# Smooth Formation Navigation of Multiple Mobile Robots for Avoiding Moving Obstacles

**Xin Chen and Yangmin Li***

**Abstract:** This paper addresses a formation navigation issue for a group of mobile robots passing through an environment with either static or moving obstacles meanwhile keeping a fixed formation shape. Based on Lyapunov function and graph theory, a NN formation control is proposed, which guarantees to maintain a formation if the formation pattern is $C^k$, $k \geq 1$. In the process of navigation, the leader can generate a proper trajectory to lead formation and avoid moving obstacles according to the obtained information. An evolutionary computational technique using particle swarm optimization (PSO) is proposed for motion planning so that the formation is kept as $C^1$ function. The simulation results demonstrate that this algorithm is effective and the experimental studies validate the formation ability of the multiple mobile robots system.

**Keywords:** Adaptive NN, formation navigation, interaction topology, particle swarm optimization.

## 1. INTRODUCTION

Formation navigation can be observed in spacecraft formation flying, robotic vehicles formation moving, and mobile robots formation surveying. Formation can be understood as a kind of information consensus in which agents (robots) interact with each other using various sensors and communication techniques. In order to understand the relationship between individual robots, graph theory is often used for the description of these interactions [1,2]. Since the formation issue of multiple mobile robots is also viewed as a distributed control problem, system stability theory such as Lyapunov method can be used effectively to analyze system performance of the formation navigation [3-6].

In this paper, in order to understand the internal structure of a formation well, the formation pattern and the interactive relations among robots must be defined clearly. The formation pattern is described by a matrix in which every entry describes relative distances between robots. The interactions among robots can be described by an adjacency matrix based on graph theory. Combining these two matrices, a unique matrix is constructed to describe the structure of a formation. Each entry in the matrix is viewed as a moving point, then one certain robot is controlled to follow this moving point. Based on Lyapunov method, it is proved that even if some parameters of an individual model are unknown, in the case of perturbations existed, neural network (NN) control can enable robots to achieve regular formation with fixed or dynamic formation patterns [7,8].

After solving individual robotic control, a formation navigation technique is applied to accomplish obstacle avoidance while keeping fixed formation shape. Relative to the real-time reactive way, such as artificial potential method, in which the motion of robots is controlled by artificial force calculated on real-time [9-11], the motion planning, namely path planning, is more convenient for evaluating paths ahead of robots moving, because it describes paths in the form of smooth splines [12], so that the paths generated are predictable. Due to the path is described in the form of high order polynomial, the computations referred in the analytical motion planning are complex and even unsolvable. To decrease computational burden, we use an evolutionary computational technique in terms of particle swarm optimization (PSO) to achieve motion planning [13-15]. Different from other evolutionary computations in which desired paths are expressed as nonsmooth ones [16,17], the PSO method can generate smooth trajectories so that the adaptive NN control strategy can be applied to control a group of

Manuscript received September 29, 2005; revised March 4, 2006; accepted April 26, 2006. Recommended by Editor Zengqi Sun. This work was supported by the Research Committee of University of Macau under grant RG066/02-03S/LYM/FST.

Xin Chen and Yangmin Li are with the Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Av. Padre Tomás Pereira S.J., Taipa, Macao S.A.R., P. R. China (e-mails: {ya27407, ymli}@umac.mo).

* Corresponding author.

robots to follow the smooth trajectories while keeping formation pattern.

This paper is organized as follows with the following section presenting the formation description based on formation pattern and interaction topology. Section 3 and Section 4 analyze the control strategy and its stability. Formation navigation with obstacle avoidance ability is studied in Section 5. Simulation is performed in Section 6. Experimental studies are conducted in Section 7. Conclusions are given finally in Section 8.

## 2. DESCRIPTION OF FORMATION

Given a group of robots labeled by $R_i$ $(i=1,2,\cdots,N)$, where $R_l$ denotes the leader of the formation. Let $P^d=\left[p_1^d \ \ p_2^d \ \ \cdots \ \ p_N^d\right]^T$ be the desired positions of all robots relative to $R_l$, where $p_l^d=0$. Let $P=\left[p_1 \ \ p_2 \ \ \cdots \ \ p_N\right]^T$ be the practical positions of robots.

Formation pattern describes the relative positions among robots over time, which is represented by a relative matrix defined as $D^d(t)=\left\{D_{ij}^d(t)\right\}_{N\times N}$, where $D_{ij}^d(t)=p_i^d(t)-p_j^d(t)$ with $D_{ii}^d(t)=0$. In response to $D^d(t)$, let $D(t)=\left\{D_{ij}(t)\right\}_{N\times N}$ denote the practical distances between robots where $D_{ij}(t)=p_i(t)-p_j(t)$. Obviously for leader-followers formation, $D^d(t)$ should be broadcasted by $R_l$ to other members.

Six robots are illustrated to form a rectangle formation as shown in Fig. 1. If projecting their positions onto a plane whose frame orientation is denoted by $X/Y$, a formation pattern is generated, whose coordinates in X-direction are shown in the matrix listed on the bottom of this page.

If a robot has known its desired position in a formation, it needs to detect his neighborhood to move to the desired position. That means it should interact with other members. A directed adjacency graph $\mathcal{G}$ is exploited to describe the interaction among robots which consists of a set of vertices (nodes) $V$ and a set of arcs $A$, where $a(v,w)\in A$ and $v,w\in V$. It is pointed from $v$ to $w$. In this paper arc $a(v,w)$ represents that $R_v$ takes $R_w$ as a reference object to decide the relative position. Since the reference position of a robot may be determined by several neighboring robots, if we let $G=\left\{g_{ij}\right\}_{N\times N}$ be adjacency matrix associated with graph $\mathcal{G}$, $g_{ij}$ is defined as a weight to denote the influence factor of $R_j$ to $R_i$ in terms of the reference position. The values of $g_{ij}$ satisfy the following property:

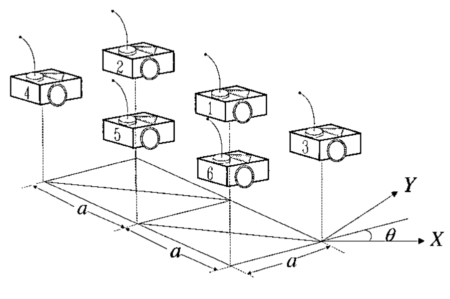
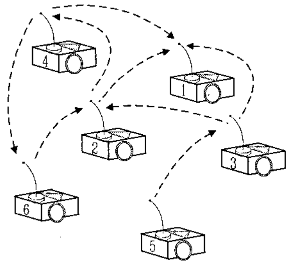


Fig. 1. A formation pattern including six robots.



$$G=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 2. An example of interaction topology among six robots.

$$\sum_{j=1}^{N} g_{ij}=1. \tag{1}$$

The interaction topology and adjacency matrix are illustrated in Fig. 2. If $R_1$ plays the role of the leader of the formation, then $g_{11}=1$ holds and other diagonal entries of $G$ are zero, ie. $g_{ii}=0, \quad (i=2,\cdots,N)$. Normally for any one robot, all of its leaders have the

$$D_x^d=\begin{bmatrix} 0 & a\cos\left(\theta-\frac{\pi}{2}\right) & a\cos\left(\theta+\frac{\pi}{2}\right) & \sqrt{2}a\cos\left(\theta-\frac{\pi}{4}\right) & a\cos(\theta) & \sqrt{2}a\cos\left(\theta+\frac{\pi}{4}\right) \\ a\cos\left(\theta+\frac{\pi}{2}\right) & 0 & 2a\cos\left(\theta+\frac{\pi}{2}\right) & a\cos(\theta) & \sqrt{2}a\cos\left(\theta+\frac{\pi}{4}\right) & \sqrt{5}a\cos(\theta+\arctan(2)) \\ a\cos\left(\theta-\frac{\pi}{2}\right) & 2a\cos\left(\theta-\frac{\pi}{2}\right) & 0 & \sqrt{5}a\cos(\theta-\arctan(2)) & \sqrt{2}a\cos\left(\theta-\frac{\pi}{4}\right) & a\cos(\theta) \\ \sqrt{2}a\cos\left(\theta+\frac{3\pi}{4}\right) & -a\cos(\theta) & -\sqrt{5}a\cos(\theta-\arctan(2)) & 0 & a\cos\left(\theta+\frac{\pi}{2}\right) & 2a\cos\left(\theta+\frac{\pi}{2}\right) \\ a\cos(\theta+\pi) & \sqrt{2}a\cos\left(\theta-\frac{3\pi}{4}\right) & \sqrt{2}a\cos\left(\theta+\frac{3\pi}{4}\right) & a\cos\left(\theta-\frac{\pi}{2}\right) & 0 & a\cos\left(\theta+\frac{\pi}{2}\right) \\ \sqrt{2}a\cos\left(\theta-\frac{3\pi}{4}\right) & -\sqrt{5}a\cos(\theta+\arctan(2)) & a\cos(\theta+\pi) & 2a\cos\left(\theta-\frac{\pi}{2}\right) & a\cos\left(\theta-\frac{\pi}{2}\right) & 0 \end{bmatrix}$$

same effect on determining its reference point. For example since $R_2$ has two leaders $R_1$ and $R_4$, we can obtain $g_{21} = g_{24} = 0.5$.

Let $E = \{e_j\}_{N\times 1} = \left(G \circ D - G \circ D^d\right) 1_{N\times 1}$ be a relative error vector, where the operator '$\circ$' refers to a Hadamard product. If let a matrix $H$ be defined as $H = I - G$, the relative error is expressed as

$$E = HP - G \circ D^d 1_{N\times 1} = P - \left(GP + G \circ D^d 1_{N\times 1}\right). \quad (2)$$

Due to the property of $G$, it holds that $\sum_{j=1}^{N} h_{ij} = 0$, and $h_{lj} = 0$, $(j = 1, 2, \cdots, N)$. Since $R_l$ is the formation leader, it holds $g_{ll} = 1$, while $g_{lj} = 0$, $(j \neq l)$, and $D_{ll}^d = 0$. Therefore the $l$-th element of the relative error, $e_l$, equals zero. It's reasonable because the adjacency graph is built relative to the leader, the relative error of $R_l$ should be zero.

The following lemmas are introduced:

**Lemma 1:** Given a nonnegative matrix $G$ satisfying the property of stochastic matrix, the matrix $H = I - G$ has at least one zero eigenvalue and all of the non-zero eigenvalues are in the open right half plane, and $\rho(H) \leq 1$. Furthermore $H$ has one zero eigenvalue and the kernel of $H$ is $span\{1\}$ if and only if the direct graph associated with $G$ has a spanning tree.

**Lemma 2:** Given an error shown in (2), if the adjacency matrix $G$ is connected, all robots will follow the leader of the formation and form a formation in case of $E = 0$.

**Proof:** Let $R_l$ be the formation leader. If we obtain $P^d$ by solving equation $HP^d = G \circ D^d 1_{N\times 1}$, then $E = H\left(P - P^d\right)$.

Two facts are given below: 1) Because $R_l$ is the leader of the formation, the entries in the $l$-th row of $H$ are zero, 2) Since $D_{ii} = 0$, the $l$-th element of $G \circ D^d 1_{N\times 1}$ is zero too. Since the equation $p_l^d = 0$ holds, the equation can be reduced to

$$\bar{H}\bar{P}^d = \bar{G} \circ \bar{D}^d 1_{(N-1)\times 1}, \quad (3)$$

where $\bar{P}^d = \left[p_1^d \ \cdots \ p_{l-1}^d \ p_{l+1}^d \ \cdots \ p_N^d\right]^T$, $\bar{H}$ is the submatrix of $H$ resulting from taking off the $l$th row and the $l$th column of $H$, $\bar{G}$ and $\bar{D}$ are the submatrices of $G$ and $D$ resulting from taking off the $l$th rows of them respectively. Obviously there is only one nonzero solution for (3). Hence the desired relative position $P^d$ is unique.

If $E = 0$, we have $H\left(P - P^d\right) = 0$. According to Lemma 1, we have $\left(P - P^d\right) \in span\{1\}$. That means all elements in vector $P - P^d$ are the same. Due to $p_l^d = 0$, we have $p_i = p_l + p_i^d$, $(i = 1, \cdots, N)$. That means all members reach the desired positions described by $D^d(t)$, in other words, formation is formed.

Lemma 2 suggests a way to realize formation control. Since $D^d$ is the information broadcasted by the formation leader and $G$ is determined by the practical interaction among robots, $GP + G \circ D^d 1_{N\times 1}$ in (2) can be measured on real-time. Hence every robot can control itself by decentralized control method to follow reference points determined by $GP + G \circ D^d 1_{N\times 1}$ so that conditions of Lemma 2 are satisfied, and formation is formed.

## 3. INDIVIDUAL CONTROL STRATEGY

3.1. Dynamic description for individual robot

A kind of two-wheel car-like mobile robot shown in Fig. 3 is exploited to achieve the formation task. If we take the center of mass as the robot's position, the dynamic equation for such individual robot is expressed as

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + \tau_d = J^T(q)\lambda + B(q)\tau, \quad (4)$$

where $q = [p_x \ \ p_y \ \ \theta]^T$ represents general coordinates, $\tau_d$ denotes bounded disturbance and unmodeled dynamics, other matrices referred in the equation are given by

$$M(q) = \begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I_0 + md^2 \end{bmatrix},$$

$$\tau = \begin{bmatrix} \tau_l & \tau_r \end{bmatrix}^T, V(q,\dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix},$$

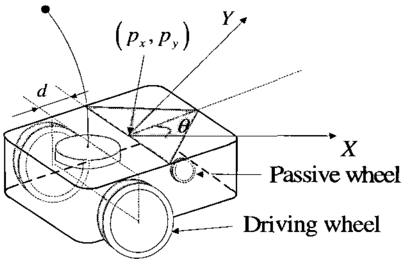


Fig. 3. A two-wheel-driven mobile robot.

$$B = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix},\ J(q) = \begin{bmatrix} \sin\theta & -\cos\theta & d \end{bmatrix}.$$

The nonholonomic constraint is expressed as $J(q)\dot{q} = 0$ or $\dot{p}_x \sin\theta - \dot{p}_y \cos\theta + d\dot{\theta} = 0$. Then the second order derivative of $\theta$ is expressed as:

$$\begin{aligned}\ddot{\theta} &= \frac{1}{d}(\ddot{p}_y \cos\theta - \ddot{p}_x \sin\theta) + \frac{1}{2d^2}(\dot{p}_x^2 - \dot{p}_y^2)\sin 2\theta \\ &\quad - \frac{1}{d^2}\dot{p}_x \dot{p}_y(\cos^2\theta - \sin^2\theta).\end{aligned} \tag{5}$$

A full rank matrix $S(q)$ is formed by the vectors spanning the null space of constraint matrix $J(q)$ such that $S^T(q)J^T(q)$=0, where $S(q) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix}$. Multiplying both sides of (4) by $S^T$ to eliminate nonholonomic constraint forces $\lambda$, yields

$$S^T M(q)\ddot{q} + S^T V(q,\dot{q})\dot{q} + \bar{\tau}_d = S^T B(q)\tau, \tag{6}$$

where $\bar{\tau}_d = S^T \tau_d$. For a formation task, the key point is how to keep relative positions among robots, so using (5) we can get a reduced system in which $\theta$ is enclosed into the system's parameters. That means if define $p = [p_x \quad p_y]^T$, $T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$, and $M_0 = \begin{bmatrix} m & 0 \\ 0 & \frac{I}{d} \end{bmatrix}$, (6) can be transformed to

$$M_0 T\ddot{p} + M_0 \dot{T}\dot{p} = S^T B\tau - \bar{\tau}_d. \tag{7}$$

3.2. Neural network controller

From Lemma 2 we know that if we take $P^d = GP + G \circ D^d \cdot 1_{N\times 1}$ as local reference position, when (2) converges to zero, the formation is formed. If $P^d$ is assigned to individual robots, the ideal reference point of $R_i$ is in the form of $p_i^d = \sum_{j=1}^{N} g_{ij}(p_j + D_{ij}^d)$. Therefore we can define the individual relative position error as

$$e_i = p_i - p_i^d = p_i - \sum_{j=1}^{N} g_{ij}(p_j + D_{ij}^d). \tag{8}$$

A filtered error is defined as $z_i = \dot{e}_i + \Lambda e_i$. If a temporal variable is defined as $\dot{p}_i^r = \dot{p}_i^d - \Lambda e_i$, then $z_i = \dot{p}_i - \dot{p}_i^r$. Substitute it into (7), and let $\tilde{z}_i = T_i z_i$, thus

$$M_0 \dot{\tilde{z}}_i = S^T B\tau_i - \bar{M}_i \ddot{p}_i^r - \bar{V}_i \dot{p}_i^r - \bar{\tau}_{id}, \tag{9}$$

where $\bar{M}_i = \begin{bmatrix} m_i & 0 \\ 0 & \frac{I_i}{d_i} \end{bmatrix}\begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}$, $\bar{V}_i = \frac{I_i}{d_i}(\dot{x}\sin\theta_i - \dot{y}\cos\theta_i)\begin{bmatrix} m_i & 0 \\ 0 & \frac{I_i}{d_i} \end{bmatrix}\begin{bmatrix} \sin\theta_i & -\cos\theta_i \\ \cos\theta_i & \sin\theta_i \end{bmatrix}$.

In practical situations, the values of matrices above are always not measured accurately. So a neural network is used to model the item $\bar{M}_i \ddot{p}_i^r + \bar{V}_i \dot{p}_i^r$ online.

A nonlinear function is defined as $f(X_i) = \bar{M}_i \ddot{p}_i^r + \bar{V}_i \dot{p}_i^r$ in which $X_i = \begin{bmatrix} \dot{p}_i & \sin\theta_i & \cos\theta_i & \ddot{p}_i^r & \dot{p}_i^r \end{bmatrix}^T$ who satisfies the following inequality:

$$\|X_i\| \le c_1 Q^d + c_2\|\tilde{z}_i\| + c_3, \tag{10}$$

where $c_1$ to $c_3$ are positive scalars, $Q^d$ is the bound satisfying

$$\left\| \begin{matrix} \ddot{p}_i^d \\ \dot{p}_i^d \\ p_i^d \end{matrix} \right\| \le Q^d. \tag{11}$$

To simplify the expression, the subscript $i$ is omitted. It is supposed that there exists a two-layer feedforward NN as shown in Fig. 4, which can approximate $f(X)$.

$$f(X) = W^T \sigma(V^T X) + \varepsilon, \tag{12}$$

where $V \in R^{N_I \times N_H}$ represent the input-to-hidden-

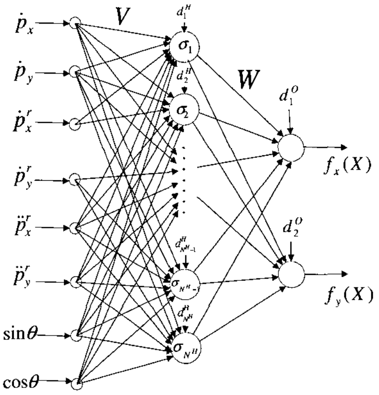


Fig. 4. A two-layer feedforward neural network.

layer interconnection weights; $W \in R^{N_H \times N_O}$ represent the hidden-layer-to-outputs interconnection weights; $N_H$, $N_I$ and $N_O$ are the numbers of neurons in the hidden layer, the input layer, and the output layer respectively. The activation function $\sigma(\cdot)$ is in the form of $\sigma(x) = \frac{1}{1+e^{-x}}$, $\varepsilon$ is the NN functional approximation error.

A NN function $\hat{f}(X)$ is constructed to estimate $f(X)$ on-line, which can be written as

$$\hat{f}(X) = \hat{W}^T \sigma(\hat{V}^T X), \tag{13}$$

where $\hat{W}$ and $\hat{V}$ are estimators of NN weights.

The estimated errors are defined as $\tilde{f} = f - \hat{f}$, $\tilde{W} = W - \hat{W}$, and $\tilde{V} = V - \hat{V}$. The hidden-layer output error is defined as $\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T X) - \sigma(\hat{V}^T X)$. Applying Taylor series expansion, we can obtain

$$\sigma(V^T X) = \sigma(\hat{V}^T X) + \sigma'(\hat{V}^T X)\tilde{V}^T X + O(\tilde{V}^T X)^2, \tag{14}$$

where $\sigma'(\hat{y}) = \left.\frac{\partial \sigma(y)}{\partial y}\right|_{y=\hat{y}}$. Therefore,

$$\tilde{\sigma} = \sigma'(\hat{V}^T X)\tilde{V}^T X + O(\tilde{V}^T X)^2, \tag{15}$$

where $O(\tilde{V}^T X)^2$ is a term with order two, which satisfies the following property:

**Property 1:**

$$\begin{aligned} \left\| O(\tilde{V}^T X)^2 \right\| &\le c_4 + c_5 Q^d \left\|\tilde{V}\right\|_F + c_6 \left\|\tilde{V}\right\|_F \|\tilde{z}\| + c_7 \left\|\tilde{V}\right\|_F \\ &\le c_4 + c_8 \left\|\tilde{V}\right\|_F + c_6 \left\|\tilde{V}\right\|_F \|\tilde{z}\|, \end{aligned} \tag{16}$$

where $c_4$ to $c_8$ are positive scalars, $c_8 = c_5 Q^d + c_7$.

Substituting the approximated $f(X)$ into (9), we have

$$\begin{aligned} M_0 \dot{\tilde{z}} &= S^T B \tau - f(X) - \bar{\tau}_d \\ &= S^T B \tau - W^T \sigma(V^T X) - \bar{\tau}_d + \varepsilon. \end{aligned} \tag{17}$$

The input-output feedback linearization control technology and adaptive backpropagation learning algorithm are applied to stabilize individual robot system, which can be expressed as

$$\tau = (S^T B)^{-1} (\hat{W}^T \sigma (\hat{V}^T X) - K\tilde{z} + \gamma), \tag{18}$$

$$\begin{aligned} \dot{\hat{W}} &= F \hat{\sigma}' \hat{V}^T X \tilde{z}^T - F \hat{\sigma} \tilde{z}^T - \kappa F \|\tilde{z}\| \hat{W} \\ \dot{\hat{V}} &= -UX(\hat{\sigma}'^T \hat{W} \tilde{z})^T - \kappa U \|\tilde{z}\| \hat{V}, \end{aligned} \tag{19}$$

where $K = \text{diag}\{k_1, k_2\}$, in which $k_1$ and $k_2$ are positive scalars; $\gamma$ is a robust control term to suppress the disturbance of unmodeled structure of dynamics $\tau_d$ and functional approximation error of NN $\varepsilon$ ; $F$ and $U$ are positive definite design parameter matrices governing the speed of learning.

Substitute control law into (17), and let $\sigma$ and $\hat{\sigma}$ be $\sigma(V^T X)$ and $\sigma(\hat{V}^T X)$ respectively, thus

$$M_0 \dot{\tilde{z}} = -K\tilde{z} - W^T \sigma + \hat{W}^T \hat{\sigma} + \gamma - \bar{\tau}_d + \varepsilon. \tag{20}$$

Adding and subtracting $W^T \hat{\sigma}$ and $\hat{W}^T \tilde{\sigma}$ , and considering (14) and (15), we have

$$M_0 \dot{\tilde{z}} = -K\tilde{z} - \tilde{W}^T (\hat{\sigma} - \hat{\sigma}' \hat{V}^T X) - \hat{W}^T \hat{\sigma}' \tilde{V}^T X + s + \gamma, \tag{21}$$

where $s(t)$ is a disturbance term in terms of

$$s(t) = -\tilde{W}^T \hat{\sigma}' V^T X - W^T O(\tilde{V}^T X)^2 - \bar{\tau}_d + \varepsilon. \tag{22}$$

## 4. STABILITY OF FORMATION CONTROL

The desired formation shape is determined by the formation pattern $D^d(t)$ and interaction matrix $G$. In practice, due to the different requirements of formation tasks, formation pattern and interaction topology may be either invariant or variant.

### 4.1. Stability under variant formation pattern and invariant interaction topology

When a formation with a kind of formation shape is passing a field with obstacles, two actions may happen to avoid obstacles:

1) A new formation pattern ($D^d(t)$) is generated to make the formation avoid obstacles.
2) A new desired path for the formation leader, $R_l$, is generated, so that $R_l$ follows this new path to lead the whole formation to pass through obstacles. This change induces change of formation pattern $D^d(t)$.

For these two changes about formation pattern, the following theorem represents the preconditions under which the control strategy ensures the convergence of the system.

**Theorem 1:** For a multi-robot system which has a predetermined leader, if formation pattern $D^d(t)$ is $C^k$ function with $k \ge 1$, and adjacency matrix associated with interaction graph is connected and invariant, the robots must converge to the formation pattern following up the individual control strategy represented in (18) and (19).

**Proof:** An important assumption is that $D^d(t)$ is at least $C^1$. Then $\dot{D}^d(t)$ is Lipschitz continuous.

The dynamic equations of the system consist of (19) and (21). According to the definition of sigmoid function, *it holds that* $\forall x \in R$, $\sigma(x) \in [0,1]$. *And its* derivative satisfies $\forall x \in R$, $\sigma'(x) \in [0, \frac{1}{4}]$. All parameters such as $F$, $U$, $\kappa$, $\overline{\tau}_d$, $\varepsilon$ are bounded. Therefore to make the dynamic functions (19) and (21) be locally bounded, the key precondition is that $X_i = \begin{bmatrix} \dot{p}_i & \sin\theta_i & \cos\theta_i & \ddot{p}_i^r & \dot{p}_i^r \end{bmatrix}^T$ is bounded. According to the definition of $p_i^d$, the following formula holds for robot $i$ :

$$X_i \leq c_9 \overline{Q}_i^d + c_{10}\, \tilde{z}_i + c_{11} + \sum_{j=1}^{N} g_{ij} \left( p_j + \dot{p}_j + \ddot{p}_j \right), \quad (23)$$

where $c_9$ to $c_{11}$ are positive scalars, and $\overline{Q}_i^d$ satisfies

$$\left\| \begin{matrix} \sum_{j=1}^{N} g_{ij} D_{ij}^d \\ \sum_{j=1}^{N} g_{ij} \dot{D}_{ij}^d \\ \sum_{j=1}^{N} g_{ij} \ddot{D}_{ij}^d \end{matrix} \right\| \leq \overline{Q}_i^d . \quad (24)$$

Obviously, to make $X_i$ be locally bounded, all positions of robots $p_i$ should be at least $C^1$.

To simplify expression, the subscript $i$ is omitted in the following proof. Assume that $X$ is bounded, then (19) and (21) are locally bounded functions. According to the properties of integral, for a finite interval of time [0 $t$], $t < \infty$, $\tilde{z}(t)$, $\hat{W}(t)$ and $\hat{V}(t)$ are Lipschitz continuous. From the definition of the filter error $\tilde{z}$, we know the position of robot $p(t)$ is smooth continuous. Considering all robots, it is concluded that if initial coordinate $p(0)$ is finite, after a finite interval $[0, t]$, $X(t)$ is bounded.

Therefore (19) and (21) are locally bounded functions. And both equations can be expressed in Filippov sense [18]. Since $\tilde{z}$, $\hat{W}$ and $\hat{V}$ are Lipschitz continuous, the following expression holds.

$$\begin{aligned} \dot{\hat{W}} &\in \mathcal{K}\left[ F\hat{\sigma}'\hat{V}^T X \tilde{z}^T - F\hat{\sigma}\tilde{z}^T - \kappa F \|\tilde{z}\| \hat{W} \right] \\ &= F \cdot \mathcal{K}\left[ \hat{\sigma}'\hat{V}^T X \tilde{z}^T - F\hat{\sigma}\tilde{z}^T \right] - \kappa F \|\tilde{z}\| \hat{W}, \end{aligned} \quad (25)$$

where $\mathcal{K}[f](\tilde{z})$ is defined as

$$\mathcal{K}[f](x,t) \equiv \bigcap_{\delta > 0} \bigcap_{\mu(N)=0} \overline{co} f(B(\tilde{z}, \delta) - N), \quad (26)$$

where $\bigcap_{\mu(N)=0}$ denotes the intersection over all set of Lebesgue measure zero. Similarly,

$$\dot{\hat{V}} \in -U \cdot \mathcal{K}\left[ X(\hat{\sigma}'^T \hat{W}\tilde{z})^T \right] - \kappa U \|\tilde{z}\| \hat{V}, \quad (27)$$

$$\begin{aligned} \dot{\tilde{z}} \in -M_0^{-1} K\tilde{z} - M_0^{-1} \cdot \mathcal{K}\Big[ &\tilde{W}^T(\hat{\sigma} - \hat{\sigma}'\hat{V}^T X) \\ &- \hat{W}^T \hat{\sigma}' \tilde{V}^T X + s \Big] + M_0^{-1}\gamma . \end{aligned} \quad (28)$$

A Lyapunov candidate is defined as

$$L = \tfrac{1}{2}\left[ \tilde{z}^T M_0 \tilde{z} + tr\left\{ \tilde{W}^T F^{-1} \tilde{W} \right\} + tr\left\{ \tilde{V}^T U^{-1} \tilde{V} \right\} \right]. (29)$$

Obviously $L(0) = 0$.

Using Lyapunov theorem on nonsmooth system [19], we have the derivative of Lyapunov function

$$\begin{aligned} \dot{L} &= \tilde{z}^T M_0 \dot{\tilde{z}} + tr\left\{ \tilde{W}^T F^{-1} \dot{\tilde{W}} \right\} + tr\left\{ \tilde{V}^T U^{-1} \dot{\tilde{V}} \right\} \\ &\subset -\tilde{z}^T M_0^{-1} K\tilde{z} + tr\left\{ \tilde{W}\left( F^{-1}\dot{\tilde{W}} - \mathcal{K}\left[ \hat{\sigma}\tilde{z}^T - \hat{\sigma}'\hat{V}^T X \tilde{z}^T \right] \right) \right\} \\ &\quad + tr\left\{ \tilde{V}^T \left( U^{-1}\dot{\tilde{V}} - \mathcal{K}\left[ X\tilde{z}^T \hat{W}^T \hat{\sigma}' \right] \right) \right\} + \tilde{z}^T(\mathcal{K}[s] + \gamma) \\ &\subset -\tilde{z}^T M_0^{-1} K\tilde{z} + tr\Big\{ \tilde{W}\Big( \mathcal{K}\left[ \hat{\sigma}\tilde{z}^T - \hat{\sigma}'\hat{V}^T X \tilde{z}^T \right] \\ &\quad - \mathcal{K}\left[ \hat{\sigma}\tilde{z}^T - \hat{\sigma}'\hat{V}^T X \tilde{z}^T \right] \Big) \Big\} \\ &\quad + tr\left\{ \tilde{V}^T \left( \mathcal{K}\left[ X(\hat{\sigma}'^T \hat{W}\tilde{z})^T \right] - \mathcal{K}\left[ X(\hat{\sigma}'^T \hat{W}\tilde{z})^T \right] \right) \right\} \\ &\quad + \kappa F \|\tilde{z}\| tr\left\{ \tilde{W}^T \hat{W} \right\} + \kappa U \|\tilde{z}\| tr\left\{ \tilde{V}^T \hat{V} \right\} + \tilde{z}^T(\mathcal{K}[s] + \gamma) \\ &= -\tilde{z}^T K\tilde{z} + \kappa \|\tilde{z}\| tr\left\{ \tilde{W}^T \left( W - \tilde{W} \right) \right\} \\ &\quad - \kappa \|\tilde{z}\| tr\left\{ \tilde{V}^T \left( V - \tilde{V} \right) \right\} + \tilde{z}^T \left( \mathcal{K}[s] + \gamma \right). \end{aligned} \quad (30)$$

It is noticed that $Y = diag\{W, V\}$, $\hat{Y} = diag\{\hat{W}, \hat{V}\}$, and $\tilde{Y} = Y - \hat{Y}$. Then (30) can be expressed as

$$\dot{L} = -\tilde{z}^T K\tilde{z} + \kappa \|\tilde{z}\| tr\left\{ \tilde{Y}^T \left( Y - \tilde{Y} \right) \right\} + \tilde{z}^T \left( \mathcal{K}[s] + \gamma \right). (31)$$

The robust term is designed as

$$\gamma = \begin{cases} -K_Y\left( \|\hat{Y}\|_F + Y_M \right)\tilde{z} - J\frac{\tilde{z}}{\|\tilde{z}\|}, & \|\tilde{z}\| \neq 0 \\ -K_Y\left( \|\hat{Y}\|_F + Y_M \right)\tilde{z}, & \|\tilde{z}\| = 0, \end{cases} \quad (32)$$

where $J$ and $K_Y$ are positive, $Y_M$ is the bound of ideal weights. Substituting (32) into (31) and considering it holds that $tr\left\{ \tilde{Y}\left( Y - \tilde{Y} \right) \right\} = \left\langle \tilde{Y}, Y \right\rangle_F - \|\tilde{Y}\|_F^2 \leq \|\tilde{Y}\|_F \|Y\|_F - \|\tilde{Y}\|_F^2$, if $\tilde{z} \neq 0$, we get

$$\begin{aligned} \dot{L} \leq &-K_{min}\|\tilde{z}\|^2 + \kappa\|\tilde{z}\|\left( \|\tilde{Y}\|_F \|Y\|_F - \|\tilde{Y}\|_F^2 \right) \\ &- K_Y\left( \|\hat{Y}\|_F + Y_M \right)\|\tilde{z}\|^2 - J\|\tilde{z}\| + \|\tilde{z}\|\|\mathcal{K}[s]\|, \end{aligned} \quad (33)$$

where $K_{min}$ is the minimum singular value of $K$.

The following property holds:

**Property 2:** Based on Property 1, since $X$ is bounded, $Y$ and $\tilde{z}$ are smooth continuous, the disturbance term $s(t)$ is bounded by

$$\left\|\mathcal{K}[s(t)]\right\| \le C_0 + C_1\left\|Y\right\|_F + C_2\left\|\tilde{Y}\right\|_F\left\|\tilde{z}\right\|, \tag{34}$$

where $C_0$, $C_1$ and $C_2$ are positive scalars. Substituting it into (33), yields

$$\begin{aligned}\dot{L} \le & -K_{min}\left\|\tilde{z}\right\|^2 + \kappa\left\|\tilde{z}\right\|\left(\left\|\tilde{Y}\right\|_F\left\|Y\right\|_F - \left\|\tilde{Y}\right\|_F^2\right)\\ & - K_Y\left(\left\|\hat{Y}\right\|_F + Y_M\right)\left\|\tilde{z}\right\|^2 + \left\|\tilde{z}\right\|\left(C_0 + C_1\left\|\tilde{Y}\right\|_F\right.\\ & \left. + C_2\left\|\tilde{Y}\right\|_F\left\|\tilde{z}\right\|\right) - J\left\|\tilde{z}\right\|.\end{aligned} \tag{35}$$

It holds that $\left\|\hat{Y}\right\|_F + Y_M \ge \left\|\hat{Y}\right\|_F + \left\|Y\right\|_F \ge \left\|Y - \hat{Y}\right\|_F = \left\|\tilde{Y}\right\|_F$. Taking $K_Y > C_2$, we can obtain

$$\begin{aligned}\dot{L} \le & -K_{min}\left\|\tilde{z}\right\|^2 + \kappa\left\|\tilde{z}\right\|\left(\left\|\tilde{Y}\right\|_F\left\|Y\right\|_F - \left\|\tilde{Y}\right\|_F^2\right)\\ & + \left\|\tilde{z}\right\|\left(C_0 + C_1\tilde{Y}_F\right) - J\left\|\tilde{z}\right\|\\ \le & -\left\|\tilde{z}\right\|\left[K_{min}\left\|\tilde{z}\right\| - \kappa\left\|\tilde{Y}\right\|_F\left(Y_M - \left\|\tilde{Y}\right\|_F\right) - C_0\right.\\ & \left. - C_1\left\|\tilde{Y}\right\|_F + J\right]\\ = & -\left\|\tilde{z}\right\|\left[K_{min}\left\|\tilde{z}\right\| + \kappa\left(\left\|\tilde{Y}\right\|_F - \frac{C_3}{2}\right)^2 - \frac{\kappa C_3^2}{4} - C_0 + J\right],\end{aligned} \tag{36}$$

where $C_3 = Y_M + \frac{C_1}{\kappa}$. Obviously, if we take $J \ge \frac{\kappa C_3^2}{4} + C_0$, it follows that

$$\dot{L} \le -\left\|\tilde{z}\right\|\left[K_{min}\left\|\tilde{z}\right\| + \kappa\left(\left\|\tilde{Y}\right\|_F - \frac{C_3}{2}\right)^2\right] \le 0. \tag{37}$$

Hence (29) is a Lyapunov function. According to the structure of (29), $\tilde{z}$, $\tilde{W}$, and $\tilde{V}$ are bounded. According to LaSalle's principle for nonsmooth system [19], the system must stabilize to the invariant set included in $\left\{\tilde{z}\middle|\dot{V}=0\right\}$, here $\tilde{z}=0$. Since the determinant of $T$ is one, it holds $z = T^{-1}\tilde{z} = 0$ in case of $\tilde{z}=0$. Furthermore $e$ and $\dot{e}$ converge to zero too. And the relative error defined in (2) converges to zero. According to Lemma 2, we can draw the conclusion that the robots converge to a formation whose pattern satisfies $C^k$, $k \ge 1$.

The proof does not assert that $\tilde{W}$ and $\tilde{V}$ converge to zero. Hence although $\hat{f}(X)$ approaches to $f(X)$, the $\hat{W}$ and $\hat{V}$ may not converge to the desired weights $W$ and $V$ without error eventually. But since for a formation, the most important thing is to keep a formation shape, it is acceptable that there exist estimated errors on NN weights.

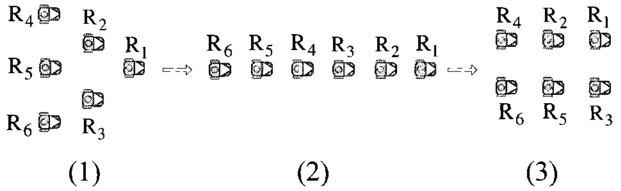


Fig. 5. A variant formation pattern.

To verify the performance of the controller, a simple simulation is illustrated where six mobile robots are required to form a formation with patterns varying three times as shown in Fig. 5.

The interaction topology among robots is invariant, i.e.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0.5 & 0.5 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Some important parameters used are listed as follows:

- Robot's size is $0.14\text{m} \times 0.08\text{m}$, and its mass is 1kg.
- The disturbance $\tau_d$ is a kind of white noise whose range is limited within $[-0.05, 0.05]$.
- Parameters used in (18) and (19) are chosen as: $K = diag\{10, 10\}$, the NN includes 40 hidden nodes, and learning speeds in both $F$ and $U$ are identically 0.1, $\kappa = 10$, and $J = 0.01$.

The trace of the formation is shown in Fig. 6. Since this simulation is to test the control strategy's performance under different formation patterns, the times for pattern change are predetermined. $R_1$ is the leader of the formation, whose desired trajectory is predetermined. $R_1$ follows the trajectory using the same control strategy as others. The tracking errors of all six robots are shown in Fig. 7. Obviously the control strategy guarantees that all tracking errors converge to zero. That means the robots can keep a regular formation according to the formation pattern.

### 4.2. Stability under variant interaction topology

Due to its perceiving range and communication bandwidth, one robot can communicate with other robots within a certain range. Hence if robots have to change their neighborhoods for exchanging information, the interaction topologies also change.

When interaction topology is changed from $\mathcal{G}_1$ to $\mathcal{G}_2$, as well as the adjacency matrix is changed from

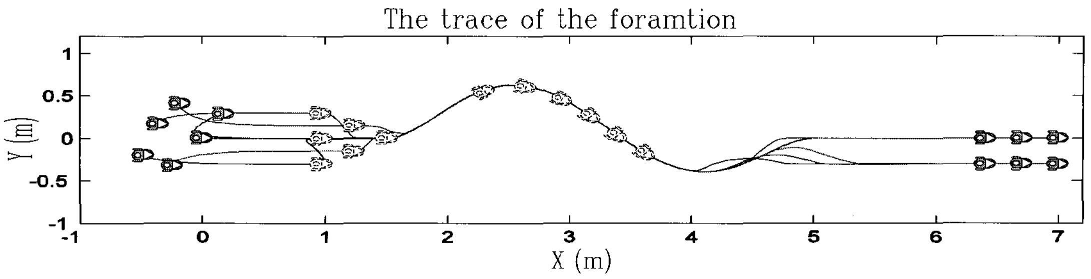


Fig. 6. The trace of the formation.

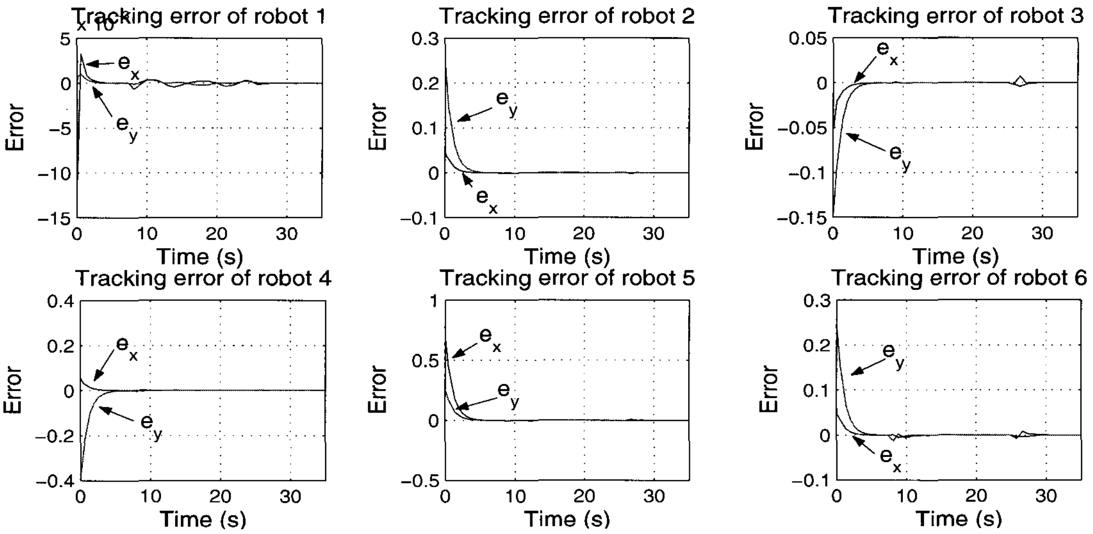


Fig. 7. The relative errors of all robots.

$G_1$ to $G_2$, even if the formation shape and the positions of robots are fixed, the relative errors associated with the two adjacency matrices are different. That means at the instant of the change, the error shown in (8) is noncontinuous, as well as (21) is noncontinuous. Hence the control strategy can not be expressed in the Filippov sense. So we have to modify control strategies to control this noncontinuous system.

Let the set be $\mathbf{G}=\{G_i : i=1,2,\cdots\}$ which consists of all possible connected adjacency matrices. If a time sequence is denoted as $\{t_0,t_1,\cdots,t_i,\cdots\}$, $t_i > t_{i-1}$, where $t_i$ is the instant when the change of adjacency matrix occurs, the duration of formation is a sequence $\{[t_0,t_1),\ [t_1,t_2),\cdots[t_{i-1},t_i),\cdots\}$, and the adjacency matrix in duration $[t_{i-1},t_i)$ is denoted by $G_{t_i}$. Since $p_l^d=0$, the equation $H_iP^d = G_i \circ D^d$ $1_{N\times 1}$ has identical solution $P^d$ for any $G_i \in \mathbf{G}$.

When the adjacency matrix is changed at the instant $t_i$, the control torque is infinity. To eliminate the infinite point, a modification of control torque is proposed as follows:

$$\tau(t_i)=\tau(t_i^-). \qquad (38)$$

Then the control strategy is not a time continuous feedback control any more. In fact, a simple "switch" can be added into the controller, so that when the robot has to change its reference objects, its controller maintains the same control torque as that ahead of this change for a short duration.

Based upon the previous analysis on formation control for invariant interaction topology, we have the following theorem describing performance of formation with variant interaction topologies:

**Theorem 2:** Given a multi-robot system with a predetermined leader, if the interaction topologies among robots change from time to time, then there exists a temporal sequence of adjacency matrices $\{G_{t_0},G_{t_1},\cdots\}$, $t_i > t_{i-1}$, describing relationship change, where $G_{t_i}$ represents the adjacency matrix during time-interval in $[t_{i-1},t_i)$. If $G_{t_i} \in \mathbf{G}$ $(i=1,2,\cdots)$, and formation pattern $D^d$ is $C^k$ $(k \geq 1)$ function, the system will converge to a unique formation according to individual control strategy shown in (18) and (19).

The proof of Theorem 2 is very similar to Theorem 2 in [20] except that in [20] only formation pattern satisfying $C^\infty$ property is analyzed. But this difference does not affect the proof on the stability of formation control with variant interaction topology too much. In a short, the main idea of the proof is based on Theorem 1. If the modification mentioned above is considered, it can be proved when interaction topology is changed, the sudden change of tracking error $e$ is bounded, and in each interval $[t_{i-1}, t_i)$ the system is convergent according to Theorem 1. Hence when time goes to infinity, the robots will still be able to form a formation, even if the changes of interaction topology make the system become a noncontinuous one.

## 5. FORMATION NAVIGATION WITH OBSTACLE AVOIDANCE

The motivation of the formation navigation is described as follows: It is assumed that the sensor range of the leader $R_l$ is bounded but large enough, so that $R_l$ can perceive any obstacles which may collide with robots in the near future. If there are no obstacles perceived by $R_l$, it will generate a beeline connecting its position with the destination, so that other robots will follow $R_l$ to reach the destination. But if any obstacles are moving into the sensor range of $R_l$, it will generate a proper path to the destination, so that all members of the formation follow it to avoid moving obstacles while keeping regular formation shape. During the process the formation pattern should be changed to avoid obstacles conveniently.

Owing to the sensor limits for detecting obstacles, $R_l$ is only able to perceive obstacles within its sensor range. With appearance and disappearance of obstacles within its sensor range, $R_l$ has to generate desired paths from time to time, which will induce the change of formation pattern $D^d(t)$. At the same time, since interactions among robots may be truncated by obstacles, robots have to find new interactions with other robots to set up new reference points. That means the interaction topology, or adjacency matrix $G$ will be timely changed. However the two theorems mentioned above imply that if formation navigation ensures the paths generated are smooth continuous, i.e. $C^k\ (k \geq 1)$, robots can keep formation using the decentralized NN controller, even if $D^d(t)$ and $G$ are variant. Hence if the formation shape is fixed, the key point of formation navigation is the path planning for $R_l$, in order that the paths generated are smooth continuous with obstacle avoidance for all robots.

This paper proposes a path planning via particle swarm optimization to fulfill these requirements.

5.1. Description of desired trajectory of the leader

Let $p^d(t) = [p_x^d(t),\ p_y^d(t)]^T$ be a virtual moving point on a desired trajectory. If the coordinate in X-direction is the function of time, i.e., $p_x^d(t) = \varphi(t)$, the smooth path is expressed as an algebraic cubic spline, $p_y^d(t) = \sum_{i=0}^{n} a_i (p_x^d(t))^i$.

Since the desired trajectory should be at least $C^1$ to apply local control strategy, not only position boundary conditions, but also velocity boundary conditions should be applied.

$$p^d(t_0) = P^{t_0}, \qquad p^d(t_f) = P^{t_f},$$

$$\left.\frac{dp_y^d}{dp_x^d}\right|_{t=t_0} = \theta^{t_0}, \qquad \left.\frac{dp_y^d}{dp_x^d}\right|_{t=t_f} = \theta^{t_f},$$

where $[t_0, t_f]$ represents the interval for the moving point from the start time of path planning to the end time of reaching destination; $\theta^{t_0}$ represents the heading angle of $R_l$ at the moment for new path planning. Therefore between two successive times of path planning, the path generated in the latter path planning must continue the former path smoothly. And the whole desired trajectory is ensured as $C^1$.

If only the boundary condition is considered, a three-order polynomial trajectory function can be chosen. To avoid moving obstacles, a five-order polynomial for path planning is chosen as

$$p_y^d = \sum_{i=1}^{5} a_i (p_x^d)^i. \tag{39}$$

There are six parameters $a_0$ to $a_5$ to be determined. According to the boundary conditions, only two of them are free parameters, and the other four parameters can be expressed by these two.

5.2. Path planning via PSO algorithm

5.2.1 PSO algorithm

Obviously each particle in a swarm represents a solution on path planning. In following analysis, we will introduce the meaning of each particle and the algorithm of PSO path planning.

Let $S$ denote the size of the swarm. For an arbitrary particle $i$, its current position is denoted by $\xi_i = [\xi_{i1}\ \ \xi_{i2}\ \ \cdots\ \ \xi_{iL}]$, where $L$ is the dimension of the solution space, and its current velocity is denoted by $v_i$. Assume that the function $F(\bullet)$ is to be minimized, $r_1 \sim U(0,1)^L$ and $r_2 \sim U(0,1)^L$ represent the two random vectors in the range of $[0,1]^L$. To ensure convergence, the adjustment of particle with a constriction factor [21] is expressed as

$$v_i(t+1) = K_c \left\{ v_i(t) + c_1 r_{1i}(t) \left[ Y_i - \xi_i(t) \right] \right.$$

$$+c_2 r_{2i}(t)\left[Y_i^g - \xi_i(t)\right]\Big\}, \qquad (40)$$
$$\xi_i(t+1) = \xi_i(t) + v_i(t+1),$$

where $c_1$ and $c_2$ represent the acceleration coefficients satisfying with $\phi = c_1 + c_2$, $\phi > 4$; $Y_i$ represents the best position found by particle $i$ so far; $Y_i^g$ represents the global best position among particle $i$'s neighborhood; the constriction factor $K_c$ is defined as $K_c = 2\Big/\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|$.

The best position recorded is updated by

$$Y_i(t+1) = \begin{cases} Y_i(t), & F(\xi_i(t+1)) \geq F(Y_i(t)) \\ \xi_i(t+1), & F(\xi_i(t+1)) < F(Y_i(t)). \end{cases} \qquad (41)$$

And the global best position found by particle $i$ 's neighborhood is modified by

$$Y_i^g(t+1) = arg \min_{j \in \Pi_i} F(Y_j(t+1)), \qquad (42)$$

where $\Pi_i$ represents the neighbors of particle $i$.

5.2.2 Fitness evaluation

The goal of a PSO is to minimize a fitness function $F(\cdot)$. For path planning, two requirements should be considered in fitness function: (i) Arriving at destination along the trajectory as soon as possible; (ii) Avoiding obstacles.

1) Fitness with respect to trajectory's length.

Instead of a direct measurement of path length, another fitness function is chosen. If the X-axis of the universal reference frame is along the beeline connecting the leader and the destination, the fitness function can be expressed as

$$F^{path} = \int_{p_x^d(t_0)}^{p_x^d(t_f)} (p_y^d)^2 dp_x^d. \qquad (43)$$

It reflects the intention that the desired trajectory should be as close as possible to the beeline connecting two ends of the trajectory.

2) Fitness with respect to obstacle avoidance.

To avoid obstacles, the shortest distance between obstacles and robots during the whole process should be larger than a critical or safety threshold. If we define such a threshold as $\rho^{eff}$, and let $\Omega$ and $\Psi$ represent the set of robots and the set of obstacles perceived by $R_l$ respectively, then this intention can be expressed as $\forall t,\ \forall i \in \Omega,\ \forall j \in \Psi,\ min\{\rho_{ij}\} \geq \rho^{eff}$, where $\rho_{ij}$ represents the distance between robot $i$ and obstacle $j$. If let $\rho_j^{min} = min\{\rho_{ij}\}$, an evaluation function for obstacle avoidance is designed as

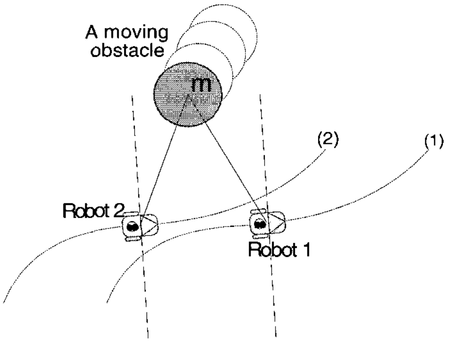


Fig. 8. A snapshot of two virtual robots at time $t_s$.

$$F_j^{obstacle} = \begin{cases} \mu(\frac{1}{\rho_j^{min}} - \frac{1}{\rho^{eff}}), & \rho_j^{min} \leq \rho_j^{eff} \\ 0, & \rho_j^{min} > \rho_j^{eff}. \end{cases} \qquad (44)$$

Therefore the key point is to find out $\rho_j^{min}$. Fig. 8 illustrates how to find such a distance, where there are two trajectories denoted by lines (1) and (2), which are designed for two robots forming a formation. Then the minimal distance between an obstacle and robots equals the minimal distance between the obstacle and all trajectories. A critical point is defined such that a beeline through obstacle position intersects with the trajectory perpendicularly on it. Then if we find this critical point, $\rho_j^{min}$ also be calculated.

Given an obstacle $m$ as shown in Fig. 8, since the path designed for the leader $R_1$ is the function of time, according to the formation pattern, the path of the follower $R_2$ is also expressed as the function of time. It is assumed that two moving points denoted by two virtual robots follow trajectories (1) and (2) generated by the PSO path planning. we draw two perpendiculars (denoted by dashed lines) with the same slope passing through the positions of robots respectively, and draw two connect lines connecting robots with the obstacle $m$ respectively. If a robot is at the critical point, the connecting line must coincide with the perpendicular there. Therefore a fitness function on evaluating critical point is expressed as

$$F_j^{crosspoint} = \left(1 + \frac{p_{jy}^o - p_{jy}^c}{p_{jx}^o - p_{jx}^c} \cdot \left.\frac{dp_y^d}{dp_x^d}\right|_{P^d = P_j^c}\right)^2, \qquad (45)$$

where $j \in \Psi$, $P_j^c = [p_{j1}^c \quad p_{j2}^c]^T$, and $P_j^o = [p_{j1}^o \quad p_{j2}^o]^T$ represent the coordinates of critical point and obstacle respectively.

If there are $M$ obstacles, the fitness function in PSO for path planning is in the form of

$$F = \omega_1 \cdot F^{path} + \omega_2 \cdot \sum_{i=1}^{M} F_i^{crosspoint} + \omega_3 \cdot \sum_{i=1}^{M} F_i^{obstacle}, (46)$$

where $\omega_1$ to $\omega_3$ represent positive weights.

5.2.3 Description of particles in swarm

Based on the analysis, the dimension of solution space can be determined. Firstly $a_5$ and $a_3$ are chosen as free parameters in order to describe a desired trajectory. And for every obstacle perceived by $R_l$, we need to find the critical point. If a desired trajectory is described as a function of time, the critical point for obstacle $j$ is also a function of time $T_j^c$. Therefore if we assume that $R_l$ can handle $M$ obstacles at the same time, the position of a particle is in the form of $\xi = [a_5 \quad a_3 \quad T_1^c \quad T_2^c \quad \cdots \quad T_M^c]^T$.

## 6. SIMULATION

To illustrate the feasibility of the design of formation navigation, a simulation is carried out in which six robots maintain a rectangle formation. Some assumptions on the environment and parameters used in the navigation are listed below:

1) There are four obstacles. A static obstacle is located at (2.5m, 0.1m). And other three obstacles are moving, which are located initially at (3.5m, 0.55m), $(3.7\text{m}, -0.45\text{m})$, and $(6\text{m}, 0.55\text{m})$. The velocities of the obstacles are $(0\text{m/s}, -0.014\text{m/s})$, $(0.025\text{m/s}, 0.012\text{m/s})$ and $(-0.012\text{m/s}, -0.02\text{m/s})$ respectively.
2) All obstacles are a kind of disc-like with radius 0.15m.
3) $R_1$ is the leader of the formation, which is required to reach (7m, 0m) with a heading angle of 0rad, while all six robots are keeping rectangle formation whose pattern is shown in Fig. 1.
4) Considering the size of robots, the distance for safe obstacle avoidance is chosen as 0.05m. Since an obstacle is a disc-like with radius 0.15m, the safe range is a disc with radius 0.2m.
5) $R_l$ handles two obstacles at one time.

Other parameters such as the size of robot and control parameters are the same as the previous simulation. The results of simulation are displayed in Figs. 9 and 10. To describe the movements of obstacles, the initial positions and final positions of obstacles are denoted by black circles, while arrows passing through them denote their moving direction. The four grey discs denote the nearest positions where robots are away from obstacles. At these locations, the red circles around the discs represent the safety ranges of obstacles. Obviously, no trajectory of robots passes through any safety range, so that the minimum distance between any robot and any obstacle must be no less than 0.05m, and the robot must avoid the obstacle.

Since $R_1$ only handles the two nearest obstacles at one time of path planning, for all four obstacles, three times of path planning are needed. Counting the duration to form formation firstly, the process of formation is divided into four sections, denoted by (1) to (4) in Fig. 9(a). When $R_1$'s coordinates in X-direction are 1.52m, 3.36m, and 4.62m, the formation executes three times of path planning.

According to the description of particles in swarm, it is known that $\xi = [a_5 \quad a_3 \quad T_1^c \quad T_2^c]^T$. Since $a_5$ and $a_3$ determine the polynomial of path, Fig. 10 displays the evolutionary processes about these two parameters in the second and the third times of path planning, where the obstacles involved are all moving obstacles. From the figure, it is observed that in every planning, all particles converge. If the desired velocity along X-direction of $R_1$ is predetermined as $p_x^d(t) = 0.2t$, the polynomials of paths after three times of path planning are:

1) Duration $t \in (7.7\text{s}, 16.8\text{s}]$:

$$p_{1y}^d(t) = 8.10\times10^{-9}t^5 - 8.80\times10^{-7}t^4 + 3.52\times10^{-5}t^3 \\ -6.31\times10^{-4}t^2 + 0.0049t - 0.0137;$$

2) Duration $t \in (16.8\text{s}, 23.1\text{s}]$:

$$p_{2y}^d(t) = -5.02\times10^{-8}t^5 + 6.77\times10^{-6}t^4 - 3.57\times10^{-4}t^3 \\ +0.0091t^2 - 0.1135t + 0.5442;$$

3) Duration $t \in (23.1\text{s}, 35\text{s}]$:

$$p_{3y}^d(t) = -5.76\times10^{-8}t^5 + 9.30\times10^{-6}t^4 - 5.87\times10^{-4}t^3 \\ +0.0182t^2 - 0.2748t + 1.6325.$$

When obstacles truncate interactions between robots, robots have no choices but try to communicate with other robots to set up new reference points. This induces a change of interaction topology. In the simulation, thirty times of such changes are recorded. Fig. 9(b) shows some topologies in simulation. Topology (1) indicates the situation at the beginning which is generated arbitrarily. The first change occurs during the formation passing obstacle 1, where the obstacle blocks the interaction between $R_1$ and $R_2$. $R_2$ abandons interaction with $R_1$ and totally turns to $R_4$ to set up the reference point. This change is illustrated in Fig. 9(b), where arc $a(2, 1)$ is disappeared in interaction topology (2), with $a(2, 4)$ remained. So the second row of adjacency matrix $G$ is changed from [0.5 0 0 0.5 0 0] to [0 0 0 1 0 0]. Since these changes of adjacency matrix occurs after the formation formed. the leaps of errors are too small to be observed in the first six figures of Fig. 9(c). But if the relative error of $R_4$ is magnified, through observing the region around 17.5s shown in the seventh subfigure of Fig. 9(c), we can find two leaps of error corresponding to two times

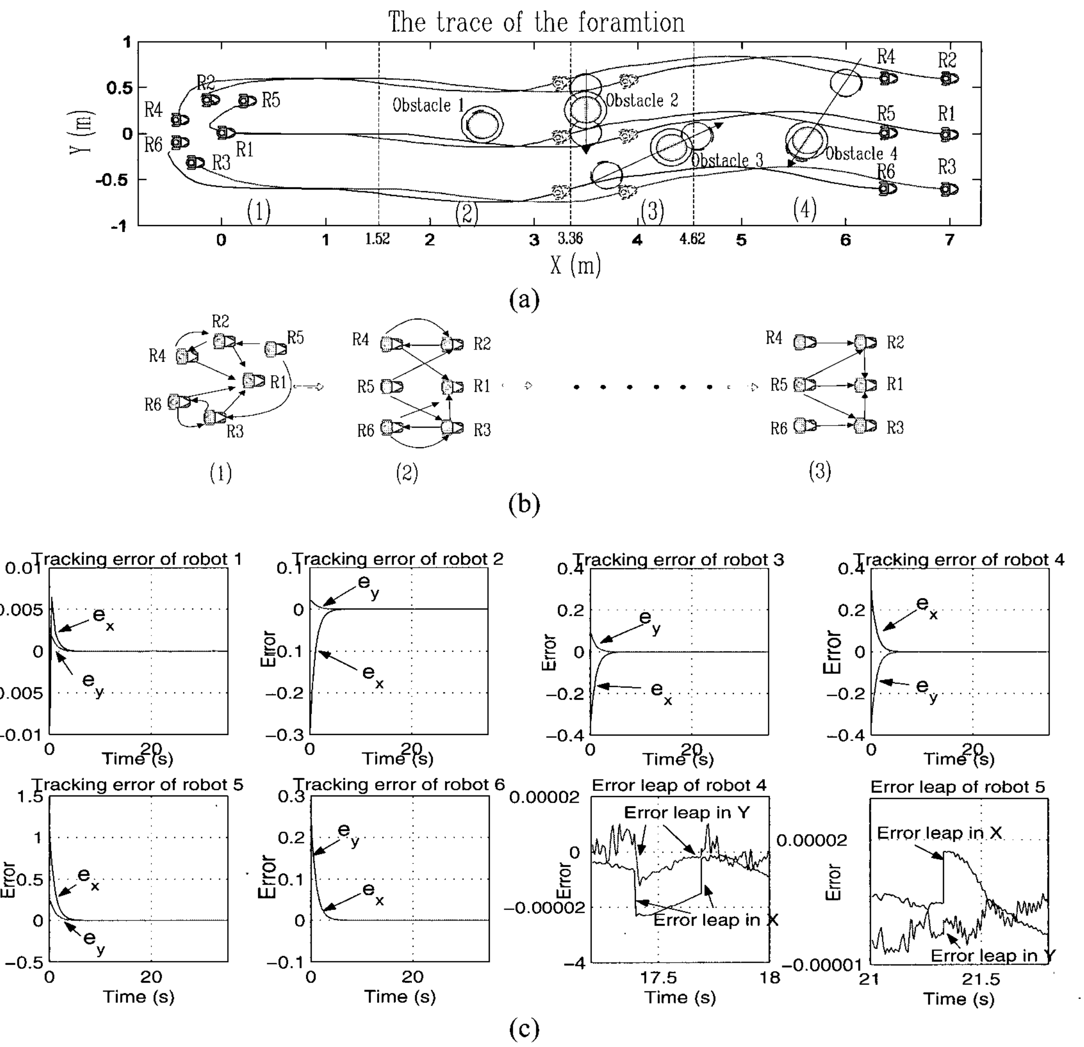


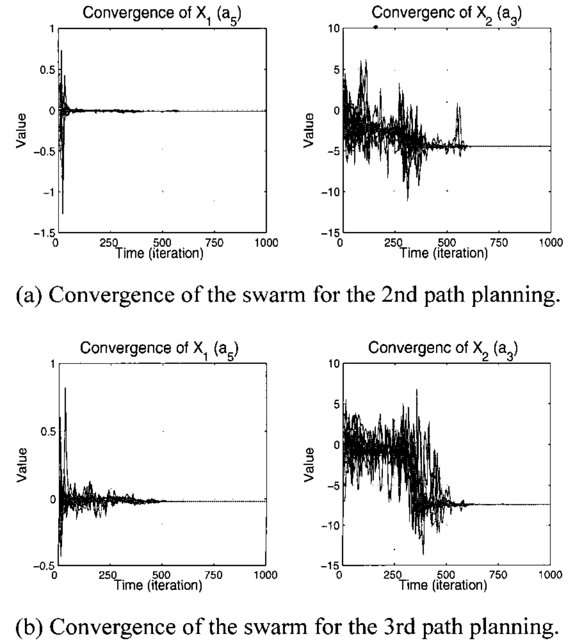
Fig. 9. Simulation results of formation navigation.

of interaction changes of $R_4$, when $R_4$ avoids obstacle 1, its reference robot is changed from $R_1$ to $R_3$, and from $R_3$ to $R_2$, respectively.

## 7. EXPERIMENTAL STUDIES

An experiment study is conducted in our lab to verify this formation navigation. Up to now we have designed and manufactured a group of robots as shown in Fig. 11. All robots can acquire the information about their relative positions using lights and photosensors.

Each individual robot has been designed as a full autonomous mobile robot, on which a Microchip PIC® microprocessor is mounted. Other necessary devices, such as communication parts, memory chips are all included. According to the mechanical design, the robot is constructed by 4 units, or layers from the top to the bottom: 1) interaction unit including the lights and photo sensors, which is mounted on the top layer, 2) extended board on the second layer, which will extend ability of input and output channels, 3)



(a) Convergence of the swarm for the 2nd path planning.

(b) Convergence of the swarm for the 3rd path planning.

Fig. 10. Evolution processes for two path planning.

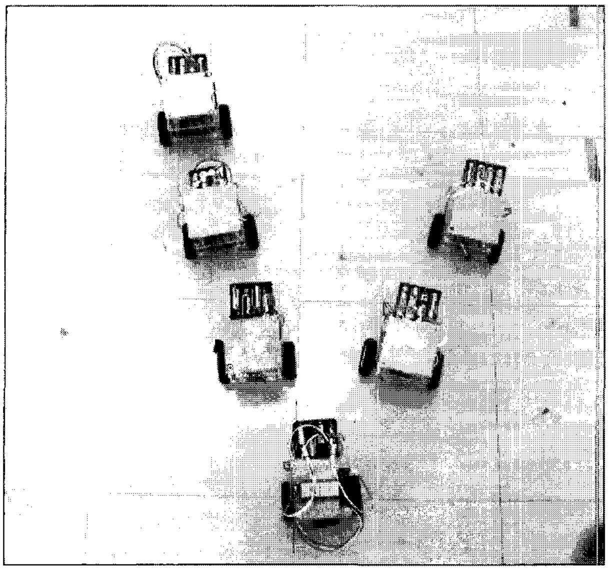


Fig. 11. The prototype of the multi-robot formation.

control unit on the third layer, which includes CPU, 4096 bytes of user program and data memory, and IR transceivers for communication with PC, 4) power source and actuator unit on the bottom layer, which include two separated battery units and two servo motors with gear boxes. The system architecture is finally shown in Fig. 12.

The following two layers of the architecture are described as follows.

1) Interaction unit

All lights mounted on robots are surrounded by light tight material, so that directional light cones are generated. Accordingly on the top layer there are several photo sensors which can perceive light from specific directions. Robots can perceive the beams to detect their reference position. Therefore the arrangement of lights and photo sensors determine the formation pattern. Till now the lights and photo sensors are fixed on the first layer, so that an invariant formation pattern can be generated, such as the formation of wild goose.

2) Power source and actuator

To avoid current disturbance resulting from action of actuator, which make power supply to the control board be unstable, the two separated 6V battery units are employed to provide power to the control unit and drive motors respectively. And the actuator unit includes two servo motors, which drive two wheels through a gear box with reduction ratio of 203:1.

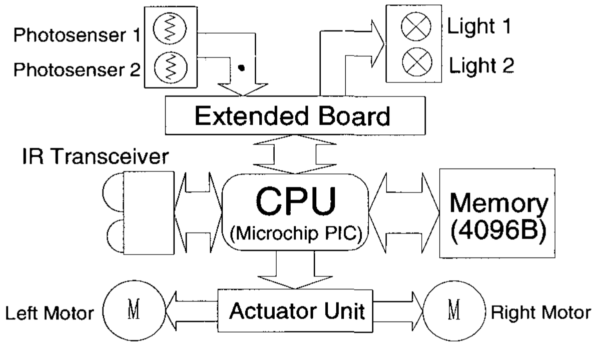


Fig. 12. The mechanical architecture of a robot.

Using this prototype, we have tested that when the robots are put together, they can perceive their leaders and compute the relative positions according to the intension of lights. In the near future the close-loop control law will be applied to the robots to test the formation performance. And finally the NN control will be added into the controller to verify the feasibility of the control strategy.

## 8. CONCLUSIONS

A formation navigation algorithm for a group of mobile robots is proposed in this paper, which can achieve formation navigation in case of moving obstacles existed. If formation pattern is $C^1$, the whole system is smooth continuous. Then the adaptive NN control can be applied to enable mobile robots follow a predetermined leader and form a formation. A PSO path planning method is proposed to generate successive paths for the leader of formation according to current obstacles perceived, and ensures that the path resulting from connection of these paths is smooth. Hence the precondition of $C^1$ formation pattern is fulfilled. Simulation results demonstrate the algorithm is effective for formation navigation of multiple robots with moving obstacles.

## REFERENCES

[1] A. Jababaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. on Automatic Control*, vol. 48, no.6, pp. 988-1001, June 2003.

[2] W. Ren and R. W. Beard, "Consensus seeking in multi-agent systems under dynamically changing interaction topologies," *IEEE Trans. on Automatic Control*, vol. 50, no. 5, pp. 655-661, 2005.

[3] Y. Liu, K. M. Passino, and M. Polycarpou, "Stability analysis of one-dimensional asynchronous swarms," *Proc. of American Control Conference*, pp. 716-721, June 2001.

[4] P. Ogren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multi-agent coordination," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 5, pp. 847-851, Oct. 2002.

[5] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formation of multiple mobile robots," *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 4, pp. 2864-2869, May 1998.

[6] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 905-908, Dec. 2001.

[7] Y. Liu and Y. Li, "Sliding mode adaptive neural-network control for nonholonomic mobile modular manipulators," *Journal of Intelligent and Robotic Systems*, vol. 44, no.3, pp. 203-224, 2005.

[8] F. L. Lewis, A. Yegildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. on Neural Networks*, vol. 7, no. 2, pp. 388-399, Mar. 1996.

[9] W. Kowalczky and K. Kozlowski, "Artificial potential based control for a large scale formation of mobile robots," *Proc. of the 4th Int. Workshop on Robot Motion and Control*, pp. 285-291, June 2004.

[10] E. Bicho and S. Monteiro, "Formation control for multiple mobile robots: A non-linear attractor dynamic approach," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, pp. 2016 - 2022, Oct. 2003.

[11] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 73-80, Apr. 2000.

[12] E. Dyllong and A. Visioli, "Planning and real-time modifications of a trajectory using spline techniques," *Robotica*, vol. 21, pp. 475-482, 2003.

[13] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," *Proc. of 6th Int. Symp. on Micro Machine and Human Science*, pp. 39-43, 1995.

[14] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proc. of IEEE Int. Conf. on Neural Network*, pp. 1942-1948, 1995.

[15] Y. Li and X. Chen, "Leader-formation navigation using dynamic formation pattern," *Proc. of IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Monterey, California, USA, pp.1494-1499, July, 2005.

[16] J. Tu and S. X. Yang, "Genetic algorithm based path planning for a mobile robot," *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 1221-1226, September 2003.

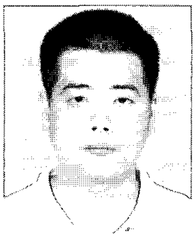
[17] C. Wang, Y. C. Soh, H. Wang, and H. Wang, "A hierarchical genetic algorithm for path planning in a static environment with obstacles," *Proc. of Congress on Evolutionary Computation*, vol. 1, pp. 500-505, May 2002.

[18] A. F. Filippov, "Differential equation with discontinuous right-hand side," *Amer. Math. Soc. Translations*, vol. 42, no. 2, pp. 191-231, 1964.

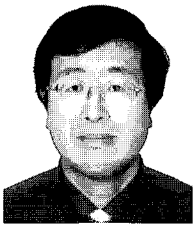
[19] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. on Automatic Control*, vol. 39, no. 9, pp. 1910-1914 1994.

[20] Y. Li and X. Chen, "Stability on multi-robot formation with dynamic interaction topologies," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1325-1330, Aug. 2005.

[21] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.



**Xin Chen** received the B.S. degree in Industrial Automation, and M.S. degree in Control Theory and Control Engineering from Central South University, Changsha, China in 1999 and 2002 respectively. He is currently working toward a Ph.D. degree in the Department of Electromechanical Engineering, University of Macau. His research interests include swarm intelligence, multiple robot coordination.



**Yangmin Li** received the B.S. and M.S degrees in Mechanical Engineering from Jilin University, Changchun, China in 1985 and 1988 respectively. He received the Ph.D. degree in Mechanical Engineering from Tianjin University, Tianjin, China in 1994. After that, he worked in South China University of Technology, International Institute for Software Technology of the United Nations University (UNU/IIST), University of Cincinnati, and Purdue University. He is currently an Associate Professor majored in robotics, mechatronics, control, and automation in University of Macau. He is an IEEE Senior Member and a Member of ASME. He has been serving as a Council Member of Chinese Journal of Mechanical Engineering since 2004. He has authored or co-authored about 140 papers in international journals and conferences.