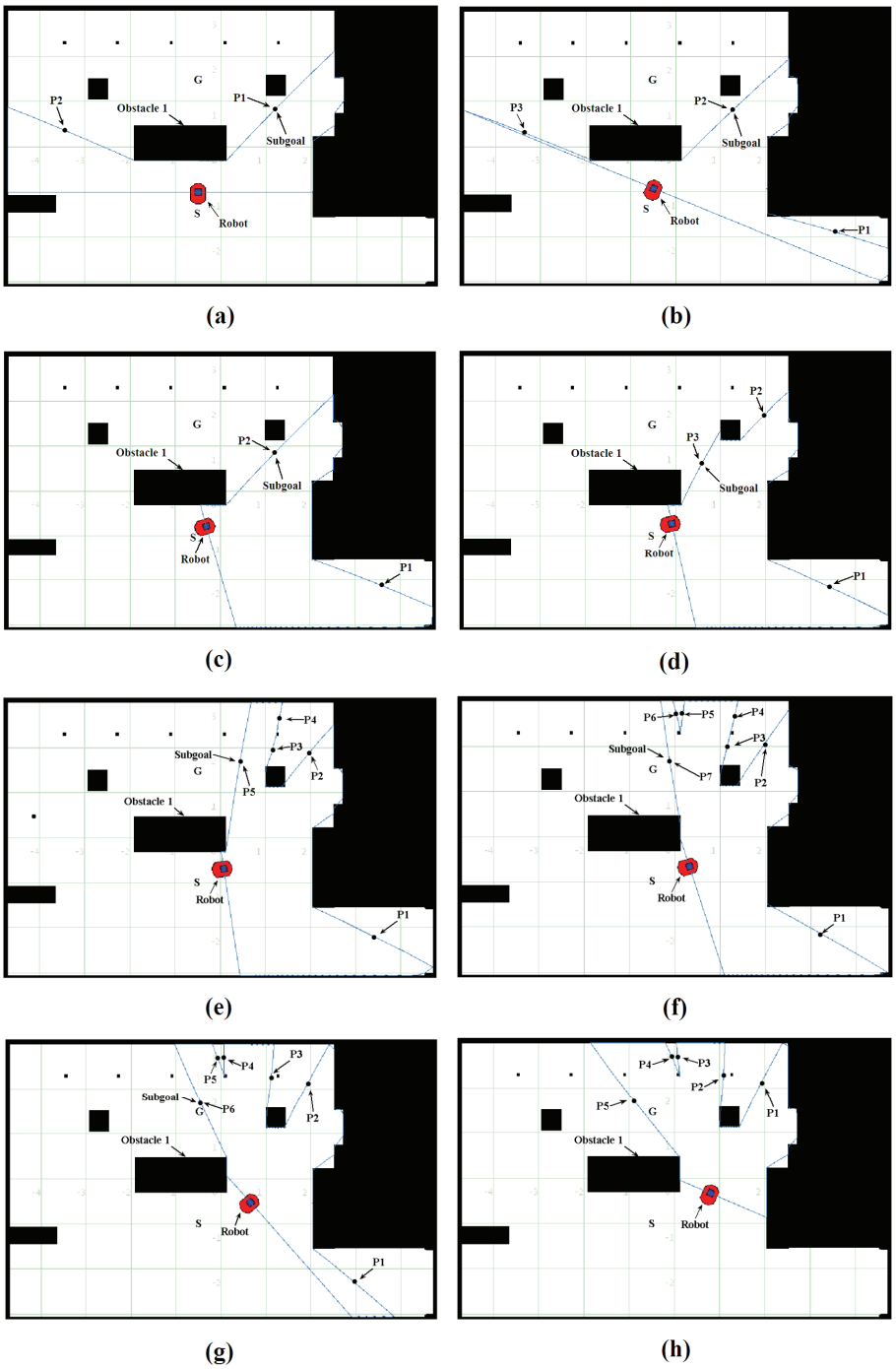


Figure 4-13 SGF^2 method: Case 1 - simulation snapshots

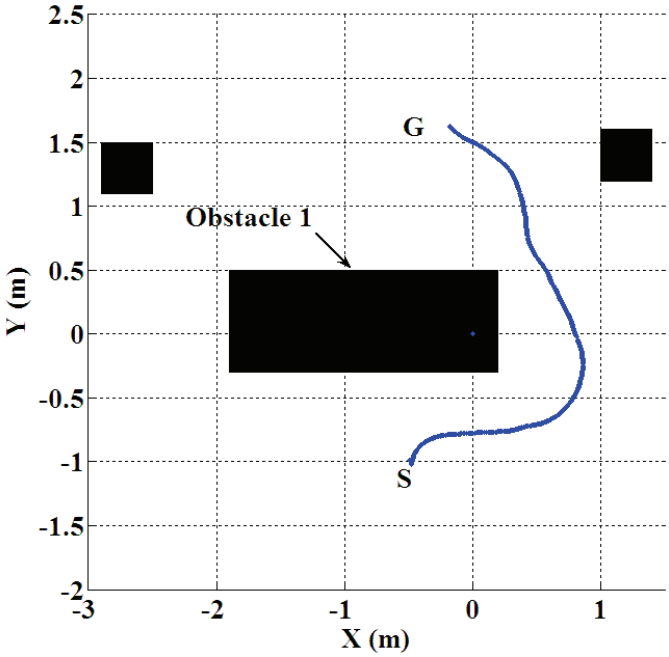


Figure 4-14 SGF^2 method: Case 1 - resultant path

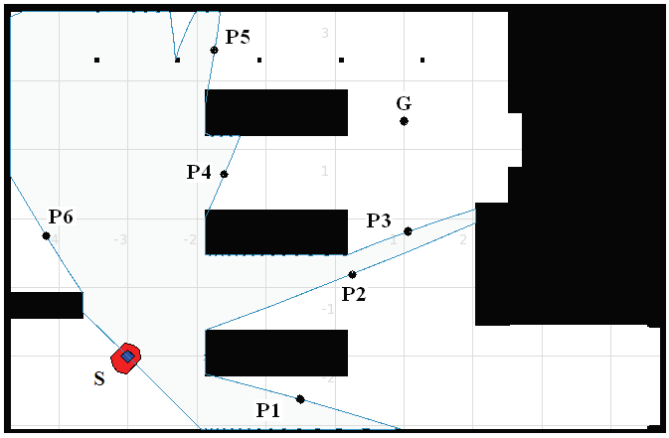


Figure 4-15 SGF^2 method: Case 2 - map

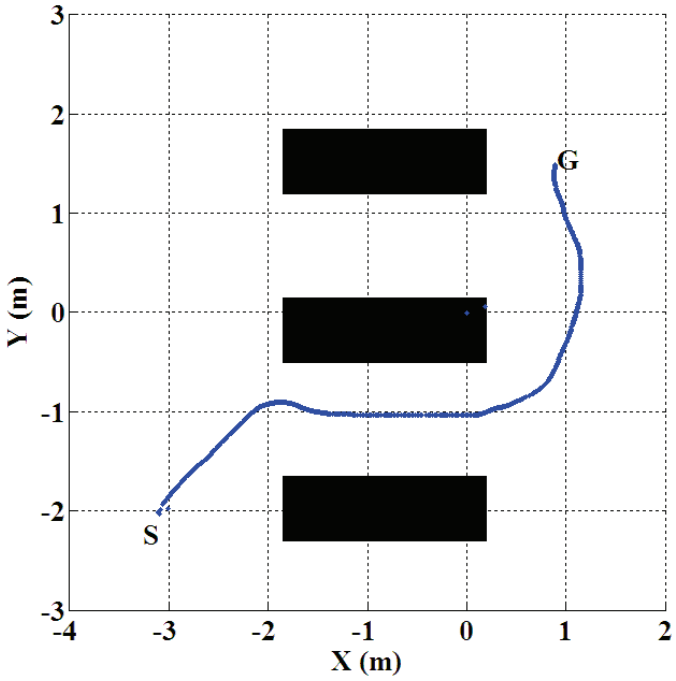


Figure 4-16 SGF^2 method: Case 2 - resultant path

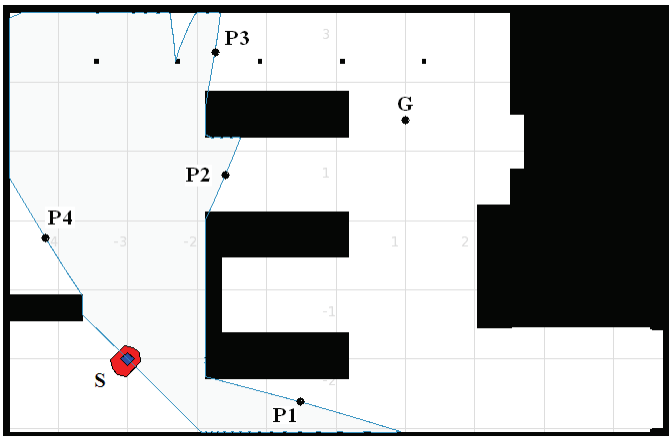


Figure 4-17 SGF^2 method: Case 2 - environment changed

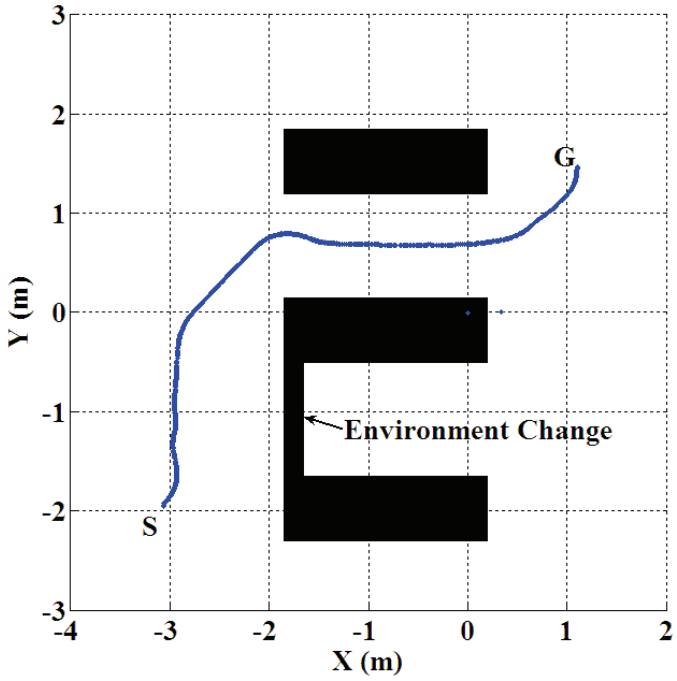


Figure 4-18 SGF^2 method: Case 2 - new path

4.2.4 Conclusion on the Subgoal-Guided Force Field Method

This section has presented a Subgoal-Guided Force Field method (SGF^2), which greatly improves the performance of the F^2 method. In the SGF^2 method, a robot first analyses the surrounding environment and finds possible passages to approach. The midpoints of these found passages are selected as subgoal candidates. These candidates are then evaluated by a heuristic cost function to determine which subgoal to move towards. The cost function is the sum of two parts: the distance from the current robot position to a subgoal candidate and the estimated distance from a subgoal candidate to the global goal. Once a subgoal is selected, the robot moves to that subgoal. The subgoal is continuously updated based on sensor information until the global goal is visible by the robot. The proposed SGF^2 method has been demonstrated feasible and efficient by simulations carried out in the Player/Stage platform.

By combining the concept of subgoal with the F^2 method, the SGF^2 is capable of reducing the 'go towards obstacles' behaviours and of being immune to local minimum problems of other F^2 based methods. A robot using the SGF^2 method needs to have the knowledge of its location

in the global map and the position of its goal, but does not need a complete global map of the whole environment. Thus, SGF^2 is suitable for real-time motion planning and collision avoidance in partially known and dynamically changing environments.

4.3 Dynamic Variable Speed Force Field Method

4.3.1 Local Obstacle Avoidance

A local obstacle avoidance approach is usually employed to change a robot's trajectory based on the updated sensor data during its movement. One of the simplest obstacle avoidance approaches is the 'bug' algorithm, in which a robot follows the contour of each obstacle in its path and circumnavigates them [144]. No kinematic or dynamic constraints are considered in this approach. The Potential Field Method has been a popular approach for real-time obstacle avoidance because of its mathematical simplicity [42]. A robot is often treated as a point under the influence of an artificial potential field and no kinematic or dynamic constraints are taken into account.

A method which is particularly relevant to the work presented here is the Elastic Band method proposed in [56]. This technique tries to combine the global path planning with real-time sensor-based collision avoidance. A pre-planned global path is deformed in real-time to keep a robot away from obstacles during its movement, while the internal contraction forces bring the robot back to its original path when the obstacle is out of the sensor range. This method also takes into account the actual shape of the robot and restricts the search space by the concept of a 'bubble'. A bubble is defined as the maximum local subset of the free space around a given configuration of the robot which can be safely traversed in any direction without collisions. Given such bubbles, a band or string of bubbles can be used along the trajectory from the robot's initial position to its goal position to show the robots' expected free space along the pre-planned path.

The obstacle avoidance techniques mentioned above and other variations have been proved to be effective in the presence of static obstacles in the environment. However, they are not directly applicable to dealing with dynamic obstacles, since the characteristics from the moving objects such as travelling speed, size, etc., are not built into the algorithms. The Dynamic Variable Speed Force Field ($DVSF^2$) method proposed here is designed around these premises and performs well in partially known and continuously changing environments. The $DVSF^2$ method adopts the definitions of force field and repulsive force in the CF^2 and VSF^2 methods, but the attractive force is generated by a temporary waypoint selected from a pre-planned path. Thus a robot is capable of tracking the pre-planned path while avoiding collisions.

4.3.2 Dynamic Variable Speed Force Field Method

$DVSF^2$ follows the definition of force field and repulsive force in the F^2 method, but changes the definition of attractive force and the way to determine a robot's motion.

A. Subgoal Selection and Attractive Force

Similar to the SGF^2 method, the $DVSF^2$ method generates a temporary goal for a robot based on the latest sensor data. This temporary goal then generates an attractive force which will attract the robot to it, but the difference is that the intermediate goal in $DVSF^2$ is selected from the points on a pre-planned optimal path. In this way, the robot tries to follow the pre-planned path while avoiding potential collision with obstacles. The optimality of the resultant path is well-preserved. The pre-planned path can be generated by any global planner. In this research, the path is planned by an optimized global planner proposed by Taha et al. in [4].

At any given time a temporary goal is selected as the furthest visible point provided by the sensor along the pre-planned global path, which is given by a high-level global planner. A virtual attractive force, denoted by $\mathbf{F}_{att\_tg}$ is associated with this temporary goal to attract the

robot from its current location towards the goal. This force directs from the centre of this robot to the temporary goal and its magnitude is set to be a positive constant Q.

$$\left|\mathbf{F}_{\mathbf{att\_tg}}\right| = Q \tag{4-12}$$

B. The Resultant Force

The resultant force acting on the robot is given by the sum of the attractive force and repulsive forces from obstacles and other robots.

$$\begin{aligned}\mathbf{F}_{\mathbf{total}} &= \mathbf{F}_{\mathbf{att}} + \mathbf{F}_{\mathbf{rep}} \\ &= \mathbf{F}_{\mathbf{att\_tg}} + \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{F}_{\mathbf{rep\_o\_i}} + \sum_{\mathbf{l=1}}^{\mathbf{m}} \mathbf{F}_{\mathbf{rep\_j\_l}}\end{aligned} \tag{4-13}$$

where n is the number of obstacles (including moving obstacles) which are detected by a robot's sensor and m is the number of detected robots. $\mathbf{F}_{\mathbf{rep\_o\_i}}$ is the repulsive force from obstacle i. $\mathbf{F}_{\mathbf{rep\_j\_l}}$ is the repulsive force from another robot l.

C. Motion Equation

A robot's motion in a 2D environment can be described by the translational speed v and the rotational movements about its centre of mass $\boldsymbol{\omega}$, which are given by:

$$v = \frac{\lambda}{1 + \dfrac{max\left(\left|\mathbf{F}_{\mathbf{rep\_o\_i}}\right|, \left|\mathbf{F}_{\mathbf{rep\_j\_l}}\right|\right)}{P}} v_{max} \tag{4-14}$$

$$\boldsymbol{\omega} = \eta\left(\angle \mathbf{F}_{\mathbf{total}} - \theta_r\right) \tag{4-15}$$

where P is the positive constant which determines the magnitude of the force field. λ and η are positive constants. Equation (4-14) guarantees that a robot's speed will be sensitively influenced by the magnitude of repulsive forces acting on it. The maximum of current repulsive forces is utilized in Equation (4-14) to ensure that the robot will decelerate adequately to avoid potential collisions. Equation (4-15) defines the angular speed as a function of the angular difference between the desired direction along the resultant force and the robot's current orientation θ_r.

4.3.3 Simulations Studies on Dynamic Variable Speed Force Field Method

The proposed DVSF^2 method is evaluated by simulations on an automated wheelchair platform in an office-like environment with narrow passages, doors, long corridors and cluttered static and dynamic obstacles. The large mechanical platform measures $1.2m$ by $0.7m$. The wheelchair has two differentially driven wheels at the rear and two passive casters at the front and can travel at speed of up to 15km/h. A laser sensor is installed on the wheelchair to obtain environment information, as shown in Figure 4-19.

Figure 4-19 An automated wheelchair [4]

In the simulation scenario, the wheelchair robot starts when an optimal path is found by the global planner, denoted by the brown line in Figure 4-20. The DVSF^2 method then tends to direct the robot away from obstacles while tracking the pre-planned path, making the path not only feasible and kinematically 'gentle' for the platform, but also maximising clearances with the obstacles known at this stage along the minimal distance path. The real path is shown by the blue line in Figure 4-21 and Figure 4-22.

Figure 4-21 depicts the scenario when a static obstacle (square box) is placed in the vicinity of the robot while the robot is navigating along the given path. It can be seen that the DVSF^2 method is capable of avoiding the obstacle, generating the tracking trajectory shown in blue. The presence of a moving obstacle in the narrow corridor (as shown in Figure 4-22) also makes the robot take evasive action. The robot slows down because of the short distance between the robot and the dynamic obstacle and then speeds up once the obstacle is passed. This is also the case when the robot crosses narrow spaces such as doors, as reaction forces from obstacles are exerted on the robot in the opposite direction. In the simulations presented here, the moving obstacle (denoted by the small box) is assumed to follow a predetermined path with a constant speed to simulate the presence of a person moving in the vicinity. In Figure 4-23, the pre-planned global path is completely blocked by the presence of the static obstacle. The environment change is detected by sensors on the wheelchair and the global planner is called to replan a new path from the robot's current location to the goal.

In this simulation, a large robot is made to travel in a challenging environment with narrow passages and dynamic obstacles. Since the size of this robot is large, the space for the robot to avoid collision is very limited. The robot must react to the environment quickly for safety reasons, so rapid changes in the robot's speed can be observed when it traverses narrow passages or is close to obstacles.

4.3.4 Conclusions on the Dynamic Variable Speed Force Field Method

This section has presented the DVSF^2 method which is a novel F^2 based approach for real-time collision avoidance. In the DVSF^2 method, an attractive force is generated by a temporary

waypoint which is selected as the furthest visible point on a pre-planned optimal path. Thus a robot is capable of tracking the optimal path while avoiding collisions with obstacles. The motion planner of the DVSF^2 method is designed to be simple and easy to implement, which makes it especially suitable for collision avoidance in dynamically changing environments. Simulation with a large wheelchair platform was carried out in an indoor environment to prove the feasibility of this approach.

This section provides a way to combine F^2 method with a global planner. With the help of a global planner, a robot using the DVSF^2 method is capable of avoiding collisions with moving obstacles and dealing with environment changes. Compared to other existing approaches, the way to generate a subgoal presented in this section can be easily implemented in real applications.

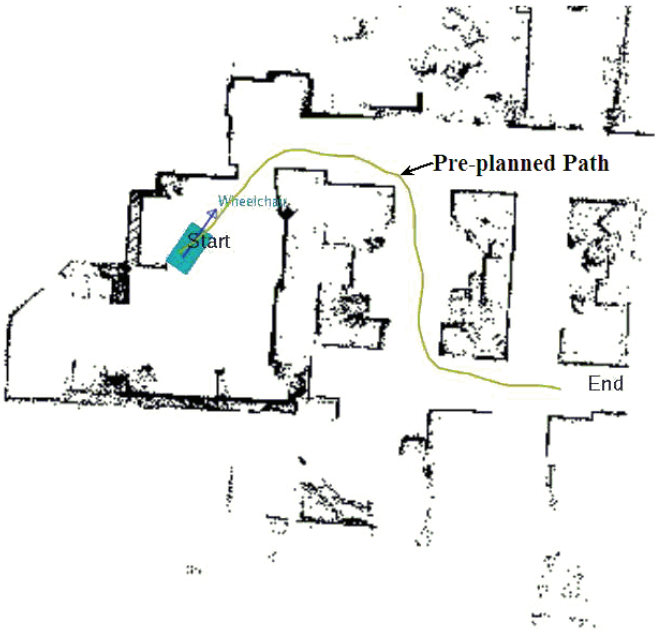


Figure 4-20 DVSF^2 simulation: snapshot 1

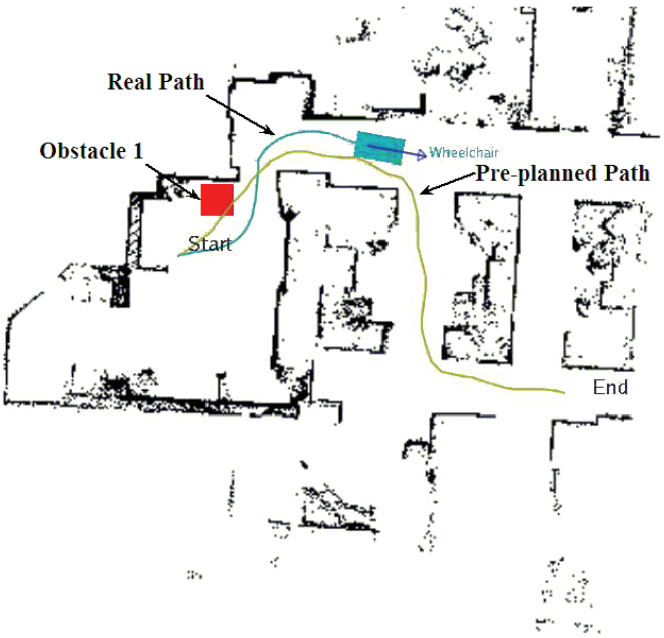


Figure 4-21 DVSF^2 simulation: snapshot 2

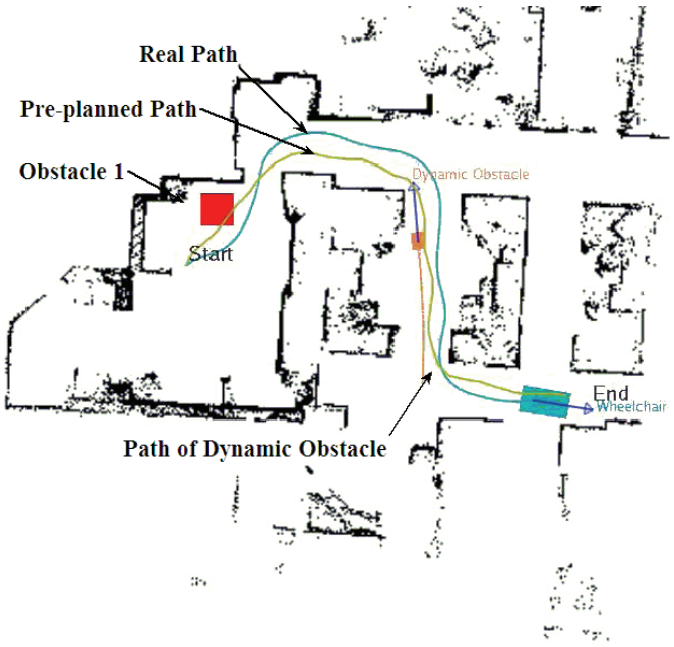


Figure 4-22 DVSF^2 simulation: snapshot 3

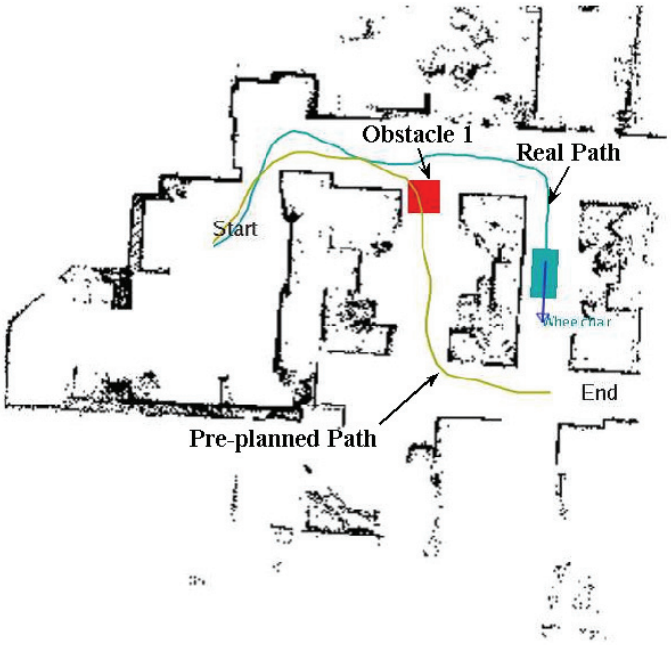


Figure 4-23 DVSF^2 simulation: snapshot 4

4.4 Discussions on Force Field Methods

Four F^2 methods have been studied in this thesis, including the CF^2, VSF^2, SGF^2 and DVSF^2 methods. These methods adopt the same definition of a robot's force field. The CF^2 is the original version of the F^2 method and can be considered as a special case of other three methods. In CF^2, a robot moves at a constant speed and its moving direction is determined by the direction of resultant forces acting on it.

The VSF^2 method improves the F^2 method by allowing a robot to alter its speed based on environment information. The motion control equations (4-1) to (4-10) are introduced to ensure that a robot fulfills its dynamic and kinematic constraints.

The SGF^2 method improves the way of generating attractive force. The concept of subgoal in SGF^2 is also applicable to the CF^2 method. CF^2 is actually a special case of SGF^2 in which the subgoal is the final destination and the robot's speed is fixed.

The DVSF^2 method improves CF^2 by varying a robot's speed according to the maximum repulsive force acting on it. If this maximum repulsive force is much smaller than the repulsive force constant *P*, the robot can be considered as traveling with constant speed. The equation to determine a robot's angular velocity in CF^2, equation (3-24), is a special case of the motion equation (4-15) , when the factor $\eta = 1/\Delta t$.

4.5 Conclusions

In this chapter, three F^2 originated algorithms are presented, including the Variable Speed Force Field method (VSF^2), Subgoal-Guided Force Field method (SGF^2) and Dynamic Variable Speed Force Field method (DVSF^2). In the VSF^2 method, a robot's dynamic and kinematic characteristics are taken into account in motion planning and collision avoidance. A robot's speed is adaptively changing based on environment information and its own status. The VSF^2 method greatly broadens the applications of the F^2 method. The SGF^2 method introduces the subgoal into the F^2 method. In this method, a robot analyses its surrounding environment using sensor information and then selects a subgoal to follow. The SGF^2 method is suitable for real-time motion planning and collision avoidance in partially known and dynamically changing environments. The DVSF^2 is suitable for integration with a global planner. In this method, a temporary waypoint is selected on a pre-planned path. A robot using the DVSF^2 method is able to follow the pre-planned path while avoiding collisions. The DVSF^2 method is suitable for real-time collision avoidance in dynamically changing environments. The feasibility and performance of these approaches is demonstrated by various simulations carried out in various environments.

Chapter 5
Optimization based Parameter Refinements

The Force Field (F^2) based methods presented in Chapter 3 and Chapter 4 use several parameters which have a significant effect on the performance of these methods. This raises the requirement of finding appropriate/optimal values of the parameters for the F^2 methods. This chapter starts by analysing the effect of parameters in the F^2 methods on motion planning and collision avoidance in Section 5.1. An emerging algorithm in the evolutionary computation family, the Particle Swarm Optimization (PSO), is then briefly reviewed in Section 5.2. The PSO algorithm is then successfully applied to search appropriate parameter values for optimising the performance of the F^2 methods in single-robot and multi-robot motion planning and coordination in Section 5.3. A Particle Swarm Optimization Tuned Force Field method (PSO-tuned F^2) is presented for solving the single objective optimization problem of robot motion planning using the F^2 method. Then a multiobjective optimization approach, a Ranked Pareto Particle Swarm Optimization (RPPSO), is proposed in Section 5.4 and utilized to solve the multiobjective motion planning and collision avoidance problems using the F^2 method in Section 5.5. Conclusions are given in Section 5.7. The Variable Speed Force Field method (VSF^2) is used in the following robot motion planning simulations, but techniques presented in this chapter are suitable for all F^2 based methods.

5.1 Introduction

The parameters in the F^2 methods include: radius of a robot R_r, robot's location in the global reference frame (X_r, Y_r, θ_r), the absolute value of robot speed v_r, the maximum speed v_{max}, robot's priority T_p, environment factor (C), scalar constant k which affects a force field's coverage area, scalar constant ρ_0 defining the coverage area of D_{min}, scalar constant P defining the magnitude of repulsive force and scalar constant Q defining the magnitude of attractive force. Among these parameters, R_r and v_{max} are fixed when a robot is selected. X_r, Y_r, θ_r and v_r are measured real-time. T_p is a robot's priority compared with other

robots and is a fixed value once assigned by a planner. So the parameters k, C, P, Q and ρ_0 need to be designed when applying the F^2 methods.

A simple example is studied here to highlight the importance of parameter settings in the F^2 methods. Figure 5-1 shows an area of $10m$ by $10m$. A robot is supposed to travel from $S(2,3)$ to $G(8,7)$. The centre of a circular obstacle with radius $R=1m$ is located at $(5,4.5)$. Simulations are carried out using several groups of parameters in the VSF^2 method. In Figure 5-1, Paths 1 to 5 are simulation results using randomly selected parameters. Path 6 (shown in solid line) is the result of optimized parameters using the approach to be presented in Section 5.3. The corresponding parameters are given in Table 5-1. It can be seen from Figure 5-1 that given different parameters, VSF^2 will return with various results for the same task. For example, the 2nd set of parameters leads to better results than using other sets of parameters in terms of travel distance; by contrast, using the 4th set of parameters fails to find a solution since collision occurs when the robot travels close to the obstacle.

This simulation shows that parameter selection is important to the performance of the F^2 methods. Thus suitable optimization approaches are required to find appropriate parameters for F^2 methods. To this end, an efficient and simple algorithm, such as the Particle Swarm Optimization (PSO) algorithm, is very attractive.

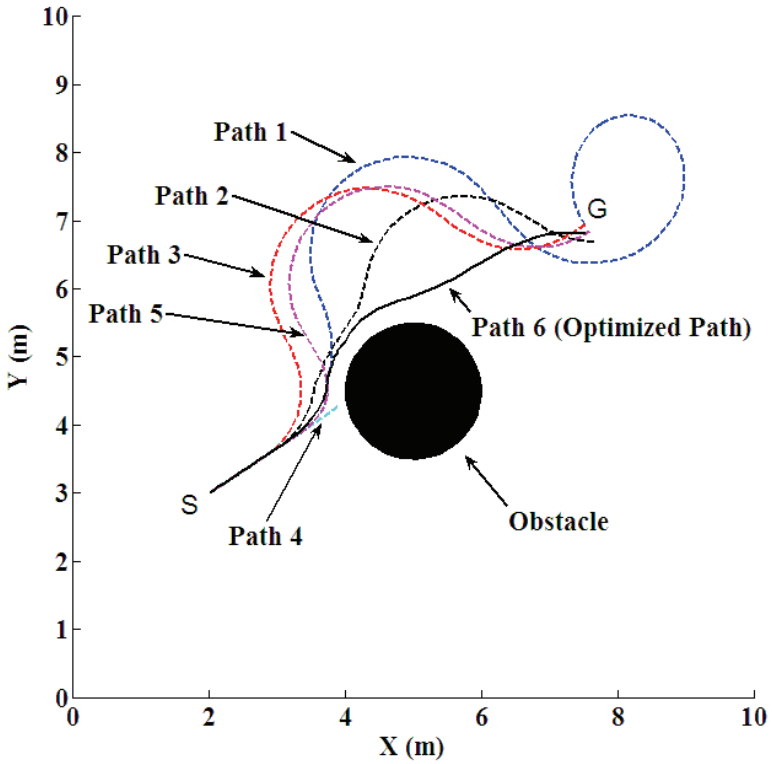


Figure 5-1 Single objective optimization Case 1: paths resulting from different parameters

Set of Parameters	Path Length (m)	k	P	Q	C	ρ_0	Success
1	15.447	6.741	17.716	5.050	2.521	0.674	Yes
2	8.196	8.198	15.318	10.209	2.088	0.551	Yes
3	9.500	9.415	17.707	6.952	1.825	0.436	Yes
4	N/A	4.578	13.053	5.092	2.627	0.216	No
5	9.586	8.707	7.571	7.318	2.517	0.522	Yes
6	7.106	4.238	9.566	9.268	1.623	0.298	Yes

Table 5-1 Parameters in single objective optimization Case 1

5.2 Particle Swarm Optimization (PSO)

PSO was proposed by Kennedy and Eberhart in 1995, motivated by observation of the social behaviour of birds flocking or fish schooling. [145]. At the beginning of PSO, a number of particles are placed in the search space of a problem or function. The objective function is evaluated by each particle at its current position. Each particle then determines its movement through the search space by considering the history of its own current and best (best-fitness) locations with those of one or more members of the swarm. The next iteration takes place after all particles have been moved. Eventually the swarm will approach an optimum of the fitness function.

Suppose the dimension of the search space is Z and the number of particles is N. The position of the i^{th} particle in the search space is represented as $\mathbf{X_i} = (x_{i1}, x_{i2}, x_{i3}..., x_{iZ})$. Each particle maintains a memory of its previous best position denoted by $\mathbf{P_{best\_i}} = (p_{i1}, p_{i2}, p_{i3}..., p_{iZ})$. The best value of all $\mathbf{P_{best}}$ is the global best $\mathbf{G_{best}}$. Vector $\mathbf{V_i} = (v_{i1}, v_{i2}, v_{i3}..., v_{iZ})$ is the velocity of the i^{th} particle. Each particle updates its position in the search space according to the following equations.

$$v_{iz} = v_{iz}\varepsilon + c_1 r_1 (P_{best} - x_{iz}) + c_2 r_2 (G_{best} - x_{iz}) \tag{5-1}$$

$$x_{iz} = x_{iz} + v_{iz} \tag{5-2}$$

where ε is the inertia factor, c_1 and c_2 are two positive constants and r_1 and r_2 are two random functions in the range $[0,1]$.

Because of its simple mechanism and promising optimization ability in various problems, PSO has been successfully applied in many research and application areas [146-165]. Specially, PSO has been proved to be able to converge faster to an acceptable solution than other existing methods [161, 164, 166-170]. For the parameter optimization problems addressed in this research work, such faster convergence behavior is highly desirable. A robot navigating in a

dynamic environment must be able to respond to changes quickly, so it is crucial for the optimizer to find an acceptable solution as quickly as possible. Therefore, PSO has been adopted for parameter optimization of the F^2 method.

5.3 Particle Swarm Optimization Tuned Force Field Method

The Particle Swarm Optimization is utilized to solve the parameter optimization problems of the VSF^2 method, which results in the PSO-tuned F^2 method. Parameters in the VSF^2 method include k, P, Q, C, ρ_0. The goal is to find suitable parameters which will minimize the total length of paths. The feasibility of the PSO-tuned F^2 method is demonstrated by the simulations in this section.

5.3.1 Single Objective Parameter Optimization

In the VSF^2 method, k is a positive multiplier which determines the coverage area of a force field. P is a positive constant which determines the magnitude of repulsive force. Q is a positive constant which determines the magnitude of attractive force. C denotes the environmental influence on the coverage of a force field. ρ_0 is a positive fractional number with $0 < \rho_0 < 1$ and determines how close the robot can be separated from obstacles.

The i^{th} particle and its fitness value are defined as:

$$\mathbf{x_i} = [k\ P\ Q\ C\ \rho_0] \tag{5-3}$$

$$f(\mathbf{x_i}) = \sum_{i=1}^{W} FF(\mathbf{x_i}) \tag{5-4}$$

where $FF(\mathbf{x_i})$ is the resultant path length obtained from the VSF^2 method using parameters $\mathbf{x_i}$. W is the number of robots in the task.

When a robot is assigned a task, for example, travelling from a start point (X_{start}, Y_{start}) to a goal point (X_{goal}, Y_{goal}), the VSF^2 method will be called to drive this robot towards its destination while avoiding any potential collision with obstacles and other robots. When a robot reaches its destination, $FF(\mathbf{x_i})$ returns the real travel distance. If a robot fails to reach its destination using parameters $\mathbf{x_i}$ or collision occurs, $FF(\mathbf{x_i})$ returns with a very high value. The sum of $FF(\mathbf{x_i})$ from all robots in the working space is then used as a fitness value in PSO.

PSO calls the F^2 method repeatedly and tries to minimize the fitness value according to Equations (5-3) and (5-4). This process continues until the user-defined stopping criteria are satisfied. In the following simulations, the stopping criterion is set to be *50* iterations, i.e. PSO returns the optimization result after running *50* iterations.

5.3.2 Simulations Studies on Single Objective Optimization

For all simulations in this section, the parameters of robots are selected based on the Amigo robot (Figure 4-2) [136]. Each robot has an assigned task and each robot knows its start and goal positions. Robots are equipped with communication devices such that they know the status of other robots, including task priorities, velocities, positions, and so forth. Task priorities of all robots are set to be $T_p = 1$, i.e. no robot has priority over other robots during collision avoidance.

The first simulation is conducted in a simple case, that is, only one robot and one static circular obstacle are in the working space. A two-robot case and a four-robot case presented in Section 4.1.2 are further studied here. The stopping criteria of these simulations are set to be 50

iterations. The results with optimized parameters are compared with previous results to show the performance of the PSO-tuned F^2 method.

A. Case 1: Single Robot and One Obstacle

This simulation study is conducted using the example shown in Figure 5-1: a robot is supposed to travel from $S(2,3)$ to $G(8,7)$, and the centre of a circular obstacle with radius $R=1m$ is located at $(5,4.5)$. The ranges of the parameters are empirically set as: $k\in[2,10]$, $P\in[5,20]$, $Q\in[5,20]$, $C\in[1.5,3]$, $\rho_0\in[0.2,1]$. Parameters for PSO, c_1 and c_2 in Equation (5-1), are set to be 2. The inertia factor ε is set to be a random decimal fraction between $[0,0.5]$. After running PSO for *50* iterations, the optimization process is shown in Figure 5-2. In Figure 5-2, the solid line shows the resultant path length at each iteration. The values of k, P, Q, C, ρ_0 in each iteration are shown by the dashed line, dashdotted line, dashdotted line, dashed line and solid line, respectively. In this case, the PSO reaches a steady state after *39* iterations. The optimal values of the parameters are: $k=4.238$, $P=9.566$, $Q=9.268$, $C=1.623$ and $\rho_0=0.298$. The optimal path length obtained is $7.106m$, which is much shorter than those using parameters listed in Rows *1* to *5* in Table 5-1.

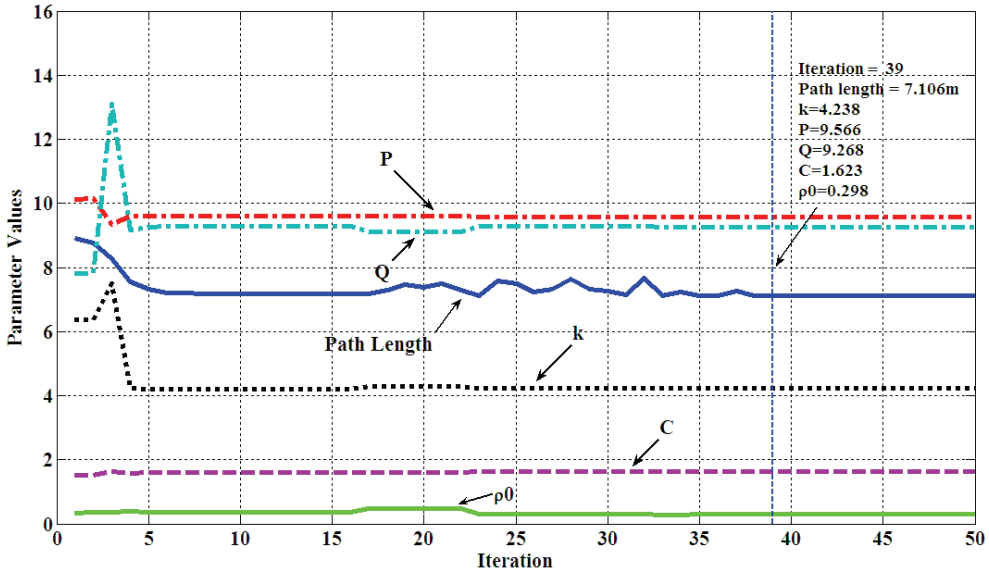


Figure 5-2 Single objective optimization Case 1: optimization results

B. Case 2: Two Robots Navigating in a Corridor

The case study scenario presented in Section 4.1.2 is further studied here as Case 2. In Figure 5-3, Robot 1 is supposed to travel from $S_1(1,5.1)$ to $G_1(9,5.4)$ and Robot 2 travels from $S_2(9,5)$ to $G_2(1,5)$. The distance between the two obstacles (walls) is $1.5m$. Previous results obtained in Section 4.1.2 are listed in the first row in Table 5-2. The path of Robot 1 is denoted by the dashed line and the path of Robot 2 is shown by the dotted line.

The parameter ranges are empirically set as: $k \in [2,10]$, $P \in [5,20]$, $Q \in [5,20]$, $C \in [1.5,3]$, $\rho_0 \in [0.2,1]$. Parameters for PSO, c_1 and c_2 in Equation (5-1), are set to be 2. The inertia factor ε is set to be a random decimal fraction between $[0,0.5]$. For each optimization, PSO was executed for *50* iterations. Rows 2 to 6 in Table 5-2 list some results obtained using the PSO-tuned F^2 method.

Figure 5-3 shows the paths using parameters listed in the 6th row of Table 5-2. Paths of Robots 1 and 2 are denoted by solid line and dashdotted line, respectively. The corresponding optimization process is shown in Figure 5-4. The solid line shows the resultant path lengths of each iteration. The values of k, P, Q, C, ρ_0 in each iteration are shown by the dotted line, dashdotted line, dashdotted line, dashed line and solid line, respectively. In this optimization process, PSO reaches a steady state after *41* iterations. The parameter values are: $k = 3.777$, $P = 7.479$, $Q = 19.356$, $C = 3.272$ and $\rho_0 = 0.488$. The path length using optimized parameters is $15.449m$, which improves 5.81% compared with the non-optimized result ($16.402m$) in terms of path length.

Set of Parameters	The Sum of Path Lengths of Two Robots (m)	k	P	Q	C	ρ_0	Improve-ment
1	16.402 (from Section 4.1.2)	5	20	5	2	0.2	N/A
2	15.441	2.301	17.019	45.979	2.154	0.411	5.86%
3	15.460	2.634	11.309	25.490	2.625	0.648	5.74%
4	15.534	3.653	9.843	30.505	2.662	0.323	5.29%
5	15.660	0.853	9.461	19.418	1.112	0.299	4.52%
6	15.449	3.777	7.479	19.356	3.272	0.488	5.81%

Table 5-2 Parameters in single objective optimization Case 2

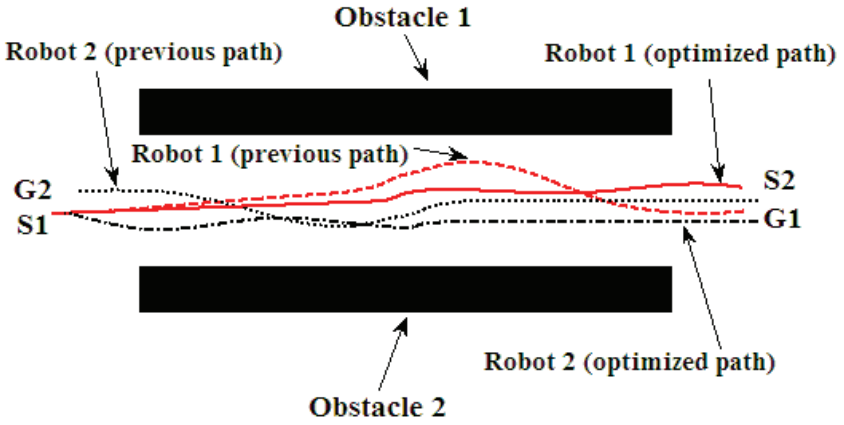


Figure 5-3 Single objective optimization Case 2: two robots in a corridor

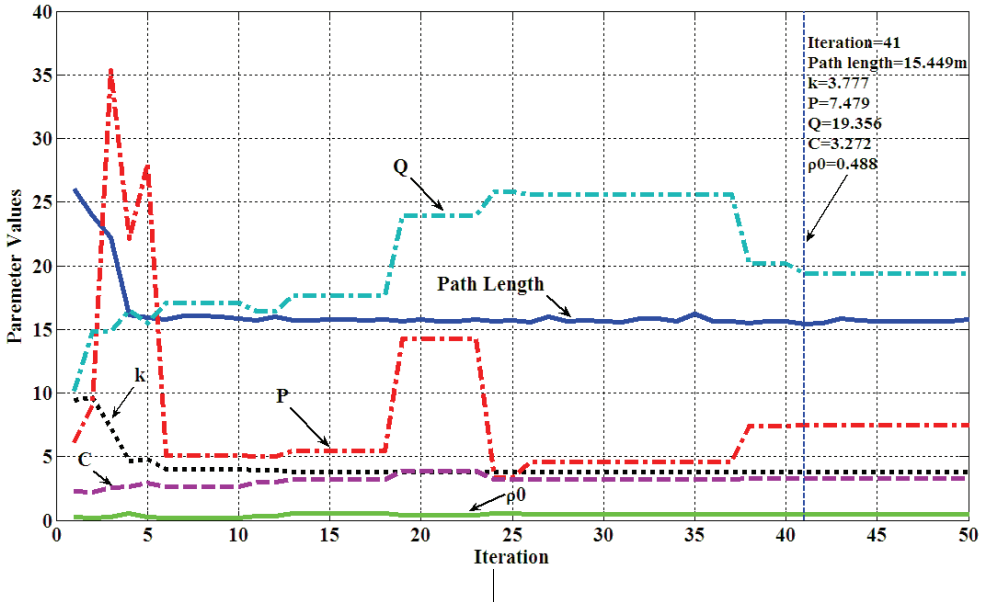


Figure 5-4 Single objective optimization Case 2: optimization results

C. Case 3: Four Robots in an Indoor Environment

The PSO-tuned F^2 method is further tested in a four-robot navigation scenario studied in Section 4.1.2. Figure 5-5 shows the indoor environment with static obstacles. The dimension of this area is $20m$ by $20m$. Robot 1 is supposed to travel from $S_1(18,1)$ to $G_1(7,18)$. Robot 2 is supposed to travel from $S_2(2,18)$ to $G_2(18,4)$. Robot 3 travels from $S_3(1,4)$ to $G_3(17,12)$ and Robot 4 from $S_4(18,18)$ to $G_4(7,2)$.The parameter ranges are the same as those used in Case 2.

The parameters and the resultant path length obtained in Section 4.1.2 are listed in Row 1 of Table 5-3. Rows 2 to 6 show the optimized parameters and the sum of the resultant path lengths of four robots obtained by the PSO-tuned F^2 method. In Figure 5-5, the paths of the four robots obtained using parameters in Row 2 of Table 5-3 are shown by the dotted line, solid line, dashed line and dashdotted line, respectively. The parameter values are $k=2.079$, $P=5.310$, $Q=11.427$, $C=1.157$ and $\rho_0=0.441$. The corresponding path lengths for Robots 1, 2, 3 and 4 are $21.023m$, $23.484m$, $18.506m$ and $19.399m$, respectively.

Compared with the previous results in Section 4.1.2, the total length is reduced from $89.061m$ to $82.412m$, an improvement of 7.47%. Path lengths for Robots 1, 2, 3 and 4 are reduced from $23.844m$, $25.27m$, $18.629m$ and $21.318m$ to $21.023m$, $23.484m$, $18.506m$ and $19.399m$, respectively.

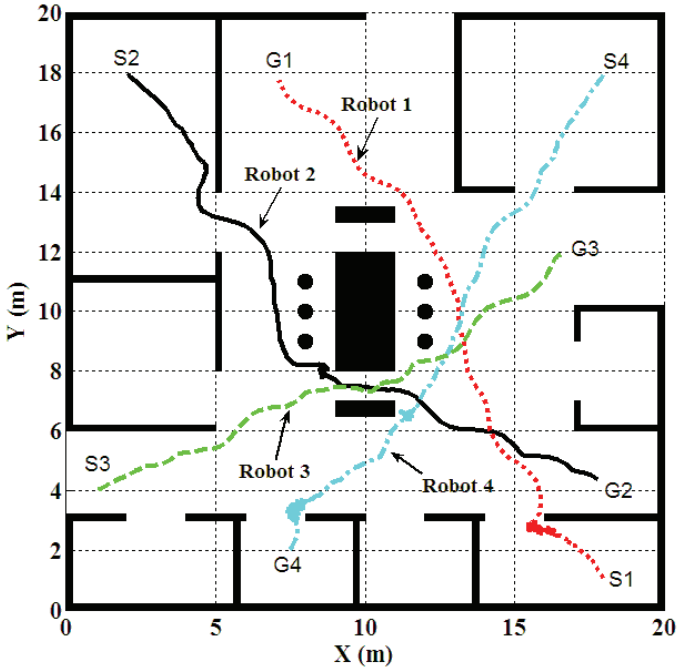


Figure 5-5 Single objective optimization Case 3: four robots navigation

Sets of Parameters	The Sum of Path Lengths of Four Robots (m)	k	P	Q	C	ρ_0	Improv e-ment
1	89.061 (from Section 4.1.2)	6	20	5	2	0.2	N/A
2	82.412	2.079	5.310	11.427	1.157	0.441	7.47%
3	85.906	8.979	29.543	13.454	2.701	0.348	3.54%
4	88.919	6.397	10.874	9.327	2.217	0.508	0.16%
5	88.297	7.312	10.389	6.939	2.575	0.584	0.86%
6	86.820	4.658	9.041	9.188	1.469	0.427	2.52%

Table 5-3 Parameters in single objective optimization Case 3

5.3.3 Conclusions on Particle Swarm Optimization Tuned Force Field Method

The PSO-tuned F^2 method has been described in detail. The parameter settings in the F^2 based methods have significant effect on the performance of the methods, which reinforces the requirement for finding optimal parameters. Parameters optimised in this optimization problem are: k, P, Q, C, ρ_0. The PSO approach is applied to find the optimal parameters which minimize the resultant path length. The proposed approach is then tested in three typical environments. Simulation results show that this approach is capable of finding appropriate parameters for the F^2 method.

5.4 Ranked Pareto Particle Swarm Optimization Method for Multiobjective Parameter Optimization

The PSO-tuned F^2 method presented in Section 5.3 is suitable for single objective optimization, i.e. to minimize a robot's path length. However, most real-world motion planning problems have several objectives. For example, a robot may need to keep as far as possible from obstacles while trying to minimize its travel distance. This necessitates multiobjective optimization. A novel multiobjective optimization method - Ranked Pareto Particle Swarm Optimization (RPPSO) - is proposed in this section for solving the multi-objective motion planning problems of multiple mobile robots by taking into account collision avoidance between robots and obstacles as well as minimum travel distance.

At the beginning of RPPSO, particles are initiated randomly in the search space. These particles are then ranked by their qualities with regard to all objectives. Those particles with good qualities constitute the Global Best which stores the current best solutions found by particles. At the same time, each particle maintains a memory of its best position, denoted by Particle Best. Particles then update their locations based on the Global Best and the Particle Best. Thus particles in RPPSO will search many possible directions. Ideally, a set of optimal solutions will be found when the termination criterion is met.

5.4.1 Key Concepts in Multiobjective Optimization Problems

The concept of *Pareto Dominance* proposed has been widely used in the research of the Multiobjective Optimization Problem (MOP). In general, an objective vector $\mathbf{F_a}$ in a minimization problem is said to dominate another objective vector $\mathbf{F_b}$, denoted by $\mathbf{F_a} \prec \mathbf{F_b}$, if and only if

$$\mathbf{f_{a,i}} \leq \mathbf{f_{b,i}} \quad \forall \; i \in \{1,2,\ldots,m\} \tag{5-5}$$

and

$$\mathbf{f_{a,j}} \leq \mathbf{f_{b,j}} \quad \exists \; j \in \{1,2,\ldots,m\} \tag{5-6}$$

where $\mathbf{f_{a,i}}$ and $\mathbf{f_{a,j}}$ are components of $\mathbf{F_a}$, $\mathbf{f_{b,i}}$ and $\mathbf{f_{b,j}}$ are components of $\mathbf{F_b}$.

For a multiobjective problem, a solution is said to be *Pareto Optimal* if it is not dominated by other solutions in the search space. The set of all Pareto Optimal solutions is called a *Pareto Set*. The aim of a Multiobjective Optimization Problem (MOP) is to derive a set of non-dominated solutions with objective values as close as possible to the objective values of the *Pareto Set*, denoted by *Pareto Front*. The concept of Pareto Optimal is a useful tool. For multiobjective robot motion planning problems, the quality of a resultant path can be evaluated from several aspects, such as the path length, path smoothness and safe distance to obstacles. For an example of the application of Pareto Optimal in robot motion planning, see [171].

The so-called Multiobjective Evolutionary Algorithm (MEA) has been shown to be suitable for solving multiobjective problems [172-176]. At each iteration, *fitness values* (relevant to *objective values*) are evaluated in order to determine better solutions for the next generation. As a result, this will ideally lead to a population of optimal solutions when some termination condition is satisfied. Following this, MEAs have to overcome two major problems [177] . The first problem is how to get close to the Pareto Optimal Front. The second is how to retain

diversity among the solutions in the obtained set. These two problems have become common criteria for most algorithmic performance comparisons.

The concept of PSO is suitable for multiobjective optimization. Firstly, PSO is able to find many nondominated solutions with a single run because of its inherent parallel nature. Secondly, the mechanism of PSO is simple so that it can be easily combined with any multiobjective optimization functions and constraints. Thirdly, in PSO, the movement of each particle depends upon its own best history and best locations found by other members of the swarm, which ensures a better balance between the speed of convergence and the search-space exploration. Last but not least, PSO has been found to be very effective and able to produce good results at a low computational cost.

5.4.2 Ranked Pareto Particle Swarm Optimization Method

The Ranked Pareto Particle Swarm Optimization method (RPPSO) follows the basic idea of the Particle Swarm Optimization method (PSO). The flowchart of the RPPSO is given in Figure 5-6 (where equations invoked in calculating the variables are indicated within brackets). Symbols and abbreviations used are listed in Table 5-4. The equations involved in each process are enclosed in brackets in the flowchart.

A. Fitness Assignment

Consider a multiobjective minimization problem,

$$\min_{P\in\phi} \mathbf{F(X)}, \quad \mathbf{X} \in \mathbf{R}^{\xi}$$

where $\mathbf{X} = \{x_1, x_2, \ldots, x_\xi\}$ is an ξ-dimensional vector having ξ decision variable or parameters and ϕ defines a feasible set of P. $\mathbf{F} = \{f_1, f_2, \ldots, f_\psi\}$ is a vector with ψ objective functions to be minimized.

At the beginning of iterations, particles are initialized randomly in the search space. Assume particles are denoted by $\mathbf{X_i}$. As stated above, $\mathbf{X_i}$ is an ξ-dimensional vector which has ξ decision variables (parameters in this research):

$$\mathbf{X_i} = \left\{ x_{i1}, x_{i2} ..., x_{i\xi} \right\}, \quad i = 1, 2, \ldots, PN \tag{5-7}$$

where *PN* is the total number of particles.

Symbols	Descriptions
ξ	number of decision variables or parameters
ψ	number of objectives to be minimized
PN	particle numbers
RFV	raw fitness value
RFVM	raw fitness value matrix
MFV	modified fitness value
MFVV	modified fitness value vector
GMFV	global modified fitness vector
PRV	particle rank value
PCN	particle copy number
G_{best}	global best
P_{best}	particle best

Table 5-4 Nomenclature in RPPSO method

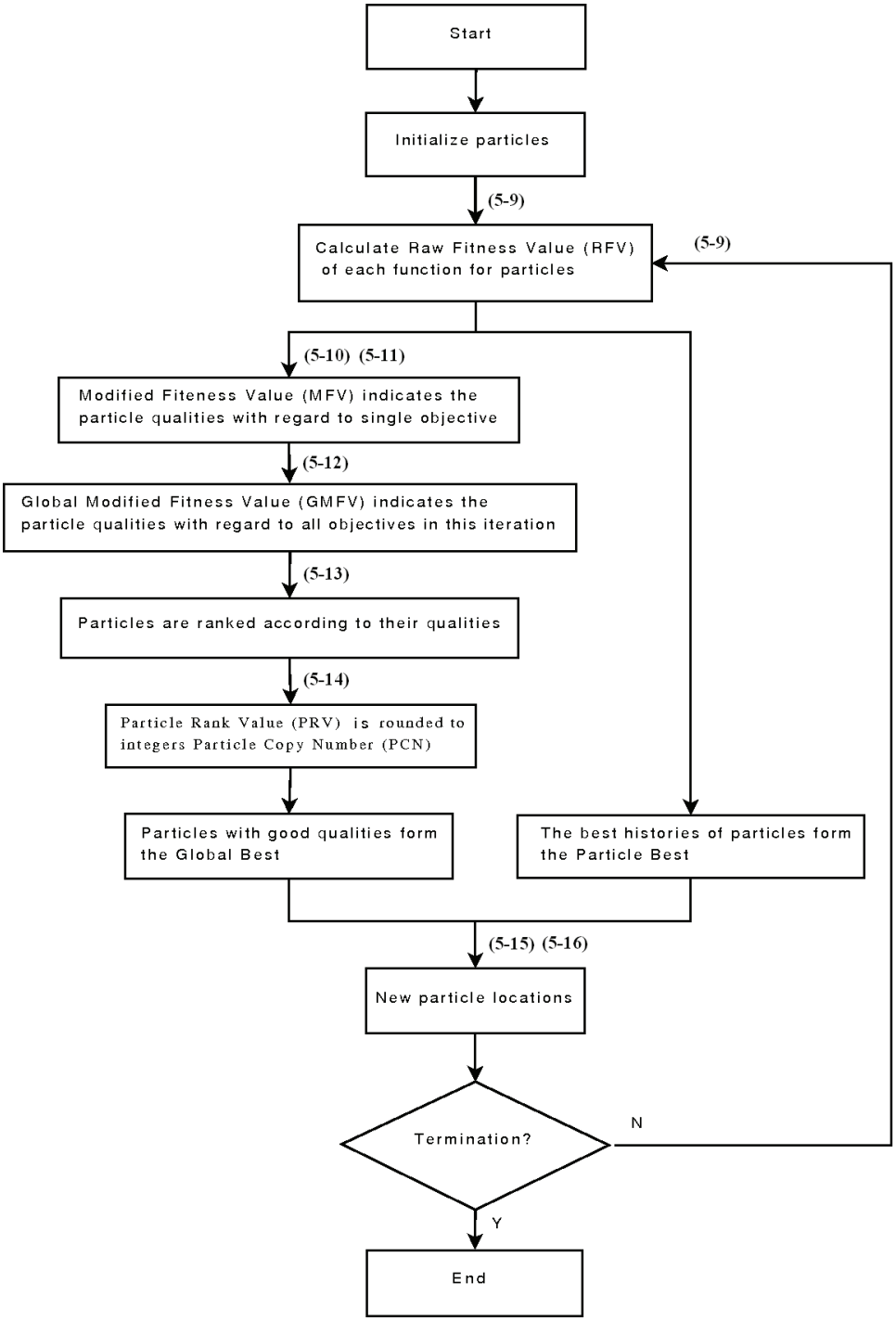


Figure 5-6 RPPSO flowchart

For any objective j, a raw fitness value (*RFV*) is obtained for each particle:

$$RFV_{j,i} = f_j\left(\mathbf{X_i}\right), \quad j = 1,2,\ldots,\psi \tag{5-8}$$

where *i* is the particle number and *j* is the objective function number. Thus with regard to each objective *j*, a raw fitness value matrix (*RFVM*) is attained for each iteration, which is a $\psi \times PN$ array of raw fitness values:

$$RFVM = \begin{bmatrix} RFV_{1,1} & RFV_{1,2} & \cdots & RFV_{1,PN} \\ RFV_{2,1} & RFV_{2,2} & \cdots & RFV_{2,PN} \\ \vdots & \vdots & \ddots & \vdots \\ RFV_{\psi,1} & RFV_{\psi,2} & \cdots & RFV_{\psi,PN} \end{bmatrix} \tag{5-9}$$

The raw fitness value of each particle is then compared with raw fitness values of all other particles. To set a comparison criterion, define that for a minimization problem f_j, particle i_1 is said to dominate particle i_2 with regard to objective f_j if and only if the raw fitness value of particle i_1 is smaller than that of particle i_2,

$$RFV_{j,i_1} < RFV_{j,i_2} \quad j = 1,2,\ldots,\psi \tag{5-10}$$

For each particle, the modified fitness value (*MFV*) on an objective is defined as the total number of other particles which are dominated by this particle. *MFV* shows the quality of particles regarding objective f_j, i.e. particles with higher *MFV* are more desirable than those particles with lower *MFV*. For objective f_j, a *PN*-dimensional vector of modified fitness value ($\mathbf{MFVV_j}$) in a generation is obtained:

$$\mathbf{MFVV_j} = \left\{MFV_{j,1}, MFV_{j,2}, \ldots, MFV_{j,PN}\right\}, \quad j = 1,2,\ldots,\psi \tag{5-11}$$

For every objective, the same process is repeated and a **MFVV** is obtained. The sum of all **MFVV** s is called a global modified fitness vector (**GMFV**), which is a *PN*-dimensional vector:

$$\mathbf{GMFV} = \sum_{j=1,2,\ldots,\psi} \mathbf{MFVV_j} \tag{5-12}$$

The **GMFV** can be treated as a benchmark of particle general quality with regard to all objectives. Particles with higher **GMFV** s have higher qualities than those with lower **GMFV** s in general. Note that a particle with higher **GMFV** does not need to have better *MFV*s (or *RFV* s) on *each* objective than those with lower **GMFV** .

B. Global Best

The Global Best in RPPSO stores the current best solutions found by particles (denoted by $\mathbf{G_{best}}$). In standard PSO, Global Best stores *one* best solution found by all particles. Global Best in RPPSO does almost the same task, except that Global Best in RPPSO is a *PN*-dimensional vector which stores *PN* best solutions found by all particles. The particle rank value (*PRV)* of i^{th} particle is defined by

$$PRV_i = \frac{\mathbf{GMFV_i}}{\sum_{i=1,2,\ldots,PN} \mathbf{GMFV_i}} \times PN, \quad i = 1,2,\ldots,PN \tag{5-13}$$

The PRV_i is a decimal fraction which is proportional to the $\mathbf{GMFV_i}$ of a particle. This value is then rounded to an integer, denoted by Particle Copy Number (*PCN*), which determines how many copies of this particle will enter the Global Best of this generation:

$$PCN_i = f\left(PRV_i\right), \quad i = 1,2,\ldots,PN \tag{5-14}$$

To ensure that there are enough candidates for Global Best, the function $f(PRV_i)$ rounds to the nearest integer towards infinity. The particles which hold higher *PRV*s have the priority to enter the Global Best vector. This process continues until the Global Best vector is filled up. Particles with lower *PRV*s, not contributing to solution improvements, will not enter the Global Best vector if Global Best vector has been filled by copies of particles with higher *PRV*s and redundant particles will be abandoned.

C. Particle Best

The best history of all particles is stored in a *PN*-dimensional vector, denoted by $\mathbf{P_{best}}$. To achieve this, the algorithm keeps the records of the best history of each particle. For each particle, its current location will replace its $\mathbf{P_{best}}$ only when its current raw fitness values (*RFV*s) for all objectives are better than those in $\mathbf{P_{best}}$.

D. Update

Each particle updates its position according to the following equations.

$$\mathbf{v_{new}} = \mathbf{v_{old}}\varepsilon + c_1 r_1 \left(\mathbf{P_{best}} - \mathbf{X_{old}}\right) + c_2 r_2 \left(\mathbf{G_{best}} - \mathbf{X_{old}}\right) \tag{5-15}$$

$$\mathbf{X_{new}} = \mathbf{X_{old}} + \mathbf{v_{new}} \tag{5-16}$$

where ε is the inertia factor, c_1 and c_2 are positive constants and r_1 and r_2 are two functions returning random numbers in the range $[0,1]$.

These equations are similar to those of standard PSO. However G_{best} in standard PSO usually only stores one best location for all particles. In RPPSO, $\mathbf{G_{best}}$ is a *PN*-dimensional vector which stores the best *PN* positions found by particles. As a consequence, particles in

RPPSO will search many possible directions. In this way, the diversity among solutions is well preserved.

5.4.3 Case Study

A multiobjective optimization problem is studied to illustrate the process of RPPSO. This problem consists of two variables (x_1 and x_2) and two objective functions (f_1 and f_2).

$$f_1(x_1, x_2) = x_1^2 - 2x_2, \quad x_1, x_2 \in [0,5] \tag{5-17}$$

$$f_2(x_1, x_2) = \sqrt{|x_1 - 0.2|} + (2x_2 + x_1)^2, \quad x_1, x_2 \in [0,5] \tag{5-18}$$

The particle number (PN) in this simulation is 5. In Equation (5-15), the inertia factor ε is set to be a random decimal fraction between $[0,0.5]$, c_1 and c_2 are set to be 2. Particles 1 to 5 are denoted by star, square, diamond, circle and plus sign, respectively. Figure 5-7 (a) shows the particle locations in the *4th* generation. (b) and (c) show the locations of $\mathbf{G}_{\mathbf{best}}$ and $\mathbf{P}_{\mathbf{best}}$ before the 4^{th} generation. (e) and (f) show the locations of $\mathbf{G}_{\mathbf{best}}$ and $\mathbf{P}_{\mathbf{best}}$ after the *4th* generation. (d) shows the updated particle locations in the *5th* generation. There are two points which should be noted in Figure 5-7.

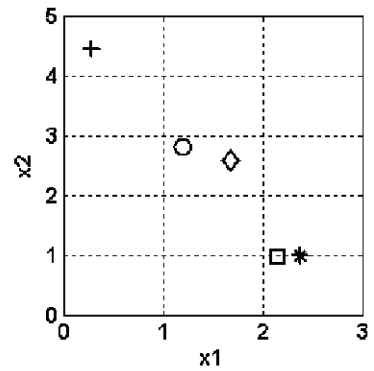
- $\mathbf{G}_{\mathbf{best}}$ in RPPSO may contain some duplicated particles, e.g., square and star in (b), circle and diamond in (e). That means that good particles have a higher chance of entering the next generation than others.

- From (c) and (f), it can be found that some values of $\mathbf{P}_{\mathbf{best}}$ changed in this generation (square and star). This is because Particle 1 (star) and Particle 2 (square) have better fitness values than before and replaced the corresponding $\mathbf{P}_{\mathbf{best}}$. See (a), (c) and (f).

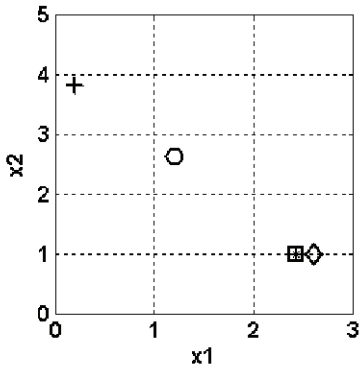
When the number of particles is increased to *2000*, Figure 5-8 shows the optimization results. The gray dots in Figure 5-8 illustrate the particle locations and the black stars show the Pareto Optimal Set.

If a third objective function $f_3(x_1, x_2) = x_2$ is added into this problem, the optimization results are given in Figure 5-9 again. The gray dots show the particle locations and black stars show the Pareto Optimal Set. The number of particles in this simulation is *2000* as well.

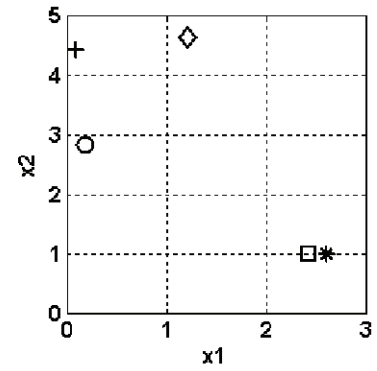
In this section, a multiobjective optimization approach, the Ranked Pareto Particle Swarm Optimization method (RPPSO), has been detailed for solving multiobjective optimization problems. In RPPSO, particles are ranked by their qualities with regard to all optimization objectives. Those particles with better qualities constitute the Global Best. It has been proved by the case study that RPPSO is capable of solving multiobjective optimization problems efficiently.



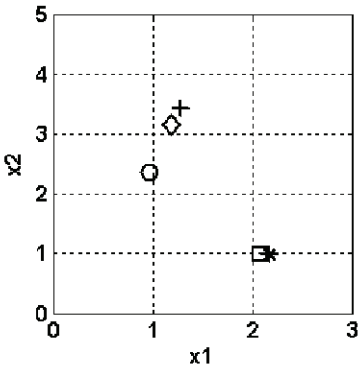
(a) Particle locations in the 4th generation



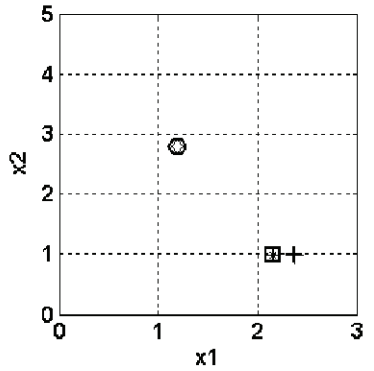
(b) Gbest locations before the 4th generation



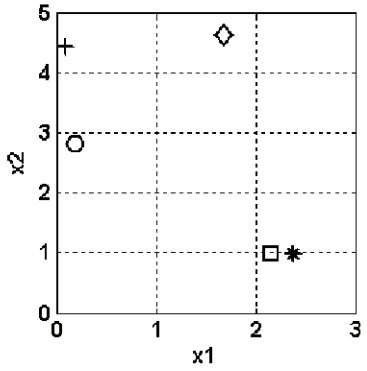
(c) Pbest locations before the 4th generation



(d) Particle locations in the 5th generation



(e) Gbest locations after the 4th generation



(f) Pbest locations after the 4th generation

Figure 5-7 Snapshots of the progress of RPPSO

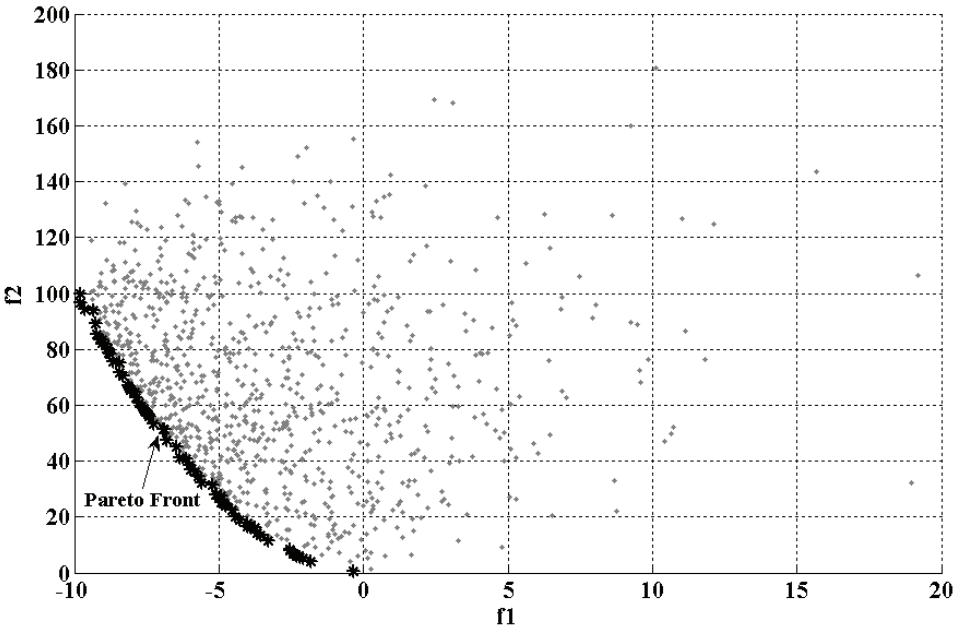


Figure 5-8 RPPSO optimization results – 2 objectives

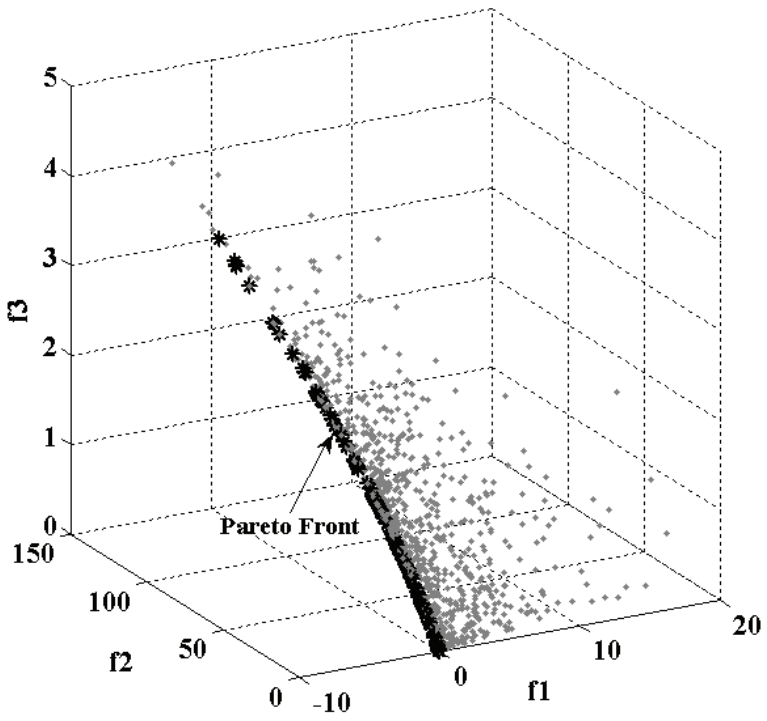


Figure 5-9 RPPSO optimization results – 3 objectives

5.5 Multiobjective Optimization of Force Field Method

The case study presented in Figure 5-1 and Section 5.3 is further studied here for comparison. Path 6 (solid line) is the optimal result obtained by the PSO-tuned F^2 method, which is much shorter than those using non-optimal parameters. In Path 6, it can be found that the robot runs very close to the obstacle. Although the path length is reduced significantly, moving very close to the obstacle may be undesirable in real applications for safety considerations. Thus, an approach which takes into account both efficiency (e.g., path length) and safety (e.g., sufficient safety distance) is needed. This multiobjective problem can be well solved by the RPPSO.

As analysed in Section 5.1, the parameters to be optimized are k, P, Q, C, ρ_0. So the i^{th} particle is defined as:

$$\mathbf{x_i} = [k, P, Q, C, \rho_0], \quad i = 1, 2, \ldots, PN \tag{5-19}$$

The optimization objectives are set to be:

- to minimize the travel length
- to keep away from obstacles and other robots as far as possible

The first objective function is given by:

$$f_1(\mathbf{x_i}) = FF(\mathbf{x_i}), \quad i = 1, 2, \ldots, PN \tag{5-20}$$

where $FF(\mathbf{x_i})$ is the resultant path length obtained from the VSF^2 method using parameters $\mathbf{x_i}$. When a robot reaches its destination, $f_1(\mathbf{x_i})$ will return the value of the real travel distance. If a robot fails to reach its destination using these parameters or collision occurs, $f_1(\mathbf{x_i})$ will return a big value.

The second objective is to let a robot keep away from obstacles as far as possible. Assume that in each time cycle, the shortest distance from a robot (with parameters $\mathbf{x_i}$) to obstacles in the working space is denoted by $D_{n,\mathbf{x_i}}$. The danger index is defined as:

$$f_2(\mathbf{x_i}) = \sum \frac{1}{D_{n,\mathbf{x_i}}}, \quad i = 1,2,\ldots,PN \tag{5-21}$$

If a robot travels close to obstacles, $f_2(\mathbf{x_i})$ will return a bigger value. On the other hand, when a robot travels far from obstacles, $f_2(\mathbf{x_i})$ will result in a smaller value. The optimization objective is then turned into finding appropriate $\mathbf{x_i}$ to minimize $f_2(\mathbf{x_i})$.

For a motion planning problem with multiple robots, each robot needs to avoid collisions with obstacles and other robots while minimizing its path length. If the number of robots is given by m, Equation (5-20) and (5-21) can be redefined as:

$$f_1(\mathbf{x_i}) = \sum_{j=1,2,\ldots m} FF(\mathbf{x_i}), \quad i = 1,2,\ldots,PN \tag{5-22}$$

$$f_2(\mathbf{x_i}) = \sum_{j=1,2,\ldots m} \sum \frac{1}{D_{n,\mathbf{x_i}}}, \quad i = 1,2,\ldots,PN \tag{5-23}$$

For all simulations in this section, robot parameters are selected based on those of the Amigo robot [136]. Each robot has an assigned goal and knows its start and goal positions. Robots are assumed to have communication devices so that they know the status of other robots, including task priorities, velocities, positions, and so on.

Task priorities of all robots are set to be $T_p = 1$, i.e. no robot has priority over other robots during collision avoidance. The ranges for parameters are same as those used in Section 5.3.2, i.e. $k \in [2,10]$, $P \in [5,20]$, $Q \in [5,20]$, $C \in [1.5,3]$, $\rho_0 \in [0.2,1]$. For all optimization

processes, the inertia factor ε is set to be a random decimal fraction between $[0,0.5]$, c_1 and c_2 are set to be 2.

A. Multiobjective Optimization Case 1: Single Robot and One Obstacle

A robot is supposed to travel from locations $S_1(2,3)$ to $G_1(8,7)$, as shown in Figure 5-10. An obstacle is located in this work space. The time cycle of simulation is *0.1s*.

The number of particles of the RPPSO process in this simulation is *1000*. The termination criterion is set to be *100* iterations. In this case, RPPSO reaches a steady state after *60* generations. The optimization results are shown in Figure 5-11, in which the optimal set is shown in black stars and other particles are shown in gray dots. Figure 5-12 shows the parameter values of particle B (the *3rd* row in Table 5-5) during the iterations. The optimal values of parameters are: $k = 4.5173$, $P = 10.8845$, $Q = 16.3748$, $C = 1.1476$ and $\rho_0 = 0.4594$.

In Table 5-5, simulation results of the PSO-tuned F^2 method are also given at the *1st* row. Rows 2 to 6 are simulation results obtained by RPPSO, denoted by points A, B, C, D, E (particles in the Pareto Front) in Figure 5-11. As shown in Table 5-5, the resultant parameters fall within the same order of magnitude. Using the parameters of points A, B, C, D and E, the corresponding path lengths obtained are $7.5916m$, $7.6023m$, $7.6091m$, $7.6177m$ and $7.7501m$, respectively, which are close to, but slightly longer than, the path lengths obtained by the PSO-tuned F^2 method ($7.106m$) in Section 5.3.2. At the same time, the danger index is reduced from 43.5490 to 5.2414, 5.2162, 5.1895, 5.1599 and 5.1409, respectively (last column in Table 5-5). That means that the distance between the robot and the obstacle during the robot's movement is larger than the previous case, as shown in Figure 5-10. Thus, the effectiveness of the proposed multiobjective optimization method is able to significantly reduce the danger index, i.e. increase the safety.

The path obtained using parameters of point B is shown in a solid line in Figure 5-10. Compared with the previous result of the PSO-tuned F^2 method denoted by the dashed line, it is clear that the new path follows a larger contour around the circular obstacle. The robot is now capable of keeping a safe distance from obstacles while minimizing the path length.

B. Multiobjective Optimization Case 2: Two Robots Navigating in a Corridor

In this simulation Robot 1 is supposed to travel from location $S_1(1,5.25)$ to $G_1(9,5)$ and Robot 2 from $S_2(9,5)$ to $G_2(1,4.75)$. The distance between the two obstacles (black patches) is $2m$. The two Robots start at the same time from their start points respectively. The time cycle is set to be $0.01s$. Robots 1 and 2 use the same set of parameters in every iteration in the RPPSO based design process.

Figure 5-13 shows the results obtained by the RPPSO after *50* generations using *50* particles. Particles of the optimal set (6 in total) are shown in black stars and other particles are shown in gray dots. Details of these optimal particles are given in Table 5-6. The parameters corresponding to Particle C indicated by the circle (Figure 5-13) are listed in the *3rd* row in Table 5-6. Its resultant paths are shown in Figure 5-14. The paths of Robots 1 and 2 are shown in dashed and solid lines, respectively.

In Figure 5-15, X coordinate is time and Y coordinate shows the nearest distances from a robot to obstacles (including obstacles and another robot). The distances to obstacles with regard to Robots 1 and 2 are shown in a dashed line and solid line, respectively. This simulation has proved that the PRRSO is suitable for solving the multiobjective parameter optimization problems in the VSF^2 parameter refinement.

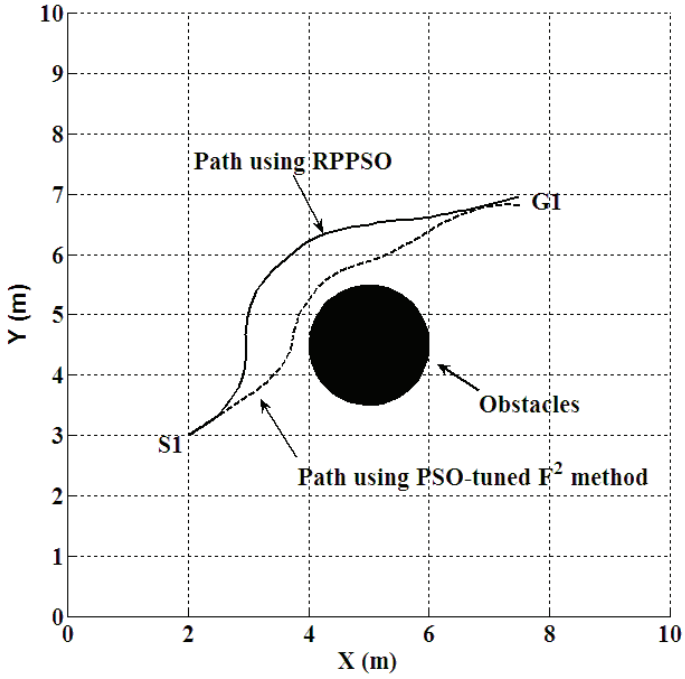


Figure 5-10 Multiobjective optimization Case 1: resultant path

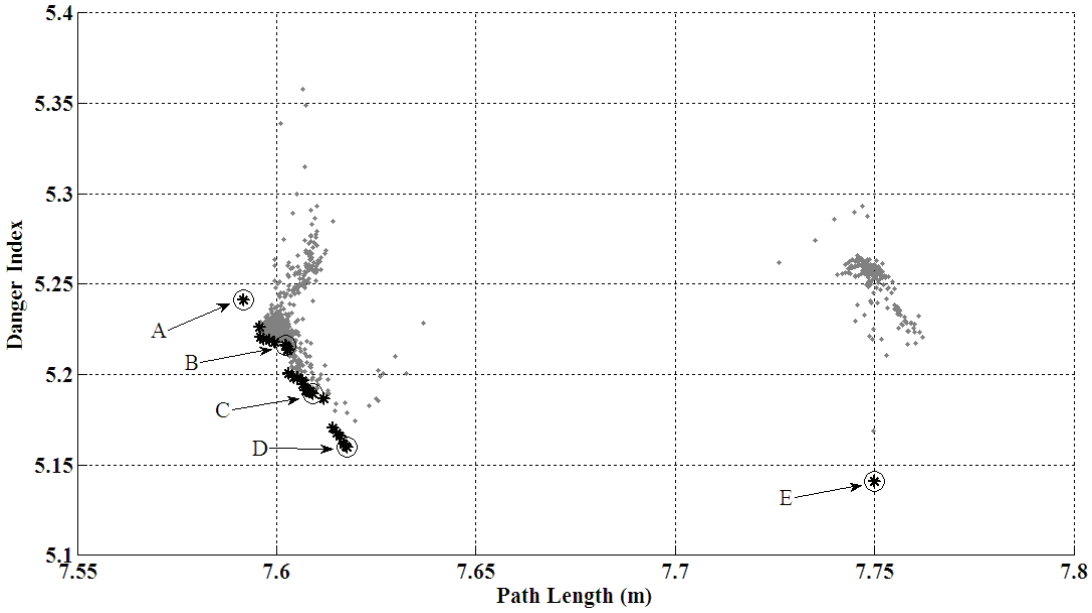


Figure 5-11 Multiobjective optimization Case 1: Pareto optimal set

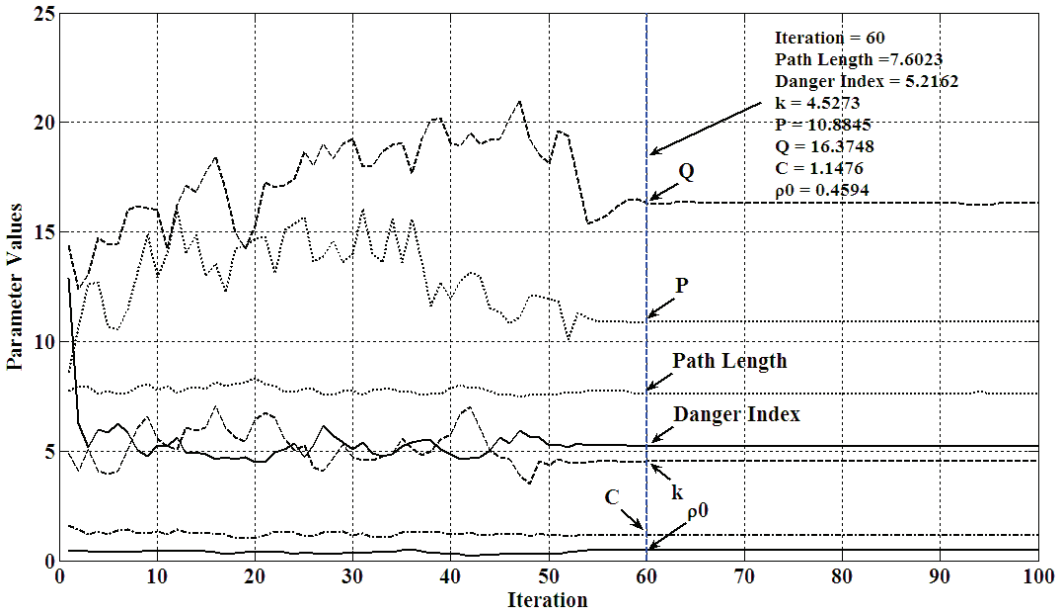


Figure 5-12 Multiobjective optimization Case 1: evaluation of optimized parameters

Test	k	P	Q	C	ρ_0	Path Length (m)	Danger Index
1	4.2380	9.5660	9.2680	1.6230	0.2980	7.1060	43.5490
A	4.5068	10.9423	15.7974	1.1488	0.4605	7.5916	5.2414
B	4.5273	10.8845	16.3748	1.1476	0.4594	7.6023	5.2162
C	4.5280	10.8821	16.5471	1.1479	0.4587	7.6091	5.1895
D	4.5265	10.9539	16.7658	1.1475	0.4574	7.6177	5.1599
E	4.7715	10.8570	16.3219	1.1497	0.4588	7.7501	5.1409

Table 5-5 Multiobjective optimization Case 1: optimization results

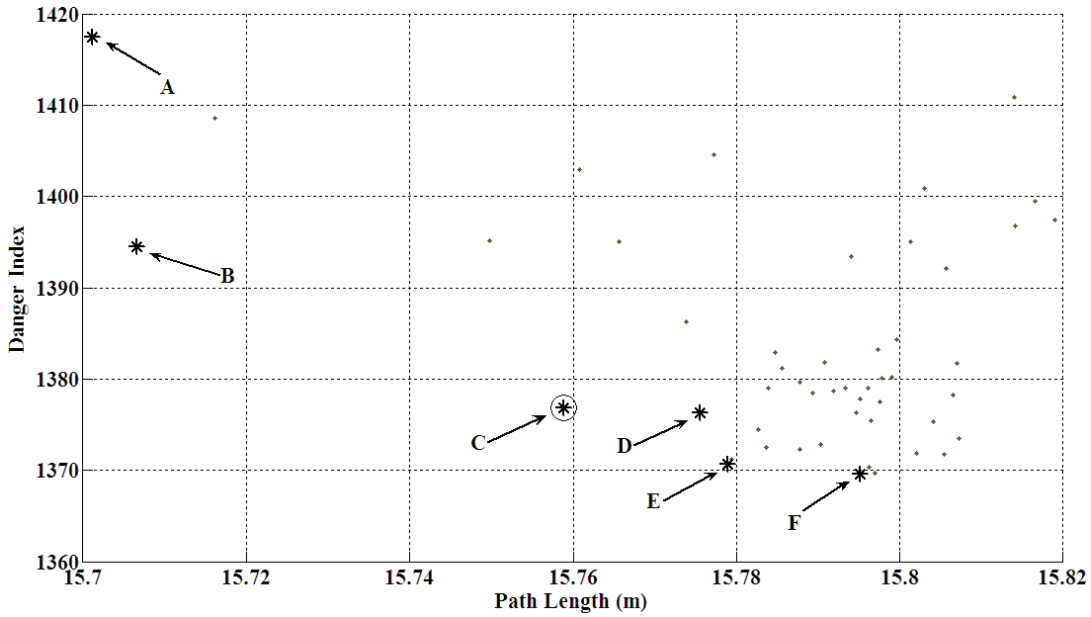


Figure 5-13 Multiobjective optimization Case 2: Pareto optimal set

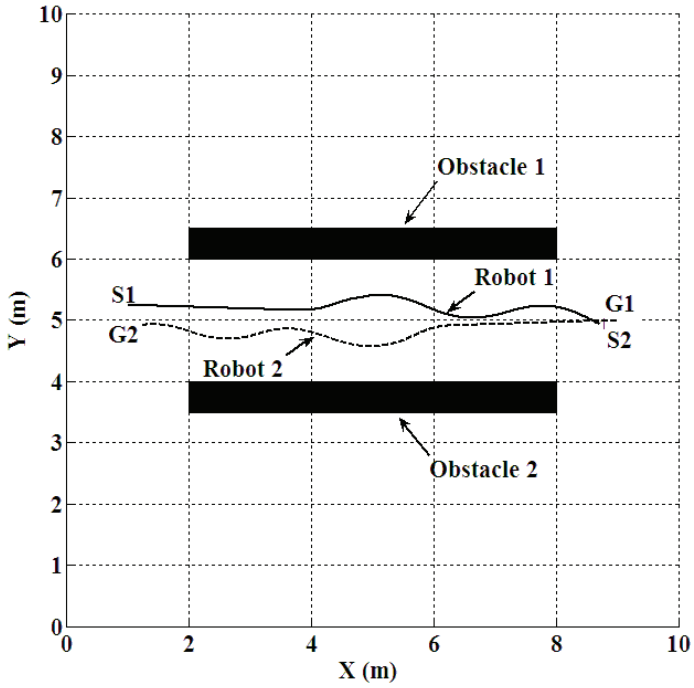


Figure 5-14 Multiobjective optimization Case 2: resultant paths

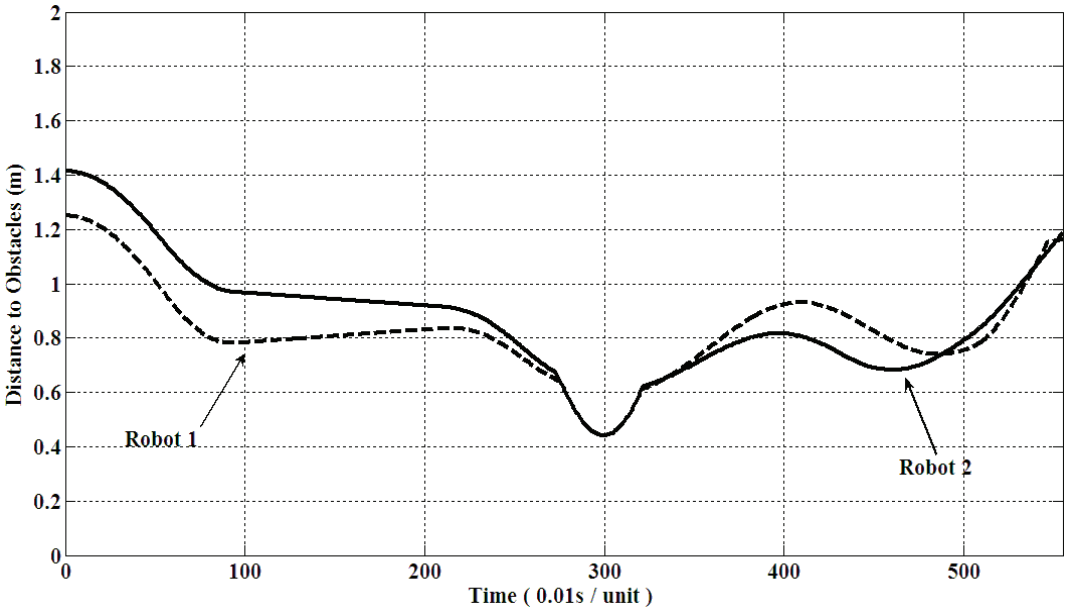


Figure 5-15 Multiobjective optimization Case 2: distance to obstacles

Test	k	P	Q	C	ρ_0	Path Length (m)	Danger Index
A	2.4208	17.1789	19.3420	1.3274	0.4159	15.7011	1417.4227
B	2.3219	17.0566	19.7220	1.3278	0.4647	15.7066	1394.4961
C	2.3407	17.8411	19.6860	1.3381	0.3980	15.7589	1376.8007
D	2.3524	17.8243	19.9060	1.3340	0.3958	15.7755	1376.3172
E	2.2883	16.7237	19.4253	1.3278	0.3920	15.7789	1370.6500
F	2.3047	16.2929	20.0047	1.3304	0.4802	15.7951	1369.6322

Table 5-6 Multiobjective optimization Case 2: optimization results

5.6 Discussions

In the multiobjective optimizaiton problem studied in Section 5.5 A, the number of particles of the RPPSO is set to be 1000 in order to get a better result, so the optimization process takes a longer time. In fact, RPPSO is able to find an acceptable solution in a short time. The

following simulation uses the same parameters as those in Section 5.5 A, except that the particle number is set to be 50. The termination criterion is set to be 10 iterations. The simulation finished in 490s on a laptop with Centrino 1.6GHz CPU and 1G RAM, running Matlab 7 in Windows XP environment.

The simulation results are shown in Figure 5-16, where gray dots give the values of particles and black stars are particles on the Pareto Front. The values of A, B, C, D, E (particles in cycles in Figure 5-16) are listed in Table 5-7. Compared with the results listed in Table 5-5, it is found that the results in Table 5-7 are not dominated by any particles in Table 5-5. This simulation proves that RPPSO is capable of finding acceptable solutions quickly.

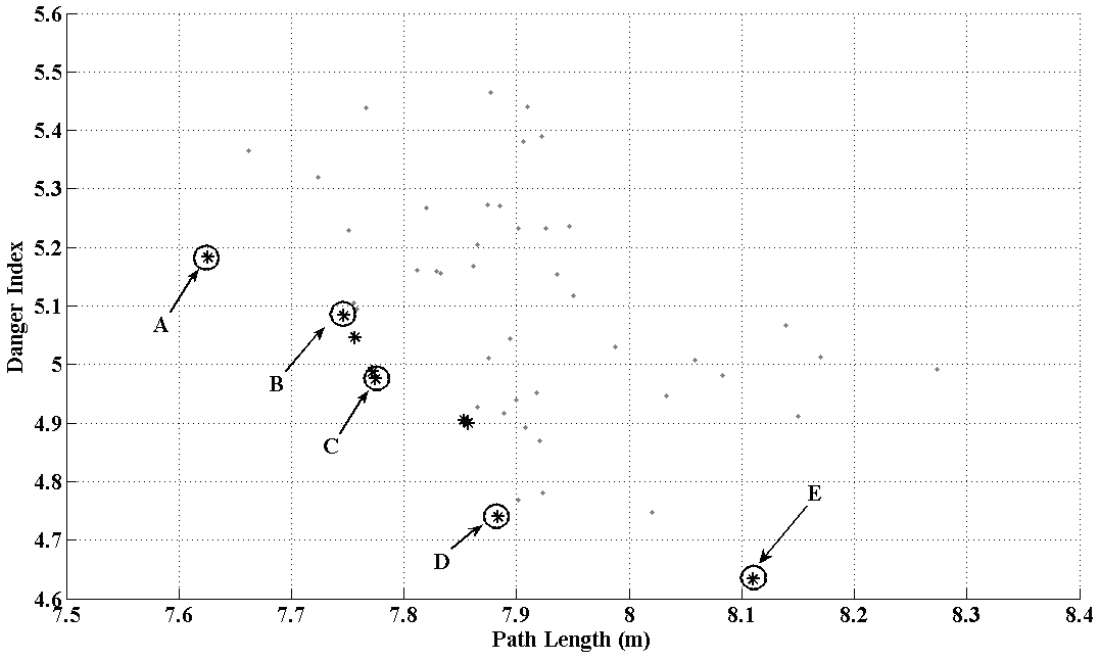


Figure 5-16 Multiobjective optimization Case 3: Pareto optimal set

Test	k	P	Q	C	ρ_0	Path Length (m)	Danger Index
A	5.1311	12.116	16.731	1.2304	0.4972	7.6258	5.1833
B	5.8008	14.31	15.902	1.2987	0.5234	7.7464	5.0838
C	6.1178	13.397	16.687	1.3077	0.5584	7.7745	4.9756
D	6.4227	14.462	15.894	1.2497	0.4859	7.8837	4.7398
E	6.3112	12.665	13.316	1.1228	0.4503	8.1098	4.6333

Table 5-7 Multiobjective optimization Case 3: optimization results

5.7 Conclusions

This chapter has investigated the optimization of parameters in the VSF^2 method. It has proved that the parameter setting in the F^2 method has a significant effect on the performance of the methods. For this reason, this chapter proposed a PSO-tuned F^2 method for single objective parameter optimization problems. A multiobjective optimization method – the Ranked Pareto Particle Swarm Optimization method (RPPSO) was then developed, and has been proved to be capable of solving multiobjective parameter optimization problems efficiently. Although the simulation studies were conducted for the VSF2 method in the research, the algorithms are generic and can be used for optimising parameters in other F^2 methods and for solving other optimization problems.

Chapter 6
Experimental Verification

In the previous chapters, a novel force field (F^2) for robot motion planning and coordination has been described in detail. Methods based on the concept of F^2 and their suitability for different applications has been presented. The feasibilities of these algorithms are demonstrated by simulation studies. This chapter examines the performance of these methods with robots in the Player/Stage in various environments. The methods to be tested in this chapter include the Canonical Force Field method (CF^2), Variable Speed Force Field method (VSF^2) and the Subgoal Guided Force Field method (SGF^2). Experiments with a Pioneer robot have been designed to prove the feasibility of the F^2 method on real robots in a real environment.

This chapter is organized as follows: the experiment setup is given in Section 6.1, which includes the introduction of the software platform, the Pioneer robot together with onboard laser sensor, and the description of software techniques used in the experiments. Experiments on single robot motion planning are carried out with a Pioneer robot in various environments in Section 6.2. Simulation studies with multiple robots in the Player/Stage platform are reported in Section 6.3. Conclusions are drawn in Section 6.4.

6.1 Experiment Setup

6.1.1 Software Platform

The Player Project (formerly the Player/Stage Project or Player/Stage/Gazebo Project), which is a project for the testing and development of robotics and sensor systems, has been adopted to develop software in experimental studies [141]. Its components include the *Player* network server and *Stage* and *Gazebo* robot platform simulators. The project was developed in 2000 by

Brian Gerkey, Richard Vaughan and Andrew Howard and is widely used in robotics research and education.

A. Player Robot Device Interface

The *Player* is a TCP/IP based network server for robot control. It supports a variety of robot hardware including the ActivMedia Pioneer robot [143] and SICK laser [178] which are used in our simulations and experiments. Player is designed to be language and platform independent, which means that the client program can be written in any language that supports TCP sockets, like C++, Java and Python, etc. In the simulations and experiments to be presented in this chapter, the control softwares are written in C++. Player is also designed to support any number of clients, which makes it suitable for complex applications such as multi-robot motion planning and coordination.

B. Stage Multiple Robot Simulator

The *Stage* has been designed to be capable of simulating a population of mobile robots moving in a two-dimensional bitmapped environment. Various sensor models are provided in this platform, including sonar, scanning laser rangefinder, pan-tilt-zoom camera with colour blob detection and odometry. Stage devices present a standard Player interface so that very few or no changes are required to move between simulation and hardware. That means that if control software has been tested successfully in the Player/Stage, it is almost ready to work on real robots [141].

C. Configuration File

A configuration file in Player defines all devices required and instantiates their drivers. A configuration file used in our experiments is given in Figure 6-1.

```
L01    #       Desc: Configuration for a Pioneer robot and a SICK laser.
L02    #       modified from the codes of B. P. Gerkey
L03    #       suit for player 2.0
L04
L05    # p2os driver
L06    driver
L07    (
L08      name "p2os"
L09      provides ["odometry:::position2d:0"]
L10      port "/dev/ttyUSB0"   # connection between robot and host
L11    )
L12
L13    # laser driver
L14    driver
L15    (
L16      name "sicklms200"
L17      provides ["laser:0"]
L18      port "/dev/ttyUSB1"   #connection between laser and host
L19      range_res "1" # 1mm precision, 8.192m max range
L20      resolution "50" # 0.5 degree increments, 361 readings @ 5Hz (38400) or 32Hz (500000)
L21    )
L22
L23    # map driver
L24    driver
L25    (
L26      name "mapfile"
L27      provides ["map:0"]
L28      filename "mapfiles/bld23201.png"   # provides the file name and location
L29      resolution 0.0147 #   length(m)/pixel
L30    )
L31
L32    #Localizer
L33    driver
L34    (
L35      name "amcl"
L36      update_thresh [0.05 0.175] # 10 degrees angle and 5 cm
L37      provides ["localize:0"]
L38      requires ["odometry:::position2d:0" "laser:0" "laser:::map:0"]
L39    )
```

Figure 6-1 A configuration file from the Player project

In this configuration file, lines L06 to L11 instantiate a *p2os* driver, which is essential to control a robot like a Pioneer robot. A host computer talks to the *po2s* driver via a USB port named *USB0*. L14 to L21 instantiates a *sicklms200* driver, which provides an interface (*laser:0*) to read data from a SICK LMS200 laser rangerfinder in L17. L18 specifies that this laser is connected to a host computer via a USB port (*USB1*). The range and resolution of the laser sensor are specified in L19 and L20. L24 to L30 configure an environment map to be used in experiments. L28 provides its name and location and L29 indicates that the map scale is *0.147m/pixel*. L33 to L30 define an *amcl* localizer which provides an interface (*localize:0*) and update its output every *0.05cm* or *0.175rad* (or *10* degrees approximately).

6.1.2 Pioneer Robot

A Pioneer robot used in the experiments is shown in Figure 6-2. A SICK LMS200 laser sensor is installed on the top of the robot. A laptop communicates with the robot via a USB port. Data from the laser sensor are transferred to the laptop through another USB port. More details of the Pioneer robot can be found at [143].

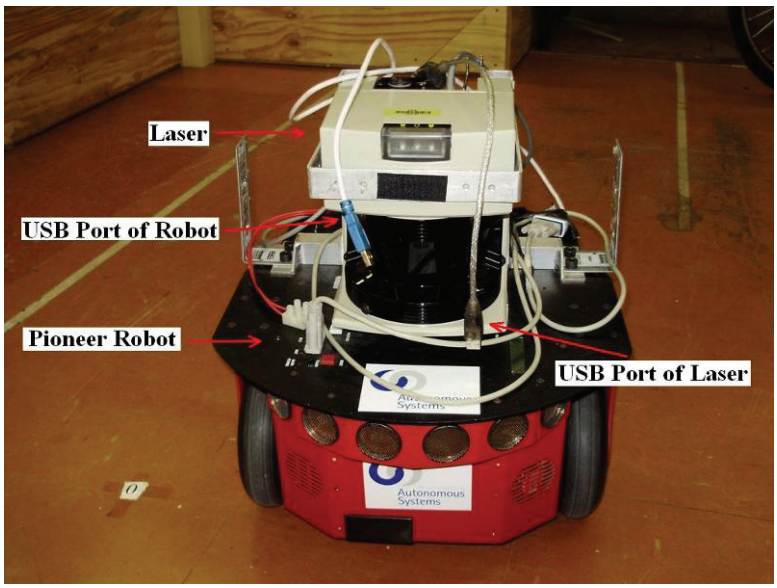


Figure 6-2 A Pioneer robot with a laser rangerfinder

6.1.3 Laser Sensor

The laser sensor used in these experiments is the SICK LMS200 [178]. This laser is installed on the top of a Pioneer robot, as shown in Figure 6-2. Given the configuration in Figure 6-1, the SICK laser in experiments will give a 180 degree view of $8.192m$ in front of it. The increment is 0.5 degree, so a single scan gives 361 readings. The laser data is useful in two aspects, i.e. obstacle detection and simultaneous localization, which will be discussed in the following sections.

6.1.4 Environmental Map

Experiments have been carried out in various indoor environments. A photo of a test environment is given in Figure 6-3. To create a map which can be used in the Player/Stage, the dimensions of all items in this area are measured and drawn in a bitmap, as shown in Figure 6-4. Note that environment data is to be acquired by a laser sensor installed on a Pioneer, so the map should only contain data which may be 'seen' by this laser, for example, the fire extinguisher (marked by A in Figure 6-3 and Figure 6-4), the table leg (marked by B in Figure 6-3 and Figure 6-4) and wooden wall (marked by C in Figure 6-3 and Figure 6-4). By contrast, the height of the table top in Figure 6-3 is higher than the laser. As a result, it cannot be found by the laser and 'disappears' in Figure 6-4. If a robot is placed in this environment (Figure 6-5), a laser reading is shown in Figure 6-6.

Figure 6-3 An experimental environment

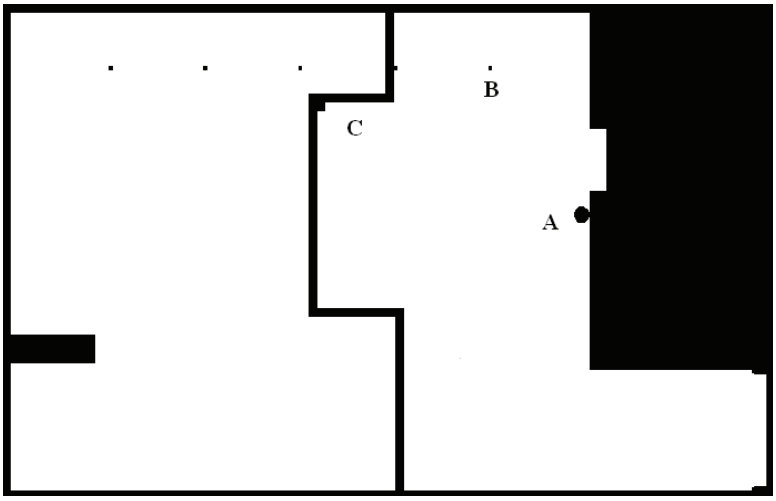


Figure 6-4 A bitmap used in Player/Stage

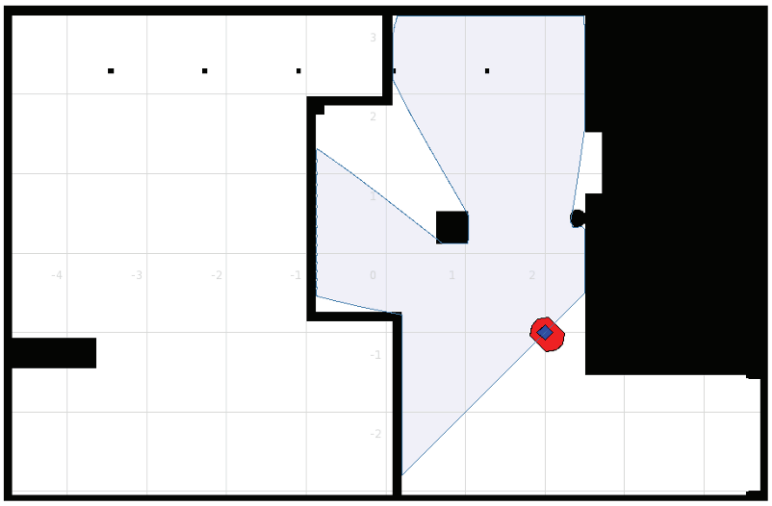


Figure 6-5 An experiment map

In the Player/Stage, a bitmap image file is read by a *mapfile* driver. The *mapfile* driver classifies each pixel of an image to one of the following three states: occupied (1), unknown (0), and free (-1). In short, 'darker' pixels are occupied, 'brighter' pixels are free, and those in between are unknown.

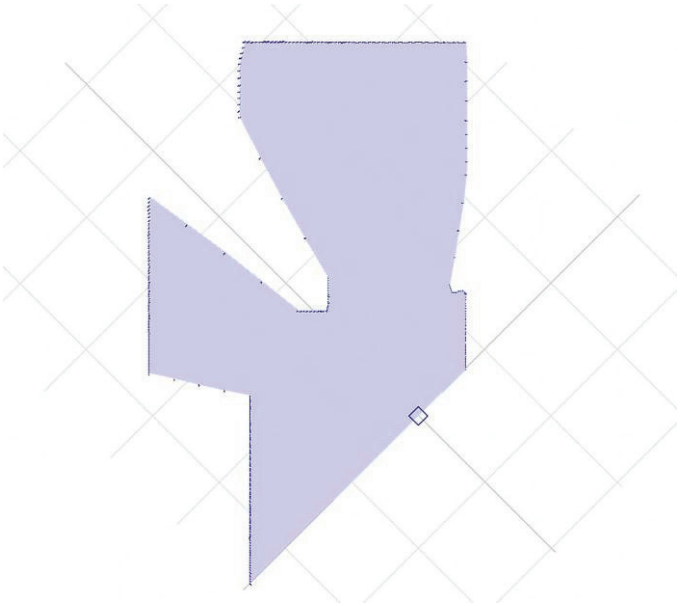


Figure 6-6 An example of laser reading

6.1.5 Localization Method

A robot needs to know its location in the environment. In experiments conducted in this chapter, this is achieved by applying an Adaptive Monte-Carlo Localization driver (*amcl*) [142]. At the conceptual level, the *amcl* driver maintains a probability distribution over a set of all possible robot poses, and updates this distribution using data from odometry, sonar and/or laser range-finders. A pre-defined map of the environment is needed to compare sensor readings with the map.

In the experiments, the *amcl* driver runs continuously and returns a set of current pose estimations and their possibilities. A pose estimation is considered 'good enough' and 'usable' if its possibility is larger than *0.9*. If no usable pose estimation is found during task execution, a robot will stop and wait until a better pose estimation is available. The update threshold is set to be $0.05m$ and $0.175rad$ which is 10 degrees approximately.

6.1.6 Obstacle Identification Approach

The obstacle identification process contains two steps: reading data from the laser sensor and identifying obstacles from the acquired data. If the distance between two points is larger that a preset threshold ε, these two points are considered to be from different entities. The setting of ε is described below:

First, ε must be larger than an arc length which is with current increment degree and max range to ensure obstacles at the laser's maximum range are correctly identified. In this case, $\varepsilon > \left(8.192 \times \frac{0.5}{180} \times \pi \right) \approx 0.0715m$

Second, ε must be larger than the diameter of the robot. Since a robot cannot pass an opening which is narrower that its diameter, a reasonable solution is to combine two obstacles into one obstacle if their distance are relatively small. A Pioneer robot's diameter is approximately *0.3m*, so $\varepsilon > 0.3m$.

Based on the analysis presented above, ε is set to be $0.3m$ in the following experiments. For the laser data shown in Figure 6-6, the result of obstacle identification is shown in Figure 6-8, in which five obstacles have been identified based on laser view. Their corresponding locations in the real environment are shown in Figure 6-7. It should be noted that the large obstacles around the robot are recognized as several small obstacles by this approach; for example, the wall on the right side of this robot is treated as two obstacles, which represents the collision possibilities from two directions in the F^2 method.

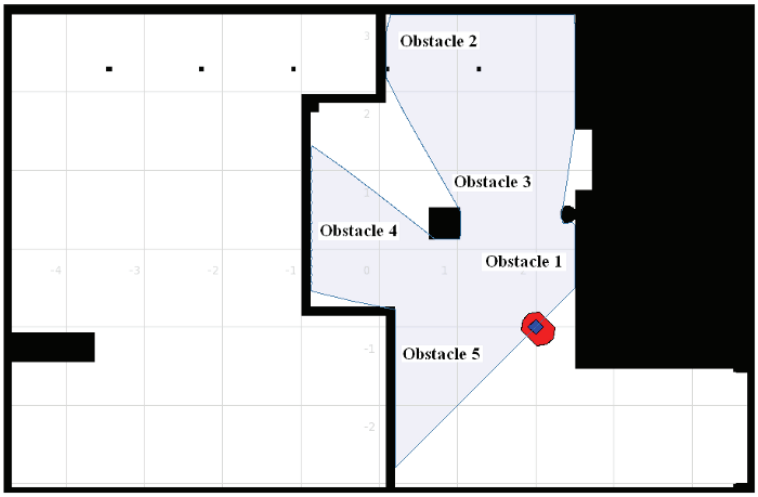


Figure 6-7 Obstacles identified

6.1.7 Curve Fitting Method

As described in the last section, the identification program has obtained the locations of some points on different obstacles. Since the geometric shape of an obstacle is needed for determining the repulsive force direction in the F^2 methods (see Chapter 3), the next step is to estimate the obstacle geometric shape based on these points. In all experiments conducted in this chapter, mobile robots travel in a flat surface, so the points on obstacles can be considered as points on a 2-dimensional curve. The Least Square Method is then utilized to find a set of coefficients for this curve. Regarding obstacles in Figure 6-8, the curve fitting result is given in Figure 6-9.

6.2 Experimental Studies on Single Robot Cases

This section introduces several experiments on cases where only one robot (single robot) is navigating within the environment. Several experiments are first carried out using the Canonical Force Field method (CF^2) to test its applicability in real robots. Experiments are designed to highlight the influence of parameters in the F^2 method. Then experiments with the

Variable Speed Force Field method (VSF^2) and the Subgoal-Guided Force Field Method (SGF^2) are described.

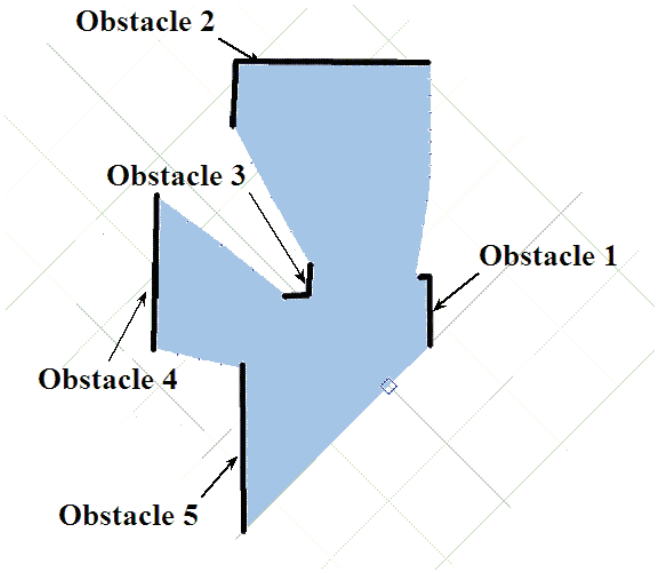


Figure 6-8 Obstacle identification

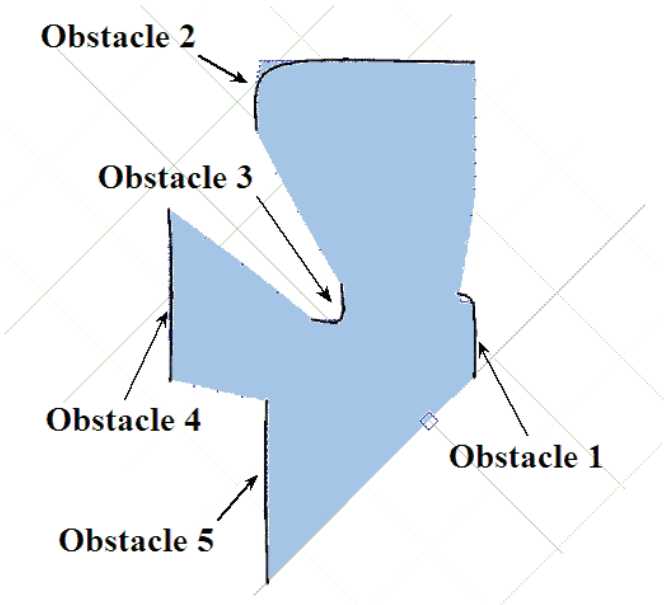


Figure 6-9 Curve fitting

6.2.1 Experimental Studies on Canonical Force Field Method

The Canonical Force Field method (CF^2) is tested in this section. Experiments are carried out using several groups of parameters for a navigation task. The results are analysed to highlight the effect of parameters on the performance of the method. Figure 6-10 shows a Pioneer robot, a laptop and a box (used as obstacle) in this experiment. The box is placed in this environment as shown in Figure 6-11. A point in this environment is set as the origin of coordinates, as shown in Figure 6-11 and Figure 6-12. The map used by this robot for localization purposes is shown in Figure 6-12. The Pioneer robot, denoted by a red patch in Figure 6-12, is placed at location $(2,-1)$ and is supposed to travel to $(0,1)$. The box size is $0.38m$ by $0.38m$. A robot is considered to be at its destination when the distance from its centre to its destination is less than its diameter, which is set to be *0.3m* in this case.

A. Canonical Force Field Method: Case 1

The parameters used in this experiment include: $k=5$, $C=1.5$, $v_{max}=0.05m/s$, $\rho_0=0.3$, *P=40*, *Q=20*, *v=0.03m/s*. From Equation (3-12), $E_r=v/(v_{max}C)=0.03/(0.05\times1.5)=0.4$. The resultant path is shown in Figure 6-13. This robot starts to move once it gets an estimation of its current location. When the robot is far from the obstacle and no obstacle is in the area of its force field, it goes straight to its goal. When it is close to the square obstacle and its force field covers the obstacle, the robot is repelled by the obstacle and steers away from it at $(1.1,-0.1)$.

Figure 6-10 CF^2 Case 1: setup

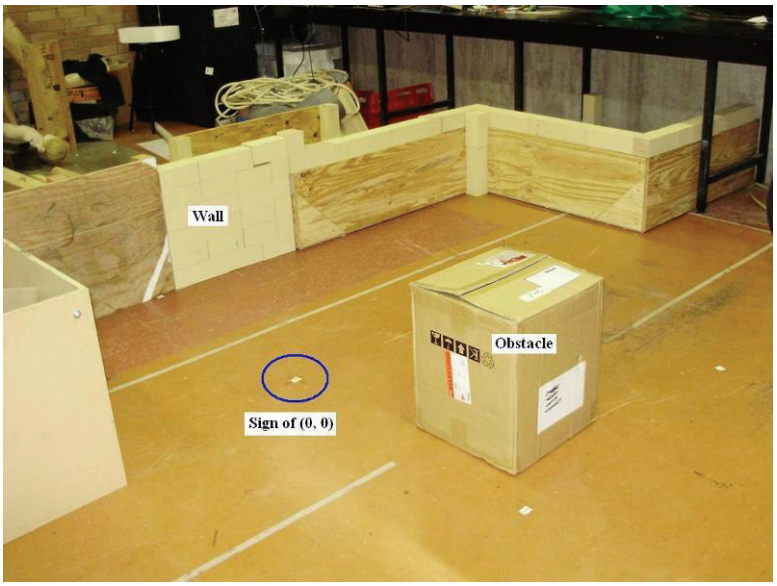


Figure 6-11 CF^2 Case 1: the environment used in the experiments

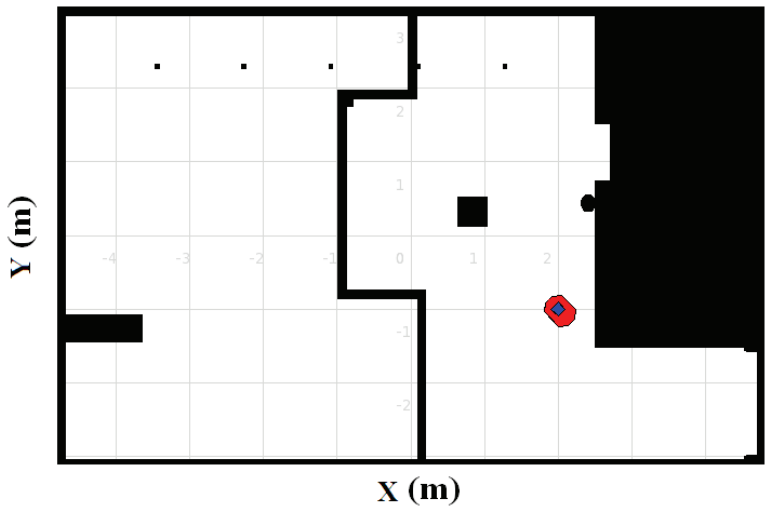


Figure 6-12 CF^2 Case 1: the map of the environment

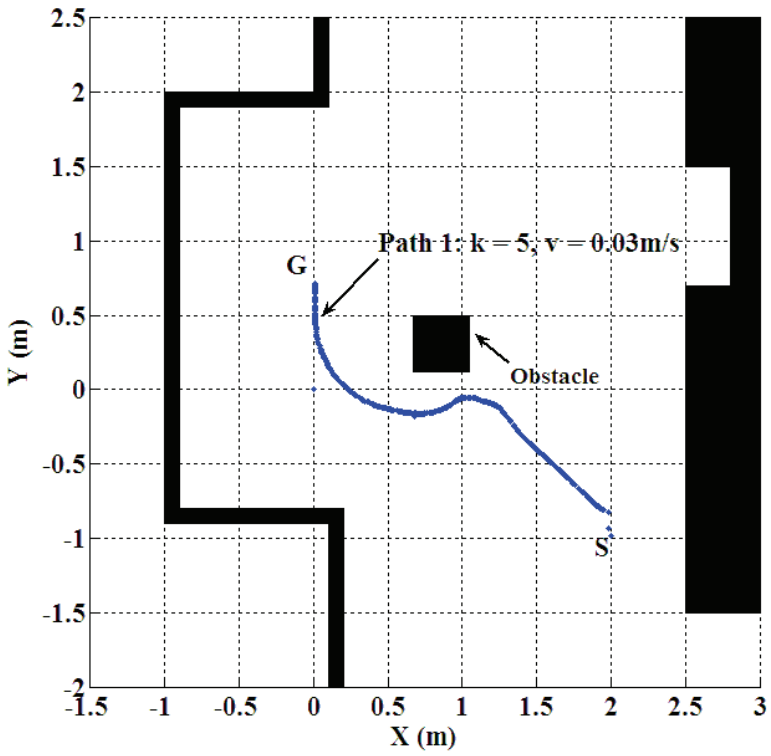


Figure 6-13 CF^2 Case 1: the path obtained by the CF^2 method

B. Canonical Force Field Method: Case 2

The purpose of this experiment is to study the effect of robot speed on the force field. The robot speed is increased to $v = 0.0375 m/s$. $E_r = v/(v_{max} C) = 0.0375/(0.05 \times 1.5) = 0.5$, which is larger than that of Case 1 ($E_r = 0.4$). The resultant path is shown in the red line (Path 2) in Figure 6-14 (lower curve). The path of case 1 (Path 1) is also plotted in Figure 6-14 (upper blue line) for comparison.

From Figure 6-14, it is observed that this robot avoids the square obstacle by taking a path further from the obstacle than in Case 1. The reason for this is that the coverage of this robot's force field is larger than that in Case 1 because of the higher speed, so this robot will suffer a repulsive force from the square obstacle when it is at a further distance from an obstacle than in Case 1.

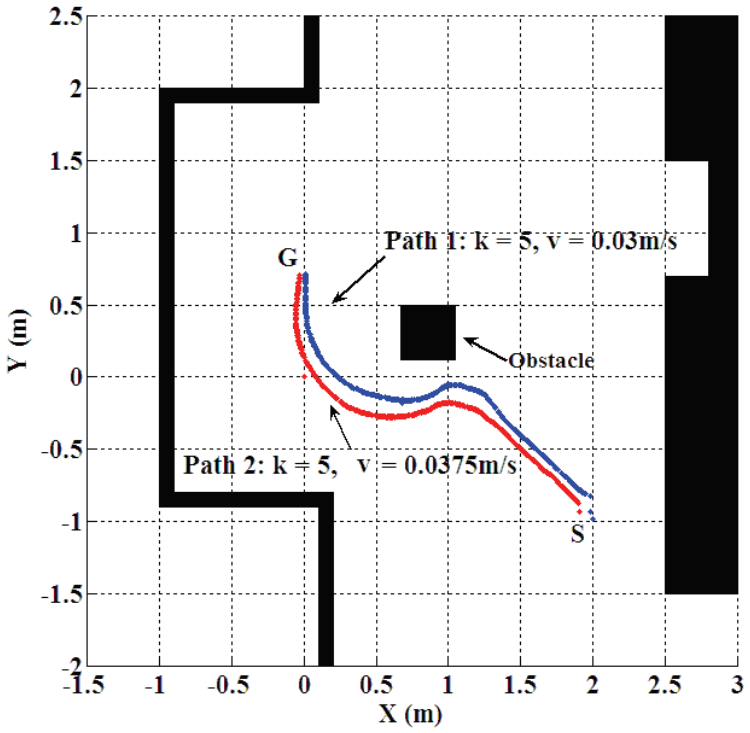


Figure 6-14 CF^2 Case 2: the path obtained by the CF^2 method

C. Canonical Force Field Method Case 3

The purpose of this experiment is to show the effect of factor k which is a scalar factor determining the coverage of force field. In this experiment, k is set to be 8 instead of 5 as in Case 1 and Case 2. Other parameters are the same as those used in Case 1, for example, $v_{max} = 0.05m/s$, $v = 0.03m/s$ and $C = 1.5$. The coverage of force field of this robot is enlarged since k is increased from *5* to *8*.

The resultant path is shown in the red line (Path 3) in Figure 6-15. The path obtained in Case 1 (Path 1) is also plotted in Figure 6-15 for comparison. It can be seen that this robot begins to avoid the square obstacle when it is further away than it does in Case 1 and then follows a larger curvature around the square obstacle. The reason is that the coverage of its force field is larger than that of Case 1. Unlike Case 2, the coverage of force field is enlarged by k instead of E_r as in Case 2.

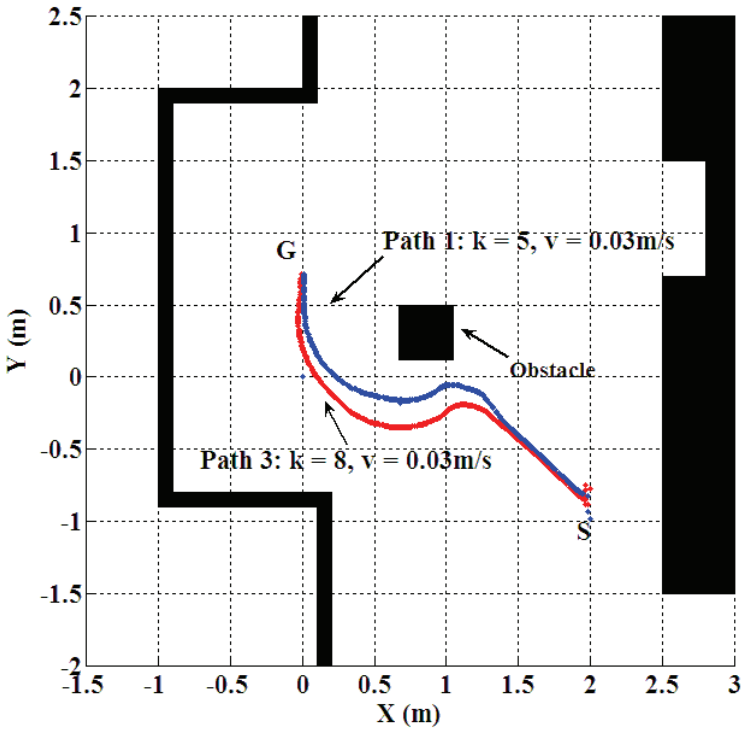


Figure 6-15 CF^2 Case 3: the path obtained by the CF^2 method

D. Conclusions on Experiments on Canonical Force Field Method

Experiments on the Canonical Force Field (CF^2) Method have been described in this section. The CF^2 method is proved to be applicable with real robots and capable of finding smooth trajectories for single robot motion planning and collision avoidance. Experiments were designed to test the influence of parameters on a robot's force field. It is shown that the setting of parameters in the CF^2 method significantly affect the performance of the method.

6.2.2 Experimental Studies on Variable Speed Force Field Method

An experiment carried out using the Variable Speed Force Field Method (VSF^2) is presented here. The environment used in this experiment is shown in Figure 6-16. A robot is supposed to travel through a narrow corridor-like environment. The map used by this robot in the Player project is shown in Figure 6-17. In this experiment, a robot is supposed to travel from $(0.8,0)$ to $(0.8,-2.5)$. Other parameters used in this experiment include $R_r = 0.3m$, $k = 5$, $C = 2$, $v_{max} = 0.03m/s$, $P = 40$, $Q = 40$, $\rho_0 = 0.2$. The resultant path is shown in Figure 6-18. The robot's orientation, linear speed and angular speed are given in Figure 6-19, Figure 6-20 and Figure 6-21 respectively.

The robot passes through this narrow corridor and arrives at its destination successfully. Since the range of the laser sensor is big, the robot is acted upon by repulsive forces from the start point all the way to the destination. As the result of the repulsive forces from Obstacle 1 and Obstacle 2, the robot travels in the middle of the two obstacles and its speed fluctuates around $0.02m/s$. In this way, the repulsive forces from the two sides are nearly balanced. Because of the slippage between the robot wheels and the floor, there are some fluctuations in the robot's linear speed and angular speed. In general, the VSF^2 approach is proved to be applicable in robot navigation.

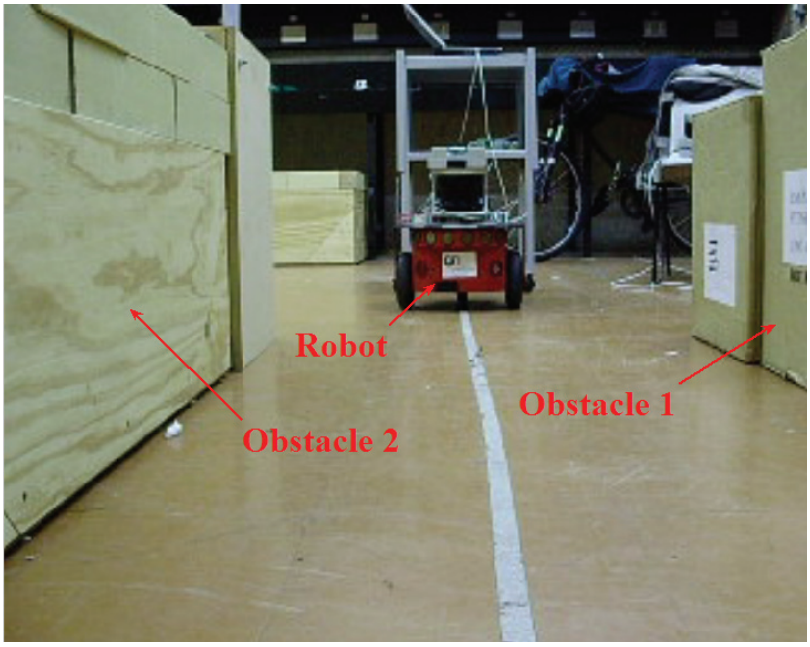


Figure 6-16 VSF^2 Case 1: the environment

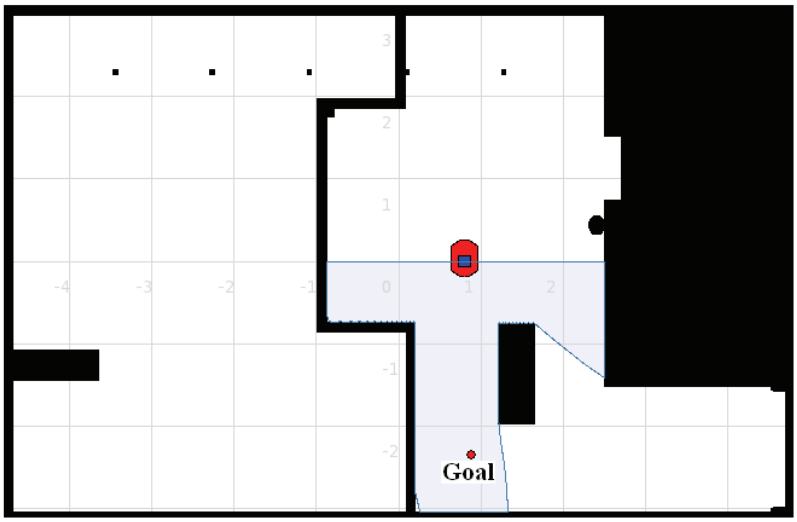


Figure 6-17 VSF^2 Case 1: the map of the environment

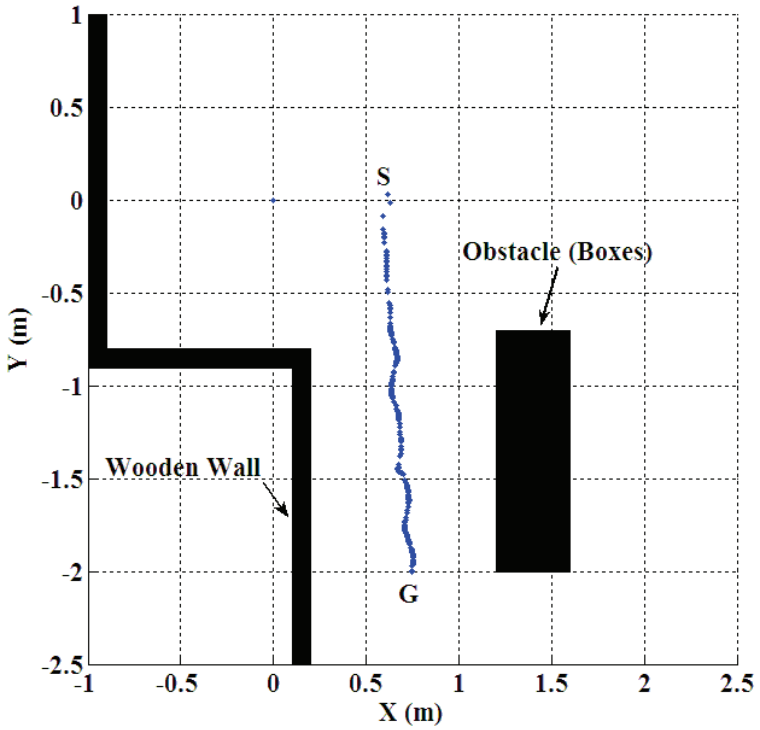


Figure 6-18 VSF^2 Case 1: the path obtained

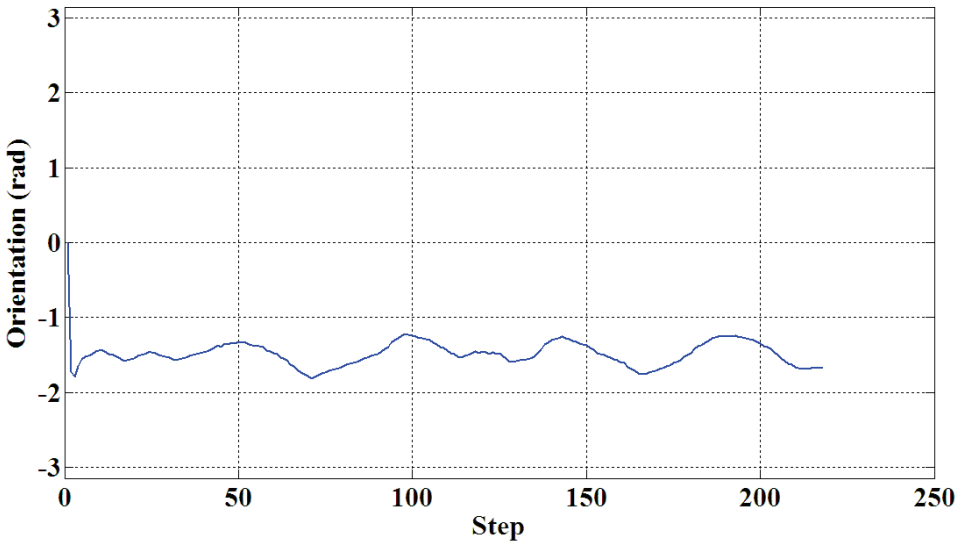


Figure 6-19 VSF^2 Case 1: variation of the robot orientation

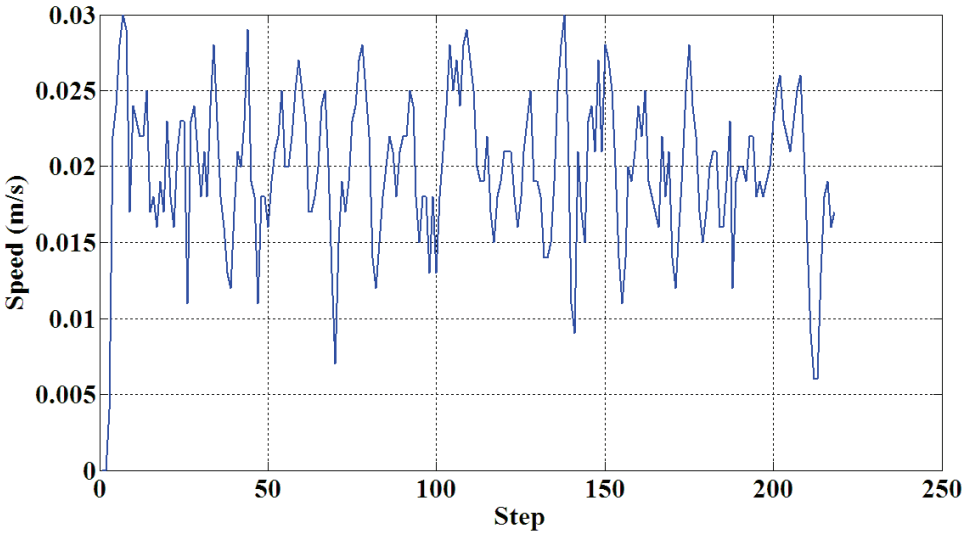


Figure 6-20 VSF^2 Case 1: the changes of the robot's linear speed with time

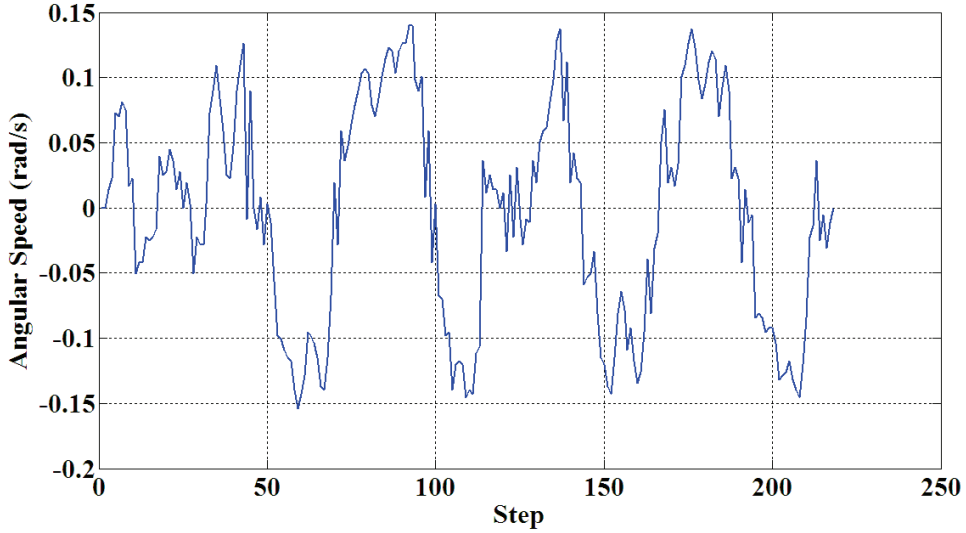


Figure 6-21 VSF^2 Case 1: the variation of the robot's angular speed

6.2.3 Experimental Studies on the Subgoal-Guided Force Field Method

An experiment carried out using the Subgoal-Guided Force Field Method (SGF^2) is introduced here. A map is given in Figure 6-22. Two obstacles are placed in this environment, forming a corridor-like passage.

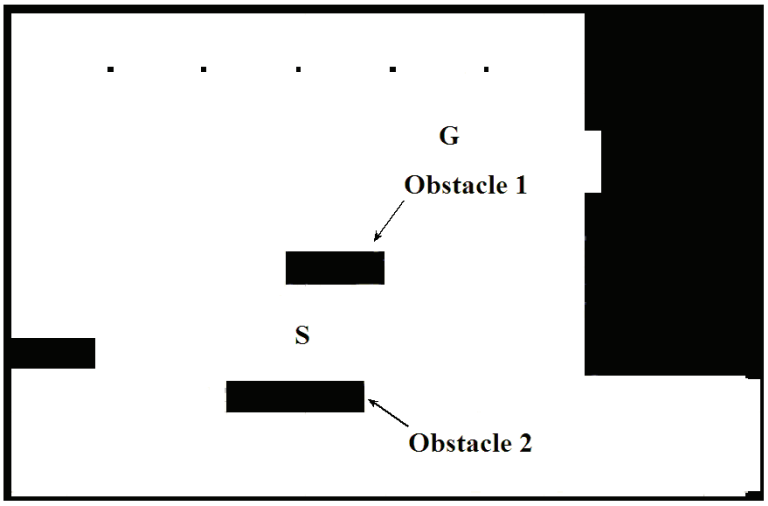


Figure 6-22 SGF^2 Case 1: the map of the environment

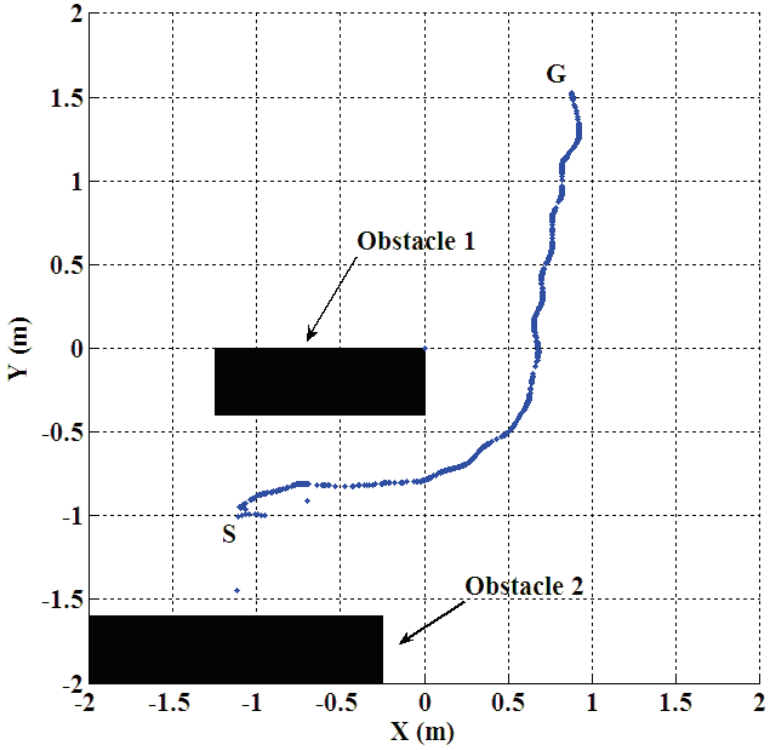


Figure 6-23 SGF^2 Case 1: the path obtained

In this experiment, the robot is supposed to travel from $S(-1,-1)$ to $G(1,1.5)$. Parameters used in this experiment are $R_r = 0.3m$, $k = 5$, $C = 2$, $v_{max} = 0.03m/s$, $P = 60$, $Q = 20$,

$\rho_0 = 0.2$. The resultant path is shown in Figure 6-23. In this experiment, the robot successfully arrives at its destination. Its movement in the corridor area is quite smooth and no severe oscillations occur. This experiment proves that the SGF^2 is applicable for real robot applications.

6.2.4 Conclusions on Single Robot Experiments

Experiments on single robot motion planning and collision avoidance have been carried out in indoor environments. Firstly, experiments have been carried out using the Canonical Force Field Method (CF^2) with different sets of parameters. The results are compared to highlight the effect of the parameter settings on the CF^2 method. Experiments were also designed to prove the feasibility of the Variable Speed Force Field method (VSF^2) and the Subgoal-Guided Force Field method (SGF^2) on real robots. It has shown that these methods are suitable for robot real-time motion planning and collision avoidance.

6.3 Experimental Studies on Multi-robot Coordination

Simulations carried out in the Player are presented here. Pioneer Robot parameters are the same as those stated in the previous experimental studies. The robots use *amcl* driver to estimate their locations and plan their motions using the VSF^2 method. Each robot is assumed to have the information of other robots, including size, speed, task priority, etc.

6.3.1 Two-Robot Cases

Experiments with two robots are conducted. Robot 1 (red) is supposed to travel from $S_1(-1,1.5)$ to $G_1(-1,-2)$ and Robot 2 (blue) travels from $S_2(-0.5,-2)$ to $G_2(-0.5,1.5)$. Three sets of force field method parameters are tested and the results are compared. The effect of task priority (T_p) and environment factor (C) on multi-robot motion coordination using the VSF^2 method is studied.

A. Two-Robot Coordination: Case 1

The VSF[2] method parameters used in this experiment are $R_r = 0.3m$, $k = 3$, $C = 2$, $v_{max} = 0.03m/s$, $P = 40$, $Q = 30$, $\rho_0 = 0.2$ for every robot. The task priorities of robots are set to $T_p = 1$, i.e. all robots have same priority. A general view of the robots' paths is shown in Figure 6-24. More details on the simulation progress are shown in Figure 6-25.

Before a robot starts, the *amcl* driver is called to get the estimation of its current location, as shown in Figure 6-25 (a). Once a believable estimation is obtained, the robot will start to move, as in (b). When the robots arrive at the positions in (c), their force fields overlap and both robots are acted on by the repulsive forces from each other. Their moving directions are therefore changed a little. The robots move closer in (d) and avoid collision with each other in (e). They move on to their destinations in (f), (g) and (h).

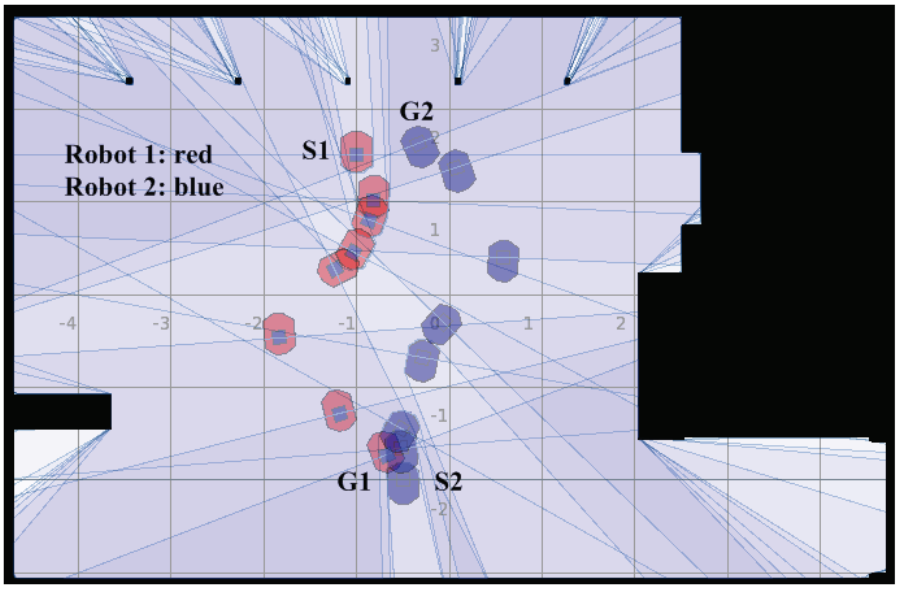


Figure 6-24 Two-robot coordination: paths of Case 1

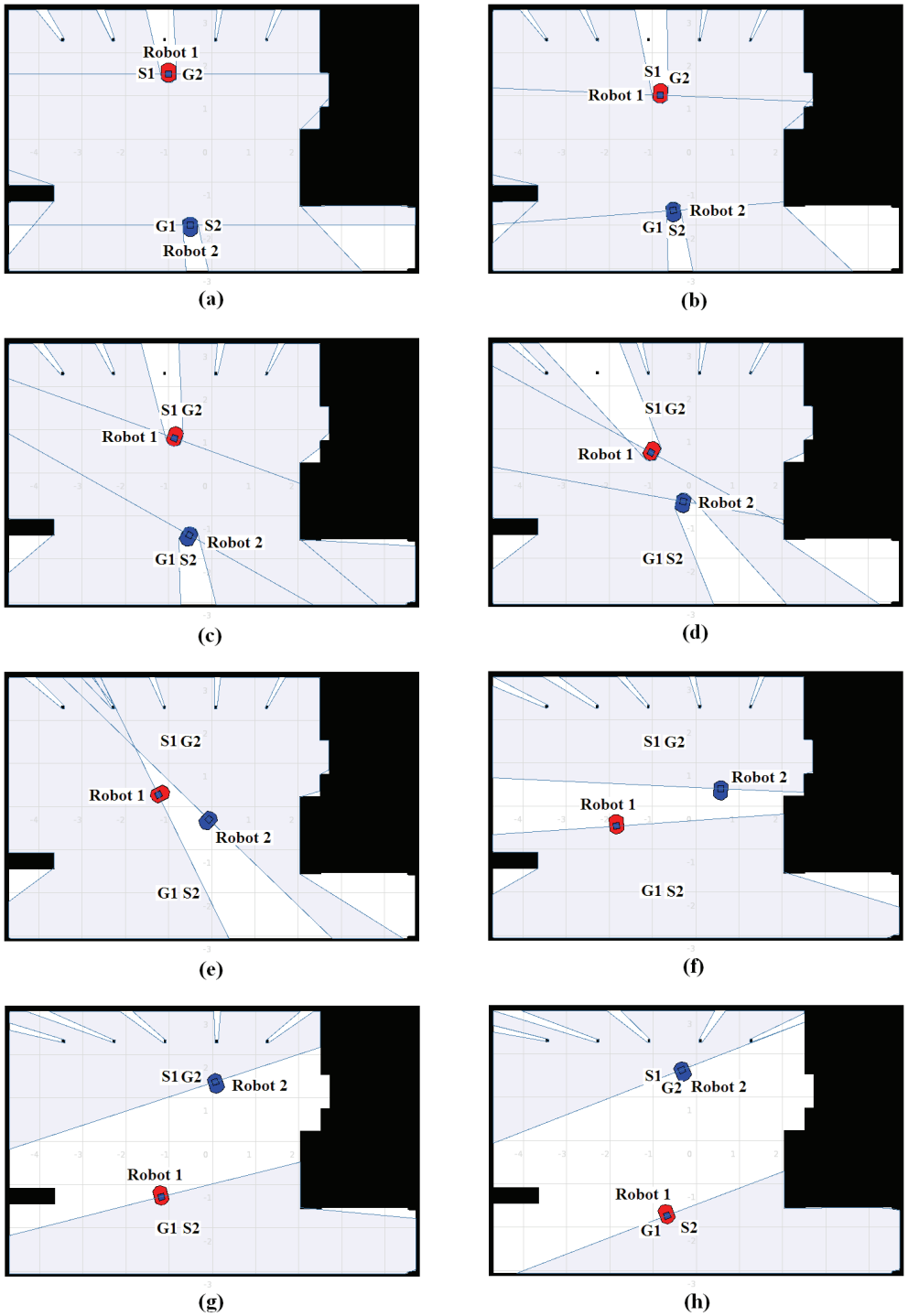


Figure 6-25 Two-robot coordination: Case 1

B. Two-Robot Coordination: Case 2

This simulation is carried out to study how the environment factor C affects multi-robot coordination using the VSF^2 method. Parameters used in this simulation are the same as those of Case 1, except that the environment factor is set to be $C = 3$, which is larger than the value used in Case 1. The robots' paths are shown in Figure 6-26. More details of the simulation can be found in Figure 6-27.

In Figure 6-27, (a) and (b) are almost the same as those of Figure 6-25. In (c), robots receive repulsive forces from each other and begin to change their moving directions. Note that the distance between the two robots at this time is smaller than that of Figure 6-25 (c). The reason is that the coverage of their force fields is smaller than the force field in Case 1 because of a larger C. In (d) and (e), the robots take smaller curvatures to avoid collision with each other. They then move towards their destinations in (f), (k) and (h). Comparing robots' paths in Figure 6-24 and Figure 6-26, it is shown that the environment factor c significantly affects the performance of the VSF^2 method.

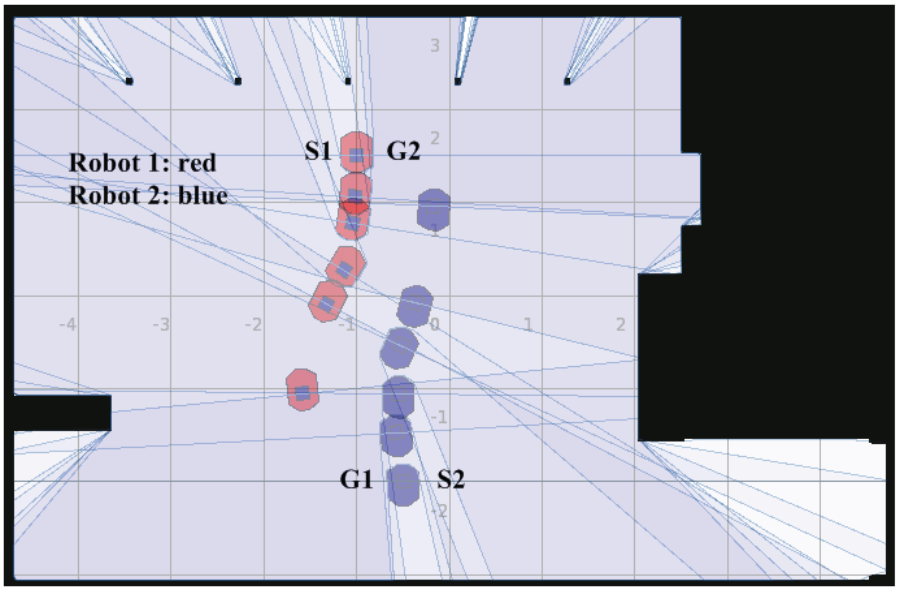


Figure 6-26 Two-robot coordination: paths of Case 2

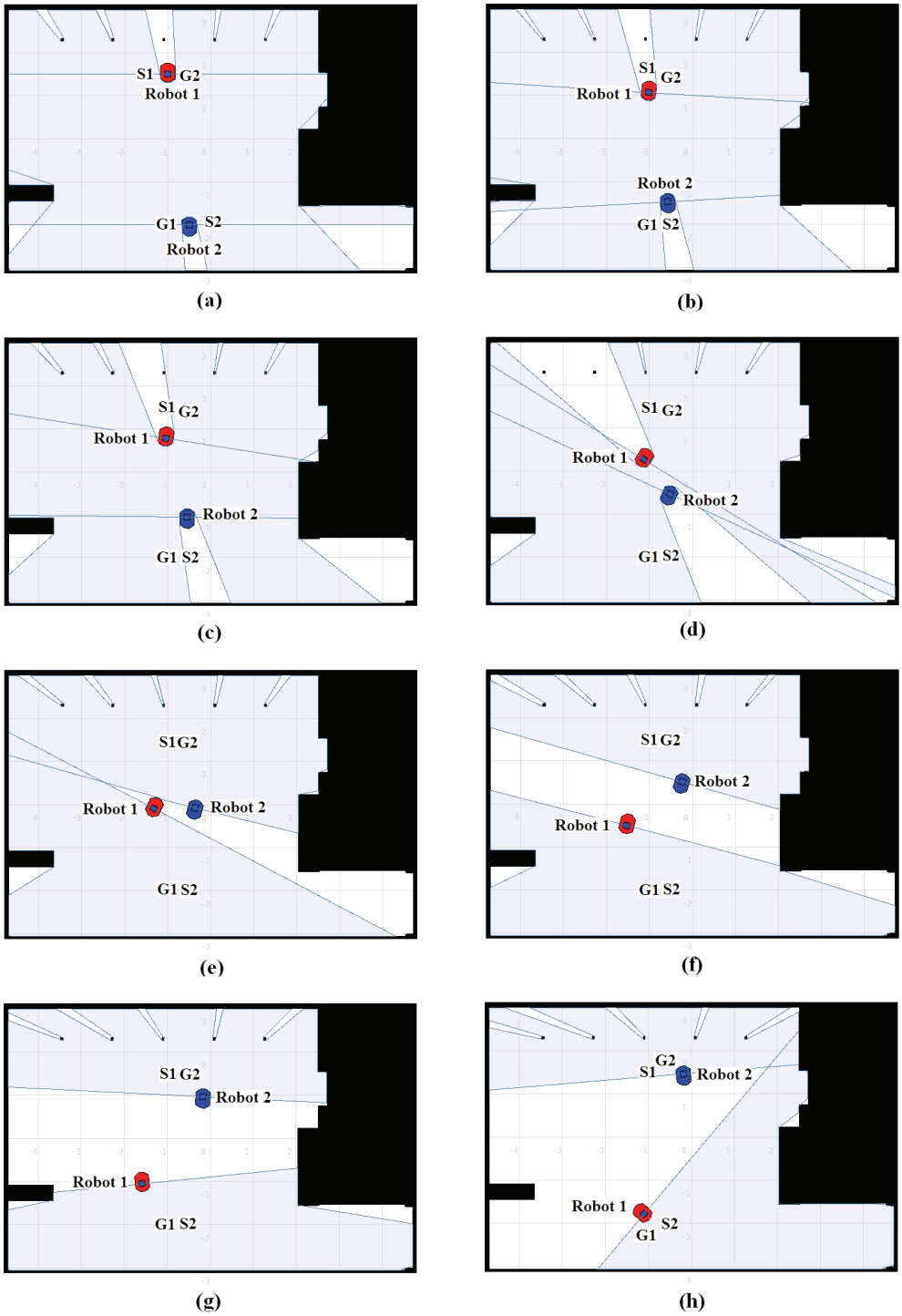


Figure 6-27 Two-robot coordination: Case 2

From Equation (3-10), the E_r decreases if a larger C is chosen. As a result, the coverage of its force field will become smaller. When a robot travels in an obstacle-cluttered area, a larger C may be chosen. On the other hand, a smaller C may be chosen when the robot travels in a free space.

C. Two-Robot Coordination: Case 3

This simulation highlights the effect of task priority T_p on multi-robot coordination. Case 3 is carried out using the same robots and the VSF[2] parameters as those in Case 1, except that Robot 1's task priority is set to be 1 and the task priority of Robot 2 is set to be 2. Therefore, Robot 2 has higher priority than Robot 1 in this case. The robots' paths are shown in Figure 6-28. Some simulation snapshots are shown in Figure 6-29.

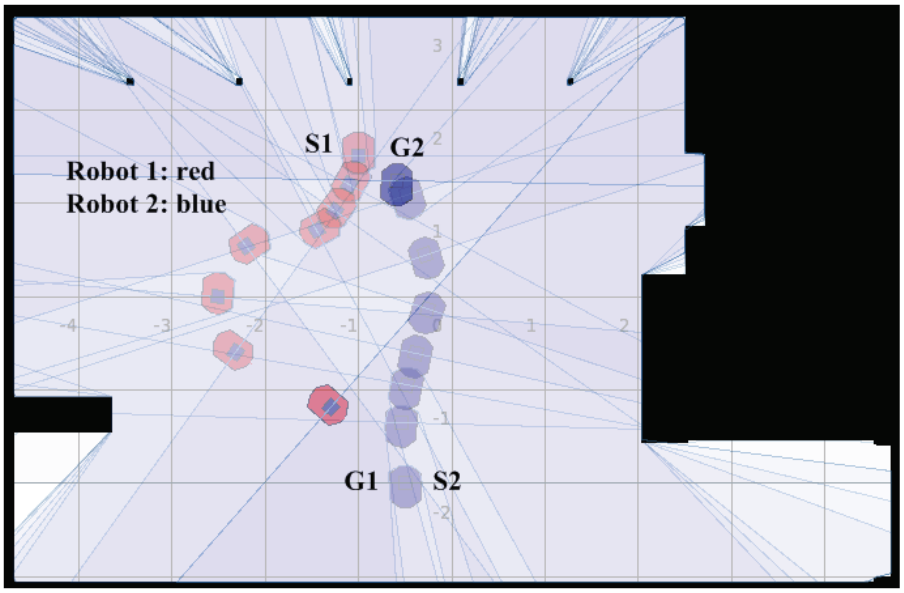


Figure 6-28 Two-robot coordination: paths of Case 3

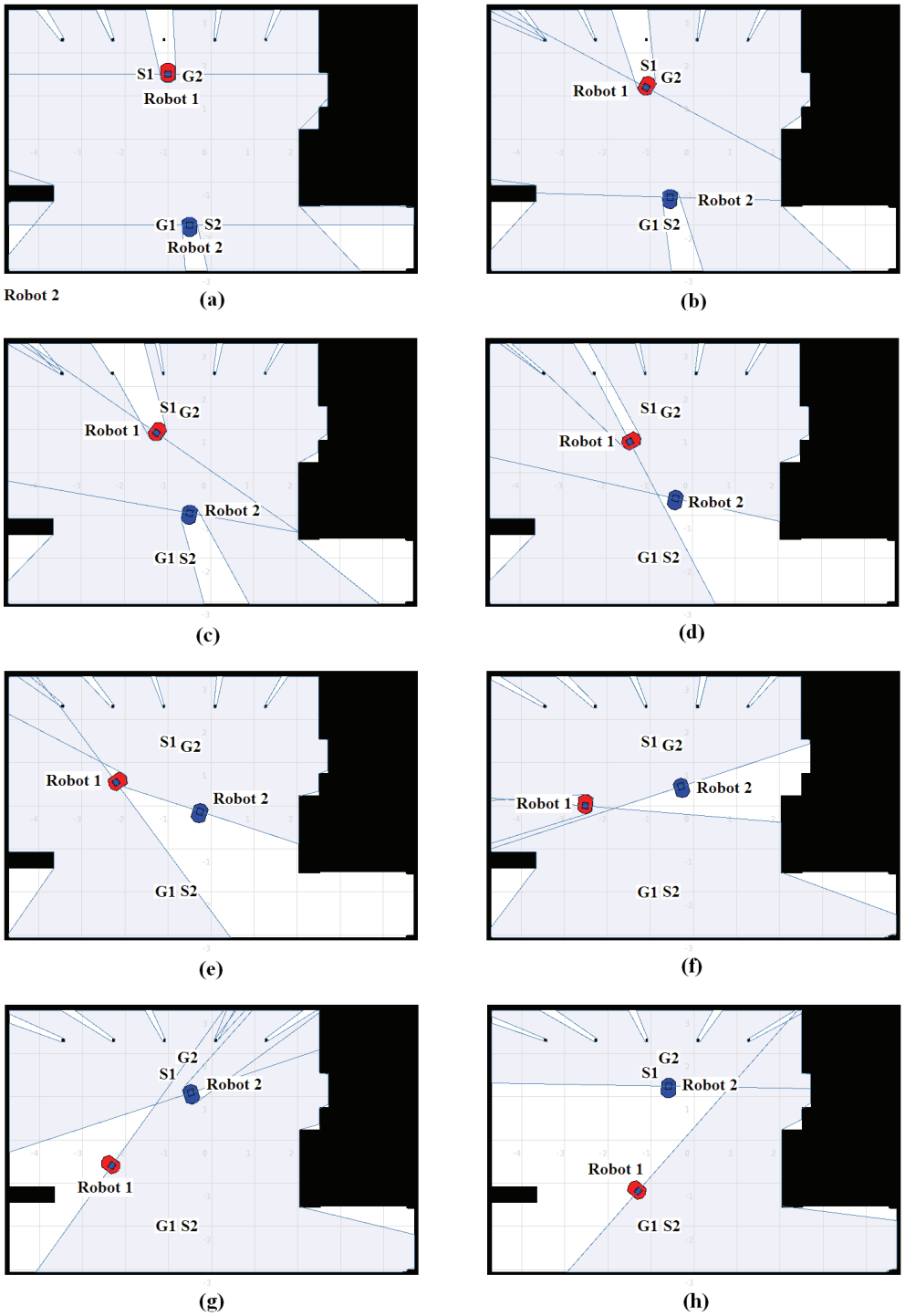


Figure 6-29 Two-robot coordination: Case 3

In Figure 6-29, Robot 1 starts to change its moving direction in (b), but Robot 2 does not. This is because by Equation (3-3), the coverage of Robot 2's force field is enlarged by a higher task priority so Robot 2 has the advantage in collision avoidance. From (c), (d), (e) and (f), the moving direction of Robot 2 is changed slightly. Robot 1 follows a path with larger curvature to avoid collision with Robot 2. Comparing robots' paths in Figure 6-24 and Figure 6-28, it is shown that in the F^2 method a robot will have priority in collision avoidance when a higher task priority is assigned.

The task priority is important in real applications. By assigning appropriate priorities to robots, a task with higher priority can be done without much effect from other robots. The F^2 method integrates the task priority into the construction of the force field, which makes it a suitable and natural approach for motion planning and collision avoidance in multi-robot coordination and control.

6.3.2 Three-Robot Coordination

A simulation study carried out with three robots is presented here to test the performance of the F^2 method in a multi-robot scenario. Each robot plans its way independently to avoid potential collisions with other robots simultaneously and makes it way to the goal. Parameters for all robots in this experiment are $R_r = 0.3m$, $k = 3$, $C = 2$, $v_{max} = 0.03m/s$, $P = 40$, $Q = 30$, $\rho_0 = 0.2$, $T_p = 1$. Robot 1 (red) is supposed to travel from $S_1(-1,1.5)$ to $G_1(-0.2,-2)$. Robot 2 (blue) travels from $S_2(1,-1)$ to $G_2(-2,1)$. Robot 3 (green) travels from $S_3(-4,0)$ to $G_3(1,1.5)$. Every robot in this simulation is aware of the other robots' status including their speeds, positions, and so on. The simulation snapshots are shown in Figure 6-30, Figure 6-31 and Figure 6-32 in time sequence. For a better view of robots' paths in the whole process, their movements in Figure 6-30, Figure 6-31 and Figure 6-32 are shown in subfigures (a), (b) and (c) of Figure 6-33, respectively.

In Figure 6-30, the robots' initial positions are shown in (a). The *amcl* driver is called to get estimations of their current locations and the robots start to move as shown in (b). As they

move closer to each other, they are acted on by repulsive forces from other robots. In (c), all robots decelerate and the moving direction of Robot 2 (blue) changes noticeably. In (d), (e) and (f), Robots 2 and 3 are forced to travel at very slow speeds to give way to Robot 1.

In Figure 6-31 (g), Robot 1 (red) has passed through the central area. Robots 2 (blue) and 3 (green) are moving in opposite directions. In (h) and (i), Robot 3 changes its moving direction to give way to Robot 2. Robot 2 moves on in (j), (k) and (l). The moving direction of Robot 1 is determined by the resultant force of the repulsive force from Robot 3 and the attractive force from its own destination.

In Figure 6-32, Robot 1 arrives at its destination. Robot 2 moves towards its goal in (m), (n) and stops in (o). Robot 3 avoids possible collision with Robot 2 in (o), (p) and makes it way to its goal in (q) and (r).

This simulation consists of complex collision avoidance processes. Figure 6-30 shows the collision avoidance among three moving robots in (b) to (f). Figure 6-31 shows the collision avoidances between two moving robots, e.g., Robot 2 and Robot 3 in (h). Figure 6-32 (o) and (p) shows the collision avoidance between a moving robot (Robot 3) and a stationary obstacle (Robot 1 stopping at its goal). This simulation proves that the F^2 method is suitable for multi-robot real-time motion planning and coordination.

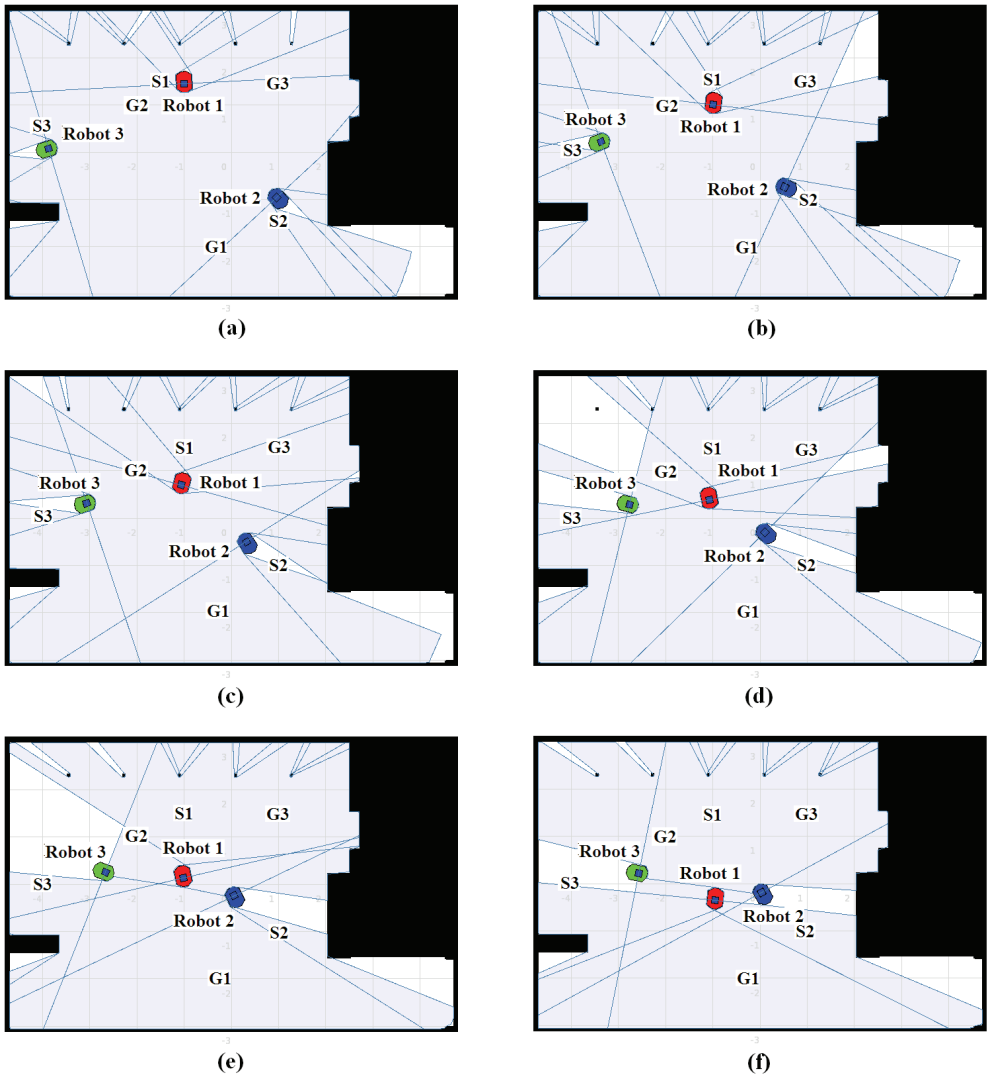


Figure 6-30 Three-robot coordination: part 1

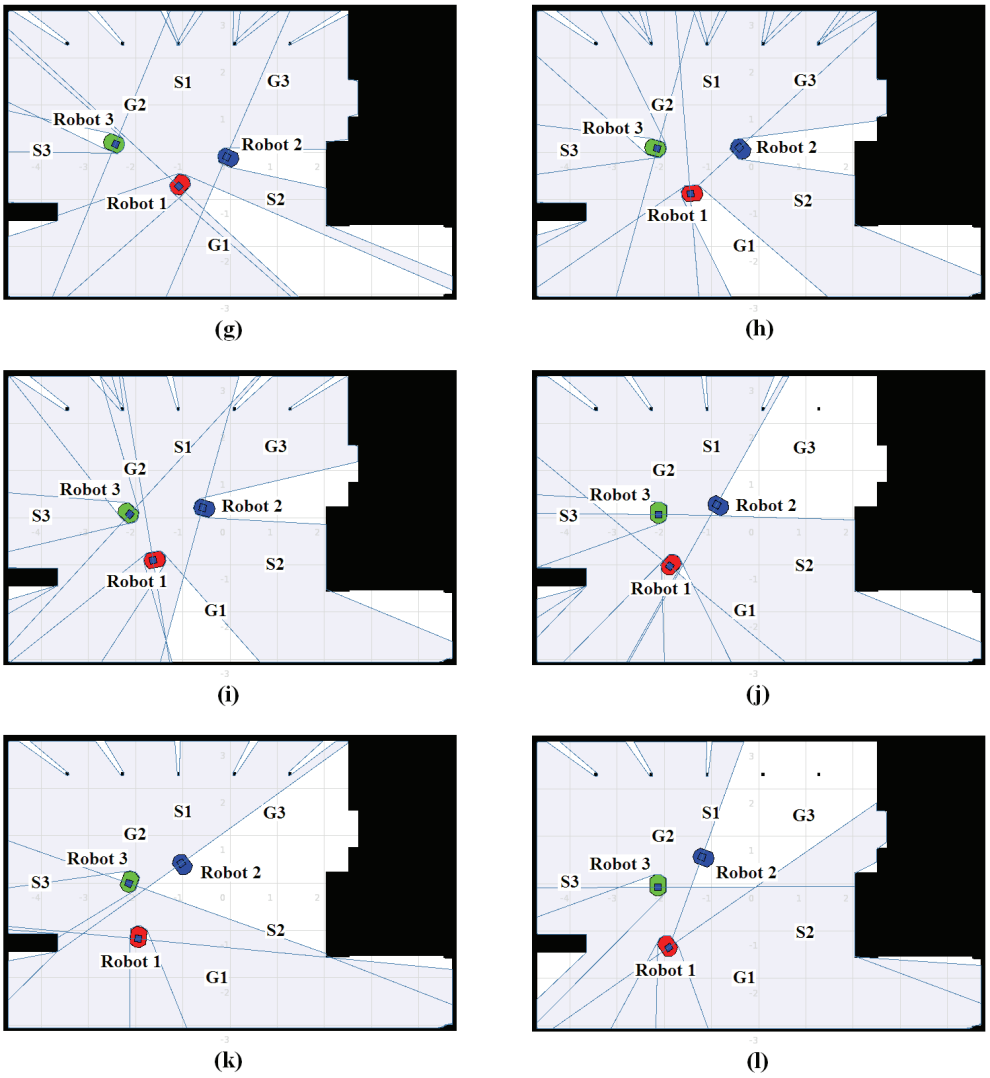


Figure 6-31 Three-robot coordination: part 2

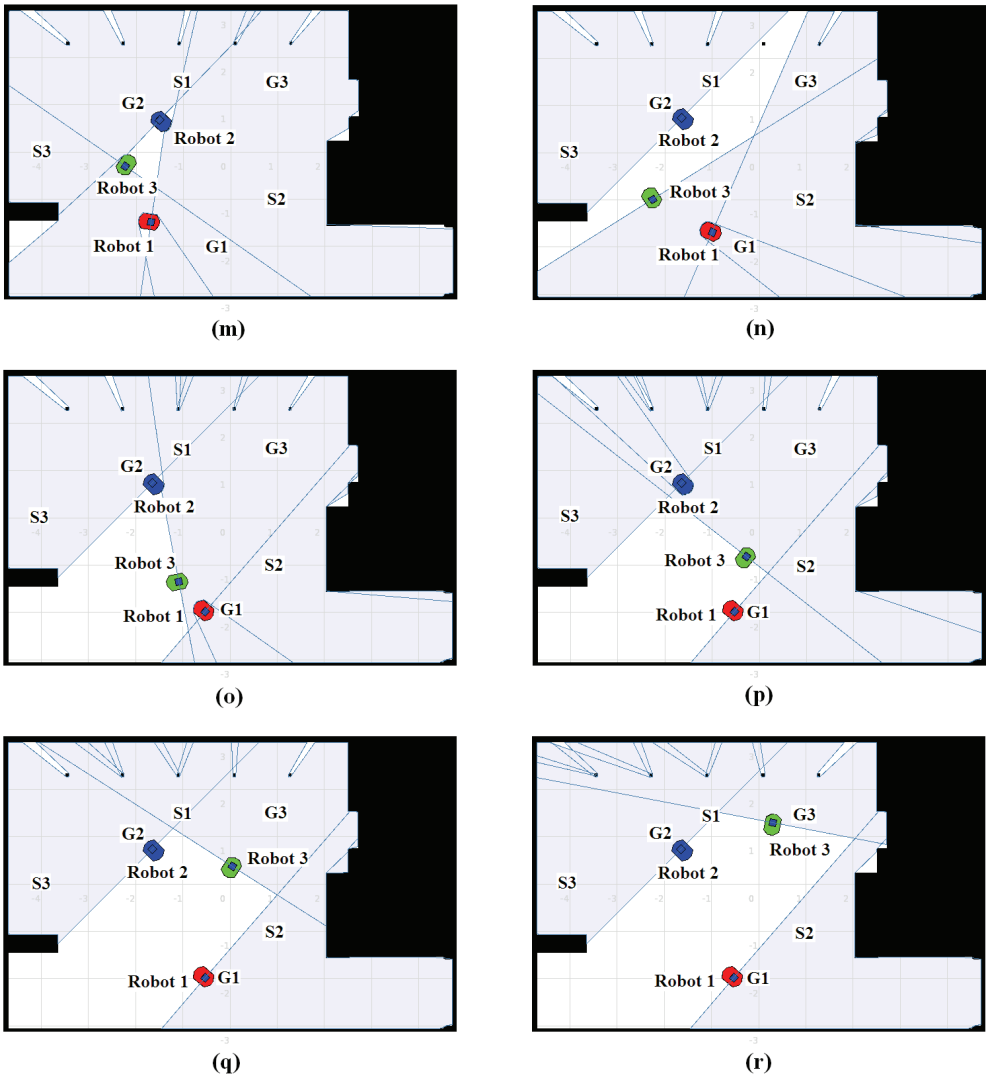
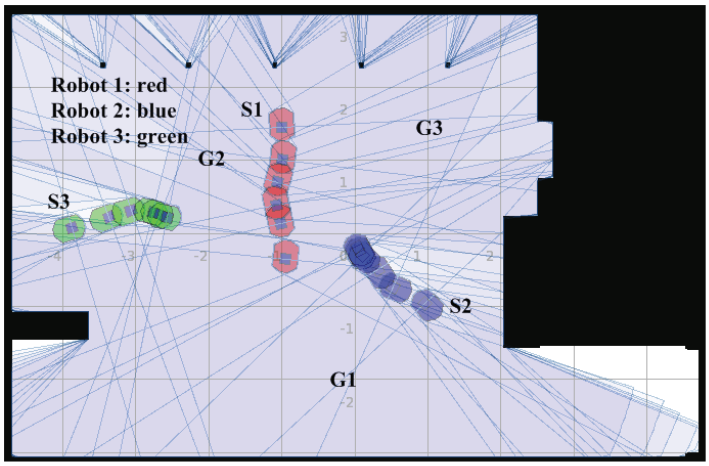
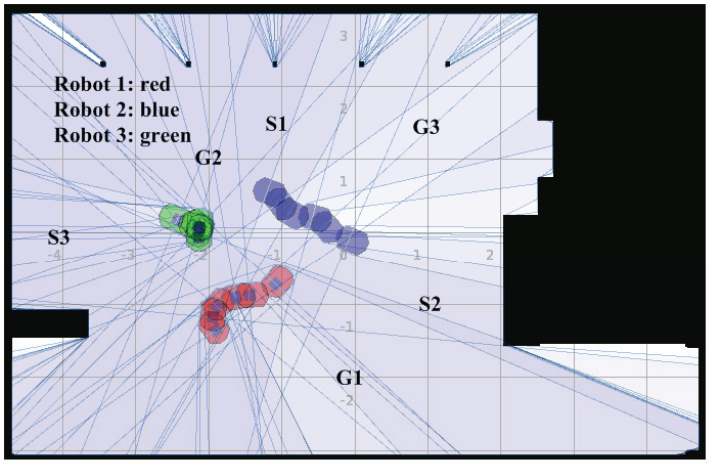


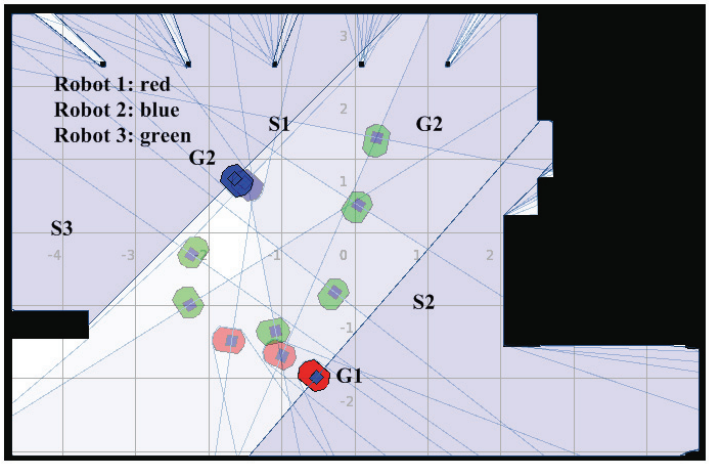
Figure 6-32 Three-robot coordination: part 3



(a) Paths of Figure 6-30



(b) Paths of Figure 6-31



(c) Paths of Figure 6-32

Figure 6-33 Three-robot coordination: a general view

6.4 Conclusions

In this chapter, experimental studies on the F^2 method have been presented in detail. Firstly, the Player project, which is the research software platform used in the experiments, was introduced in Section 6.1. Then the hardware, including robot and laser sensor, and software techniques, including localization, obstacle identification and curve fitting, were described.

Experiments on single robot motion planning and collision avoidance were carried out using a Pioneer robot equipped with a laser rangerfinder in an indoor environment. Experiments using several sets of parameters were also conducted to highlight the different effects of parameters on the F^2 methods. Algorithms tested experimentally were the Canonical Force Field method (CF^2), the Variable Speed Force Field method (VSF^2) and the Subgoal-Guided Force Field method (SGF^2). The feasibility of these methods is fully demonstrated by the experiments.

The experimental studies with multiple robots were carried out in the Player/Stage platform in Section 6.3. The results are shown with snapshots of simulation processes. Three simulations on two-robot navigation in an indoor environment are described in detail. The simulation results demonstrate the effect of task priority (T_p) and environment factor (C) on motion planning and coordination using VSF^2 method. A simulation with three robots was then presented and verified that the VSF^2 method can be applied in multi-robot motion planning and coordination. Although the VSF^2 method is the only one of the four discussed that is studied experimentally in this section, yhe other methods, CF^2, SGF^2 and $DVSF^{2,}$ can be experimentally tested in the same way, and we believe they are able to coordinate the motion of a team of robots equally well because they are based on the same concept of force field.

Chapter 7
Conclusions and Future Work

In this thesis, a novel concept of force field (F^2) has been developed and presented in detail. Based on this concept, four F^2 based methods (or 'F^2 methods') have been proposed and investigated: the Canonical Force Field method (CF^2), Variable Speed Force Field method (VSF^2), Subgoal-Guided Force Field method (SGF^2) and Dynamic Variable Speed Force Field method ($DVSF^2$). Unlike other existing approaches such as potential field methods, the F^2 methods generate a virtual force field for each robot based on its own status, including position and orientation in the work space, size, travelling speed, task priority with respect to other robots and environmental factors. This force field locates in the robot's vicinity and varies with its status during the robot movement. If there are obstacles or other robots existing in a robot's force field, the robot will be acted on by repulsive forces and steered away to avoid possible collisions. If a robot has a larger size, higher speed or higher priority, its force field will cover a greater area than other robots and it will have the advantage during collision avoidance.

The 'walking towards a goal' behaviour of a robot using the F^2 methods is achieved by an attractive force which directs a robot to a goal point. This goal point can be a robot's global destination (as in the CF^2 method and VSF^2 method), a temporary subgoal which is generated continuously using updated sensor data (as in the SGF^2 method) or a temporary point which is selected on a pre-planned path (as in the $DVSF^2$).

The four algorithms developed based on the concept of F^2 focus on the requirements of various tasks and environments. In the CF^2 method, a robot travels at a constant speed and its moving direction is determined by the resultant force of attractive force and repulsive force. The CF^2 method is especially suitable for robots with limited motion control and computing capabilities. The VSF^2 method broadens the applications of the F^2 method greatly by taking a robot's dynamic and kinematic characteristics into account in motion planning and collision avoidance. A robot's speed is adaptively changed based on environment information and its own status in this method. The SGF^2 method introduces the subgoal into the F^2 method. A robot analyses its surrounding environment using sensor information and then selects a subgoal to follow. The

SGF^2 method is suitable for real-time motion planning and collision avoidance in partially known and dynamically changing environments. The $DVSF^2$ is suitable for integration with a global planner and acting as a local collision avoidance approach. In this method, a temporary waypoint is selected on a pre-planned path. A robot using the $DVSF^2$ method is able to follow the pre-planned path while avoiding collisions. The $DVSF^2$ method is suitable for real-time collision avoidance in a dynamically changing environment. The feasibility and performance of these approaches have been demonstrated by simulation studies and real robot experiments in this thesis.

The parameter setting was shown to be critical to the performance of all four F^2 based methods. This results in an emphasis on optimization approaches to find appropriate parameters for the F^2 based method. The Particle Swarm Optimization method (PSO) has been successfully utilized to solve single objective parameter optimization problems for the F^2 methods. To solve multiobjective parameter optimization problems, a Ranked Pareto Particle Swarm Optimization approach (RPPSO) was proposed, and has been proved to be feasible and efficient for solving the multiobjective parameter optimization problems for robot motion planning and coordination using F^2 based methods.

The F^2 based methods have been proved, by extensive simulation studies and experimental studies, to be applicable in both single robot motion planning and multi-robot coordination. Robots using the F^2 based methods work in a decentralized manner. Each robot determines its motion based on its own status and the updated information. The increase in the number of robots in the work space will thus not cause an exponential increase in computing burden as many conventional approaches do. Another advantage is that the task priority is integrated into the construction of the force field, so a robot with higher task priority will have advantage in collision avoidance, which is an important issue in multi-robot applications.

Experiments and simulations with real robots in Player/Stage have been carried out in various environments. The control softwares were developed in C++ and Player project. The experiments on single robot motion planning and collision avoidance were carried out with a Pioneer robot with a laser rangerfinder installed. Experimental studies on multi-robot motion

planning and collaboration were carried out in the Player/Stage. The experimental results have demonstrated the feasibility and performance of the proposed F^2 based methods.

Future research directions may include the following:

A. Extending F^2 methods from 2D to 3D. F^2 methods provide a generic approach and can be easily extended to three dimensions and applied to robotic manipulators in 3D environments. Some preliminary research works on a 3D F^2 method have been carried out with my colleagues and many promising results have been obtained [68, 69, 71]. Appendix A gives a brief description of the 3-Dimensional Force Field method (3D-F^2).

B. Extending the F^2 methods for various types of robots. The robot model used in this thesis is relative simple, however real robots vary greatly in structures, locomotion mechanisms, and shapes. It is expected that the feasibility and performance of the F^2 methods will be improved by taking further robots' characteristics into consideration.

C. Enhancing the controllability of the F^2 method. There is always a gap between an ideal plan and the real world. An underlying assumption of research work presented in this thesis is that all environmental information, including information from sensors or other robots, is accurate. Robots are assumed to be able to execute the planners' commands instantly and precisely. This is not true in the physical world. Further research will include developing a control framework considering uncertainty and feedback control.

Appendix A
3-Dimensional Force Field

The 3-Dimensional Force Field (3D-F^2) method is a novel approach to real-time motion planning and collision avoidance for robot manipulators. In 3D-F^2, virtual ellipsoid force fields are designed to cover all the links in a robotic manipulator. The interaction between the 3D ellipsoid force fields and obstacles provides a feasible and efficient way for real-time motion planning and collision avoidance for robotic manipulators with a high degree of freedom. Algorithms have been developed based on the concept of 3D-F^2 for different applications [68, 69, 71]. This appendix does not cover all the details of these algorithms, but briefly describes the definition of attractive force, 3-dimensional force field and repulsive force in the 3D-F^2 method [68]. For more details, see [68, 69, 71]

An industrial robotic manipulator (Denso VMD6556) used in this research is illustrated in Figure 7-1 [179]. Each link is modelled as a spring damp joint.

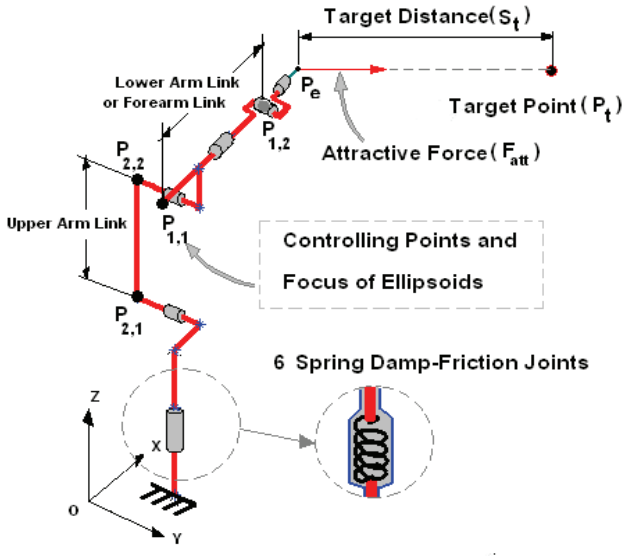


Figure 7-1 Spring damp-friction joints represent the robot arm [68]

i. Definition of 3-Dimensional Force Field

The definition of the 3D-F^2 is described here. To protect a manipulator from possible collision, the proposed force field should envelop the manipulator completely. Since a manipulator usually consists of several links, each link should be covered by a force field.

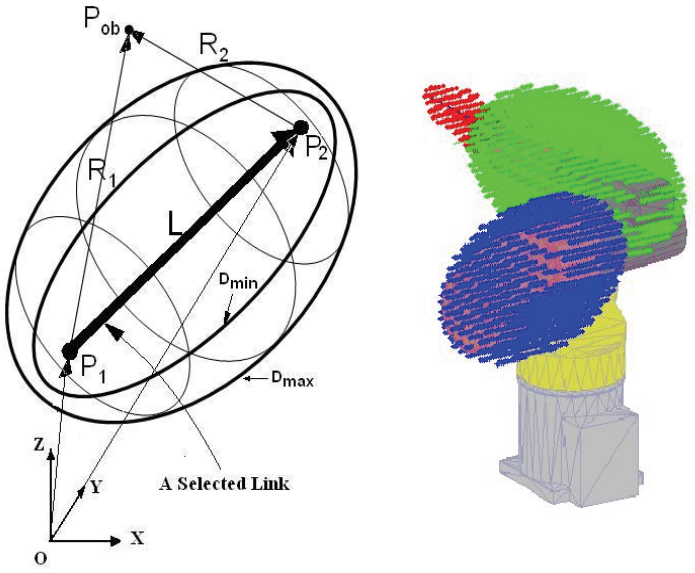


Figure 7-2 (a) Parameters of 3D-F^2 and (b) a robot arm covered by force fields [68]

To determine the ellipsoid covering a link, two points on a link are selected as the focus of the resulting ellipsoid (P_1 and P_2 in Figure 7-2). For any point in 3D space (P_{ob} in Figure 7-2), $\mathbf{R_1}$ and $\mathbf{R_2}$ are vectors from foci P_1 and P_2 to P_{ob}, respectively. Let D_{max} be the influence area of an ellipsoid force field and D_{min} be the area where the magnitude of the repulsive force reaches its maximum. To ensure that the ellipsoid force field will cover the whole body of the link, the length of major axis of D_{min} is set to be equal to $L \times K_p$, where L is the distance between foci and K_p is a constant larger than 1. Define:

$$C_x = \frac{\|R_1\| + \|R_2\|}{L} \qquad \text{(A-1)}$$

For any point P_{ob} in 3D space, if $C_x = K_p$, this point is on the ellipsoid D_{min} and the repulsive force reaches it maximum value. If $C_x < K_p$, this point is inside D_{min}. On the contrary, if $C_x > K_p$, the point is outside D_{min} (see Figure 7-2).

To take the linear speed of the end-effector into account, a new factor E_r is defined to be the ratio of the link's instant speed (v_i) and maximum speed (v_{max}), so $0 \le E_r \le 1$.

$$E_r = \frac{v_i}{v_{max}} \tag{A-2}$$

The length of the major axis of D_{max} is then set to be $L \times \left(K_p + E_r\right)$. So for any point on the ellipsoid D_{max}, $C_x = K_p + E_r$.

The amplitude of repulsive force is defined by

$$f_{rep} = K_f - \left(\frac{K_f}{1 + e^{\left(-\frac{10\left(C_x - K_p - 0.5E_r\right)}{E_r} \right)}} \right) \tag{A-3}$$

$$\mathbf{F_{rep}} = f_{att} \frac{\left(\mathbf{R}_1 - dot(\mathbf{R}_1, \frac{\mathbf{L}}{\|\mathbf{L}\|}) * \frac{\mathbf{L}}{\|\mathbf{L}\|} \right)}{\left\| \mathbf{R}_1 - dot(\mathbf{R}_1, \frac{\mathbf{L}}{\|\mathbf{L}\|}) * \frac{\mathbf{L}}{\|\mathbf{L}\|} \right\|} \tag{A-4}$$

where K_f is the maximum magnitude of repulsive force. The repulsive force is a vector and its direction is defined to be the unit vector that points from P_{ob} to the link perpendicularly.

Figure 7-3 shows how the amplitude of a repulsive force varies with distance when $K_f = 10$, $K_p = 1.05$.

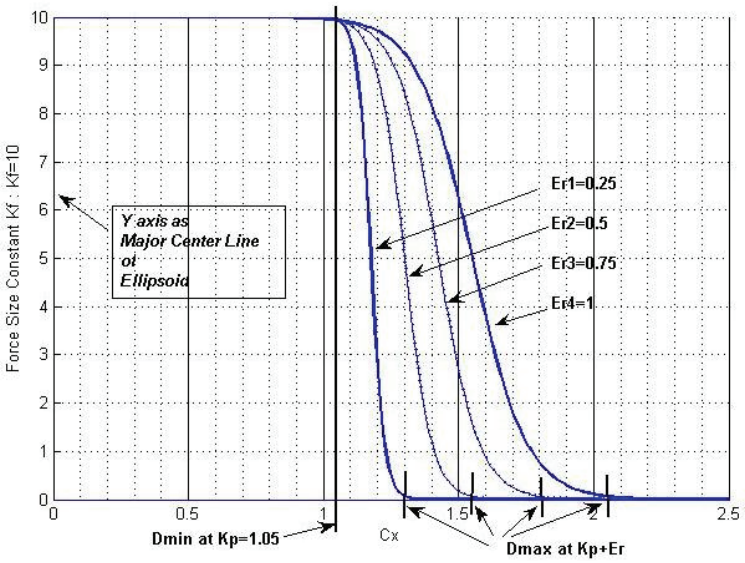


Figure 7-3 The magnitude of force field [68]

From Equation (A-2), E_r increases with the increase of the link's speed V_i. The coverage area of force field D_{max} also increases, as shown in Figure 7-3. When a point in space is far from the link ($C_x > K_p + E_r$), the magnitude of repulsive force is equal to 0. When it moves from the D_{max} to D_{min} ($K_p < C_x < K_p + E_r$), the magnitude of repulsive force increases smoothly from 0 to its maximum (10 in this case). When this point enters D_{min} ($C_x < K_p$), the magnitude is at the maximum.

ii. Attractive Force

As in the F^2 method, an attractive force is generated from a target point in the 3D-F^2 method. The attractive force $\vec{\mathbf{F}}_{\mathbf{att}}$ is defined as a function of the distance between the manipulator end-effecter P_e and the target's position P_t.

$$S_t = \left\| \vec{\mathbf{P}}_{\mathbf{t}} - \vec{\mathbf{P}}_{\mathbf{e}} \right\| \tag{A-5}$$

$$f_{att} = \frac{K_{att}}{1 + \left(\frac{e^{-K_s S_t}}{K_{zero}} \right)} \tag{A-6}$$

$$\vec{\mathbf{F}}_{\mathbf{att}} = f_{att} \frac{\vec{\mathbf{P}}_{\mathbf{t}} - \vec{\mathbf{P}}_{\mathbf{e}}}{S_t} \tag{A-7}$$

where $\vec{P}_e$ is the position vector of a manipulator's end-effector and $\vec{P}_t$ is the position vector of its target point in 3-dimensional space. S_t denotes the distance between the end-effector and the target point. K_{att} is a positive constant which determines the magnitude of the attractive force. K_{zero} is a small non-zero positive constant and K_s is a constant which will determine how fast the attractive force varies with distance between the end-effector and the target point. This attractive force ties at the end-effector like a pulling string, which is tightly tense to the target point, and its direction points from the current position of the end-effector to the target, as given by Equation (A-7).

REFERENCE

[1] "iRobot," http://www.irobot.com/index.cfm, accessed 20 Dec. 2008.

[2] J. L. Jones, "Robots at the tipping point: the road to iRobot Roomba," *IEEE Robotics & Automation Magazine,* vol. 13, pp. 76-78, 2006.

[3] T. Nishiyama, F. Takiuchi, K. Ando, and M. Arisawa, "The functional evaluation of future wheelchairs contributing to ecological aid in traveling," *Proceedings of the Fourth International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, 2005, pp. 844-849.

[4] T. Taha, J. V. Miro, and D. Liu, "An efficient path planner for large mobile platforms in cluttered environments," *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics*, Bangkok, Thailand, 2006, pp. 225-230.

[5] I. R. Nourbakhsh, C. Kunz, and T. Willeke, "The mobot museum robot installations: a five year experiment," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 3636-3641.

[6] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "MINERVA: a second-generation museum tour-guide robot," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 1999-2005.

[7] M. Bennewitz, "Mobile robot navigation in dynamic environments," PhD Thesis, University of Freiburg, 2004.

[8] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes, "Field and service applications - an autonomous straddle carrier for movement of shipping containers - from research to operational autonomous Systems," *IEEE Robotics & Automation Magazine,* vol. 14, pp. 14-23, 2007.

[9] "Patrick: technology & systems: autostrad," http://www.patrick.com.au/IRM/Content/technology/autostrad.html, accessed 20 Dec. 2008.

[10] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: controller design, experimental validation and racing," *Proceedings of the 2007 American Control Conference*, 2007, pp. 2296-2301.

[11] "DARPA Grand Challenge Information," http://www.darpa.org/, accessed at Dec. 2008.

[12] J.-C. Latombe, "Robot motion planning," Kluwer Academic Publishers, Boston, 1991.

[13] Y. K. Hwang and N. Ahuja, "Gross motion planning - a survey," *ACM Computing Surveys,* vol. 24, 1992.

[14] "Robot specifications," http://www.activrobots.com/ROBOTS/specs.html, accessed 20 Dec. 2008.

[15] D. K. Liu and A. K. Kulatunga, "Simultaneous planning and scheduling for multi-autonomous vehicles," *Evolutionary Scheduling*, K. P. Dahal, K. C. Tan, and P. I. Cowling, Eds.: Springer-Verlag, 2007, pp. 437-464.

[16] H. J. Chang, C. S. G. Lee, Y. C. Hu, and L. Yung-Hsiang, "Multi-robot SLAM with topological/metric maps," *Proceedings of the 2007 IEEE/RSJ International Conference onIntelligent Robots and Systems*, 2007, pp. 1467-1472.

[17] X. S. Zhou and S. I. Roumeliotis, "Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1785-1792.

[18] M. Bryson and S. Sukkarieh, "Co-operative localisation and mapping for multiple UAVs in unknown environments," *Proceedings of the IEEE Aerospace Conference*, 2007, pp. 1-12.

[19] "RoboCup Official Site," http://www.robocup.org/, accessed 20 Dec. 2008.

[20] H. Kitano, S. Suzuki, and J. Akita, "RoboCup Jr.: RoboCup for edutainment," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, pp. 807-812.

[21] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*. vol. 1: Springer, 1959, pp. 269 - 271.

[22] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics,* vol. 4, pp. 100-107, 1968.

[23] D. Rina and P. Judea, "Generalized best-first search strategies and the optimality of A*," *Journal of the ACM,* vol. 32, pp. 505-536, 1985.

[24] T. Lozano-Pérez and M. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM,* vol. 22, pp. 560-570, 1979.

[25] A. Franz, "Voronoi diagrams - a survey of a fundamental geometric data structure," *ACM Computing Surveys,* vol. 23, pp. 345-405, 1991.

[26] K. Hoff, III, T. Culver, J. Keyser, M. C. Lin, and D. Manocha, "Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams,"

Proceedings of the IEEE International Conference on Robotics and Automation, 2000, pp. 2931-2937.

[27] E. Masehian, M. R. Amin-Naseri, and S. E. Khadem, "Online motion planning using incremental construction of medial axis," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 2928-2933.

[28] R. Wein, J. v. d. Berg, and D. Halperin, "Planning high-quality paths and corridors amidst obstacles," *The International Journal of Robotics Research,* vol. 27, pp. 1213-1231, 2008.

[29] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996, pp. 113-120.

[30] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Transactions on Robotics and Automation,* vol. 14, pp. 166-171, 1998.

[31] C. Nissoux, T. Simeon, and J. P. Laumond, "Visibility based probabilistic roadmaps," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1999, pp. 1316-1321.

[32] J. P. van den Berg, D. Nieuwenhuisen, L. Jaillet, and M. H. Overmars, "Creating robust roadmaps for motion planning in changing environments," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1053-1059.

[33] S. Hrabar, "3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 807-814.

[34] B. Frank, M. Becker, C. Stachniss, W. Burgard, and M. Teschner, "Efficient path planning for mobile robots in environments with deformable objects," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3737-3742.

[35] Z. Liangjun and D. Manocha, "An efficient retraction-based RRT planner," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3743-3750.

[36] J. J. Kuffner, Jr. and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, pp. 995-1001.

[37] M. Zucker, J. Kuffner, and J. A. Bagnell, "Adaptive workspace biasing for sampling-based planners," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3757-3762.

[38] G. F. Liu and J. C. Trinkle, "Complete path planning for planar closed chains among point obstacles," *Robotics: Science and Systems*: MIT Press, 2005.

[39] P. C. Chen and Y. K. Hwang, "SANDROS: a dynamic graph search algorithm for motion planning," *IEEE Transactions on Robotics and Automation,* vol. 14, pp. 390-403, 1998.

[40] D. J. Zhu and J. C. Latombe, "New heuristic algorithms for efficient hierarchical path planning," *IEEE Transactions on Robotics and Automation,* vol. 7, pp. 9-20, 1991.

[41] L. Zhang, Y. J. Kim, and D. Manocha, "Efficient cell labelling and path non-existence computation using c-obstacle query," *The International Journal of Robotics Research,* vol. 27, pp. 1246-1257, 2008.

[42] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research,* vol. 5, pp. 90-98, 1986.

[43] C. I. Connolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1990, pp. 2102-2106

[44] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Autonomous Robots,* vol. 13, pp. 207-222, 2002.

[45] J. Ren, K. A. McIsaac, R. V. Patel, and T. M. Peters, "A potential field model using generalized sigmoid functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 37, pp. 477-484, 2007.

[46] J. Jih-Gau, "Collision avoidance using potential fields," *International Journal of Industrial Robot,* vol. 25, no. 6, pp. 408-415, 1998.

[47] K. Nishiwaki and K. i. Yano, "Variable impedance control of meal assistance robot using potential method," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* 2008, pp. 3242-3247.

[48] J. Ramirez-Gordillo, E. A. Merchan-Cruz, E. Lugo-Gonzalez, R. Ponce-Reynoso, R. G. Rodiguez-Canizo, G. Urriolagoitia-Sosa, and B. Subudhi, "The Laplacian artificial potential field (LAPF) for the path planning of robotic manipulators," *Proceedings of Electronics, Robotics and Automotive Mechanics Conference*, 2008, pp. 508-513.

[49] K.-S. Hwang, M.-Y. Ju, and Y.-J. Chen, "Speed alteration strategy for multijoint robots in co-working environment," *IEEE Transactions on Industrial Electronics,* vol. 50, pp. 385-393, 2003.

[50] C. W. Warren, "Multiple robot path coordination using artificial potential fields," *Proceedings of 1990 IEEE International Conference on Robotics and Automation*, 1990, pp. 500-505

[51] A. A. Masoud, "Using hybird vector-harmonic potential fields for multi-robot, multi-target navigation in a stationary environment," *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, 1996, pp. 3564-3571.

[52] E. G. Hernandez-Martinez and E. Aranda-Bricaire, "Non-collision conditions in multi-agent robots formation using local potential functions," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3776-3781.

[53] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Transactions on Robotics,* vol. 24, pp. 1213-1223, 2008.

[54] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Transactions on Robotics,* vol. 24, pp. 1416-1428, 2008.

[55] H. Seraji and B. Bon, "Real-time collision avoidance for position-controlled manipulators," *IEEE Transactions on Robotics and Automation,* vol. 15, pp. 670-677, 1999.

[56] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1993, pp. 802-807

[57] O. Brock and O. Khatib, "Real-time re-planning in high-dimensional configuration spaces using sets of homotopic paths," *Proceedings of IEEE International Conference on Robotics and Automation*, 2000, pp. 550-555.

[58] F. Seto, K. Kosuge, and Y. Hirata, "Self-collision avoidance motion control for human robot cooperation system using RoBE," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3143-3148.

[59] R. Simmons, "The curvature-velocity method for local obstacle avoidance," *Proceedings of the IEEE International Conference on Robotics and Automation*. vol. 4, 1996, pp. 3375-3382.

[60] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine,* vol. 4, pp. 23-33, 1997.

[61] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," *Proceedings of IEEE International Conference on Robotics and Automation*, 1999, pp. 341-346.

[62] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation,* vol. 7, pp. 278-288, 1991.

[63] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," *Proceedings of the IEEE International Conference on Robotics and Automation*. vol. 2, 1998, pp. 1572-1577.

[64] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with look-ahead verification," *Proceedings of the IEEE International Conference on Robotics and Automation,* vol. 3, 2000, pp. 2505-2511.

[65] D. L. Wang, D. K. Liu, X. Wu, and K. C. Tan, "A force field method for robot navigation," *Proceedings of the Third International Conference on Computational Intelligence, Robotics and Autonomous Systems*, 2005, pp. 662-667.

[66] D. K. Liu, D. Wang, and G. Dissanayake, "A force field method based multi-robot collaboration," *Proceedings of the IEEE International Conference on Robotics, Autonomous & Mechatronics*, Bangkok, Thailand, 2006, pp. 662-667.

[67] D. Wang, D. Liu, and G. Dissanayake, "A variable speed force field method for multi-robot collaboration," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2697-2702.

[68] P. Chotiprayanakul, D. K. Liu, D. Wang, and G. Dissanayake, "Collision-free trajectory planning for manipulators using virtual force based approach," *Proceedings of the International Conference on Engineering, Applied Sciences, and Technology*, Bangkok, Thailand, 2007.

[69] P. Chotiprayanakul, D. K. Liu, D. Wang, and G. Dissanayake, "A 3-dimensional force field method for robot collision avoidance in complex environments," *Proceedings of the 24th International Symposium on Automation and Robotics in Construction*, Kochi, Kerala, India, 2007, pp. 139-145.

[70] D. Wang, N. M. Kwok, D. K. Liu, and G. Dissanayake, "PSO-tuned F2 method for multi-robot navigation," *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, California, USA, 2007, pp. 3765-3770.

[71] P. Chotiprayanakul, D. Wang, N. M. Kwok, and D. K. Liu, "A haptic based human robot interaction approach for robotic grit blasting," *Proceedings of the 25th International Symposium on Automation and Robotics in Construction*, Vilnius, Lithuania, 2008, pp. 148-154.

[72] J. V. Miró, T. Taha, D. Wang, G. Dissanayake, and D. Liu, "An efficient strategy for robot navigation in cluttered environments in the presence of dynamic obstacles," *Proceedings of the Eighth International Conference on Intelligent Technology*, Sydney, Australia, 2007, pp. 74-81.

[73] J. V. Miró, T. Taha, D. Wang, and G. Dissanayake, "An adaptive manoeuvring strategy for mobile robots in cluttered dynamic environments," *International Journal of Automation and Control,* vol. 2, Nos. 2/3, pp. 178-194, 2008.

[74] D. Wang, D. K. Liu, N. M. Kwok, and K. J. Waldron, "A subgoal-guided force field method for robot navigation," *Proceedings of the 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Beijing, China, 2008, pp. 488-494.

[75] D. Wang, N. M. Kwok, D. K. Liu, and Q. P. Ha, "Ranked Pareto particle swarm optimization for mobile robot motion planning," *Design and Control of Intelligent Robotic Systems,* Berlin Heidelberg: Springer-Verlag, 2009, pp. 97-118.

[76] M. Clifton, G. Paul, N. Kwok, D. Liu, and D. Wang, "Evaluating performance of multiple RRTs," *Proceedings of the 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, Beijing, China, 2008, pp. 564-569.

[77] J. Barraquand, B. Langlois, and J. C. Latombe, "Numerical potential field techniques for robot path planning," *Proceedings of the Fifth International Conference on Advanced Robotics*, 1991, pp. 1012-1017.

[78] J. G. Juang, "Robot collision avoidance control using distance computation," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* vol. 3, 1995, pp. 2564-2569.

[79] L. Chien-Chou, P. Chi-Chun, and C. Jen-Hui, "A novel potential-based path planning of 3-D articulated robots with moving bases," *Proceedings of the IEEE International Conference on Robotics and Automation,* vol. 3, 2003, pp. 3365-3370.

[80] M. G. Park and M. C. Lee, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronic*, 2003, pp. 735 - 740.

[81] M. T. Wolf and J. W. Burdick, "Artificial potential functions for highway driving with collision avoidance," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3731-3736.

[82] M. C. Mora and J. Tornero, "Path planning and trajectory generation using multi-rate predictive artificial potential fields," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2990-2995.

[83] N. A. Scott and C. R. Carignan, "A line-based obstacle avoidance technique for dexterous manipulator operations," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 3353-3358.

[84] G. Besseron, C. Grand, F. Ben Amar, and P. Bidaud, "Decoupled control of the high mobility robot Hylos based on a dynamic stability margin," *Proceedings of the IEEE/RSJ International Conference onIntelligent Robots and Systems*, 2008, pp. 2435-2440.

[85] G. C. Karras and K. J. Kyriakopoulos, "Visual servo control of an underwater vehicle using a laser vision system," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 4116-4122.

[86] R. Jing, K. A. McIsaac, and R. V. Patel, "Modified Newton's method applied to potential field-based navigation for mobile robots," *IEEE Transactions on Robotics,* vol. 22, pp. 384-391, 2006.

[87] O. Brock, "Generating robot motion: the integration of planning and execution," PhD Thesis, Stanford University, 1999.

[88] T. Sattel and T. Brandt, "Ground vehicle guidance along collision-free trajectories using elastic bands," *Proceedings of the 2005 American Control Conference*, 2005, pp. 4991-4996.

[89] S. K. Gehrig and F. J. Stein, "Elastic bands to enhance vehicle following," *Proceedings of the 2001 IEEE Intelligent Transportation Systems*, 2001, pp. 597-602.

[90] M. Khatib, H. Jaouni, R. Chatila, and J. P. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, 1997, pp. 2920-2925.

[91] S. Quinlan, "Real-time modification of collision-free paths," PhD Thesis, Stanford University, 1995.

[92] V. Delsart and T. Fraichard, "Navigating dynamic environments using trajectory deformation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 226-233.

[93] O. Brock, O. Khatib, and S. Viji, "Task-consistent obstacle avoidance and motion behavior for mobile manipulation," *Proceedings of the IEEE International Conference onRobotics and Automation*, 2002, pp. 388-393.

[94] O. Brock and O. Khatib, "Executing motion plans for robots with many degrees of freedom in dynamic environments," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998, pp. 1-6.

[95] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation,* vol. 16, pp. 615-620, 2000.

[96] Y. Lu and Y. Yixin, "An improved potential field method for mobile robot path planning in dynamic environments," *Proceedings of the 7th World Congress on Intelligent Control and Automation*, 2008, pp. 4847-4852.

[97] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1991, pp. 1398-1404.

[98] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots in cluttered environments," *Proceedings of IEEE International Conference on Robotics and Automation*, 1990, pp. 572-577.

[99] D. An and H. Wang, "VPH: a new laser radar based obstacle avoidance method for intelligent mobile robots," *Proceedings of Fifth World Congress on Intelligent Control and Automation*, 2004, pp. 4681-4685.

[100] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," *Proceedings of the IEEE International Conference on Robotics and Automation*. vol. 2, 1985, pp. 116-121.

[101] D. Fox, W. Burgard, and S. Thrun, "Controlling synchro-drive robots with the dynamic window approach to collision avoidance," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems,* vol. 3, 1996, pp. 1280-1287.

[102] P. Ogren and N. E. Leonard, "A tractable convergent dynamic window approach to obstacle avoidance," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System,* vol. 1, 2002, pp. 595-600.

[103] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics,* vol. 21, pp. 188-195, 2005.

[104] A. Stentz, "The focussed D* algorithm for real-time replanning," *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[105] M. Seder, K. Macek, and I. Petrovic, "An integrated approach to real-time mobile robot control in partially known indoor environments," *Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society*, 2005, pp. 1785-1790.

[106] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 1986-1991.

[107] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation,* vol. 8, pp. 501-518, 1992.

[108] K. Nak Yong and R. G. Simmons, "The lane-curvature method for local obstacle avoidance," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998, pp. 1615-1621.

[109] P. Tournassoud, "A strategy for obstacle avoidance and its application to mullti-robot systems," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986, pp. 1224-1229.

[110] J. Barraquand and J. C. Latombe, "A Monte-Carlo algorithm for path planning with many degrees of freedom," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1990, pp. 1712-1717.

[111] J. Barraquand, B. Langlois, and J. C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 22, pp. 224-241, 1992.

[112] P. Svestka and M. H. Overmars, "Coordinated motion planning for multiple car-like robots using probabilistic roadmaps," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1995, pp. 1631-1636.

[113] P. Svestka and M. H. Overmars, "Coordinated path planning for multiple robots," *Robotics and Autonomous Systems,* vol. 23, pp. 125-152, 1998.

[114] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986, pp. 1419-1424.

[115] Y. H. Liu, S. Kuroda, T. Naniwa, H. Noborio, and S. Arimoto, "A practical algorithm for planning collision-free coordinated motion of multiple mobile robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 1427-1432.

[116] S. J. Buckley, "Fast motion planning for multiple moving robots," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 322-326.

[117] H. Chu and H. A. ElMaraghy, "Real-time multi-robot path planner based on a heuristic approach," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992, pp. 475-480.

[118] K. Azarm and G. Schmidt, "A decentralized approach for the conflict-free motion of multiple mobile robots," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996, pp. 1667-1675.

[119] D. Herrero-Perez and H. Matinez-Barbera, "Decentralized coordination of autonomous AGVs in flexible manufacturing systems," *Proceedings of the IEEE/RSJ International Conference onIntelligent Robots and Systems*, 2008, pp. 3674-3679.

[120] J. Zheng, H. Yu, W. Liang, and P. Zeng, "A distributed and optimal algorithm to coordinate the motion of multiple mobile robots," *Proceedings of the 7th World Congress on Intelligent Control and Automation*, 2008, pp. 3027-3032.

[121] S. Kato, S. Nishiyama, and J. Takeno, "Coordinating mobile robots by applying traffic rules," *Proceedings of the 1992 lEEE/RSJ International Conference on Intelligent Robots and Systems*, 1992, pp. 1535-1541.

[122] K. Azarm and G. Schmidt, "Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1997, pp. 3526-3533.

[123] G. Sanchez and J. C. Latombe, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems," *Proceedings of the IEEE International Conference on Robotics and Automation,* 2002, vol.2, pp. 2112-2119.

[124] Y. Guo and L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," *Proceedings of the IEEE International Conference on Robotics and Automation,* vol. 3, 2002, pp. 2612-2619.

[125] J. P. v. d. Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," *Proceedings of the IEEE/RSL International Conference on International Robots and Systems*, 2005, pp. 2217-2222.

[126] M. Bennewitz, W. Burgard, and S. Thrun, "Optimizing schedules for prioritized path planning of multi-robot systems," *Proceedings of the IEEE International Conference on Robotics and Automation,* vol. 1, 2001, pp. 271-276.

[127] D.K.Liu, X.Wu, A.K.Kulatunga, and G. Dissanayake, "Motion coordination of multiple autonomous vehicles in dynamic and strictly constrained environments," *Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems,* Bangkok, Thailand, 2006, pp. 204-209.

[128] Z. Bien and J. Lee, "A minimum-time trajectory planning method for two robots," *IEEE Transactions on Robotics and Automation,* vol. 8, pp. 414-418, 1992.

[129] P. A. O'Donnell and T. Lozano-Periz, "Deadlock-free and collision-free coordination of two robot manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1989, pp. 484-489.

[130] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *International Journal of Robotics Research,* vol. 5, pp. 72-89, 1986.

[131] C. Ferrari, E. Pagello, M. Voltolina, J. Ota, and T. Arai, "Varying paths and motion profiles in multiple robot motion planning," *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997, pp. 186-193.

[132] M. Jager and B. Nebel, "Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 1213-1219.

[133] C. Ferrari, E. Pagello, J. Ota, and T. Arai, "Multirobot motion coordination in space and time," *Robotics and Autonomous Systems,* vol. 25, pp. 219-229, 1998.

[134] A. Stentz, "Optimal and efficient path planning for partially-known environments," *Proceedings of the IEEE International Conference on Robotics and Automation*. vol. 4, 1994, pp. 3310-3317.

[135] Y.-C. Chang and Y. Yoshio, "Dynamic decision making of mobile robot under obstructed environment," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 4091-4096.

[136] "AmigoBot robot for education & collaborative research," http://www.activrobots.com/ROBOTS/amigobot.html, accessed 20 Dec. 2008.

[137] C. W. Warren, "A vector based approach to robot path planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1991, pp. 1021-1026.

[138] S. Ando, "A fast collision-free path planning method for a general robot manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 2871-2877.

[139] B. H. Krogh and D. Feng, "Dynamic generation of subgoals for autonomous mobile robots using local feedback information," *IEEE Transactions on Automatic Control,* vol. 34, pp. 483-493, 1989.

[140] Y. Xiaoyu, M. Moallem, and R. V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B,* vol. 35, pp. 1214-1224, 2005.

[141] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: tools for multi-robot and distributed sensor systems," *Proceedings of the International Conference on Advanced Robotics*, Coimbra, Portugal, 2003, pp. 317-323.

[142] D. Fox, "Adapting the sample size in particle filters through KLD-sampling," *The International Journal of Robotics Research,* vol. 22, 2003.

[143] "Pioneer robot," http://www.activrobots.com/ROBOTS/p2dx.html, accessed 20 Dec. 2008.

[144] V. J. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 20, pp. 1058-1069, 1990.

[145] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks*, Perth, WA, 1995, pp. 1942-1948.

[146] Sheetal and G. K. Venayagamoorthy, "Unmanned vehicle navigation using swarm intelligence," *Proceedings of the International Conference on Intelligent Sensing and Information Processing*, 2004, pp. 249-253.

[147] X. Chen and Y. Li, "Smooth path planning of a mobile robot using stochastic particle swarm optimization," *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, 2006, pp. 1722-1727.

[148] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," *Proceedings of the IEEE Swarm Intelligence Symposium*, 2007, pp. 120-127.

[149] M. Qianzhi, L. Xiujuan, and Z. Qun, "Mobile Robot Path Planning with Complex Constraints Based on the Second-Order Oscillating Particle Swarm Optimization Algorithm," *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering,* 2009, pp. 244-248.

[150] L. Dongmei, Q. Shenshan, and W. Dong, "Particle swarm optimization based on neighborhood encoding for traveling salesman problem," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 1276-1279.

[151] Eberhart and S. Yuhui, "Particle swarm optimization: developments, applications and resources," *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, vol. 1, pp. 81-86.

[152] S. Mei-Ping and G. Guo-Chang, "Research on particle swarm optimization: a review," *Proceedings of 2004 International Conference on Machine Learning and Cybernetics,* 2004, pp. 2236-2241 vol.4.

[153] H. Xiaohui, S. Yuhui, and R. Eberhart, "Recent advances in particle swarm," *Proceedings of the 2004 Congress on Evolutionary Computation*, 2004, vol.1, pp. 90-97.

[154] D. Tsou and C. MacNish, "Adaptive particle swarm optimisation for high-dimensional highly convex search spaces," *Proceedings of the 2003 Congress on Evolutionary Computation*, 2003, vol. 2, pp. 783-789.

[155] R. C. Eberhart and S. Yuhui, "Tracking and optimizing dynamic systems with particle swarms," *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001, vol. 1, pp. 94-100.

[156] A. Atyabi, S. Phon-Amnuaisuk, and C. K. Ho, "Robot navigation with enhanced versions of particle swarm optimization," *Applied Soft Computing, doi:10.1016 /j.asoc.2009.06.017,* 2009.

[157] M. Reyes-Sierra and C. A. C. Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research,* vol. 2, No.3, pp. 287-308, 2006.

[158] L. Wen-Fung and G. G. Yen, "PSO-Based Multiobjective Optimization With Dynamic Population Size and Adaptive Local Archives," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 38, pp. 1270-1293, 2008.

[159] A. Shubham, Y. Dashora, M. K. Tiwari, and S. Young-Jun, "Interactive Particle Swarm: A Pareto-Adaptive Metaheuristic to Multiobjective Optimization," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 38, pp. 258-277, 2008.

[160] D. Y. Sha and H. H. Lin, "A Multi-objective PSO for job-shop scheduling problems," *Proceedings of the International Conference on Computers & Industrial Engineering*, 2009, pp. 489-494.

[161] S. Tsung-Ying, W. Wun-Ci, T. Shang-Jeng, L. Chan-Cheng, C. Shih-Yuan, and H. Sheng-Ta, "Particle swarm optimizer for multi-objective problems based on proportional distribution and cross-over operation," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2008, pp. 2658-2663.

[162] M. A. Abido, "Multiobjective particle swarm for environmental/economic dispatch problem," *Proceedings of the International Power Engineering Conference*, 2007, pp. 1385-1390.

[163] L. Xiaodong, J. Branke, and M. Kirley, "On performance metrics and particle swarm methods for dynamic multiobjective optimization problems," *Proceedings of the IEEE Congress on Evolutionary Computation*, 2007, pp. 576-583.

[164] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," *Proceedings of the 2002 Congress on Evolutionary Computation,* 2002, pp. 1051-1056.

[165] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: an review," *Swarm Intelligence,* vol. 1, pp. 33-57, 2007.

[166] H.-X. Yi, L. Xiao, and P.-K. Liu, "Intelligent Algorithms for Solving Multiobjective Optimization Problems," *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1-5.

[167] G. G. Yen and L. Wen Fung, "Dynamic Multiple Swarms in Multiobjective Particle Swarm Optimization," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 39, pp. 890-911, 2009.

[168] Z. Qian and M. Mahfouf, "A modified PSO with a dynamically varying population and its application to the multi-objective optimal design of alloy steels," *Proceedings of the IEEE Congress on Evolutionary Computation,* 2009, pp. 3241-3248.

[169] H. Xiaohui, R. C. Eberhart, and S. Yuhui, "Particle swarm with extended memory for multiobjective optimization," *Proceedings of the 2003 IEEE Swarm Intelligence Symposium,* 2003, pp. 193-197.

[170] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation,* vol. 8, pp. 256-279, 2004.

[171] J. Jae Bum and R. Christ, "Pareto optimal multi-robot coordination with acceleration constraints," *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008, pp. 1942-1947.

[172] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proceedings of the First International Conference on Genetic Algorithms: Genetic Algorithms and their Applications*, 1985, pp. 93-100.

[173] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," *Proceedings of the Fifth International Conference on Genetic Algorithms*, 1993, pp. 416-423.

[174] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," *Proceedings of the 1994 Congress on Evolutionary Computation*, 1994, pp. 82-87.

[175] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation,* vol. 3, pp. 257-271, 1999.

[176] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland 2001.

[177] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation,* vol. 8, pp. 173-195, 2000.

[178] "Laser Measurement Sensors," http://mysick.com/eCat.aspx?go=DataSheet&Cat=Gus&At=Fa&Cult=English&Category=Produktfinder&ProductID=9168, accessed 20 Dec. 2008.

[179] "Industrial Robots Comparative Sheet," http://www.denso-wave.com/en/robot/product/specslist/index.html, accessed at Dec. 2008.