Optimal Formation of Multirobot Systems Based on a Recurrent Neural Network

Yunpeng Wang, Long Cheng, *Senior Member, IEEE*, Zeng-Guang Hou, *Senior Member, IEEE*, Junzhi Yu, *Senior Member, IEEE*, and Min Tan

***Abstract*— The optimal formation problem of multirobot systems is solved by a recurrent neural network in this paper. The desired formation is described by the shape theory. This theory can generate a set of feasible formations that share the same relative relation among robots. An optimal formation means that finding one formation from the feasible formation set, which has the minimum distance to the initial formation of the multirobot system. Then, the formation problem is transformed into an optimization problem. In addition, the orientation, scale, and admissible range of the formation can also be considered as the constraints in the optimization problem. Furthermore, if all robots are identical, their positions in the system are exchangeable. Then, each robot does not necessarily move to one specific position in the formation. In this case, the optimal formation problem becomes a combinational optimization problem, whose optimal solution is very hard to obtain. Inspired by the penalty method, this combinational optimization problem can be approximately transformed into a convex optimization problem. Due to the involvement of the Euclidean norm in the distance, the objective function of these optimization problems are nonsmooth. To solve these nonsmooth optimization problems efficiently, a recurrent neural network approach is employed, owing to its parallel computation ability. Finally, some simulations and experiments are given to validate the effectiveness and efficiency of the proposed optimal formation approach.**

***Index Terms*— Combinational optimization problem, multirobot system, optimal formation, recurrent neural network, shape theory.**

I. Introduction

A. Formation Problem of Multirobot Systems

RECENT decades have witnessed a significant progress in coordination and cooperation of multirobot systems due to its great potential in many practical applications, such as the urban search and rescue [1], surveillance and reconnaissance [2], environmental monitoring [3], and mapping of unknown environments [4]. Among different coordination tasks of multirobot systems, the formation problem is always a fundamental concern since an appropriate organization of robots can dramatically increase the coordination efficiency. For example, in the practice of unknown area exploration, a well-organized formation can have a broader coverage and reduce the possibility of reduplicate searching.

Basically, the formation problem of multirobot systems includes two steps: 1) determine a desired formation and 2) design the corresponding control algorithm for reaching this formation. Recently, most results regarding the formation problem concentrate on the latter one. Those control strategies include the leader-follower strategy [5]–[9], the distributed control strategy [10]–[14], the graph-based strategy [15], [16], and the intelligent control strategy [17], [18]. The leader-follower strategy is the most popular method in the formation control of multirobot systems. In [5], a vision-based formation controller was proposed in the leader-follower framework. By using the controller, a group of robots can achieve and maintain a prescribed formation when they move along a planned trajectory. When considering the controller's physical limitation, a leader-follower formation controller with bounded input was proposed in [6], where the conditions on the formation keeping and asymptotic stabilization were presented. Park *et al.* [7] designed an adaptive leader-follower formation approach for the case where the robot's velocity is unavailable. In the leader-follower case, leader plays a central role on the success of formation. If leader is attacked or malfunctioning, then the whole system may lead to some unpredicted behaviors. Distributed formation control means that every robot is functionally equal in the system, and the formation can be achieved by exchanging information among robots. It should be noted that each robot can only receive information from its neighbors rather than all robots in the system. For example, Lin *et al.* [10] showed that if the positions of all robots could be convergent to a same point, then more general formation could be achieved as well. Based on this observation, three formation strategies were proposed. In [10], each robot is modeled by the single-integrator dynamics, which neglects the mobile robot's orientation. In [11], each robot is modeled as a unicycle and the corresponding formation strategy was proposed too. Furthermore, in [13], a formation algorithm based on the small-gain theorem was presented for multiple unicycles, which was robust to the position measurement error. In [14], the formation control of a group of underactuated ships was investigated. In addition, the collision avoidance was also considered in the formation controller design. In those distributed control strategies, the relationship among robots is usually described by a graph.

Manuscript received August 13, 2014; revised June 4, 2015; accepted July 18, 2015. Date of publication August 25, 2015; date of current version January 18, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61370032, Grant 61422310, Grant 61375102, Grant 61225017, and Grant 61421004, and in part by the Beijing Nova Program under Grant Z121101002512066. *(Corresponding author: Long Cheng.)*

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yunpeng_wang@foxmail.com; chenglong@compsys.ia.ac.cn; hou@compsys.ia.ac.cn; junzhi.yu@ia.ac.cn; tan@compsys.ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2015.2464314

Besides designing specific controllers, how does the graph (connection among robots) affect the controller's performance and the formation achievement should be answered. This is the motivation of the graph-based formation strategy. In [15], a Nyquist criterion was proposed to determine whether certain formation can be stabilized under different graphs. In addition, Anderson *et al.* [16] revealed which graph could cause the difficulty of controlling a formation. Most aforementioned formation strategies are based on the traditional control methods, which can hardly deal with uncertainties in the robot's dynamics. Intelligent control approaches such as neural networks and fuzzy logics are very suitable for this case. For example, in [17], the fuzzy approach was employed to approximate uncertainties in the robot's dynamics and the sliding-mode approach was employed to counteract the external disturbances. In [18], the fuzzy logic was replaced by the neural network and the linear and angular velocities of robots were estimated by a neural network as well.

The above paragraph reviews some recent advances on the formation control strategies of multirobot systems. However, the first step of formation problem, how to determine a desired formation, is still not well studied. There are typically two ways to describe a formation. One is to use the relative distances and angles between robots. The other one is to use the shape theory, which can provide a much more concise description of formation. The shape theory only cares about the essential geometric relationship between robots. As long as two graphs are similar, they have the same shape. For example, by the shape theory, we can use the term equilateral triangle to describe a formation, however, the orientation and size of this equilateral triangle are free to choose. Therefore, the shape theory can provide a set of feasible formations with a same shape. How to choose one formation from this set is the kernel part to determine a desired formation. To the best of our knowledge, most papers concerning the formation problems pay little attention on this. Even so, some researchers still did some seminal works [19], [20]. In [20], finding a desired/optimal formation became a standard optimization problem by the shape theory, and a numerical algorithm was proposed to solve the optimization problem. For conventional numerical algorithms, the computation time required to obtain the solution is usually dependent of the problem size [21]–[23]. In other words, the efficiency of the numerical algorithms decreases as long as the problem size increases. It becomes a bottleneck when the size of the problem is huge. Recently, this bottleneck is relaxed to some extent by parallel numerical algorithms, especially the ones that can be implemented on graphics processing units [24], [25]. Even so, the computation time of the parallel numerical algorithms still has a positive correlation with the problem size. Therefore, the bottleneck still exists. In addition, it is noted that in [20], each robot corresponds to one specific position in the formation, however, this is unnecessary if robots are all functionally identical since the positions of robots are exchangeable in this case. To deal with this difficulty, we extend the optimization problem in [20] to a combinational optimization problem in this paper. It turns out that the optimization problem in [20] is a special case of the proposed combinational optimization problem. Unfortunately, the extended optimization problem is a NP-hard problem, which is much more difficult to be solved by the traditional numerical methods.

B. Recurrent Neural Networks for Optimization Problems

From the discussion in Section I-A, the key point of the optimal formation is to solve an optimization problem effectively. Due to the rapid development of intelligent techniques, scholars in the computational intelligence community find that the recurrent neural network has shown a great potential as a powerful weapon for solving optimization problems because of its parallel computation nature. In contrast with conventional numerical methods, the efficiency of the recurrent neural network does not decrease when the size of the optimization problem increases [21]–[23], [26]. Moreover, the convergence rate of the neural network can be arbitrarily increased by properly adjusting the parameters in the neural network. This paper is devoted to solving the optimal formation problem by the recurrent neural network. A brief review of recurrent neural networks for optimization problems is given as follows.

In [27], the traveling salesman problem was successfully solved by a recurrent neural network (Hopfield neural network), which is the first recurrent neural network model for optimization problems and indicates the world-wide renaissance of neural network studies. After that, the various recurrent neural network models were presented for different types of optimization problems. For example, the nonlinear programming neural network [28], the Lagrange neural network [29], the primal-dual neural network [30], the dual neural network [31]–[33], the projection neural network [34], and its delayed versions [35]–[37], to name a few. And many engineering applications have been solved by the recurrent neural networks, such as the image processing [38], the model predictive control [39], [40], the manipulator control [41], the coordination of interconnected systems [42], [43], and the scientific computing [44].

One common assumption of the aforementioned neural networks is that the optimization problem in concern is a smooth optimization problem. However, the optimal formation problem considered in this paper may not be transformed into a smooth optimization problem. This is because if the desired formation is the one, which has the minimal distance from the initial formation of the multirobot system, the Euclidean norm of decision variables is included in the objective function, which results in the nonsmooth optimization problem. Fortunately, there are a few recurrent neural network models regarding the nonsmooth optimization problems in the literature. In [45], a generalized neural network based on the model proposed in [28] was presented for nonsmooth nonlinear programming problems. Extensions to the nonsmooth convex optimization problem were made in [46]. However, these models can only obtain an approximate solution to the optimization problem. In order to achieve the exact optimal solution, a one-layer projection neural network was proposed for nonsmooth optimization problems with the

linear equality constraints and the bound constraints [47]. Since inequality constraints frequently occur in engineering problems, recurrent neural networks were also designed for the nonsmooth optimization problem with the linear equality constraints, inequality constraints, and bound constraints [48], [49]. These models represent the state of the art of solving nonsmooth optimization problems based on recurrent neural networks.

C. Contribution and Organization

In this paper, the optimal formation problem is solved by the recurrent neural network proposed in [49]. Although, the neural network proposed in [48] has a simpler structure than the one in [49], there is a parameter in the neural network model of [48] which is strongly related with the minima of inequality constraints and then is hard to determine in practice. The main contributions of this paper lie in the following three aspects.

1) A new neural-network-based approach for solving the optimal formation of multirobot systems is proposed. This approach is featured by the parallel computation ability of neural networks, and is especially effective to deal with the formation problem where the size of multirobot systems is large.
2) The orientation, scale, and admissible range of the desired formation are also considered, which can be transformed into the constraints of the optimization problem.
3) The extended formation problem where all robots' positions in the formation are exchangeable is studied. The extended formation problem is transformed into a combinational optimization problem, which can also be solved by the employed recurrent neural network based on the penalty method.

Finally, several simulation examples are conducted to verify the theoretical analysis and testify the effectiveness of the neural network based solution.

The remainder of this paper is organized as follows. In Section II, some preliminaries are given, which include the shape theory, the subgradient of nonsmooth function and the penalty method. The problem formulation is formally presented in Section III. In Section IV, the extended formation problem is proposed. The recurrent neural network model for solving the optimal formation problem is given in Section V. Three simulation examples and two experiments are given in Section VI. Section VII concludes this paper with final remarks.

The following notations will be used throughout this paper: $1_n = (1, 1, \ldots, 1) \in \mathbb{R}^n$; $0_n = (0, 0, \ldots, 0) \in \mathbb{R}^n$; I_n denotes the $n \times n$ dimensional identity matrix; $\mathbb{R}^n_+$ denotes the nonnegative quadrant of the n-dimensional real space $\mathbb{R}^n$; and $\otimes$ denotes the Kronecker operator. For a given matrix X, X^T denotes its transpose. For a given vector $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, $\|x\|_2$ is the Euclidean-norm of x which is defined by $\|x\|_2 = (\sum_{i=1}^n x_i^2)^{1/2}$. For a 2-D vector (x_1, x_2) in a Cartesian plane, arctan (x_2, x_1) denotes the angle between vector (x_1, x_2) and the horizontal axis. For a set S, int(S) denotes the set of interiors of S.

II. Preliminaries

A. Shape Theory

In this paper, the formation problem of multirobot systems is described by the shape theory [20]. On a Cartesian plane, there are n robots, which are required to form a certain formation. Let $p_i = (p_i^x, p_i^y)^T$ be the Cartesian coordinate of robot i related to the world frame $\mathcal{W}$. The formation of the multirobot system can be defined by the desired positions of all robots in the system

$$P = (p_1, p_2, \ldots, p_n) \in \mathbb{R}^{2\times n}.$$

However, many applications are only concerned with the relative relationship between robots rather than the absolute location. In this case, the shape theory becomes a more appropriate way to describe the formation. Shape is an abstractive concept. The shape of an object is the geometrical information that remains when location, scale, and rotational effects are filtered from the object [50], [51]. It is noted that location, scale, and rotational effects correspond to translation transformation, scaling transformation, and rotation transformation, respectively. That is to say, two formations have the same shape if one can been transformed into the other one by a similarity transformation. Therefore, the shape of formation P can be defined as an equivalence class of formations which have the same shape as P [20]

$$[P] = \left\{\alpha RP + 1_n^T \otimes d | \alpha > 0, R \in \mathrm{SO}(2), d \in \mathbb{R}^2\right\} \quad (1)$$

where α is the scale parameter, $R \in \mathrm{SO}(2)$ denotes the set of 2-D rotation matrices, and $d = (d^x, d^y)^T$ is the translation vector. All formations belonging to $[P]$ are similar to each other. And for any two formations $P_1, P_2 \in [P]$, the similarity relation between them is denoted by $P_1 \sim P_2$.

B. Subgradient and Subdifferential

For a function $\phi(x) : \mathbb{R}^n \to \mathbb{R}$, $(\partial\phi(x)/\partial x_i)$ denotes the partial derivative of $\phi(x)$ at point $x = (x_1, x_2, \ldots, x_n)^T$ with respect to x_i. If all the partial derivatives $(\partial\phi(x)/\partial x_i)$ $(i = 1, 2, \ldots, n)$ exist at point x, then the gradient of $\phi(x)$ exists at point x and is denoted by

$$\nabla\phi(x) = \left(\frac{\partial\phi(x)}{\partial x_1}, \frac{\partial\phi(x)}{\partial x_2}, \ldots, \frac{\partial\phi(x)}{\partial x_n}\right)^T.$$

The gradient plays an important role in the neural network model introduced in Section V. However, for some nonsmooth functions, their gradients do not exist at some point. For example, $f(x) = \|x\|_2$ with $x \in \mathbb{R}^2$ is nonsmooth at point $x = 0_2$. To find an alternative of gradient for nonsmooth functions, the concept of subgradient is introduced. A vector $d \in \mathbb{R}^n$ is a subgradient of $\phi(x)$ at x if

$$\phi(x + v) - \phi(x) \geq d^T v \quad \forall v \in \mathbb{R}^n.$$

The set of all subgradients of ϕ at x is call the subdifferential of ϕ at x, which is denoted by

$$\partial\phi(x) = \{d \in \mathbb{R}^n | \phi(x + v) - \phi(x) \geq d^T v \quad \forall v \in \mathbb{R}^n\}. \quad (2)$$

Lemma 1 [52], [53]: If $\phi(x)$ is a convex function, then $\partial\phi(x) \neq \varnothing$. Furthermore, if the gradient of ϕ exists at x,

then $\partial\phi(x) = \nabla\phi(x)$. In reverse, if the subdifferential of ϕ at x contains exactly one subgradient, then the gradient of ϕ exists at x and is equal to the subgradient.

For example, it is easy to see $f(x) = \|x\|_2$ with $x = (x_1, x_2)^T \in \mathbb{R}^2$ is a convex function and is nonsmooth only at point $x = (0,0)^T$. Therefore, by Lemma 1 and (2), it can be calculated that the subgradient of $f(x)$ is

$$\partial f(x) = \begin{cases} \left(\dfrac{x_1}{\|x\|_2}, \dfrac{x_2}{\|x\|_2}\right)^T, & x \neq (0,0)^T \\ \left\{(c_1, c_2)^T | c_1^2 + c_2^2 \leq 1\right\}, & x = (0,0)^T \end{cases} \tag{3}$$

where c_1 and c_2 are two real constants.

C. Penalty Method

The penalty method is a class of algorithms for solving constrained optimization problems. Since the unconstrained optimization problems are relatively easier to solve, the penalty methods seek the solution by replacing the original constrained optimization problem with a sequence of unconstrained optimization problems. Consider the following constrained optimization problem:

$$\min_x \quad f(x) \tag{4a}$$
$$\text{s.t.} \quad c_i(x) = 0, \quad i \in \mathcal{E} \tag{4b}$$
$$c_i(x) \geq 0, \quad i \in \mathcal{I}. \tag{4c}$$

The penalty function is defined as

$$Q(x, M) = f(x) + M \sum_{i \in \mathcal{E}} c_i^2(x) + M \sum_{i \in \mathcal{I}} (\min\{0, c_i(x)\})^2 \tag{5}$$

where $M > 0$ is called the penalty parameter.

Lemma 2 [54, Th. 17.1]: For a given penalty parameter sequence $\{M_k\}$, suppose x_k is the exactly global minimizer of (5) with $M = M_k$. Assume x^* is a limit point of $\{x_k\}$. Then, x^* is an optimal solution to (4) if $\lim_{k\to\infty} M_k = \infty$.

Consider another constrained optimization problem

$$\min_x \quad f(x) \tag{6a}$$
$$\text{s.t.} \quad g_i(x) \geq 0, \quad i = 1, 2, \ldots, m \tag{6b}$$
$$Ax = b \tag{6c}$$
$$h_j(x) = 0, \quad j = 1, 2, \ldots, l. \tag{6d}$$

If $f(x)$ and $-g_i(x)$ are all convex functions and there is no nonlinear equality constraint (in the other word $l = 0$), then the optimization problem (6) is a convex optimization problem [54]. In general, it is much easier to solve a convex optimization problem than a nonconvex optimization problem. To eliminate the nonlinear equality constraints, (6) is transformed into the following problem by the penalty function method:

$$\min_x \quad f(x) + M \sum_{j=1}^{l} h_j^2(x) \tag{7a}$$
$$\text{s.t.} \quad g_i(x) \geq 0, \quad i = 1, 2, \ldots, m \tag{7b}$$
$$Ax = b \tag{7c}$$

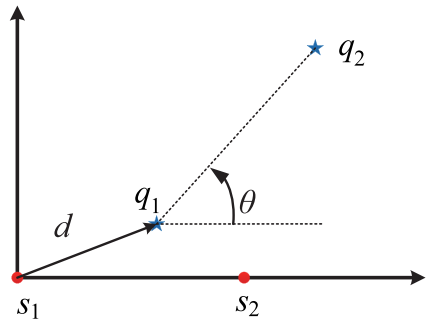

Fig. 1. Relationship between S and Q in the world frame $\mathcal{W}$.

where $M > 0$ is the penalty parameter. It is noted that (7) is a convex optimization problem if all $h_j(x)$ are convex functions.

The following Lemma is a direct extension to Lemma 2.

Lemma 3: For a given penalty parameter sequence $\{M_k\}$, suppose x_k is the exactly global solution of (7) with $M = M_k$. Assume x^* is a limit point of $\{x_k\}$ and $h_j(x)$ $(j = 1, \ldots, l)$ are continuous functions. Then, x^* is an optimal solution to (6) if $\lim_{k\to\infty} M_k = \infty$.

Proof: Please see the proof in the Appendix. ■

Lemma 3 provides an approach to obtain an approximation solution to the original nonconvex optimization problem. If the penalty parameter M_k is sufficiently large, then x_k can be seen as an approximate solution. Moreover, the approximation error can be arbitrarily close to zero.

III. Problem Formulation

The multirobot system is required to achieve a formation, which has the same shape as a given formation (we call it the shape icon). The shape icon is composed of n points $\{s_1, \ldots, s_n\}$ $[s_i = (s_i^x, x_i^y)^T$ is the Cartesian coordinate of the ith point in the world frame $\mathcal{W}]$ and is denoted by $S = (s_1, \ldots, s_n) \in \mathbb{R}^{2\times n}$.

Let $P = (p_1, p_2, \ldots, p_n) \in \mathbb{R}^{2\times n}$ denote the initial formation of the multirobot system. There must exist a new formation $Q \in [S](Q = (q_1, q_2, \ldots, q_n))$ which minimizes $\|P - Q\|$ where $\|\cdot\|$ is certain matrix norm. In general, $\|P - Q\|$ can be viewed as the distance between P and Q. Then, the formation problem is equivalent to the following constrained optimization problem:

$$\min_{Q \in \mathbb{R}^{2\times n}} \quad \|P - Q\| \tag{8a}$$
$$\text{s.t.} \quad Q \in [S] \quad (\text{or } Q \sim S). \tag{8b}$$

It is noted that the constraint $Q \in [S]$ in (8) is difficult to deal with. In [20], a method of translating this constraint into a standard linear constraint is provided. The following part gives the detailed description of this method.

For any formation $Q \in [S]$, by (1), we know that there must exist a scale factor α, a rotation matrix R, and a translation vector d such that $q_i = \alpha R s_i + d$. Without loss of generality, it is assumed that s_1 is at the origin of $\mathcal{W}$ and s_2 is on the x-axis of $\mathcal{W}$, which is illustrated in Fig. 1. From this figure,

it can be calculated by the basic geometric principle that

$$\begin{cases} d = q_1 \\ \alpha = \|q_2 - q_1\|_2 / \|s_2\|_2 \\ R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \\ \theta = \arctan\left(q_2^y - q_1^y, q_2^x - q_1^x\right). \end{cases} \tag{9}$$

By (9), the constraint $Q \in [S]$ in (8) can be transformed into the following linear equality constraint $Aq = 0_{2(n-2)}$, where $q = (q_1^T, q_2^T, \ldots, q_n^T)^T \in \mathbb{R}^{2n}$:

$$A = \begin{bmatrix} -A_1 + A_3 & -A_3 & A_1 & \Theta_2 & \cdots & \Theta_2 \\ -A_1 + A_4 & -A_4 & \Theta_2 & A_1 & \cdots & \Theta_2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -A_1 + A_n & -A_n & \Theta_2 & \Theta_2 & \cdots & A_1 \end{bmatrix}$$

and

$$A_1 = \begin{bmatrix} \|s_2\|_2 & 0 \\ 0 & \|s_2\|_2 \end{bmatrix}, \quad A_i = \begin{bmatrix} s_i^x & -s_i^y \\ s_i^y & s_i^x \end{bmatrix}, \quad \Theta_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Hence, the optimization problem (8) can be rewritten in the following form:

$$\min_{Q \in \mathbb{R}^{2\times n}} \quad \|P - Q\| \tag{10a}$$

$$\text{s.t.} \quad Aq = 0 \tag{10b}$$

where $q = [q_1^T, q_2^T, \ldots, q_n^T]^T$.

Remark 1: The matrix norm $\|\cdot\|$ can be defined in different senses. For example, $\|X\| = \sum_{i=1}^n \|x_i\|_2$ or $\|X\| = \max_i \|x_i\|_2$, where $X = (x_1, \ldots, x_n) \in \mathbb{R}^{2\times n}$. If $\|X\| = \sum_{i=1}^n \|x_i\|_2$, the optimization problem (8) is equivalent to minimizing the total distance that all robots travel. From the practical point of view, this definition may result in the least amount of energy consumption. If $\|X\| = \max_i \|x_i\|_2$, the optimization problem (8) is equivalent to minimizing the maximal distance that one robot can possibly travel. In this definition, the time for achieving certain formation is minimized because the time for achieving a formation is determined by the robot who travels the maximal distance provided speeds of all robots are the same. For the convenience of easy understanding, it is assumed that $\|X\| = \sum_{i=1}^n \|x_i\|_2$ throughout this paper.

Remark 2: For any norm $\|\cdot\|$, it is worth noting that $f(x) = \|x\|$ is a convex function. Consequently, the optimization problem (10) is a standard convex optimization problem. In addition, by the same analysis of (3), it can be obtained that the objective function of (10) is nonsmooth.

By the above procedure, the formation problem is transformed into a nonsmooth convex optimization problem. However, the desired formation in practice may have some particular requirements. For example, the multirobot system can only work in a predesigned range (safe range); if the system is out of this range, it may be apt to be attacked. The following section introduces how to add some practical constraints (the orientation, the scale, and the admissible range of the formation) on the desired formation.

1) *Admissible Range:* It is assumed that the desired formation is located in a predesigned convex set $\Omega \subset \mathbb{R}^2$.

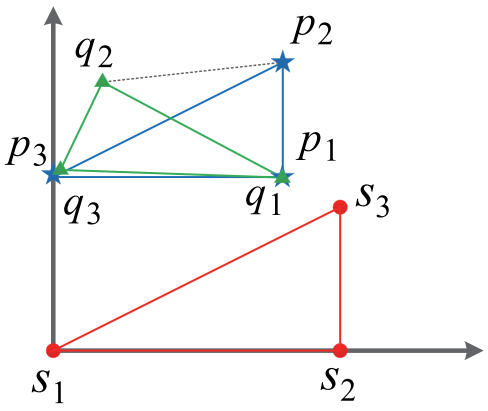

Fig. 2. Triangle formation composed of three robots where all three robots are functionally identical.

In this case, the optimization problem (10) can be written in the following form:

$$\min_{Q} \quad \|P - Q\| \tag{11a}$$

$$\text{s.t.} \quad Aq = 0 \tag{11b}$$

$$q_i \in \Omega, \quad i = 1, 2, \ldots, n. \tag{11c}$$

2) *Orientation and Scale:* It is assumed that the orientation of the desired formation should be a specific angle θ_0. From Fig. 1, it can be seen that the orientation is described by the angle θ. Then, by (9), the rotation matrix is calculated as follows:

$$R_0 = \begin{bmatrix} \cos(\theta_0) & -\sin(\theta_0) \\ \sin(\theta_0) & \cos(\theta_0) \end{bmatrix}.$$

Hence, the orientation constraint is equivalent to

$$q_i - q_1 = \alpha R_0 s_i, \quad i = 2, \ldots, n.$$

It is also worth noting that the scale factor α is also a free parameter in this case. Here, the scale α of the desired formation is assumed to satisfy that $\alpha_{\min} \le \alpha \le \alpha_{\max}$. By summarizing the above analysis, the formation problem with practical constraints can be reformulated into the following optimization problem:

$$\min_{Q} \quad \|P - Q\| \tag{12a}$$

$$\text{s.t.} \quad \bar{A}q = \alpha(I_{n-1} \otimes R_0)s \tag{12b}$$

$$\alpha_{\min} \le \alpha \le \alpha_{\max} \tag{12c}$$

$$q_i \in \Omega, \quad i = 1, 2, \ldots, n \tag{12d}$$

where $\bar{A} = [-1_{n-1} \otimes I_2, \ I_{2n-2}]$ and $s = (s_2^T, \ldots, s_n^T)^T$.

IV. Extended Formation Problems

In this section, the extended formation problem is discussed, where each robot is not specific to one fixed position in the formation. First, let us look at an example shown in Fig. 2. There are three identical robots whose initial formation is

$$P = [p_1, p_2, p_3] = \begin{bmatrix} 8 & 8 & 0 \\ 6 & 10 & 6 \end{bmatrix}.$$

The goal here is to achieve a formation $Q = [q_1, q_2, q_3]$, which has the same shape as a given shape icon

$$S = \begin{bmatrix} 0 & 10 & 10 \\ 0 & 0 & 5 \end{bmatrix}.$$

Intuitively, formation P and formation S have the same shape, because $\triangle s_1 s_2 s_3 \sim \triangle p_3 p_2 p_1$. However, by the shape theory, we know that the shape of P and the shape of S are different ($P \nsim S$). This is because, in the definition (1), the point p_i corresponds to the point s_i ($i = 1, 2, 3$). From one side, by solving the formation problem defined by (10), we can obtain an optimal formation

$$Q = \begin{bmatrix} 7.9995 & 1.6367 & 0.0339 \\ 5.9988 & 9.2044 & 6.0230 \end{bmatrix}.$$

From the other side, if all three robots are functionally identical, their positions in the system are exchangeable. In this case, the multirobot system is actually unnecessary to move from the formation P to Q since the triangles P and S are similar.

To deal with this dilemma, the following transformation matrix $X \in \mathbb{R}^{n\times n}$ is introduced:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}$$

where $x_{ij} \in \{0, 1\}$, $\sum_{i=1}^{n} x_{ij} = n$ and $\sum_{j=1}^{n} x_{ij} = n$. Let $T(n)$ denote the set of all such n-dimensional matrices. Then, for a given formation $S \in \mathbb{R}^{2\times n}$, the set of formations which have the same shape as S is extended as follows:

$$[S]_e = \left\{\alpha RSX + 1_n^T \otimes d | \alpha > 0, R \in \mathrm{SO}(2), X \in T(n), d \in \mathbb{R}^2\right\}. \tag{13}$$

As a result, the extended optimal formation problem (8) is

$$\min_{Q\in\mathbb{R}^{2\times n}} \quad \|P - Q\| \tag{14a}$$
$$\text{s.t.} \quad Q \in [S]_e. \tag{14b}$$

The constraint $Q \in [S]_e$ means that there exist a scale factor α, a rotation matrix R, a transformation matrix $X \in T(n)$, and a vector $d \in \mathbb{R}^2$ such that $Q = \alpha RSX + 1_n^T \otimes d$. It is noted that $\forall X \in T(n)$, $X^{-1} = X$ and $(1_n \otimes d)X = 1 \otimes d$. Let $\bar{Q} \triangleq QX^{-1} = [\bar{q}_1, \bar{q}_2, \ldots, \bar{q}_n]$ then $\bar{Q} = \alpha RS + 1_n^T \otimes d \in [S]$.

It is easy to see that $\bar{Q} \in [S]$ and $X \in T(n)$ minimize $\|P - \bar{Q}X\|$ if and only if $\bar{Q} \in [S]$ and $X \in T(n)$ minimize $\|PX - \bar{Q}\|$. It is easy to see that $\bar{Q} \in [S]$ minimizes $\|P - \bar{Q}X\|$ if and only if $\bar{Q} \in [S]$ minimizes $\|PX^{-1} - \bar{Q}\|$. Moreover, $\|P - \bar{Q}X\|$ is equal to $\|P - Q\|$. Therefore, the extended optimal formation problem (14) can be reformulated as

$$\min_{\bar{Q}\in\mathbb{R}^{2\times n}} \quad \|PX - \bar{Q}\| \tag{15a}$$
$$\text{s.t.} \quad \bar{Q} \in [S] \tag{15b}$$
$$X \in T(n). \tag{15c}$$

By the transformation method described in Section III, the optimal formation problem (14) is further reformulated as

$$\min_{\bar{Q}\in\mathbb{R}^{2\times n}} \quad \|PX - \bar{Q}\| \tag{16a}$$
$$\text{s.t.} \quad A\bar{q} = 0 \tag{16b}$$
$$x_{ij} \in \{0, 1\}, \quad i, j = 1, \ldots, n \tag{16c}$$
$$\sum_{i=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1 \tag{16d}$$

where $\bar{q} = (\bar{q}_1^T, \bar{q}_2^T, \ldots, \bar{q}_n^T)^T$.

In fact, X gives the corresponding relationship between the robot and its position in the formation. The optimal formation problem (10) is just a special case of (16) with $X = I_n$. The extended formation problem (16) is essentially a combinational optimization problem. This problem can be solved by the enumeration method if the number of robots is relatively small. However, the complexity of (16) grows dramatically with the increase of the number of robots.

The optimization problem (16) is not a convex optimization problem, which further increases the difficulty of solving this problem. However, the nonconvex optimization problem (16) can be transformed into a convex optimization problem using the penalty method introduced in Section II-C.

Lemma 4: Constraints (16c) and (16d) are equivalent to the following n^2 inequality constraints and $4n$ equality constraints:

$$x_{ij} \geq 0, \quad i, j = 1, 2, \ldots, n \tag{17}$$
$$\sum_{j=1}^{n} x_{ij} = 1, \quad \sum_{i=1}^{n} x_{ij} = 1 \tag{18}$$
$$\sum_{j=1}^{n} x_{ij}^2 = 1, \quad \sum_{i=1}^{n} x_{ij}^2 = 1. \tag{19}$$

Proof: Squaring both sides of (18) obtains

$$\begin{cases} \sum_{j=1}^{n} x_{ij}^2 + \sum_{k\neq l} x_{ik}x_{il} = 1, \quad i = 1, \ldots, n \\ \sum_{i=1}^{n} x_{ij}^2 + \sum_{k\neq l} x_{kj}x_{lj} = 1, \quad j = 1, \ldots, n. \end{cases} \tag{20}$$

Substituting (19) into (21) obtains that

$$\begin{cases} \sum_{k\neq l} x_{ik}x_{il} = 0, \quad i = 1, \ldots, n \\ \sum_{k\neq l} x_{kj}x_{lj} = 0, \quad j = 1, \ldots, n \end{cases} \tag{21}$$

which together with (17) leads to that there are at least $n - 1$ zero entries in each row of X and $n - 1$ zero entries in each column of X. Therefore, there is one and only one 1 in each row of X and one. In addition, there is only one 1 in each column of X. Then, conditions (17)–(19) are compatible with constraints (16c) and (16d). ■

By Lemmas 3 and 4, the optimal solution to (16) can be approximated by the following convex

optimization problem:

$$\begin{aligned}\min\ & \|PX-\bar{Q}\| + M\sum_{i=1}^{n}\left[\left(\sum_{j=1}^{n}x_{ij}-1\right)^2+\left(\sum_{j=1}^{n}x_{ij}^2-1\right)^2\right] \\ & + M\sum_{j=1}^{n}\left[\left(\sum_{i=1}^{n}x_{ij}-1\right)^2+\left(\sum_{i=1}^{n}x_{ij}^2-1\right)^2\right] \\ \text{s.t. } & A\bar{q}=0 \\ & x_{ij}\geq 0,\quad i,j=1,\ldots,n \end{aligned} \tag{22}$$

where $M>0$ is a constant which should be a relatively large.

Remark 3: By solving the optimization problem (22), one can obtain an approximated solution (Q_a, X_a) to the optimization problem (16). Since every entry of X can only be 0 or 1, the optimal solution X^* to the optimization problem (16) can be deduced from the approximated solution X_a. By substituting $X = X^*$ into (16), the combinational optimization problem (16) is transformed into the following common convex optimization problem:

$$\min_{\bar{Q}\in\mathbb{R}^{2\times n}} \quad \|PX^*-\bar{Q}\| \tag{23a}$$

$$\text{s.t.} \quad A\bar{q}=0. \tag{23b}$$

Therefore, one can obtain the optimal solution $\bar{Q}^*$ to (16) by solving (23). Then, the optimal solution to (14) is $Q^*=\bar{Q}^*X^*$.

V. Recurrent Neural Network Model

If the size n of the multirobot system is a very large, the optimization problems discussed in Sections III and IV are difficult to be solved by the conventional numerical algorithm. In particular, the combination optimization problem (16) is almost impossible to be solved by the conventional numerical algorithm. As discussed in Section I-B, the recurrent neural network paves another avenue for solving optimization problems effectively. With the aid of hardware implementation, the recurrent neural network can solve optimization problems in real time. In this section, a recurrent neural network model, which is capable for solving nonsmooth optimization problems discussed in Sections III and IV, is introduced.

Consider the following convex optimization problem:

$$\min_{x} \quad f(x) \tag{24a}$$

$$\text{s.t.} \quad g(x)\geq 0_m \tag{24b}$$

$$Cx=b \tag{24c}$$

$$x\in\Phi \tag{24d}$$

where $x=(x_1,x_2,\ldots,x_n)^T\in\mathbb{R}^n$ is the decision variable; $f(x):\mathbb{R}^n\to\mathbb{R}$ is the objective function; $g(x)=(g_1(x),g_2(x),\ldots,g_m(x))^T:\mathbb{R}^n\to\mathbb{R}^m$ are the inequality constraints; $C\in\mathbb{R}^{p\times n}$ is a row full rank matrix; $b\in\mathbb{R}^p$ is a constant vector; and $\Phi\subset\mathbb{R}^n$ is a convex admissible set for the decision variable. It is assumed that $f(x)$ and $-g_i(x)$ are all convex functions on Φ, however, they are not necessary to be smooth. Moreover, the feasible set of (24) is nonempty and the optimization problem (24) has at least one finite optimal solution x^*. In addition, there exists a feasible point $\tilde{x}\in\text{int}(\Phi)$ such that $g(\tilde{x})>0_m$ and $C\tilde{x}=d$.

To deal with (24), a recurrent neural network model was proposed in [49], which can be described by the following differential inclusion:

$$\frac{dx}{dt}\in -2\beta(x-\hat{x}) \tag{25a}$$

$$\frac{d\lambda}{dt}=-\beta(\lambda-\tilde{\lambda}) \tag{25b}$$

$$\frac{d\mu}{dt}=-\beta(Cx-b) \tag{25c}$$

where $\beta>0$ is the scaling constant; $\lambda\in\mathbb{R}^m$; $\mu\in\mathbb{R}^p$; $\hat{x}=\{F_\Phi(x-\gamma+G^T\tilde{\lambda}+C^T(\mu-Cx+b))|\gamma\in\partial f(x), G\in G(x)\}$; $G(x)=[v_1(x),v_2(x),\ldots,v_m(x)]^T\in\mathbb{R}^{m\times n}$; $\tilde{\lambda}=(\lambda-g(x))^+=\max\{0,\lambda-g(x)\}$; $v_i(x)\in\partial g_i(x)$ $(i=1,2,\ldots,m)$; $\partial f(x),\partial g_1(x),\ldots,\partial g_m(x)$ are the subdifferentials of $f(x),g_1(x),\ldots,g_m(x)$, respectively; and $F_\Phi(x):\mathbb{R}^n\to\Phi$ is the projection operator defined by

$$F_\Phi(x)=\arg\min_{y\in\Phi}\|x-y\|. \tag{26}$$

Remark 4: Since $f(x)$ and $g_i(x)$ may be nonsmooth at some points, the recurrent neural network must be modeled by the differential inclusion rather than the differential equation. Here, the mathematical symbol $\in$ means that the time derivative of x can be any value in the set described by the right side of (25a).

By the convergence analysis in [49], for any initial state $(x^T(0),\lambda^T(0),\mu^T(0))^T$ of the recurrent neural network defined by (25), its state trajectory $(x^T(t),\lambda^T(t),\mu^T(t))^T$ is convergent to an equilibrium point. In addition, the equilibrium point is an optimal solution to the optimization problem (24).

Since (10)–(12) are all nonsmooth convex optimization problems, they can be solved by (25) with specific parameters. For example, the following recurrent neural network can solve the optimization problem (10):

$$\frac{dq}{dt}\in -2\beta(\partial\|P-Q\|-A^T(\mu-Aq)) \tag{27a}$$

$$\frac{d\mu}{dt}=-\beta Aq \tag{27b}$$

where $\mu\in\mathbb{R}^{2n-4}$ and

$$\partial\|P-Q\|=[\partial^T\|p_1-q_1\|_2,\ldots,\partial^T\|p_n-q_n\|_2]^T$$

$$\partial\|p_i-q_i\|_2=\begin{cases}\left(\dfrac{q_i^x-p_i^x}{\|p_i-q_i\|_2},\dfrac{q_i^y-p_i^y}{\|p_i-q_i\|_2}\right)^T, & q_i\neq p_i \\ \left\{(c_1,c_2)^T|c_1^2+c_2^2\leq 1\right\}, & q_i=p_i.\end{cases}$$

Analogously, the optimization problem (11) can be solved by the following recurrent neural network:

$$\frac{dq}{dt}\in -2\beta(q-F_{\bar{\Omega}}(q-\partial\|P-Q\|+A^T(\mu-Aq))) \tag{28a}$$

$$\frac{d\mu}{dt}=-\beta Aq \tag{28b}$$

where $F_{\bar{\Omega}}$ is the projection operator defined by (26) and $\bar{\Omega}=\{(x_1^T,x_2^T,\ldots,x_n^T)^T|x_i\in\Omega\}$. The recurrent neural

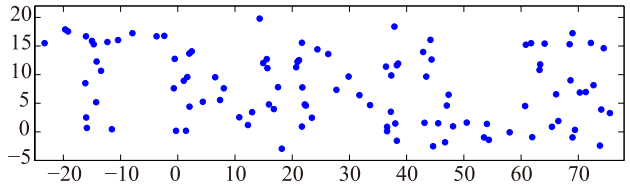

Fig. 3. Initial formation of the multirobot system in Simulation Example 1.

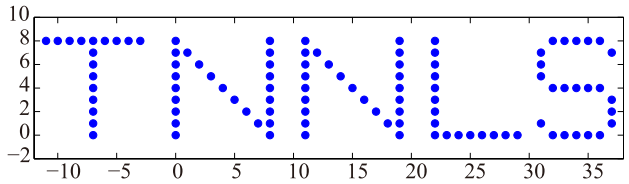

Fig. 4. Reference shape icon TNNLS.

network for solving the optimization problem (12) is designed as follows:

$$\frac{d\hat{q}}{dt} \in -2\beta(\hat{q} - F_{\hat{\Omega}}(\hat{q} - \gamma + \hat{A}^T(\mu - \hat{A}\hat{q}))) \quad (29a)$$

$$\frac{d\mu}{dt} = -\beta\hat{A}\hat{q} \quad (29b)$$

where $\hat{q} = (q^T, \alpha)^T$, $\gamma = (\partial^T\|P - Q\|, 0)^T$, $\hat{A} = [\bar{A}, -(I_{n-1} \otimes R_0)s]$, and $\hat{\Omega} = \{(x_1^T, \ldots, x_n^T, \alpha)^T | x_i \in \Omega, \quad \alpha_{\min} \leq \alpha \leq \alpha_{\max}\}$.

Likewise, the optimization problem (22) can be solved by the following neural network:

$$\frac{d\xi}{dt} \in -2\beta(\partial f(\xi) - G^T\tilde{\lambda} - A^T(\mu - \bar{A}\xi)) \quad (30a)$$

$$\frac{d\lambda}{dt} = -\beta(\lambda - \tilde{\lambda}) \quad (30b)$$

$$\frac{d\mu}{dt} = -\beta\bar{A}\xi \quad (30c)$$

where $\xi = [\bar{q}^T, x_{11}, \ldots, x_{1n}, x_{2n}, \ldots, x_{nn}]^T = [\xi_1, \ldots, \xi_{n^2+2n}]^T$; $\lambda \in \mathbb{R}^{n^2}$, $\tilde{\lambda} = (\lambda - [\xi_{2n+1}, \ldots, \xi_{2n+n^2}]^T)^+$; $\mu \in \mathbb{R}^{2n-4}$; $\bar{A} = [A, \Theta_{(2n-4)\times n^2}]$; $G = [\Theta_{n^2\times 2n}, I_{n^2}]$; and

$$f(\xi) = \|PX - \bar{Q}\| + M\sum_{i=1}^{n}\left[\left(\sum_{j=1}^{n}x_{ij} - 1\right)^2 + \left(\sum_{j=1}^{n}x_{ij}^2 - 1\right)^2\right] + M\sum_{j=1}^{n}\left[\left(\sum_{i=1}^{n}x_{ij} - 1\right)^2 + \left(\sum_{i=1}^{n}x_{ij}^2 - 1\right)^2\right].$$

VI. Simulations and Experiments

In this section, we provide three simulation examples and two experiments to verify the effectiveness of the proposed approach.

A. Simulation Example 1

Consider a multirobot system composed of 106 robots in a 2-D plane. The dynamics of the robots is not considered in this paper. Each robot is regarded as a particle and denoted by its coordinate. The initial formation of the mutlirobot system is shown in Fig. 3. The mission is to seek a new formation for the multirobot system, which has the same shape as the shape icon illustrated in Fig. 4. In addition, the new formation should minimize the distance that all robots travel. This formation problem can be dealt with by solving the optimization problem (10). The neural network (27) is used to solve this optimization problem. The MATLAB ode solver is employed to simulate this neural network. The scaling constant β is set to be 20. In this simulation, if $q_i = p_i$, then let $\partial\|p_i - q_i\|_2 = 0$. Let $s(t) = (q^T(t), \mu^T(t))^T$ denote the state vector of the neural network. By [49], the neural network has an equilibrium point $s^* = (q^{*T}, \mu^{*T})^T$, where q^* is the optimal solution to the optimization problem (10).

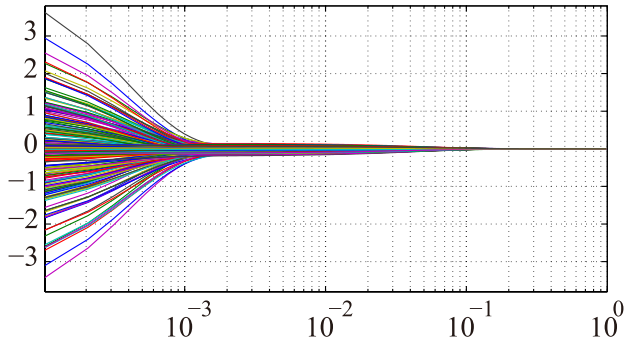

Fig. 5. Trajectory of $s(t) - s*$, where $s(t)$ and s^* are the state vector and equilibrium point of the recurrent neural network, respectively.

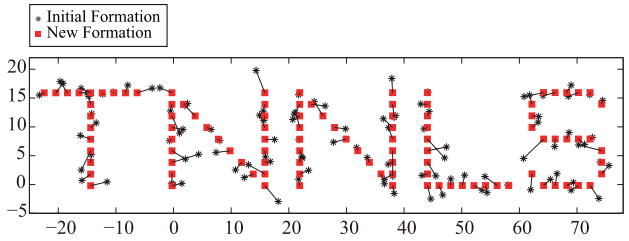

Fig. 6. Obtained optimal formation in Simulation Example 1.

The simulation results show that the state vector $s(t)$ is convergent to a constant vector which is the equilibrium point s^*. Fig. 5 gives the trajectory of the difference between the state vector $s(t)$ and the equilibrium point s^*, namely, $s(t) - s^*$. From Fig. 5, the state vector $s(t)$ of the recurrent neural network is convergent to the equilibrium point s^* within 1 s. This means that if the recurrent neural network is implemented in hardware, the computation time is <1 s. Fig. 6 shows that the obtained formation has the same shape as the shape icon TNNLS shown in Fig. 4. It is worth noting that the convergence rate of the neural network can be further speeded up by increasing the scaling constant β.

B. Simulation Example 2

Consider the same multirobot system as the one in Simulation Example 1. In addition, the reference shape icon for the

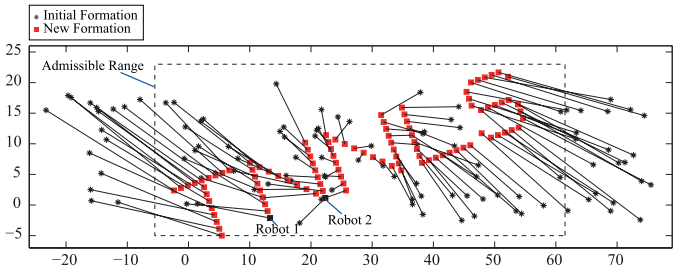

Fig. 7. Optimal formation with the admissible range, the orientation, and the scale constraints in Simulation Example 2.

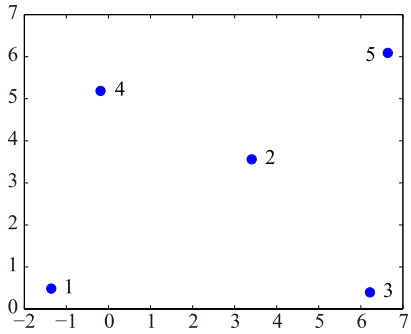

Fig. 8. Initial formation of the multirobot system in Simulation Example 3.

multirobot system is the same as the one in Example 1. This section requires that all robots work in a rectangle area R. In the other word, the new formation of the multirobot system should be in this rectangle admissible range. The center of this rectangle is $(28, 9)$. The height and width of this rectangle are 28 and 67, respectively. Meanwhile, the constraints on the formation's orientation and scale are also considered. The orientation and the scale of the new formation are required to be 20° and 1.2°, respectively. The desired formation can be achieved by solving the optimization problem (12) with $\theta = 20°$, $\alpha_{\min} = \alpha_{\max} = 1.2$, and $\Omega = R$. The recurrent neural network (29) is used to solve this optimization problem.

The obtained formation is shown in Fig. 7. The coordinates of robot 1 and robot 2 in the formation are $q_1 = (13.3252, -2.1271)$ and $q_2 = (22.3463, 1.1563)$, respectively. Hence, the orientation of this formation is $\arctan\ (q_2^y - q_1^y, q_2^x - q_1^x) = \arctan\ (3.2834, 9.0210) = 20.00°$ and the scale is $\alpha = \|q_2 - q_1\|_2 / \|s_2\|_2 = 9.60/8 = 1.20$. Therefore, the new formation satisfies the constraints on orientation and scale.

C. Simulation Example 3

In this example, we consider a multirobot systems composed of five robots. The initial formation of this multirobot system is shown in Fig. 8. The multirobot system is required to achieve a new formation which has the same shape as the shape icon shown in Fig. 9. The new formation obtained by employing the recurrent neural network (27) is shown in Fig. 10. To achieve this new formation, the total distance that all robots should travel is 10.21.

Next, we assume that all robots are functional identical. In this case, the new formation shown in Fig. 10 may not be the optimal formation. From Section IV, this formation problem can be formulated as (16), which is a combinational

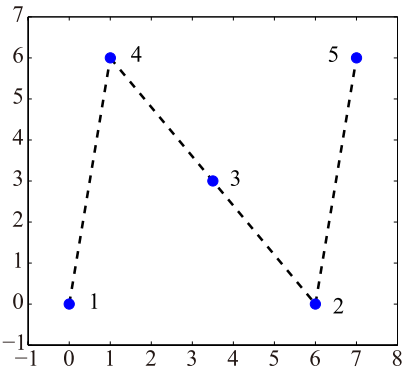

Fig. 9. Reference icon N.

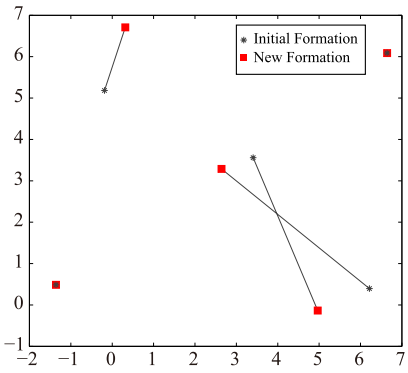

Fig. 10. Optimal formation obtained by applying the neural network (27) in Example 3.

optimization problem. An approximated solution (X_a, Q_a) to (16) can be obtained by solving (22). In this example, the recurrent neural network (30) is used to solve the optimization problem (22). In addition, the solution is

$$X_a = \begin{bmatrix} 0.95 & 0.09 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.96 & 0.04 & 0.00 \\ 0.21 & 0.92 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.06 & 0.00 & 0.98 & 0.06 \\ 0.02 & 0.00 & 0.00 & 0.15 & 0.97 \end{bmatrix}. \quad (31)$$

By Remark 3, the optimal solution X^* to (16) can be deduced from X_a. In addition, it is

$$X^* = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (32)$$

By substituting X^* into (16), the combinational optimization problem (16) is transformed into a common convex optimization problem (23), which can be solved by the recurrent neural network (27). The obtained optimal formation is shown in Fig. 11. To achieve this formation, the total distance that all robots travel is 3.49, which is much smaller than the total distance that all robots travel to achieve the formation shown in Fig. 10.

D. Experiments

In this section, the simulation results of Example 3 are implemented on a group of five e-puck mobile robots,

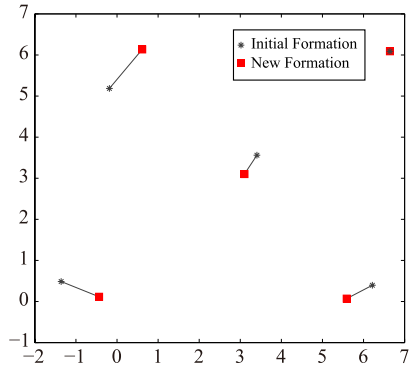

Fig. 11. Optimal formation in Example 3 in the case that all robots are functionally identical.

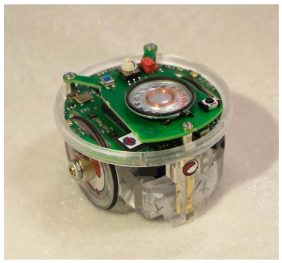

Fig. 12. e-puck mobile robot.

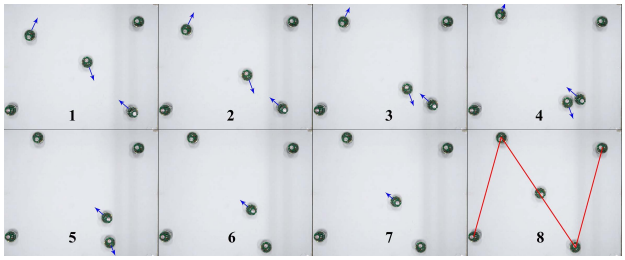

Fig. 13. Snapshots of Experiment I. Blue arrows: directions of motions of the e-puck robots.

one of which is shown in Fig. 12. Each e-puck robot is equipped with two encoders to measure the travel distance. At the beginning of the experiments, each e-puck robot is placed at its initial position shown in Fig. 8. The e-puck robots are equipped with Bluetooth modules. They can communicate with other devices via the Bluetooth. A personal computer equipped with the Bluetooth module sends the information of the final positions to the e-puck robots via the Bluetooth.

1) *Experiment I:* The group of e-puck robots are required to achieve the new formation shown in Fig. 10. The snapshots of this experiment are illustrated in Fig. 13.
2) *Experiment II:* All e-puck robots are assumed to be functionally identical. Hence, the optimal formation for the group of robots is the one shown in Fig. 11. The snapshots of this experiment are shown in Fig. 14.

In both experiments, the e-puck robots successfully accomplish the formation task. The travel distance of each robot is given in Table I. It is easy to calculated that the total distances of all robots travel in Experiments I and II are 10.01 dm and 3.38 dm, respectively. This is consistent with the simulation results of Simulation Example 3, within the range of error permission. Obviously, the total distance that all robots travel is much smaller in the case that all robots are functionally identical.

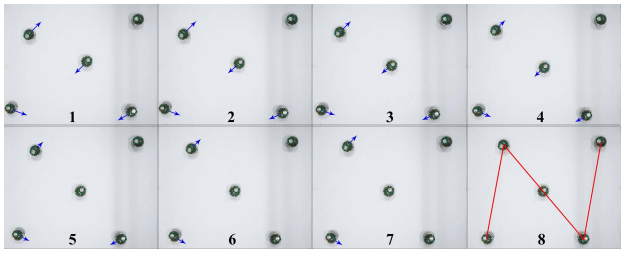

Fig. 14. Snapshots of Experiment II. Blue arrows: the directions of motions of the e-puck robots.

TABLE I
TRAVEL DISTANCE OF EACH e-PUCK ROBOT IN TWO EXPERIMENTS

Robot	Experiment I (dm)	Experiment II (dm)
1	0	0.97
2	3.89	0.47
3	4.50	0.71
4	1.53	1.23
5	0	0

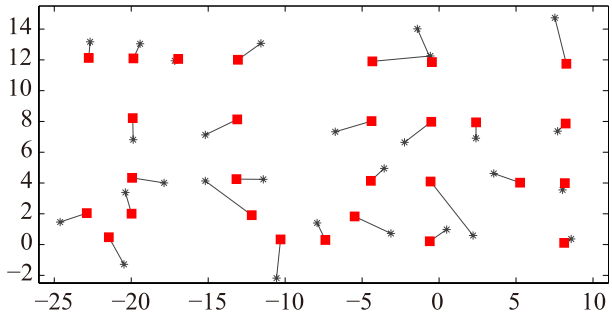

Fig. 15. "JUN" formation made of 28 robots.

VII. CONCLUSION

In this paper, an optimal formation problem is transformed into a nonsmooth convex optimization problem, which is then solved by a recurrent neural network proposed in [49]. Compared with the existing work, the recurrent neural network based approach can solve the large-scale optimal formation problems more effectively due to its parallel computation nature; and the orientation, the scale, and the admissible range of the desired formation are also considered in this paper. In addition, the optimal formation problem is extended to be a combinational optimization problem when the positions of all robots in the formation are exchangeable. By combining the penalty method and the recurrent neural network, this extended formation problem can be satisfactorily solved as well. Finally, the theoretical analysis is verified by several simulation and experiment examples.

APPENDIX
PROOF OF LEMMA 3

Let $\bar{x}$ be a global optimal solution to the problem (6), then $\bar{x}$ is also a feasible solution to problem (7). Hence

$$f(x_k) \le f(x_k) + M_k \sum_{j=1}^{n} h_j^2(x_k)$$
$$\le f(\bar{x}) + M_k \sum_{j=1}^{n} h_j^2(\bar{x}) = f(\bar{x}) \qquad (33)$$

which follows that:

$$\sum_{j=1}^{n} h_j^2(x_k) \leq \frac{f(\bar{x}) - f(x_k)}{M_k}. \tag{34}$$

Without loss of generality, assume that $\lim_{k\to\infty} x_k = x^*$. By taking the limitation operation ($k \to \infty$) on both sides of (34), it is obtained that

$$\sum_{j=1}^{n} h_j^2(x^*) = \lim_{k\to\infty} \sum_{j=1}^{n} h_j^2(x_k) \leq \lim_{k\to\infty} \frac{f(\bar{x}) - f(x_k)}{M_k} = 0 \tag{35}$$

which leads to that x^* is a feasible solution to problem (6). In addition, by taking the limitation operation on both sides of (33), it is obtained that

$$f(x^*) \leq f(\bar{x}). \tag{36}$$

That is to say, x^* is also a global optimal solution to problem (6).

ACKNOWLEDGMENT

The authors would like to thank Prof. Jun Wang, who made pioneering contributions in the field of neurodynamic systems for optimization problems. Without his encouragement, they would probably not have continued working on this interesting topic. To honor his contribution, 28 robots are required to achieve a formation of the shape icon JUN, which is shown in Fig. 15.

REFERENCES

[1] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 3, pp. 367–385, Jun. 2003.

[2] M. Saptharishi *et al.*, "Distributed surveillance and reconnaissance using multiple autonomous ATVs: CyberScout," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 826–836, Oct. 2002.

[3] D. A. Paley, F. Zhang, and N. E. Leonard, "Cooperative control for ocean sampling: The glider coordinated control system," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 4, pp. 735–744, Jul. 2008.

[4] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 610–621, Jul. 2007.

[5] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct. 2002.

[6] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, May 2008.

[7] B. S. Park, J. B. Park, and Y. H. Choi, "Adaptive formation control of electrically driven nonholonomic mobile robots with limited information," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 1061–1075, Aug. 2011.

[8] X. Wang, W. Ni, and X. Wang, "Leader-following formation of switching multirobot systems via internal model," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 817–826, Jun. 2012.

[9] Z. Peng, D. Wang, Z. Chen, X. Hu, and W. Lan, "Adaptive dynamic surface control for formations of autonomous surface vehicles with uncertain dynamics," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 2, pp. 513–520, Mar. 2013.

[10] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 622–629, Apr. 2004.

[11] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Trans. Autom. Control*, vol. 50, no. 1, pp. 121–127, Jan. 2005.

[12] Z.-G. Hou, L. Cheng, and M. Tan, "Decentralized robust adaptive control for the multiagent system consensus problem using neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 3, pp. 636–647, Jun. 2009.

[13] T. Liu and Z.-P. Jiang, "Distributed formation control of nonholonomic mobile robots without global position measurements," *Automatica*, vol. 49, no. 2, pp. 592–600, Feb. 2013.

[14] K. D. Do, "Formation control of underactuated ships with elliptical shape approximation and limited communication ranges," *Automatica*, vol. 48, no. 7, pp. 1380–1388, Jul. 2012.

[15] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.

[16] B. D. O. Anderson, C. Yu, and J. M. Hendrickx, "Rigid graph control architectures for autonomous formations," *IEEE Control Syst. Mag.*, vol. 28, no. 6, pp. 48–63, Dec. 2008.

[17] Y.-H. Chang, C.-W. Chang, C.-L. Chen, and C.-W. Tao, "Fuzzy sliding-mode formation control for multirobot systems: Design and implementation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 444–457, Apr. 2012.

[18] T. Dierks and S. Jagannathan, "Neural network output feedback control of robot formations," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 2, pp. 383–399, Apr. 2010.

[19] J. R. Spletzer and R. Fierro, "Optimal positioning strategies for shape changes in robot teams," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, Apr. 2005, pp. 742–747.

[20] J. C. Derenick and J. R. Spletzer, "Convex optimization strategies for coordinating large-scale robot formations," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1252–1259, Dec. 2007.

[21] Y. Xia and J. Wang, "A general methodology for designing globally convergent optimization neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1331–1343, Nov. 1998.

[22] Y. Xia and J. Wang, "Recurrent neural networks for optimization: The state of the art," in *Recurrent Neural Networks: Design and Applications*, L. Medsker and L. C. Jain, Eds. Boca Raton, FL, USA: CRC Press, 1999, ch. 2, pp. 13–45.

[23] M. S. Kamel and Y. Xia, "Cooperative recurrent modular neural networks for constrained optimization: A survey of models and applications," *Cognit. Neurodyn.*, vol. 3, no. 1, pp. 47–81, Mar. 2009.

[24] Y. Fei, G. Rong, B. Wang, and W. Wang, "Parallel L-BFGS-B algorithm on GPU," *Comput. Graph.*, vol. 40, pp. 1–9, May 2014.

[25] N. Ploskas and N. Samaras, "Efficient GPU-based implementations of simplex type algorithms," *Appl. Math. Comput.*, vol. 250, pp. 552–570, Jan. 2015.

[26] J. Wang, W. S. Tang, and C. Roze, "Neural network applications in intelligent manufacturing: An updated survey," in *Computational Intelligence in Manufacturing Handbook*, J. Wang and A. Kusiak, Eds. Boca Raton, FL, USA: CRC Press, 2001.

[27] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, no. 3, pp. 141–152, Jul. 1985.

[28] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, no. 5, pp. 554–562, May 1988.

[29] S. Zhang and A. G. Constantinides, "Lagrange programming neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 7, pp. 441–452, Jul. 1992.

[30] J. Wang, "Primal and dual assignment networks," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 784–790, May 1997.

[31] Y. Xia and J. Wang, "A dual neural network for kinematic control of redundant robot manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 1, pp. 147–154, Feb. 2001.

[32] L. Cheng, Z.-G. Hou, M. Tan, and X. Wang, "A simplified recurrent neural network for solving nonlinear variational inequalities," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Hong Kong, Jun. 2008, pp. 104–109.

[33] L. Cheng, Z.-G. Hou, H. Homma, M. Tan, and M. M. Gupta, "Solving convex optimization problems using recurrent neural networks in finite time," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Atlanta, GA, USA, Jun. 2009, pp. 538–543.

[34] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 4, pp. 447–458, Apr. 2002.

[35] L. Cheng, Z.-G. Hou, and M. Tan, "A neutral-type delayed projection neural network for solving nonlinear variational inequalities," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 55, no. 8, pp. 806–810, Aug. 2008.

[36] L. Cheng, Z.-G. Hou, and M. Tan, "Solving linear variational inequalities by projection neural network with time-varying delays," *Phys. Lett. A*, vol. 373, no. 20, pp. 1739–1743, Apr. 2009.
[37] L. Cheng, Z.-G. Hou, and M. Tan, "A delayed projection neural network for solving linear variational inequalities," *IEEE Trans. Neural Netw.*, vol. 20, no. 6, pp. 915–925, Jun. 2009.
[38] L. Cheng, Z.-G. Hou, and M. Tan, "Relaxation labeling using an improved Hopfield neural network," *Springer Lecture Notes Control Inf. Sci.*, vol. 345, pp. 430–439, 2006.
[39] L. Cheng, Z.-G. Hou, and M. Tan, "Constrained multi-variable generalized predictive control using a dual neural network," *Neural Comput. Appl.*, vol. 16, no. 6, pp. 505–512, Oct. 2007.
[40] Z. Yan and J. Wang, "Model predictive control for tracking of underactuated vessels based on recurrent neural networks," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 717–726, Oct. 2012.
[41] Y. S. Xia, G. Feng, and J. Wang, "A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 1, pp. 54–64, Feb. 2005.
[42] Z.-G. Hou, M. M. Gupta, P. N. Nikiforuk, M. Tan, and L. Cheng, "A recurrent neural network for hierarchical control of interconnected dynamic systems," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 466–481, Mar. 2007.
[43] Z.-G. Hou, L. Cheng, and M. Tan, "Multicriteria optimization for coordination of redundant robots using a dual neural network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1075–1087, Aug. 2010.
[44] L. Cheng, Z.-G. Hou, and M. Tan, "A simplified neural network for linear matrix inequality problems," *Neural Process. Lett.*, vol. 29, no. 3, pp. 213–230, Jun. 2009.
[45] M. Forti, P. Nistri, and M. Quincampoix, "Generalized neural network for nonsmooth nonlinear programming problems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1741–1754, Sep. 2004.
[46] X. Xue and W. Bian, "Subgradient-based neural networks for nonsmooth convex optimization problems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 8, pp. 2378–2391, Sep. 2008.
[47] Q. Liu and J. Wang, "A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 812–824, May 2013.
[48] Q. Liu and J. Wang, "A one-layer recurrent neural network for constrained nonsmooth optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 5, pp. 1323–1333, Oct. 2012.
[49] L. Cheng, Z.-G. Hou, Y. Lin, M. Tan, W. C. Zhang, and F.-X. Wu, "Recurrent neural network for non-smooth convex optimization problems with application to the identification of genetic regulatory networks," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 714–726, May 2011.
[50] D. G. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bull. London Math. Soc.*, vol. 16, no. 2, pp. 81–121, 1984.
[51] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. Hoboken, NJ, USA: Wiley, 1998.
[52] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.
[53] F. H. Clarke, *Optimization and Nonsmooth Analysis*. New York, NY, USA: Wiley, 1983.
[54] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer-Verlag, 1999.

Yunpeng Wang received the B.S. degree in information and computing science from Shandong University, Jinan, China, in 2010, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2015.

He is currently a Post-Doctoral Researcher with the Beijing Institute of Control Engineering, Beijing. His current research interests include the multiagent system and the robotic control.

Long Cheng (SM'14) received the B.S. (Hons.) degree in control engineering from Nankai University, Tianjin, China, in 2004, and the Ph.D. (Hons.) degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009.

He was a Post-Doctoral Research Fellow with the Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, SK, Canada, in 2010. From 2010 to 2011, he was a Post-Doctoral Research Fellow with the Mechanical and Industrial Engineering Department, Northeastern University, Boston, MA, USA. From 2013 to 2014, he was a Visiting Scholar with the Electrical and Computer Engineering Department, University of California at Riverside, Riverside, CA, USA. He is currently a Professor with the Laboratory of Complex Systems and Intelligent Science, Institute of Automation, Chinese Academy of Sciences. He has authored over 50 technical papers in peer-refereed journals and prestigious conference proceedings. His current research interests include intelligent control of smart materials, coordination of multiagent systems, neural networks, and their applications to robotics.

Dr. Cheng is also an Editorial Board Member of *Neurocomputing* and the *International Journal of Systems Science*.

Zeng-Guang Hou (SM'09) received the B.E. and M.E. degrees in electrical engineering from Yanshan University, Qinhuangdao, China, in 1991 and 1993, respectively, and the Ph.D. degree in electrical engineering from the Beijing Institute of Technology, Beijing, China, in 1997.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. His current research interests include neural networks, optimization algorithms, robotics, and intelligent control systems.

Dr. Hou is also an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and an Editorial Board Member of *Neural Networks*.

Junzhi Yu (SM'14) received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North China Institute of Technology, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2003.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, IACAS. He has authored over 100 journal and conference papers. His current research interests include biomimetic robots, intelligent control, and intelligent mechatonic systems.

Dr. Yu is also an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS and the IEEE TRANSACTIONS ON MECHATRONICS/*ASME Transactions on Mechatronics*.

Min Tan received the B.S. degree in control engineering from Tsinghua University, Beijing, China, in 1986, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1990.

He is currently a Professor with the Laboratory of Complex Systems and Intelligent Science, Institute of Automation, Chinese Academy of Sciences. His current research interests include advanced robot control, multirobot systems, biomimetic robots, and manufacturing systems.