Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline Conference Paper Archive

IFAC PapersOnLine 56-2 (2023) 6952–6957

Distributed Estimation of the Laplacian Spectrum via Wave Equation and Distributed Optimization

Diego Deplano * **Claudia Congiu** * **Alessandro Giua** * **Mauro Franceschelli** *

* *Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy.*

Abstract: This paper presents a distributed algorithm to estimate all distinct eigenvalues of the Laplacian matrix encoding the unknown topology of a multi-agent system. The agents interact according to the discrete-time wave equation so that their state trajectory persistently oscillates with modes that depend on the eigenvalues of the Laplacian matrix. In this way, the problem of distributed estimation of the eigenvalues of the Laplacian is recast into that of estimating the modes of evolution of the state-trajectory of a linear dynamical system. Unlike previous literature, this paper formulates a distributed optimization problem where, by considering its own state trajectory, each agent estimates all distinct eigenvalues of the Laplacian matrix. The main advantages of the proposed algorithm are the ability of each agent to estimate also eigenvalues corresponding to modes unobservable from its own state trajectory, a much greater numerical stability, and therefore improved scalability to large networks wit h respect to competing approaches, as evidenced by the numerical comparisons.

1. INTRODUCTION

The need for decentralized and distributed architectures has been growing for decades in a rapidly increasing number of applications, especially those who deal with large networks of interacting autonomous agents, such as multi-robot systems [Zareh et al., 2018, Park and Yoo, 2021], smart grids and power networks [Dörfler et al., 2018, Sadamoto et al., 2019], blockchain [Zhang et al., 2020], social networks [Amelkin et al., 2017, Zhai and Zheng, 2021], sensor networks [Martínez and Bullo, 2006, Muniraju et al., 2019], to name a few.

Problem of interest and motivation. The pattern of interaction among the agents in a network strongly influences the behavior of the overall multi-agent system, and it can be effectively modeled by a graph, where nodes represent agents and edges represent point-to-point interaction or coupling links. Algebraic graph theory provides several powerful tools and the spectral properties of the Laplacian matrix associated to such graphs have emerged as pivotal in the analysis and design of interconnected systems. For instance, the spectrum of the Laplacian matrix can be used to estimate several topological properties of a graph, e.g., algebraic connectivity [Li et al., 2019, Kan et al., 2018] and Fiedler vector [Deplano et al., 2020, Doshi and Eun, 2020], min/max-cut [Dory et al., 2021], diameter and radius [Oliva et al., 2016, Deplano et al., 2021a,b], spectral gap [Vizuete et al., 2021]. Computing eigenvalues of a graph Laplacian is therefore a key problem that has been thoroughly investigated when the full state of the system is accessible from a centralized computational unit, but it becomes more challenging in the context of distributed estimation in multi-agent systems im which each agent has access only to its own state, which is the main problem addressed in this work.

Literature review. A recent branch of work uses the strategy of looking at the past history of the agents while running specific distributed protocols in order to retrieve information about the Laplacian spectrum. Charalambous et al. [2016] and Kibangou and Commault [2012] proposed distributed protocols based on the execution of a consensus protocol to enable the agents to compute the set of observable eigenvalues, where the former deals with weighted digraphs and the latter deals with undirected graphs. In all the above methods, the agents can only estimate the eigenvalues associated with the modes that are observable, and thus some agents may observe only a subset of the eigenvalues. To overcome this issue, some authors have recently investigated the strategy of recasting the problem into a distributed optimization problem [Tran and Kibangou, 2015, Zareh et al., 2018, Fan, 2017, Gusrialdi and Qu, 2020]. However, as the network becomes larger, these approaches suffer from ill-conditioning of the problem, due to the vanishing trajectories generated by the consensus dynamics. In contrast, a number of works force the agents' states to oscillate and then retrieve information about the eigenvalues via Fourier Transform strategies [Franceschelli et al., 2009, Sahai et al., 2012, Franceschelli et al., 2013]. These methods involve the estimation of the frequency at which the states' oscillates via peak-detection algorithms, which are known to be not accurate due to the occurrence of spectral leakage events.

Main contribution. In this preliminary work we propose a distributed algorithm that enables each agent to estimate

⋆ This work was partially supported by the Fondazione Banco di Sardegna with the grant "Formal Methods and Technologies for the Future of Energy Systems", cup F72F20000350007. Emails: {diego.deplano, claudia.congiu, giua,mauro.franceschelli}@unica.it
M. Franceschelli is the corresponding author.

10.1016/j.ifacol.2023.10.519

the distinct eigenvalues of the Laplacian spectrum with superior accuracy with respect to the state of the art for large networks. The idea of the approach is to make the agents interact according to the wave equation and cooperatively solve a distributed optimization problem based on their own state trajectories to estimate the modes of the persistently oscillating network. The time constants corresponding to these modes are related to the eigenvalues of the Laplacian matrix which encodes the unknown graph topology. Unlike previous literature, the proposed approach combines the wave equation method and the formulation of a distributed optimization problem for improved numerical stability in large networks.

The structure of this paper is as follows. We introduce our notations in Section II. In Section III, we present our novel algorithm to solve the distributed eigenvalue estimation problem. In Section IV numerical simulations to compare the proposed algorithm with a competing approach are provided. Finally, future directions and concluding remarks are contained in Section V.

2. NOTATION

The set of real and integer numbers are denoted by $\mathbb{R}$ and $\mathbb{Z}$, while $\mathbb{R}_{\geq 0}$, $\mathbb{N}$ and $\mathbb{R}_+$,$\mathbb{N}_+$ denote their restriction to nonnegative and positive entries, respectively. Matrices are denoted by uppercase letters and vectors by bold lowercase letters, whose entries are denoted by lowercase, nonbold symbols. For instance, $\boldsymbol{x} = [x_1, \ldots, x_n]^\top$ denotes a vector of $n \in \mathbb{N}_+$ entries $x_i \in \mathbb{R}$ with $i = 1, \ldots, n$, and $M = \{m_{ij}\}$ denotes a square matrix of dimension $n \in \mathbb{N}_+$ with entries $m_{ij} \in \mathbb{R}$ with $i, j = 1, \ldots, n$. Moreover, the identity matrix is denoted by I_n while $\mathbf{1}_n$ denotes a vector of ones of dimension $n \in \mathbb{N}_+$. When clear from the context, the subscript is omitted. Given a signal $\boldsymbol{x}(k) \in \mathbb{R}^n$ for $k \in \mathbb{N}$, we denote by $[\boldsymbol{x}]_b^a$, where $a, b \in [1, n]$ and $a \leq b$, its restriction to the interval $[a, b]$, namely $[\boldsymbol{x}]_b^a = [\boldsymbol{x}(a)^\top, \cdots, \boldsymbol{x}(b)^\top]^\top$. Moreover, for any $m \in \mathbb{N}_+$, we denote by $[X]_m^a$ the square Hankel matrix of dimension m with first entry $\boldsymbol{x}(a)$, namely

$$[X]_m^a = \begin{bmatrix} \boldsymbol{x}(a) & \cdots & \boldsymbol{x}(a+m-1) \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}(a+m-1) & \cdots & \boldsymbol{x}(2m-2+a) \end{bmatrix}.$$

2.1 Networks and Graphs

We consider multi-agent systems (MASs) consisting of $n \in \mathbb{N}_+$ interconnected agents modeled as discrete-time dynamical systems. The pattern of interactions is described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of nodes modeling the agents, and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$ is the set of edges modeling the point-to-point interactions. The interactions among the agents are assumed to be bidirectional, and therefore the graph is *undirected*, i.e., if $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$. An undirected graph $\mathcal{G}$ is said to be *connected* if between any pair of nodes $i, j \in \mathcal{V}$ there exists a path. Nodes $i, j \in \mathcal{V}$ are said to be *neighbors* if there exists an edge between them, i.e., $(i, j) \in \mathcal{E}$. The set of neighbors of the i-th node is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in E\}$. We consider graphs without self-loops, i.e., $i \notin \mathcal{N}_i$. The *degree matrix* $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix, whose diagonal elements are the degrees $|\mathcal{N}_i|$ of the nodes. The *adjacency matrix* $A = \{a_{i,j}\} \in \mathbb{R}^{n \times n}$ associated to a graph $\mathcal{G}$ is such that the entry $a_{i,j}$ is equal to 1 if there is an edge between nodes $i, j \in \mathcal{V}$, i.e., $(i, j) \in \mathcal{E}$, and it is equal to 0 otherwise. The *normalized Laplacian matrix* $L = \{\ell_{ij}\}$ is given by $L = I_n - AD^{-1}$. For undirected graphs, the normalized Laplacian matrix is symmetric, i.e., $L = L^\top$, and it satisfies the following two properties:

- The eigenvalues of L denoted by λ_i with $i \in \mathcal{V}$ are real and nonnegative, $\lambda_i \in \mathbb{R}_{\geq 0}$, and can be ordered such that $\lambda_i \leq \lambda_{i+1}$ for $i = 1, \ldots, n-1$.
- The smallest eigenvalue of L is null, $\lambda_1 = 0$, and the largest eigenvalue is upper bounded by $\lambda_n \leq 2$.

3. DISTRIBUTED SPECTRUM ESTIMATION

The problem we address is that of making each agent in the network estimate all distinct eigenvalues of the normalized Laplacian matrix $L = \{\ell_{ij}\}$ associated to $\mathcal{G}$ while only exchanging local information with its neighbors.

Such problem can be solved by Algorithm 1 given on the next page, which envisages three main steps to be performed by each agent $i \in \mathcal{V}$ in the network:

- **Step 1:** Update its own state $x_i \in \mathbb{R}$ according to the discretized wave equation,
$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k) \quad (1)$$
and keep memory of past iterations. In this way, the agents' states persistently oscillate with modes that depend on the eigenvalues of L.
- **Step 2:** Derive a data-driven model with order m of the whole interconnected system by solving the distributed optimization problem
$$\begin{aligned} \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n \in \mathbb{R}^m}{\operatorname{argmin}} \quad & \sum_{i \in \mathcal{V}} \left\| [X_i]_m^1 \boldsymbol{\theta}_i - [\boldsymbol{x}_i]_{2m}^{m+1} \right\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad \forall (i,j) \in \mathcal{E} \end{aligned}, \quad (2)$$
where matrix $[X_i]_m^1$ and vector $[\boldsymbol{x}_i]_{2m}^{m+1}$ are constructed exploiting only the knowledge of the history of the local state x_i.
- **Step 3:** Compute the eigenvalues of L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$.

In the following sections we provide a step-by-step explanation of the algorithm, showing a series of intermediate results which are instrumental to prove the main result of the paper given in Theorem 1, whose proof sketch is thus postponed to Section 3.4.

Theorem 1. *Consider a MAS with n agents interacting according to graph $\mathcal{G}$ and executing Algorithm 1. If:*

- *the graph $\mathcal{G}$ is undirected and connected;*
- *the order of the model satisfies $m \geq 2n$;*

then each agent asymptotically estimates all distinct eigenvalues of the Laplacian matrix L for almost every initial condition.

Algorithm 1: Distributed Estimation of Distinct Eigenvalues of the Laplacian Matrix

Input: Order $m \in \mathbb{N}_+$, wave speed $c \in (0, \sqrt{2})$,
design parameters $\alpha \in (0,1)$, $\rho > 0$, $\varepsilon > 0$
Init.: $x_i(1) = x_i(0) \in \mathbb{R}$ for all $i \in \mathcal{V}$
$\boldsymbol{y}_{ij}(2m) \in \mathbb{R}^m$ for all $(i,j) \in \mathcal{E}$
Output: Estimated eigenvalues $\hat{\lambda}_j$
for $k = 1, 2, 3, \dots$ **each node** $i \in \mathcal{V}$ **does**
  if $k \leq 2m-1$ **then** // Step 1
    gather $x_j(k)$ from each neighbor $j \in \mathcal{N}_i$
    $x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_i(k)$
    send $x_i(k+1)$ to each neighbor $j \in \mathcal{N}_i$
    if $k = 2m-1$ **then**
      $N_i = (2[X_i]_m^{1\,\top}[X_i]_m^1 + \rho|\mathcal{N}_i| I_m)^{-1}$,
      $\boldsymbol{v}_i = 2[X_i]_m^{1\,\top}[\boldsymbol{x}_i]_{2m}^{m+1}$
  else // Step 2
    $\boldsymbol{\theta}_i(k+1) = N_i\big(\boldsymbol{v}_i + \sum_{j \in \mathcal{N}_i} \boldsymbol{y}_{ij}(k)\big)$
    $\boldsymbol{p}_{ij} = -\boldsymbol{y}_{ij}(k) + 2\rho\boldsymbol{\theta}_i(k+1)$
    send $\boldsymbol{p}_{ij}$ to each neighbor $j \in \mathcal{N}_i$
    gather $\boldsymbol{p}_{ji}$ from each neighbor $j \in \mathcal{N}_i$
    $\boldsymbol{y}_{ij}(k+1) = (1-\alpha)\boldsymbol{y}_{ij}(k) + \alpha \boldsymbol{p}_{ji}$
    compute the roots of $r^m = \sum_{j=0}^{m-1} \theta_{i,j}(k+1) r^j$
    for each complex root $r_j \in \mathbb{C}$ // Step 3
      if $|\Re\{r_j\}^2 + \Im\{r_j\}^2 - 1| < \varepsilon$ **then**
        output $\hat{\lambda}_j = 2(1 - \Re\{r_j\})/c^2$

3.1 Step 1: Local state updates based on the discretized wave equation

Denoting with $\boldsymbol{z}(k) = \left[\boldsymbol{x}^\top(k+1)\ \boldsymbol{x}^\top(k)\right]^\top \in \mathbb{R}^{2n}$ the state of the MAS when the agents execute the wave equation (1), its dynamics can be written in compact form

$$\boldsymbol{z}(k+1) = R\boldsymbol{z}(k), \quad \text{with} \quad R = \begin{bmatrix} 2I_n - c^2 L & -I_n \\ I_n & \mathbf{0}_{n\times n} \end{bmatrix}. \quad (3)$$

We recall in the next Lemma two known results: a sufficient condition for marginal stability of the wave equation and the relation between the eigenvalues of the transition matrix R and the Laplacian matrix L (for instance, see Proposition 3.1 in [Sahai et al., 2012]).

Proposition 1. *A MAS in which the agents update their state according to the wave equation is marginally stable if*

$$c^2 \in (0,2), \quad (4)$$

with an initial condition $\boldsymbol{x}(1) = \boldsymbol{x}(0)$. If the graph and connected, the eigenvalues $r_i \in \mathbb{C}$ of R in eq. (3) are related to the eigenvalues $\lambda_i \in \mathbb{R}$ of L by

$$\Re\{r_i\} = \frac{2 - c^2\lambda_i}{2}, \quad \Im\{r_i\} = \pm\frac{c}{2}\sqrt{(4 - c^2\lambda_i)\lambda_i}. \quad (5)$$

The strategy of making the agents' state oscillate and then retrieve an estimate of the Laplacian eigenvalues has been previously adopted by other state-of-art works, such as that of Sahai *et al.* in [Sahai et al., 2012] and Franceschelli *et al.* in [Franceschelli et al., 2013]. Unlike these works, which employ a frequency domain analysis of the wave equation iteration, we aim at exploiting the history of the state trajectories of the agent executing the wave equation in order to determine a data-driven model of the MAS in a distributed way (see Section 3.2) from which the eigenvalues of the Laplacian can be derived (see Section 3.3).

We claim that adopting of the wave equation in Algorithm 1 reduces the condition number of the problem for large networks with respect to standard consensus dynamics as it has been done recently by Charalambous *et al.* in [Charalambous et al., 2016]. Indeed, as the number of agents within the network becomes larger, a wider time window of observation of the state trajectories is needed. Since the execution of the wave equation makes the agents' states persistently oscillate over time, neither vanishing nor diverging [Evans, 2010, Friedman and Tillich, 2004], it allows to observe the state trajectories for larger time windows and without loss of information if compared to the consensus dynamics. Although formal proof of this claim will be the object of future work, in this preliminary work we corroborate the claim by means of numerical simulations in Section 4.

3.2 Step 2: Distributed Data-Driven Model Identification

An equivalent representation of the MAS in state-space form is given by the following Auto-Regressive (AR) model, which specifies that the state variable $\boldsymbol{z}(k) \in \mathbb{R}^{2n}$ at time k depends linearly on its own previous values,

$$\boldsymbol{z}(k) = \theta_m^* \boldsymbol{z}(k-1) + \theta_{m-1}^* \boldsymbol{z}(k-2) + \cdots + \theta_1^* \boldsymbol{z}(k-m) \quad (6)$$

where $m \in \mathbb{N}_+$ denotes the number of past values. With this representation, usually employed in data-driven system identification [Bittanti, 2019], the eigenvalues of the transition matrix R, which defines the dynamics of the system in its state-space form (3), are a subset of the roots of the monic polynomial (6).

The following Lemma 1 shows that the coefficients of this polynomial can be computed in a distributed way by the agents by solving the optimization problem in eq. (2). In other words, the agents can cooperate to agree upon a common AR model of the system that exactly describes the dynamics of each state trajectory $x_i(k)$, without the need to know the full state $\boldsymbol{x}(k)$.

Lemma 1. *Consider a MAS in which the agents update their state according to the wave equation. If the graph is connected, the distributed optimization problem in eq.* (2) *is equivalent to the centralized optimization problem*

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \left\| [Z]_m^1 \boldsymbol{\theta} - [\boldsymbol{z}]_{2m}^{m+1} \right\|_2^2, \quad (7)$$

whose solution is the unique vector of coefficients $\boldsymbol{\theta}^* = [\theta_1^*, \cdots, \theta_m^*]^\top$ in eq. (6) *if $m \geq 2n$.*

Proof. Enumerating the signal samples in eq. (6) from step $m+1$ to step $2m$ results in

$$\underbrace{\begin{bmatrix} \boldsymbol{z}(m+1) \\ \vdots \\ \boldsymbol{z}(2m) \end{bmatrix}}_{[\boldsymbol{z}]_{2m}^{m+1}} = \underbrace{\begin{bmatrix} \boldsymbol{z}(1) & \cdots & \boldsymbol{z}(m) \\ \vdots & \ddots & \vdots \\ \boldsymbol{z}(m) & \cdots & \boldsymbol{z}(2m-1) \end{bmatrix}}_{[Z]_m^1} \underbrace{\begin{bmatrix} \theta_1^* \\ \vdots \\ \theta_m^* \end{bmatrix}}_{\boldsymbol{\theta}^*}. \quad (8)$$

From the above linear system, it is clear that $\boldsymbol{\theta}^*$ defined by the coefficients of eq. (6) is a solution to the centralized optimization problem in eq. (7). Recalling that $\boldsymbol{z}(k) = [\boldsymbol{x}^\top(k+1),\, \boldsymbol{x}^\top(k)]^\top$, we notice that the rows of the matrix $[Z]_m^1$ can be rearranged into a matrix formed by two blocks $[X]_m^1$ and $[X]_m^2$, leading to

$$\left\|[Z]_m^1\boldsymbol{\theta} - [\boldsymbol{z}]_{2m}^{m+1}\right\|_2^2 = \left\|[X]_m^1\boldsymbol{\theta} - [\boldsymbol{x}]_{2m}^{m+1}\right\|_2^2 + \left\|[X]_m^2\boldsymbol{\theta} - [\boldsymbol{x}]_{2m+1}^{m+2}\right\|_2^2$$

By construction, both the Hankel matrices $[X]_m^1$ and $[X]_m^2$ have dimension m and describe the same system of dimension $2n$. Therefore, if $m \geq 2n$, the minimizers of those terms coincide, it is unique and equal to $\boldsymbol{\theta}^*$, i.e.,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \quad \left\|[X]_m^1\boldsymbol{\theta} - [\boldsymbol{x}]_{2m}^{m+1}\right\|_2^2.$$

Furthermore, since matrix $[X]_{T-1}^{1,m}$ and vector $[\boldsymbol{x}]_T^{m+1}$ contains the entries of the state trajectory of each agent $i = 1, \ldots, n$, we can write

$$\left\|[X]_m^1 - [\boldsymbol{x}]_{2m}^{m+1}\right\|_2^2 = \sum_{i\in\mathcal{V}} \left\|[X_i]_m^1\boldsymbol{\theta} - [\boldsymbol{x}_i]_{2m}^{m+1}\right\|_2^2,$$

from which the equivalence between eqs. (7)-(2) follows. □

Some important remarks are in order.

Remark 1. *Due to Lemma 1, when the agents solve the distributed optimization problem in eq.* (2)*, they estimate a common model by achieving consensus on the one that better approximates all the local state trajectories, thus only exploiting the local information of their own state trajectory and not the global information of the whole system trajectory. Moreover, since each mode of the system is observable from at least one agent in the network, then all the modes are encoded into the estimated model in such a distributed way.*

Remark 2. *Based on the methods used, it is reasonable to believe that the estimates generated by Algorithm 1 are unbiased in the presence of noise or disturbances with zero means. Furthermore, these estimates are expected to improve over time, particularly for low values of the variances.*

Remark 3. *While executing Algorithm 1, each agent $i \in \mathcal{V}$ needs to store matrix $N_i \in \mathbb{R}^{m\times m}$ and vector $\boldsymbol{v}_i \in \mathbb{R}^m$ given in eq.* (11)*, where it is assumed $m \geq 2n$. However, numerical simulations highlighted that smaller values of m could be sufficient, for instance in the case of eigenvalues with algebraic multiplicity greater than one. A more detailed analysis is left for future investigation.*

Among the several methods proposed in the current literature to solve problem (2), in this work, we resort to the distributed relaxed version of the alternating direction method of multipliers (R-ADMM) formalized by Bastianello *et al.* in [Bastianello et al., 2020].

Lemma 2. *The R-ADMM applied to the distributed optimization problem in eq.* (2) *over a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is characterized by the following updates,*

$$\boldsymbol{\theta}_i(k+1) = N_i\big(\boldsymbol{v}_i + \sum_{j\in\mathcal{N}_i} \boldsymbol{y}_{ij}(k)\big) \tag{9}$$

$$\boldsymbol{y}_{ij}(k+1) = (1-\alpha)\boldsymbol{y}_{ij}(k) + \alpha \boldsymbol{p}_{ji} \tag{10}$$

where $\boldsymbol{p}_{ij} = -\boldsymbol{y}_{ij}(k) + 2\rho\boldsymbol{\theta}_i(k+1)$ are messages sent from agent i to agent j if $(i,j) \in \mathcal{E}$, $\boldsymbol{y}_{ij}$ are auxiliary local variables, $\alpha \in (0,1)$, $\rho > 0$ are design parameters, and

$$\begin{aligned} N_i &= (2[X_i]_m^{1\ \top}[X_i]_m^1 + \rho|\mathcal{N}_i|I_m)^{-1}, \\ \boldsymbol{v}_i &= 2[X_i]_m^{1\ \top}[\boldsymbol{x}_i]_{2m}^{m+1}. \end{aligned} \tag{11}$$

Proof. The optimization problem in eq. (2) is of the type

$$\begin{aligned} \min_{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n} \quad & \sum_{i\in\mathcal{V}} f_i(\boldsymbol{\theta}_i) \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad \forall (i,j) \in \mathcal{E} \end{aligned},$$

Where the costs $f_i(\theta_i) = \left\|[X_i]_m^1\boldsymbol{\theta}_i - [\boldsymbol{x}_i]_{2m}^{m+1}\right\|_2^2$ are strongly convex. Following [Bastianello et al., 2020], the R-ADMM depends on two design parameters $\alpha \in (0,1)$ and $\rho > 0$ and it is characterized by the updates of the auxiliary variables $\boldsymbol{y}_{ij}$ as in eq. (10) and the updates of $\boldsymbol{\theta}_i$ are given by the following minimization problem

$$\boldsymbol{\theta}_i(k+1) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ f_i(\boldsymbol{\theta}) - \sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}^\top \boldsymbol{y}_{ij}(k) + \frac{\rho|\mathcal{N}_i|}{2}\|\boldsymbol{\theta}\|_2^2 \right\}$$

whose closed-form solution is computed as detailed next. Let $g(\boldsymbol{\theta})$ be the function to be minimized, i.e.,

$$g(\boldsymbol{\theta}) = \left\|[X_i]_m^1\boldsymbol{\theta}_i - [\boldsymbol{x}_i]_{2m}^{m+1}\right\|_2^2 - \sum_{j\in\mathcal{N}_i} \boldsymbol{\theta}^\top \boldsymbol{y}_{ij}(k) + \frac{\rho|\mathcal{N}_i|}{2}\|\boldsymbol{\theta}\|_2^2$$

The minimizer of $g(\boldsymbol{\theta})$ is obtained by finding the zeros of its gradient, $\nabla g(\boldsymbol{\theta}) =$

$$\underbrace{(2[X_i]_m^{1\ \top}[X_i]_m^1 + \rho|\mathcal{N}_i|I_m)}_{N_i^{-1}}\boldsymbol{\theta} - \underbrace{2[X_i]_m^{1\ \top}[\boldsymbol{x}_i]_{2m}^{m+1}}_{\boldsymbol{v}_i} + \sum_{j\in\mathcal{N}_i} \boldsymbol{y}_{ij}(k))$$

from which the thesis follows. □

3.3 Step 3: Eigenvalue estimation procedure

Let $\boldsymbol{\theta}^*$ be the solution to the distributed optimization problem in (2) and consider the polynomial whose coefficients are the entries of $\boldsymbol{\theta}^*$, namely,

$$r^m - \theta_m^* r^{m-1} - \theta_{m-1}^* r^{m-2} - \ldots - \theta_2^* r - \theta_1^*. \tag{12}$$

In Lemma 3 we provide a method for the agents to discriminate among these roots those that correspond to the eigenvalues of the system.

Lemma 3. *Consider a MAS in which the agents update their state according to the wave equation and let $\boldsymbol{\theta}^*$ be the solution to the distributed optimization problem in eq.* (2) *with $m \geq 2n$. Then a root $r \in \mathbb{C}$ of the monic polynomial in eq.* (12) *defined by the coefficients of $\boldsymbol{\theta}^*$ is an eigenvalue of the system in eq.* (3) *only if*

$$\Re\{r\}^2 + \Im\{r\}^2 = 1 \tag{13}$$

Proof. It is a direct consequence of Proposition 1 and Lemma 1.

Remark 4. *We believe that the condition* (13) *of Lemma 3 is not only necessary but also sufficient. This is consistent with the simulation results, but a formal proof is currently missing.*

3.4 Proof sketch of Theorem 1

The agents update their state $x_i(k)$ according to the wave equation in eq. (1) up to step $k = 2m - 1$, thus generating non vanishing/diverging trajectories according to Proposition 1. Afterwards, for $k \geq 2m$ the agents solve the distributed optimization problem in eq. (2) via R-ADMM algorithm by the iterations in eqs. (9)-(10). For $k \to \infty$ the agents converge linearly to the optimal solution $\boldsymbol{\theta}^*$, see [Bastianello et al., 2020, Propositions 1-2], which, according to Lemma 1, uniquely defines the monic polynomial in eq. (6) if $m \geq 2n$. Thus, the roots of the polynomial include all the distinct eigenvalues of system (3), but also additional roots when the order of the system is not precisely known, i.e., when $m > 2n$. Finally, Lemma 3 shows that the roots of the polynomial in eq. (12) that corresponds to an eigenvalue of the system must satisfy eq. (13), which thus constitutes a criterion for the agents to discriminate the useful roots. The proof is completed by remarking that from the eigenvalues of the system one can infer the eigenvalues of the Laplacian matrix according to eq. (5).

4. NUMERICAL SIMULATIONS

In this section, we show numerical simulations corroborating the effectiveness of Algorithm 1 and a comparison with the algorithm proposed by Charalambous *et al.* [Charalambous et al., 2016]. All the simulations are run within Matlab environment, using variable precision floating point arithmetic with 32 decimal digit accuracy.

4.1 Example 1: Unobservable eigenvalues

Consider a network of $n = 7$ agents interacting according to an undirected graph with line topology. The set of the eigenvalues $\Lambda = \{\lambda_i\}$ of the Laplacian matrix L is

$$\Lambda = \{0,\ 0.134,\ 0.5,\ 1,\ 1.5,\ 1.866,\ 2\}.$$

In this setup, the system is fully observable from the agents at the periphery of the network, but it is only partially observable from the central node, which we call node $i^* = 4$.

The algorithm proposed by Charalambous *et al* in [Charalambous et al., 2016] allows each agent to estimate only a number of eigenvalues equal to the rank of the observability matrix computed from that agent. Therefore, the agent i^* is able to estimate a subset $\Lambda^* \subset \Lambda$ of only 4 eigenvalues, that is $\Lambda^* = \{0,\ 0.5,\ 1.5,\ 2\}$. In contrast, by means of our Algorithm 1, agent i^* is able to estimate all the eigenvalues. In particular, having chosen the inputs according to

$$c = \sqrt{2},\ \ \alpha = 0.99,\ \ \rho = 10,\ \ m = 20,\ \ \varepsilon = 10^{-6}, \qquad (14)$$

the agents asymptotically agree upon the vector of coefficients

$$\boldsymbol{\theta}^* = \begin{matrix}[340,\ -359,\ 376,\ 918,\ 24.3,\ 24.6,\ 24.9,\ 25.1, \\ 25.1,\ 25.0,\ 24.8,\ 24.5,\ 296,\ 276,\ 255\,] \cdot 10^{-3}\end{matrix}$$

whose roots are

$$\begin{aligned} r_1 &= -0.990 \pm j0.141, & r_2 &= -0.857 \pm j0.516 \\ r_3 &= -0.492 \pm j0.870, & r_4 &= +0.005 \pm j1.000 \\ r_5 &= +0.503 \pm j0.865, & r_6 &= +0.867 \pm j0.499 \\ r_7 &= +0.338 \pm j0.475, & r_8 &= +1.000 \pm j0.000 \end{aligned}$$

By means of eq. (13), each agent can discriminate the roots that correspond to an eigenvalue of the Laplacian matrix L, indeed, the only root that does not meet eq. (13) is r_7, indeed, $\Re\{r_7\}^2 + \Im\{r_7\}^2 = 0.34 \ll 1$. On the other hand, the other roots r_j with $j \neq 7$ meet eq. (13) up to an error equal to $\varepsilon = 10^{-6}$ and lead to the eigenvalues of L by $\hat{\lambda}_j = 2(1 - \Re\{r_j\})/c^2$,

$$\begin{aligned} &\hat{\lambda}_1 = 2, \quad \hat{\lambda}_2 = 1.866, \quad \hat{\lambda}_3 = 1.5, \quad \hat{\lambda}_4 = 1, \\ &\hat{\lambda}_5 = 0.5, \quad \hat{\lambda}_6 = 0.134, \quad \hat{\lambda}_8 = 0, \end{aligned}$$

and the mean square error in the estimation is $\approx 10^{-15}$.

4.2 Example 2: Scalability for large-networks

In this section we run Algorithm 1 and Algorithm 3 in [Charalambous et al., 2016] over networks with line topology by increasing the number of agents $n = 3, 4, \ldots, 19, 20, 30, 40, \ldots, 100$. We consider 10 different instances of each problem with different initial conditions, and show in Figs. 1-2 the average results. Differently from Example 1, we consider an agent at the periphery of the line graph, called i^*, from which all the modes of the system are observable. If agents are numbered in ascending order according to the line topology, then $i^* = 1$. With this setup, the comparison is more fair since both algorithms theoretically allow agent i^* to estimate all the eigenvalues of the Laplacian matrix. If agents are numbered in ascending order according to the line topology, then $i^* = 1$.

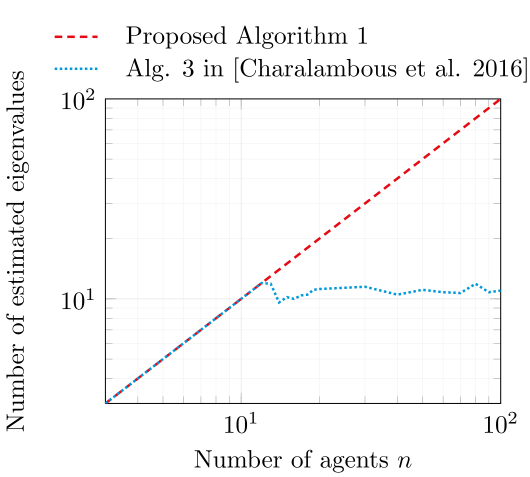

Fig. 1. Number of estimated eigenvalue estimation in line networks of different size

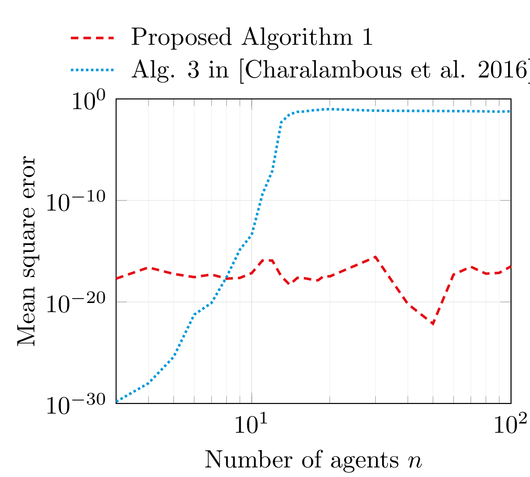

Fig. 2. Mean square error of eigenvalue estimation in line networks of different size.

Figs. 1-2 show that when Algorithm 1 is executed by the MAS with input parameters as in eq. (14), with a large bound $m = 5n$, agent i^* estimates all the eigenvalues of the Laplacian matrix for any network size and with a small mean square error $\approx 10^{-16}$ that does not increase with the size of the network. On the other hand, Figs. 1-2 show that when algorithm in [Charalambous et al., 2016] is executed by the MAS, agent i^* can compute all the eigenvalues of the Laplacian matrix only for small networks, namely $n \leq 12$. For larger networks, $n > 12$, node i^* fails in computing all the eigenvalues, meaning that some of the eigenvalues are not estimated at all and some eigenvalues are estimated with large errors. In particular, in our experiments, the agent i^* was not able to estimate more than 12 eigenvalues regardless of the size of the network, and the error made in such estimation grows up to ≈ 0.1.

5. CONCLUSIONS

In this paper, we have proposed a novel protocol for the distributed estimation of the eigenvalues of the Laplacian matrix in undirected networks. The protocol allows each agent to estimate all the eigenvalues with high accuracy and it is scalable for large networks due to the linear growth of the size of locally exchanged messages with respect to the size of the network. Future research directions include the generalization of the proposed protocol to a real-time scenario, where the agents estimate the eigenvalues when the number of agents and their interconnections may vary over time.

REFERENCES

V. Amelkin, F. Bullo, and A. K. Singh. Polar opinion dynamics in social networks. *IEEE Transactions on Automatic Control*, 62(11):5650–5665, 2017.

N. Bastianello, R. Carli, L. Schenato, and M. Todescato. Asynchronous distributed optimization over lossy networks via relaxed admm: Stability and linear convergence. *IEEE Transactions on Automatic Control*, 66(6):2620–2635, 2020.

S. Bittanti. *Model identification and data analysis*. John Wiley & Sons, 2019.

T. Charalambous, M. G. Rabbat, M. Johansson, and C. N. Hadjicostis. Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum. *IEEE Transactions on Control of Network Systems*, 3 (2):137–148, 2016.

D. Deplano, M. Franceschelli, A. Giua, and L. Scardovi. Distributed fiedler vector estimation with application to desynchronization of harmonic oscillator networks. *IEEE Control Systems Letters*, 5(2):659–664, 2020.

D. Deplano, M. Franceschelli, and A. Giua. Dynamic min and max consensus and size estimation of anonymous multi-agent networks. *IEEE Transactions on Automatic Control*, pages 1–1, 2021a.

D. Deplano, M. Franceschelli, and A. Giua. Distributed tracking of graph parameters in anonymous networks with time-varying topology. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6258–6263. IEEE, 2021b.

F. Dörfler, J. W. Simpson-Porco, and F. Bullo. Electrical networks and algebraic graph theory: Models, properties, and applications. *Proceedings of the IEEE*, 106(5):977–1005, 2018.

M. Dory, Y. Efron, S. Mukhopadhyay, and D. Nanongkai. Distributed weighted min-cut in nearly-optimal time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1144–1153, 2021.

V. Doshi and D. Y. Eun. Fiedler vector approximation via interacting random walks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–28, 2020.

L. C. Evans. *Partial differential equations*, volume 19. American Mathematical Soc., 2010.

L. Fan. Data fusion-based distributed prony analysis. *Electric Power Systems Research*, 143:634–642, 2017.

M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu. Decentralized laplacian eigenvalues estimation for networked multi-agent systems. In *48th IEEE Conference on Decision and Control (CDC)*. IEEE, 2009.

M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu. Decentralized estimation of laplacian eigenvalues in multi-agent systems. *Automatica*, 49, 2013.

J. Friedman and J.-P. Tillich. Wave equations for graphs and the edge-based laplacian. *Pacific Journal of Mathematics*, 216(2):229–266, 2004.

A. Gusrialdi and Z. Qu. Data-driven distributed algorithms for estimating eigenvalues and eigenvectors of interconnected dynamical systems. *IFAC-PapersOnLine*, 53(2):52–57, 2020.

Z. Kan, E. A. Doucette, and W. E. Dixon. Distributed connectivity preserving target tracking with random sensing. *IEEE Transactions on Automatic Control*, 64(5):2166–2173, 2018.

A. Y. Kibangou and C. Commault. Decentralized laplacian eigenvalues estimation and collaborative network topology identification. *IFAC Proceedings Volumes*, 45, 2012.

C. Li, Z. Qu, D. Qi, and F. Wang. Distributed finite-time estimation of the bounds on algebraic connectivity for directed graphs. *Automatica*, 107:289–295, 2019.

S. Martínez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4): 661–668, 2006.

G. Muniraju, C. Tepedelenlioglu, and A. Spanias. Distributed spectral radius estimation in wireless sensor networks. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1506–1510. IEEE, 2019.

G. Oliva, R. Setola, and C. N. Hadjicostis. Distributed finite-time calculation of node eccentricities, graph radius and graph diameter. *Systems & Control Letters*, 92:20–27, 2016.

B. S. Park and S. J. Yoo. Connectivity-maintaining and collision-avoiding performance function approach for robust leader–follower formation control of multiple uncertain underactuated surface vessels. *Automatica*, 127:109501, 2021.

T. Sadamoto, A. Chakrabortty, T. Ishizaki, and J.-i. Imura. Dynamic modeling, stability, and control of power systems with distributed energy resources: Handling faults using two control methods in tandem. *IEEE Control Systems Magazine*, 39(2):34–65, 2019.

T. Sahai, A. Speranzon, and A. Banaszuk. Hearing the clusters of a graph: A distributed algorithm. *Automatica*, 48(1):15–24, 2012.

T.-M.-D. Tran and A. Y. Kibangou. Distributed estimation of laplacian eigenvalues via constrained consensus optimization problems. *Systems & Control Letters*, 80, 2015.

R. Vizuete, F. Garin, and P. Frasca. The laplacian spectrum of large graphs sampled from graphons. *IEEE Transactions on Network Science and Engineering*, 8(2):1711–1721, 2021.

M. Zareh, L. Sabattini, and C. Secchi. Distributed laplacian eigenvalue and eigenvector estimation in multi-robot systems. In *Distributed autonomous robotic systems*, pages 191–204. Springer, 2018.

S. Zhai and W. X. Zheng. Dynamic behavior for social networks with state-dependent susceptibility and antagonistic interactions. *Automatica*, 129:109652, 2021.

Z. Zhang, M. Zargham, and V. M. Preciado. On modeling blockchain-enabled economic networks as stochastic dynamical systems. *Applied Network Science*, 5(1):1–24, 2020.