Flocking of Second-Order Multiagent Systems With Connectivity Preservation Based on Algebraic Connectivity Estimation

Hao Fang, *Member, IEEE*, Yue Wei, Jie Chen, *Senior Member, IEEE*, and Bin Xin, *Member, IEEE*

***Abstract*—The problem of flocking of second-order multiagent systems with connectivity preservation is investigated in this paper. First, for estimating the algebraic connectivity as well as the corresponding eigenvector, a new decentralized inverse power iteration scheme is formulated. Then, based on the estimation of the algebraic connectivity, a set of distributed gradient-based flocking control protocols is built with a new class of generalized hybrid potential fields which could guarantee collision avoidance, desired distance stabilization, and the connectivity of the underlying communication network simultaneously. What is important is that the proposed control scheme allows the existing edges to be broken without violation of connectivity constraints, and thus yields more flexibility of motions and reduces the communication cost for the multiagent system. In the end, nontrivial comparative simulations and experimental results are performed to demonstrate the effectiveness of the theoretical results and highlight the advantages of the proposed estimation scheme and control algorithm.**

***Index Terms*—Connectivity preservation, distributed consensus, generalized potential fields, inverse power iteration, multiagent systems.**

I. Introduction

RECENTLY, distributed flocking and coordination of networked agents has grown to pervade many scientific disciplines, and it is amenable to solve a wide variety of spatially distributed tasks [1]–[9]. Reynolds [3] proposed the boid model which consists of three heuristic interaction rules of separation, cohesion, and alignment. Inspired by, many flocking algorithms were proposed [10]–[12]. As a common property, graph connectivity is one of the most fundamental requirements for the sensing and communication topology.

Manuscript received June 12, 2015; revised November 16, 2015 and January 25, 2016; accepted February 20, 2016. Date of publication March 15, 2016; date of current version March 15, 2017. This work was supported in part by the Projects of Major International (Regional) Joint Research Program NSFC under Grant 61120106010, in part by the Program for Changjiang Scholars and Innovative Research Team in University under Grant IRT1208, in part by the Changjiang Scholars Program, in part by the National Natural Science Foundation of China under Grant 61573062 and Grant 61304215, in part by the Beijing Education Committee Cooperation Building Foundation Project, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61321002, and in part by the Beijing Outstanding Ph.D. Program Mentor under Grant 20131000704. This paper was recommended by Associate Editor Q. Liu. *(Corresponding author: Jie Chen.)*

The authors are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China, and also with the Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China (e-mail: chenjie@bit.edu.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/TCYB.2016.2537307

Connectivity-preserving flocking of multiagent systems is rapidly becoming a hot research topic, and various strategies have been developed including both centralized [13], [14] and decentralized approaches [15]–[23], which can be further divided into two main categories: 1) conservative connectivity preservation and 2) flexible connectivity preservation. The conservative connectivity preservation methods aim at preserving all the existing links as system topology evolves. Ji and Egerstedt [16], [18] introduced an edge tension function to keep all the neighboring agents within the maximum sensing range, and a similar situation was addressed in [20]. Zavlanos *et al.* [14], [19] proposed a gradient-based control strategy to guarantee that the set of the initial topological links is invariant as time evolves, the main drawback of which is that the connectivity has to be computed in a centralized way. Hybrid control laws adopting market-based auctions with gossip algorithms for addition and deletion of topological links were introduced in [23] and [24]. In addition, bounded artificial potential fields were introduced in [25]–[29] to cope with the connectivity-preserving rendezvous and formation control problems.

Although connectivity preservation could be formally guaranteed by using the above control algorithms, they are rather conservative and often impose too many redundant constraints on the system behavior. Moreover, maintaining all the existing edges of the underlying topology may hinder realization of flocking configuration. Alternatively, flexible connectivity preservation approaches allow the underlying communication topology to switch among different connected topologies, which means that some links are allowed to be removed or added as long as the overall graph is connected. Schuresko and Cortés [30] developed an algorithm which could maintain only the edges that define the spanning tree of the original graph. The concept of k-hop connectivity was introduced to allow the agents to move while maintaining connections [31], [32]. However, the connectivity preservation strategies require the k-hop information. Further relaxations on the constraints for connectivity maintenance can be introduced by keeping the global connectedness indicators, such as the second smallest eigenvalue λ_2 of the graph Laplacian, to be above a certain threshold over time. Kim and Mesbahi [13] proposed a centralized semidefinite programming method to optimally place a set of robots in order to realize a given value of λ_2. A similar objective was also pursued in [17] by devising a supergradient decentralized solution combined with gradient controllers. Gradient-like controllers aimed at maximizing the value of λ_2 over time based on the power iteration eigenvalue/eigenvector estimation technique were presented without considering obstacle or inter-robot collision avoidance [17], [33], [34].

A nice bit of the algorithms to estimate the eigenvalues of the Laplacian matrix are based on the centralized methods, which rely on the knowledge of all the elements of the Laplacian matrix. Some decentralized estimation algorithms have been reported [14], [33], [35]–[37]. Yang *et al.* [33] proposed a decentralized estimation scheme based on the power iteration algorithm combined with the proportional-integral (PI) average consensus estimators [38]. However, when the subdominant eigenvalue of the Laplacian matrix is close in magnitude to the dominant eigenvalue, the convergence rate of the power iteration algorithm could be very slow, which cannot satisfy the control requirements.

With respect to the state-of-the-art, the goal of this paper is to present a novel decentralized flocking control strategy with connectivity preservation which extends and generalizes the connectivity preservation methods in a flexible way. The contributions of this paper are twofold.

1) A new scheme of decentralized estimation of λ_2 as well as its corresponding eigenvector ϑ_2 based on the inverse power iteration algorithm is provided.
2) A new class of distributed connectivity-preserving flocking control protocols integrating the decentralized algebraic connectivity estimators are further proposed for second-order multiagent systems relying on the design of the novel hybrid potential fields. This set of hybrid potential fields exploited here can guarantee no-collision, desired distance stabilization, and connectivity of the underlying communication network simultaneously. Compared with the conservative connectivity-preserving strategies [16], [18], [19], the proposed connectivity-maintenance algorithm has flexible performance which allows the existing links to be switched. Consequently, the agent team can be separated into two connected parts and the communication burden can be reduced with the decreasing number of the links.

The remainder of this paper is organized as follows. In Section II, the problem is formally formulated and preliminaries are provided. A decentralized scheme for fully decentralized algebraic connectivity estimation based on inverse power iteration is presented in Section III. Section IV develops the distributed flocking protocols with flexible connectivity preservation based on the generalized hybrid potential fields. Stability of the overall estimation-based control system is analyzed in Section V. Comparative simulation and experimental results are provided in Section VI. Concluding remarks as well as directions for future work are stated in Section VII.

II. Problem Formulation and Preliminaries

Consider a group of N mobile agents moving in the 2-D Euclidean plane with second-order dynamics, which is described by

$$\begin{aligned} \dot{q}_i &= p_i \\ \dot{p}_i &= u_i \qquad i = 1, 2, \ldots, N \end{aligned} \tag{1}$$

where $q_i \in \mathbb{R}^2$ is the position vector of agent i, $p_i \in \mathbb{R}^2$ is the velocity vector of agent i, $u_i \in \mathbb{R}^2$ is the control input of agent i. Let $q = [q_1^T, q_2^T, \ldots, q_N^T]$ denotes the stack position vector of the multiagent systems. Each agent is considered to have the same limited communication radius R. The state dependent, switched system (1) induces the undirected dynamic graph

$$\mathcal{G} = \{\mathcal{V}, \mathcal{A}, \mathcal{E}\} \tag{2}$$

where $\mathcal{V} = \{1, 2, \ldots, N\}$ corresponds to the N agents. $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of communication links among agents. $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ is the neighbor set of an agent i with symmetry property $i \in \mathcal{N}_j \leftrightarrow j \in \mathcal{N}_i$. Proximity-limited communication is modeled by assigning to each link between agents i and j a weight which is defined as

$$a_{ij}\big(\|q_{ij}\|_2\big) = \begin{cases} 1, & \|q_{ij}\|_2 \in [0, \tau R) \\ \dfrac{1}{2}\left[1 + \cos\left(\pi \dfrac{\frac{\|q_{ij}\|}{R} - \tau}{1-\tau}\right)\right], & \|q_{ij}\|_2 \in [\tau R, R) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $q_{ij} = q_i - q_j$, τ is the parameter showing how the communication quality changes with the distance between agent i and j, and $0 < \tau < 1$. We assume $a_{ii} = 0$. Define the degree matrix $D = \mathrm{diag}(d_i)$ with the weighted degree $d_i = \sum_{j\in\mathcal{N}_i} a_{ij}$ as the diagonal elements. The Laplacian matrix of $\mathcal{G}$ is given by

$$L = D - \mathcal{A}. \tag{4}$$

Let $0 = \lambda_1 \le \lambda_2 \le \cdots \le \lambda_N$ be the eigenvalues of L in the nondecreasing order with the corresponding unit eigenvectors $\{v_1, v_2, \ldots, v_n\}$. The graph Laplacian exhibits the following remarkable properties.

Lemma 1 [39]: Given an undirected graph $\mathcal{G}$:

1) $L(\mathcal{G})$ is always symmetric and positive semidefinite and satisfies

$$x^T L x = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} a_{ij}\big(x_i - x_j\big)^2 \tag{5}$$

where $x = (x_1^T, x_2^T, \ldots, x_N^T)^T \in R^N$;
2) $\lambda_1 = 0, \lambda_2 > 0$ if $\mathcal{G}$ is connected;
3) $v_i^T L v_j = 0, \forall 1 \le i, j \le N, i \ne j$. Especially, $v_1 = \mathbf{1}_N$, where $\mathbf{1}_N$ is an N-dimensional column vector with all its elements being one;
4) the spectral radius is $\rho(L) = \lambda_N \le N$.

In 2), λ_2 is referred to as the algebraic connectivity which acts as an indicator of network connectedness. Therefore, the goal of the decentralized controller is to steer the system to

Algorithm 1 Decentralized Summation Estimation

Step 1. For each agent i, the PI average consensus estimator has the following form

$$\begin{cases} \dot{H}_i = K_1(f_i - H_i) - K_2 \sum_{j\in\mathcal{N}_i}\left(H_i - H_j\right) - K_3 \sum_{j\in\mathcal{N}_i}\left(\mathcal{M}_i - \mathcal{M}_j\right) \\ \dot{\mathcal{M}}_i = -K_3 \sum_{j\in\mathcal{N}_i}\left(H_i - H_j\right) \end{cases} \tag{6}$$

where f_i is the initial scalar measured by agent i, H_i is the average estimate of $\frac{\sum_{i=1}^{N} f_i}{N}$ for agent i, K_1 is the rate that new information replaces old information, and K_2, K_3 are estimator gains.

Step 2. The estimation of the sum of the initial scalars $\sum_{i=1}^{N} f_i$ for each agent can be obtained by

$$F_i = H_i * N \tag{7}$$

where N is the number of the agents in the MAS network, and F_i is the estimation result. When the PI average consensus estimator (6) converges, F_i can be treated as the final result of Algorithm 1.

achieve the stable flocking configuration while guaranteeing connectivity of the underlying graph, i.e., $\lambda_2(L) > 0$, provided the given graph is initially connected.

III. Decentralized Estimation of Algebraic Connectivity λ_2

In order to overcome the drawback of the power iteration algorithm, we devise a decentralized inverse power iteration scheme, which can estimate the connectivity much faster than the standard power iteration scheme with properly chosen parameters. The operation of the decentralized inverse power iteration can be converted into solving a series of related sets of nonhomogeneous linear equations. For each set of nonhomogeneous linear equations, the conjugate gradient (CG) algorithm is employed to solve the nonhomogeneous linear equations. And a decentralized summation computation algorithm inspired by [33] is further applied in this paper to make the CG algorithm fully decentralized.

A. Decentralized Conjugate Gradient Algorithm

It is well known that CG algorithm has been widely used in solving linear equations, but normally the key procedure in computation of CG algorithm, for example, the inner product calculation, has to rely on the knowledge of all the elements in the correlation matrix, which makes the traditional CG algorithm centralized. In order to cope with this problem, a decentralized CG algorithm is proposed based on the decentralized summation computation in which, inspired by [33], the distributed PI average consensus algorithm is applied in this paper to decentralize the summation operation.

By combining the traditional CG algorithm [40] with the proposed decentralized summation computation algorithm, a decentralized CG algorithm is provided as follows.

Algorithm 2 Decentralized CG

Step 1. For nonhomogeneous linear equations $Ax = b$, where A is an N-dimensional positive square matrix, b is a known N-dimensional vector, and x is the solution vector of these equations. $w^{(0)}$ is setted associated to the random initial vector $x^{(0)}$ and the equations by

$$w^{(0)} = -r^{(0)} = b - Ax^{(0)} \tag{8}$$

for the initial iteration step $k = 0$, where $w^{(k)}$ is the direction-revising vector and $r^{(k)}$ is the residual vector in the CG iteration step k.

Step 2. At the iteration step k, $x^{(k+1)}$ can be updated as

$$x_i^{(k+1)} = x_i^{(k)} + \alpha^{(k)} w_i^{(k)}, \quad i = 1, \ldots, N \tag{9}$$

where

$$\alpha^{(k)} = -\frac{\left\langle r^{(k)}, w^{(k)}\right\rangle}{\left\langle w^{(k)}, Aw^{(k)}\right\rangle} \tag{10}$$

and $\left\langle r^{(k)}, w^{(k)}\right\rangle = \sum_{i=1}^{N} r_i^{(k)} w_i^{(k)}$, $\left\langle w^{(k)}, Aw^{(k)}\right\rangle = \sum_{i=1}^{N}(w_i^{(k)} \sum_{m=1}^{N} A_{i,m} w_m^{(k)})$. Since the computations of $\left\langle r^{(k)}, w^{(k)}\right\rangle$ and $\left\langle w^{(k)}, Aw^{(k)}\right\rangle$ have to rely on all the elements of $r^{(k)}$, $w^{(k)}$ and the matrix A, Algorithm 1 is applied to make these computations decentralized.

Step 3. If the current iteration step $k < N - 1$, $r_i^{(k+1)}$, $\beta^{(k+1)}$, $w_i^{(k+1)}$ can be updated for the next iteration step as follows.

$$\begin{cases} r_i^{(k+1)} = \sum_{m=1}^{N} A_{i,m} x_i^{(k+1)} - b_i \\ \beta^{(k+1)} = \frac{\left\langle r^{(k+1)}, Aw^{(k)}\right\rangle}{\left\langle w^{(k)}, Aw^{(k)}\right\rangle} \\ w_i^{(k+1)} = -r_i^{(k+1)} + \beta^{(k)} w_i^{(k)} \end{cases} \tag{11}$$

where the computation of $\left\langle r^{(k+1)}, Aw^{(k)}\right\rangle = \sum_{i=1}^{N}(r_i^{(k+1)} \cdot \sum_{m=1}^{N} A_{i,m} w_m^{(k)})$ and $\sum_{m=1}^{N} A_{i,m} x_i^{(k+1)}$ can be also decentralized by Algorithm 1. After that, return to step 2 of the decentralized CG algorithm and set $k = k + 1$.

Step 4. If the current iteration step is $k \geq N - 1$ and $\|x_i^{(k+1)} - x_i^{(k)}\| \leq e_c$, where e_c is the required accuracy of the algorithm, then each element $x_i^{(k)}$ can be obtained to compose the solution to the equation $Ax = b$ as $x = x^{(k)}$.

Remark 1: In Algorithm 2, $\langle\cdot\rangle$ denotes the inner product of two vectors; $A_{i,m}$ is the (i,m)th element of matrix A; $w_m^{(k)}(x_i^{(k+1)}, x_i^{(k)}, r_i^{(k+1)}, w_i^{(k+1)})$ is the mth (ith) element of vector $w^{(k)}(x^{(k+1)}, x^{(k)}, r^{(k+1)}, w^{(k+1)})$. Please note that the computation of $\alpha^{(k)}$ and $\beta^{(k+1)}$ has been decentralized by using Algorithm 1. Derived from the conclusion of [40], it can be concluded that when the decentralized CG algorithm is accomplished, the vector $x^{(N-1)}$ would converge to the solution of $Ax = b$ in a fully decentralized way.

B. Decentralized Inverse Power Iteration Scheme

Since the decentralized inverse power iteration can be converted into solving a series of related sets of the nonhomogeneous linear equations, the following deflation of L is applied:

$$\hat{L} = L + \frac{N+\delta}{N}\mathbf{1}\mathbf{1}^T \tag{12}$$

where $\delta \in \mathbb{R}$, $\delta > 0$, the set of eigenvalues of $\hat{L}$ is $\{\lambda_2, \dots, \lambda_N, N+\delta\}$ with the associated set of eigenvectors $\{v_2, \dots, v_n, \mathbf{1}_N/\sqrt{N}\}$.

Define the matrix $(\hat{L} - \mu I)^{-1}$ with $\mu \in \mathbb{R}$. It follows that the set of eigenvectors is the same as that of $\hat{L}$ with the corresponding set of the eigenvalues:

$$\left\{(N+\delta-\mu)^{-1}, (\lambda_N - \mu)^{-1}, \dots, (\lambda_2 - \mu)^{-1}\right\}. \tag{13}$$

The inverse power iteration procedure converges to the dominant eigenpair of $(\hat{L}-\mu I)^{-1}$, which is $\{(\lambda_2-\mu)^{-1}, v_2\}$, with the following iteration:

$$\left(\hat{L} - \mu I\right)^{-1}\hat{v}_2^{(k)} = \hat{v}_2^{(k+1)} \tag{14}$$

where $k+1$ is the step number of the inverse power iteration.

The convergence rate of $\hat{\lambda}_2$ approaching to λ_2 is related with the convergence factor $\gamma = |((\lambda_2 - \mu)/(\lambda_3 - \mu))|$, where the closer γ is to 0, the higher the convergence rate is. The symmetric and positive definiteness of $(L-\mu I)^{-1}$ can be ensured when $0 \le \mu < \lambda_2$. Hence the convergence rate of the inverse power iteration would be much higher than that of the standard power iteration when the parameter μ is thoughtfully adjusted [40]. In this paper, μ can be deliberately chosen according to the preliminary knowledge about λ_2 and the properties of λ_2 proposed in [41].

In the inverse iteration algorithm, computation of matrix inversion is not only necessary, but very time-consuming. Moreover, matrix inversion computation usually has to rely on all the elements of the matrix, for example, all the elements of the Laplacian L in this case, which unfortunately is inaccessible for each agent. Therefore, direct matrix inversion cannot be employed in multiagent systems, so instead of any direct method, an indirect iteration method for computing matrix inversion is applied.

At the inverse power iteration step $k+1$, assume that $\hat{v}_2^{(k)}$ has been figured out from the preceding step, (14) can be considered as a set of nonhomogeneous linear equations. This idea can be summarized by the following update law:

$$\left(\hat{L} - \mu I\right)\hat{v}_2^{(k+1)} = \hat{v}_2^{(k)}. \tag{15}$$

To realize this update law in a decentralized way, the proposed decentralized CG algorithm is used to figure out the $(k+1)$th iteration vector $\hat{v}_2^{(k+1)}$ of $E\hat{v}_2^{(k+1)} = \hat{v}_2^{(k)}$, where $E = (\hat{L} - \mu I)$. Hence, based on the above work, a decentralized algebraic connectivity estimation scheme is exploited.

Theorem 1: Agent i can work out its corresponding element $\hat{v}_{2,i}^{(k+1)}$ in the estimated eigenvector $\hat{v}_2^{(k+1)}$ at the inverse power iteration step $k+1$ in a decentralized way by Scheme 1,

Scheme 1 Decentralized Algebraic Connectivity Estimation

Step 1. Firstly, each element of the corresponding row of the preconditioned matrix E is calculated by agent i, where

$$E_{i,j} = \left(L_{i,j} + \frac{N+\delta}{N} - \zeta\mu\right), \quad j = 1, \dots, N, \tag{16}$$

and $\zeta = \begin{cases} 0, j \neq i \\ 1, j = i \end{cases}$.

Step 2. For each agent i, Algorithm 2 is applied to deal with the corresponding work in solving the nonhomogeneous linear equations $E\hat{v}_2^{(k+1)} = \hat{v}_2^{(k)}$. At the iteration step $k+1$, agent i can obtain the corresponding element $\hat{v}_{2,i}^{(k+1)}$ of the estimated solution vector.

Step 3. $\hat{\lambda}_2^{(k+1),i}$ can be estimated by agent i as

$$\hat{\lambda}_2^{(k+1),i} = \frac{\sum\limits_{j \in (\mathcal{N}_i \cup i)} L_{i,j}\hat{v}_{2,j}^{(k+1)}}{\hat{v}_{2,i}^{(k+1)}} \tag{17}$$

If $\|\hat{\lambda}_2^{(k+1),i} - \hat{\lambda}_2^{(k),i}\| \le e$, where e is the estimation accuracy that the final estimated eigenvalue should satisfy, then the estimation result can be obtained as $\hat{\lambda}_2^i = \hat{\lambda}_2^{(k+1),i}$; otherwise, return $\hat{v}_{2,i}^{(k+1)}$ to step 2 of the estimation scheme and set $k = k+1$.

then the overall eigen-pair estimation scheme can successfully deal with the estimated connectivity $\hat{\lambda}_2$ and its corresponding eigenvector $\hat{v}_2$ in a fully decentralized way.

Proof: Since the convergence time of the PI average consensus estimators in Algorithm 1 is significantly less than the time consumed by steps 2 and 3 in Algorithm 2, according to the situations in [40] and [42], the decentralized CG algorithm 2 will agree with the standard CG algorithm. So with the values of $\alpha^{(j)}$ and $\beta^{(j+1)}$ obtained by Algorithm 1, we can accomplish Algorithm 2 in the decentralized way, obtaining the final $\hat{v}_{2,i}^{(k+1)}$ by each corresponding agent at the inverse power iteration step $k+1$. As the results of every decentralized inverse power iteration step will converge, it can be shown that the entire algebraic connectivity estimation scheme runs successfully. So we can get the estimation value of eigenvector $\hat{v}_2$ and eigenvalue $\hat{\lambda}_2$ in a decentralized way. ∎

Remark 2: For the heterogeneous multi-agent system (MAS) networks with directed topology, the necessary condition to perform the overall algorithm is that the directed graph is strongly connected. Then the proposed scheme should be improved to guarantee that for the directed graph of $\mathcal{A}$, the matrix $\mathcal{A}$ is irreducible, which is the future work to be carried out next.

In conclusion, the flow chart describing the proposed algebraic connectivity estimation scheme for agent i is displayed as Fig. 1.

IV. Controller Development

Flocking control aims at steering the agents to asymptotically achieve velocity alignment, the desired interagent distance stabilization, and collision avoidance. Besides, the connectivity of the underlying network should be maintained.

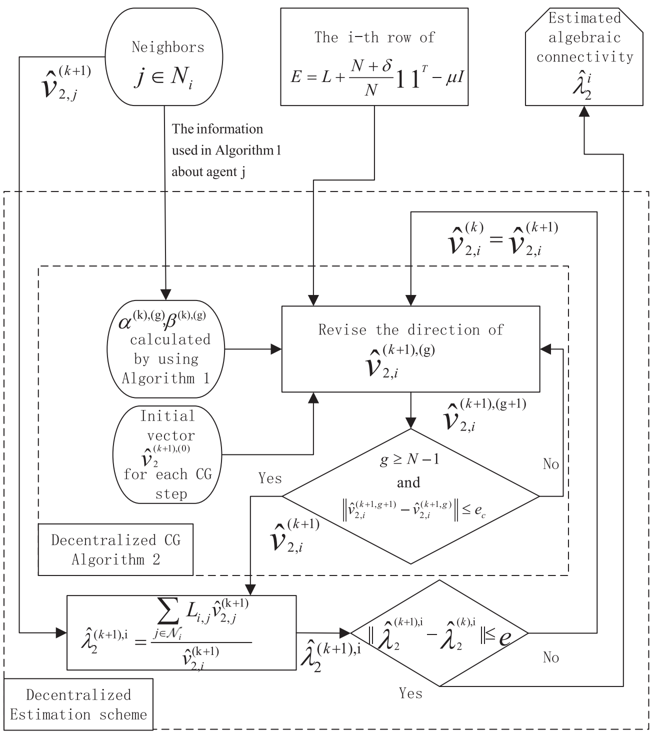


Fig. 1. Flow chart for the algebraic connectivity estimation scheme for agent i.

The corresponding distributed control protocol can be designed as follows:

$$u_i = -\left(\nabla_{q_i} V_i^a + l_i \cdot \nabla_{q_i} V_i^t\right) - \sum_{j \in \mathcal{N}_i} \nabla_{q_i} V_{ij}^c - \sum_{j \in \mathcal{N}_i} a_{ij}\left(p_i - p_j\right) \tag{18}$$

where V_i^a is the artificial potential field with obstacle avoidance, V_i^t is the artificial potential field with target tracking, $l_i = 1$ if agent i can observe the target, otherwise $l_i = 0$. V_i^a and V_i^t can be simply chosen as $V_i^a = (1/((\|q_{io}\| - r)^2))$ and $V_i^t = \|q_{it}\|^2$, where $q_{io} = q_i - q_o$, q_o is the center of the obstacle, r is the radio of the obstacle, $q_{it} = q_i - q_t$, q_t is the center of the target area. V_{ij}^c is the artificial potential field with connectivity preservation, a_{ij} smoothly decreases from 1 to 0 when the relative distance between agents i and j increases from 0 to R, as can be seen from (3).

Actually, from Lemma 1, it is sufficient for connectivity preservation by controlling $\lambda_2 > 0$. Since $\hat{\lambda}_2$ has been estimated by each agent only relying on local information, similar to our previous work [35], a hybrid potential field could be formulated as follows:

$$V_{ij}^c = \begin{cases} \left\| \dfrac{1}{\|q_{ij}\|} - \dfrac{1}{s} \right\|^{k_1} \dfrac{1}{\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2}} & \|q_{ij}\| \in (0, s) \\ \dfrac{k_3}{2} \dfrac{1 - \cos\left(\pi \frac{\|q_{ij}\| - s}{R - s}\right)}{\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2}} & \|q_{ij}\| \in [s, R) \\ \dfrac{k_3}{\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2}} & \|q_{ij}\| \in [R, \infty) \end{cases} \tag{19}$$

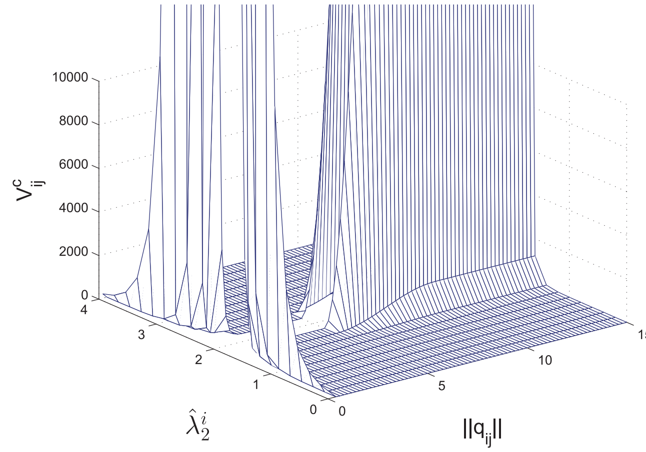


Fig. 2. Hybrid potential field V_{ij}^c.

where $k_3 > 0$, $k_1, k_2 > 1$, $0 < s < R$ is the desired interagent distance, $\hat{\lambda}_2^i$ is the estimate of λ_2 by agent i, and R is the communication radius of agent. A possible shape of V_{ij}^c is illustrated in Fig. 2 with $R = 10, k_1 = 12, k_2 = 6, k_3 = 5$, $s = 5$, and $\tilde{\varepsilon} = 2$.

Correspondingly, we have

$$\nabla_{q_i} V_{ij}^c = \begin{cases} \dfrac{k_2}{\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2+1}} \left\| \dfrac{1}{\|q_{ij}\|} - \dfrac{1}{s} \right\|^{k_1} \dfrac{\partial \hat{\lambda}_2^i}{\partial q_i} \\ - \dfrac{k_1}{\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2}} \left\| \dfrac{1}{\|q_{ij}\|} - \dfrac{1}{s} \right\|^{k_1 - 1} \dfrac{q_{ij}}{\|q_{ij}\|^3}, & \|q_{ij}\| \in (0, s) \\ - \dfrac{k_3 k_2}{2\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2+1}} \left[1 - \cos\left(\pi \dfrac{\|q_{ij}\| - s}{R - s}\right)\right] \dfrac{\partial \hat{\lambda}_2^i}{\partial q_i} \\ + \dfrac{k_3 \pi}{2\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2} (R - s)} \sin\left(\pi \dfrac{\|q_{ij}\| - s}{R - s}\right) \dfrac{q_{ij}}{\|q_{ij}\|}, & \|q_{ij}\| \in [s, R) \\ - \dfrac{k_3 k_2}{\left(\hat{\lambda}_2^i - \tilde{\varepsilon}\right)^{k_2+1}} \dfrac{\partial \hat{\lambda}_2^i}{\partial q_i}, & \|q_{ij}\| \in [R, \infty) \end{cases} \tag{20}$$

where $\tilde{\varepsilon}$ is the connectivity threshold which will be determined later, and

$$\frac{\partial \hat{\lambda}_2^i}{\partial q_i} = \frac{\partial\left(\left(\hat{v}_2^i\right)^T L \hat{v}_2^i\right)}{\partial q_i}. \tag{21}$$

From (19) and (20), it is obvious that V_{ij}^c is continuously differentiable, and the desired relative distance between agent i and agent j can be uniquely determined by minimizing V_{ij}^c. Furthermore, the gradient of V_{ij}^c indicates that if the initial value of λ_2 locates in the area where $\lambda_2 > \tilde{\varepsilon}$, then the gradient-based control protocol (18) can guarantee that the value of λ_2 never goes below $\tilde{\varepsilon}$, maintaining the connectivity during the whole maneuver process.

V. Stability Analysis

With the estimated value of λ_2 and v_2 denoted by $\hat{\lambda}_2^i$ and $\hat{v}_2$ for agent i, the distributed connected flocking control protocol can be obtained. By utilizing the normalized eigenvector

$\hat{v}_2$ obtained by inverse power iteration, (21) can be further deduced as follows:

$$\frac{\partial \hat{\lambda}_2^i}{\partial q_i} = \frac{\partial\left(\left(\hat{v}_2\right)^T L \hat{v}_2\right)}{\partial q_i} = \frac{\partial\left(\hat{v}_2\right)^T}{\partial q_i} L \hat{v}_2 + \left(\hat{v}_2\right)^T \frac{\partial L}{\partial q_i} \hat{v}_2 + \left(\hat{v}_2\right)^T L \frac{\partial \hat{v}_2}{\partial q_i}. \tag{22}$$

Since L is always symmetric positive semi-definite for undirected graphs from Lemma, it can be guaranteed that

$$\left(\hat{v}_2\right)^T L \frac{\partial \hat{v}_2}{\partial q_i} = \frac{\partial\left(\hat{v}_2\right)^T}{\partial q_i} L \hat{v}_2 = \lambda_2 \frac{\partial\left(\hat{v}_2\right)^T}{\partial q_i} \hat{v}_2 = \frac{\lambda_2}{2} \frac{\partial\left[\left(\hat{v}_2\right)^T \hat{v}_2\right]}{\partial q_i} = 0. \tag{23}$$

Then, we can obtain that

$$\frac{\partial \hat{\lambda}_2}{\partial q_i} = \left[\left(\hat{v}_2\right)^T \frac{\partial L}{\partial q_{i,x}} \hat{v}_2 \quad \left(\hat{v}_2\right)^T \frac{\partial L}{\partial q_{i,y}} \hat{v}_2\right]^T = \sum_{j \in \mathcal{N}_i} \frac{\partial a_{ij}}{\partial q_i} \left(\hat{v}_{2,i} - \hat{v}_{2,j}\right)^2. \tag{24}$$

Combining (20) and (24) with (18), it is obvious that the control protocol of each agent can be obtained in a fully distributed way.

The stability of group evolution is discussed by considering the following positive semi-define Lyapunov function candidate:

$$V = \sum_{i=1}^{N} \left(\sum_{j \in \mathcal{N}_i} V_{ij}^c + V_i^a + l_i \cdot V_i^t \right) + \frac{1}{2} \sum_{i=1}^{N} p_i^T p_i. \tag{25}$$

Then the stability result of the overall system can be deduced by the following theorem.

Theorem 2: Consider the multiagent systems consisting of N agents steered by double integrator dynamics (1) and steered by the distributed control law (18). If $V(0)$ is finite and the initial value of $\lambda_2(0) > \varepsilon + 2e$, then $\lambda_2 > \varepsilon$ for all time $t \geq 0$, where ε is related to $\widetilde{\varepsilon}$ with $\widetilde{\varepsilon} = \varepsilon + e$. e is the estimation error between $\hat{\lambda}_2$ and λ_2. The agents asymptotically approach the same velocities and reach the target point, collisions among agents and collisions between obstacle and agents are avoided, and the global connectivity of the underlying network is maintained.

Proof: Take the derivative of (25) with respect to time, we have

$$\dot{V} = \sum_{i=1}^{N} p_i^T \left(\sum_{j \in \mathcal{N}_i} \nabla_{q_i} V_{ij}^c + \nabla_{q_i} V_i^a + l_i \cdot \nabla_{q_i} V_i^t \right) + \sum_{i=1}^{N} p_i^T \dot{p}_i = -p^T (L \otimes I_2) p \leq 0 \tag{26}$$

where $\otimes$ denotes the Kronecker product. Since L is always symmetric and positive semidefinite, we have $\dot{V} \leq 0$. Because the initial energy is finite, at the same time, as indicated in (25) and (26), $V(0)$ is finite, the non-negative V will keep finite for all time. Note that, according to (26), the fact that $\lambda_2(0) > \varepsilon + 2e$ implies $\hat{\lambda}_2^i(0) \geq \varepsilon + e = \widetilde{\varepsilon}$, $\forall i$. Then from (18), (26), and the boundedness of the initial energy, we have $\hat{\lambda}_2^i(t) \geq \widetilde{\varepsilon}$, $\forall t \geq 0$ for each agent i, which ensures that the value of $\lambda_2(t) \geq \widetilde{\varepsilon} - e = \varepsilon$. Therefore, the global connectivity of the underlying network is preserved and bounded below by $\varepsilon > 0$ for all time. Furthermore, from (25), we have $(1/2)\sum_{i=1}^{N} p_i^T p_i \leq V(t) \leq V(0)$, which implies $\|p_i(t)\| \leq \sqrt{2V(0)}$, $\forall i$, and $V_{ij}(t) \leq V(t) \leq V(0) \ll \infty$, $\forall j \in \mathcal{N}_i$. From (18), $V_{ij}^c(t) \ll \infty$ guarantees $\|q_{ij}(t)\| > 0$ and $\lambda_2(t) > \varepsilon$. The former guarantees that the collision avoidance between neighbors i and j is always achieved. The latter implies that $\mathcal{G}(t) \in \mathbb{C}$ for all time. Thus any two agents in the group are connected by a path with the distance no more than $(N-1)R$. Therefore, the set $\Omega = \{(q_{ij}(t), p_i(t)) | V(t) \leq V(0), \forall t \geq 0\}$ is closed and bounded, hence compact. Since system (1) with control input (18) is an autonomous system on the concerned time interval $[0, \infty)$, LaSalle's invariance principle can be applied to infer that for any initial state starting from Ω, its trajectories will converge to the largest positively invariant set inside the region $M = \{(q_{ij}, p_i \in \Omega) | \dot{V} = 0\}$

$$\dot{V} = -p^T (L \otimes I_2) p = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \left(p_i - p_j\right)^2 = 0. \tag{27}$$

Since $\mathcal{G} \in \mathbb{C}$, where $\mathbb{C}$ denotes the set of all the connected graphs, (27) holds if and only if $p_1 = p_2 = \cdots = p_N = p^*$, which implies that all agents asymptotically move with the same velocity, which yields that

$$\dot{p} = \begin{bmatrix} -\sum_{j \in \mathcal{N}_1} \nabla_{q_1} V_{1j}^c\left(\|q_{1j}\|\right) - \nabla_{q_1} V_1^a(\|q_{1o}\|) - l_1 \cdot \nabla_{q_1} V_1^t(\|q_{1t}\|) \\ -\sum_{j \in \mathcal{N}_2} \nabla_{q_2} V_{2j}^c\left(\|q_{2j}\|\right) - \nabla_{q_2} V_2^a(\|q_{2o}\|) - l_2 \cdot \nabla_{q_2} V_2^t(\|q_{2t}\|) \\ \vdots \\ -\sum_{j \in \mathcal{N}_N} \nabla_{q_N} V_{Nj}^c\left(\|q_{Nj}\|\right) - \nabla_{q_N} V_N^a(\|q_{No}\|) - l_N \cdot \nabla_{q_N} V_N^t(\|q_{Nt}\|) \end{bmatrix} = 0. \tag{28}$$

From (28), almost all the solutions of the system dynamics except for saddle points and local maxima converge to one of a local minimum of V. After the analysis similar to [7], each local minima of V corresponds to a stable flocking configuration, i.e., all the agents asymptotically converge to the desired interagent distances, reach the target point and avoid collision with the obstacles. ■

VI. Simulations and Experiments

In this section, simulations and experiments are performed to validate our proposed decentralized estimation algorithm with effective performance and distributed flocking control strategies with flexible connectivity preservation.

A. Simulation Results

The initial position of a seven-agent network was depicted in Fig. 3 along with interagent communication links represented by solid lines. The Laplacian spectrum of the initial topology is $\{0, 0.555, 0.634, 2.045, 2.778, 3.632, 4.746\}$, where the eigenvector associated with λ_2 is $v_2 = \{-0.156, -0.496, 0.136, -0.221, 0.487, -0.320, 0.569\}$. The communication radius is set to be $R = 8$ m, the corresponding

TABLE I
SIMULATION RESULTS OF DIFFERENT μ WITH DIFFERENT λ_2, λ_3, AGENT NUMBER, AND DENSITY

Time \ (λ_2,λ_3,N,D) / μ	(0.5545,0.6336,7,0.42)	(0.8550,1.0106,10,0.47)	(0.9449,1.8607,7,0.63)
μ =0.45	0.56s	2.32s	0.90s
μ =0.55	0.38s	2.02s	0.88s
μ =0.75	None	1.52s	0.74s
μ =0.85	None	0.51s	0.58s
μ =0.93	None	None	0.35s
Power Iteration	0.85s	0.54s	0.43s

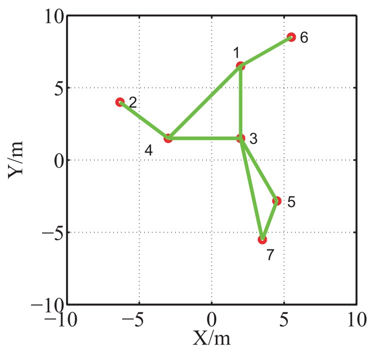


Fig. 3. Initially connected team position and topology of seven-agents network.

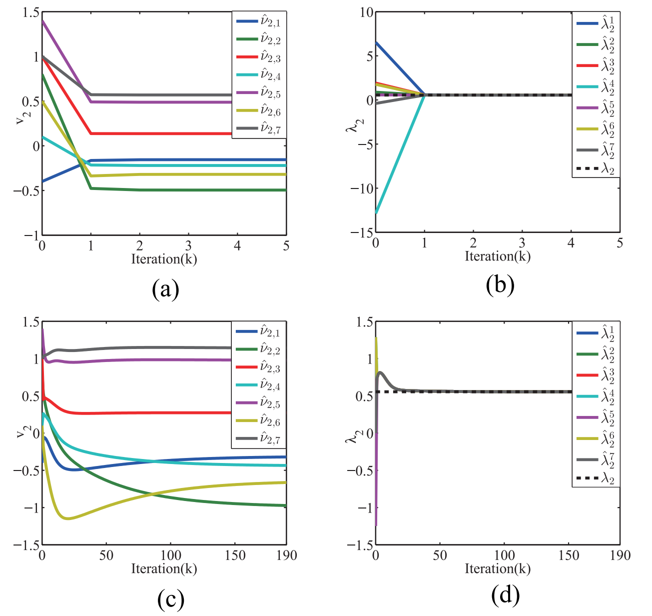


Fig. 4. Estimation of λ_2 and v_2 about estimation accuracy $e = |\|\hat{\lambda}_2^{(k+1)}\| - \|\hat{\lambda}_2^{(k)}\|| = 10^{-7}$. Estimation of (a) v_2 using inverse power iteration, (b) λ_2 using inverse power iteration, (c) v_2 using power iteration, and (d) λ_2 using power iteration.

parameters are properly chosen as $\tau = 0.8$ in (3), $K_1 = 5$, $K_2 = 1000$, $K_3 = 10$ in (6), $\sigma = 0.7$ in (12), and $\mu = 0.55$ in (14).

Please note that the initial value of μ is chosen based on the preliminary knowledge about λ_2 of the MAS networks. Moreover, μ would be updated dynamically less than the converged value of $\hat{\lambda}_2$ during the process when the agents are driven to move. Other parameters used in the connectivity estimator such as K_1, K_2, and K_3 are determined to guarantee that the time constant of the PI average consensus estimation in Algorithm 1 is significantly less than that of the eigenvector estimation, and the time constant of the eigenvector estimation is even greatly less than that of the connectivity maintenance controller.

First, the comparative convergence results of the estimated $\hat{v}_2$ and $\hat{\lambda}_2$ by using both the power iteration in [33] and the inverse power iteration are demonstrated in Fig. 4. The estimation of the eigenvector and eigenvalue are denoted by colored solid lines, and the actual value of the eigenvalue is denoted by black dashed line.

With the estimation accuracy $e = |\|\hat{\lambda}_2^{(k+1)}\| - \|\hat{\lambda}_2^{(k)}\|| = 10^{-7}$, it is clear that by exploiting our proposed inverse power iteration algorithm, each element of the estimated eigenvector $\hat{v}_2$ and $\hat{\lambda}_2$ converges to the fine value v_2 and λ_2 much faster than the power iteration. The iteration step and consumed time of our algorithm are, respectively, 5 and 0.38 s, while the iteration step and consumed time of the power iteration are, respectively, 190 and 0.85 s.

In order to fully demonstrate the proposed scheme, some comparative simulations are carried out with both different network density and different number of agents. Here the network density is defined as the ratio of existing links to the total number of possible links [43]. The comparative simulation results are shown by Table I. From this table, it is clear that the closer μ is set to λ_2, the higher convergence rate we can obtain. Meanwhile with almost the same value of μ, the convergence rate would be lower when the MAS network becomes denser, which is because μ is farther from λ_2 when λ_2 becomes larger under this condition. While it is also noticed that for MAS networks with relatively larger size, much more convergence time has to be consumed when μ is set far from λ_2. However, the results can be greatly improved by setting a proper value of μ with our proposed algorithm.

For the case when λ_3 is close to λ_2, the proposed inverse power iteration scheme can provide much faster convergence rate than the power iteration method even when μ is set with more flexibility only less than λ_2. While when λ_3 is relatively far from λ_2, the value of μ must be chosen quite precisely close to λ_2 whose exact value is rather difficult to obtain. The drawback of this operation can be moderated by further designing more powerful algorithms to reduce the computation time cost for each iteration step, which would make the convergence rate not sensitive to the precise value of λ_2.

Furthermore, to highlight the significance of connectivity preservation in achieving stable flocking behavior, comparative

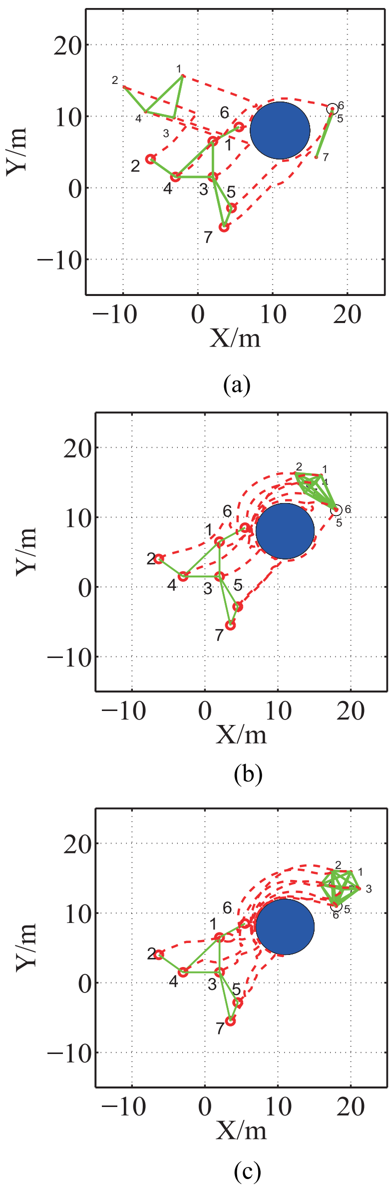


Fig. 5. Trajectory history of multiagent system in flocking task. Trajectory history (a) without connectivity preservation, (b) with our connectivity preservation algorithm, and (c) with conservative connectivity preservation algorithm.

simulations are also performed to compare among the flocking algorithm without using connectivity preservation, with the flexible connectivity preservation proposed in this paper, and the conservative connectivity preservation algorithm [19]. These results are demonstrated by Fig. 5, in which seven agents are labeled with red dots.

The initial positions and velocities are randomly chosen in the range of $[-10, 10]m \times [-10, 10]m$ and $[-2, 4]m/s \times [-2, 4]m/s$, respectively, such that the network is initially connected. $\tilde{\varepsilon}$ in (19) and (20) is chosen as $\tilde{\varepsilon} = 0.3$. There is a circular obstacle centered at $[\text{Obstacle}_x = 11, \text{Obstacle}_y = 8]$ with a reasonable diameter of $[\text{Obstacle}_r = 8]$; and we chose a circular area with the center $[\text{Termini}_x = 18, \text{Termini}_y = 11]$ and the diameter $[\text{Termini}_r = 1]$ as the target area. Only agents 5 and 6 know the information of the target area.

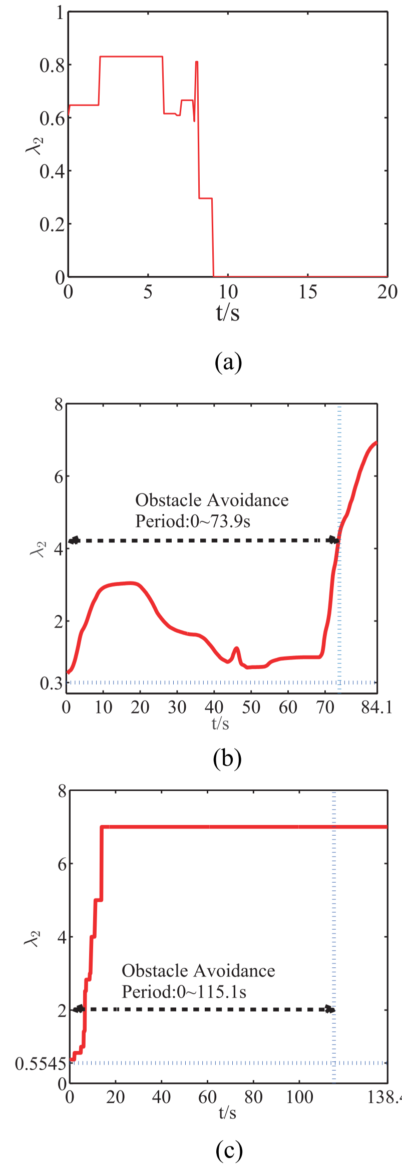


Fig. 6. Algebraic connectivity λ_2 over time. (a) λ_2 without connectivity preservation. (b) λ_2 with our connectivity preservation algorithm. (c) λ_2 with conservative connectivity preservation algorithm.

Obviously, Fig. 5(a) shows that without connectivity preservation, the flocking task failed in this obstacle avoidance situation, while as Fig. 5(b) and (c) displays, with connectivity preservation algorithms, the multiagent system can accomplish this flocking mission. At the same time, with our flexible connectivity preservation algorithm, the team allows the links to be broken while the connectivity is still maintained. Consequently the group of the agents can be separated into two parts and the connected two parts can move toward the target while avoiding the obstacle. While in Fig. 5(c), all the agents have to move in a block when avoiding the obstacle, which imposes too many redundant constraints on the movement.

The evolving curves of the algebraic connectivity λ_2 are depicted in Fig. 6. Due to the connectivity preservation, it can be observed that the value of λ_2 never decreases to zero. And because of the flexibility of our method, the obstacle avoidance time in Fig. 6(b) is 73.9 s less than that in Fig. 6(c).

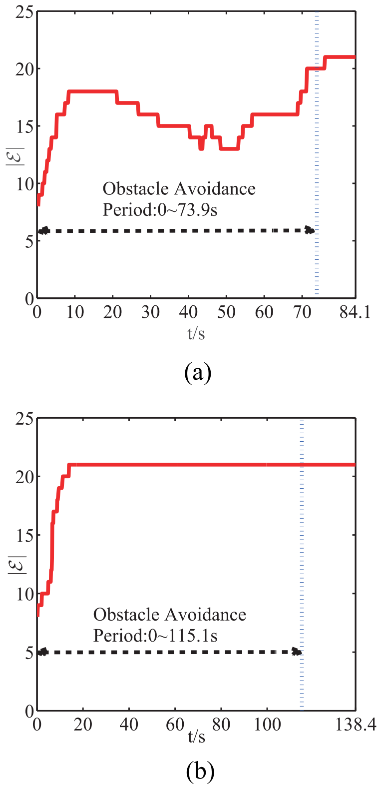


Fig. 7. Total number of edges in the network over time in simulations with (a) our connectivity preservation algorithm and (b) conservative connectivity preservation algorithm.

The numbers of the edges in the underlying communication topology are shown in Fig. 7. As expected, it can be seen from Fig. 7(a) and (b) that the number of the edges is not monotonically increasing with adoption of the flexible connectivity maintenance strategy, which costs less energy on communication in the obstacle-avoidance process.

B. Experimental Results

The experiments of flocking with obstacle avoidance are carried out with four Pioneer3-AT mobile robots and one Pioneer3-DX mobile robot to validate the practical effectiveness of the proposed distributed flocking control algorithm.

The task presented is to avoid the obstacle and reach the target area, only two of the robots can sense the target. As shown in Figs. 8 and 9, the dimension of the experiment place is 7 m $\times$ 8 m in which one pole is used as the obstacle area. The comparative experiments are also performed to compare the proposed flexible connectivity preservation algorithm with the conservative connectivity preservation algorithm. The experiment results show that it costs 17 s for the proposed distributed flocking algorithm to avoid the obstacle which is less than that of the conservative one. On the other hand, because the resulting team formation becomes more loose, the proposed method has to take more time to rebuild the team formation and finally achieve the target. Whereas, it is illustrated that our method is still more effective than the conservative one.

The snapshots on several important instants are depicted in Figs. 8 and 9, which testify the practical advantages of the proposed distributed flocking algorithm. The evolution of the numbers of the edges and λ_2 for the two situations are depicted in Figs. 10 and 11, respectively. As shown in the figures, the number of the edges with our flexible algorithm changes during the obstacle-avoidance period, leading to a flexible team formation. On the contrary, the number of the edges with the conservative algorithm is monotonically increasing, which is not always feasible for obstacle avoidance under some complex conditions.

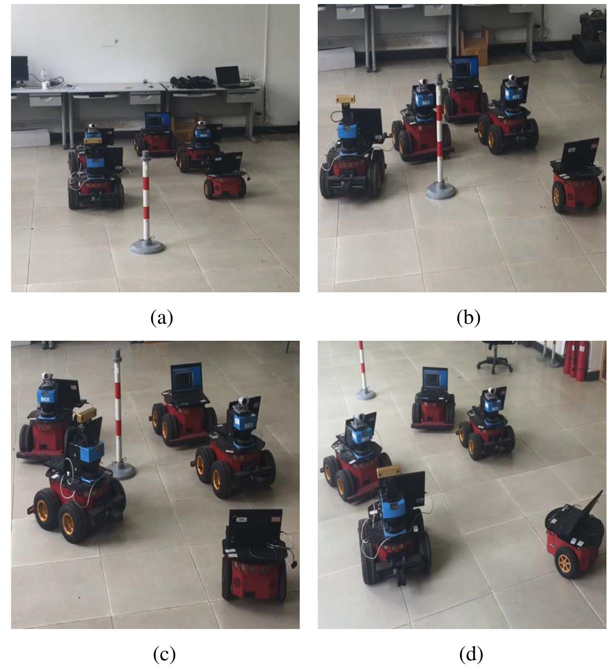


Fig. 8. Formation of MAS with our connectivity preservation algorithm. (a) $T = 0$ s. (b) $T = 8$ s. (c) $T = 15$ s. (d) $T = 30$ s.

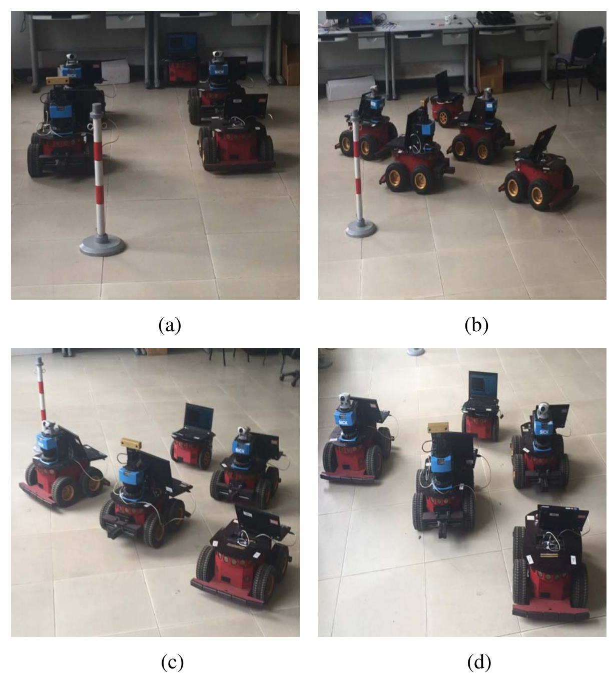


Fig. 9. Formation of MAS with conservative connectivity preservation algorithm. (a) $T = 0$ s. (b) $T = 7$ s. (c) $T = 26$ s. (d) $T = 35$ s.

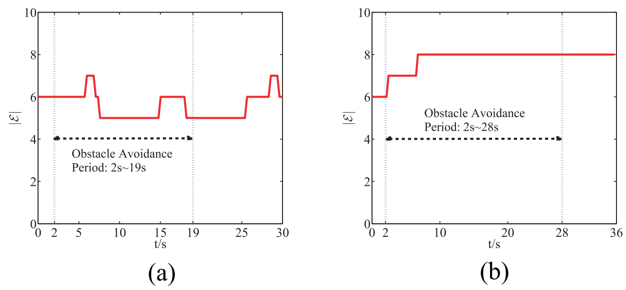


Fig. 10. Total number of edges in the network over time in experiments with (a) our connectivity preservation algorithm and (b) conservative connectivity preservation algorithm.

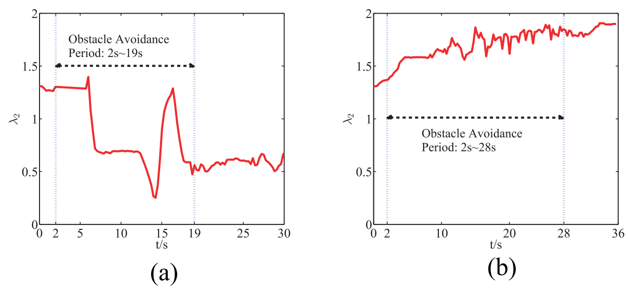


Fig. 11. Algebraic connectivity λ_2 over time. (a) λ_2 with our connectivity preservation algorithm. (b) λ_2 with conservative connectivity preservation algorithm.

VII. Conclusion

The problem of flocking control with flexible connectivity preservation for multiagent systems with second-order dynamics has been investigated. A fully decentralized flocking control strategy is presented via integrating a decentralized estimator of the algebraic connectivity into a distributed gradient-based flocking control protocol. The proposed estimation-based control protocol could steer the multiagent systems to achieve stable connectivity-preserving flocking motion asymptotically by flexibly keeping λ_2 larger than a certain given threshold. While there are still some issues that need to be addressed in the future. It is interesting and challenging to consider more realistic communication models especially failures of communication links and directed communication graphs. The upper bound of the radius of obstacles also deserves further investigation.

References

[1] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proc. IEEE*, vol. 99, no. 9, pp. 1525–1540, Sep. 2011.

[2] S. Zhu, C. Chen, W. Li, B. Yang, and X. Guan, "Distributed optimal consensus filter for target tracking in heterogeneous sensor networks," *IEEE Trans. Cyber.*, vol. 43, no. 6, pp. 1963–1976, Dec. 2013.

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

[4] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1226–1229, Aug. 1995.

[5] Y. Tang, H. Gao, W. Zou, and J. Kurths, "Distributed synchronization in networks of agent systems with nonlinearities and random switchings," *IEEE Trans. Cyber.*, vol. 43, no. 1, pp. 358–370, Feb. 2013.

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

[7] 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.

[8] J. Chen, M. Gan, J. Huang, L. Dou, and H. Fang, "Formation control of multiple Euler–Lagrange systems via null-space-based behavioral control," *Sci. China Inf. Sci.*, vol. 59, no. 1, pp. 1–11, Jan. 2016.

[9] Q. Wang and Y. Wang, "Cluster synchronization of a class of multi-agent systems with a bipartite graph topology," *Sci. China Inf. Sci.*, vol. 57, no. 1, pp. 1–11, Jan. 2014.

[10] S. H. Semnani and O. A. Basir, "Semi-flocking algorithm for motion control of mobile sensors in large-scale surveillance systems," *IEEE Trans. Cyber.*, vol. 45, no. 1, pp. 129–137, Jan. 2015.

[11] X. Wang, S. Li, and P. Shi, "Distributed finite-time containment control for double-integrator multiagent systems," *IEEE Trans. Cyber.*, vol. 44, no. 9, pp. 1518–1528, Sep. 2014.

[12] 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.

[13] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 116–120, Jan. 2006.

[14] M. M. Zavlanos and G. J. Pappas, "Potential fields for maintaining connectivity of mobile networks," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 812–816, Aug. 2007.

[15] Z. Chen, M.-C. Fan, and H.-T. Zhang, "How much control is enough for network connectivity preservation and collision avoidance?" *IEEE Trans. Cyber.*, vol. 45, no. 8, pp. 1647–1656, Aug. 2015.

[16] M. Ji and M. Egerstedt, "Distributed formation control while preserving connectedness," in *Proc. IEEE Conf. Decis. Control*, San Diego, CA, USA, Dec. 2006, pp. 5962–5967.

[17] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. IEEE Conf. Decis. Control*, San Diego, CA, USA, Dec. 2006, pp. 3628–3633.

[18] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Trans. Robot.*, vol. 23, no. 4, pp. 693–703, Aug. 2007.

[19] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, "Flocking while preserving network connectivity," in *Proc. IEEE Conf. Decis. Control*, New Orleans, LA, USA, Dec. 2007, pp. 2919–2924.

[20] D. V. Dimarogonas and K. J. Kyriakopoulos, "Connectedness preserving distributed swarm aggregation for multiple kinematic robots," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1213–1223, Oct. 2008.

[21] H. Su, X. Wang, and G. Chen, "A connectivity-preserving flocking algorithm for multi-agent systems based only on position measurements," *Int. J. Control*, vol. 82, no. 7, pp. 1334–1343, May 2009.

[22] X. Li and Y. Xi, "Distributed cooperative coverage and connectivity maintenance for mobile sensing devices," in *Proc. IEEE Conf. Decis. Control*, Shanghai, China, Dec. 2009, pp. 7891–7896.

[23] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid control for connectivity preserving flocking," *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2869–2875, Dec. 2009.

[24] M. M. Zavlanos and G. J. Pappas, "Distributed connectivity control of mobile networks," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2008.

[25] D. V. Dimarogonas and K. H. Johansson, "Bounded control of network connectivity in multi-agent systems," *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1330–1338, Aug. 2010.

[26] H. Su, X. Wang, and G. Chen, "Rendezvous of multiple mobile agents with preserved network connectivity," *Syst. Control Lett.*, vol. 59, no. 5, pp. 313–322, May 2010.

[27] A. Ajorlou, A. Momeni, and A. G. Aghdam, "A class of bounded distributed control strategies of connectivity preservation in multi-agent systems," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2828–2832, Dec. 2010.

[28] Z. Kan, A. P. Dani, J. M. Shea, and W. E. Dixon, "Network connectivity preserving formation stabilization and obstacle avoidance via a decentralized controller," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1827–1832, Jul. 2012.

[29] L. Wang, X. Wang, and X. Hu, "Connectivity preserving flocking without velocity measurement," *Asian J. Control*, vol. 15, no. 2, pp. 521–532, Mar. 2013.

[30] M. Schuresko and J. Cortés, "Distributed motion constraints for algebraic connectivity of robotic networks," *J. Intell. Robot. Syst.*, vol. 56, nos. 1–2, pp. 99–126, Apr. 2009.

[31] R. Aragues, G. Shi, D. V. Dimarogonas, C. Sagues, and K. H. Johansson, "Distributed algebraic connectivity estimation for adaptive event-triggered consensus," in *Proc. Amer. Control Conf. (ACC)*, Montreal, QC, Canada, Jun. 2012, pp. 32–37.

[32] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity management in mobile robot teams," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, May 2008, pp. 1525–1530.

[33] P. Yang *et al.*, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, Feb. 2010.

[34] L. Sabattini, C. Secchi, and N. Chopra, "Decentralized connectivity maintenance for networked Lagrangian dynamical systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, USA, May 2012, pp. 2433–2438.

[35] Y. Mao, L. Dou, H. Fang, J. Chen, and T. Cai, "Distributed flocking of second-order multi-agent systems with global connectivity maintenance," in *Proc. Amer. Control Conf. (ACC)*, Washington, DC, USA, Jun. 2013, pp. 976–981.

[36] M. C. De Gennaro and A. Jadbabaie, "Decentralized control of connectivity for multi-agent systems," in *Proc. IEEE Conf. Decis. Control*, San Diego, CA, USA, Dec. 2006, pp. 3628–3633.

[37] D. E. Marelli and M. Y. Fu, "Distributed weighted least-squares estimation with fast convergence for large-scale systems," *Automatica*, vol. 51, pp. 27–39, Jan. 2015.

[38] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. IEEE Conf. Decis. Control*, San Diego, CA, USA, Dec. 2006, pp. 338–343.

[39] C. D. Godsil and G. F. Royle, *Algebraic Graph Theory*. New York, NY, USA: Springer Science+Business Media, 2001.

[40] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Washington, DC, USA: Johns Hopkins Univ. Press, 1996.

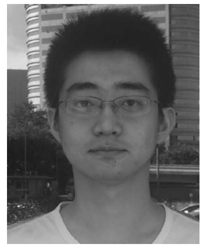
[41] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.

[42] H. J. Sussmann and P. V. Kokotovic, "The peaking phenomenon and the global stabilization of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 36, no. 4, pp. 424–440, Apr. 1991.

[43] S. Wasserman and K. Faust, *Social Networks Analysis: Methods and Application*. Cambridge, U.K.: Cambridge Univ. Press, 1994.

Hao Fang (M'15) received the B.S. degree from the Xi'an University of Technology, Xi'an, China, in 1995, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, in 1998 and 2002, respectively.

He held two Post-Doctoral appointments at the INRIA/France Research Group of COPRIN and the LASMEA (UNR6602 CNRS/Blaise Pascal University, Clermont-Ferrand, France). Since 2011, he has been a Professor with the Beijing Institute of Technology, Beijing, China. His current research interests include all-terrain mobile robots, robotic control, and parallel manipulators.



Yue Wei received the bachelor's degree from the Beijing Institute of Technology Zhuhai Campus, Zhuhai, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Automation, Beijing Institute of Technology, Beijing, China.

His current research interests include parameter estimation, multiagent systems, and human-in-loop research.

Jie Chen (M'09–SM'12) received the B.S. degree, the M.S. and Ph.D. degrees from Beijing Institute of Technology, Beijing, China, in 1986, 1996, and 2001, respectively.

He is a Professor with the School of Automation, Beijing Institute of Technology, Beijing, China. His current research interests include complex system, multi-agent system, multiobjective optimization and decision, constrained nonlinear control, and optimization methods.

Bin Xin (S'09–M'10) received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively.

He is currently an associate professor with the School of Automation, Beijing Institute of Technology. His current research interests include search and optimization, multi-agent system, evolutionary computation, operations research, and combinatorial optimization.