



Brief paper

Distributed algebraic connectivity estimation for undirected graphs with upper and lower bounds[☆]



Rosario Aragues^{a,1}, Guodong Shi^b, Dimos V. Dimarogonas^c, Carlos Sagüés^d,
Karl Henrik Johansson^c, Youcef Mezouar^a

^a Institut Pascal, UMR 6602 CNRS - UBP - IFMA, F-63000 Clermont-Ferrand, France

^b College of Engineering and Computer Science, The Australian National University, ACT 0200, Canberra, Australia

^c Access Linnaeus Centre, School of Electrical Engineering, Royal Institute of Technology (KTH), 10044 Stockholm, Sweden

^d Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, 50018 Zaragoza, Spain

ARTICLE INFO

Article history:

Received 8 March 2013

Received in revised form

18 March 2014

Accepted 30 July 2014

Available online 13 November 2014

Keywords:

Multi-agent systems

Control of networks

Sensor networks

Algebraic connectivity

ABSTRACT

The algebraic connectivity of the graph Laplacian plays an essential role in various multi-agent control systems. In many cases a lower bound of this algebraic connectivity is necessary in order to achieve a certain performance. Lately, several methods based on distributed Power Iteration have been proposed for computing the algebraic connectivity of a symmetric Laplacian matrix. However, these methods cannot give any lower bound of the algebraic connectivity and their convergence rates are often unclear. In this paper, we present a distributed algorithm for estimating the algebraic connectivity for undirected graphs with symmetric Laplacian matrices. Our method relies on the distributed computation of the powers of the adjacency matrix and its main interest is that, at each iteration, agents obtain both upper and lower bounds for the true algebraic connectivity. Both bounds successively approach the true algebraic connectivity with the convergence speed no slower than $O(1/k)$.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

The diverse applications of multi-agent systems, e.g., sensor fusion, flocking, formation, or rendezvous (Olfati-Saber, Fax, & Murray, 2007), have led to tremendous research interest in the past decade. A typical multi-agent system is a network of cooperative agents targeting a collective aim using the distributed control design and local information exchange. An underlying communication graph is thus naturally associated with any given multi-agent network. The second smallest eigenvalue of the Laplacian matrix of this graph, known as the algebraic connectivity, plays an important role in various multi-agent applications and in many cases serves

as a fundamental performance measure (Bullo, Cortés, & Martínez, 2009).

The magnitude of the algebraic connectivity determines the connectivity of the communication graph. We first remark some efforts in the literature on maintaining or computing the connectivity of the graph. Control laws for rendezvous and formation control that keep the initial topology have been proposed in Dimarogonas and Johansson (2010) and Ji and Egerstedt (2007). Then in Zavlanos and Pappas (2005), it was shown how to compute the k -hop connectivity matrix of the graph in a centralized fashion. Several distributed methods were then proposed on computing spanning subgraphs (Zavlanos & Pappas, 2008), specifying Laplacian eigenvectors (Qu, Li, & Lewis, 2011), estimating moments of the Laplacian eigenvalue spectrum (Preciado, Zavlanos, Jadbabaie, & Pappas, 2010), or maximizing the algebraic connectivity through motion control (Simonetto, Keviczky, & Babuska, 2011).

How to estimate the value of this algebraic connectivity becomes an intriguing problem for the study of multi-agent networks. In Franceschelli, Gasparri, Giua, and Seatzu (2009), the Laplacian eigenvalues were estimated by making the agents execute a local interaction rule that makes their states oscillate at frequencies corresponding to these eigenvalues and then agents use the Fast Fourier Transform (FFT) on their states to identify these

[☆] A preliminary version of this work appears in Aragues et al. (2012). The material in this paper was partially presented at the 2012 American Control Conference (ACC2012), June 27–29, 2012, Montreal, Canada. This paper was recommended for publication in revised form by Associate Editor Tamas Keviczky under the direction of Editor Frank Allgöwer.

E-mail addresses: raragues@unizar.es (R. Aragues), guodongs@kth.se (G. Shi), dimos@kth.se (D.V. Dimarogonas), csagues@unizar.es (C. Sagüés), kallej@ee.kth.se (K.H. Johansson), Youcef.MEZOUAR@ubpmes.univ-bpclermont.fr (Y. Mezouar).

¹ Tel.: +34 976 76 24 72; fax: +34 976 76 19 14.

Table 1
Notation.

n	Number of agents.
i, j	Agent indices.
k	Iteration, $k \in \mathbb{N}$.
<i>Special matrices and vectors</i>	
\mathbf{I}	Identity matrix.
$\mathbf{0}, \mathbf{1}$	Vectors with all entries equal to 0 and 1.
\mathcal{A}	Adjacency matrix of the graph.
\mathcal{L}	Laplacian matrix, $\mathcal{L} = \text{diag}(\mathcal{A}\mathbf{1}) - \mathcal{A}$.
D	Perron matrix $D = \mathbf{I} - \beta\mathcal{L}$.
C	Deflated matrix, $C = D - \mathbf{1}\mathbf{1}^T/n$.
<i>Matrix operations, eigenvalues and eigenvectors</i>	
$A_{ij}, [A]_{ij}$	(i, j) entry of matrix A .
$\text{diag}(b_1, \dots, b_r)$	matrix A with $A_{ii} = b_i$ and $A_{ij} = 0$.
$\lambda_i(A)$	i th eigenvalue of A .
$\mathbf{v}_i(A)$	i th eigenvector of A .
$\lambda_*(\mathcal{L})$	Algebraic connectivity.
$\ A\ _\infty$	Induced ∞ -norm, $\max_i \sum_{j=1}^n A_{ij} $.
$\ A\ _2$	Spectral norm, $\max_i \sqrt{\lambda_i(A^T A)}$.
$\rho(A)$	Spectral radius, $\max_i \lambda_i(A) $.

eigenvalues. A framework for computing the algebraic connectivity was then introduced in Montijano, Montijano, and Sagues (2011) by iteratively bisecting the interval where it is supposed to belong to. Most of the remaining Laplacian spectra estimation solutions relied on the Power Iteration method or variations (De Gennaro & Jadbabaie, 2006; Kempe & McSherry, 2008; Li & Qu, 2013; Oreshkin, Coates, & Rabbat, 2010; Sabattini, Chopra, & Secchi, 2011; Yang et al., 2010). Power Iteration (Householder, 1964) selects an initial vector and then repeatedly multiplies it by a matrix and normalizes it. This vector converges to the eigenvector associated to the leading eigenvalue (the one with the largest absolute value). The original matrix can be previously deflated so that a particular eigenvalue becomes the leading one. The distributed implementations of the power iteration method let each agent maintain one entry of the state vector. The operations that require global knowledge (normalization and deflation) are usually replaced with averaging iterations, as in Sabattini et al. (2011) and Yang et al. (2010) for continuous-time systems, and in De Gennaro and Jadbabaie (2006) and Li and Qu (2013) for discrete-time systems. A brief summary of the power iteration method can be found in Appendix.

Most of these existing algebraic connectivity estimation methods have asymptotic convergence. However, in order to combine in parallel these methods with some other algorithms or control laws that require the knowledge of the algebraic connectivity, it is necessary to have accurate lower and upper bounds as well as the convergence rate of the algebraic connectivity estimation algorithms (see, e.g., Seyboth, Dimarogonas, & Johansson, 2013), which are typically missing in the literature (Oreshkin et al., 2010; Sabattini et al., 2011; Yang et al., 2010).

In this paper, we present an alternative distributed method for computing the algebraic connectivity (Section 3), whose main interest is that it provides upper and lower bounds for the true algebraic connectivity at each iteration. We prove that both bounds converge to the true algebraic connectivity, with a convergence speed no slower than $O(1/k)$.

2. Preliminaries

We use the notation defined in Table 1.

Consider a set of $n \in \mathbb{N}$ agents with indices $i \in \{1, \dots, n\}$. The agents can exchange information with nearby nodes. This information is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ are the agents, and \mathcal{E} are the edges. There is an edge $(i, j) \in \mathcal{E}$ between nodes i and j if they can exchange data. We assume that \mathcal{G} is connected. We use \mathcal{N}_i for the set of neighbors of a node i with whom i can exchange data, $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$, and

we let d_i be the *degree* of node i defined as the cardinality of \mathcal{N}_i , and $d_{\max} = \max_{i \in \mathcal{V}} d_i$. We say an $n \times n$ matrix C is *compatible* with \mathcal{G} if $C_{ij} = 0$ iff $(i, j) \notin \mathcal{E}$ for $j \neq i$; we let the elements in the diagonal C_{ii} be either equal or different from 0. The adjacency matrix $\mathcal{A} \in \{0, 1\}^{n \times n}$ of \mathcal{G} is

$$\mathcal{A}_{ij} = 1 \quad \text{if } (i, j) \in \mathcal{E}, \quad \mathcal{A}_{ij} = 0 \quad \text{otherwise, for } i, j \in \mathcal{V}.$$

The Laplacian matrix $\mathcal{L} \in \mathbb{R}^{n \times n}$ of \mathcal{G} is the positive-semidefinite matrix

$$\mathcal{L} = \text{diag}(\mathcal{A}\mathbf{1}) - \mathcal{A}, \quad (1)$$

where $\mathbf{1}$ is as in Table 1. Note that the same Laplacian \mathcal{L} is obtained for graphs with $(i, i) \in \mathcal{E}$ and without $(i, i) \notin \mathcal{E}$ self-loops. Both \mathcal{A} and \mathcal{L} are compatible with the graph. We sort the eigenvalues $\lambda_i(\mathcal{L})$ of \mathcal{L} as follows,

$$\lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \dots \leq \lambda_n(\mathcal{L}).$$

The Laplacian matrix \mathcal{L} has the following well known properties, see e.g., Olfati-Saber et al. (2007): (i) its eigenvalues are upper bounded by $\lambda_n(\mathcal{L}) \leq 2d_{\max}$; (ii) it has an eigenvector $\mathbf{v}_1(\mathcal{L}) = \mathbf{1}/\sqrt{n}$ with associated eigenvalue $\lambda_1(\mathcal{L}) = 0$, $\mathcal{L}\mathbf{1}/\sqrt{n} = \mathbf{0}$; and (iii) when \mathcal{G} is connected, all the other eigenvalues are strictly greater than zero.

The *algebraic connectivity* of \mathcal{G} denoted by $\lambda_*(\mathcal{L})$ is defined as the second-smallest eigenvalue $\lambda_2(\mathcal{L})$ of the Laplacian \mathcal{L} . Usually, the distributed algorithms that estimate the algebraic connectivity have asymptotic convergence, i.e., if we let $\hat{\lambda}_i(k)$ be the estimated algebraic connectivity after k iterations of the algorithm, then $\lim_{k \rightarrow \infty} \hat{\lambda}_i(k) = \lambda_*(\mathcal{L})$, but for a finite k , we have $\hat{\lambda}_i(k) \neq \lambda_*(\mathcal{L})$. If we do not know how $\hat{\lambda}_i(k)$ approaches $\lambda_*(\mathcal{L})$, then the selection of the number of steps k and the adjust of a parameter α satisfying $\alpha < \lambda_*(\mathcal{L})$ are non-trivial. Instead, if we know that our estimate approaches $\lambda_*(\mathcal{L})$ satisfying $\hat{\lambda}_i(k) \leq \lambda_*(\mathcal{L})$ for all k , then we can just choose $\alpha < \hat{\lambda}_i(k) \leq \lambda_*(\mathcal{L})$ at any step k .

Problem 2.1. Our goal is to design distributed algorithms to allow the agents to compute $\lambda_*(\mathcal{L})$, and/or a lower bound of $\lambda_*(\mathcal{L})$ in a distributed fashion. ■

From now on, we let C be the following deflated version of the Perron matrix of the Laplacian \mathcal{L} , (Aragues, Shi, Dimarogonas, Sagues, & Johansson, 2012; Olfati-Saber et al., 2007; Xiao & Boyd, 2004; Yang et al., 2010)

$$C = \mathbf{I} - \beta\mathcal{L} - \mathbf{1}\mathbf{1}^T/n, \quad (2)$$

where the eigenvalues $\lambda_1(\mathcal{L}), \dots, \lambda_n(\mathcal{L})$ of the Laplacian and of C are related by

$$\lambda_1(C) = 0, \quad \lambda_i(C) = 1 - \beta\lambda_i(\mathcal{L}), \quad \text{for } i \in \{2, \dots, n\},$$

so that the spectral radius $\rho(C)$ of C is associated to the algebraic connectivity $\lambda_*(\mathcal{L})$ by

$$\lambda_*(\mathcal{L}) = (1 - \rho(C))/\beta, \quad \text{if } 0 < \beta < 1/\lambda_n(\mathcal{L}). \quad (3)$$

We let D be the not-deflated matrix,

$$D = \mathbf{I} - \beta\mathcal{L}, \quad \text{so that } C = D - \mathbf{1}\mathbf{1}^T/n. \quad (4)$$

3. Distributed computation of the algebraic connectivity

We present a distributed method for estimating the algebraic connectivity $\lambda_*(\mathcal{L})$ of an undirected graph, which is not only convergent but also provides lower and upper bounds at each step k . We begin with a brief summary of the method, which is then discussed in detail along this section. The method computes the spectral radius of the deflated matrix C , which is related to the Laplacian \mathcal{L} by Eqs. (2), (3). Agents compute the induced ∞ -norm $\|\cdot\|_\infty$ of matrix C^k , which is the maximum absolute row sum of

the matrix,

$$\|C^k\|_\infty = \max_{i \in \mathcal{V}} \{|[C^k]_{i1}| + \dots + |[C^k]_{in}|\}. \quad (5)$$

This can be easily obtained by the agents using a distributed max-consensus algorithm, provided that each agent i knows the i th row of C^k . The following properties (Horn & Johnson, 1985, Chap. 5.6) relating $\rho(C)$ and the spectral $\|\cdot\|_2$ and induced ∞ -norms $\|\cdot\|_\infty$ of symmetric matrices C allow us to give lower and upper bounds for our estimated algebraic connectivity at each step:

$$\rho(C) \leq \|C\|_\infty, \quad \rho(C) = \|C\|_2 = \|C\|_2^{\frac{1}{k}},$$

$$\|C^k\|_\infty \leq \sqrt{k} \|C\|_2, \quad \rho(C) = \lim_{k \rightarrow \infty} \|C^k\|_\infty^{\frac{1}{k}}.$$

Observing Eq. (5), agents must compute the powers C^k of matrix C which is not compatible with the graph. As the following result shows, C^k can be easily obtained if agents know the powers D^k of matrix $D = \mathbf{I} - \beta \mathcal{L} = C + \mathbf{1}\mathbf{1}^T/n$ in Eq. (4), which is compatible with \mathcal{G} .

Proposition 3.1. Consider a symmetric Laplacian matrix \mathcal{L} , let $D = \mathbf{I} - \beta \mathcal{L}$, with $\beta > 0$, and let $C = D - \mathbf{1}\mathbf{1}^T/n$. The k th powers of matrices C and D are related as follows, for all $k \geq 1$,

$$C^k = (D - \mathbf{1}\mathbf{1}^T/n)^k = D^k - \mathbf{1}\mathbf{1}^T/n. \quad (6)$$

Proof. Since $\mathbf{1}/\sqrt{n}$ is the eigenvector $\mathbf{v}_1(C)$ of C associated to the eigenvalue $\lambda_1(C) = 0$, $D = C + \mathbf{1}\mathbf{1}^T/n$ and C have the same eigenvectors $\mathbf{v}_i(C) = \mathbf{v}_i(D)$ for all $i \in \{1, \dots, n\}$, and all the eigenvalues equal $\lambda_i(C) = \lambda_i(D)$ for all $i \in \{2, \dots, n\}$, but the first one being $\lambda_1(C) = 0$ whereas $\lambda_1(D) = 1$. Thus, their k th powers have the following expressions:

$$C^k = (0)^k \mathbf{1}\mathbf{1}^T/n + \sum_{i=2}^n (\lambda_i(C))^k \mathbf{v}_i(C) \mathbf{v}_i(C)^T,$$

$$D^k = (1)^k \mathbf{1}\mathbf{1}^T/n + \sum_{i=2}^n (\lambda_i(C))^k \mathbf{v}_i(C) \mathbf{v}_i(C)^T,$$

where $0^k = 0$ and $1^k = 1$, so that we obtain Eq. (6). ■

The powers D^k of a matrix D compatible with the graph can be computed in a distributed fashion by letting each agent i store the i th row of D^k and update each element $[D^k]_{ij}$, for $j = 1, \dots, n$, with its neighbors' data,

$$[D^{k+1}]_{ij} = \sum_{j' \in \mathcal{N}_i \cup \{i\}} D_{ij'} [D^k]_{j'j}. \quad (7)$$

Since D is compatible with \mathcal{G} , $D_{ij'} = 0$ for non-neighbor agents $j' \notin \mathcal{N}_i \cup \{i\}$ and Eq. (7) gives

$$[D^{k+1}]_{ij} = \sum_{j=1}^n D_{ij'} [D^k]_{j'j}, \quad (8)$$

which is exactly the expression for computing the powers of a matrix, i.e., agents obtain exactly D^k at each step k . Observing Eq. (7), when agent i receives $[D^k]_{j'j}$ from its neighbor j' , it needs to know to which agent j (possibly a non-neighbor) this information refers to, for combining it with its own $[D^k]_{ij}$. The following algorithm is a more flexible version of (7), where each agent keeps a list $l_i(k)$ of the identifiers j associated to each of its elements $[D^k]_{ij}$ (in real implementations, these identifiers can be, e.g., the IP addresses). Agents exchange these identifiers $l_i(k)$ together with their elements $[D^k]_{ij}$. Moreover, when agent i discovers in its neighbors' data $j' \in \mathcal{N}_i$ the identifier of a new agent $j \in l_j(k)$, it initializes a new variable $[D^k]_{ij}$ and updates $l_i(k)$ accordingly.

Algorithm 3.2 (Distributed Power Matrix Computation). Each node $i \in \mathcal{V}$ maintains a set of node identifiers $l_i(k)$ and an estimate $\hat{D}_{ij}(k)$

of the (i, j) entries of the k th power of D , $[D^k]_{ij}$, associated to the nodes j such that $j \in l_i(k)$.

1: At $k = 0$, each node $i \in \mathcal{V}$ initializes a single variable $\hat{D}_{ii}(k)$ and a single identifier,

$$\hat{D}_{ii}(0) = 1, \quad l_i(0) = \{i\}, \quad (9)$$

and sends this data to its neighbors \mathcal{N}_i .

2: At each step $k \geq 1$, node i first looks for new nodes in the information $l_j(k)$ received from its neighbors $j' \in \mathcal{N}_i$ and updates its identifiers $l_i(k)$ accordingly,

$$l_i(k+1) = \bigcup_{j' \in \mathcal{N}_i \cup \{i\}} l_j(k). \quad (10)$$

3: Then, node i creates a new variable $\hat{D}_{ij}(k)$ initialized with $\hat{D}_{ij}(k) = 0$, for each recently discovered node j ,

$$j \in l_i(k+1) \quad \text{and} \quad j \notin l_i(k).$$

4: Finally, node i updates all its variables $\hat{D}_{ij}(k)$,

$$\hat{D}_{ij}(k+1) = \sum_{j' \in \mathcal{N}_i \cup \{i\}, j \in l_j(k)} D_{ij'} \hat{D}_{j'j}(k), \quad (11)$$

for $j \in l_i(k+1)$, and sends to its neighbors these variables $\hat{D}_{ij}(k+1)$ and the identifiers $l_i(k+1)$. ■

Proposition 3.3. When D is compatible with the graph, (i) the outcomes $\hat{D}_{ij}(k)$ of Algorithm 3.2 at step $k \geq 0$, with $j \in l_i(k)$, are exactly the entries of the k th power of D , D^k , and (ii) for $j \notin l_i(k)$, the entries of the k th power of D , D^k , equal zero, $[D^k]_{ij} = 0$.

Proof. We first prove (ii). Note that $l_i(k)$ contains the identifiers of the nodes j which are at k or less hops far from node i . Thus, if $j \notin l_i(k)$, then agents i and j are farther than k hops and, since matrix D is compatible with the graph, $[D^k]_{ij} = 0$ (Bullo et al., 2009, Lemma 1.32).

Thus, at step k , agent i has variables $\hat{D}_{ij}(k)$ for all the agents $j \in l_i(k)$ that are at k or less hops from i . For the remaining agents $j \notin l_i(k)$, agent i does not have variables $\hat{D}_{ij}(k)$ yet, but this is not important because their value $[D^k]_{ij}$ is zero anyway.

Now we focus in (i). For $k = 0$, it is straightforward to see that it is true, since $D^0 = \mathbf{I}$, and each agent $i \in \{1, \dots, n\}$ has a single variable $\hat{D}_{ii} = 1$ (Eq. (9)). For $k \geq 1$, we consider Eq. (7) that contains the explicit expression for $D^{k+1} = DD^k$, with D compatible with the graph. From (ii), when agents j', j are farther than k hops, i.e., when $j \notin l_j(k)$, the entry $[D^k]_{j'j} = 0$. Thus, each (i, j) entry of D^{k+1} can be expressed as

$$[D^{k+1}]_{ij} = \sum_{j' \in \mathcal{N}_i \cup \{i\}} D_{ij'} [D^k]_{j'j}$$

$$= \sum_{j' \in \mathcal{N}_i \cup \{i\}, j \in l_j(k)} D_{ij'} [D^k]_{j'j},$$

which is the update rule for $\hat{D}_{ij}(k+1)$ in Eq. (11). ■

Agents use Algorithm 3.2 for computing the powers D^k of matrix D , whereas the aim (Eq. (5)) was to compute C^k . From Proposition 3.1, $C^k = D^k - \mathbf{1}\mathbf{1}^T/n$. The value $\hat{D}_{ij}(k)$ at each agent i is the exact value (not an estimate of it) for the entries $[D^k]_{ij}$ of D^k associated to the agents $j \in l_i(k)$ which are at k or less hops; thus, the equivalent entry of C^k is obtained by subtracting $1/n$ from $\hat{D}_{ij}(k)$. Besides, the entries $[D^k]_{ij}$ for which $j \notin l_i(k)$ equal zero (Proposition 3.3); thus, for each of these $n - |l_i(k)|$ entries, the associated entry in C^k equals $-1/n$. Each agent i uses this transformation for computing $c_i(k) = |[C^k]_{i1}| + \dots + |[C^k]_{in}|$.

Algorithm 3.4 (Distributed Algebraic Connectivity). Let $\varepsilon \in (0, 1)$ a number pre-given to the agents and $\beta = \varepsilon/(2n)$. Consider the

agents execute [Algorithm 3.2](#) for computing the powers of matrix $D = \mathbf{I} - \beta \mathcal{L}$. Then, at each step $k \geq 1$, each node $i \in \mathcal{V}$ has variables $\hat{D}_{ij}^k(k)$, for $j \in I_i(k)$, containing the (i, j) entries of the k th power of matrix D , $[D^k]_{ij}$.

At each step k , each node i locally computes²

$$c_i(k) = \sum_{j \in I_i(k)} |\hat{D}_{ij}^k(k) - 1/n| + (n - |I_i(k)|)/n, \quad (12)$$

and starts a max-consensus ([Tahbaz-Salehi & Jadbabaie, 2006](#)) to get $\max_{j \in \mathcal{V}} c_j(k)$. The algebraic connectivity $\hat{\lambda}_i(k)$ estimated by each agent $i \in \mathcal{V}$ at step $k \geq 1$ is given by³

$$\hat{\lambda}_i(k) = (1 - \hat{\rho}_i(k)) / \beta, \quad \hat{\rho}_i(k) = (\max_{j \in \mathcal{V}} c_j(k))^{1/k}. \quad \blacksquare \quad (13)$$

The execution of Eq. (12) starts a new max-consensus process, indicating in the messages the new step number k . At every time instant, in addition to [Algorithm 3.2](#), there are up to $\text{diam}(\mathcal{G})$ max-consensus processes that run simultaneously in the network. Agents keep on executing new iterations of [Algorithm 3.2](#) for computing C^{k+1} , C^{k+2} , etc., and simultaneously, they update the max values associated to the last $\text{diam}(\mathcal{G})$ steps.

Theorem 3.5. *Let each node i execute [Algorithm 3.4](#) with \mathcal{G} undirected and connected. As $k \rightarrow \infty$, the variables $\hat{\lambda}_i(k)$ converge to the Laplacian algebraic connectivity $\lambda_*(\mathcal{L})$ for all $i \in \mathcal{V}$,*

$$\lim_{k \rightarrow \infty} \hat{\lambda}_i(k) = \lambda_*(\mathcal{L}), \quad (14)$$

and for all $i \in \mathcal{V}$ and all k we have lower and upper bounds for $\lambda_*(\mathcal{L})$:

$$\hat{\lambda}_i(k) \leq \lambda_*(\mathcal{L}) \leq n^{-1/k} \hat{\lambda}_i(k) + \left(1 - n^{-1/k}\right) / \beta. \quad (15)$$

Proof. First note that $\beta = \varepsilon/(2n)$ satisfies $0 < \beta < 1/\lambda_n(\mathcal{L})$ since $\varepsilon \in (0, 1)$ and $\lambda_n(\mathcal{L}) \leq 2d_{\max} < 2n$, where d_{\max} is the maximum degree in the graph. Therefore, the algebraic connectivity is related to the spectral radius of matrix C as in Eq. (3), $\lambda_*(\mathcal{L}) = (1 - \rho(C))/\beta$. From [Proposition 3.3](#), for all $i \in \mathcal{V}$, the variables $\hat{D}_{ij}^k(k)$ are equal to $[D^k]_{ij}$ for $j \in I_i(k)$, whereas $[D^k]_{ij} = 0$ for $j \notin I_i(k)$. Linking this with the fact that $C^k = D^k - \mathbf{1}\mathbf{1}^T/n$ ([Proposition 3.1](#)) yields

$$[C^k]_{ij} = \hat{D}_{ij}^k(k) - 1/n, \quad \text{for } j \in I_i(k),$$

$$[C^k]_{ij} = -1/n, \quad \text{for } j \notin I_i(k), \text{ for all } i \in \mathcal{V}, k \geq 1.$$

When \mathcal{G} is connected, $c_i(k)$ in Eq. (12) is the absolute i th row sum of C^k , and $\hat{\rho}_i(k)$ in Eq. (13) is

$$\hat{\rho}_i(k) = \|C^k\|_{\infty}^{1/k}, \quad \text{for all } i \in \{1, \dots, n\}, k \geq 1. \quad (16)$$

For any induced norm, in particular for the ∞ -norm, it holds ([Horn & Johnson, 1985, Chap. 5.6](#)),

$$\rho(C) = \lim_{k \rightarrow \infty} \|C^k\|_{\infty}^{1/k} = \lim_{k \rightarrow \infty} \hat{\rho}_i(k), \quad (17)$$

and $\rho(C) \leq \|C\|_{\infty}$; therefore,

$$\rho(C) = (\rho(C^k))^{1/k} \leq \|C^k\|_{\infty}^{1/k} = \hat{\rho}_i(k). \quad (18)$$

Since C is symmetric, its spectral norm $\|C\|_2 = \max_i \sqrt{\lambda_i(C^2)}$ equals $\rho(C) = \max_i |\lambda_i(C)|$,

$$\rho(C) = \|C\|_2 = \|C^k\|_2^{1/k}. \quad (19)$$

The spectral $\|C^k\|_2$ and induced infinite $\|C^k\|_{\infty}$ norms of C^k are related by $(\sqrt{n})^{-1} \|C^k\|_{\infty} \leq \|C^k\|_2$, ([Horn & Johnson, 1985, Chap. 5.6](#)), which combined with Eq. (19) give

$$n^{-1/k} \hat{\rho}_i(k) = n^{-1/k} \|C^k\|_{\infty}^{1/k} \leq \|C^k\|_2^{1/k} = \rho(C). \quad (20)$$

From Eqs. (17)–(20),

$$\rho(C) = \lim_{k \rightarrow \infty} \hat{\rho}_i(k), \quad n^{-1/k} \hat{\rho}_i(k) \leq \rho(C) \leq \hat{\rho}_i(k), \quad (21)$$

which combined with Eqs. (3) and (13) give

$$\lim_{k \rightarrow \infty} (1 - \beta \hat{\lambda}_i(k)) = 1 - \beta \lambda_*(\mathcal{L}), \quad \text{and} \\ \left(n^{-1/k} - \beta n^{-1/k} \hat{\lambda}_i(k)\right) \leq (1 - \beta \lambda_*(\mathcal{L})) \leq (1 - \beta \hat{\lambda}_i(k)),$$

which lead to Eqs. (14) and (15). \blacksquare

Corollary 3.6. *Let each node i execute [Algorithm 3.4](#) with \mathcal{G} undirected and connected. The estimation errors, between the estimated $\hat{\lambda}_i(k)$ and the true Laplacian algebraic connectivity $\lambda_*(\mathcal{L})$, evolve according to:*

$$|\lambda_*(\mathcal{L}) - \hat{\lambda}_i(k)| \leq \frac{1}{k} \log \sqrt{n} (1/\beta - \lambda_*(\mathcal{L})) + O(1/k^2).$$

Proof. Noticing

$$|\lambda_*(\mathcal{L}) - \hat{\lambda}_i(k)| \leq \left(\sqrt{n}^{1/k} - 1\right) (1/\beta - \lambda_*(\mathcal{L})) \\ = \left(e^{\frac{1}{k} \log \sqrt{n}} - 1\right) (1/\beta - \lambda_*(\mathcal{L})),$$

the desired conclusion becomes clear in light of the Taylor series of $e^{\frac{1}{k} \log \sqrt{n}} - 1$ with respect to $1/k$. \blacksquare

While most of the results in this paper refer to fixed undirected graphs, [Algorithm 3.2](#) can be also used with non-symmetric matrices D , and with time-varying graphs ($D(k)$ depends on the step k), in which case it gives

$$\hat{D}_{ij}^k(k) = [D(k)D(k-1) \cdots D(1)]_{ij}, \quad \text{for } j \in \mathcal{V}.$$

Equivalently, the estimate $\hat{\rho}_i(k)$ of [Algorithm 3.4](#) in Eqs. (13) and (16) for time-varying graphs is

$$\hat{\rho}_i(k) = \|C(k)C(k-1) \cdots C(1)\|_{\infty}^{1/k},$$

which captures the connectivity up to the current time step. This is related, e.g., to the asymptotic convergence rate of certain distributed systems ([Xiao, Boyd, & Lall, 2005](#)).

Remark 3.7 (Knowledge of n). Our method requires knowing n for: (i) computing $\beta = \varepsilon/(2n)$ satisfying $\beta < 1/\lambda_n(\mathcal{L})$, and for (ii) executing Eqs. (12)–(13). For (i), it is enough to know an upper bound of n or to compute the max degree d_{\max} with a max-consensus algorithm on the node degrees. In fact, this is common to all the methods deflating \mathcal{L} ([Oreshkin et al., 2010](#); [Sabattini et al., 2011](#); [Yang et al., 2010](#)).

However, (ii) requires the exact value of n , like, e.g., in [De Genaro and Jadbabaie \(2006\)](#). If \mathcal{G} is fixed and connected, agents can automatically obtain n as follows. Note that each agent i can always compute the powers of matrix D ([Algorithm 3.2](#)) within [Algorithm 3.4](#); let k_i be the first instant for which agent i has discovered the identifiers of all the agents in the network, and thus $I_i(k)$ stops changing, $k_i = \min\{k \mid I_i(k) = I_i(k-1)\}$; then, $n = |I_i(k_i)|$. At this moment, agent i can proceed with Eqs. (12)–(13). Moreover, it can stop executing steps 2 to 3 in [Algorithm 3.2](#) and exchanging variables $I_i(k)$. Alternatively, n can be computed in an initial phase.

² Recall that node i maintains the i th row of D^k .

³ It is available at the nodes when the associated max-consensus finishes $\text{diam}(\mathcal{G})$ iterations later.

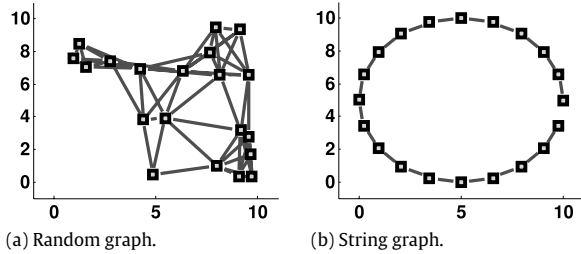


Fig. 1. Scenarios tested. (a) 20 agents (black squares) are placed randomly in a region of 10×10 m; there is an edge $e = (i, j) \in \mathcal{E}$ (gray lines) between pairs of agents that are closer than 4 m; $\lambda_*(\mathcal{L}) = 0.7103$. (b) String graph with agents 1 and 20 in the extremes; $\lambda_*(\mathcal{L}) = 0.0246$.

A method that does not require n at all is Franceschelli et al. (2009); however, it assumes continuous communication, and it is unclear if the discrete versions will require knowing n to choose the step size or to determine triggering conditions. ■

Remark 3.8 (Essential Spectral Radius). The essential spectral radius $\rho_{\text{ess}}(\mathcal{W}) = \rho(\mathcal{W} - \mathbf{1}\mathbf{1}^T/n)$ plays an important role in systems that rely on stochastic weight matrices \mathcal{W} , such as the Metropolis weights (Xiao et al., 2005). Algorithm 3.4 can be adapted for computing $\rho_{\text{ess}}(\mathcal{W})$ by replacing matrices C (Eq. (2)) and D (Eq. (4)) with $C = \mathcal{W} - \mathbf{1}\mathbf{1}^T/n$, $D = \mathcal{W}$, so that $\rho_{\text{ess}}(\mathcal{W}) = \rho(C)$.

The essential spectral radius $\hat{\rho}_i(k)$ (Eq. (13)) estimated at step k by agent i converges to the true $\rho_{\text{ess}}(\mathcal{W})$, with upper and lower bounds, as in Eq. (21). ■

Remark 3.9 (Complexity). Our method has memory, computational, and communication complexities order n per agent and iteration. This is a benefit compared with De Gennaro and Jadbabaie (2006), whose complexities are order n^2 . Other algebraic connectivity estimation algorithms (Franceschelli et al., 2009; Sabattini et al., 2011; Yang et al., 2010) have lighter memory costs, constant per agent. However, they assume continuous communication whereas messages are sent at discrete time instances in real multi-agent systems. To ensure that the discrete-time version of the system properly resembles the original one, a small step size should be chosen, giving rise to high communication costs. The most efficient method is Oreshkin et al. (2010), which considers discrete-time communication, and that has memory, computational, and communication complexities constant per agent and iteration. In Section 4 we compare the performance of our method and Oreshkin et al. (2010) experimentally, concluding that we achieve similar speed rates, with the additional benefit that we give upper and lower bounds. ■

4. Simulations

We have performed a set of simulations with $n = 20$ nodes placed as in the two scenarios in Fig. 1.

Fig. 2 shows the results of our Distributed Algebraic Connectivity method (Algorithm 3.4). The estimated algebraic connectivity $\hat{\lambda}_i(k)$ at step k (Fig. 2, *dac*, dark red dashed) is the same for all the agents $i \in \mathcal{V}$; it lower bounds the true algebraic connectivity $\lambda_*(\mathcal{L})$ (black solid) and asymptotically converges to $\lambda_*(\mathcal{L})$. The expression $(\sqrt{n})^{-\frac{1}{k}} \hat{\lambda}_i(k) + (1 - (\sqrt{n})^{-\frac{1}{k}})/\beta$ (*dac up*, red solid) upper bounds $\lambda_*(\mathcal{L})$ and asymptotically converges to $\lambda_*(\mathcal{L})$. The rates of convergence for the random and string graphs are very similar.

Fig. 3 compares our method (Algorithm 3.4) against the Centralized Power Iteration (Algorithm A.1), which converges exponentially (order r^k , where r is the rate between the two largest modulus eigenvalues); thus it is expected to exhibit a fast convergence, whereas our method has a convergence rate order $\frac{1}{k}$

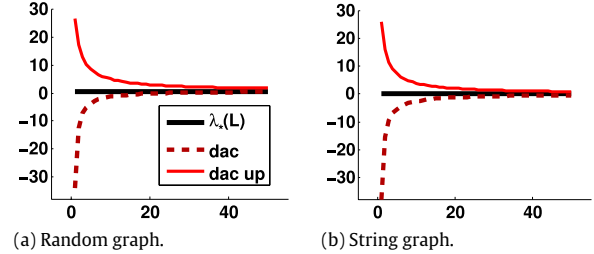


Fig. 2. Algebraic connectivity $\hat{\lambda}_i(k)$ estimated for 40 steps of our method (Algorithm 3.4). $\hat{\lambda}_i(k)$ (*dac*, dark red dashed) and the expression $(\sqrt{n})^{-\frac{1}{k}} \hat{\lambda}_i(k) + (1 - (\sqrt{n})^{-\frac{1}{k}})/\beta$ (*dac up*, red solid) respectively lower and upper bounds the true $\lambda_*(\mathcal{L})$ (black solid), and both converge to $\lambda_*(\mathcal{L})$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

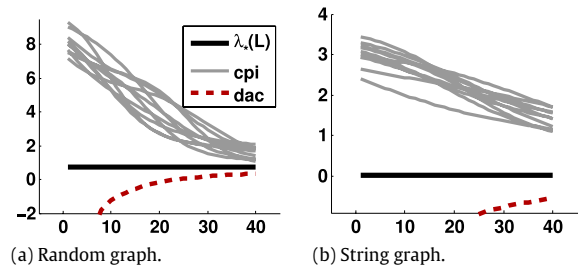


Fig. 3. Estimates for 40 steps of our method ($\hat{\lambda}_i(k)$, *dac*, dark red dashed), and of the Centralized Power Iteration (Algorithm A.1) with 10 different initial vectors ($\hat{\lambda}_{cpi}(k)$, *cpi*, gray solid). The true $\lambda_*(\mathcal{L})$ is shown in black solid. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(Corollary 3.6). In practice, the estimates of the Centralized Power Iteration $\hat{\lambda}_{cpi}(k)$ (*cpi*, gray solid) converged to the true $\lambda_*(\mathcal{L})$ (black solid) slowly for the string graph, where the two largest modulus eigenvalues have similar values; we show the results for 10 different initial vectors. Besides, the estimates $\hat{\lambda}_{cpi}(k)$ are larger than $\lambda_*(\mathcal{L})$, as opposed to the goal stated in Problem 2.1. Our method (*dac*, dark red dashed) converged fast to $\lambda_*(\mathcal{L})$ in both cases, and it produced $\hat{\lambda}_i(k)$ smaller than $\lambda_*(\mathcal{L})$; additionally, it gives an upper bound in case it is needed.

Fig. 4 analyzes the estimates (y -axis) versus the total size of messages sent per agent (x -axis), for our method ($\hat{\lambda}_i(k)$, *dac*, dark red dashed), and for two distributed versions of the power iteration. In the first one, the normalization and deflation operations in Eq. (A.1) are replaced with $T_{\text{cons}} = \{20, 50, 100\}$ averaging steps, using the discrete-time rule $\mathbf{w}(t+1) = \hat{\mathbf{W}}\mathbf{w}(t)$, with $\hat{\mathbf{W}}$ the Metropolis weight matrix (Xiao et al., 2005). As T_{cons} increases, the estimates (*api*, blue solid) converge to a value closer to $\lambda_*(\mathcal{L})$ (black solid), but the communication load increases. The second method (Oreshkin et al., 2010) avoids the deflation by building a zero-average initial vector, and it normalizes using the ∞ -norm (computed with max-consensus). Its estimates (*mpi*, green solid) converge to the true $\lambda_*(\mathcal{L})$ (black solid), using a similar number of messages as our method $\hat{\lambda}_i(k)$ (*dac*, dark red dashed), although without giving any bounds. A similar behavior has been observed for scenarios with different network sizes.

5. Conclusions

We have presented a distributed method to compute the algebraic connectivity $\lambda_*(\mathcal{L})$ and the essential spectral radius $\rho_{\text{ess}}(\mathcal{W})$ for networked agent systems with limited communication. At each iteration, the algorithm produces both upper and lower bound estimates of $\lambda_*(\mathcal{L})$. We have proved theoretically and experimentally

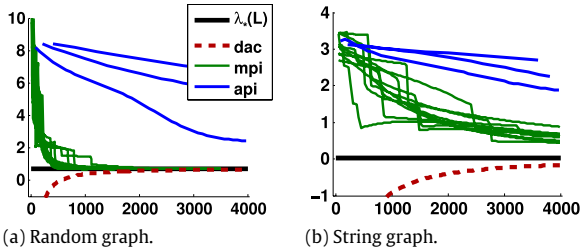


Fig. 4. Estimates (y -axis) vs. messages exchanged (x -axis) for our method $\hat{\lambda}_i(k)$ (dac, dark red dashed); and for the Dist. Power Iter. with max-consensus (mpi, green solid) and with averaging (api, blue solid) with $T_{\text{cons}} = \{20, 50, 100\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

that both estimates asymptotically converge to the true $\lambda_*(\mathcal{L})$. This ability to give upper and lower bounds has a great importance for combining this method with higher level algorithms, executing both processes simultaneously. Although our agents send messages of size n at each step, we have shown that our method has similar communication load as distributed implementations of the Power Iteration method.

Acknowledgments

Partially supported by the Spanish projects and grants Ministerio de Economía y Competitividad DPI2009-08126, DPI2012-32100, BES-2007-14772, by grants from the French program investissement d'avenir managed by the National Research Agency (ANR), the European Commission (Auvergne FEDER funds) and the Région Auvergne in the framework of the LabEx IMobS3 (ANR-10-LABX-16-01), and by the Swedish Research Council.

Appendix. Power iteration

Algorithm A.1 (Centralized Power Iteration). Let $\mathbf{z}(k) \in \mathbb{R}^n$ be an estimate of the leading eigenvector, initialized with any value, and updated at each step k with

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{C}\mathbf{z}(k) / \|\mathbf{C}\mathbf{z}(k)\| \\ &= (\mathbf{D}\mathbf{z}(k) - \mathbf{1}\mathbf{1}^T\mathbf{z}(k)/n) / \|\mathbf{D}\mathbf{z}(k) - \mathbf{1}\mathbf{1}^T\mathbf{z}(k)/n\|. \end{aligned} \quad (\text{A.1})$$

The estimate $\hat{\rho}_{cpi}(k)$ of the leading eigenvalue $\rho(C)$ at step k is given by the Rayleigh quotient, and the estimated algebraic connectivity $\hat{\lambda}_{cpi}(k)$ is as in Eq. (3),

$$\hat{\rho}_{cpi}(k) = \frac{\mathbf{z}(k)^T \mathbf{C} \mathbf{z}(k)}{\mathbf{z}(k)^T \mathbf{z}(k)}, \quad \hat{\lambda}_{cpi}(k) = \frac{1 - \hat{\rho}_{cpi}(k)}{\beta}. \quad \blacksquare \quad (\text{A.2})$$

Symbol $\|\cdot\|$ in Eq. (A.1) denotes a vector norm. In fact, the same $\hat{\rho}_{cpi}(k)$, $\hat{\lambda}_{cpi}(k)$ are obtained regardless of the specific values used for normalizing.

This method upper bounds $\lambda_*(\mathcal{L})$ whereas our goal was to obtain a lower bound (Problem 2.1), i.e., if we consider the estimate after a finite number of steps k , the method believes that the network has a connectivity slightly higher than the actual one.

Lemma A.2. The algebraic connectivity $\hat{\lambda}_{cpi}(k)$ estimated with Algorithm A.1 for a symmetric matrix C is related to the true Laplacian algebraic connectivity $\lambda_*(\mathcal{L})$ as follows, for all $k \geq 0$:

$$\hat{\lambda}_{cpi}(k) \geq \lambda_*(\mathcal{L}). \quad (\text{A.3})$$

Proof. Consider $\hat{\rho}_{cpi}(k)$ in Eq. (A.2),

$$\hat{\rho}_{cpi}(k) = \frac{\left\| \mathbf{C}^{\frac{1}{2}} \mathbf{z}(k) \right\|_2^2}{\left\| \mathbf{z}(k) \right\|_2^2} \leq \frac{\left\| \mathbf{C}^{\frac{1}{2}} \right\|_2^2 \left\| \mathbf{z}(k) \right\|_2^2}{\left\| \mathbf{z}(k) \right\|_2^2} = \left\| \mathbf{C}^{\frac{1}{2}} \right\|_2^2,$$

and for symmetric matrices, $\|C\|_2 = \rho(C)$, and

$$\left\| \mathbf{C}^{\frac{1}{2}} \right\|_2^2 = \rho(\mathbf{C}^{1/2})^2 = \rho(C),$$

giving $\hat{\rho}_{cpi}(k) \leq \rho(C)$, and thus

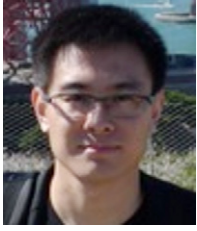
$$\hat{\lambda}_{cpi}(k) = (1 - \hat{\rho}_{cpi}(k)) / \beta \geq (1 - \rho(C)) / \beta = \lambda_*(\mathcal{L}). \quad \blacksquare$$

References

- Aragues, R., Shi, G., Dimarogonas, D. V., Sagues, C., & Johansson, K. H. (2012). Distributed algebraic connectivity estimation for adaptive event-triggered consensus. In *American control conference, Montreal, Canada, June* (pp. 32–37).
- Bullo, F., Cortés, J., & Martínez, S. (2009). *Applied mathematics series, Distributed control of robotic networks*. Princeton University Press, Electronically available at: <http://coordinationbook.info>.
- De Gennaro, M. C., & Jadbabaie, A. (2006). Decentralized control of connectivity for multi-agent systems. In *IEEE conf. on decision and control, San Diego, CA, December* (pp. 3628–3633).
- Dimarogonas, D. V., & Johansson, K. H. (2010). Bounded control of network connectivity in multi-agent systems. *IET Control Theory & Applications*, 4(8), 1330–1338.
- Franceschelli, M., Gasparri, A., Giua, A., & Seatzu, C. (2009). Decentralized Laplacian eigenvalues estimation for networked multi-agent systems. In *IEEE conf. on decision and control, Shanghai, P. R. China, December* (pp. 2717–2722).
- Horn, R. A., & Johnson, C. R. (1985). *Matrix analysis*. Cambridge, UK: Cambridge University Press.
- Householder, A. S. (1964). *The theory of matrices in numerical analysis*. New York: Dover Publications.
- Ji, M., & Egerstedt, M. (2007). Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4), 693–703.
- Kempe, D., & McSherry, F. (2008). A decentralized algorithm for spectral analysis. *Journal of Computer and System Sciences*, 74(1), 70–83.
- Li, C., & Qu, Z. (2013). Distributed estimation of algebraic connectivity of directed networks. *Systems & Control Letters*, 62, 517–524.
- Montijano, E., Montijano, J. I., & Sagues, C. (2011). Adaptive consensus and algebraic connectivity estimation in sensor networks with chebyshev polynomials. In *IEEE conf. on decision and control, Orlando, FL, USA, December* (pp. 4296–4301).
- Olfati-Saber, R., Fax, J. A., & Murray, R. M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.
- Oreshkin, B. N., Coates, M. J., & Rabbat, M. G. (2010). Optimization and analysis of distributed averaging with short node memory. *IEEE Transactions on Signal Processing*, 58(5), 2850–2865.
- Preciado, V. M., Zavlanos, M. M., Jadbabaie, A., & Pappas, G. J. (2010). Distributed control of the Laplacian spectral moments of a network. In *American control conference, Baltimore, Maryland, USA, July* (pp. 4462–4467).
- Qu, Z., Li, C., & Lewis, F. (2011). Cooperative control based on distributed estimation of network connectivity. In *American control conference, San Francisco, CA, USA, July* (pp. 3441–2446).
- Sabattini, L., Chopra, N., & Secchi, C. (2011). On decentralized connectivity maintenance for mobile robotic systems. In *IEEE conf. on decision and control, Orlando, FL, USA, December* (pp. 988–993).
- Seyboth, G. S., Dimarogonas, D. V., & Johansson, K. H. (2013). Event-based broadcasting for multi-agent average consensus. *Automatica*, 49(1), 245–252.
- Simonetto, A., Keviczky, T., & Babuska, R. (2011). On distributed maximization of algebraic connectivity in robotic networks. In *American control conference, San Francisco, CA, USA, July* (pp. 2180–2185).
- Tahbaz-Salehi, A., & Jadbabaie, A. (2006). A one-parameter family of distributed consensus algorithms with boundary: from shortest paths to mean hitting times. In *IEEE conf. on decision and control, San Diego, CA, USA, December* (pp. 4664–4669).
- Xiao, L., & Boyd, S. (2004). Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1), 65–78.
- Xiao, L., Boyd, S., & Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *Symposium on information processing of sensor networks, IPSN, Los Angeles, US, April* (pp. 63–70).
- Yang, P., Freeman, R. A., Gordon, G. J., Lynch, K. M., Srinivasa, S. S., & Sukthankar, R. (2010). Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2), 390–396.
- Zavlanos, M. M., & Pappas, G. J. (2005). Controlling connectivity of dynamic graphs. In *IEEE conf. on decision and control, Seville, Spain, December* (pp. 6388–6393).
- Zavlanos, M. M., & Pappas, G. J. (2008). Distributed connectivity control of mobile networks. *IEEE Transactions on Robotics*, 24(6), 1416–1428.



Rosario Aragues is a Postdoc at the Clermont Université, Institut Pascal, CNRS, UMR 6602, France, since April 2012. She received her M.S. and Ph.D. degrees in System Engineering and Computer Science from the University of Zaragoza, Spain, in 2008 and 2012. Her research interests include multi-robot perception, map merging, and distributed consensus in robotic networks.



Guodong Shi received his B.Sc. degree in Mathematics from Shandong University, Jinan, China, in 2005, and Ph.D. degree in Systems Theory from Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, in 2010, respectively. He was a postdoctoral researcher at the ACCESS Linnaeus Centre, School of Electrical Engineering, Royal Institute of Technology, Sweden, from Aug. 2010 to April 2014. Now he is with the College of Engineering and Computer Science, The Australian National University, Canberra, Australia. His research interests include distributed control, computation, and optimization methods over large-scale networks.



Dimos V. Dimarogonas was born in Athens, Greece, in 1978. He received the Diploma in Electrical and Computer Engineering in 2001 and the Ph.D. in Mechanical Engineering in 2006, both from the National Technical University of Athens (NTUA), Greece. Between May 2007 and February 2009, he was a Postdoctoral Researcher at the Automatic Control Laboratory, School of Electrical Engineering, ACCESS Linnaeus Center, Royal Institute of Technology (KTH), Stockholm, Sweden. Between February 2009 and March 2010, he was a Postdoctoral Associate at the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT), Boston, MA, USA. He is currently an Associate Professor at the Automatic Control Laboratory, ACCESS Linnaeus Center, Royal Institute of Technology (KTH), Stockholm, Sweden.

His current research interests include Multi-Agent Systems, Hybrid Systems and Control, Robot Navigation and Networked Control. He was awarded a Docent in Automatic Control from KTH in 2012. He serves in the Editorial Board of *Automatica* and the *IET Control Theory and Applications* and is a member of IEEE and the Technical Chamber of Greece.

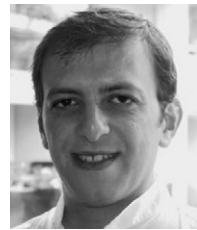


Carlos Sagiés received the M.Sc. and Ph.D. degrees from the University of Zaragoza, Spain. During the course of his Ph.D. he worked on force and infrared sensors for robots. Since 1994 he has been Associate Professor and, since 2009, Full Professor with the Department of Computer Science and Systems Engineering being also Head Teacher in this Department. His current research interest includes control systems, computer vision, visual servoing, distributed consensus and multi-vehicle cooperative control.



Karl Henrik Johansson is Director of the KTH ACCESS Linnaeus Centre and Professor at the School of Electrical Engineering, Royal Institute of Technology, Sweden. He is a Wallenberg Scholar and has held a six-year Senior Researcher Position with the Swedish Research Council. He is Director of the Stockholm Strategic Research Area ICT The Next Generation. He received M.Sc. and Ph.D. degrees in Electrical Engineering from Lund University. He has held visiting positions at UC Berkeley (1998–2000) and California Institute of Technology (2006–2007). His research interests are in networked control systems,

hybrid and embedded system, and applications in transportation, energy, and automation systems. He has been a member of the IEEE Control Systems Society Board of Governors and the Chair of the IFAC Technical Committee on Networked Systems. He has been on the Editorial Boards of several journals, including *Automatica*, *IEEE Transactions on Automatic Control*, and *IET Control Theory and Applications*. He is currently on the Editorial Board of *IEEE Transactions on Control of Network Systems* and the *European Journal of Control*. He has been Guest Editor for special issues, including the one on “Wireless Sensor and Actuator Networks” of *IEEE Transactions on Automatic Control* 2011. He was the General Chair of the ACM/IEEE Cyber-Physical Systems Week 2010 in Stockholm and IPC Chair of many conferences. He has served on the Executive Committees of several European research projects in the area of networked embedded systems. In 2009, he received the Best Paper Award of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems. In 2009, he was also awarded Wallenberg Scholar, as one of the first ten scholars from all sciences, by the Knut and Alice Wallenberg Foundation. He was awarded an Individual Grant for the Advancement of Research Leaders from the Swedish Foundation for Strategic Research in 2005. He received the triennial Young Author Prize from IFAC in 1996 and the Peccei Award from the International Institute of System Analysis, Austria, in 1993. He received Young Researcher Awards from Scania in 1996 and from Ericsson in 1998 and 1999. He is a Fellow of the IEEE.



Youcef Mezouar received the Ph.D. degree in computer science from the Université de Rennes 1, Rennes, France, in 2001 and the “Habilitation Diriger les Recherches” degree from Université Blaise Pascal, Clermont-Ferrand, France, in 2009. For one year, he was a Postdoctoral Associate with the Robotics Laboratory, Department of Computer Science, Columbia University, New York, NY. In 2002, he joined the Robotics and Vision Group, Laboratoire des Sciences et Matériaux pour l'Electronique et d'Automatique (LASMEA), Centre National de la Recherche Scientifique (CNRS)/Université Blaise Pascal, Aubiere, France. He is currently a Member of Institut Pascal, UMR 6602, CNRS/Université Blaise Pascal, Aubiere. He is a Coleader of the Image, Perception Systems, Robotics Group, Institut Pascal, and he leads the Robotics and Autonomous Complex Systems Team, Institut Pascal. His research interests include automatics, robotics, and computer vision, particularly visual servoing and mobile-robot navigation.

He is currently a Member of Institut Pascal, UMR 6602, CNRS/Université Blaise Pascal, Aubiere. He is a Coleader of the Image, Perception Systems, Robotics Group, Institut Pascal, and he leads the Robotics and Autonomous Complex Systems Team, Institut Pascal. His research interests include automatics, robotics, and computer vision, particularly visual servoing and mobile-robot navigation.