On the control of the algebraic connectivity and clustering of a mobile robotic network

Fabio Morbidi

***Abstract*— In this paper two related problems are studied: the control of the algebraic connectivity and clustering of a network of single-integrator agents. A steepest-descent algorithm is presented for the first problem, so that a smooth approximation of the algebraic connectivity of the underlying undirected communication graph converges to an *assigned value*. For the second problem, a new gradient-based control strategy is proposed to *automatically partition* the mobile robotic network into two predefined groups: our spectral-clustering method leverages a continuous-time power-iteration algorithm on the normalized Laplacian matrix which provides an estimate of its Fiedler vector at each time instant. The results of numerical simulations are provided to illustrate our theoretical findings.**

I. Introduction

A. Motivation and related work

This paper presents original algorithms for controlling the *algebraic connectivity* and *clustering* of a mobile robotic network. Connectivity is a critical issue in numerous problems involving groups of cooperating agents. Since connectivity is *not* generally maintained during the execution of a given coordination task (e.g., rendezvous at a point), several methods have emerged in the recent literature to preserve it at all times. The majority of the existing approaches rely on *global connectivity criteria*, such as, notably, the "algebraic connectivity" or "Fiedler value" of the underlying communication graph (i.e., the second smallest eigenvalue of the graph Laplacian). In the seminal work by De Gennaro and Jadbabaie [1], an iterative decentralized supergradient algorithm is presented for maximizing the connectivity of a robotic network. In [2], instead, the authors leveraged the determinant of the reduced Laplacian to define an artificial potential field that drives the agents to configurations away from the undesired space of disconnected networks. In [3], the same authors used the notion of k-connectivity and tools from hybrid-systems theory to address an analogous problem, while recently, in [4], a new method that relies on the so-called spectral moments has been proposed to achieve a desired set of adjacency eigenvalues. In [5], the Fiedler-value maximization problem was cast as a semi-definite program (SDP) and in [6] it has been made distributedly implementable via a non-iterative method that solves local SDPs using only the information from the nearest neighbors. In [7], distributed game-theoretic algorithms that work under imperfect information caused by delays in communication and robots' mobility, have been proposed for connectivity maintenance. Finally, in [8], an original estimation procedure based on dynamic average consensus estimators has been introduced for tracking the algebraic connectivity and a gradient-based control strategy has been designed to maintain the connectivity at all times. The focus of a second stream of research, has been on the synthesis of (decentralized) controllers that enable each agent to maintain *local connectivity*. For discrete-time double integrators, a feasible control space is computed in [9] for each agent to maintain all existing pairwise connections: in [10], instead, each agent tries to maintain its two-hop communication neighbors. Finally, in [11], the authors studied the rendezvous and the formation control problems over dynamic graphs, and by adding appropriate weights to their edges they guaranteed that connectivity is preserved during robots' motion.

Differently from Fiedler-value maximization which tends to increase the cohesion of the agents, *clustering control* consists of partitioning a network into two (or more) predefined groups of nodes. The intuition behind clustering is that of separating nodes in different groups according to their "similarities": in other words, we seek a partition of the graph such that the edges between different groups have a very low weight (which means that nodes in different clusters are "dissimilar" from each other) and the edges within a group have high weight (i.e. nodes within the same cluster are "similar" to each other). *Spectral clustering* has lately emerged as a powerful tool for network decomposition purposes [12], [13]. In its most general form, this method assigns nodes to clusters according to the signs of the components of the eigenvectors of the (normalized) Laplacian corresponding to increasing eigenvalues. In [14], the authors have developed a distributed algorithm for spectral clustering of *static networks*: the algorithm involves performing random walks and neglecting at every step probabilities below a threshold value. More recently, in [15], a distributed wave equation-based algorithm that is orders of magnitude faster than that in [14], has been proposed for the same problem. Indeed, as pointed out in [16], [17], the clustering of *dynamic networks* could be highly beneficial in many mobile-robot applications, e.g., to maxime a mission's probability of success or economize limited energy resources.

B. Original contributions and organization

This paper is divided into two parts. In the first part, adopting a formulation similar to that in [2], we propose a steepest-descent algorithm to steer the algebraic connectivity of a network of single-integrator agents towards a predefined value. By using an exponential-decay model to characterize the connectivity relation between the robots, our main contribution consists in approximating the algebraic connectivity (that is *not* a differentiable function, in general: in fact the eigenvalues are not differentiable quantities at points where they coalesce [18]), with a smooth symmetric penalty function. Note that although some previous papers in the connectivity-control literature have acknowledged the nonsmooth nature of the Fiedler value (see, e.g., [19], [20]),

The author was with the Institute for Design and Control of Mechatronical Systems, Johannes Kepler University, Altenbergerstraße 69, 4040 Linz, Austria. He is currently with the Networked Controlled System (NeCS) team, Inria Grenoble Rhône-Alpes, 655 Avenue de l'Europe, Montbonnot, 38334 Saint Ismier, France. *Email:* fabio.morbidi@inria.fr

to the best of the author's knowledge *no explicit solution* to this problem has been yet proposed. It is also worth pointing out that differently from most of the existing works which merely deal with the maximization of the Fiedler value (that leads to the collapse of the agents into a single point, unless e.g., a leader-follower mechanism is adopted [8]), for the first time in this article we synthesize a more flexible *set-point controller* of the algebraic connectivity.

In the second part of the paper, we continue to adopt an exponential-decay model for describing agents' interaction and inspired by [14], [15] where the spectral clustering of a graph with *static nodes* is considered, we propose a new gradient-based control strategy to *automatically* partition a *mobile robotic network* into two predefined clusters. Our approach relies on a continuous-time power iteration algorithm on the normalized Laplacian matrix which provides an estimate of the Fiedler vector (i.e. the eigenvector associated to the Fiedler value) at each time instant, and bears some resemblance with the eigenstructure-assignment techniques routinely used in the field of active-vibration control. Note that in this paper we are not specifically interested in *decentralized* control strategies, but rather in providing preliminary proof-of-concepts. Our algorithms, indeed, can be implemented in a distributed fashion by tailoring existing approaches (e.g., that in [8]) to our specific setting: this extension is the subject of ongoing research.

The rest of this paper is organized as follows. Sect. II presents some preliminaries on agents' modeling and spectral graph theory. The main theoretical results of the work are introduced in Sects. III and IV, which deal with the connectivity and clustering control problems. Finally, in Sect. V, the theory is illustrated via numerical simulations, and in Sect. VI the main contributions of the paper are summarized and possible future research directions are outlined.

II. Preliminaries

Consider a team of $n > 2$ mobile agents modeled as single integrators, $\dot{\mathbf{p}}_i(t) = \mathbf{u}_i(t)$, $i \in \{1, \ldots, n\}$, where $\mathbf{p}_i(t) = [p_{i1}(t), \ldots, p_{id}(t)]^T \in \mathbb{R}^d$, $d \in \{2, 3\}$, denotes the position of agent i and $\mathbf{u}_i(t) = [u_{i1}(t), \ldots, u_{id}(t)]^T \in \mathbb{R}^d$ its control input at time t. The dynamics of the team can be rewritten in a compact form as,

$$\dot{\mathbf{p}}(t) \;=\; \mathbf{u}(t), \tag{1}$$

where $\mathbf{p}(t) \triangleq [\mathbf{p}_1^T(t), \ldots, \mathbf{p}_n^T(t)]^T \in \mathbb{R}^{dn}$ and $\mathbf{u}(t) \triangleq [\mathbf{u}_1^T(t), \ldots, \mathbf{u}_n^T(t)]^T \in \mathbb{R}^{dn}$.

Note that the n-agent network defined by (1), gives rise to a graph $\mathcal{G}(\mathbf{p}(t)) = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of nodes or vertices indexed by the mobile agents and $E = \{\{i, j\} \mid j \in \mathcal{N}(i)\}$ is the set of edges, where $\mathcal{N}(i)$ denotes the set of neighbors of agent i in the undirected communication network.

***Definition** 1 (Weighted adjacency matrix $\mathbf{A}$):* The $n \times n$ weighted adjacency matrix $\mathbf{A} = [a_{ih}]$ is defined as,

$$a_{ih} \;=\; \begin{cases} a_{hi} > 0 & \text{if } \; h \in \mathcal{N}(i), \\ 0 & \text{otherwise.} \end{cases}$$

Since self-loops are not allowed, we define $a_{ii} = 0$, $\forall\, i \in \{1, 2, \ldots, n\}$. ◇

***Definition** 2 (Weighted Laplacian matrix $\mathbf{L}$):* The weighted Laplacian matrix is defined as,

$$\mathbf{L} \;=\; \mathbf{D} - \mathbf{A},$$

where the diagonal matrix $\mathbf{D} = \text{diag}(\mathbf{A}\mathbb{1})$ is called the degree matrix, and $\mathbb{1}$ is a column vector of n ones. ◇

***Property** 1 (Spectral properties of $\mathbf{L}$):* Let $\lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}) \leq \ldots \leq \lambda_n(\mathbf{L})$ be the ordered eigenvalues of the Laplacian $\mathbf{L}$. Then, we have that:

1) $\lambda_1(\mathbf{L}) = 0$ with corresponding eigenvector $\mathbb{1}$. The algebraic multiplicity of $\lambda_1(\mathbf{L})$ is equal to the number of connected components in the graph $\mathcal{G}$.
2) $\lambda_2(\mathbf{L}) > 0$ if and only if the graph $\mathcal{G}$ is connected. λ_2 is called the *algebraic connectivity* or *Fiedler value* of $\mathcal{G}$ and the associated eigenvector is called *characteristic valuation* or *Fiedler vector* of $\mathcal{G}$. ◇

***Definition** 3 (Reduced Laplacian $\mathbf{L}^*$):* Let

$$\text{Orth}(\mathbb{1}) \triangleq \{\mathbf{P} \in \mathbb{R}^{n \times (n-1)} \mid \mathbf{P}^T\mathbf{P} = \mathbf{I}_{n-1} \text{ and } \mathbf{P}^T\mathbb{1} = \mathbf{0}\},$$

where $\mathbf{I}_{n-1}$ is the $(n-1) \times (n-1)$ identity matrix. By fixing some $\mathbf{P} \in \text{Orth}(\mathbb{1})$, we define the reduced Laplacian to be the $(n-1) \times (n-1)$ symmetric matrix:

$$\mathbf{L}^* \;=\; \mathbf{P}^T\,\mathbf{L}\,\mathbf{P}.$$ ◇

***Property** 2 (Spectrum of $\mathbf{L}^*$):* Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the ordered eigenvalues of $\mathbf{L}$. Then $\lambda_2 \leq \ldots \leq \lambda_n$ are the eigenvalues of $\mathbf{L}^*$. ◇

There exist two matrices which are called *normalized Laplacian* in the literature [21]. This paper deals with the normalized Laplacian $\mathbf{L}_{\text{sym}}$, which is a symmetric matrix.

***Definition** 4 (Normalized Laplacian $\mathbf{L}_{sym}$):* The normalized Laplacian $\mathbf{L}_{\text{sym}} \in \mathbb{R}^{n \times n}$ is defined as,

$$\mathbf{L}_{\text{sym}} \;\triangleq\; \mathbf{D}^{-1/2}\,\mathbf{L}\,\mathbf{D}^{-1/2} \;=\; \mathbf{I}_n - \mathbf{D}^{-1/2}\,\mathbf{A}\,\mathbf{D}^{-1/2}.$$ ◇

***Property** 3 (Spectral properties of $\mathbf{L}_{sym}$ [21]):* $\mathbf{L}_{\text{sym}}$ has the following spectral properties:

1) 0 is an eigenvalue of $\mathbf{L}_{\text{sym}}$ with eigenvector $\mathbf{D}^{1/2}\,\mathbb{1}$. The algebraic multiplicity of the eigenvalue 0 is equal to the number of connected components in the graph.
2) $\mathbf{L}_{\text{sym}}$ is a symmetric positive semidefinite matrix and it has n nonnegative real-valued eigenvalues $0 = \lambda_1(\mathbf{L}_{\text{sym}}) \leq \ldots \leq \lambda_n(\mathbf{L}_{\text{sym}}) \leq 2$. If $\mathcal{G}$ is a bipartite graph, then $\lambda_n(\mathbf{L}_{\text{sym}}) = 2$.
3) $\text{tr}(\mathbf{L}_{\text{sym}}) = n$ for any graph $\mathcal{G}$. ◇

***Definition** 5 (Spectral clustering of graph $\mathcal{G}$):* The signs of the n components of the Fiedler vector $\mathbf{v}_2$ of $\mathbf{L}_{\text{sym}}$, define a partition of the connected graph $\mathcal{G}$ into two *vertex-cutsets*[1]. For example, if $\mathbf{v}_2 = [0.35,\, 0.7,\, 0.06,\, -0.2,\, 0.12,\, -0.7]^T$, then the graph $\mathcal{G}$ is partitioned into the vertex-cutsets $\{1,\, 2,\, 3,\, 5\}$ and $\{4,\, 6\}$, according to the sign pattern. ◇

Through all the paper, we will assume that the components a_{ih}, $i \neq h$, of the weighted adjacency matrix $\mathbf{A}$ are defined according to the following *position-dependent Gaussian similarity function*,

$$a_{ih}(\mathbf{p}(t)) \;\triangleq\; \exp\!\left(-\frac{\|\mathbf{p}_i(t) - \mathbf{p}_h(t)\|^2}{2\,\sigma^2}\right),$$

where $\|\cdot\|$ denotes the standard Euclidean norm and parameter $\sigma > 0$ controls the width of the neighborhoods[2]. In this way, by suitably moving the n agents, we are ultimately able to control the algebraic connectivity and clustering of the robotic network.

[1] As pointed out in [12, Sect. 8.5], the normalized Laplacian $\mathbf{L}_{\text{sym}}$ offers distinctive advantages over the standard Laplacian $\mathbf{L}$ for spectral clustering.

[2] For the sake of simplicity, as in [2], we do *not* consider here a threshold value on $\|\mathbf{p}_i(t) - \mathbf{p}_h(t)\|$ over which an edge ceases to exist.

III. Set-point connectivity control

The first problem studied in this paper is that of connectivity control.

***Problem** 1 (Set-point connectivity control):* Let the initial configuration $\mathbf{p}(t_0)$ of the n agents be arbitrary. Design the control inputs $\mathbf{u}_1, \ldots, \mathbf{u}_n$ of the agents, so that the quadratic cost function,

$$J_1 = \gamma\,(\lambda_2(\mathbf{L}) \,-\, \lambda_2^d)^2, \tag{2}$$

is minimized, where γ is a positive gain and $\lambda_2^d > 0$ represents the *desired algebraic connectivity* of the network. $\diamond$

Note that the cost function (2) can be equivalently rewritten as (cf. Property 2),

$$J_1 = \gamma\,\big(\lambda_{\min}(\mathbf{L}^*) \,-\, \lambda_2^d\big)^2 = \gamma\,\big(\lambda_{\max}(-\mathbf{L}^*) \,+\, \lambda_2^d\big)^2,$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalue of a matrix, respectively.

Since the cost function $\lambda_{\max}(-\mathbf{L}^*)$ is *nondifferentiable* in general (in fact, the gradient of $\lambda_{\max}(-\mathbf{L}^*)$ does not exist when the maximum eigenvalue of $-\mathbf{L}^*$ is not simple, i.e., it has algebraic multiplicity greater than one), in the remaining of this section we will use its *smoothed version* (a symmetric exponential penalty function [22, p. 293])

$$\Phi_\varepsilon(-\mathbf{L}^*) \;\triangleq\; \varepsilon \ln\Big(\sum_{k=1}^{n-1} \exp\Big(-\frac{\lambda_k(\mathbf{L}^*)}{\varepsilon}\Big)\Big), \tag{3}$$

where $\varepsilon > 0$ is a smoothing parameter and $\lambda_k(\mathbf{L}^*)$ denotes the k-th eigenvalue of $\mathbf{L}^*$. Note that (3) is a $\mathcal{C}^\infty$ convex function with respect to $-\mathbf{L}^*$ and it possesses the following uniform approximation property to $\lambda_{\max}(-\mathbf{L}^*)$,

$$0 \;\le\; \Phi_\varepsilon(-\mathbf{L}^*) - \lambda_{\max}(-\mathbf{L}^*) \;\le\; \varepsilon \ln(n-1), \;\; \forall\, \varepsilon > 0,$$

from which it follows that $\lim_{\varepsilon \downarrow 0} \Phi_\varepsilon(-\mathbf{L}^*) = \lambda_{\max}(-\mathbf{L}^*)$.

The next proposition provides a solution to Problem 1. Our idea simply consists in considering

$$J_1 = \gamma\,\big(\lambda_{\max}(-\mathbf{L}^*) \,+\, \lambda_2^d\big)^2 = \lim_{\varepsilon \downarrow 0} \gamma\,\big(\Phi_\varepsilon(-\mathbf{L}^*) \,+\, \lambda_2^d\big)^2, \tag{4}$$

as an artificial potential function, and in using it to define a steepest-descent control law for each of the n agents.

***Proposition** 1 (Solution to Problem 1):* Problem 1 is solved if the following control input is applied to agent i for $\varepsilon \downarrow 0$:

$$\begin{aligned}\mathbf{u}_i = -2\,\gamma\,(\Phi_\varepsilon(-\mathbf{L}^*) + \lambda_2^d)\big[&\mathrm{tr}\big(\mathbf{R}\,\mathrm{diag}(\mathbf{\Pi})\,\mathbf{R}^T\,\mathbf{P}^T\,\mathbf{\Theta}_{i1}\,\mathbf{P}\big),\\ &\ldots,\, \mathrm{tr}\big(\mathbf{R}\,\mathrm{diag}(\mathbf{\Pi})\,\mathbf{R}^T\,\mathbf{P}^T\,\mathbf{\Theta}_{id}\,\mathbf{P}\big)\big]^T,\end{aligned} \tag{5}$$

where $\mathbf{R}$, $\mathrm{diag}(\mathbf{\Pi})$, $\mathbf{P}$, $\mathbf{\Theta}_{i1}$, $\ldots$, $\mathbf{\Theta}_{id}$ are matrices whose explicit expression is given in the proof below.

Proof: Consider J_1 in (4) as a potential function and define the control of agent i as follows:

$$\begin{aligned}\mathbf{u}_i = -\nabla_{\mathbf{p}_i} J_1 &= \lim_{\varepsilon \downarrow 0} -\nabla_{\mathbf{p}_i} \gamma\big(\Phi_\varepsilon(-\mathbf{L}^*) + \lambda_2^d\big)^2\\ &= \lim_{\varepsilon \downarrow 0} -2\gamma\big(\Phi_\varepsilon(-\mathbf{L}^*) + \lambda_2^d\big)\nabla_{\mathbf{p}_i}\Phi_\varepsilon(-\mathbf{L}^*),\end{aligned} \tag{6}$$

where $\nabla_{\mathbf{p}_i} J_1$ denotes the gradient of J_1 with respect to $\mathbf{p}_i$. By using [22, Th. B.17], we can rewrite the d-dimensional gradient vector $\nabla_{\mathbf{p}_i}\,\Phi_\varepsilon(-\mathbf{L}^*)$ in (6), as,

$$\begin{aligned}\nabla_{\mathbf{p}_i}\,\Phi_\varepsilon(-\mathbf{L}^*) = \Bigg[&\mathrm{tr}\bigg(\bigg(\frac{\partial\,\Phi_\varepsilon(-\mathbf{L}^*)}{\partial\,\mathbf{L}^*}\bigg)^T \frac{\partial\,\mathbf{L}^*}{\partial\,p_{i1}}\bigg), \ldots,\\ &\mathrm{tr}\bigg(\bigg(\frac{\partial\,\Phi_\varepsilon(-\mathbf{L}^*)}{\partial\,\mathbf{L}^*}\bigg)^T \frac{\partial\,\mathbf{L}^*}{\partial\,p_{id}}\bigg)\Bigg]^T.\end{aligned} \tag{7}$$

Let now $\mathcal{O}(n-1)$ denote the group of $(n-1)\times(n-1)$ real orthogonal matrices. Then, by leveraging the results in [23], we have that,

$$\frac{\partial\,\Phi_\varepsilon(-\mathbf{L}^*)}{\partial\,\mathbf{L}^*} \;=\; \mathbf{R}\;\mathrm{diag}\big(\mathbf{\Pi}(-\mathbf{L}^*,\varepsilon)\big)\,\mathbf{R}^T, \tag{8}$$

where $\mathbf{R} \in \mathcal{O}(n-1)$ is such that $\mathbf{R}^T\,\mathbf{L}^*\,\mathbf{R} = \mathrm{diag}(\boldsymbol{\lambda}(\mathbf{L}^*))$, being $\boldsymbol{\lambda}(\mathbf{L}^*) \in \mathbb{R}^{n-1}$ the vector of eigenvalues of $\mathbf{L}^*$ in nondecreasing order. Moreover, $\mathbf{\Pi}(-\mathbf{L}^*,\varepsilon) = [\Pi_1(-\mathbf{L}^*,\varepsilon),\ldots,\Pi_{n-1}(-\mathbf{L}^*,\varepsilon)]^T$, where

$$\Pi_k(-\mathbf{L}^*,\varepsilon) \;=\; \frac{e^{\,-\lambda_k(\mathbf{L}^*)/\varepsilon}}{\sum_{\ell=1}^{n-1} e^{\,-\lambda_\ell(\mathbf{L}^*)/\varepsilon}},\; k \in \{1,\ldots,n-1\}.$$

Note that $\Pi_k(-\mathbf{L}^*,\varepsilon) > 0$, $k \in \{1,\ldots,n-1\}$, and that $\sum_{k=1}^{n-1}\Pi_k(-\mathbf{L}^*,\varepsilon) = 1$. Finally, observe that,

$$\frac{\partial\,\mathbf{L}^*}{\partial\,p_{ij}} \;=\; \mathbf{P}^T\frac{\partial\,\mathbf{L}}{\partial\,p_{ij}}\,\mathbf{P} \;\triangleq\; \mathbf{P}^T\,\mathbf{\Theta}_{ij}\,\mathbf{P},\;\; j \in \{1,\ldots,d\}, \tag{9}$$

where $\mathbf{\Theta}_{ij} \in \mathbb{R}^{n\times n}$ is given in equation (10) at the top of the next page. By collecting equations (6)-(10) together, we obtain the control input in (5). ■

IV. Clustering control

The second problem studied in this paper is formulated as follows (recall Def. 5).

***Problem** 2 (Clustering control):* Let the initial configuration $\mathbf{p}(t_0)$ of the n agents be arbitrary. Design the control inputs $\mathbf{u}_1, \ldots, \mathbf{u}_n$ of the agents, so that the following quadratic cost function,

$$J_2 \;=\; (\mathbf{v}_2 - \mathbf{v}_2^d)^T\,\mathbf{\Gamma}\,(\mathbf{v}_2 - \mathbf{v}_2^d)\,, \tag{11}$$

is minimized, where $\mathbf{\Gamma} \in \mathbb{R}^{n\times n}$ is a symmetric positive-definite gain matrix, $\mathbf{v}_2 \in \mathbb{R}^n$ is the Fiedler vector of the normalized Laplacian $\mathbf{L}_{\text{sym}}$, and $\mathbf{v}_2^d \in \ker(\lambda_2(\mathbf{L}_{\text{sym}})\mathbf{I}_n - \mathbf{L}_{\text{sym}})$ is an n-dimensional vector that specifies the *desired clustering* of the network. $\diamond$

The next proposition, an extension of Theorem 1 in [8], is instrumental in presenting the main result of this section (Prop. 3). It introduces a *continuous-time power-iteration algorithm* for estimating the Fiedler vector $\mathbf{v}_2$ of $\mathbf{L}_{\text{sym}}$ at each time instant. For the sake of conciseness, the proof is omitted.

***Proposition** 2 (Continuous-time power iteration on $\mathbf{L}_{sym}$):* Let $\boldsymbol{\delta} \in \mathbb{R}^n$ be such that $\mathrm{diag}(\boldsymbol{\delta}) = \mathbf{D}^{1/2}$. Given any initial condition $\mathbf{x}(t_0)$ and positive gains k_1, k_2 and k_3, as long as $\mathbf{v}_2^T\mathbf{x}(t_0) \neq 0$, the conditions,

$$k_1 > k_2\,\frac{n}{\|\boldsymbol{\delta}\|^2}\;\lambda_2(\mathbf{L}_{\text{sym}}), \qquad k_3 > k_2\,\lambda_2(\mathbf{L}_{\text{sym}}), \tag{12}$$

are necessary and sufficient for the following system,

$$\dot{\mathbf{x}} \;=\; -\left(\frac{k_1}{n}\,\boldsymbol{\delta}\,\boldsymbol{\delta}^T \,+\, k_2\,\mathbf{L}_{\text{sym}} \,+\, k_3\left[\frac{\|\mathbf{x}\|^2}{n} - 1\right]\mathbf{I}_n\right)\mathbf{x}\,, \tag{13}$$

$$\boldsymbol{\Theta}_{ij} = \frac{1}{\sigma^2}\begin{bmatrix} -(p_{1j}-p_{ij})\,e^{-\frac{\|\mathbf{p}_1-\mathbf{p}_i\|^2}{2\sigma^2}} & \cdots & 0 & \cdots & (p_{1j}-p_{ij})\,e^{-\frac{\|\mathbf{p}_1-\mathbf{p}_i\|^2}{2\sigma^2}} & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & (p_{2j}-p_{ij})\,e^{-\frac{\|\mathbf{p}_2-\mathbf{p}_i\|^2}{2\sigma^2}} & \cdots & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & \cdots & \cdots & 0 & \cdots & 0 \\ (p_{ij}-p_{1j})\,e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_1\|^2}{2\sigma^2}} & & \cdots & & \sum_{\substack{k=1\\k\neq i}}^{n}(p_{ij}-p_{kj})\,e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_k\|^2}{2\sigma^2}} & & \cdots & & (p_{ij}-p_{nj})\,e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_n\|^2}{2\sigma^2}} \\ 0 & \cdots & 0 & \cdots & \cdots & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & \cdots & (p_{nj}-p_{ij})\,e^{-\frac{\|\mathbf{p}_n-\mathbf{p}_i\|^2}{2\sigma^2}} & \cdots & 0 & \cdots & -(p_{nj}-p_{ij})\,e^{-\frac{\|\mathbf{p}_n-\mathbf{p}_i\|^2}{2\sigma^2}} \end{bmatrix}. \tag{10}$$

to converge to an eigenvector $\widetilde{\mathbf{v}}_2$ corresponding to $\lambda_2(\mathbf{L}_{\text{sym}})$, satisfying $\|\widetilde{\mathbf{v}}_2\| = \sqrt{n\left(1-\frac{k_2}{k_3}\,\lambda_2(\mathbf{L}_{\text{sym}})\right)}$. ■

Note that in order to accommodate the continuous-time power iteration on $\mathbf{L}$ in [8, Th. 1] to the normalized Laplacian $\mathbf{L}_{\text{sym}}$, we had to modify the so-called "*deflation step*" (cf. the first term on the right-hand side of (13)), as follows:

$$\dot{\mathbf{x}} \;=\; -\,\frac{k_1}{n}\;\mathbf{D}^{1/2}\,\mathbb{1}\mathbb{1}^T(\mathbf{D}^{1/2})^T\,\mathbf{x} \;=\; -\,\frac{k_1}{n}\,\boldsymbol{\delta}\,\boldsymbol{\delta}^T\mathbf{x}. \tag{14}$$

From a geometric viewpoint, system (14) drives $\mathbf{x}$ to $\ker(\mathbf{D}^{1/2}\,\mathbb{1})$, i.e., the space spanned by the eigenvectors $\mathbf{v}_2,\,\ldots,\mathbf{v}_n$ of $\mathbf{L}_{\text{sym}}$. In the case of repeated eigenvalues $\lambda_2(\mathbf{L}_{\text{sym}}) = \ldots = \lambda_r(\mathbf{L}_{\text{sym}}) < \lambda_{r+1}(\mathbf{L}_{\text{sym}})$, Prop. 2 is still valid. In this case, as shown in [8], all trajectories with $\mathbf{v}_2^T\mathbf{x}(t_0) \neq 0$ converge to an equilibrium point on the r-dimensional manifold,

$$\mathcal{M} = \Big\{\mathbf{q} = [q_1,\,\ldots,q_n]^T \;\Big|\; \|\mathbf{q}\| = \sqrt{n\,(1-\tfrac{k_2}{k_3}\,\lambda_2(\mathbf{L}_{\text{sym}}))}\,,\; q_1 = 0,\;\; q_i = 0,\;\; \forall\; i > r\Big\}.$$

However, for the existence of *unique cuts* that divide the network into two clusters, we henceforth assume that $\lambda_2(\mathbf{L}_{\text{sym}}) \neq \lambda_3(\mathbf{L}_{\text{sym}})$, (cf. [15]).

***Remark** 1 (On the fulfillment of the conditions in (12)):* Note that the conditions in (12) can be satisfied also with a partial knowledge of the network topology. In fact, we have that $\lambda_2(\mathbf{L}_{\text{sym}}) < \text{tr}(\mathbf{L}_{\text{sym}}) = n$ (recall Property 3.3), from which it follows that the conditions in (12) are fulfilled if we simply choose $k_1 > k_2\,(n/\|\boldsymbol{\delta}\|)^2$, $k_3 > k_2\,n$. ◇

We are now in a position to present our solution to Problem 2. As in Sect. III, we will leverage a gradient-based method to define the control input of the n agents.

***Proposition** 3 (Solution to Problem 2):* Problem 2 is solved if the following control input is applied to agent i,

$$\mathbf{u}_i \;=\; -\,2\left((\mathbf{x}-\mathbf{v}_2^d)^T\,\boldsymbol{\Gamma}\;[\,\mathbf{z}_{i,1},\,\ldots,\,\mathbf{z}_{i,d}]\right)^T, \tag{15}$$

where $\mathbf{x} \in \mathbb{R}^n$ in (15) is obtained from integration of equation (13) at each time instant, and $\mathbf{z}_{i,j} \in \mathbb{R}^n$, $j \in \{1,\ldots,d\}$, is obtained from integration of the following linear time-varying system,

$$\dot{\mathbf{z}}_{i,j} = -\Big(\frac{k_1}{n}\,\boldsymbol{\delta}\,\boldsymbol{\delta}^T + \,k_2\,\mathbf{L}_{\text{sym}} + \,\frac{2\,k_3}{n}\,\mathbf{x}\,\mathbf{x}^T + k_3\Big[\frac{\|\mathbf{x}\|^2}{n}-1\Big]\,\mathbf{I}_n\Big)\,\mathbf{z}_{i,j} - \Big(\frac{2\,k_1}{n}\frac{\partial\,\boldsymbol{\delta}}{\partial\,p_{ij}}\;\boldsymbol{\delta}^T + k_2\,\frac{\partial\,\mathbf{L}_{\text{sym}}}{\partial\,p_{ij}}\Big)\,\mathbf{x}, \tag{16}$$

where

$$\frac{\partial\,\boldsymbol{\delta}}{\partial\,p_{ij}} = \frac{1}{2\,\sigma^2}\Big[\Big(\sum_{k=2}^{n} e^{-\frac{\|\mathbf{p}_1-\mathbf{p}_k\|^2}{2\,\sigma^2}}\Big)^{-1/2}(p_{1j}-p_{ij})\,e^{-\frac{\|\mathbf{p}_1-\mathbf{p}_i\|^2}{2\,\sigma^2}},$$
$$\ldots,\,-\Big(\sum_{\substack{k=1\\k\neq i}}^{n} e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_k\|^2}{2\,\sigma^2}}\Big)^{-1/2}\sum_{\substack{k=1\\k\neq i}}^{n}(p_{ij}-p_{kj})\,e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_k\|^2}{2\,\sigma^2}},$$
$$\ldots,\,\Big(\sum_{k=1}^{n-1} e^{-\frac{\|\mathbf{p}_n-\mathbf{p}_k\|^2}{2\,\sigma^2}}\Big)^{-1/2}(p_{nj}-p_{ij})\,e^{-\frac{\|\mathbf{p}_n-\mathbf{p}_i\|^2}{2\,\sigma^2}}\Big]^T,$$

$$\frac{\partial\,\mathbf{L}_{\text{sym}}}{\partial\,p_{ij}} \;=\; \frac{\partial\,\mathbf{D}^{-1/2}}{\partial\,p_{ij}}\,\mathbf{L}\,\mathbf{D}^{-1/2} + \Big(\frac{\partial\,\mathbf{D}^{-1/2}}{\partial\,p_{ij}}\,\mathbf{L}\,\mathbf{D}^{-1/2}\Big)^T + \;\mathbf{D}^{-1/2}\,\boldsymbol{\Theta}_{ij}\,\mathbf{D}^{-1/2},$$

$\boldsymbol{\Theta}_{ij} \in \mathbb{R}^{n\times n}$ is given in (10) above, and

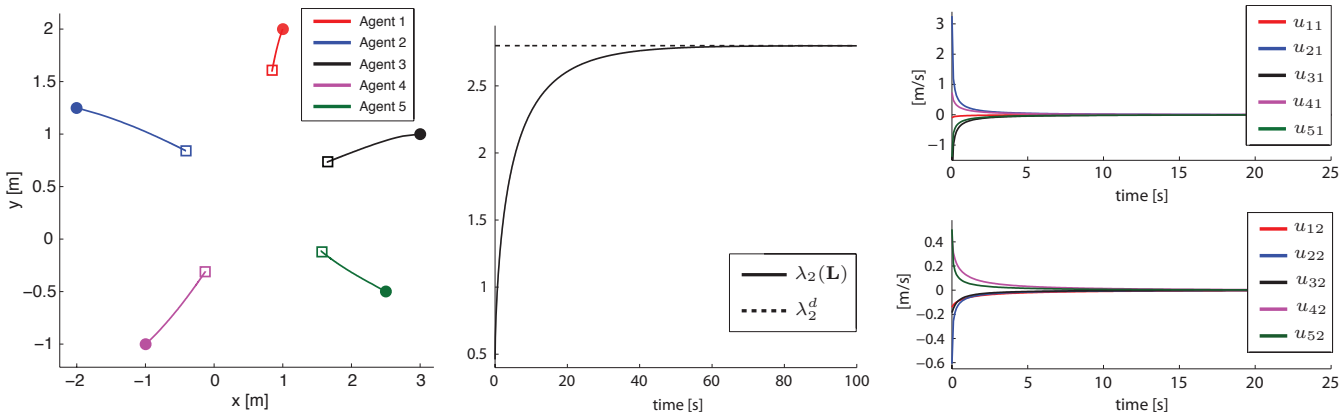
$$\frac{\partial\,\mathbf{D}^{-1/2}}{\partial\,p_{ij}} =$$
$$-\frac{1}{2\,\sigma^2}\,\text{diag}\Big(\Big(\sum_{k=2}^{n} e^{-\frac{\|\mathbf{p}_1-\mathbf{p}_k\|^2}{2\,\sigma^2}}\Big)^{-3/2}(p_{1j}-p_{ij})\,e^{-\frac{\|\mathbf{p}_1-\mathbf{p}_i\|^2}{2\,\sigma^2}},$$
$$\ldots,-\Big(\sum_{\substack{k=1\\k\neq i}}^{n} e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_k\|^2}{2\,\sigma^2}}\Big)^{-3/2}\sum_{\substack{k=1\\k\neq i}}^{n}(p_{ij}-p_{kj})\,e^{-\frac{\|\mathbf{p}_i-\mathbf{p}_k\|^2}{2\,\sigma^2}},$$
$$\ldots,\Big(\sum_{k=1}^{n-1} e^{-\frac{\|\mathbf{p}_n-\mathbf{p}_k\|^2}{2\,\sigma^2}}\Big)^{-3/2}(p_{nj}-p_{ij})\,e^{-\frac{\|\mathbf{p}_n-\mathbf{p}_i\|^2}{2\,\sigma^2}}\Big).$$

Proof: Consider J_2 in (11) as a potential function, and define the control input of agent i as follows:

$$\mathbf{u}_i = -\nabla_{\mathbf{p}_i}\,J_2 = -2\left((\mathbf{v}_2-\mathbf{v}_2^d)^T\,\boldsymbol{\Gamma}\left[\frac{\partial\,\mathbf{v}_2}{\partial\,p_{i1}},\ldots,\frac{\partial\,\mathbf{v}_2}{\partial\,p_{id}}\right]\right)^T.$$

In order to explicitly compute this control, we need to know $\mathbf{v}_2$ and the partial derivatives $\frac{\partial\,\mathbf{v}_2}{\partial\,p_{i1}},\ldots,\frac{\partial\,\mathbf{v}_2}{\partial\,p_{id}}$, at each time instant. In place of $\mathbf{v}_2$, we can use its estimate $\mathbf{x}$ provided by equation (13). Moreover, if we define the vector, $\mathbf{z}_{i,j} \triangleq \frac{\partial\,\mathbf{x}}{\partial\,p_{ij}}$, $j \in \{1,\ldots,d\}$, and compute the partial derivative with respect to p_{ij} of both sides of (13), we obtain the linear time-varying system in (16), which gives us an estimate of $\frac{\partial\,\mathbf{v}_2}{\partial\,p_{i1}},\ldots,\frac{\partial\,\mathbf{v}_2}{\partial\,p_{id}}$ at each time instant. ■

***Remark** 2 (On the computation of multiple clusters):* If the control input in (15) is applied to system (1), the robotic network is partitioned into two desired clusters. Additional clusters can be computed from the sign of the components



(a) (b) (c)

Fig. 1. *Set-point connectivity control*: (a) Trajectory of the 5 agents (the initial and final positions are marked with bullets and squares, respectively); (b) Time history of $\lambda_2(\mathbf{L})$ (solid) and desired algebraic connectivity λ_2^d (dash); (c) Time evolution of the control inputs $\mathbf{u}_i = [u_{i1},\, u_{i2}]^T$, $i \in \{1, \ldots, 5\}$.

of the eigenvectors $\mathbf{v}_3$, $\mathbf{v}_4$, $\ldots$, $\mathbf{v}_k$, $k < n$, associated to the eigenvalues $\lambda_3(\mathbf{L}_{\text{sym}}) \leq \lambda_4(\mathbf{L}_{\text{sym}}) \leq \ldots \leq \lambda_k(\mathbf{L}_{\text{sym}})$ [12]. In this case, at node i one computes the sign of the i-th component of the eigenvectors $\mathbf{v}_2$, $\mathbf{v}_3$, $\ldots$, $\mathbf{v}_k$, and the cluster assignment is obtained by interpreting the vector of k signs as a binary number. The eigenvectors $\mathbf{v}_3$, $\mathbf{v}_4$, $\ldots$, $\mathbf{v}_k$ can be determined, for example, by using the (decentralized) orthogonal iteration algorithm presented in [1], the method described in [15] or by suitably modifying the power iteration (13). Recursive spectral partitioning represents an alternative to "higher-order" eigenvectors. In fact, once the graph has been divided into two clusters, control (15) can be run again independently on both clusters to compute two additional vertex-cutsets. This procedure can be repeated until either a desired number of clusters is found or no further clusters can be determined [15]. $\diamond$

V. Simulation Results

Simulation experiments have been conducted to illustrate the theory presented in the previous two sections.

Fig. 1 shows the performance of our *set-point connectivity controller*. The initial position of the 5 mobile agents is $\mathbf{p}(t_0) = \mathbf{p}(0) = [1,\, 2,\, -2,\, 5/4,\, 3,\, 1,\, -1,\, -1,\, 5/2,\, -1/2]^T$ (marked with bullets in Fig. 1(a)), we chose $\sigma = 1.8$, $\epsilon = 4 \times 10^{-3}$, $\gamma = 10^{-2}$, and selected,

$$\mathbf{P} = \begin{bmatrix} -\sqrt{2/15} & 0 & 0 & -\sqrt{6}/3 \\ \sqrt{3/10} & \sqrt{2}/2 & 0 & 0 \\ -\sqrt{2/15} & 0 & \sqrt{2}/2 & \sqrt{6}/6 \\ -\sqrt{2/15} & 0 & -\sqrt{2}/2 & \sqrt{6}/6 \\ \sqrt{3/10} & -\sqrt{2}/2 & 0 & 0 \end{bmatrix}.$$

Since the initial algebraic connectivity of the network is $\lambda_2(\mathbf{L}) = 0.4610$ and we selected $\lambda_2^d = 2.8$, the relative distance between the robots decreases over time (see Fig. 1(a)). Fig. 1(b) shows the time evolution of $\lambda_2(\mathbf{L})$ (solid) and the desired algebraic connectivity λ_2^d (dash). Finally, Fig. 1(c) reports the time history of the control inputs $\mathbf{u}_i = [u_{i1},\, u_{i2}]^T$, $i \in \{1, \ldots, 5\}$ in the first 25 seconds of the simulation.

Fig. 2 shows the performance of our *clustering controller*. The initial position of the 5 agents is $\mathbf{p}(0) = [1,\, 1.85,\, -1,\, 0.625,\, 4,\, 0.25,\, -0.25,\, -0.1,\, 1.75,\, -0.15]^T$ (marked with bullets in Fig. 2(a)), we set $\sigma = 6.5$, $\Gamma = 0.14\,\mathbf{I}_5$, $k_2 = 0.65$, $k_1 = k_3 = 4.8125$, and we initialized equations (13) and (16) with $\mathbf{x}(0) = [-0.1379,\, 0.6199,\, -0.0056,\, 1.1072,\, -0.1856]^T$ and with random vectors $\mathbf{z}_{i,j}(0)$, $i \in \{1, \ldots, 5\}$, $j \in \{1,\, 2\}$, respectively. Since $\mathbf{v}_2(0) = [-0.0230,\, -0.5468,\, 0.6981,\, -0.3757,\, 0.2683]^T$, the initial partition of the network is $\{1,\, 2,\, 4\}$, $\{3,\, 5\}$. Our choice of $\mathbf{v}_2^d = [-0.05,\, 0.8,\, -1.5,\, 1.12,\, -0.55]^T$, instead, leads to the vertex-cutsets $\{1,\, 3,\, 5\}$ and $\{2,\, 4\}$. Fig. 2(a) shows the trajectory of the 5 agents and the initial and final clusters (indicated with dashed and dotted sets, respectively), while Figs. 2(b) and 2(c) report the time history of the estimated Fiedler vector $\mathbf{x}$ (solid) and $\mathbf{v}_2^d$ (dash), and the time history of the normalized $\mathbf{v}_2$ (solid) and $\mathbf{x}/\|\mathbf{x}\|$ (dash), respectively. Note that after about 75 seconds, the actual and estimated Fiedler vectors coincide. Finally, Figs. 2(d)-(e) show the time evolution of the control input of the 5 agents and of the cost function J_2.

VI. Conclusions and Future Work

In this paper we have presented original algorithms for controlling the algebraic connectivity and clustering of a network of mobile agents. We have defined quadratic potential functions depending on the Fiedler vector and on a smooth approximation of Fiedler value of the Laplacian of the underlying communication graph, and synthesized the controls of the single-integrator agents using gradient-descent methods.

There are several interesting problems that this paper has not addressed and that will be studied in future works. These include the determination of convex approximations to the cost functions J_1 and J_2 with respect to the position of the agents, the extension of the strategy in Sect. IV for the generation of multiple clusters, and the validation of our control policies in the presence of imperfect information. We also aim at gaining more insight on how to translate an assigned clustering of the network into a desired Fiedler vector $\mathbf{v}_2^d$ (and vice versa), and we are examining the possibility to partition the robotic network according to time-varying external events (e.g., the position of two or more evasive targets).

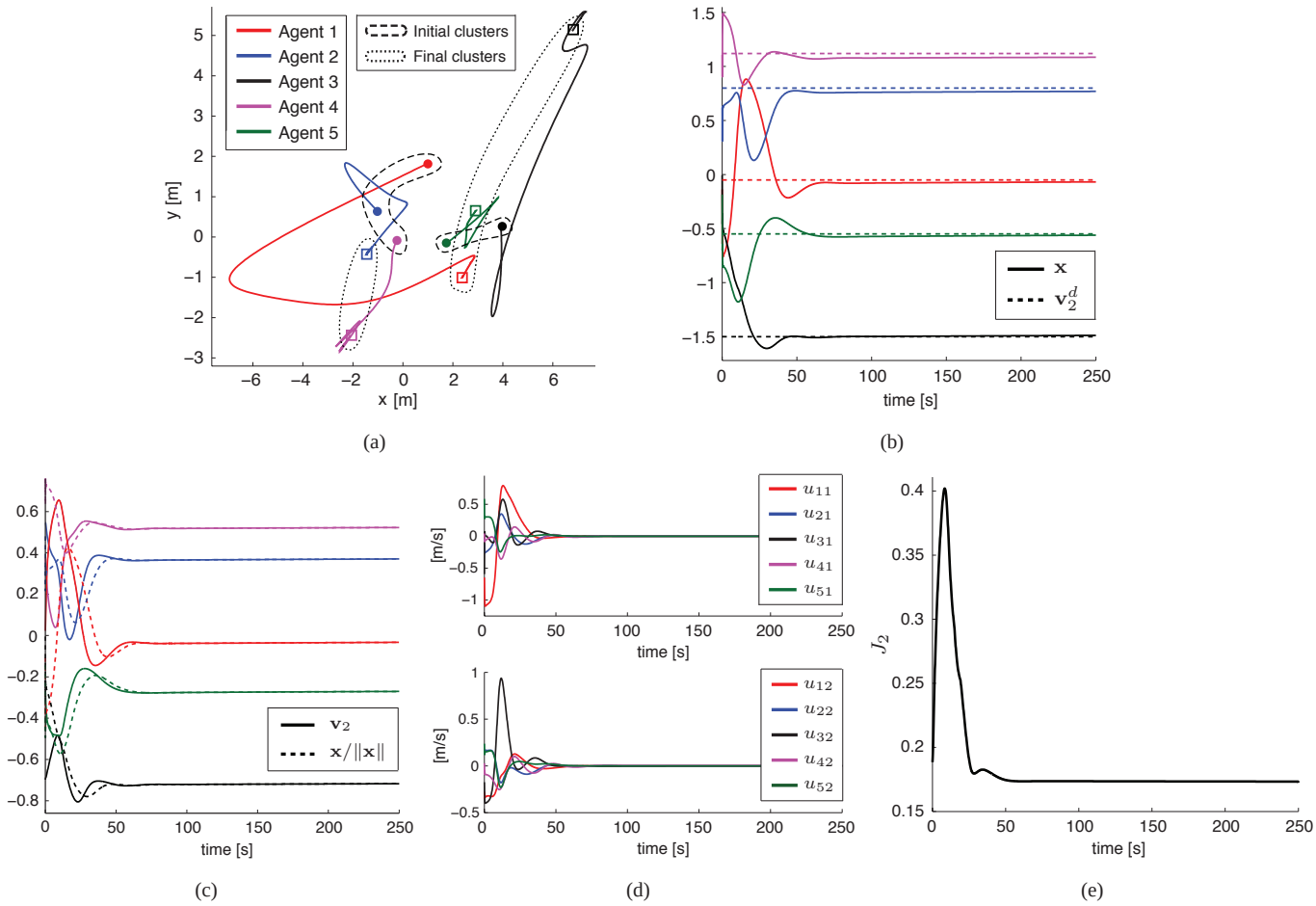


Fig. 2. *Clustering control*: (a) Trajectory of the 5 agents (the initial and final positions are marked with bullets and squares, respectively): the initial and final clusters are indicated with dashed and dotted sets, respectively; (b) Time history of the estimated Fiedler vector $\mathbf{x}$ (solid) and $\mathbf{v}_2^d$ (dash); (c) Time evolution of normalized $\mathbf{v}_2$ (solid) and of $\mathbf{x}/\|\mathbf{x}\|$ (dash); (d) Time history of the control inputs $\mathbf{u}_i = [u_{i1}, u_{i2}]^T$, $i \in \{1, \ldots, 5\}$; (e) Time evolution of the cost function J_2. Note that in (b) and (c) we used the same colors' convention as in (a) and (d).

REFERENCES

[1] M.C. De Gennaro and A. Jadbabaie. Decentralized control of connectivity for multi-agent systems. In *Proc. 45th IEEE Conf. Dec. Contr*, pages 3628–3633, 2006.

[2] M.M. Zavlanos and G.J. Pappas. Potential Fields for Maintaining Connectivity of Mobile Networks. *IEEE Trans. Robot.*, 23(4):812–816, 2007.

[3] M.M. Zavlanos and G.J. Pappas. Distributed Connectivity Control of Mobile Networks. *IEEE Trans. Robot.*, 24(6):1416–1428, 2008.

[4] M.M. Zavlanos, V.M. Preciado, and A. Jadbabaie. Spectral Control of Mobile Robot Networks. In *Proc. American Contr. Conf*, pages 3245–3250, 2011.

[5] Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Trans. Automat. Contr.*, 51(1):116–120, 2006.

[6] A. Simonetto, T. Keviczky, and R. Babuška. On Distributed Maximization of Algebraic Connectivity in Robotic Networks. In *Proc. American Contr. Conf*, pages 2180–2185, 2011.

[7] M. Schuresko and J. Cortés. Distributed motion constraints for algebraic onnectivity of robotic networks. *J. Intell. Robot. Syst.*, 56(1):99–126, 2009.

[8] P. Yang, R. Freeman, G. Gordon, K. Lynch, S. Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2):390–396, 2010.

[9] K. Savla, G. Notarstefano, and F. Bullo. Maintaining Limited-Range Connectivity among Second-Order Agents. *SIAM J. Contr. Optim.*, 48(1):187–205, 2009.

[10] D.P. Spanos and R.M. Murray. Robust connectivity of networked vehicles. In *Proc. 43rd IEEE Conf. Dec. Contr*, volume 3, pages 2893–2898, 2004.

[11] M. Ji and M. Egerstedt. Distributed Coordination Control of Multiagent Systems while Preserving Connectedness. *IEEE Trans. Robot.*, 23(4):693–703, 2007.

[12] U. von Luxburg. A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416, 2007.

[13] A. LaViers, A. Rahmani, and M. Egerstedt. Dynamic Spectral Clustering. In *Proc. Math. Theory Net. Syst.*, 2010.

[14] D.A. Spielman and S.-H. Teng. Convex optimization of graph Laplacian eigenvalues. In *Proc. 36th Annual ACM Symp. Theory Comp.*, pages 81–90, 2004.

[15] T. Sahai, A. Speranzon, and A. Banaszuk. Hearing the Clusters of a Graph: A Distributed Algorithm. *Automatica*, 48(1):15–24, 2012.

[16] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh. Optimal energy aware clustering in sensor networks. *Sensors*, 2(7):258–269, 2002.

[17] B.D.O. Anderson, C. Yu, B. Fidan, and J. Hendrickx. Rigid graph control architectures for autonomous formations. *IEEE Contr. Syst. Mag.*, 28(6):48–63, 2008.

[18] A.S. Lewis and M.L. Overton. Eigenvalue optimization. *Acta Numer.*, 5:149–190, 1996.

[19] E. Stump, A. Jadbabaie, and V. Kumar. Connectivity Management in Mobile Robot Teams. In *Proc. IEEE Int. Conf. Robot. Automat*, pages 1525–1530, 2008.

[20] M. Zavlanos, M. Egerstedt, and G. Pappas. Graph Theoretic Connectivity Control of Mobile Robot Networks. *Proc. IEEE*, 99(9):1525–1540, 2011.

[21] F.R.K. Chung. *Spectral graph theory*. CBMS Regional Conference Series in Mathematics, No. 92. American Mathematical Soc., 1997.

[22] D. Uciński. *Optimal Measurement Methods for Distributed Parameter System Identification*. CRC Press, 2005.

[23] X. Chen, H. Qi, L. Qi, and K.L. Teo. Smooth convex approximation to the maximum eigenvalue function. *J. Global Optim.*, 30(2):253–270, 2004.