Consensus-based Distributed Estimation of Laplacian Eigenvalues of Undirected Graphs

Thi Minh Dung Tran and Alain Y. Kibangou

Abstract— In this paper, we present a novel algorithm for estimating eigenvalues of the Laplacian matrix associated with the graph describing the network topology of a multi-agent system or a wireless sensor network. As recently shown, the average consensus matrix can be written as a product of Laplacian based consensus matrices whose stepsizes are given by the inverse of the nonzero Laplacian eigenvalues. Therefore, by solving the factorization of the average consensus matrix, we can infer the Laplacian eigenvalues. We show how solving such a matrix factorization problem in a distributed way. In particular, we formulate the problem as a constrained consensus problem. The proposed algorithm does not require great resources in both computation and storage. This algorithm can also be viewed as a way for decentralizing the design of finite-time average consensus protocol recently proposed in the literature. Eventually, the performance of the proposed algorithm is evaluated by means of simulation results.

I. INTRODUCTION

In order to model and analyze networks of agents or sensors, it is now well established to resort to algebraic graph theory that provides powerful tools and abstractions. In particular, the network topology is generally represented by means of graphs where vertices represent agents whereas edges represent the existence of an interaction between them. Several features of a given network are captured by the graph Laplacian matrix. A comprehensive survey on properties of Laplacian of undirected graphs can be found in [1]. For example, the second smallest eigenvalue of a Laplacian matrix, i.e., the graph algebraic connectivity of the graph, which can be retrieved from the spectrum of the graph, is known to have the main role in the convergence time of various distributed algorithms. It is also a critical parameter that influences the performance and robustness properties of dynamical systems operating over an information network. From the spectrum of the graph Laplacian matrix we can also infer bounds on the graph diameter and state the connectedness of the graph.

In the recent literature devoted to multi-agent dynamic systems and wireless sensor networks, several issues are formulated as consensus problems, which consist in designing a network protocol based on the local information obtained by each agent, such that all agents finally reach an agreement on certain quantities of interest. The network protocol is an interaction rule, which ensures that the whole group can achieve a consensus on the shared data in a distributed manner, i.e. without the coordination of a central authority. In the study of consensus problems, the speed of

convergence is an important index for assessing the proposed protocols. When using Laplacian based consensus matrices, it has been shown that, for speeding up the convergence in an average consensus problem, the optimal consensus matrix is associated with both the largest and the smallest nonzero Laplacian eigenvalues [2]. More recently, it has been shown that all the spectrum of the Laplacian matrix can also be used for designing consensus matrices in order to achieve average consensus in a finite number of steps [3], [4].

Therefore, computing the eigenvalues of the Laplacian matrix is an important issue. Unfortunately, this information, up to now, can be obtained only by centralized algorithms where a global knowledge about the network topology is available. However, during the current decade, various studies have been carried out on decentralized algorithms for estimating the Laplacian eigenvalues.

For instance in [5], the second smallest Laplacian eigenvalue was estimated by resorting to a decentralized power iteration method. In [6] and [7], Fast Fourrier Transform (FFT) based methods were suggested. In these works, the idea is to make the state of each agent oscillates only at frequencies corresponding to the eigenvalues of the network topology. The problem is then mapped into a signal processing one that can be efficiently and independently solved by each agent in applying the FFT algorithm. The approach in [6] involved twice communication burden compared to that in [7]. However, both methods inherit on the limitations of the FFT algorithm. In particular, the resolution of the estimated eigenvalues is strongly dependent on that of the FFT method and the accuracy depends on the amount of stored data. In contrast, in [8], the authors resort to an algebraic method using observability properties of the network. With this method, the eigenvalues of the network matrix can be recovered by solving a local eigenvalue decomposition on an appropriately constructed matrix of observed data. However, this method is only applicable to networks having nodes with sufficient storage and computation capabilities.

In this paper, motivated by the results in [3], we propose a novel algorithm for estimating the Laplacian eigenvalues. By distributively solving the factorization of the average consensus matrix, we show that the Laplacian eigenvalues can be computed as the inverse of the stepsizes in each estimated factor, where these factors are constrained to be structured as Laplacian based consensus matrices. We formulate a constrained optimization problem that can be solved in a distributed way by means of a gradient descent algorithm. The problem, as formulated herein, can be viewed as a consensus problem with equality constraints.

Thi Minh Dung Tran and Alain Y. Kibangou are with the Department of Automatic control, Gipsa-Lab, CNRS, University Joseph Fourier, 11 rue des Mathï£¡matiques, Grenoble Campus, France. email: thi-minh-dung.tran@gipsa-lab.fr, alain.kibangou@ujf-grenoble.fr

The remainder of this paper is organized as follows: in Section II, we first formulate the problem under study. Then, a gradient descent algorithm is derived in Section III for solving a constrained optimization problem. The performance of the proposed algorithm is evaluated in Section IV by means of simulation results before concluding the paper.

II. BACKGROUND AND PROBLEM STATEMENT

In this section, we first recall some graph and matrix properties that will be useful in the derivations carried out in the sequel. Then, we formulate the problem to be studied.

A. Matrix Khatri-Rao Product

Given two matrices $\mathbf{A} \in R^{I \times F}$, $\mathbf{B} \in R^{J \times F}$ with the same number of columns, the Khatri-Rao product $\mathbf{A} \odot \mathbf{B} \in R^{IJ \times F}$ is defined as:

$$\mathbf{A} \odot \mathbf{B} := [\mathbf{a}_1 \otimes \mathbf{b}_1 \cdots \mathbf{a}_F \otimes \mathbf{b}_F] = \mathbf{B} \begin{pmatrix} \text{diag}_1(\mathbf{A}) \\ \vdots \\ \text{diag}_I(\mathbf{A}) \end{pmatrix} \quad (1)$$

where $\mathbf{a}_f$ is the fth column of $\mathbf{A}$, similarly for $\mathbf{b}_f$, and $\otimes$ denotes the Kronecker product of two column vectors, whereas $\text{diag}_i(\mathbf{A})$ stands for the diagonal matrix built with the ith row of $\mathbf{A}$. The Khatri-Rao product can be viewed as a column-wise Kronecker product. Its main property, which will be used herein, is related to the vectorization operation of a given matrix:

$$\text{vec}(\mathbf{A} diag(\mathbf{d})\mathbf{B}^T) = (\mathbf{B} \odot \mathbf{A})\mathbf{d}. \quad (2)$$

where the $vec(.)$ operator stacks the columns of its matrix argument in a vector and $\mathbf{d} \in R^{F \times 1}$. Therefore we can deduce the following identity:

$$\mathbf{A} diag(\mathbf{d})\mathbf{b} = (\mathbf{b}^T \odot \mathbf{A})\mathbf{d}. \quad (3)$$

B. Average consensus Problems

Through out this paper, we consider a connected undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots N\}$ is the set of vertices of graph G, and $E \subset V \times V$ is the set of edges. Vertices V are nodes in a network connected according to E. The neighbors of node i are denoted by $N_i = \{j \in V : (i, j) \in E\}$. We denote by $\mathbf{A}$ the adjacency matrix of the graph. Its entries $a_{i,j}$ being equal to one if $(i, j) \in E$ and zero elsewhere. The graph Laplacian $\mathbf{L}$ is defined as the matrix with entries $l_{i,j}$ given by: $l_{i,j} = \begin{cases} \sum_{k=1, k \neq i}^n a_{i,k}, & \text{if } j = i, \\ -a_{i,j}, & \text{elsewhere.} \end{cases}$

For an undirected graph G, the graph Laplacian $\mathbf{L}$ is symmetric and positive semidefinite. Its eigenvalues, $\lambda_1 \leqslant \lambda_2 \leqslant \ldots \leqslant \lambda_N$, contain very significant information about the topology of the graph G. In particular, we have $\lambda_1 = 0$ that admits $\mathbf{1}$, an N-length all ones column-vector, as eigenvector. Additionally, the graph G is said to be connected if the eigenvalue $\lambda_1 = 0$ is simple.

For each node $i \in V$, let $x_i(t)$ denotes the value of node i at time-step t. Define $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_N(t)]^T$, where N is the number of nodes in the network. Average consensus algorithms can be seen as the distributed solution of an optimization algorithm whose goal is the minimization of the disagreement between the nodes in the network [9]. In other words, average consensus resort to minimizing the cost function

$$\Phi_G(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{(i,j) \in E} (x_j - x_i)^2. \quad (4)$$

Using a steepest descent method, the following linear iteration scheme is obtained:

$$x_i(t) = x_i(t-1) + \alpha \sum_{j \in N_i} (x_j(t-1) - x_i(t-1)) \quad (5)$$

that can be written in matrix form as:

$$\mathbf{x}(t) = (\mathbf{I}_N - \alpha \mathbf{L})\mathbf{x}(t-1). \quad (6)$$

By appropriately selecting the stepsize α, all nodes converge asymptotically to the same value that is the average of the initial ones:

$$\lim_{t \to \infty} \mathbf{x}(t) = \frac{1}{N} \mathbf{1}\mathbf{1}^T \mathbf{x}(0).$$

It has been shown in [3] and [4] that the average consensus matrix can be factored as

$$\prod_{t=D}^{1} \mathbf{W}_t = \frac{1}{N} \mathbf{1}\mathbf{1}^T, \quad (7)$$

where $\mathbf{W}_t = \vartheta_t \mathbf{I}_N + \alpha_t \mathbf{L}$, ϑ_t and α_t being parameters to be designed. In [3], the solution was given by $\vartheta_t = 1$ and $\alpha_t = -\frac{1}{\lambda_{t+1}}$, λ_t being a nonzero Laplacian eigenvalue. Owing to the above factorization, average consensus can then be reached in D steps, D being the number of distinct nonzero Laplacian eigenvalues:

$$\mathbf{x}(D) = \prod_{t=D}^{1} \mathbf{W}_t \mathbf{x}(0) = \frac{1}{N} \mathbf{1}\mathbf{1}^T \mathbf{x}(0) \quad \text{for all } \mathbf{x}(0) \in R^N. \quad (8)$$

C. Problem statement

From the results above, we can note that by factorizing the average consensus matrix, while constraining the factor matrices to be in the form $\mathbf{I}_N - \alpha_t \mathbf{L}$, we can deduce the eigenvalues of the Laplacian matrix as the inverse of α_t. Now, we first state the uniqueness of such a factorization in the following lemma:

Lemma 1: Let $\lambda_2, \cdots, \lambda_{D+1} \neq 0$ be the D distinct nonzero eigenvalues of the graph Laplacian matrix $\mathbf{L}$, then, up to permutation, the sequence $\{\alpha_i\}_{i=1,\cdots,D}$, with $\alpha_i = \frac{1}{\lambda_{i+1}}$, $i = 1, 2, \cdots, D$, is the unique sequence allows getting the minimal factorization of the average consensus matrix as $\frac{1}{N} \mathbf{1}\mathbf{1}^T = \prod_{i=1}^{D} (\mathbf{I}_N - \alpha_i \mathbf{L})$.

Proof: Let us consider the following factorization of the average consensus matrix: $\frac{1}{N} \mathbf{1}\mathbf{1}^T = \prod_{i=1}^{\theta} (\mathbf{I}_N - \alpha_i \mathbf{L})$. The Laplacian matrix $\mathbf{L}$ being symmetric, then:

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Delta}\mathbf{U}^T, \mathbf{U}^T \mathbf{U} = \mathbf{I}_N, \mathbf{U}\mathbf{U}^T = \mathbf{I}_N$$

where $\boldsymbol{\Delta} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ and $\mathbf{U} = (\frac{1}{\sqrt{N}}\mathbf{1} \quad \tilde{\mathbf{U}})$ with $\tilde{\mathbf{U}}^T \tilde{\mathbf{U}} = \mathbf{I}_{N-1}$ and $\tilde{\mathbf{U}}^T \mathbf{1} = \mathbf{0}$. Therefore, the above factorization

can be rewritten as:

$$\mathbf{U}(\prod_{i=1}^{\theta}(\mathbf{I}_N - \alpha_i \mathbf{\Delta}))\mathbf{U}^T = \frac{1}{N}\mathbf{1}\mathbf{1}^T$$

or equivalently:

$$\mathbf{U}(\prod_{i=1}^{\theta}(\mathbf{I}_N - \alpha_i \mathbf{\Delta}))\mathbf{U}^T = \mathbf{U}\text{diag}(1 \quad 0\cdots0)\mathbf{U}^T. \qquad (9)$$

From equation (9), we can see that $\prod_{i=1}^{\theta}(1 - \alpha_i \lambda_1) = 1$, that is always fulfilled since $\lambda_1 = 0$. In addition, we have:

$$\prod_{i=1}^{\theta}(1 - \alpha_i \lambda_j) = 0, \quad j = 2, \cdots, N. \qquad (10)$$

Taking into account the multiplicities of the eigenvalues, we only have D distinct equations. It is obvious that the D equalities above are fulfilled only if $1/\alpha_i$ belong to the Laplacian spectrum and if there are at least D distinct α_i. Therefore, the Laplacian spectrum implies the minimal factorization of the average consensus matrix. ∎

Our goal is to design a protocol for estimating the Laplacian eigenvalues of a given undirected graph in a distributed manner. More precisely, a novel algorithm is derived by making use of the Lagrange method to solve a distributed constrained optimization problem.

III. DISTRIBUTED ESTIMATION OF LAPLACIAN EIGENVALUES THROUGH DISTRIBUTED FACTORIZATION OF THE AVERAGE CONSENSUS MATRIX

Given an initial input-output pair $\{\mathbf{x}(0), \bar{\mathbf{x}}\}$, with $\bar{\mathbf{x}} = \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{x}(0)$ and an initial guess of the minimal number of factors D, the matrix factorization problem (8) is equivalent to minimize the cost function:

$$E(\mathbf{W}) = \|\mathbf{x}(D) - \bar{\mathbf{x}}\|^2$$

that can also be rewritten as

$$E(\mathbf{W}) = \left\|\prod_{t=D}^{1} \mathbf{W}_t \mathbf{x}(0) - \bar{\mathbf{x}}\right\|^2, \qquad (11)$$

where D is the number of steps before reaching average consensus and $\mathbf{W}_t = \mathbf{I}_N - \alpha_t \mathbf{L}$.

Note that there is no need for a central node for setting the initial input-output pair. Indeed, such a pair can be obtained after running a standard average consensus algorithm. Each node has to keep in memory its own initial value $x_i(0)$ and the final consensus value $\bar{x}$.

Solving the factorization problem (11) consists in finding the sequence of stepsizes $\{\alpha_t\}_{t=1,\cdots,D}$. It is obvious that α_t are global parameters. To relax these constraints, we define the factor matrices as $\mathbf{W}_t = \mathbf{I}_N - \mathbf{\Lambda}_t\mathbf{L}$, where $\mathbf{\Lambda}_t = diag(\boldsymbol{\alpha}_t)$, $\boldsymbol{\alpha}_t = [\alpha_{t,1}, \alpha_{t,2}, \ldots, \alpha_{t,N}]$, $t = 1, 2, \ldots, D$. We reformulate the problem above as a constrained consensus problem, that is to compute the sequence of stepsizes $\{\boldsymbol{\alpha}_t\}$ so that $\alpha_{t,1} = \alpha_{t,2} = \ldots = \alpha_{t,N}$.

For distributively carrying out the factorization of the average consensus matrix as factors of Laplacian based consensus matrices, the idea is to minimize the disagreement between neighbors on the stepsizes α_t while ensuring that the factorization of the average consensus matrix is achieved. Such a factorization is assessed by constraining the values of the nodes after D iterations of the consensus algorithm to be equal to the average of the initial values:

$$\min_{\boldsymbol{\alpha}_t \in R^{N \times 1}, t=1,2,\ldots,D} \quad \frac{1}{2}\sum_{t=1}^{D}\sum_{i \in V}\sum_{j \in N_i}(\alpha_{t,j} - \alpha_{t,i})^2, \quad (12)$$

$$\text{subject to} \quad \mathbf{x}(D) = \bar{\mathbf{x}}.$$

Similarly to (4), we can rewrite the problem (12) as follows:

$$\min_{\boldsymbol{\alpha}_t \in R^{N \times 1}, t=1,2,\ldots,D} \quad \frac{1}{2}\sum_{t=1}^{D}\boldsymbol{\alpha}_t^T\mathbf{L}\boldsymbol{\alpha}_t, \qquad (13)$$

$$\text{subject to} \quad \mathbf{x}(D) = \bar{\mathbf{x}}.$$

The constrained optimization problem (13) can then be performed as an unconstrained optimization problem by means of a Lagrange method with a Lagrange function defined as follows:

$$H(\boldsymbol{\alpha}_{1:D}, \mathbf{y}) = \frac{1}{2}\sum_{t=1}^{D}\boldsymbol{\alpha}_t^T\mathbf{L}\boldsymbol{\alpha}_t + \mathbf{y}^T(\mathbf{x}(D) - \bar{\mathbf{x}}), \qquad (14)$$

where $\mathbf{y} \in R^{N \times 1}$ stands for the Lagrange multipliers.

The problem (14) can be viewed as D consensus problems to be solved simultaneously with a constraint that provides a kind of external reference to be tracked.

The solution of this unconstrained optimization problem can be obtained iteratively by using a gradient descent method:

$$\boldsymbol{\alpha}_t[k+1] = \boldsymbol{\alpha}_t[k] - \beta\frac{\partial H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})}{\partial \boldsymbol{\alpha}_t[k]},$$

$$\mathbf{y}[k+1] = \mathbf{y}[k] + \mu(\mathbf{x}_D[k] - \bar{\mathbf{x}}),$$

where β and μ stand for the stepsizes of the gradient descent method. Note that to avoid misunderstanding between time-step D and iteration k, we denote $\mathbf{x}(D)$ as $\mathbf{x}_D$.

In order to derive the corresponding distributed algorithm, we now state the following technical lemma:

Lemma 2: The derivatives of the cost function $H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})$ defined in (13) can be computed as follows:

$$\frac{\partial H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})}{\partial \boldsymbol{\alpha}_t} = \mathbf{L}\boldsymbol{\alpha}_t - \text{diag}^{-1}(\boldsymbol{\alpha}_t)\text{diag}(\mathbf{x}_{t-1} - \mathbf{x}_t)\boldsymbol{\delta}_t, \quad (15)$$

where

$$\boldsymbol{\delta}_D = \mathbf{y} \text{ and } \boldsymbol{\delta}_{t-1} = \mathbf{W}_t\boldsymbol{\delta}_t, \quad t = 1, \ldots, D. \qquad (16)$$

Proof: The consensus network being a linear system we know that $\mathbf{x}(t) = \mathbf{W}_t\mathbf{x}(t-1)$, therefore we can explicitly write the output according to the weighting matrix of interest, i.e. $\mathbf{x}(D) = \mathbf{W}_D\mathbf{x}(D-1)$ and $\mathbf{x}(D) = \prod_{j=D}^{t+1}\mathbf{W}_j\mathbf{W}_t\mathbf{x}(t-1)$,

$t = 1, \cdots, D-1$. The cost function can be written as:

$$H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})$$

$$= \frac{1}{2} \sum_{t=1}^{D} \boldsymbol{\alpha}_t^T \mathbf{L} \boldsymbol{\alpha}_t + \mathbf{y}^T (\prod_{i=D}^{t+1} \mathbf{W}_i \mathbf{W}_t \mathbf{x}_{t-1} - \bar{\mathbf{x}})$$

$$= \frac{1}{2} \sum_{t=1}^{D} \boldsymbol{\alpha}_t^T \mathbf{L} \boldsymbol{\alpha}_t + \mathbf{y}^T (\prod_{i=D}^{t+1} \mathbf{W}_i (\mathbf{I}_N - \text{diag}(\boldsymbol{\alpha}_t) \mathbf{L} \mathbf{x}_{t-1} - \bar{\mathbf{x}})$$

$$= \frac{1}{2} \sum_{t=1}^{D} \boldsymbol{\alpha}_t^T \mathbf{L} \boldsymbol{\alpha}_t + \mathbf{y}^T (\prod_{i=D}^{t+1} \mathbf{W}_i \mathbf{x}_{t-1} - \bar{\mathbf{x}})$$

$$- \mathbf{y}^T \prod_{i=D}^{t+1} \mathbf{W}_i \text{diag}(\boldsymbol{\alpha}_t) \mathbf{L} \mathbf{x}_{t-1}).$$

Now, using the property (3) of the Khatri-Rao product, we get:

$$H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})$$

$$= \frac{1}{2} \sum_{t=1}^{D} \boldsymbol{\alpha}_t^T \mathbf{L} \boldsymbol{\alpha}_t + \mathbf{y}^T (\prod_{i=D}^{t+1} \mathbf{W}_i \mathbf{x}_{t-1} - (\mathbf{x}_{t-1}^T \mathbf{L}^T \odot \prod_{i=D}^{t+1} \mathbf{W}_i) \boldsymbol{\alpha}_t - \bar{\mathbf{x}}).$$

Therefore, we can easily deduce that

$$\frac{\partial H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})}{\partial \boldsymbol{\alpha}_t} = \mathbf{L} \boldsymbol{\alpha}_t - (\mathbf{x}_{t-1}^T \mathbf{L}^T \odot \prod_{i=D}^{t+1} \mathbf{W}_i)^T \mathbf{y}.$$

Taking the symmetry of the consensus matrices into account, we get:

$$\frac{\partial H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})}{\partial \boldsymbol{\alpha}_t} = \mathbf{L} \boldsymbol{\alpha}_t - \text{diag}(\mathbf{x}_{t-1}^T \mathbf{L}^T)^T \prod_{i=t+1}^{D} \mathbf{W}_i \mathbf{y}$$

$$= \mathbf{L} \boldsymbol{\alpha}_t - \text{diag}(\mathbf{L} \mathbf{x}_{t-1}) \prod_{i=t+1}^{D-2} \mathbf{W}_i \mathbf{W}_{D-1} \underbrace{\mathbf{W}_D \mathbf{y}}_{\boldsymbol{\delta}_{D-1}}$$

$$= \mathbf{L} \boldsymbol{\alpha}_t - \text{diag}(\mathbf{L} \mathbf{x}_{t-1}) \prod_{i=t+1}^{D-2} \mathbf{W}_i \underbrace{\mathbf{W}_{D-1} \boldsymbol{\delta}_{D-1}}_{\boldsymbol{\delta}_{D-2}}$$

$$= \mathbf{L} \boldsymbol{\alpha}_t - \text{diag}(\mathbf{L} \mathbf{x}_{t-1}) \boldsymbol{\delta}_t.$$

Since $\mathbf{x}_t = (\mathbf{I}_N - diag(\boldsymbol{\alpha}_t) \mathbf{L}) \mathbf{x}_{t-1}$, we can deduce that $\mathbf{L} \mathbf{x}_{t-1} = \text{diag}^{-1}(\boldsymbol{\alpha}_t)(\mathbf{x}_{t-1} - \mathbf{x}_t)$. Hence, $\text{diag}(\mathbf{L} \mathbf{x}_{t-1}) = \text{diag}^{-1}(\boldsymbol{\alpha}_t) \text{diag}(\mathbf{x}_{t-1} - \mathbf{x}_t)$. As a consequence,

$$\frac{\partial H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})}{\partial \boldsymbol{\alpha}_t} = \mathbf{L} \boldsymbol{\alpha}_t - \text{diag}^{-1}(\boldsymbol{\alpha}_t) \text{diag}(\mathbf{x}_{t-1} - \mathbf{x}_t) \boldsymbol{\delta}_t.$$

∎

Applying the results of Lemma 2, the updating scheme of the optimization algorithm is as follows:

$$\boldsymbol{\alpha}_t[k+1] = \boldsymbol{\alpha}_t[k] - \beta \frac{\partial H(\boldsymbol{\alpha}_{1:D}, \mathbf{y})}{\partial \boldsymbol{\alpha}_t[k]}$$

$$= (\mathbf{I}_N - \beta \mathbf{L}) \boldsymbol{\alpha}_t[k] + \beta \text{diag}^{-1}(\boldsymbol{\alpha}_t) \text{diag}(\mathbf{x}_{t-1} - \mathbf{x}_t) \boldsymbol{\delta}_t, \quad (17)$$

$$\mathbf{y}[k+1] = \mathbf{y}[k] + \mu(\mathbf{x}_D[k] - \bar{\mathbf{x}}). \quad (18)$$

The proposed distributed algorithm can then be described as follows:

Algorithm 1: Distributed estimation of Laplacian eigenvalues

1) *Initialization*:
   - Number of steps D,
   - Initial input-output values $\{x_i(0), \bar{x}_i\}$, $i = 1, \cdots, N$, with $\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i(0)$ obtained from a standard average consensus algorithm.
   - Random initial stepsizes $\boldsymbol{\alpha}_t(0), t = 1, \ldots, D$ and initial Lagrange multipliers $y_i(0)$, $i = 1, \cdots, N$
   - Learning rate: $0 < \beta, \mu < 1$;
2) Set $k = 0$;
   a) Set $k := k+1$,
   b) Lagrange multiplier:
      $$\delta_{D,i}[k] = y_i[k].$$
   c) Propagate the Lagrange multipliers for $t = D, \cdots, 2$:
      $$\delta_{t-1,i}[k] = \delta_{t,i}[k] + \sum_{j \in N_i} (\delta_{t,j}[k] - \delta_{t,i})[k].$$
   d) Finite-time average consensus steps:
      i) $x_{0,i}[k] = x_i(0)$.
      ii) For $t = 1, 2, \cdots, D$, each node sends a message $m_{i,t}[k]$ containing its current local value $x_{t,i}[k]$, and the local stepsize $\alpha_{t,i}[k]$.
      iii) After receiving the messages $m_{j,t}[k]$ from its neighbors $j \in N_i$, each node i carries out the following updates:
          $$x_{t,i} = x_{t-1,i} + \alpha_{t,i}[k] \sum_{j \in N_i} (x_{t-1,j} - x_{t-1,i}),$$
          $$\alpha_{t,i}[k+1] = \alpha_{t,i}[k] - \beta (\sum_{j \in N_i} (\alpha_{t,j}[k] - \alpha_{t,i}[k]))$$
          $$- \beta \delta_{t,i}[k] \sum_{j \in N_i} (x_{t-1,j}[k] - x_{t-1,i}[k]).$$
   e) Update the Lagrange multiplier
      $$y_i[k+1] = y_i[k] + \mu(x_{D,i}[k] - \bar{x}_i).$$
   f) Return to 2a or stop the iterations if a stopping criterion is reached.
3) Each node deduce the Laplacian eigenvalues as $\lambda_{t,i} = \frac{1}{\alpha_{t,i}[k+1]}$.

This algorithm can also be viewed as a way for decentralizing the design of the finite-time average consensus protocol proposed in [3]. The convergence of this Lagrange method has been well studied in the literature [10]. Convergence to a local minimum is guaranteed if we pick appropriate constants β, μ in the interval of $(0,1)$.

IV. SIMULATION RESULTS

In this section, we evaluate the efficiency of the proposed distributed algorithm by considering two different examples. The performance is evaluated by means of the mean square error between actual and desired values of nonzero Laplacian eigenvalues:

$$MSE = \frac{1}{DN} \sum_{t=1}^{D} \sum_{i=1}^{N} (\lambda_{t,i} - \hat{\lambda}_{t,i})^2.$$

In addition, we also check the results of the distributed factorization of the average consensus matrix. Then, based on Lemma 1, we can conclude on the correctness of the estimation of the Laplacian eigenvalues.
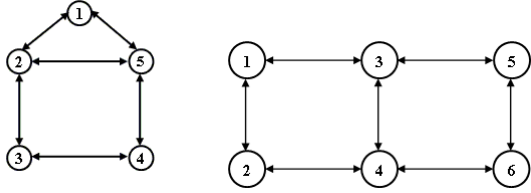

Fig. 1. Network graphs

A. *Example 1*

The network considered in this first example contains 5 nodes whose interactions are modeled with the graph depicted in Figure 1 (left). The corresponding Laplacian matrix $\mathbf{L}$ admits the following spectrum:$\{0, 1.382, 2.382, 3.618, 4.618\}$. Therefore the number of nonzero distinct eigenvalues, that is also the number of steps in the finite-time average consensus protocol, is $D = 4$.

According to the specifications of Algorithm 1, each node randomly initialized its stepsize parameters $\alpha_{1,i}$, $\alpha_{2,i}$, $\alpha_{3,i}$ and $\alpha_{4,i}$. These initial values are drawn from a uniform distribution on the unit interval. The algorithm is stopped when the local disagreement Figure 2 depicts the MSE according to the number of iterations of the proposed distributed algorithm. We can note a globally monotonic behavior.
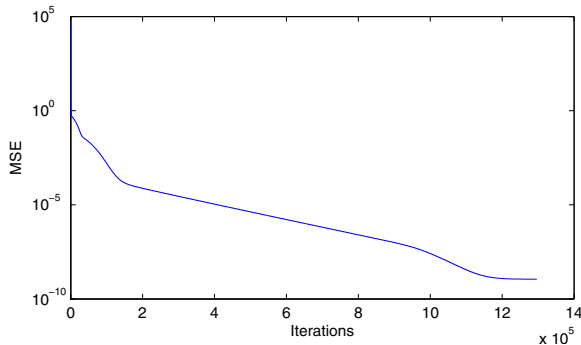

Fig. 2. Mean square error (MSE) between the actual and desired values of Laplacian eigenvalues of the 5-nodes graph

After convergence, the obtained values of the stepsizes are: $\{0.7236, 0.4198, 0.2764, 0.2165\}$. With these values, we can check that perfect factorization of average consensus matrix is obtained. The respective inverses of these values correspond to the Laplacian eigenvalues since $\alpha_1 = \frac{1}{1.382}$, $\alpha_2 = \frac{1}{2.382}$, $\alpha_3 = \frac{1}{3.618}$, $\alpha_4 = \frac{1}{4.618}$.

Figure 3 depicts the trajectories of each node for each values of λ_t, $t = 1, 2, 3, 4$. We can note that the nodes achieve consensus and the consensus values are exactly equal to Laplacian eigenvalues. Therefore,
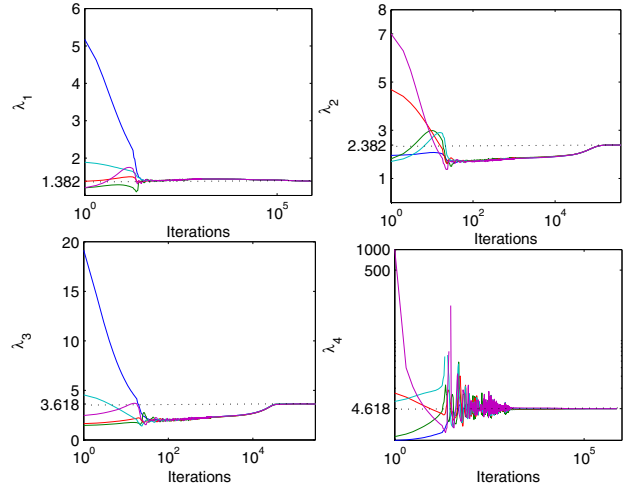

Fig. 3. Nodes trajectories converging to the nonzero distinct eigenvalues of the graph Laplacian matrix

B. *Example 2*

Now, let us consider the graph depicted in Figure 1 (right).

The associated graph Laplacian matrix $\mathbf{L}$ admits the following spectrum: $\{0, 1, 2, 3, 3, 5\}$. As for the previous example the number of factors of the average consensus matrix is $(D = 4)$. The MSE and the nodes trajectories for each value of α_t are depicted in Figures 4 and 5 respectively.
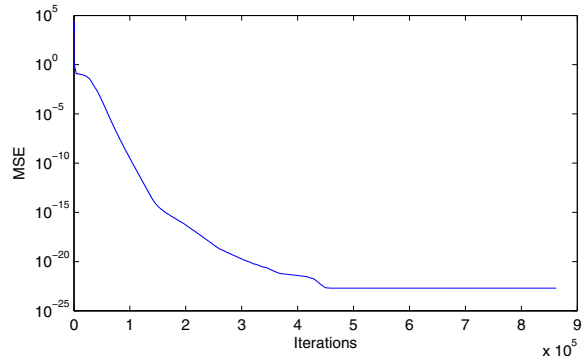

Fig. 4. Mean square error (MSE) between the actual and desired values of Laplacian eigenvalues of the 6-nodes graph

We can note that the nodes reach consensus on the inverse of Laplacian eigenvalues leading to the following factorization of the average consensus matrix:

$$(\mathbf{I}_N - 0.2\mathbf{L})(\mathbf{I}_N - 0.5\mathbf{L})(\mathbf{I}_N - \mathbf{L})(\mathbf{I}_N - 0.3333\mathbf{L}) = \frac{1}{6}\mathbf{1}\mathbf{1}^T.$$

Let us analyze the nodes trajectories during few first iterations by considering Figure 6 that depicts these trajectories α_t, $t = 1, 2, 3, 4$ (instead of the Laplacian eigenvalues illustrated in Figure 3). We can note that after starting the estimation process with random initial values the nodes attempt to reach an agreement on common values before tracking the actual inverse Laplacian eigenvalues. Such an
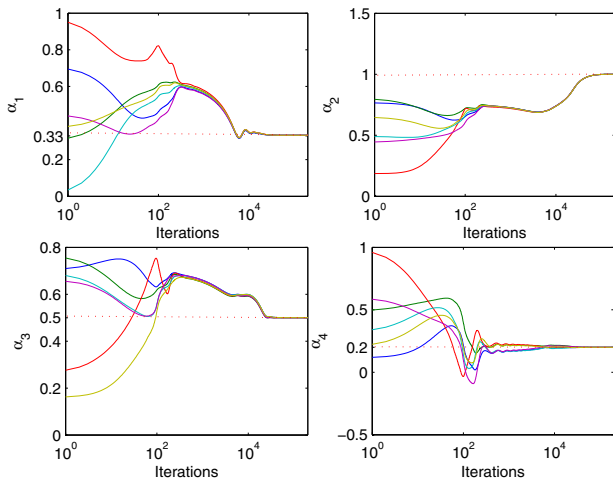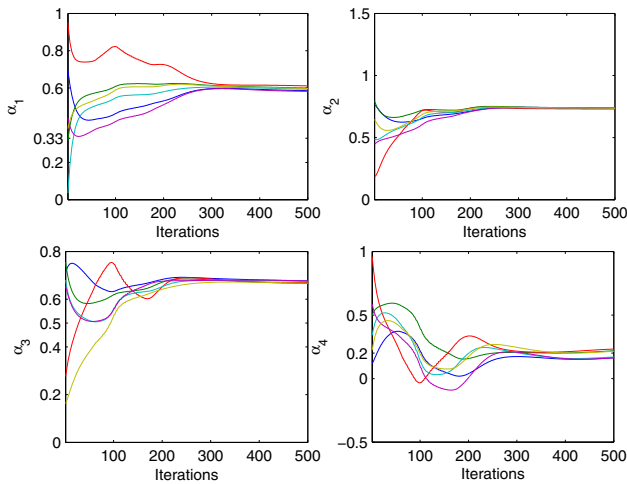
Fig. 5.   Nodes trajectories for the 6-nodes graph

Fig. 6.   First iterations of estimation of the inverses of Laplacian eigenvalues.

C. Discussion

In the simulation results presented herein we have implicitly assumed that D is known. We can make such an assumption if the graph is assumed to be distance regular or if it is a path. Indeed, for distance regular graphs D is equal to the diameter of the graph [11] while for a path with N nodes D is equal to $N-1$ (all the eigenvalues are simple). However, in general only bounds exist. More precisely, we know that $diameter(G) \leq D \leq N-1$. Both N and $diameter(G)$ can be computed in a distributed way [12]. The knowledge of the bounds on D can then be carried out without a central node. However, according to Lemma 1, we know that the factorization problem cannot be solved if D is lower than the actual number of distinct nonzero Laplacian eigenvalues. Therefore, the estimated parameters should contain the actual

values else the algorithm does not converge. Then, the question is: how retrieving the actual inverse of Laplacian eigenvalues from a larger set of estimated parameters? This question is still under investigation.

Slowness of the estimation process is also to be pointed out. However, this behavior is also observed in FFT based methods using the wave equation [7].

V. CONCLUSION

In this paper, we have proposed a way for distributively estimating Laplacian eigenvalues of an undirected graph. Based on a recent result showing that the average consensus matrix can be factored in D Laplacian based consensus matrices, where D stands for the number of nonzero distinct Laplacian eigenvalues, we have shown how carrying out such a factorization in a fully distributed way. The proposed solution results on a distributed solution of a constrained consensus problem. The efficiency of the proposed method has been evaluated by means of simulations. However several issues should be considered in future works. First, the convergence proof of the proposed method is to be stated. Second, without strict a priori knowledge of D, as in distance regular graphs, methods for retrieving the Laplacian eigenvalues in a set of estimated parameters are to be designed. Third, speed of convergence should be improved by adding memory for instance.

REFERENCES

[1] R. Merris, "Laplacian matrices of a graph: a survey." *Linear Algebra and its Applications*, vol. 197, pp. 143–176, 1994.
[2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems Control Lett.*, vol. 53, pp. 65–78, 2004.
[3] A. Y. Kibangou, "Graph Laplacian based matrix design for finite-time distributed average consensus," *Proc. of the American Control Conference (ACC)*, Montréal, Canada, 2012.
[4] ——, "Finite-time average consensus based protocol for distributed estimation over awgn channels," *Proc. of the IEEE Conference on Decision and Control (CDC)*, Orlando, Fl, USA, 2011.
[5] P. Yang, R. A. Freeman, G. Gordon, K. Lynch, S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity in mobile sensor networks," *Proc. of American Control Conference*, June 2008.
[6] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu, "Decentralized Laplacian eigenvalues estimation for networked multi-agent systems," *Proc. of joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pp. 2717–2722, Shanghai, P.R, China, 2009.
[7] T. Sahai, A. Speranzon, and A. Banaszuk, "Hearing the cluster of a graph: A distributed algorithm," *Automatica*, vol. 48(1), pp. 15–24, January, 2012.
[8] A. Y. Kibangou and C. Commault, "Decentralized laplacian eigenvalues estimation and collaborative network topology identification," *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12)*, Santa Barbara, USA 2012.
[9] L. Pescosolido, S. Barbarossa, and G. Scutari, "Average consensus algorithms robust against channel noise," in *Proc. of the IEEE Workshop SPAWC*, Recife, Brazil, 2008, pp. 261–265.
[10] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, 3rd ed., J. W. . Sons, Ed.   John Wiley & Sons, 2012.
[11] A. Brouwer, A. Cohen, and A. Neumaier, *Distance-regular graphs*. Springer, 1989.
[12] F. Garin, D. Varagnolo, and K. Johansson, E., "Distributed estimation of diameter, radius and eccentricities in anonymous networks," in *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'12)*, Santa Barbara (CA), United States, 2012, pp. 13–18.

observation implies that the stopping criterion should not be focused on the local disagreement but also on the error between the desired output $\bar{x}$ and the current one.