Using Fuzzy Logic to Design Separation Function in Flocking Algorithms

Dongbing Gu, *Senior Member, IEEE*, and Huosheng Hu, *Senior Member, IEEE*

***Abstract*—Flocking algorithms essentially consist of three components: alignment, cohesion, and separation. To track a desired trajectory, the flock center should move along the desired trajectory, and thus, the fourth component, navigation, is necessary. The alignment, cohesion, and navigation components are well implemented through consensus protocols and tracking controls, while the separation component is designed through heuristic-based approaches. This paper proposes a fuzzy logic solution to the separation component. The TS rules and Gaussian membership functions are used in fuzzy logic. For fixed network flocking, a standard stability proof by using LaSalle's invariance principle is provided. For dynamic network flocking, a Filipov solution definition is given for nonsmooth dynamics. Then, a LaSalle's invariance principle for nonsmooth dynamics is used to prove the stability. A group of mobile robots with double integrator dynamics is simulated for the flocking algorithms in a 2-D environment.**

***Index Terms*—Cooperative control, flocking behavior, multirobot systems, nonsmooth systems.**

I. Introduction

It is well known that flocking behavior of living beings has certain advantages, such as avoiding predators, increasing the chance of finding food, saving energy, etc. Examples include flocks of birds, schools of fish, herds of wildebeest, and colonies of bacteria. Potential engineering applications of such cooperative behavior include automated highway systems [1], [2], cooperative robot reconnaissance [3], manipulation operation [4], formation flight control [5], [6], deployment of distributed sensor arrays [7], etc.

Flocking behavior has been studied in robotics within three structures—behavior-based structure, leader–follower structure, and virtual leader structure. The flocking control in a behavior-based structure is established by building up a group of formation-related behaviors [3]. It is suitable for uncertain environments, but lacks a rigorous theoretic analysis. The leader–follower structure is constructed by a string of chains where each robot follows a single robot ($l-\phi$ model) or two robots ($l-l$ model) [8], [9]. However, the chain structure leads to a poor disturbance rejection property. The virtual leader structure is inspired by the biological model where the center of all robot positions jointly represents a single, possible fictitious leader. Reynolds [10] simulated a flock of birds in flight with a common average heading, and they avoid colliding with each other. There is no leader broadcasting instructions in the flock and each bird has a local control strategy. There are three components in the local strategy of each bird: separation steering to avoid crowding, alignment steering toward the average heading of neighbors, and cohesion steering toward the average position of neighbors. A similar model was proposed by Vicsek in [11] and its convergence proofs were recently given in [12] and [13].

For robots with double integrator dynamics, flocking algorithms have been proven stable for both fixed and dynamic networks [14], [15]. In particular, a navigation component is added to the flocking algorithm, which is very demanding in formation control and other practical applications [15]. Further, the use of a navigation component eliminates the requirement of the connectivity property of a graph.

Speed consensus protocol [16], [17] and standard linear tracking control approaches are used for alignment, cohesion, and navigation components, while the separation component relies on a repulsive potential function, of which a minimum value exists at the point where the distance between neighbor robots is fixed. For fixed network flocking where the neighbors of a robot include all other robots, the stability can be analyzed by using LaSalle's invariance principle. For dynamic or switched network flocking where the neighbors of a robot are those robots who can be ranged through limited ranging sensors or communication channels, nonsmooth system analysis has been used in [14]. To avoid nonsmooth control analysis, special potential functions can be designed [15].

In this paper, we use fuzzy logic to design a control function for the separation component. This control function is the gradient of a potential-like function, which can guarantee the flocking stability. The fuzzy logic control function consists of T-K rules and Gaussian membership functions. They are designed based on stability analysis. For practical applications, it is expected that minimized control effects are used. The advantage of using fuzzy logic to design the separation control function is the guarantee of system stability with constrained magnitude of control inputs. The popularly used artificial potential function [14] generates the repulsive force, which is inversely proportional to the distance. It cannot constrain the magnitude of control inputs as the magnitude of control inputs becomes very large when the distance between neighbor robots becomes small. Also, it could cause erratic behavior when the large force drives robot to move far away from the flocking. The control inputs of potential functions contain the robot's position and neighbors' positions. Thus, the noise of control inputs comes from robot localization and communication between robots. The fuzzy potential function can suppress the input noise. We analyze the flocking stability under a class of input uncertainties. In addition, the use of fuzzy control function provides an opportunity for learning of the flocking algorithms through using fuzzy policy reinforcement learning.

Manuscript received February 9, 2006; revised November 26, 2006, March 22, 2007, and July 12, 2007.

The authors are with the Department of Computer Science, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: dgu@essex.ac.uk; hhu@essex.ac.uk).

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/TFUZZ.2008.917289

The robots with double integrator dynamics are used in this paper. An algebraic graph is used to model the flocking dynamics. We discuss the stability of both fixed and dynamic network flocking. For dynamic network flocking, Filippov solutions are used to define the solution concept [18]. Clarke's generalized gradient [19] and set-value derivative [20] are used to simplify the stability analysis. The nonsmooth stability analysis for dynamic network flocking based on [20, Th. 3] is presented.

The rest of the paper is organized as follows. Section II presents a flocking model for robots with double integrator dynamics. Section III describes distributed flocking algorithms. Section IV proposes a fuzzy separation control function and nonsmooth control approach. The stability analysis of both fixed and dynamic network flocking is given in Section V. The simulations of the proposed flocking algorithms are provided in Section VI. Finally, our conclusions and future work are briefly summarized in Section VII.

II. Flock Model

A. Robot Dynamics

Given N robots in a group, each robot is described by its double integrator dynamics. For a robot i with n dimensional coordinates $q_i = [x_i^1, \ldots, x_i^n]^T \in \mathcal{R}^n$, the state and control vectors are $z_i(t) = [q_i(t)^T, \dot{q}_i(t)^T]^T \in \mathcal{R}^{2n}$ and $u_i(t) \in \mathcal{R}^n$ $(i = 1, \ldots, N)$. The robot dynamics is

$$\dot{z}_i = az_i + bu_i \tag{1}$$

where

$$a = \begin{bmatrix} 0 & I_{(n)} \\ 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ I_{(n)} \end{bmatrix}.$$

The matrix $I_{(n)}$ is the identity matrix of dimension n. The states of all N robots in a group can be concatenated into vectors $q = [q_1^T, \ldots, q_N^T]^T \in \mathcal{R}^{nN}$, $\dot{q} = [\dot{q}_1^T, \ldots, \dot{q}_N^T]^T \in \mathcal{R}^{nN}$, $z = [z_1^T, \ldots, z_N^T]^T \in \mathcal{R}^{2nN}$, and $u = [u_1^T, \ldots, u_N^T]^T \in \mathcal{R}^{nN}$. The group collective state dynamics is

$$\dot{z} = Az + Bu \tag{2}$$

where $A = aI_{(N)}$, $B = b\mathbf{1}^T$, and $\mathbf{1} = [1, \ldots, 1]$.

The group center can be denoted by a virtual center state $z_c = [(q_c)^T, (\dot{q}_c)^T]^T$. q_c is the virtual center's configuration, which is the average of all states:

$$q_c = \frac{1}{N}\sum_{i=1}^{N} q_i. \tag{3}$$

Then, the following results are true for the virtual center:

$$\dot{q}_c = \frac{1}{N}\sum_{i=1}^{N} \dot{q}_i, \quad z_c = \frac{1}{N}\sum_{i=1}^{N} z_i, \quad u_c = \frac{1}{N}\sum_{i=1}^{N} u_i. \tag{4}$$

Thus, the dynamics of the virtue center is

$$\dot{z}_c = az_c + bu_c. \tag{5}$$

Let $z_d = [(q_d)^T, (\dot{q}_d)^T]^T$ be the desired state vector for the group center. The desired state z_d should also have the same dynamics with the robot dynamics (1):

$$\dot{z}_d = az_d + bu_d. \tag{6}$$

Definition 1 (Flock definition): The configuration q_i is called flock if:

1) the distances $\|q_i - q_j\|$ between any two neighbor robots are asymptotically convergent to a fixed value;
2) the velocity mismatches $\dot{q}_i - \dot{q}_j$ of all robots are asymptotically convergent to zero;
3) the center trajectory q_c is asymptotically convergent to a desired trajectory q_d;
4) no collision between robots occurs during the flocking.

B. Flock Graph

A graph can be used to represent the flocking interconnection between robots. A vertex of the graph corresponds to a robot, and edges of the graph capture the dependence of interconnections. Formally, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $\mathcal{V} = \{v_1, \ldots, v_N\}$, indexed by robots in the group, and a set of edges $\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V}\}$, containing unordered pairs of distinct vertices. Assuming the graph has no loops, that is, $(v_i, v_j) \in \mathcal{E}$ implies $v_i \neq v_j$.

A graph is connected if, for any vertices $(v_i, v_j) \in \mathcal{V}$, there exists a path of edges in $\mathcal{E}$ from v_i to v_j. The incidence matrix $D(\mathcal{G})$ of a graph $\mathcal{G}$ is the $\{0, 1\}$-matrix with rows and columns indexed by vertices $\mathcal{V}$ and edges of $\mathcal{E}$, respectively, such that the ijth element of $D(\mathcal{G})$ is equal to 1 if the vertex i is in the edge j and 0 otherwise. If the graph $\mathcal{G}$ has N vertices and $|\mathcal{E}|$ edges, then incidence matrix $D(\mathcal{G})$ of the graph $\mathcal{G}$ has order $N \times |\mathcal{E}|$ [21]. Because graph $\mathcal{G}$ is undirected, the matrix $D(\mathcal{G})$ is symmetric. Let $\Delta(\mathcal{G})$ be the diagonal $N \times N$ matrix with rows and columns indexed by $\mathcal{V}$ with ii entry equal to the valency of vertex i. Following [21], Laplacian of a graph $\mathcal{G}$ is defined as $L(\mathcal{G})$:

$$L(\mathcal{G}) = \Delta(\mathcal{G}) - D(\mathcal{G}). \tag{7}$$

For a graph $\mathcal{G}$, it is known the rank of its Laplacian $L(\mathcal{G})$ is $N - |\mathcal{E}|$. Laplacian $L(\mathcal{G})$ is symmetric and positive semidefinite. Let r denote the distance that a robot can range via ranging sensors or communicate via wireless radio links.

Definition 2 (Neighbor definition): Robot j is called a neighbor of robot i if the Euclidean distance r_{ij} between robots i and j is less than or equal to r. All neighbors of robot i compose its neighbor set $\mathcal{N}_i$.

For fixed network flocking, each robot i can range or communicate with all other group members. Therefore, the graph $\mathcal{G}$ is fixed and will not vary with time. However, for dynamic or switching network flocking, the neighbor set $\mathcal{N}_i$ of robot i changes with time due to the limited ranging value r. Accordingly, the graph $\mathcal{G}$ will vary with time and can be denoted as $\mathcal{G}^{\sigma(t)}$ where $\sigma(t)$ is a piecewise constant function of time, called a switching signal. The neighbor set and Laplacian of a graph are denoted as $\mathcal{N}_i^{\sigma(t)}$ and $L^{\sigma(t)}$, respectively. The number of neighbors of robot i is denoted as $|\mathcal{N}_i^{\sigma(t)}|$.

III. Distributed Flocking Control

A. Alignment, Cohesion, and Navigation Control

For distributed control, robot i should have the ability to access all its neighbors' states $z_j, j \in \mathcal{N}_i^{\sigma(t)}$. This means each robot i should acquire z_j through its ranging sensors or wireless communication at each time step t. Each robot i should also have the ability to fully observe its own state z_i at each time step t. Based on the states z_i and z_j, each robot can use state feedback control approaches to flocking.

For the alignment component, each robot needs a velocity state feedback control to reduce velocity mismatches and accordingly align its heading with its neighbors. The state feedback control for this component is denoted as u_i^a and defined as

$$u_i^a = -k^a \sum_{j \in \mathcal{N}_i^{\sigma(t)}} (\dot{q}_i - \dot{q}_j) \tag{8}$$

where $k^a > 0$ is a n-dimensional diagonal matrix. This is the so-called velocity consensus protocol [16], [12], which can asymptotically make the velocity of robot i equal to the average velocity of its neighbors. Collectively, the alignment control function of all robots can be written as a vector u^a

$$u^a = -k^a \otimes L^{\sigma(t)} \dot{q} \tag{9}$$

where $\otimes$ is the Kronecker product.

For cohesion and navigation components, all robots' states z_i should tend to be the same as the desired state z_d. The state feedback control function for robot i should be as

$$u_i^c = -\mathbf{k}^c (z_i - z_d) + u_d \tag{10}$$

where $\mathbf{k}^c = k^c \otimes I_{(n)}$ is the feedback gain and $k^c > 0$ is a n-dimensional diagonal matrix. From (4), (5), and (10), it can be seen that the cohesion and navigation control function of the virtual center has the same state feedback law:

$$u_c^c = -\mathbf{k}^c (z_c - z_d) + u_d. \tag{11}$$

Using the sum of u_i^a in (8) and u_i^c in (10) as a control function, all robots will move toward the desired trajectory. Fig. 1 shows that 20 2-D robots randomly starting from different positions with different velocities will asymptotically converge to the desired trajectory, a circle. It can be seen that separation control is necessary for flocking, which can separate all robots with a fixed distance and avoid collision with each other during flocking.

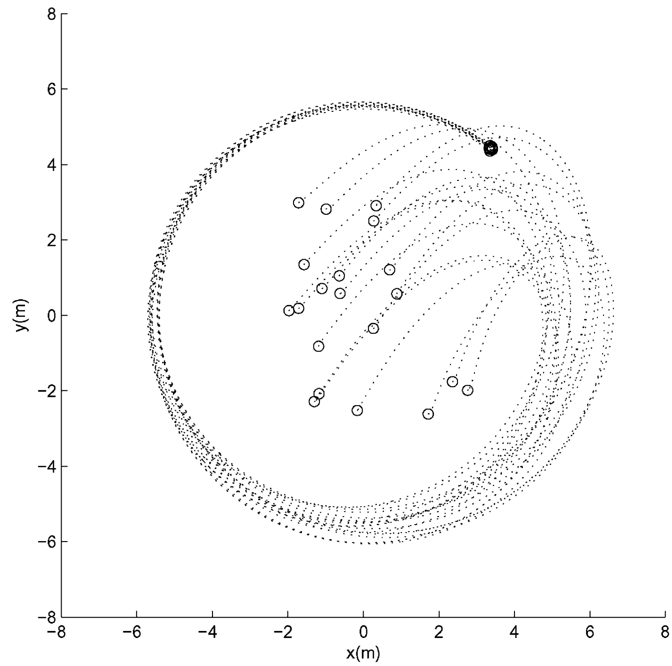

Fig. 1. All robots converge to a circle.

B. Separation Control

In general, a function $g_{ij}(q_i, q_j)$ can be defined as the separation control function between robot i and its neighbor robot j. This function is a state feedback function. The separation control function of robot i should include all effects from all its neighbors:

$$u_i^s = -\sum_{j \in \mathcal{N}_i^{\sigma(t)}} g_{ij}(q_i, q_j). \tag{12}$$

A smooth repulsive potential function $V_{ij}(r_{ij})$ can be used to generate the separation control function. The repulsive potential function $V_{ij}(r_{ij})$ should have a minimum value at $r_{ij} = d$ where d is a predefined distance. When the Euclidian distance $r_{ij} = \|q_i - q_j\|$ is less than d, the control function should be negative to repulse neighbor robots from getting too close. When the Euclidian distance r_{ij} is larger than d, the control function should be positive to attract neighbor robots to move closer. The gradient of repulsive potential function $V_{ij}(r_{ij})$ is the separation function $g_{ij}(q_i, q_j)$:

$$g_{ij}(q_i, q_j) = \nabla_{q_i} V_{ij}(r_{ij}). \tag{13}$$

The separation control function of robot i is

$$u_i^s = -\sum_{j \in \mathcal{N}_i^{\sigma(t)}} \nabla_{q_i} V_{ij}(r_{ij}). \tag{14}$$

All control functions can be added together to form a state feedback control function u_i for flocking:

$$u_i = u_i^a + u_i^c + u_i^s. \tag{15}$$

In the next section, the separation control function is implemented by fuzzy logic.

IV. Fuzzy Separation Control

A. Fuzzy Logic Control Function

Fuzzy logic can build any nonlinear function mappings based on human experience. The separation control function can be built through a set of fuzzy logic rules. Therefore, fuzzy logic employed as the separation function can be used for flocking control.

A set of fuzzy logic rules can map an input s to a deterministic control $h(s)$. For the kth dimension state $x_i^k (k = 1, \ldots, n)$, robot i uses states (x_i^k, x_j^k) to build a P-dimensional vector $s_{ij}^k = \{s_{ij}^{k,1}, \ldots, s_{ij}^{k,P}\}$ as fuzzy input. The corresponding fuzzy input sets are $F_i^1, \ldots, F_i^P$. The following expression is a fuzzy rule for the separation control function between robots i and j:

$$\text{Rule } m\text{: IF } s_{ij}^{k,1} \text{ is } F_i^{m,1} \text{ AND}, \ldots, s_{ij}^{k,P} \text{ is } F_i^{m,P}$$

$$\text{THEN } h_{ij}^m = k_i^{m,0} + \sum_{p=1}^{P} k_i^{m,p} s_{ij}^{k,p}$$

where $s_{ij}^{k,p}$ is the pth input ($p = 1, \ldots, P$), $F_i^{m,p}$ is the pth input fuzzy set, h_{ij}^m is the output, and $k_i^{m,p}$ is the parameters of the fuzzy output. The fuzzy set $F_i^{m,p}$ and the output parameter $k_i^{m,p}$ are the same for any neighbors j. The activation degree of rule m is calculated by the production operation

$$\lambda_{ij}^m = \prod_{p=1}^{P} \eta_{F_i^{m,p}}. \tag{16}$$

The Gaussian function can be used to define the fuzzy set membership function

$$\eta_{F_i^{m,p}} = \exp\left[-\frac{(s_{ij}^{k,p} - \mu_i^{m,p})^2}{2(\sigma_i^{m,p})^2}\right] \tag{17}$$

where $\mu_i^{m,p}, \sigma_i^{m,p}$ ($p = 1, \ldots, P$) are the mean and variance, respectively. The activation degree (16) is rewritten as follows:

$$\lambda_{ij}^m = \exp\left[-\sum_{p=1}^{P} \frac{(s_{ij}^{k,p} - \mu_i^{m,p})^2}{2(\sigma_i^{m,p})^2}\right]. \tag{18}$$

The crisp output u_{ij} is calculated by the center of area method

$$h_{ij}(s_{ij}^k) = \frac{\sum_{m=1}^{M} \lambda_{ij}^m h_{ij}^m}{\sum_{m=1}^{M} \lambda_{ij}^m} \tag{19}$$

where M is the number of fuzzy rules.

B. Fuzzy Separation Control Function

The separation control function u_i^s is an n-dimensional vector. Let g_{ij} denote the nth-dimension fuzzy separation control function between robots i and j. Due to the symmetry of the robot dynamics (1), control functions in different dimensions are the same. For the kth dimension, the fuzzy input s_{ij}^k consists of $r_{ij}^2 - d^2$ and $x_i^k - x_j^k$, i.e., $P = 2$ and $s_{ij}^k = [s_{ij}^{k,1}, s_{ij}^{k,2}] = [r_{ij}^2 - d^2, x_i^k - x_j^k]$. The fuzzy output is defined as

$$h_{ij}^m = k_i^{m,2}(x_i^k - x_j^k) \tag{20}$$

which implies $k_i^{m,0} = 0$ and $k_i^{m,1} = 0$. A fuzzy rule m is defined as

$$\text{IF } r_{ij}^2 - d^2 \text{ is } F_i^{m,1} \text{ THEN } h_{ij}^m = k_i^{m,2}(x_i^k - x_j^k).$$

Denote the gradient of the repulsive potential function $V_{ij}(r_{ij})$ as

$$\nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij}) = \frac{1}{2} \frac{\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2}}{\sum_{m=1}^{M} \lambda_{ij}^m}. \tag{21}$$

From (19), we have

$$\begin{aligned} h_{ij}(s_{ij}^k) &= \frac{\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2}}{\sum_{m=1}^{M} \lambda_{ij}^m}(x_i^k - x_j^k) \\ &= 2\nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij})(x_i^k - x_j^k) \\ &= \nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij}) \nabla_{x_i^k} s_{ij}^{k,1} \\ &= \nabla_{x_i^k} V_{ij}(r_{ij}). \end{aligned} \tag{22}$$

The fuzzy control function between robots i and j is

$$g_{ij}(q_i, q_j) = \begin{bmatrix} h_{ij}(s_{ij}^1) \\ \vdots \\ h_{ij}(s_{ij}^n) \end{bmatrix}. \tag{23}$$

For distributed control, robot i needs to sum up all control functions $g_{ij}(q_i, q_j)$ from its neighbors to generate the separation control function u_i^s:

$$\begin{aligned} u_i^s &= -\sum_{j \in \mathcal{N}_i^{\sigma(t)}} g_{ij}(q_i, q_j) \\ &= -\begin{bmatrix} \sum_{j \in \mathcal{N}_i^{\sigma(t)}} \nabla_{x_i^1} V_{ij}(r_{ij}) \\ \vdots \\ \sum_{j \in \mathcal{N}_i^{\sigma(t)}} \nabla_{x_i^n} V_{ij}(r_{ij}) \end{bmatrix} \\ &= -\sum_{j \in \mathcal{N}_i^{\sigma(t)}} \nabla_{q_i} V_{ij}(r_{ij}). \end{aligned} \tag{24}$$

C. Nonsmooth Control

For fixed network flocking, the control function u_i is smooth as u_i^s is smooth. For dynamic network flocking, the control function u_i is not a smooth function as the neighbors of each robot are changing during flocking, i.e., the graph $\mathcal{G}$ is changing. The energy function used to analyze the stability is also a nonsmooth function.

To analyze nonsmooth dynamics, some known facts pertaining to differential inclusions and nonsmooth analysis are necessary. As the classical notion of solution is not appropriate and needs to be generalized, a solution definition for the nonsmooth dynamics, the Filippov solution [18], is used first. Its formal definition is given in the Appendix. Other terms will be introduced in the next section.

With the state feedback in fixed network flocking, the robot dynamics (2) is an autonomous system:

$$\dot{z} = f(z, t). \tag{25}$$

In dynamic network flocking, the autonomous system is defined as a differential inclusion

$$\dot{z} \in^{a.e.} K[f](z, t) \tag{26}$$

where $K[f](z, t)$ is a differential inclusion and $a.e.$ stands for "almost everywhere." As the differential inclusion $K[f]$ is upper semicontinuous with nonempty, compact, convex values and locally bounded, for each $z_0 \in \mathcal{R}^{2nN}$, there exists at least one solution of the differential inclusion with the initial condition $z(0) = z_0$ [18].

The separation control function u_i^s with dynamic neighbor set $\mathcal{N}_i^{\sigma(t)}$ is

$$u_i^s = \begin{cases} -\sum_{j\in\mathcal{N}_i^{\sigma(t)}} \nabla_{q_i} V_{ij}(r_{ij}), & r_{ij} < R \\ 0, & r_{ij} \geq R \end{cases} \tag{27}$$

In [22], a calculus is provided to simplify the calculation of differential inclusions. As $f(z,t) = Az + Bu = [\dot{q}^T, u^T]^T$, using the calculus, we have

$$\begin{aligned} K[f] &= \begin{bmatrix} \dot{q} \\ K[u] \end{bmatrix} \\ &= \begin{bmatrix} \dot{q} \\ u^a + u^c + K[u^s] \end{bmatrix} \\ &= \begin{cases} \begin{bmatrix} \dot{q} \\ u^a + u^c - \begin{bmatrix} \vdots \\ \sum_{j\in\mathcal{N}_i^{\sigma(t)}} \nabla_{q_i} V_{ij} \\ \vdots \end{bmatrix} \end{bmatrix} \\ \text{if} \quad r_{ij} < R \\ \begin{bmatrix} \dot{q} \\ u^a + u^c \end{bmatrix}, \quad \text{if} \quad r_{ij} \geq R. \end{cases} \end{aligned} \tag{28}$$

D. Saturation Constraint of Separation Control Signal

The using of fuzzy logic to design the separation function can constrain the magnitude of input signals. This is desirable for practical systems with input saturation constraints. From (22) to (24), we know

$$u_i^s = -\sum_{j\in\mathcal{N}_i^{\sigma(t)}} \sum_{m=1}^{M} \frac{\lambda_{ij}^m}{\sum_{m=1}^{M} \lambda_{ij}^m} k_i^{m,2}(x_i^k - x_j^k).$$

Due to $\|(x_i^k - x_j^k)\| \leq R + d$ for all $x_i^k - x_j^k$ and $0 \leq \lambda_{ij}^m / (\sum_{m=1}^{M} \lambda_{ij}^m) \leq 1$, we have

$$\begin{aligned} \|u_i^s\| &= \left\| \sum_{j\in\mathcal{N}_i^{\sigma(t)}} \sum_{m=1}^{M} \frac{\lambda_{ij}^m}{\sum_{m=1}^{M} \lambda_{ij}^m} k_i^{m,2}(x_i^k - x_j^k) \right\| \\ &\leq \sum_{j\in\mathcal{N}_i^{\sigma(t)}} \sum_{m=1}^{M} \|k_i^{m,2}\|(R+d) \\ &= |\mathcal{N}_i^{\sigma(t)}|(R+d) \sum_{m=1}^{M} \|k_i^{m,2}\| \\ &\leq \kappa \end{aligned} \tag{29}$$

where κ is a positive constant. Equation (29) shows that the separation control signal is saturated constrained.

V. Flocking Stability

For fixed network flocking, the LaSalle invariance principle can be employed to analyze the stability. For dynamic network flocking, a similar version of LaSalle's invariance principle for nonsmooth functions can be employed to analyze the stability.

A. Fixed Network Flocking Stability

The flocking control with fixed neighbor set $\mathcal{N}_i$ is

$$\begin{aligned} u_i &= u_i^a + u_i^c + u_i^s \\ &= -k^a \sum_{j\in\mathcal{N}_i} (\dot{q}_i - \dot{q}_j) - \mathbf{k}^c(z_i - z_d) \\ &\quad - \sum_{j\in\mathcal{N}_i} g_{ij}(q_i, q_j) + u_d. \end{aligned} \tag{30}$$

An energy function is defined as

$$\begin{aligned} V(z) &= V^s(q) + V^a(z) + V^c(z) \\ &= \sum_{i=1}^{N} V_i(q) + \frac{1}{2}\|z - \mathbf{1}^T \otimes z_c\|_{\mathbf{K}^c}^2 \\ &\quad + \frac{1}{2}\|z_c - z_d\|^2 \end{aligned} \tag{31}$$

where $V^a(z) = \frac{1}{2}\|z - \mathbf{1}^T \otimes z_c\|_{\mathbf{K}^c}^2$, $V^c(z) = \frac{1}{2}\|z_c - z_d\|^2$, $V^s(q) = \sum_{i=1}^{N} V_i(q)$, and

$$\mathbf{K}^c = I_N \otimes \begin{bmatrix} k^c & 0 \\ 0 & \mathbf{1} \end{bmatrix}.$$

$V_i(q)$ is the repulsive potential function:

$$V_i(q) = \sum_{j\in\mathcal{N}_i} V_{ij}(r_{ij}). \tag{32}$$

Theorem 1 (Fixed network flocking stability theorem): For fixed network flocking, starting from $\Omega_c = \{z : V(z) \leq c\}$, the flocking as defined in Definition 1 is asymptotically stable under the state feedback control u_i (30) with an appropriate control gain k^c.

Proof:

1) The energy function $V(z)$ is positive definite.

As $V^a(z)$ and $V^c(z)$ are positive definite, the fact that $V^s(z)$ is positive definite can guarantee that $V(z)$ is also positive definite. From the gradient of the repulsive potential function (21), the following analysis can be obtained.

a) For $r_{ij} - d < 0$, if $k_i^{m,2} < 0$ for those rules whose $\mu_i^{m,1} < 0$, then $\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2} < 0$, i.e., $\nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij}) < 0$.

b) For $r_{ij} - d > 0$, if $k_i^{m,2} > 0$ for those rules whose $\mu_i^{m,1} > 0$, then $\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2} > 0$, i.e., $\nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij}) > 0$.

c) For $r_{ij} - d = 0$, if $k_i^{m,2} = 0$ for the rule whose $\mu_i^{m,1} = 0$ and for other rules $\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2}|_{r_{ij}=d} = 0$, then $\nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij}) = 0$.

In summary, we can let $V_{ij}(r_{ij})$ have a minimum value at $r_{ij} = d$ and $\nabla_{s_{ij}^{k,1}} V_{ij}(r_{ij}) = 0$ at $r_{ij} = d$ through selecting the values of $k_i^{m,2}$ and $\mu_i^{m,1}$. Hence, $V_{ij}(r_{ij})$ can be designed positive definite by using proper fuzzy rules, i.e., $V^s(z)$ can be positive definite.

2) The derivative of the energy function $V(z)$ is seminegative definite.

$$\dot{V}^s(q) = \sum_{i=1}^{N} \dot{V}_i(q) = \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} \nabla_{q_i} V_{ij}(r_{ij})\dot{q}_i = -\sum_{i=1}^{N} u_i^s \dot{q}_i \quad (33)$$

$$\dot{V}^a(z) = (z - \mathbf{1}^T \otimes z_c)^T \boldsymbol{K}^c(\dot{z} - \mathbf{1}^T \otimes \dot{z}_c) = k^c \sum_{i=1}^{N}(q_i - q_c)^T(\dot{q}_i - \dot{q}_c) + \sum_{i=1}^{N}(\dot{q}_i - \dot{q}_c)^T(\ddot{q}_i - \ddot{q}_c).$$

From $\ddot{q}_i = u_i^a + u_i^c + u_i^s$ and $\sum_{i=1}^{N}(\dot{q}_i - \dot{q}_c) = 0$, we have $\sum_{i=1}^{N}(\dot{q}_i - \dot{q}_c)(u_d - \ddot{q}_c) = 0$.
Then,

$$\dot{V}^a(z) = k^c \sum_{i=1}^{N}(q_i - q_c)^T(\dot{q}_i - \dot{q}_c) + \sum_{i=1}^{N}(\dot{q}_i - \dot{q}_c)^T \left[-k^a \sum_{j\in\mathcal{N}_i^{\sigma(t)}}(\dot{q}_i - \dot{q}_j) - \mathbf{k}^c(z_i - z_d) + u_i^s\right].$$

Due to $\sum_{i=1}^{N} u_i^s = 0$, it is further simplified as

$$\dot{V}^a(z) = -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 + \sum_{i=1}^{N}(\dot{q}_i - \dot{q}_c)^T u_i^s = -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 + \sum_{i=1}^{N} \dot{q}_i^T u_i^s - \sum_{i=1}^{N} \dot{q}_c^T u_i^s = -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 - \dot{V}^s(q) \quad (34)$$

where $Q = k^a \otimes L^{\sigma(t)} + k^c \otimes I_{(N)}$.

For $V^c(z)$, first we have

$$u_c - u_d = \frac{1}{N}\sum_{i=1}^{N} u_i - u_d = \frac{1}{N}\sum_{i=1}^{N}(u_i^a - \mathbf{k}^c(z_i - z_d) + u_i^s) = -\frac{1}{N}\sum_{i=1}^{N}\mathbf{k}^c(z_i - z_d) = -\mathbf{k}^c(z_c - z_d).$$

Then,

$$\dot{V}^c(z) = (z_c - z_d)^T(\dot{z}_c - \dot{z}_d) = (z_c - z_d)^T[a(z_c - z_d) + b(u_c - u_d)] = \|z_c - z_d\|_{(a-b\mathbf{k}^c)}^2. \quad (35)$$

From (33) to (35), we have

$$\dot{V}(z) = \dot{V}^s(q) + \dot{V}^a(z) + \dot{V}^c(z) = -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 + \|z_c - z_d\|_{(a-b\mathbf{k}^c)}^2. \quad (36)$$

Therefore, as long as $(a - b\mathbf{k}^c) \leq 0$, $\dot{V}$ is seminegative definite. This can be achieved by selecting the control gain k^c as $[a, b]$ is controllable.

3) LaSalle's invariance principle.

As $\dot{V}(z)$ is seminegative definite, $V(z)$ is therefore monotonically decreasing for all $t \geq 0$. Based on this nonincreasing property of $V(z)$ and given $\Omega_c = \{z : V(z) \leq c\}$, Ω_c is an invariant set. This bounded set and

$$\|z_c - z_d\|^2 \leq V(z) \leq c$$

guarantee $(z_c - z_d)$ is bounded. Given the desired state z_d is bounded and $z_c = 1/N\sum_{i=1}^{N} z_i$, we know z_i is bounded.

From LaSalle's invariance principle, all states starting in Ω_c converge to the largest invariant set $E = \{z \in \Omega_c : \dot{V}(z) = 0\}$. Hence, asymptotically, all states converge to the the largest invariant set $E = \{z \in \Omega_c : z_c = z_d, \dot{q}_i = \dot{q}_c\}$. Or it can be stated that the flock center state z_c converges asymptotically to z_d and all robot velocities $\dot{q}_i$ converge to the center speed $\dot{q}_c$.

Further, when the flocking converges to stable state $(z_c \to z_d, \dot{q}_i \to \dot{q}_c)$, $V(z) \to V^s(q) + 1/2\sum_{i=1}^{N} \|q_i - q_c\|_{k^c}^2 = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} V_{ij}(r_{ij}) + 1/2\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} \|q_i - q_j\|_{k^c}^2$. There is an equilibrium point at $r_{ij} = d$ where $\nabla_{r_{ij}} V(z) = 0$. Thus, the distance between two neighbor robots converges to the equilibrium point.

4) There are no collisions with each other.

By contradiction, assume there exists a time $t = t_1$ when two robots k, l collide, i.e., $q_k(t_1) = q_l(t_1)$. The potential function $V_{kl}(q_k, q_l)$ at $q_k(t_1) = q_l(t_1)$ is a constant

$$V^s(q) = \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} V_{ij}(q_i, q_j) = V_{kl}(q_k, q_l) + \sum_{i\in\mathcal{V}\backslash\{k,l\}}\sum_{j\in\mathcal{V}\backslash\{i,k,l\}} V_{ij}(q_i, q_j) \geq V_{kl}(q_k, q_l).$$

At $t = t_1$, defining the constant $V_{kl}(q_k, q_l)|_{q_k=q_l}$ larger than c leads to $V^s(q(t_1)) \geq c$, which is in contradiction with the invariant set Ω_c. Therefore, no two robots collide at any time $t \geq 0$. ■

Remark 1: To consider some input uncertainties, the flocking control (30) can also be expressed as

$$u_i = u_i^a + u_i^c + u_i^s + w(t) = -k^a \sum_{j\in\mathcal{N}_i}(\dot{q}_i - \dot{q}_j) - \mathbf{k}^c(z_i - z_d)$$

$$- \sum_{j \in \mathcal{N}_i} g_{ij}(q_i, q_j) + u_d + w(t) \tag{37}$$

where $w(t) \in \mathcal{R}^{nN}$ denotes the unknown uncertainties with a known upper bound $w_u \geq \|w(t)\|$. Based on the similar results as in the proof of Theorem 1, we have

$$\begin{aligned}
\dot{V}(z) &= \dot{V}^s(q) + \dot{V}^a(z) + \dot{V}^c(z) \\
&= -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 + \|z_c - z_d\|^2_{(a - b\mathbf{k}^c)} \\
&\quad + \sum_{i=1}^{N} (\dot{q}_i - \dot{q}_c)^T w_i(t) \\
&\leq -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 + (\dot{q} - \mathbf{1}^T \otimes \dot{q}_c)^T w(t) \\
&= -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 \\
&\quad - \left\| \frac{1}{2}(\dot{q} - \mathbf{1}^T \otimes \dot{q}_c) - w(t) \right\|^2 \\
&\quad + \frac{1}{2}\|(\dot{q} - \mathbf{1}^T \otimes \dot{q}_c)\|^2 + \|w(t)\|^2 \\
&\leq -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|_Q^2 \\
&\quad + \frac{1}{2}\|(\dot{q} - \mathbf{1}^T \otimes \dot{q}_c)\|^2 + \|w(t)\|^2 \\
&= -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|^2_{Q - \frac{1}{2}I_{(nN)}} + \|w(t)\|^2 \\
&\leq -\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|^2_{Q_1} + w_u^2 \\
&\leq -\lambda_{\min}(Q_1)\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\|^2 + w_u^2
\end{aligned} \tag{38}$$

where $Q_1 = Q - (1/2)I_{(nN)}$ and $\lambda_{\min}(Q_1)$ is the smallest eigenvalue of Q_1. For the connected graph, all eigenvalues of Laplacian matrix $\lambda_i(L) \geq 0$. We can select k^a and k^c to make $\lambda_{\min}(Q_1) > 0$. Then, we have the following result.

Whenever $\|\dot{q} - \mathbf{1}^T \otimes \dot{q}_c\| > w_u/\sqrt{\lambda_{\min}(Q_1)}$, $\dot{V}(z) < 0$. Then, the asymptotical stability of the perturbed closed-loop system is proved according to Lyapunov's stability theorem.

B. Dynamic Network Flocking Stability

The flocking control with dynamic neighbor set $\mathcal{N}_i^{\sigma(t)}$ is

$$\begin{aligned}
u_i &= u_i^a + u_i^c + u_i^s \\
&= -k^a \sum_{j \in \mathcal{N}_i^{\sigma(t)}} (\dot{q}_i - \dot{q}_j) - \mathbf{k}^c(z_i - z_d) \\
&\quad - \sum_{j \in \mathcal{N}_i^{\sigma(t)}} g_{ij}(q_i, q_j).
\end{aligned} \tag{39}$$

The separation control function u_i^s is no longer a smooth function as $\sigma(t)$ varies with time. For the differential equations with discontinuous right-hand sides, nonsmooth energy functions nationally arise.

Definition 3 (Nonsmooth repulsive potential function): The repulsive potential function between two robots i and j for the nonsmooth separation control function (27) can be defined as follows:

$$U_{ij}(r_{ij}) = \begin{cases} V_{ij}(r_{ij}), & r_{ij} < R \\ V_{ij}(R), & r_{ij} \geq R. \end{cases} \tag{40}$$

This function is in the form of the pointwise maximum of two smooth functions. Hence, it is a regular function [23].

Due to the use of nonsmooth repulsive potential function, the classical notion of gradient is not appropriate. Clarke's generalized gradient can be used to analyze the nonsmooth stability [19]. The generalized gradient at a point z can be viewed as a set-valued map equal to the convex closure of the limiting gradient near z. The formal definition of the generalized gradient is given in the Appendix.

The generalized gradient of $U_{ij}(r_{ij})$ can be calculated based on the control function (27) and the directional generalized derivative given in the Appendix as follows:

$$\partial U_{ij}(r_{ij})) = \begin{cases} \nabla_{q_i} V_{ij}(r_{ij}), & r_{ij} < R \\ 0, & r_{ij} \geq R. \end{cases} \tag{41}$$

The repulsive potential function $U^s(q)$ of the group is defined as

$$U^s(q) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^{\sigma(t)}} U_{ij}(r_{ij}). \tag{42}$$

The energy function $U(z) = U^s(q) + V^c(z) + V^a(z)$ is, therefore, a nonsmooth and regular function. Its generalized gradient is

$$\partial U(z) = \partial U^s(q) + \nabla V^c(z) + \nabla V^a(z). \tag{43}$$

There are three versions of LaSalle's invariance principle in the nonsmooth analysis. In [23], the regular function and solution uniqueness are used. In [24], the regular function and solution uniqueness are not required. It needs to define the worst case of the gradient. In [20], the regular function is required, but not solution uniqueness. In the following, the version in [20] is used to prove the stability.

Theorem 2 (Dynamic network flocking stability theorem): For dynamic network flocking, starting from $\Omega_c = \{z : V(z) \leq c\}$, the flocking as defined in Definition 1 is asymptotically stable under the state feedback control u_i (39).

Proof: LaSalle's invariance principle in the nonsmooth analysis provided in [20] needs to calculate a set-valued derivative. The definition of the set-valued derivative is given in the Appendix. For the differential inclusion (26), by using the generalized gradient (41) and (43), the set-valued

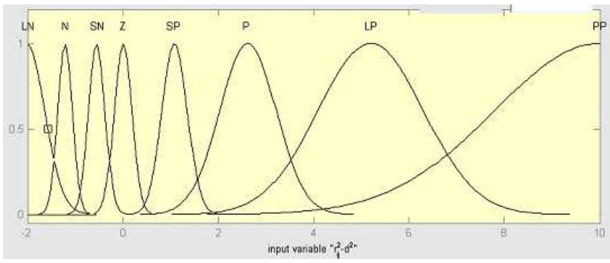

Fig. 2. Fuzzy sets.

derivative is

$$\dot{\overline{V}}(z) = \bigcap_{\xi \in \partial U(z)} \xi^T K[f](z,t)$$
$$= \sum_{i=1}^{N} \bigcap_{\xi_i} \xi_i^T \dot{q}_i + \|(z_c - z_d)\|^2_{(a-b\mathbf{k}^c)}$$
$$- \|\dot{q}_i - \dot{q}_c\|^2_Q + \sum_{i=1}^{N} \dot{q}_i^T u_i^s$$
$$= \|(z_c - z_d)\|^2_{(a-b\mathbf{k}^c)} - \|\dot{q}_i - \dot{q}_c\|^2_Q \quad (44)$$

where $\xi_i \in \sum_{j \in \mathcal{N}_i^{\sigma(t)}} \partial U_{ij}(r_{ij})$.

Let

$$Z_V = \{z \in \mathbb{R}^{2nN} : 0 \in \dot{\overline{V}}(z)\}. \quad (45)$$

The largest weakly invariant subset of $\overline{Z_V}$ is $\{\|(z_c - z_d)\|^2_{(a-b\mathbf{k}^c)} = 0, \|\dot{q}_i - \dot{q}_c\|^2_Q = 0\}$, i.e., $\{z_c = z_d, \dot{q}_i = \dot{q}_c\}$. Applying Theorem 3 provided in the Appendix to the system described by (26), it follows that, for the initial conditions in Ω_c, Filippov trajectories of the system converge to the largest weakly invariant set $E = \{z \in \Omega_c : z_c = z_d, \dot{q}_i = \dot{q}_c\}$. Or, it can be stated that the flock center state z_c converges asymptotically to z_d and all robot velocities $\dot{q}_i$ converge to the center speed $\dot{q}_c$.

Finally, with the same argument as that in the fixed network flocking stability, the distance between neighbor robots converges to a fixed value and the collision between neighbor robots can be avoided. ■

VI. Simulations

Eight fuzzy sets are designed for the fuzzy control input $r_{ij}^2 - d^2$. They are LN, N, SN, Z, SP, P, LP, and PP as shown in Fig. 2. Eight fuzzy rules are designed as follows:

IF $r_{ij}^2 - d^2$ is LN THEN $h_{ij}^1 = -100(x_i^k - x_j^k)$

IF $r_{ij}^2 - d^2$ is N THEN $h_{ij}^2 = -80(x_i^k - x_j^k)$

IF $r_{ij}^2 - d^2$ is SN THEN $h_{ij}^3 = -50(x_i^k - x_j^k)$

IF $r_{ij}^2 - d^2$ is Z THEN $h_{ij}^4 = 0$

IF $r_{ij}^2 - d^2$ is SP THEN $h_{ij}^5 = 0.5(x_i^k - x_j^k)$

IF $r_{ij}^2 - d^2$ is P THEN $h_{ij}^6 = (x_i^k - x_j^k)$

IF $r_{ij}^2 - d^2$ is LP THEN $h_{ij}^7 = 1.5(x_i^k - x_j^k)$

IF $r_{ij}^2 - d^2$ is PP THEN $h_{ij}^8 = 2(x_i^k - x_j^k)$.

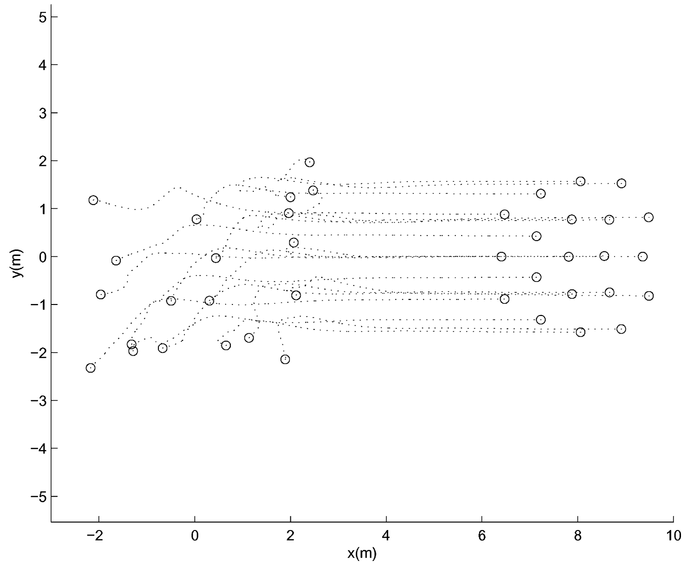

Fig. 3. Fixed network flocking in straight line trajectory.

It can be seen that for fuzzy sets LN, N, and SN, $k_i^{m,2} < 0$; for fuzzy sets SP, P, LP, and PP, $k_i^{m,2} > 0$; for fuzzy set Z, $k_i^{m,2} = 0$. Thus, when $r_{ij} - d < 0$, $\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2} < 0$. When $r_{ij} - d > 0$, $\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2} > 0$. $\sum_{m=1}^{M} \lambda_{ij}^m k_i^{m,2}|_{r_{ij}=d}$ is not exactly equal to zero, but very close to zero. Therefore, these rules meet the conditions analyzed in the stability proof of Theorem 1.

2-D robots are used ($n = 2$). The control gains are selected to be $k^c = I_{(2)}$ and $k^a = I_{(2)}$. The four eigenvalues of $(a - b\mathbf{k}^c)$ are $-0.5 + 0.866i$, $-0.5 - 0.866i$, $-0.5 + 0.866i$, $-0.5 - 0.866i$. Therefore, the state feedback control is stable. The desired trajectory is a straight line

$$q_i^d = \begin{bmatrix} 0.5t \\ 0 \end{bmatrix}$$

or a circle

$$q_i^d = \begin{bmatrix} 5\sin(0.5t) \\ 5\cos(0.5t) \end{bmatrix}.$$

A. Simulation Results by Using Fuzzy Potential Function

The fixed network flocking algorithm is tested first. Twenty robots are used, and all of them are connected with each other. The simulation results are shown in Fig. 3 for the straight line and Fig. 4 for the circle. From these two figures, it can be seen that 20 robots are initially randomly located within a square of $[-2.5m, 2.5m]$ and $[-2.5m, 2.5m]$. The velocities are initially randomly set to $[-1m/s, 1m/s]$. After a certain time, the flocking is stable and its center tracks the desired trajectory. The top two diagrams in Fig. 5 show velocities and separation control signals of all 20 robots tracking the circle $\dot{q}_i = [v_x, v_y]^T, u_i = [u_x, u_y]^T$. All of them are gradually convergent to their stable states.

The cohesion radius is defined as the maximum distance between any two robots. It converges to a stable level as shown in the left bottom of Fig. 5 for the circle tracking. The velocity

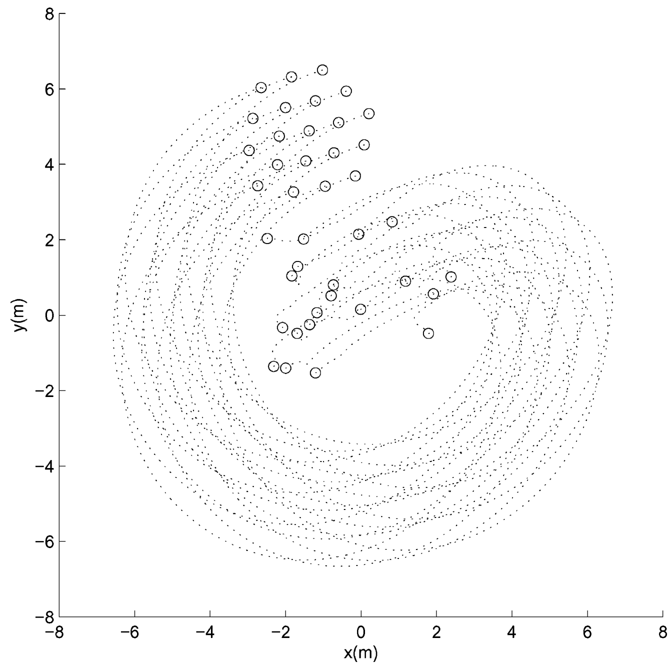

Fig. 4. Fixed network flocking in circle trajectory.

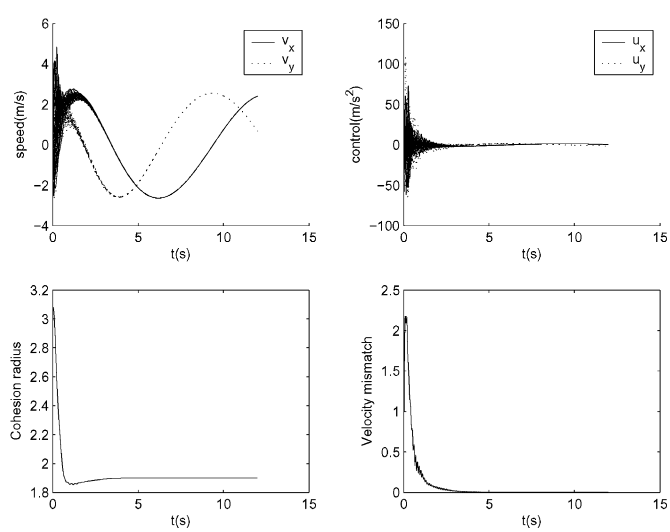

Fig. 5. Fixed network flocking performances.

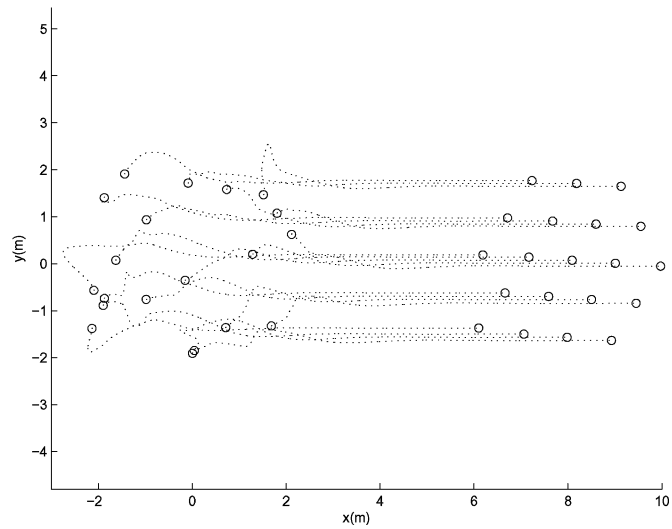

Fig. 6. Dynamic network flocking in straight line trajectory.

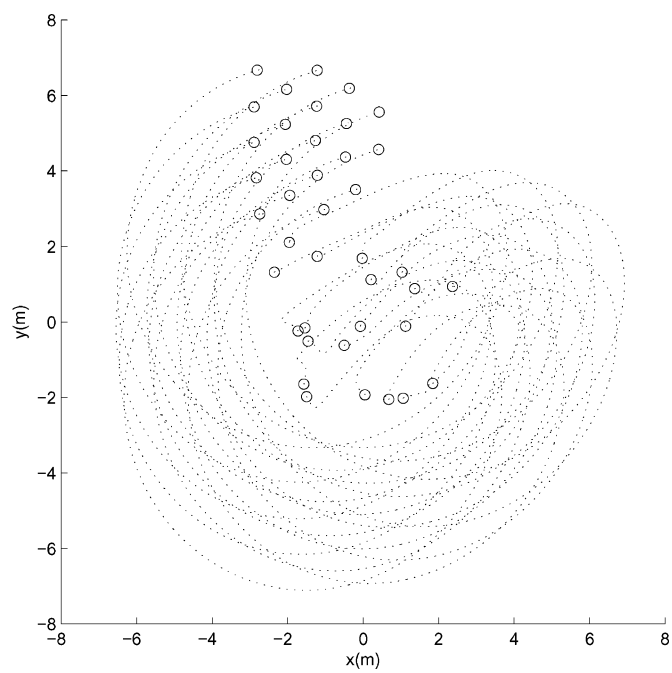

Fig. 7. Dynamic network flocking in circle trajectory.

mismatch is defined as $\sum_{i=1}^{N} \|\dot{q}_i - \dot{q}_c\|$. It converges to zero as shown in the right bottom of Fig. 5 for the circle tracking. The similar results can be obtained for the straight line tracking.

The dynamic network flocking algorithm is tested by using 20 robots with the same initial setting as the fixed network flocking algorithm. The variables $d = 1$ and $r = 1.2$ are used to switch neighbor robots. The simulation results are shown in Fig. 6 for the straight line tracking and Fig. 7 for the circle tracking. Both results show the convergent performance. For the circle tracking, the results including velocities, separation control signals, cohesion radius, and velocity mismatch are shown in Fig. 8. They have a similar convergent property as the fixed network flocking algorithm.

A random noise $w(t)$ with the upper bound $w_u = 0.25$ is used to test the control robustness discussed in Section V. All other parameters are the same as earlier. The tracking of the straight line is simulated with the noise signal. The results are shown in Fig. 9 for trajectories and Fig. 10 for the convergent performance. As they are expected, the trajectories and performances are not as smooth as the simulation results without noise. However, the flocking is still stable and its center still tracks the desired trajectory. The velocities, separation control signals, cohesion radius, and velocity mismatch also show the convergent performance.

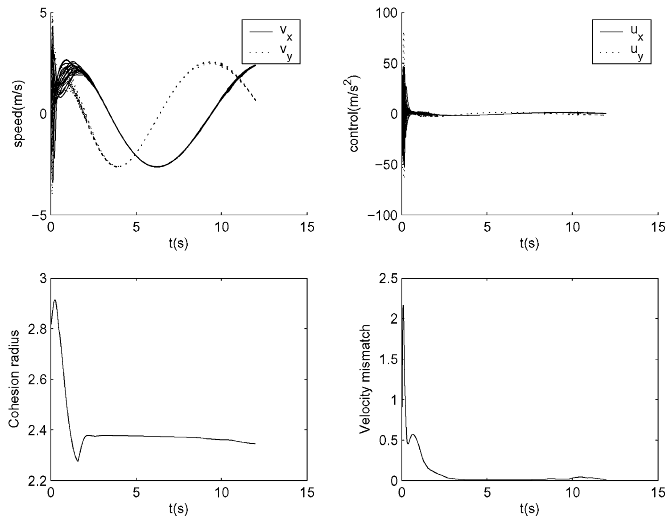

Fig. 8. Dynamic network flocking performances.

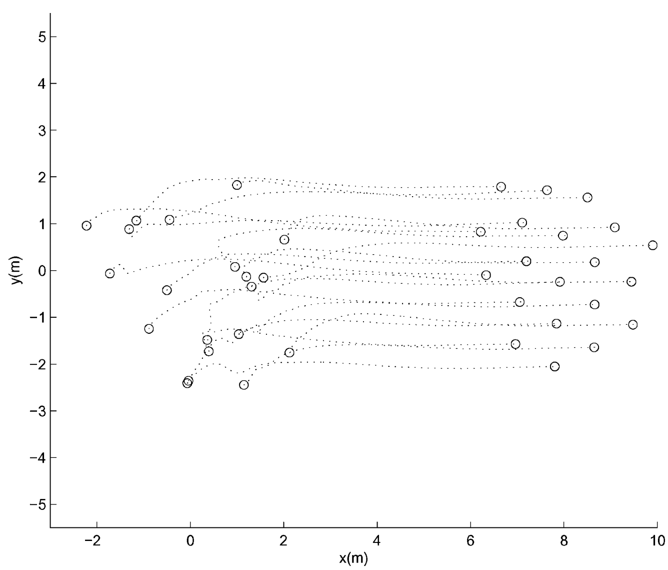

Fig. 9. Dynamic network flocking in straight line trajectory with noised control.

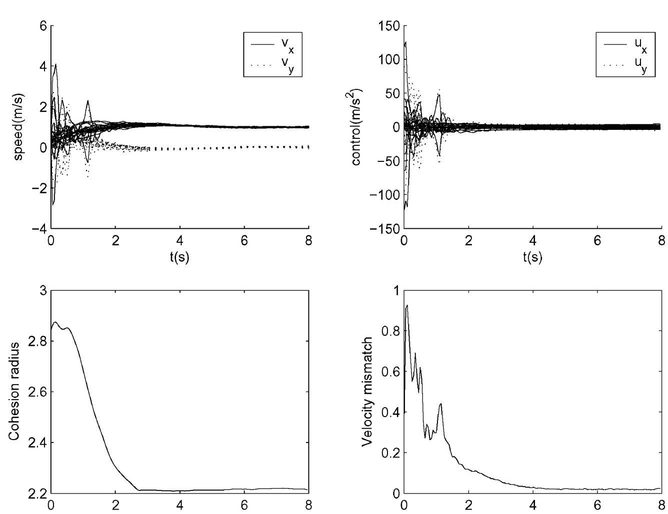

Fig. 10. Dynamic network flocking performances with noised control.

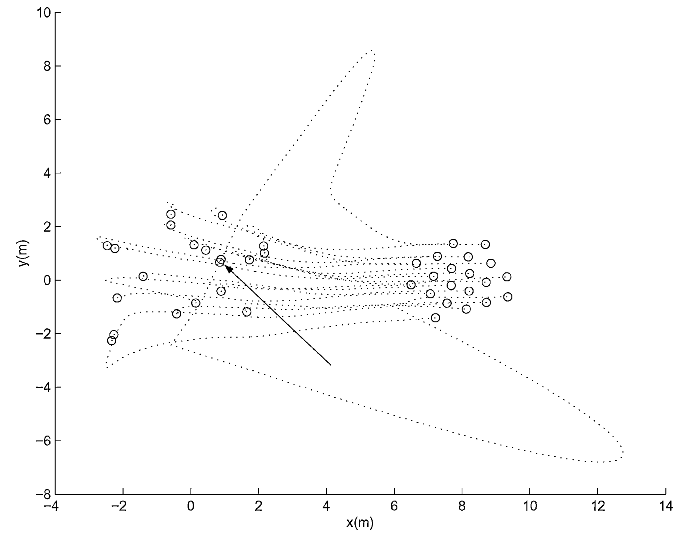

Fig. 11. Dynamic network flocking in straight line trajectory with the potential function (46).

B. Comparison Results

A typical potential function for separation control [14] is the one defined as

$$V_{ij}(r_{ij}) = \frac{1}{\left(\frac{\|r_{ij}\|}{d}\right)^2} + \log\left(\frac{\|r_{ij}\|}{d}\right)^2 \tag{46}$$

and its gradient can be found as

$$\nabla_{x_i^k} V_{ij}(r_{ij}) = \frac{2(x_i^k - x_j^k)(r_{ij}^2 - d^2)}{r_{ij}^4}. \tag{47}$$

This potential function can generate a very large repulsive force between two neighbor robots when the distance is very small. Sometimes, the force is so large that the robots move too far away from the flocking center. Fig. 11 shows one of these scenarios. As can be seen, the two robots pointed to by an arrow are initially very close to each other. The very large repulsive force moves them away. Although finally they move back to the flocking cluster under the control of the cohesion component, the flocking undergoes erratic behavior. The large repulsive force and large velocity can be found from the top of Fig. 12. Obviously, the cohesion radius and the velocity mismatch are also large, as can be seen from the bottom of Fig. 12.

Our proposed fuzzy potential function can avoid such erratic behavior. We use the same simulation setting as the one described earlier except for the potential function $V_{ij}(r_{ij})$. We also use the same initial positions and velocities. The simulation result is shown in Fig. 13. It clearly shows the two robots pointed to by an arrow do not move too far away and the flocking behavior is fairly smooth.

The simulation performances shown in Fig. 14 further confirm the result. Especially, the separation control signals are limited (the right top of Fig. 14). Also, the cohesion radius and the velocity mismatch are much smaller than those using the potential function (46).

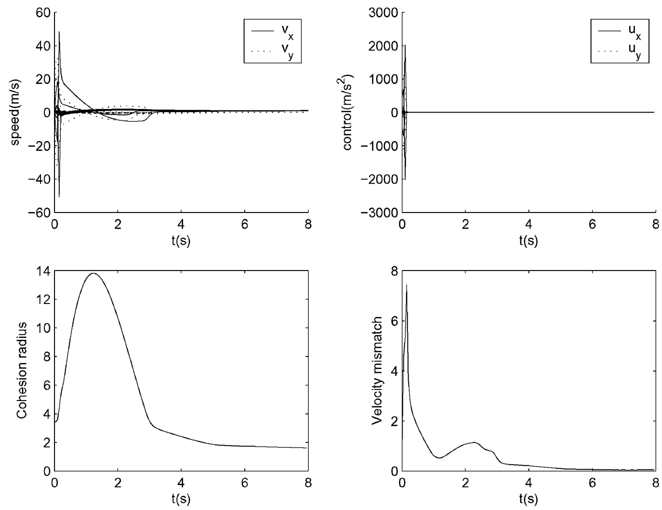

Fig. 12. Dynamic network flocking performances with the potential function (46).

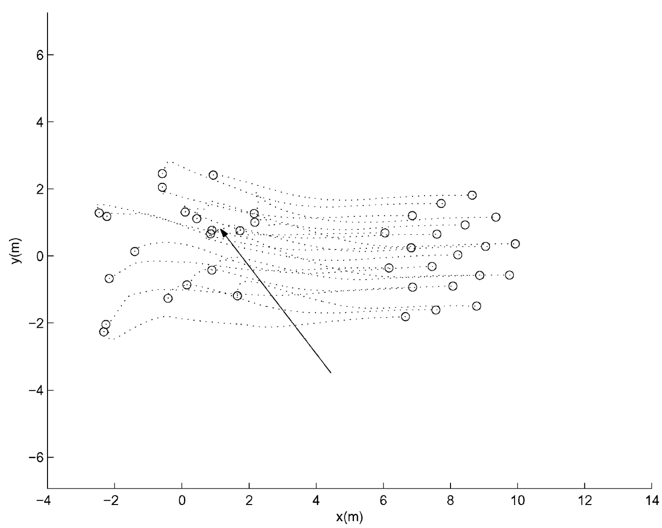

Fig. 13. Dynamic network flocking straight line trajectory with the fuzzy potential function.

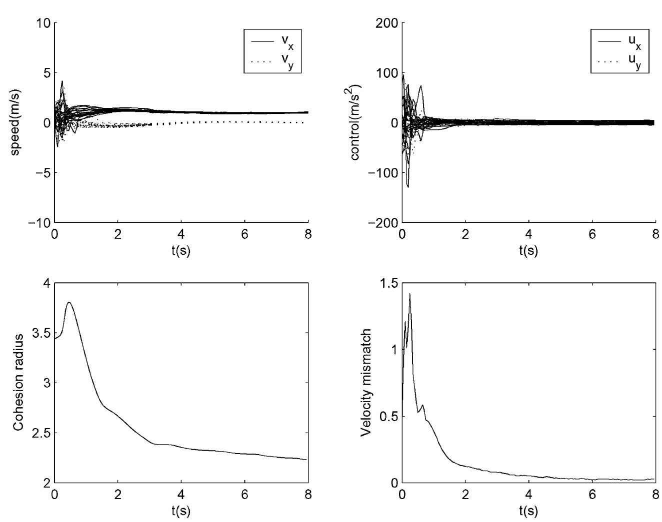

Fig. 14. Dynamic network flocking performances with the fuzzy potential function.

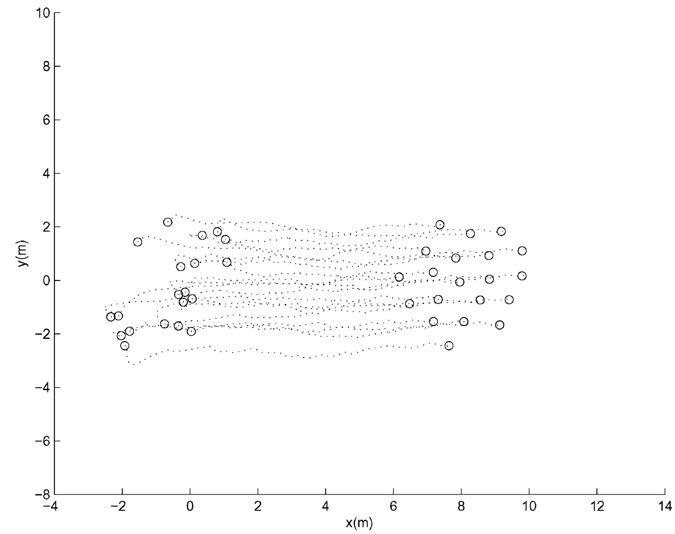

Fig. 15. Dynamic network flocking in straight line trajectory with the fuzzy potential function.

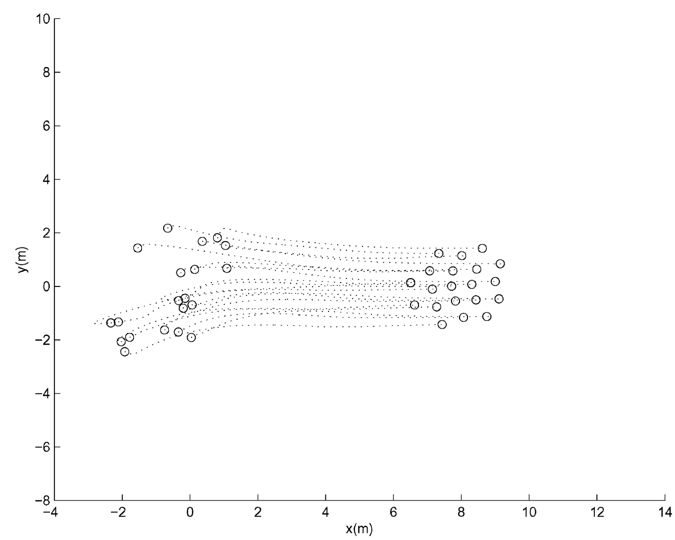

Fig. 16. Dynamic network flocking in straight line trajectory with the potential function (46).

Furthermore, the separation control signals in all simulations using the fuzzy potential function are constrained within the limitation as discussed in Section IV. The limitation depends on the parameter κ defined in (29). It does not relate to the initial positions, velocities, or the desired trajectory. This conclusion can be observed from Figs. 5, 8, 10, and 14.

The noise from self-localization (x_i) and communication (x_j) can also be used to reflect the effectiveness of using the fuzzy potential function. We can add a random noise $v(t)$ with the upper bound $v_x = 0.2$ to $x_i^k - x_j^k$. It can be shown that the fuzzy potential function can suppress noise, while the potential function in (46) does not have such a property.

The simulation result with noise $v(t)$ by using the fuzzy potential function is shown in Fig. 15. Although the tracking trajectories are not smooth, the flocking behavior is still reasonable. We have not shown the simulation result without noise $v(t)$

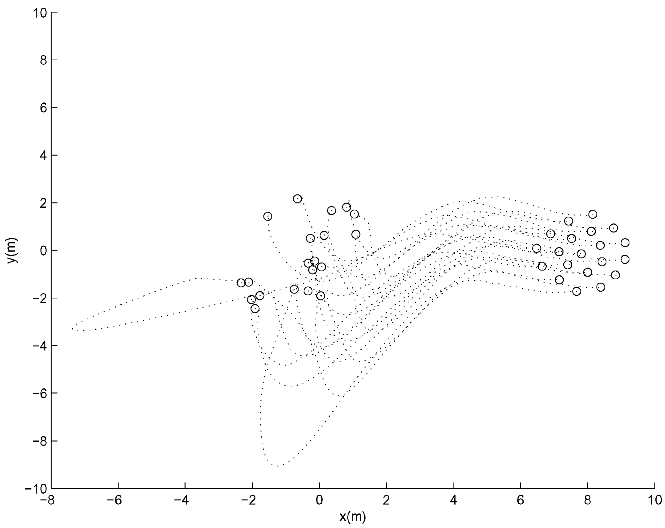

Fig. 17. Dynamic network flocking in straight line trajectory with the potential function (46) and noise.

by using the fuzzy potential function as it has a better behavior than in Fig. 15.

The simulation result without noise $v(t)$ by using the potential function (46) is shown in Fig. 16. The flocking behavior is fairly smooth. However, it becomes erratic after adding noise $v(t)$ as shown in Fig. 17. By comparing Figs. 15 and 17, it can be seen that the fuzzy potential function has a good property in suppressing the noise.

VII. Conclusion

In flocking algorithms, consensus protocol and tracking control have been formally developed. However, the separation control is defined through *ad hoc* potential functions. This paper presents a fuzzy logic approach to the separation control. By using fuzzy logic, a set of basic fuzzy rules plays the roles of separation control functions. These rules can be designed by human intuition and experience. The designed fuzzy separation function can constrain the control inputs as the constrained control inputs are demanding in practical applications.

For both fixed and dynamic network flocking, the stability analysis shows that the fuzzy separation control function can be used to stabilize the flocking. In the dynamic network flocking, this is achieved by using nonsmooth analysis. The proposed control algorithm also has the robustness property with respect to a class of input uncertainty.

In the next step, the fuzzy separation control function will be learned through simulation. A policy gradient reinforcement learning algorithm can be used to learn the fuzzy logic control function. To do so, the fuzzy separation control function needs to be parameterized and stochastic. We also plan to handle the obstacle avoiding problem in the flocking algorithm through modeling obstacles as the neighbors of robots.

Appendix

Definition 4 (Filippov solution definition): A vector function $z(\cdot)$ is called a solution of (26) on $[t_0, t_1]$ if $z(\cdot)$ is absolutely continuous on $[t_0, t_1]$ and for almost all $t \in [t_0, t_1]$

$$\dot{z} \in K[f](z,t) \tag{48}$$

where $K[f](z,t)$ is a differential inclusion. There exists $N_f \subset \mathcal{R}^m, \mu(N_f) = 0$ such that, for all $N \subset R^m, \mu(N) = 0$

$$K[f](z,t) = \overline{co}\{\lim_{z_t \to z}, z_t \notin N_f \cup N\} \tag{49}$$

where μ is the Lebesgue measure and $\overline{co}$ denotes the convex closure. The content of Filippov's solution is that the tangent vector to a solution must lie in the convex closure of the limiting values of the vector filed in progressively smaller neighborhoods around the solution to be defined at the points even where the vector filed itself is not defined, such as at the interface of two regions in a piece-wise defined vector field. Using this definition, nonsmooth of f occurs in a set of zero measure, and this nonsmooth occurrence will not affect the differential inclusions, i.e., Filippov's solution [18], [23].

Clarke's generalized gradient and generalized directional derivative are particularly useful in simplifying proofs of nonsmooth stability [19].

Definition 5 (Clarke's generalized gradient): For a locally Lipschitz function $V : \mathcal{R}^m \to \mathcal{R}$, define the generalized gradient of V at z by

$$\partial V(z) = \overline{co}\{\lim_{z_i \to z} \nabla V(z), z_i \notin \Omega_v\} \tag{50}$$

where Ω_v is the set of measure zero where the gradient of V is not defined.

Definition 6 (Generalized directional derivative): The generalized directional gradient is defined as

$$V^\circ(z,w) = \limsup_{y \to z, t \downarrow 0} \frac{V(y+tw) - V(y)}{t}. \tag{51}$$

The following lemma links the generalized gradient and generalized directional derivative.

Lemma 1: Let V be Lipschitz near z, then

$$V^\circ(z,w) = \max\{\langle \xi, w\rangle | \xi \in \partial V(z)\}. \tag{52}$$

In [20], a set-valued derivative V with respect to (26) is defined to analyze the stability.

Definition 7 (Set-valued derivative): The set-valued derivative of V is defined as

$$\dot{\overline{V}}(z) = \{a \in \mathcal{R} : \exists v \in K[f](z,t), pv = a, \forall p \in \partial V(z)\}. \tag{53}$$

In case V is differentiable at z, one has $\dot{\overline{V}}(z) = \{\nabla V(z)v, v \in K[f](z,t)\}$. Moreover, $\dot{\overline{V}}(z)$ is, in general, a proper subset of the set $\dot{\tilde{V}}(z)$ used in [23].

Lemma 2: Let z be a solution of the differential inclusions (26) and let V be a locally Lipschitz continuous and regular function. Then, $(d/dt)V(z)$ exists almost everywhere and $(d/dt)V(z) \in \dot{\overline{V}}(z)$ almost everywhere.

To use LaSalle's invariance principle for nonsmooth analysis, a weakly invariant set is required [20].

Definition 8 (Weakly invariant set definition): A set Ω_m is called a weakly invariant set for (26) if through each point $z \in \Omega_m$, there exists a maximal solution of (26) lying in Ω_m.

Finally, a LaSalle's invariance principle for nonsmooth analysis is given [20].

Theorem 3: Let $V : \mathcal{R}^m \to \mathcal{R}$ be a locally Lipschitz continuous and regular function for (26). Assume for some $l > 0$, the connected component L_l of the level set $\{z \in \mathcal{R}^n : V(z) \leq l\}$ such that $0 \in L_l$ is bounded. Let

$$Z_V = \{z \in \mathcal{R}^m : 0 \in \dot{\overline{V}}(z)\} \tag{54}$$

and let M be the largest weakly invariant subset of $\overline{Z_V} \cap L_l$. Then $z(\cdot)$ approaches to M as $t \to \infty$.

References

[1] A. Pant, P. Seiler, and K. Hedric, "Mesh stability of look-ahead interconnected systems," *IEEE Trans. Autom. Control*, vol. 47, no. 2, pp. 403–407, Feb. 2002.

[2] D. Swaroop and J. K. Hedrick, "String stability of interconnected systems," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 349–357, Mar. 1996.

[3] T. Balch and R. Arkin, "Behavior-based formation control for multirobot systems," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 926–939, Dec. 1998.

[4] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of multiple mobile manipulators," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 53–64, Feb. 2003.

[5] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *IEEE Control Syst. Mag.*, vol. 20, no. 6, pp. 34–44, Dec. 2000.

[6] D. M. Stipanovic, G. Inalhan, R. Teo, and C. Tomlin, "Decentralised overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285–1296, 2004.

[7] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.

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

[9] J. P. O. J. P. Desai and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, Dec. 2001.

[10] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioural model," *Comput. Graph.*, vol. 21, no. 4, pp. 25–34, 1987.

[11] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995.

[12] A. Jadbabaie, J. Lin, and S. A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbour rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[13] A. V. Savkin, "Coordinated collective motion of groups of autonomous mobile robots: Analysis of vicsek's model," *IEEE Trans. Autom. Control*, vol. 49, no. 6, pp. 981–983, Jun. 2004.

[14] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 863–868, May 2007.

[15] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[16] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delay," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 101–115, Sep. 2004.

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

[18] A. F. Filippov, *Differential Equations With Discontinuous Righthand Side*. Norwell, MA: Kluwer, 1988.

[19] F. H. Clarke, "Optimization and nonsmooth analysis," in *Classics in Applied Mathematics*. Philadelphia, PA: SIAM, 1990.

[20] A. Bacciotti and F. Ceragioli, "Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions," *ESAIM: Control Optim. Calculus Variations*, vol. 4, pp. 361–376, 1999.

[21] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer-Verlag, 2001.

[22] B. Paden and S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.

[23] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39, no. 9, pp. 1910–1914, Sep. 1994.

[24] E. P. Ryan, "An integral invariance principle for differential inclusions with application in adaptive control," *SIAM J. Control Optim.*, vol. 36, no. 3, pp. 960–980, 1998.

Dongbing Gu (M'01–SM'07) received the B.Sc. and M.Sc. degrees in control engineering from Beijing Institute of Technology, Beijing, China in 1985 and 1988, respectively, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2004.

He joined the Department of Electronic Engineering, Changchun Institute of Optics and Fine Mechanics, Changchun, China, as a Lecturer in 1988, became an Associate Professor in 1993, and a Professor in 1999. During 1996–1997, he was an Academic Visiting Scholar in the Department of Engineering Science, University of Oxford, Oxford, U.K. Since 2000, he has been a Lecturer in the Department of Computer Science, University of Essex. He is the author or coauthor of over 70 published papers in international journals and conferences. His current research interests include predictive control, distributed formation control, machine learning, autonomous robots, and statistical image processing. He is an associate editor for the International Journal of Modelling, Identification and Control.

Dr. Gu is the recipient of several research awards in China.

Huosheng Hu (M'90-SM'00) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor in computer science at the University of Essex, Colchester, U.K., leading the Human Centred Robotics (HCR) Group. Since 2000, he has been a Guest Professor at six universities in China, such as the Central South University, Shanghai University, Wuhan University of Science and Engineering, Kunming University of Science and Technology, and the Northeast Normal University. He is the author or coauthor of over 250 published papers in journals, books, and conferences within these areas. His current research interests include mobile robotics, sensors integration, data fusion, distributed computing, intelligent control, behavior and hybrid control, cooperative robotics, telerobotics, and service robots. He is currently the Editor-in-Chief for the *International Journal of Automation and Computing*, and a member of the editorial board for the International Journal of Communication Networks and Distributed Systems.

Prof. Hu is currently a reviewer for a number of international journals such as the IEEE TRANSACTIONS ON ROBOTICS, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He is a Chartered Engineer and a member of the Institution of Electrical Engineers (IEE), the Association for the Advancement of Artificial Intelligence (AAAI), the Association for Computing Machinery (ACM), the Institute of Aeronautical Sciences (IAS), and the International Association of Science and Technology for Development (IASTED).