Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

PSO-Tuned F^2 Method for Multi-Robot Navigation

D. Wang, N. M. Kwok, D. K. Liu, H. Lau, G. Dissanayake

Abstract—**The F^2 (Force Field) method is a novel approach for multi-robot motion planning and collision avoidance. The setting of parameters is however vital to its performance. This paper presents an approach using Particle Swarm Optimization (PSO) to properly determine the control parameters for the F^2 method. The goal of the optimization is to minimize the resultant path lengths. The approach presented in this paper can be used as a tool to obtain optimal parameters for various tasks before their execution. Simulations are carried out in various environments to show the feasibility of this approach.**

I. Introduction

THIS paper addresses the problem of motion planning for multiple robots. Although many algorithms have been proven to be feasible and efficient for single-robot motion planning and collision avoidance, they cannot be transferred directly to multi-robot systems. The existing methods are often categorized into centralized and decentralized approaches [1]. Centralized approaches consider all robots together as if they are forming a high degree of freedom system and are capable of providing complete and optimal solutions. However, as the number of robots and obstacles in the working environment increases, such approaches will suffer from the exponentially increasing computational complexity. Decentralized approaches generate each path for individual robot independently and avoid collisions locally. Decentralized approaches are not affected by the number of robots but are usually incomplete, i.e., they may fail to find a solution even if it exists.

Priority planning is an efficient approach for multi-robot motion planning [2-7]. Techniques of this class assign priorities to each robot and compute paths in order of decreasing priority. A robot with higher priority is treated as an obstacle in the planning of a robot with lower priority. This reduces the multi-robot motion planning problem into several single-robot motion planning problems.

The configuration-space-time method was first introduced in [2]. By introducing time as an additional dimension, this approach discretizes the configuration space to a sequence of slices of the configuration space at successive time intervals.

Manuscript received March 23, 2007. This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government, Australia.

All Authors are with the ARC Centre of Excellence for Autonomous Systems (CAS), Faculty of Engineering, University of Technology, Sydney, (UTS), NSW 2007, Australia. (e-mail: {Da-Long.Wang, ngai.kwok, dkliu, hlau, gdissa}@ eng.uts.edu.au).

Some approaches decomposed the motion planning problem into two smaller sub-problems: path planning and velocity planning [8-10]. Under these approaches, robots are kept on their preplanned paths and speed changing strategies are applied to avoid collisions.

The F^2 method is a force field based method for multi-robot path planning and collaboration [11]. Instead of generating potential fields or force fields based on obstacles, a virtual force field is constructed for each robot based on its status, including traveling speed, dimension, priority, location and environmental factors, etc. The force field of a robot is different from those of other robots due to its different status and varies with the robot during its movement. A robot with larger volume, traveling with higher speed, or with higher task priority than other robots will have priority in collision avoidance. The interaction among the robots' force fields and obstacles provides a feasible and efficient way for multi-robot motion planning and collaboration.

This paper builds upon the F^2 method proposed in our previous work [11], which operates using a number of parameters. The setting of these parameters can noticeably affect its performance. Accordingly, this paper analyses the effect of parameters in the F^2 method. Since the problem is highly coupled, non-linear and incomplete, closed-form solutions are not always available. An emerging algorithm in the evolutionary computation family, the Particle Swarm Optimization (PSO), is then utilized to obtain appropriate parameters in single-robot and multi-robot cases. PSO was proposed by Kennedy and Eberhart in 1995, motivated by social behavior of organisms such as bird flocking [12]. Due to its simple mechanism and high performance of global optimization, PSO has been applied for many optimization problems successfully, including motion planning problems [13, 14].

This paper is organized as follows. In Section II, we give the detailed description of the F^2 method. Then a simple case is studied to show how the selections of parameters affect its performance. Section III introduces the concept of PSO and describes how to use PSO for parameter optimization. The feasibility of the presented approach is supported by simulations in Section IV. Conclusion and future work are given in Section V.

II. F^2 Method

In the F^2 method, a robot is assumed to travel in a 2-D environment and its location can be precisely known. Each robot is aware of the updated information of all other robots,

TABLE I
PARAMETERS IN F^2 METHOD

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

such as their locations, speeds, etc. For simplicity of explanation, robots are represented as round discs, but can be easily extended to other complex geometric shapes. For more information regarding the F^2 method, refer to [15, 16].

A. Definition of a Force Field

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 with the decrease of the distance to the robot. Parameters in the F^2 method are listed in Table I.

We define:

$$\theta = \angle(\theta_o, \theta_r) \tag{1}$$

$$E_r = v_r / (v_{max} \times C) \tag{2}$$

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

$$D_{min} = \rho_0 \times D_{max} \tag{4}$$

For any point (x, y) in the 2-D space, θ denotes the relative angle of this point to the robot's orientation (Fig. 1). C is a positive number which denotes the environment influence to the force field with $C > 1$. E_r is a positive decimal fraction with $0 \le E_r < 1$. k is a positive multiplier which determines the coverage area of the force field. D_{max} is the maximum active 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 $0 < \rho_0 < 1$ and heavily influences how close the robot can get to obstacles. T_p represents the priority of a task which is undertaken by the robot with $T_p \ge 1$.

The repulsive force generated by a robot is defined by:

$$\left|\vec{F}_{repusive\_robot}\right| = \begin{cases} 0; & when\ D > D_{max} \\ P \times \dfrac{D_{max} - D}{D_{max} - D_{min}}; & when\ D_{max} \le D < D_{min} \\ F_{max}; & when\ D < D_{min} \end{cases} \tag{5}$$

where D is the shortest distance from point (x, y) 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. F_{max} is the maximum repulsive force which will cause the maximum deceleration on the robot. P and F_{max} should be selected based on the robot's characteristics, with $F_{max} >> P$. (5) shows that the magnitude of repulsive force varies with distance. To represent the force field in contours, we further define:

$$\rho = D / D_{max} \tag{6}$$

Then (5) can be re-presented in an alternative form as:

$$\left|\vec{F}_{repusive\_robot}\right| = \begin{cases} 0; & when\ \rho > 1 \\ P \times \dfrac{1-\rho}{1-\rho_0}; & when\ \rho_0 \le \rho < 1 \\ F_{max}; & when\ \rho < \rho_0 \end{cases} \tag{7}$$

When ρ changes from ρ_0 to 1, the magnitude of the repulsive force changes from P to 0.

B. Attractive Force

When a robot is allocated a particular task, for example, traveling from start point (X_{start}, Y_{start}) to goal point (X_{goal}, Y_{goal}), a virtual attractive force which attracts the robot from the start point to the goal point is generated.

$$\left|\vec{F}_{field}\right| = Q \tag{8}$$

where Q is a positive constant scalar which determines the magnitude of the attractive force. The attractive force directs to the goal point from the centre of the robot and can be assumed to be a constant. It drives a robot to its destination (X_{goal}, Y_{goal}).

C. Reaction Force

When a robot approaches an obstacle or other robots' force fields, its force field is suppressed and the robot is repelled away by the virtual reaction force from obstacles or other robots. This reaction force can be calculated by:

$$\left|\vec{F}_{rep}\right| = \left|\vec{F}_{repulsive\_robot}\right| \tag{9}$$

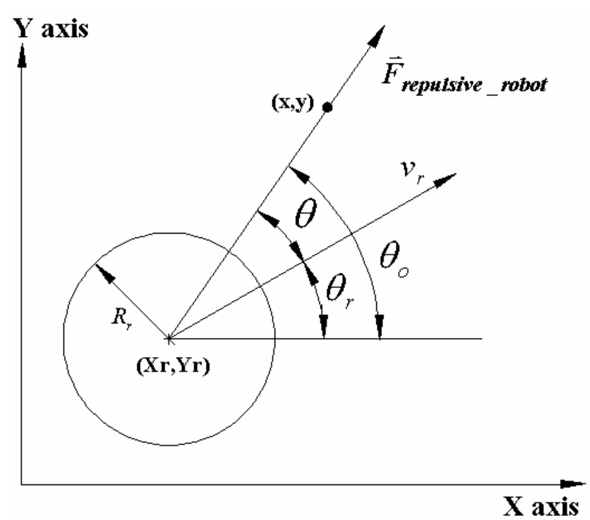

Fig. 1. Illustration of a robot's parameters

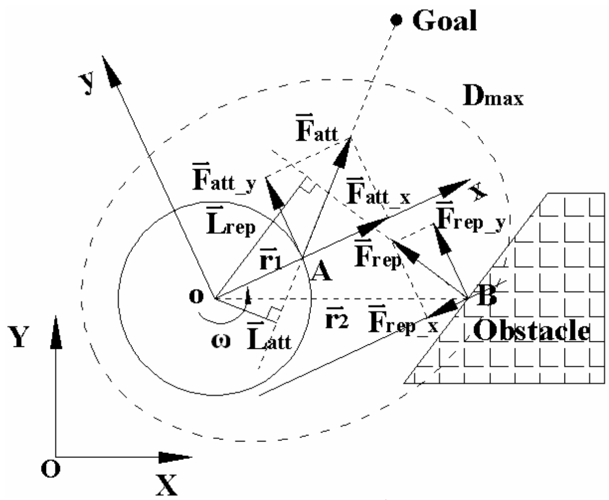

Fig. 2. Illustration of VSF^2 method

$\bar{F}_{repulsive\_robot}$ is the virtual repulsive force generated by this robot at the interaction point *(x, y)* in the 2-D space. The reaction force is a vector. Its direction of reaction force is defined to be the normal line of the interaction contour with respect to the interaction point.

D. VSF^2 Method

The Variable Speed Force Field method (VSF^2) is an approach using the concept of the F^2 method for multi-robot motion planning and collision avoidance. This section introduces the VSF^2 briefly. For more details, refer to [11].

Define a fixed reference frame *O (X, Y)* and a moving reference frame *o (x, y)* attached to the robot's body. Let $\dot{x}$, $\dot{y}$ be the longitudinal and lateral velocity of the robot in frame *o (x, y)*, the absolute velocities $\dot{X}$, $\dot{Y}$ in the fixed reference frame *O (X, Y)* are:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = R(\theta_r) \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{10}$$

Differentiation with respect the time gives

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \end{bmatrix} = R(\theta_r) \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{11}$$

where a_x, a_y are the absolute accelerations expressed in the moving frame *o*.

For a robot of mass *m* and inertia *I* about its centre of mass, equations of motions in the frame *o* are:

$$ma_x = \vec{F}_{att\_x} + \sum \vec{F}_{rep\_x} \tag{12}$$

$$ma_y = \vec{F}_{att\_y} + \sum \vec{F}_{rep\_y} \tag{13}$$

$$I\dot{\omega} = M_{att} + \sum M_{rep} \tag{14}$$

where $\vec{F}_{att\_x}$, $\vec{F}_{att\_y}$ are the components of the virtual attractive force along the longitudinal direction and the lateral direction respectively. $\vec{F}_{rep\_x}$, $\vec{F}_{rep\_y}$ are the components of the virtual repulsive forces along the longitudinal direction and the lateral direction respectively (see Fig. 2). ω is the angular velocity of this robot about its center of mass in frame *o*. M_{att} and M_{rep} are the moments generated by the virtual attractive force and repulsive force respectively. The calculations of forces and moment are defined as:

$$M_{att} = \vec{r}_1 \times \vec{F}_{att} \tag{15}$$

$$M_{rep} = \vec{r}_2 \times \vec{F}_{rep} \tag{16}$$

where $\vec{r}_1$ is the distance from the center of robot *o* to the attractive force ($\vec{F}_{att}$) acting point *A*. $\vec{r}_2$ is the distance from the center of robot to the repulsive force ($\vec{F}_{rep}$) acting point *B*. Please note that we assume the attractive force acts on the robot's front edge *A*, which is the interaction of *x* coordinate and a robot's body (see Fig.2).

It should be noted that a robot needs to satisfy the limitations of translational speed (v_r), translational acceleration ($\dot{v}_r$), angular speed (ω) and angular acceleration ($\dot{\omega}$). These constraints are:

$$-v_{max} \le v_r \le v_{max} \tag{17}$$

$$-a_{max} \le \dot{v}_r \le a_{max} \tag{18}$$

$$-\omega_{max} \le \omega \le \omega_{max} \tag{19}$$

$$-\dot{\omega}_{max} \le \dot{\omega} \le \dot{\omega}_{max} \tag{20}$$

These parameters should be chosen based on the robot's dynamic and kinematics characteristics.

E. A Simple Case

In this section, we use a simple case to highlight how the performance of the F^2 method is affected by parameters. Fig. 3 shows an area of $10m \times 10m$. A robot is supposed to travel from *(2, 3)* to *(8, 7)*. A circular obstacle is located at the center. Simulations are carried out using different parameters for the F^2 method. In Fig. 3, Paths 1 to 5 are simulation results using randomly selected parameters. Path 6 (shown in black solid line) is the result of optimized parameters using the approach to be presented in Section III.

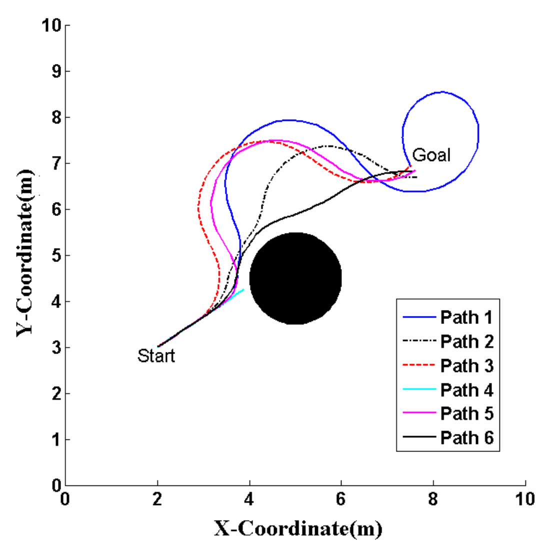

Fig. 3. Case 1: Paths resulted from different parameters

TABLE II
PARAMETERS IN CASE 1

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

The corresponding parameters are given in Table II. Please note that using the 4th group of parameters fail to achieve a solution. Collision occurs when the robot travels close to the obstacle. This is a case of unsuccessful parameter selection.

This example shows that the parameter selection is very important to the performance of the F^2 method. The use of PSO for parameter optimization is explained in detail in the following section.

III. PSO

In this section we will introduce the basic idea of PSO and then describe how to use it in the parameter optimization for the F^2 method.

A. *Particle Swarm Optimization*

In Particle Swarm Optimization (PSO), a problem space is covered with an initial population of random solutions in which they are guided to search for the optimum over a number of generations. The concept of PSO is that each particle randomly searches through the problem space by updating itself with its own memory and the social information gathered from other particles. An attractive feature of PSO is that its implementation is simple and effective.

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

$$v_{iz} = \varepsilon \times v_{iz} + c_1 \times rand_1() \times (P_{best} - x_{iz}) + c_2 \times rand_2() \times (G_{best} - x_{iz}) \quad (21)$$

$$x_{iz} = x_{iz} + v_{iz} \quad (22)$$

where ε is the inertia factor, c_1 and c_2 are two positive constants and *rand1()* and *rand2()* are two random functions in the range [0, 1].

B. *The Utilization of PSO for Parameter Optimization*

Parameters of the F^2 method included in optimization are: k, P, Q, C, ρ_0. The goal is to find these parameters which will minimize the total length of resulted paths.

The definition of the ith particle and its fitness value are:

$$x_i = [k\ P\ Q\ C\ \rho_0] \quad (23)$$

$$f(x_i) = \sum_{i=1}^{W} FF(x_i) \quad (24)$$

where $FF(x_i)$ is the resultant path length obtained from the F^2 method using parameters x_i. W is the number of robots.

When a robot is assigned a task, for example, traveling from start point (X_{start}, Y_{start}) to goal point (X_{goal}, Y_{goal}), the F^2 method will be called to drive this robot toward its destination while avoiding collision with obstacles and other robots. When a robot reaches its destination, $FF(x_i)$ will return the value of the real travel distance. If a robot fails to reach its destination using parameters x_i or collision occurs, $FF(x_i)$ will return with a very high value. The sum of all $FF(x_i)$ value 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 (21) and (22). This process continues until the user-defined stopping criteria are satisfied. In this paper, the stopping criterion is set to be 50 iterations, i.e. PSO returns the optimization result after running 50 iterations.

IV. SIMULATIONS

This paper focuses on multi-robot motion planning and collision avoidance. For all simulations in this paper, we assume that all robots are identical and their characteristics are selected based on those of the Amigo robot [17], which has a radius of 0.18(m) and a weight of 3.6(kg). The maximum speed is 0.75(m/s). Each robot has an assigned goal 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, etc. Task priorities of all robots are set to $T_p=1$, i.e., no robot has priority over other robots during collision avoidance.

We start from a simplest case, that is, only one robot and one static circular obstacle are in the working place. Then a case containing two robots passing a corridor is studied. The last simulation is with four robots navigating in an indoor environment.

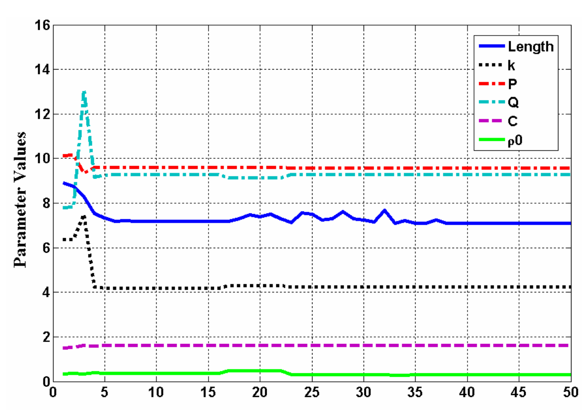

Fig. 4. Case 1: Optimization results

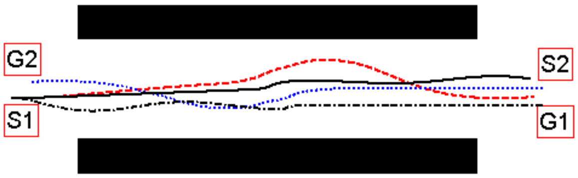

Fig. 5. Case 2: Two robots in a corridor

A. Case 1: Single Robot and One Obstacle

In Fig. 3, a robot is supposed to travel from *(2, 3)* to *(8, 7)*. The obstacle is located at the center of the work space. Simulations are carried out using different parameters. Paths 1 to 5 are simulation results using randomly selected parameters. Path 6 (shown in black solid line) is the result of optimized parameters using the approach. The corresponding parameters are given in Table II.

To start, we first set ranges for parameters. Based on past experience, we set: $k \in (2,10)$, $P \in (5,20)$, $Q \in (5,20)$, $C \in (1.5,3)$, $\rho_0 \in (0.2,1)$.

After running PSO for 50 iterations, we obtained the results shown in Fig. 4. In Fig. 4, the blue solid line shows the resultant path length of each iteration. The values of k, P, Q, C, ρ_0 in each iteration are shown by the black dashed line, red dashdot line, cyan dashdot line, magenta dashed line and green solid line, respectively. In this case, PSO reaches a stable state after 39 iterations. The path length using optimized parameters is *7.106m*, which is much shorter than those using parameters listed in rows 1 to 5 in Table II.

B. Case 2: Two Robots Navigating in a Corridor

In this section, we use a case studied in [11]. In Fig. 5 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 distance between two walls (black patches) is *1.5m*. Previous results in [11] are listed in row 1st in Table III. The path of robot 1 is denoted by the red dashed line and the path of robot 2 is denoted by the blue dotted line.

After running PSO for 50 iterations, we have the results shown in Fig. 6. The blue 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 black dashed line, red dashdot line, cyan dashdot line, magenta dashed line and green solid line respectively.

Please note that the solution is not unique. Rows 2 to 6 in Table III lists the results obtained using the approach presented in this paper. Fig. 5 shows the paths using parameters listed in the 6th row. Paths of robot 1 and 2 are denoted by the black solid line and the black dashdot line, respectively.

In the PSO searching process, obtained parameters may exceed the pre-set ranges. For example, the P values of the 3rd, 4th, 5th iteration are *35.361m*, *22.096m* and *28.800m* respectively and are out of range of [5, 20], which indicated that PSO is able to operate irrespective of incorrectly-determined initial parameters.

TABLE III
PARAMETERS IN CASE 2

	Length (m)	k	P	Q	C	ρ_0	Im-prove-ment
1	16.402	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	15.749	2.971	11.838	16.724	2.577	0.354	3.98%

C. Case 3: Four Robots in an Indoor Environment

In this section, we use the four robot navigation case used in [11]. Fig. 7 shows an indoor environment with some static obstacles. The dimension of this area is $20m \times 20m$. Robot 1 is supposed to travel from *(18, 1)* to *(7, 18)*. Robot 2 is supposed to travel from *(2, 18)* to *(18, 4)*. Robot 3 is supposed to travel from *(1, 4)* to *(17, 12)* and robot 4 is supposed to travel from *(18, 18)* to *(7, 2)*.

The parameters used in [11] and the resultant path length are listed in row 1 of Table IV. Row 2 to 6 show the parameters and path lengths obtained by the PSO approach presented in Section III. In Fig. 7, the paths of the four robots obtained using parameters in row 2 of Table III are shown by the red dotted line, black solid line, green dashed line and cyan dashdot line, respectively. The corresponding path lengths for robots 1, 2, 3 and 4 are *21.023m*, *23.484m*, *18.506m* and *19.399m*. Compared with the previous result in [11], 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.

D. Discussion

The PSO-tuned F^2 method is tested in a number of environments in this section. Simulation results show that this approach is capable of finding appropriate parameters for the F^2 method. The resultant path lengths are shorter than those obtained in our previous work [11].

In this paper, the stopping criterion of PSO is set to be 50 iterations. The approach in this paper undertook a single-objective optimization with the goal of finding parameters that minimize the total length of resultant paths. In the above cases, there are some oscillations in the robots' movements.

V. CONCLUSIONS AND FUTURE WORK

This paper has presented an approach using PSO to optimize parameters for the F^2 method. First, a simple case is studied to show the importance of parameter selection in

TABLE IV
PARAMETERS IN CASE 3

	Length (m)	k	P	Q	C	ρ_0	Im-prove-ment
1	89.061	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%

the F^2 method. We then described how to use PSO for parameter optimization. The approach was tested in various environments, and show that this algorithm is capable of finding much improved parameters for the F^2 method.

In the future we will work with multi-objective optimization models. Other factors, such as success rate, task execution time, task priority, will be taken into consideration.

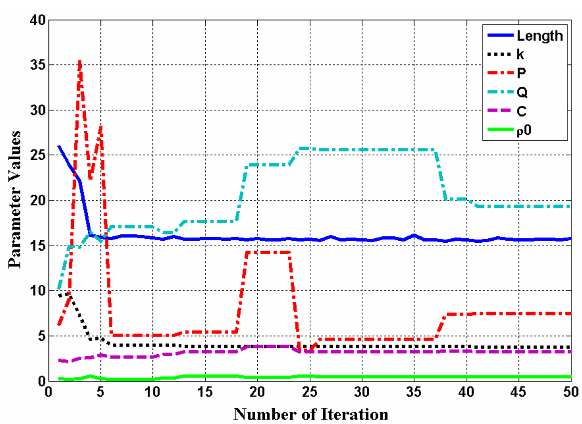

Fig. 6. Case 2: Optimization results

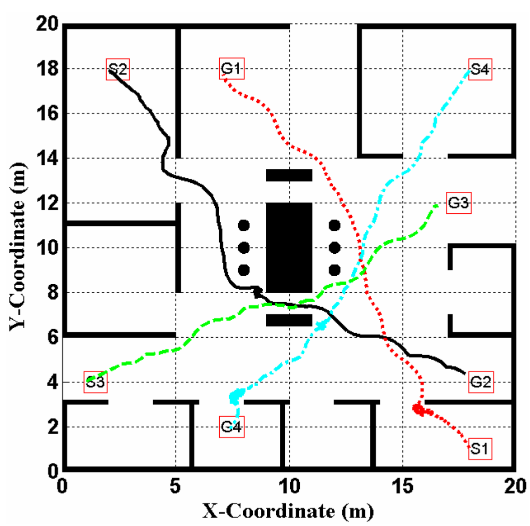

Fig. 7. Case 3: Four robots navigation

ACKNOWLEDGMENT

This work is supported in part by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government, Australia.

REFERENCE

[1] J.-C. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, Boston, 1991.

[2] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," in *Proceeding of IEEE International Conference on Robotics and Automation*, 1986, pp. 1419-1424, vol. 3.

[3] S. J. Buckley, "Fast motion planning for multiple moving robots," in *Proceeding of IEEE International Conference on Robotics and Automation*, 1989, pp. 322-326, vol.1.

[4] J. P. van den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 430-435.

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

[6] C. W. Warren, "Multiple robot path coordination using artificial potential fields," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1990, pp. 500-505, vol.1.

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

[8] 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.

[9] P. A. O'Donnell and T. Lozano-Periz, "Deadlock-free and collision-free coordination of two robot manipulators," in *Proceeding of IEEE International Conference on Robotics and Automation*, 1989, pp. 484-489, vol.1.

[10] G. Yi and L. E. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," in *Proceeding of IEEE International Conference on Robotics and Automation*, 2002, pp. 2612-2619, vol. 3.

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

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, Perth, WA, 1995, pp. 1942-1948, vol.4.

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

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

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

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

[17] Mobile Robot Bases Specifications, ActivMedia Robotics, http://www.activrobots.com/ROBOTS/specs.html