

# Data-Driven Distributed Algorithms for Estimating Eigenvalues and Eigenvectors of Interconnected Dynamical Systems

Azwirman Gusrialdi\* Zhihua Qu\*\*

\* *Automation Technology and Mechanical Engineering Unit, Faculty of  
Engineering and Natural Sciences, Tampere University, Tampere  
33014 Finland (e-mail: azwirman.gusrialdi@tuni.fi).*

\*\* *Department of Electrical and Computer Engineering, University of  
Central Florida, Orlando 32816 USA. (e-mail: qu@ucf.edu)*

---

**Abstract:** The paper presents data-driven algorithms to estimate in a distributed manner the eigenvalues, right and left eigenvectors of an unknown linear (or linearized) interconnected dynamic system. In particular, the proposed algorithms do not require the identification of the system model in advance before performing the estimation. As a first step, we consider interconnected dynamical system with distinct eigenvalues. The proposed strategy first estimates the eigenvalues using the well-known Prony method. The right and left eigenvectors are then estimated by solving distributively a set of linear equations. One important feature of the proposed algorithms is that the topology of communication network used to perform the distributed estimation can be chosen arbitrarily, given that it is connected, and is also independent of the structure or sparsity of the system (state) matrix. The proposed distributed algorithms are demonstrated via a numerical example.

*Keywords:* distributed estimation, data-driven algorithm, eigenvalues and eigenvectors

---

## 1. INTRODUCTION

Eigenvalues and the corresponding eigenvectors of an interconnected dynamical system play an important role in analyzing and controlling the dynamical system. Let us take power system as an example. One of the most critical wide-area monitoring applications in power system is the low-frequency inter-area oscillation involving two coherent generator groups swinging against each other which may lead to a small-signal stability concern for modern inter-connected power systems and thus needs to be constantly monitored and controlled (Chow, 2012). Small signal stability analysis (i.e., the behavior of the system linearized around an operating point) using modal techniques is a widely used method to study and control inter-area oscillation. In particular, the eigenvalues of the linearized dynamical model show the frequency and damping of the oscillations. Moreover, the right and left eigenvectors provide information about the observability and controllability of the oscillations respectively while their combination indicates the location of the controllers to damp the undesired oscillations (Martins and Lima, 1990; Rouco, 1998). In addition to power system, the left eigenvector of a Laplacian matrix also plays an important role in designing cooperative control algorithm for a network of heterogeneous nonlinear systems as discussed in (Qu and Simaan, 2014).

In practice, the global topology or overall dynamics of a network (interconnected system) is typically not available (for example due to privacy issue (McDaniel and McLaughlin, 2009)) and as a result, the eigenvalues and

eigenvectors cannot be computed directly. To overcome this issue, various *distributed* algorithms have been proposed in the literature to estimate the eigenvalues and/or eigenvectors of a matrix using only local information available to individual subsystems (Gusrialdi and Qu, 2017; Franceschelli et al., 2013; Kibangou and Commault, 2012; Tran and Kibangou, 2015; Yang and Tang, 2015; Charalambous et al., 2016). Even though the proposed algorithms allow distributed estimation, those work still assume that the (local) system model is available for performing the distributed estimation. However, the system model (i.e., system (state) matrix in dynamical system) is often unknown or not available due to geographical constraint, it may change due to perturbation or simply because it is too complicated to obtain as observed in power system (Gusrialdi et al., 2019). This motivates the development of data-driven (distributed) algorithm to estimate the eigenvalues and eigenvectors of unknown (linear or linearized) dynamical systems. Data-driven centralized algorithms using principal components and maximum likelihood methods to estimate dominant eigenvalues of a dynamical system are proposed in (Petrie and Zhao, 2012) under assumption that only a few of eigenvalues are dominant. Data-driven distributed algorithms based on Prony method are proposed in (Nabavi et al., 2015; Khazaei et al., 2016) to estimate the eigenvalues of power system model. However, the proposed algorithms are geared towards estimating eigenvalues only; they cannot estimate eigenvectors. Power iteration allows distributed estimation of the greatest eigenvalue (in absolute value) of a dynamical system together with the associated right eigenvector

using only available measurements or data under certain conditions (Golub and Van Loan, 1996; Gusrialdi et al., 2019). However, it is not clear if the method can be extended to estimate the left eigenvectors and to deal with complex eigenvalues and eigenvectors using only available data. Recently, the work (Gusrialdi et al., 2018) proposed a distributed algorithm to estimate the eigenvalues together with the right eigenvectors of a power system model. The idea is by first learning in a distributed manner the system model and compute the eigenvalues together the right eigenvectors using the learned system model. However, the approach heavily depends on the accuracy of the learned dynamical model and the eigenvalues and eigenvectors estimation are sensitive to the identification error.

The paper proposes data-driven algorithms to estimate in a distributed manner the eigenvalues, right and left eigenvectors of an unknown linear (or linearized) interconnected dynamical system. In contrast to the related work (Gusrialdi et al., 2018), the eigenvalues and corresponding eigenvectors are estimated directly from data and without requiring identification of the system model in advance. As a first step, in the paper we consider interconnected dynamical system with distinct eigenvalues. To this end, the eigenvalues are first estimated using the well-known Prony method. The right and left eigenvectors are then estimated by solving distributively a set of linear equations. One important feature of the proposed algorithms is that the communication network topology used to perform the distributed estimation can be chosen arbitrarily, given that it is connected. Furthermore, the structure of the communication network is also independent of the structure or sparsity of the system (state) matrix.

The paper is organized as follows. Data-driven distributed eigenvalue and eigenvector estimation problem is formulated in Section 2. The proposed algorithms to estimate distributively the eigenvalues together with both the left and right eigenvectors of linear (or linearized) dynamical system with unknown system (state) matrix are described in Section 3. The proposed distributed algorithms are demonstrated using a numerical example in Section 4. Concluding remarks and future work are presented in Section 5.

## 2. PROBLEM FORMULATION

In this section, we first introduce notations used in the paper followed by a brief overview of graph theory and the problem formulation.

### 2.1 Notation and Preliminary

For a complex number  $a$ , let  $\Re\{a\}$  and  $\Im\{a\}$  denote its real and imaginary parts respectively. The identity matrix of size  $n$  is denoted by  $I_n$ . For a matrix  $A \in \mathbb{R}^{n \times n}$ , let  $[A]_{i*} \in \mathbb{R}^n$  and  $[A]_{*i} \in \mathbb{R}^n$  represent vectors whose elements are equal to the  $i$ -th row and column of  $A$  respectively. Let  $\lambda_i(A)$  denote the eigenvalues of matrix  $A$ . Furthermore, let  $\nu_i$  and  $w_i$  respectively denote the left and right eigenvectors of  $A$  associated with the eigenvalues  $\lambda_i(A)$ . Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be a graph with a set of nodes  $\mathcal{V} = \{1, 2, \dots, n\}$  and a set of edges  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ . An edge  $(i, j) \in \mathcal{E}$  denotes that node  $i$  can obtain information from node  $j$ . The set of neighbors of node  $i$  is denoted by

$\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$ . A graph is undirected if the edges are bidirectional, that is,  $i \in \mathcal{N}_j \Leftrightarrow j \in \mathcal{N}_i$ . An undirected graph is connected if there is no isolated nodes (Qu, 2009).

### 2.2 Problem Statement

Consider a network of  $n$  (physically) interconnected nodes or subsystems whose overall dynamics is given by

$$\dot{x} = Ax, \quad (1)$$

where  $x = [x_1, \dots, x_n]^T$  denotes the state of the overall dynamical system with  $x_i \in \mathbb{R}$  represents the state of the  $i$ -th node. Even though we assume that  $x_i$  is scalar, the results of the paper can be extended in a straightforward manner to the case where state  $x_i$  is a vector. It is assumed that matrix  $A$  in (1) is unknown and its eigenvalues are distinct. Furthermore, we also assume that matrix  $A$  is Lyapunov stable and thus  $x$  is bounded. The  $i$ -th node has access only to its own sampled state  $x_i(k) \triangleq x_i(t)|_{t=kT}$ , ( $k = 0, 1, \dots$ ) where  $T$  denotes the sampling time, corresponding to discrete-time model of (1) given by

$$x(k+1) = A_d x(k) \quad (2)$$

where  $A_d = e^{AT}$ . It is also assumed that the subsystems can communicate (i.e., exchange information) with some other nodes in the network, denoted by  $\mathcal{N}_i$ , via the communication network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  whose topology is given by a connected undirected graph. It should be noted that the communication network topology  $\mathcal{G}$  is independent of the sparsity or structure of matrix  $A$  in (1). Our objective is to solve the following problem.

*Problem 1.* Assume that matrix  $A$  is unknown and given  $x_i(k)$  for  $k = \{0, 1, \dots\}$  available to the  $i$ -th subsystem together with a communication network whose topology is associated with a connected undirected graph, estimate in a distributed manner all the eigenvalues of  $A$  together with the corresponding left and right eigenvectors.

## 3. MAIN RESULT

First, observe that the relationship between the eigenvalues of matrices  $A$  and  $A_d$  is given by

$$\lambda_i(A) = \frac{\ln(\lambda_i(A_d))}{T} \quad (3)$$

and dynamics (1) and (2) share the same left and right eigenvectors. Since matrix  $A$  has distinct eigenvalues, we can write the solution to (1) as

$$x(t) = \sum_{i=1}^n \nu_i^T x(0) e^{\lambda_i(A)t} w_i. \quad (4)$$

In addition, let us define matrices  $W$  and  $V$  as

$$W = \begin{bmatrix} w_{1,1} & \cdots & w_{n,1} \\ \vdots & \vdots & \vdots \\ w_{1,n} & \cdots & w_{n,n} \end{bmatrix}, V = \begin{bmatrix} \nu_{1,1} & \cdots & \nu_{1,n} \\ \vdots & \vdots & \vdots \\ \nu_{n,1} & \cdots & \nu_{n,n} \end{bmatrix} \quad (5)$$

where  $w_{i,j}$  (resp.  $\nu_{i,j}$ ) denotes the  $j$ -th element of the vector  $w_i$  (resp.  $\nu_i$ ). From  $Aw_i = \lambda_i w_i$  and  $A^T \nu_i = \lambda_i \nu_i$  we have the following relationship

$$V = W^{-1}. \quad (6)$$

The proposed distributed algorithms to solve Problem 1 are summarized as follows:

- (1) Each node estimates distributively (i.e., cooperatively) the eigenvalues  $\lambda_i(A)$  for  $i = 1, \dots, n$  using Prony method
- (2) Each node then estimates the right eigenvector  $w_i$  by solving (4) in a distributed fashion, given the estimated eigenvalues  $\lambda_i(A)$
- (3) Each node finally estimates distributively the left eigenvector  $\nu_i$  from (6)

Details of each step will be described in the following subsections. For the sake of simplicity and clarity, in the remaining of the paper we assume that all the eigenvalues are real. However, the proposed strategy can also be extended in a straightforward manner to the case of complex eigenvalues as will be demonstrated in Section 4.

### 3.1 Distributed Eigenvalue Estimation

Each node first distributively estimates all the eigenvalues  $\lambda_i(A)$  using distributed Prony method proposed in the literature, e.g. (Fan, 2017). For the sake of completeness, in this subsection we provide a summary of distributed Prony method to estimate  $\lambda_i(A)$  from the data series  $x_i(k)$ . First, the solution (4) can be written as

$$x(t) = \sum_{i=1}^n R_i x(0) e^{\lambda_i(A)t}$$

where  $R_i = w_i \nu_i^T$  is a residue matrix. Considering the sampled state  $x(k)$ , the above equation can be recasted in the following form

$$x(k) = \sum_{i=1}^n R_i x(0) z_i^k, k = 1, \dots, N \quad (7)$$

where  $N$  is the number of samples. Moreover,  $z_i = e^{\lambda_i(A)T} = \lambda_i(A_d)$  are eigenvalues of discrete-time model and thus the roots of the following characteristic polynomial function of the system

$$z^n - (a_1 z^{n-1} + a_2 z^{n-2} + \dots + a_n z^0) = 0. \quad (8)$$

Hence, if the coefficients  $a_i$  in (8) can be computed from the sampled state  $x(k)$ , the values  $z_i$  can then be computed by finding the roots of (8) and as a result the eigenvalues  $\lambda_i(A)$  can be computed from (3). Substituting (8) into (7) for  $k = n$  results in the following linear prediction model

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + \dots + a_n x(0). \quad (9)$$

Furthermore, from (9) and by enumerating the signal samples from steps  $n$  to  $N$  yield

$$\underbrace{\begin{bmatrix} x(n-1) & \dots & x(0) \\ \vdots & \vdots & \vdots \\ x(N-1) & \dots & x(N-n) \end{bmatrix}}_H \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}}_{\bar{a}} = \underbrace{\begin{bmatrix} x(n) \\ \vdots \\ x(N) \end{bmatrix}}_Y. \quad (10)$$

Hence, the vector  $\bar{a}$  can be computed from (10) by solving the following least square (LS) problem:

$$\min_{\bar{a}} \frac{1}{2} \|H\bar{a} - Y\|^2. \quad (11)$$

In order to solve (11) in a distributed manner, the set of equations (10) can be rewritten as

$$\begin{bmatrix} H_1 \\ \vdots \\ H_n \end{bmatrix} \bar{a} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$$

where

$$H_i = \begin{bmatrix} x_i(n-1) & \dots & x_i(0) \\ \vdots & \vdots & \vdots \\ x_i(N-1) & \dots & x_i(N-n) \end{bmatrix}, Y_i = \begin{bmatrix} x_i(n) \\ \vdots \\ x_i(N) \end{bmatrix}$$

are locally known to the  $i$ -th node. Hence, LS problem can be reformulated as the following distributed optimization

$$\begin{aligned} & \text{minimize}_{\bar{a}_1, \dots, \bar{a}_n} \sum_{i=1}^n \frac{1}{2} \|H_i \bar{a}_i - Y_i\|^2 \\ & \text{subject to } \bar{a}_1 = \dots = \bar{a}_n \end{aligned} \quad (12)$$

with  $\bar{a}_i \in \mathbb{R}^n$  denotes the estimation of  $\bar{a}$  at the  $i$ -th subsystem. Given that the communication network topology is connected, optimization (12) can be solved distributively using the standard distributed optimization algorithms developed in the literature, for example the one combining the gradient and consensus algorithms (Khazaei et al., 2016; Yang et al., 2019) or the one based on distributed alternating direction method of multipliers (Nabavi et al., 2015). After solving (12) distributively, each node knows  $\bar{a}$  and thus it can compute all the eigenvalues  $\lambda_i(A)$  from (8) and (3).

### 3.2 Distributed Right Eigenvector Estimation

After each node distributively estimates  $\lambda_i(A)$ , it then estimates the right eigenvectors  $w_i$ . Note that since  $(\nu_i^T x(0))$  is scalar, vector  $(\nu_i^T x(0))w_i$  is also the right eigenvector of matrix  $A$  w.r.t.  $\lambda_i(A)$  and for simplicity is also denoted by  $w_i$ . Hence, (4) can be written as

$$x(t) = \sum_{i=1}^n e^{\lambda_i(A)t} w_i$$

or similarly for discrete time system from (7) we can write the sampled state as

$$x(k) = \sum_{i=1}^n w_i z_i^k, z_i = e^{\lambda_i(A)T}. \quad (13)$$

Given  $n$  number of samples, from (13) we can write for the  $i$ -th node (i.e., the  $i$ -th linear equation in (13))

$$\underbrace{\begin{bmatrix} z_1^{k_1^i} & z_2^{k_1^i} & \dots & z_n^{k_1^i} \\ z_1^{k_2^i} & z_2^{k_2^i} & \dots & z_n^{k_2^i} \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{k_n^i} & z_2^{k_n^i} & \dots & z_n^{k_n^i} \end{bmatrix}}_{\Omega_i} \underbrace{\begin{bmatrix} w_{1,i} \\ w_{2,i} \\ \vdots \\ w_{n,i} \end{bmatrix}}_{\tilde{w}_i} = \underbrace{\begin{bmatrix} x_i(k_1^i) \\ x_i(k_2^i) \\ \vdots \\ x_i(k_n^i) \end{bmatrix}}_{\tilde{x}_i}. \quad (14)$$

Hence, if each node can find sampled time  $k_1^i, k_2^i, \dots, k_n^i$  such that the matrix  $\Omega_i$  in (14) is non-singular, it can then compute the vector  $\tilde{w}_i = [w_{1,i}, \dots, w_{n,i}]^T$  according to

$$\tilde{w}_i = \Omega_i^{-1} \tilde{x}_i. \quad (15)$$

Note that since each node knows all the estimated eigenvalues of matrix  $A$  (see Section 3.1), no communication is required between the nodes (subsystems) to compute (15) as can be observed from (14). Furthermore, from (15) each node will be able to estimate the  $i$ -th element of the right eigenvectors  $w_j$  for  $j = 1, \dots, n$ .

The previously described algorithm (15) requires the existence of non-singular matrix  $\Omega_i$ . Conditions that guarantee the existence of such matrix is an ongoing work. In case

Table 1. Comparison between algorithms (15) and (16) for estimating right eigenvectors  $w_i$

	Algorithm (15)	Algorithm (16)
communication	not required	required
number of samples	$n$	larger than $n$
samples selection $k_1, \dots, k_N$	independent for each node	similar for all nodes
estimated $w_i$ at node $i$	$w_{1,i}, \dots, w_{n,i}$	entire $w_j$ for $j = 1, \dots, n$

that the nodes cannot construct non-singular matrix  $\Omega_i$ , the right eigenvectors  $w_i$  can be alternatively estimated by solving distributively an LS problem. To this end, for a number of samples  $N > n$  we can write (13) as

$$\Phi \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \tilde{x}_1^N \\ \vdots \\ \tilde{x}_n^N \end{bmatrix}$$

where  $\Phi \in \mathbb{R}^{Nn \times n^2}$  is a sparse matrix whose non-zero elements are  $z_i^k$  for with  $k = k_1, \dots, k_N$ . Furthermore, the vector  $\tilde{x}_i^N$  is given by  $\tilde{x}_i^N = [x_i(k_1), \dots, x_i(k_N)]^T$ . Hence, the right eigenvectors  $w_i$  can be estimated by solving the following LS problem.

$$\min_{[w_1^T, \dots, w_n^T]^T} \frac{1}{2} \left\| \Phi \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} - \begin{bmatrix} \tilde{x}_1^N \\ \vdots \\ \tilde{x}_n^N \end{bmatrix} \right\|^2. \quad (16)$$

Furthermore, using only local information available to each node, that is  $\tilde{x}_i^N$ , optimization (16) can be solved in a distributed manner by reformulating it as in (12) and applying the standard distributed optimization algorithm proposed in the literature, for example (Wang et al., 2019).

In contrast to (15), the nodes need to communicate via a connected undirected communication network in order to solve optimization problem (16). In addition, after solving (16) each node will know the entire right eigenvectors of matrix  $A$ , that is  $w_i$  for  $i = 1, \dots, n$ . Detailed comparison between the two approaches is summarized in Table 1.

### 3.3 Distributed Left Eigenvector Estimation

After estimating the eigenvalues and corresponding right eigenvectors, the final step is to estimate in a distributed manner the left eigenvectors  $\nu_i$ . As mentioned in the previous subsection, when the nodes cannot construct the non-singular matrix  $\Omega_i$  in (14) they then estimate the right eigenvectors  $w_i$  by solving optimization problem (16) in a distributed manner. As a result, each node will know all the right eigenvectors  $w_j$  for  $j = 1, \dots, n$ . Since each node knows the entire right eigenvectors, it can then compute the left eigenvectors from (6) without requiring communication with other nodes in the network.

On the other hand, if there exists a non-singular matrix  $\Omega_i$  in (14), the  $i$ -th node can then estimate independently parts of the right eigenvectors given by  $w_{1,i}, \dots, w_{n,i}$  from (15). The next step is to estimate the left eigenvectors  $\nu_i$ . To this end, from (6) we have

$$WV = I_n. \quad (17)$$

with matrices  $W$  and  $V$  are defined in (5). Note that from (15) each node knows the  $i$ -th row of matrix  $W$ ,

**Algorithm 1** Distributed estimation algorithms (there exists a non-singular matrix  $\Omega_i$ )

**Require:** Communication network topology is bidirectional and connected

- 1: each node collects  $x_i(k)$
- 2: each node **cooperatively** estimates all the eigenvalues by solving optimization (12)
- 3: node  $i$  **independently** (i.e., requires no communication) computes the right eigenvector  $w_{1,i}, \dots, w_{n,i}$  from (15) using  $n$  sampled data
- 4: each node **cooperatively** estimates the left eigenvector  $\nu_{1,l}, \dots, \nu_{n,l}$  for  $l = 1, \dots, n$  using update rule (19)

**Algorithm 2** Distributed estimation algorithms (the node cannot construct a non-singular matrix  $\Omega_i$ )

**Require:** Communication network topology is bidirectional and connected

- 1: each node collects  $x_i(k)$
- 2: each node **cooperatively** estimates all the eigenvalues by solving optimization (12)
- 3: each node **cooperatively** computes all the right eigenvectors  $w_1, \dots, w_n$  by solving optimization (16) using  $N > n$  sampled data
- 4: node  $i$  **independently** (i.e., requires no communication) estimates all the left eigenvector  $\nu_1, \dots, \nu_n$  from (6)

that is  $[W]_{i*}^T$ . Each node can then estimate  $\nu_i$  by solving distributively a set of linear equations given in (17) as proposed in (Gusrialdi and Qu, 2017), originally developed for directed graph. Specifically, from (17) each node can estimate the left eigenvector  $\nu_{1,l}, \dots, \nu_{n,l}$ , that is vector  $[V]_{*l}$  by solving the following linear equation

$$W[V]_{*l} = [I_n]_{*l}. \quad (18)$$

In order to solve (18) in a distributed manner, node  $i$  maintains a local estimate of  $[V]_{*l}$  denoted by  $\hat{\nu}_l^i$  and updates its estimate according to the following update rule

$$\begin{aligned} \hat{\nu}_l^i(m+1) = & \hat{\nu}_l^i(m) - P_i \left( \hat{\nu}_l^i(m) - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \hat{\nu}_l^j(m) \right) \\ & - [W]_{i*} ([W]_{i*}^T [W]_{i*})^{-1} ([W]_{i*}^T \hat{\nu}_l^i - c_i), \\ & m = 0, 1, \dots \end{aligned} \quad (19)$$

where  $\hat{\nu}_l^i(0)$  is chosen to be an arbitrary vector,  $c_i = 1$  if  $i = l$  and zero otherwise. Furthermore, matrix  $P_i$  is defined as

$$P_i = I_n - [W]_{i*} ([W]_{i*}^T [W]_{i*})^{-1} [W]_{i*}^T.$$

In order to execute update rule (19), the  $i$ -th node needs to communicate with its neighbors given by the set  $\mathcal{N}_i$ . It is shown in (Wang et al., 2017) that for a connected (undirected) graph, update rule (19) exponentially converges to the left eigenvector  $[V]_{*l}$ . Therefore, each node can cooperatively estimate all the left eigenvectors by solving a set of linear equations (18) using update rule (19) for  $l = 1, \dots, n$ . The complete distributed algorithms for estimating the eigenvalues together with both the left and right eigenvectors of matrix  $A$  are summarized in Algorithms 1 and 2.

Note that after distributively estimating  $\lambda_i(A)$  together with the eigenvectors  $w_i, \nu_i$  using Algorithm 1, node  $i$

can then also estimate in a distributed manner the local dynamics, that is the  $i$ -th row of matrix  $A$  denoted by  $[A]_{i*}$ . To this end, we can write

$$A = W\Lambda V \quad (20)$$

where  $\Lambda$  is a diagonal matrix whose diagonal elements correspond to the eigenvalues of matrix  $A$ . Hence, since the  $i$ -th node knows  $[W]_{i*}$ ,  $\lambda_i(A)$  for all  $i$  and  $[V]_{*l}$  for all  $l$ , it can then compute  $[A]_{i*}$  from (20).

#### 4. NUMERICAL EXAMPLE

In order to illustrate the proposed distributed algorithms, we consider an interconnected system consisting of four subsystems as illustrated in Fig. 1 whose overall dynamics is given by

$$A = \begin{bmatrix} -4 & 1 & 3 & 2 \\ 1 & -3 & -7 & -2 \\ 1 & 2 & -4 & 1 \\ -2 & 2 & 1 & -5 \end{bmatrix}. \quad (21)$$

Each node first estimates all the eigenvalues of matrix  $A$  using the Prony method described in Section 3.1, i.e. by solving distributively optimization problem (12) using the combination of gradient and consensus algorithms proposed in the literature. The number of sampled data used for estimating the eigenvalues is equal to 40 with sampling time  $T = 0.02s$  and the communication network topology  $\mathcal{G}$  for the distributed estimation is shown in Fig. 1. As can be observed from the figure, the communication network topology can be chosen to be sparse (given that it is connected) even though matrix  $A$  is dense. Moreover, the communication network topology is independent of the structure of matrix  $A$  (or the structure of physical interconnection of the dynamical system). By solving distributively optimization (12), each node is able to estimate all the complex eigenvalues of  $A$  given by

$$\begin{aligned} \lambda_1(A) &= -4.8091 + 0.9646i, \lambda_2(A) = -4.8091 - 0.9646i \\ \lambda_3(A) &= -3.1909 + 4.0471i, \lambda_4(A) = -3.1909 - 4.0471i. \end{aligned}$$

Next, each node distributively estimates the right eigenvectors associated with  $\lambda_i(A)$  using the method described in Section 3.2. Since the eigenvalues are complex number, the complex right eigenvectors will have the following form

$$\begin{aligned} w_1 &= w_1^r + w_1^m i, w_3 = w_3^r + w_3^m i, \\ w_2 &= w_1^r - w_1^m i, w_4 = w_3^r - w_3^m i, \end{aligned}$$

where  $w_1^r, w_1^m, w_3^r, w_3^m \in \mathbb{R}^4$ . Using these expressions together with the values of  $\lambda_i(A)$  estimated previously and after some manipulations, linear equations (14) can then be written as

$$\underbrace{\begin{bmatrix} g_1(k_1^i) & -b_1(k_1^i) & g_3(k_1^i) & -b_3(k_1^i) \\ g_1(k_2^i) & -b_1(k_2^i) & g_3(k_2^i) & -b_3(k_2^i) \\ g_1(k_3^i) & -b_1(k_3^i) & g_3(k_3^i) & -b_3(k_3^i) \\ g_1(k_4^i) & -b_1(k_4^i) & g_3(k_4^i) & -b_3(k_4^i) \end{bmatrix}}_{\Omega_i} \begin{bmatrix} w_{1,i}^r \\ w_{1,i}^m \\ w_{3,i}^r \\ w_{3,i}^m \end{bmatrix} = \begin{bmatrix} x_i(k_1^i) \\ x_i(k_2^i) \\ x_i(k_3^i) \\ x_i(k_4^i) \end{bmatrix}$$

where

$$\begin{aligned} g_1(k_j^i) &= 2\Re\{e^{\lambda_1(A)k_j^i T}\}, b_1(k_j^i) = 2\Im\{e^{\lambda_1(A)k_j^i T}\} \\ g_3(k_j^i) &= 2\Re\{e^{\lambda_3(A)k_j^i T}\}, b_3(k_j^i) = 2\Im\{e^{\lambda_3(A)k_j^i T}\}. \end{aligned}$$

Each node then finds time samples  $k_1^i, k_2^i, k_3^i, k_4^i$  which makes matrix  $\Omega_i$  non-singular. In this case, node  $i$  can

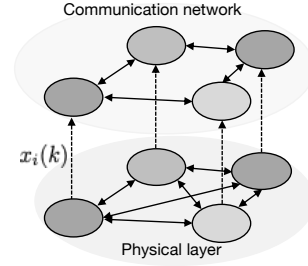


Fig. 1. Interconnected system and the corresponding communication network topology used for distributed estimation

choose  $k_1^i = 1, k_2^i = 10, k_3^i = 20, k_4^i = 30$  resulting in non-singular matrix

$$\Omega_i = \begin{bmatrix} 1.8163 & -0.0350 & 1.8702 & -0.1517 \\ 0.7502 & -0.1465 & 0.7289 & -0.7648 \\ 0.2707 & -0.1099 & -0.0268 & -0.5575 \\ 0.0935 & -0.0611 & -0.2229 & -0.1929 \end{bmatrix}.$$

Using this matrix and from (15), the  $i$ -th node can then estimate  $w_{1,i}^r, w_{1,i}^m, w_{3,i}^r, w_{3,i}^m$ . All the estimated right eigenvectors are given by

$$w_1 = \begin{bmatrix} -3.2548 + 1.3018i \\ 0.0284 + 0.8194i \\ -0.9406 + 1.9423i \\ 2.0857 - 5.4195i \end{bmatrix}, w_3 = \begin{bmatrix} 3.7548 - 7.8041i \\ 2.4716 + 12.2771i \\ 5.4406 - 3.0170i \\ 7.9143 + 2.8276i \end{bmatrix}$$

and  $w_2$  (resp.  $w_4$ ) is equal to the complex conjugate of  $w_1$  (reps.  $w_3$ ).

Finally, each node estimates all the left eigenvectors by solving a set of linear equations (17) in a distributed manner as described in Section 3.3. Since the right eigenvectors are complex numbers, we first define

$$W_r = \Re\{W\}, W_m = \Im\{W\},$$

that is, matrix  $W$  can be written as  $W = W_r + W_m i$ . Similarly, we also define for matrix  $V$  (to be estimated)

$$V_r = \Re\{V\}, V_m = \Im\{V\}.$$

From (17), we then have the following relationships

$$\begin{aligned} W_r V_r - W_m V_m &= I_n, \\ W_r V_m + W_m V_r &= 0. \end{aligned} \quad (22)$$

Each node can estimate the complex left eigenvector  $\nu_{1,j}, \dots, \nu_{n,j}$ , that is vector  $[V]_{*j}$  by solving (22) in a distributed fashion. For example, to estimate  $[V]_{*l}$ , each node needs to cooperatively solve the following linear equations

$$\begin{aligned} W_r [V_r]_{*l} - W_m [V_m]_{*l} &= [I_n]_{*l}, \\ W_r [V_m]_{*l} + W_m [V_r]_{*l} &= [0]_{*l} \end{aligned}$$

which can also be written as

$$\underbrace{\begin{bmatrix} W_r & -W_m \\ W_m & W_r \end{bmatrix}}_{\bar{W}} \begin{bmatrix} [V_r]_{*l} \\ [V_m]_{*l} \end{bmatrix} = \begin{bmatrix} [I_n]_{*l} \\ [0]_{*l} \end{bmatrix}. \quad (23)$$

In comparison to (18), in (23) each node knows two rows of the matrix  $\bar{W}$  corresponding to both the real and imaginary parts of the right eigenvectors  $w_{1,i}, w_{2,i}, w_{3,i}, w_{4,i}$ . Nevertheless, each node can still execute update law (19) to estimate the vectors  $[V_r]_{*l}, [V_m]_{*l}$  by incorporating the fact that  $[\bar{W}]_{i*} \in \mathbb{R}^{2n \times 2}$  and  $c_i \in \mathbb{R}^2$ . All the estimated

left eigenvectors are given by (note that node  $i$  has the estimates of  $\nu_{1,j}, \dots, \nu_{n,j}$ )

$$\nu_1 = - \begin{bmatrix} 0.2049 + 0.0802i \\ -0.0868 + 0.0018i \\ 0.1785 - 0.1438i \\ 0.0016 + 0.0603i \end{bmatrix}, \nu_3 = \begin{bmatrix} -0.0032 - 0.0065i \\ 0.0075 - 0.0395i \\ 0.0593 + 0.0219i \\ 0.0216 + 0.0003i \end{bmatrix}$$

and  $\nu_2$  (resp.  $\nu_4$ ) is equal to the complex conjugate of  $\nu_1$  (reps.  $\nu_3$ ).

## 5. CONCLUSION AND FUTURE WORK

We propose data-driven distributed algorithms to estimate (learn) the eigenmodes of an unknown (linear or linearized) dynamical system. An important feature is that topology of the communication network used for the distributed estimation can be chosen arbitrarily and is independent of the sparsity of system (state) matrix as long as it is connected. The proposed framework can also be extended to the case of directed communication network topology given that the least square solutions can be computed distributively under strongly connected graph. Ongoing and several possible future work are listed as follows.

- (1) We are currently investigating the conditions which guarantee the existence of non-singular matrix  $\Omega_i$  in (14) for estimating the right eigenvector.
- (2) In this paper, as a first step we assume that the system matrix has distinct eigenvalues. The next step is to extend the proposed strategies to deal with system matrix which has repeated eigenvalues.
- (3) It is also of importance to extend the proposed strategies to deal with noisy data
- (4) Finally, we will also extend the method by taking into account the non-ideal communication network, i.e., by introducing time delay.

## REFERENCES

- Charalambous, T., Rabbat, M.G., Johansson, M., and Hadjicostis, C.N. (2016). Distributed finite-time computation of digraph parameters: left-eigenvector, out-degree and spectrum. *IEEE Transactions on Control of Network Systems*, 3(2), 137–148.
- Chow, J.H. (2012). *Power System Coherency and Model Reduction*. Springer.
- Fan, L. (2017). Data fusion-based distributed prony analysis. *Electric Power Systems Research*, 143, 634–642.
- Franceschelli, M., Gasparri, A., Giua, A., and Seatzu, C. (2013). Decentralized estimation of laplacian eigenvalues in multi-agent systems. *Automatica*, 49(4), 1031–1036.
- Golub, G.H. and Van Loan, C.F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.
- Gusrialdi, A., Chakraborty, A., and Z.Qu (2018). Distributed learning of mode shapes in power system models. In *IEEE Conference on Decision and Control*, 4002–4007.
- Gusrialdi, A. and Qu, Z. (2017). Distributed estimation of all the eigenvalues and eigenvectors of matrices associated with strongly connected digraphs. *IEEE control systems letters*, 1(2), 328–333.
- Gusrialdi, A., Qu, Z., and Hirche, S. (2019). Distributed link removal using local estimation of network topology. *IEEE Transactions on Network Science and Engineering*, 6(3), 280–292.
- Gusrialdi, A., Xu, Y., Qu, Z., and Simaan, M.A. (2019). A real-time big data control-theoretical framework for cyber-physical-human systems. In *Computational Intelligence and Optimization Methods for Control Engineering*, 149–172. Springer.
- Khazaei, J., Fan, L., Jiang, W., and Manjure, D. (2016). Distributed prony analysis for real-world pmu data. *Electric Power Systems Research*, 133, 113–120.
- Kibangou, A.Y. and Commault, C. (2012). Decentralized laplacian eigenvalues estimation and collaborative network topology identification. In *3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 7–12.
- Martins, N. and Lima, L.T. (1990). Determination of suitable locations for power system stabilizers and static var compensators for damping electromechanical oscillations in large scale power systems. *IEEE Transactions on Power Systems*, 5(4), 1455–1469.
- McDaniel, P. and McLaughlin, S. (2009). Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3), 75–77.
- Nabavi, S., Zhang, J., and Chakraborty, A. (2015). Distributed optimization algorithms for wide-area oscillation monitoring in power systems using interregional pmu-pdc architectures. *IEEE Transactions on Smart Grid*, 6(5), 2529–2538.
- Petrie, A. and Zhao, X. (2012). Estimating eigenvalues of dynamical systems from time series with applications to predicting cardiac alternans. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 468(2147), 3649–3666.
- Qu, Z. (2009). *Cooperative Control of Dynamical Systems*. Springer Verlag, London.
- Qu, Z. and Simaan, M.A. (2014). Modularized design for cooperative control and plug-and-play operation of networked heterogeneous systems. *Automatica*, 50(9), 2405–2414.
- Rouco, L. (1998). Eigenvalue-based methods for analysis and control of power system oscillations. In *IEEE Colloquium on Power System Dynamics Stabilisation*, 3/1–3/6.
- Tran, T.M.D. and Kibangou, A.Y. (2015). Distributed estimation of laplacian eigenvalues via constrained consensus optimization problems. *Systems & Control Letters*, 80, 56–62.
- Wang, X., Zhou, J., Mou, S., and Corless, M.J. (2019). A distributed algorithm for least squares solutions. *IEEE Transactions on Automatic Control*, 64(10), 4217–4222.
- Wang, X., Mou, S., and Sun, D. (2017). Improvement of a distributed algorithm for solving linear equations. *IEEE Transactions on Industrial Electronics*, 64(4), 3113–3117.
- Yang, M. and Tang, C.Y. (2015). Distributed estimation of graph spectrum. In *American Control Conference*, 2703–2708.
- Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., Hong, Y., Wang, H., Lin, Z., and Johansson, K.H. (2019). A survey of distributed optimization. *Annual Reviews in Control*, 47, 278–305.