

# Distributed Principal Subspace Estimation in Wireless Sensor Networks

Lin Li, *Student Member, IEEE*, Anna Scaglione, *Fellow, IEEE*, and Jonathan H. Manton, *Senior Member, IEEE*

**Abstract**—Motivated by applications in multi-sensor array detection and estimation, this paper studies the problem of tracking the principal eigenvector and the principal subspace of a signal’s covariance matrix adaptively in a fully decentralized wireless sensor network (WSN). Sensor networks are traditionally designed to simply gather raw data at a fusion center, where all the processing occurs. In large deployments, this model entails high networking cost and creates a computational and storage bottleneck for the system. By leveraging both sensors’ abilities to communicate and their local computational power, our objective is to propose distributed algorithms for principal eigenvector and principal subspace tracking. We show that it is possible to have each sensor estimate only the corresponding entry of the principal eigenvector or corresponding row of the  $p$ -dimensional principal subspace matrix and do so by iterating a simple computation that combines data from its network neighbors only. This paper also examines the convergence properties of the proposed principal eigenvector and principal subspace tracking algorithms analytically and by simulations.

**Index Terms**—Asynchronous time, average consensus, distributed algorithm, gossiping, signal detection, stochastic approximation, subspace estimation, subspace tracking, synchronous time.

## I. INTRODUCTION

SUBSPACE-BASED signal processing methods have been applied successfully to both spatial and temporal spectral analysis. These methods are essentially applications of the so called principal component analysis (PCA). Let  $\mathbf{r}_t \in \mathbb{C}^N$  be a data vector sampled at  $t$  and zero-mean over  $t$ . In most of the relevant applications, the covariance matrix of a zero-mean vector of measurements  $\mathbf{r}_t$  can be characterized by the following eigenvalue-decomposition (EVD)

$$\mathbf{R} = \mathbb{E} [\mathbf{r}_t \mathbf{r}_t^H] = [\mathbf{U}_s \mathbf{U}_n] \Lambda [\mathbf{U}_s \mathbf{U}_n]^H$$

where  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$  with ordered eigenvalues  $\lambda_1 \geq \dots \geq \lambda_N \geq 0$ , and the submatrix  $\mathbf{U}_s \in \mathbb{C}^{N \times p}$  identifies a set of  $p$  principal components that captures as much “energy”

Manuscript received July 30, 2010; revised December 02, 2010; accepted February 02, 2011. Date of publication February 24, 2011; date of current version July 20, 2011. This work was supported by the National Science Foundation (NSF) under Grant CCF-0729074. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Michael Gastpar.

L. Li and A. Scaglione are with the Department of Electrical and Computer Engineering, University of California, Davis CA 95616 USA (e-mail: llli@ucdavis.edu; ascaglione@ucdavis.edu).

J.H. Manton is with the Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Vic. 3010, Australia (e-mail: jmanton@unimelb.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTSP.2011.2118742

of the sampled signal  $\mathbf{r}_t$  as possible. The submatrix  $\mathbf{U}_n$  specifies a vector space (i.e., the noise space) which is orthogonal to  $\mathbf{U}_s$ . A *principal subspace* is defined as the vector space spanned by the  $p$  principal components. Subspace estimation is essential in many applications, for example, noise reduction, data compression, direction of arrival (DOA) estimation, etc. Motivated by the significance of subspace estimation in signal processing, this paper studies the following subspace tracking problem in a fully decentralized WSN.

### A. Problem Statement

Consider a network of  $N$  sensor nodes, each collecting a zero-mean measurement  $r_t(i)$  at time  $t$ , for  $i = 1, \dots, N$ . Denote by  $\mathbf{r}_t \in \mathbb{C}^N$  a vector obtained by stacking these  $N$  measurements. Its covariance matrix is defined as  $\mathbf{R} = \mathbb{E} [\mathbf{r}_t \mathbf{r}_t^H]$ . From this information, and using only limited local communication between neighboring nodes, the network needs to distributively determine the first principal component of  $\mathbf{R}$ , that is, an eigenvector associated with the largest eigenvalue of  $\mathbf{R}$ . Precisely, in our first algorithm, each node computes a scalar  $\hat{u}(i)$  such that the vector  $\hat{\mathbf{u}} = [\hat{u}(1), \dots, \hat{u}(N)]^T$  is an estimate of the principal eigenvector of  $\mathbf{R}$ .

In our second algorithm, the network is required to distributively track the principal subspace of  $\mathbf{R}$  for any given dimension  $p$ , where  $1 < p \leq N$ . Suppose  $\mathbf{u}_1, \dots, \mathbf{u}_p$  are the  $p$  eigenvectors associated with the  $p$  largest eigenvalues of  $\mathbf{R}$ . Let  $[\mathbf{U}] \in \mathbb{C}^{N \times p}$  denote an *equivalence class* to the matrix  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_p]$  with an *equivalence relation*

$$[\mathbf{U}] = \left\{ \hat{\mathbf{U}} \in \mathbb{C}^{N \times p} \mid \hat{\mathbf{U}} = \mathbf{U} \mathbf{Q}, \text{ where } \mathbf{Q} \in \mathbb{C}^{p \times p} \text{ is a unitary matrix} \right\}. \quad (1)$$

As a result of the algorithm proposed in this paper, each node in the network asymptotically computes a row vector  $[\hat{u}(i, 1), \dots, \hat{u}(i, p)]$ , which is the  $i$ th row of the matrix  $\hat{\mathbf{U}}$  given by

$$\hat{\mathbf{U}} = \begin{bmatrix} \hat{u}(1, 1) & \cdots & \hat{u}(1, p) \\ \vdots & \vdots & \vdots \\ \hat{u}(N, 1) & \cdots & \hat{u}(N, p) \end{bmatrix}$$

such that  $\hat{\mathbf{U}} \in [\mathbf{U}]$  is a principal subspace of  $\mathbf{R}$ .

### B. Previous Works

Early approaches to estimate the subspace of a covariance matrix are based on the batch singular value decomposition (SVD) or the eigenvalue decomposition (EVD) of a data matrix

[1]. However, they are very computationally expensive because repeated computations of SVD/EVD are required for adaptive processing. To overcome this difficulty, many adaptive subspace estimation techniques have been proposed, such as the Oja's Learning Rule [2], the modified SVD/ED (e.g., QR decomposition, Lanczos method, power method, Jacobi rotation) [3]–[6], the Rank-one update algorithm [7] and a number of subspace estimation methods derived as a constrained or unconstrained optimization problem [8]–[11]. Most of the literature assumes that all the information is available at a single location. Recently, several authors [12]–[16] have considered alternative approaches to enable a distributed in-network computation by means of network diffusion (gossiping) using only local communication. These mechanisms significantly decrease the overall network discovery cost, and they are resilient to link failures and changes in network topology. Examples of in-network signal processing via gossiping are [17]–[23], and also distributed stochastic optimization [24], distributed Kalman filtering [25], distributed Least Mean Squares (LMS) filtering [26], [27], decentralized maximum-likelihood (ML) estimation [28], decentralized minimax detection [29], distributed covariance estimation [20], and decentralized subspace estimation [30]. A Gossip-like approach to compute  $p$  principal eigenvectors called Decentralized Orthogonal Iteration (DOI) was first proposed in [31]. It also utilizes a gossiping algorithm presented in [32] for distributed computation of an optimal weight matrix. A refined version of DOI was proposed in [33] for computing the second largest eigenvalue and its corresponding eigenvector. Both papers assume that the covariance matrix, whose sparsity structure mimics the static network, is already calculated and its entries are distributed across the network. In this paper, we propose a different distributed approach for adaptively estimating the principal subspace, given only the received signals. We extend the work in [30] to the principal subspace estimation considering both a deterministic scheme and a randomized scheme for message passing. Our primary goal is to demonstrate that the proposed distributed eigenvector and subspace tracking algorithms can achieve similar performances as compared to their centralized computations.

### C. Data Fusion Schemes

The need for distributed subspace estimation is motivated by the many applications that resort to PCA to extract information from sensors' measurements. A typical WSN consists of spatially distributed sensors, which cooperatively monitor the environmental conditions of interest. Oftentimes, sensors have limited communication range, computation resources and memory. Also, wireless networks are subject to changes in communication topology due to fading. Therefore, in the context of WSN, it is widely recognized that a distributed data fusion protocol that is resilient to link failures and self-adaptive to changes in network topology, is an appealing option.

As shown in Fig. 1, we assume that all the sensors in the network are restricted to near-neighbor/local communication. The network is assumed to be connected in expectation. To simplify the analysis, we first consider an ideal case, where local communication is performed through a static network, in which time is equally divided into time slots and every node simultaneously

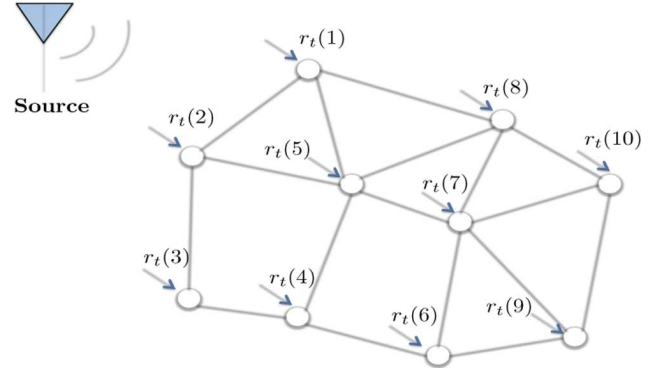


Fig. 1. Near-neighbor communication topology.

updates its own variable with a weighted average of its neighbors' values. Data exchanges are assumed to be perfect with no link failure. To extend the results beyond this ideal case, this paper introduces a second version of the algorithms using a randomized gossip scheme for data fusion, where, in the time that elapses between one sample and the next, the sensor nodes exchange messages at random times to update their corresponding computations. Also, link failures are modeled and analyzed in this framework.

### Notation and Outline

Let the EVD of  $\mathbf{R}$  be  $\mathbf{R} = \sum_{i=1}^N \lambda_i(\mathbf{R}) \mathbf{u}_i \mathbf{u}_i^H$ , in which, for  $1 \leq i \leq N$ ,  $\mathbf{u}_i$  are the eigenvectors of  $\mathbf{R}$  and their associated eigenvalues are ordered from high to low. Denote the principal eigenvector of  $\mathbf{R}$  by  $\mathbf{u}_1 \in \mathbb{C}^{N \times 1}$  and an estimated one by  $\hat{\mathbf{u}} \in \mathbb{C}^{N \times 1}$ . Let  $\mathbf{U} \in \mathbb{C}^{N \times p}$  represent a matrix containing  $p$  columns of the principal components of  $\mathbf{R}$ , that is  $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_p]$ . Denote by  $[\mathbf{U}] \subset \mathbb{C}^{N \times p}$  an equivalence class to  $\mathbf{U}$  and an estimated principal subspace by  $\hat{\mathbf{U}}$ . Our goal is to iteratively compute a vector  $\hat{\mathbf{u}}$  and a matrix  $\hat{\mathbf{U}} \in \mathbb{C}^{N \times p}$  such that  $\hat{\mathbf{u}} = \mathbf{u}_1$  and  $\hat{\mathbf{U}} \in [\mathbf{U}]$ , respectively.

Denote by  $[\mathbf{A}]_{i,j}$  the  $(i,j)$ th element of the matrix  $\mathbf{A}$ . The symbol  $\odot$  represents an element by element/Hadamard matrix product. Also,  $\text{tr}(\cdot)$  denotes the trace of a matrix in the parenthesis. Frobenius norm of a matrix  $\mathbf{A}$  is defined by  $\|\mathbf{A}\|_F = \sqrt{\text{tr}(\mathbf{A}^H \mathbf{A})}$ . Furthermore, as mentioned before, we discriminate the true principal eigenvector and the subspace spanning the  $p$  principal components using the roman letters  $\mathbf{u}_1$  and  $[\mathbf{U}]$  respectively, and the calligraphic versions  $\hat{\mathbf{u}}$  and  $\hat{\mathbf{U}}$  for their estimates.

The organization of this paper is as follows. After briefly presenting the system's mathematical model in Section II, Section III introduces a deterministic data fusion scheme which is called *distributed average consensus protocol*. Following that, we derive adaptive algorithms incorporating the distributed average consensus protocol in a synchronous time model for computing the principal eigenvector and the  $p$ -dimensional principal subspace of a signal's covariance matrix, and provide stability analysis in Sections IV and V, respectively. Section VI presents the distributed eigenvector and subspace tracking algorithms using a *randomized gossip algorithm*. Connections between the deterministic algorithms with synchronous updates and the randomized gossip algorithms for subspace tracking are also highlighted.

## II. PROBLEM SETUP IN A STATIC NETWORK

*Assumption 1:* The received signal  $\mathbf{r}_t$  is a zero-mean stationary process. We assume that  $t$  is sufficiently large such that its covariance  $\mathbf{R} = \mathbb{E}[\mathbf{r}_t \mathbf{r}_t^H]$  is well approximated by the sample covariance  $\hat{\mathbf{R}} = (1/t) \sum_{s=1}^t \mathbf{r}_s \mathbf{r}_s^H$ .

$\mathcal{G} = (\mathcal{N}, \mathcal{E})$  represents the undirected graph associated with the given connected time-invariant network. The set  $\mathcal{N} = \{1, 2, \dots, N\}$  denotes the nodes, and the set  $\mathcal{E}$  is a collection of edges  $\{i, j\}$  which describes the available links in the network. Let Assumption 1 hold. The principal eigenvector tracking problem consists of adaptively computing the eigenvector of  $\mathbf{R}$  which is approximated by the sample covariance  $\hat{\mathbf{R}}$ , given the sampled vectors  $\mathbf{r}_t$  for  $t = 0, 1, \dots, T$ .

To extract the principal eigenvector of  $\mathbf{R}$ , consider the well-known Oja's learning rule [2], [34],

$$\hat{\mathbf{u}}_t = \hat{\mathbf{u}}_{t-1} + \gamma_t \left[ \mathbf{r}_t \mathbf{r}_t^H \hat{\mathbf{u}}_{t-1} - \hat{\mathbf{u}}_{t-1} \hat{\mathbf{u}}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \hat{\mathbf{u}}_{t-1} \right] \quad (2)$$

where  $\hat{\mathbf{u}}_0$  is an unit vector. Over the time, the Oja's learning rule in (2) converges uniformly to the principal eigenvector of  $\mathbf{R}$  if certain conditions are met [34]. Notice that the computation of the learning rule (2) requires access to the entire vector of observations  $\mathbf{r}_t$ . In a centralized network, this can be achieved by passing all the observations to a data fusion site. However, in a decentralized network, each node only possesses the knowledge of its own observation  $r_t(i)$ . The key idea to decentralize the computation in (2) is to have each node compute only its corresponding entry of the eigenvector estimate, rather than compute the entire vector  $\hat{\mathbf{u}}_t$ . Observe from (2) that each component of  $\hat{\mathbf{u}}_t$  is updated as follows:

$$\hat{u}_t(i) = \hat{u}_{t-1}(i) + \gamma_t \left( r_t(i) \mathbf{r}_t^H \hat{\mathbf{u}}_{t-1} - \hat{u}_{t-1}(i) \left| \mathbf{r}_t^H \hat{\mathbf{u}}_{t-1} \right|^2 \right).$$

Therefore, the  $i$ th node can compute  $\hat{u}_t(i)$  in a distributed way provided that it can obtain (a reasonably good estimate of) the inner product  $\mathbf{r}_t^H \hat{\mathbf{u}}_{t-1}$ . Hence, all it takes to decentralize the application of Oja's rule is a distributed method to compute an inner product. Section III shows how an average consensus protocol can be used to compute this inner product in a distributed manner.

## III. DISTRIBUTED AVERAGE CONSENSUS PROTOCOL

Recall that it is required to compute an inner product  $\mathbf{r}_t^H \hat{\mathbf{u}}_{t-1} = \sum_{i=1}^N r_t(i) \hat{u}_{t-1}(i)$  distributively, where *a priori* the  $i$ th node only has access to the  $i$ th components of  $\mathbf{r}_t$  and  $\hat{\mathbf{u}}_{t-1}$ . Since addition is distributive, it can be anticipated that this is achievable, and indeed, the problem is essentially equivalent to the well-studied average consensus problem [14], [16], [35] of computing distributively the average value  $(1/N) \sum_{i=1}^N r_t(i) \hat{u}_{t-1}(i)$  of each node's state, which is set to be  $r_t(i) \hat{u}_{t-1}(i)$ .

Let  $z_0(i) = r_t(i) \hat{u}_{t-1}(i)$  be the initial value. Hence,

$$\mathbf{z}_0 = [r_t(1) \hat{u}_{t-1}(1), \dots, r_t(N) \hat{u}_{t-1}(N)]^T \quad (3)$$

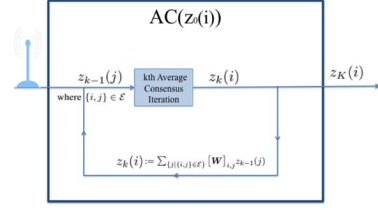


Fig. 2. Average Consensus Protocol performed at the  $i$ th node.  $\mathbf{W}$  denotes the weight matrix.

represents the initial state vector to the network. A distributed average consensus protocol iteratively computes the average  $\bar{z}_0 = (1/N) \sum_{i=1}^N z_0(i)$  at each node as illustrated in Fig. 2. Specifically, at the  $k$ th average consensus iteration, node  $i$  receives data  $z_{k-1}(j)$ 's from its near neighbors  $j$  for  $(i, j) \in \mathcal{E}$ . It then updates the value  $z_{k-1}(i)$  to  $z_k(i)$  by taking a weighted average of all the received data  $z_{k-1}(j)$ 's and the value  $z_{k-1}(i)$  obtained from the previous iteration.

Define a symmetric weight matrix  $\mathbf{W} = \mathbf{W}^T$  for the given undirected network graph  $\mathcal{G}$ . The  $(i, j)$ th entry  $[\mathbf{W}]_{i,j}$  of  $\mathbf{W}$  is the weight associated with the edge  $\{i, j\}$ . In particular,  $[\mathbf{W}]_{i,j} = [\mathbf{W}]_{j,i} \in (0, 1)$  if and only if  $\{i, j\} \in \mathcal{E}$  and  $[\mathbf{W}]_{i,j} = 0$  otherwise. Notice that  $\mathbf{W}$  has the same sparsity pattern as the network graph. Moreover,  $\mathbf{W}$  is doubly stochastic (i.e.,  $\mathbf{W} \mathbf{1}_N = \mathbf{1}_N$ ,  $\mathbf{1}_N^T \mathbf{W} = \mathbf{1}_N^T$ ), and hence  $\mathbf{1}_N$  is an eigenvector associated with its largest eigenvalue  $\lambda_1(\mathbf{W}) = 1$ .

The average consensus iteration performed at node  $i$  is  $z_k(i) = [\mathbf{W}]_{i,i} z_{k-1}(i) + \sum_{(i,j) \in \mathcal{E}} [\mathbf{W}]_{i,j} z_{k-1}(j)$ , for  $1 \leq k \leq K$ , where  $\mathbf{z}_0$  is defined in (3) and  $k$  is an index counting the number of consensus iterations performed. A vector form of this iteration is  $\mathbf{z}_k = \mathbf{W} \mathbf{z}_{k-1}$ . For finite  $K$ , a vector-valued approximation for the average  $\bar{z}_0$  in terms of  $\mathbf{z}_0$  is

$$\mathbf{z}_K = \mathbf{W}^K \mathbf{z}_0. \quad (4)$$

The following assumption, which summarizes the results from [14], states conditions for the network to achieve asymptotic average consensus.

*Assumption 2:* For a given undirected network graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  and its associated doubly stochastic weight matrix  $\mathbf{W} \in \mathbb{R}^{N \times N}$ , the convergence conditions imposed on the network are (1) the graph  $\mathcal{G}$  is connected, (2)  $\mathbf{W}$  is doubly stochastic, and (3)  $\lambda_2(\mathbf{W}) < 1$ .

If Assumption 2 is satisfied, this indicates [14]

$$\mathbf{z}_K = \mathbf{W}^K \mathbf{z}_0 \xrightarrow{k \rightarrow \infty} \bar{z}_0 \mathbf{1}_{N \times 1} \quad (5)$$

where  $K$  denotes the number of average consensus iterations performed. As  $K$  approaches to infinity, the value  $z_K(i)$  computed at each node converges to the averaged value (i.e.,  $(1/N) \mathbf{r}_t^H \hat{\mathbf{u}}_{t-1}$ ) of the initial state vector  $\mathbf{z}_0$ . For a finite number of iterations, the error [14] is bounded by

$$\|\mathbf{z}_K - \bar{z}_0 \mathbf{1}\|_2 \leq \lambda_2^K \|\mathbf{z}_0 - \bar{z}_0 \mathbf{1}\|_2. \quad (6)$$

#### IV. DECENTRALIZED PRINCIPAL EIGENVECTOR COMPUTATION VIA AVERAGE CONSENSUS

##### Algorithm

Consider the problem of distributively computing the first principal component of  $\mathbf{R}$  based on the Oja's learning rule (2) via near-neighbor communication. Our idea is simple. We propose to compute, at each time  $t$ , the  $i$ th component of the estimate  $\hat{\mathbf{u}}_t$ , following the Oja's learning rule updated locally

$$\hat{u}_t(i) = \hat{u}_{t-1}(i) + \gamma_t \left( r_t(i) \mathbf{r}_t^H \hat{\mathbf{u}}_{t-1} - \hat{u}_{t-1}(i) |\mathbf{r}_t^H \hat{\mathbf{u}}_{t-1}|^2 \right) \quad (7)$$

via several iterations (i.e.,  $K$ ) of the average consensus protocol for distributively estimating the inner product  $\mathbf{r}_t^H \hat{\mathbf{u}}_{t-1}$ . Once each node approximately reaches a *consensus*, it then *updates* its local estimate using (7).

---

##### Algorithm 1: Principal Eigenvector Estimation (at node $i$ )

---

**input:** A sequence of  $r_t(i)$

**output:** The  $i$ th element of the principal eigenvector estimate  $\hat{u}$

1: **Initialization:**  $t = 0$ ;

2:  $\hat{u}_i = \sqrt{1/N}$ ;

3: **Recursion:**

4: **foreach**  $t = 1 : T$  **do**

5:  $\mathbf{ac}_i \leftarrow N \cdot \mathbf{AC}(\hat{u}_i r_{i,t}^*, K)$ ;

6:  $\hat{u}_i \leftarrow \hat{u}_i + \gamma_t (\mathbf{ac}_i) (r_{i,t} - \mathbf{ac}_i^* \cdot \hat{u}_i)$ ;

7: **end**

---

*Algorithm 1* summarizes the proposed distributed principal eigenvector estimation, where the scalar  $\gamma_t$  denotes the stepsize. We refer to [8], [34], and [36] for more detailed discussions on the stepsize rule. However, the asymptotic performance of a stochastic approximation is not affected by the choice of the stepsize [36]. Here,  $\hat{\mathbf{u}}_0$  is initialized as an unit vector. For convenience, we denote by  $\mathbf{AC}(\text{argument}, K)$  the output of  $K$  average consensus iterations at node  $i$  given the initial value argument. Its implementation is illustrated in Fig. 2. We remark that the algorithm as stated requires knowing the number of nodes  $N$  in the network. This number itself can be estimated using average consensus, for example using the method proposed in [37]. We will discuss the implementation of an average consensus protocol given a matrix-valued argument in Section V.

For future reference, we first state some properties of Hadamard product [38].

1) *Property 1:* Let  $\mathbf{a}, \mathbf{b}, \hat{\mathbf{u}} \in \mathbb{C}^{N \times 1}$ ,  $\mathbf{v} \in \mathbb{R}^{N \times 1}$ , and  $\mathbf{A} \in \mathbb{C}^{N \times N}$  be a Hermitian matrix, then

$$\begin{aligned} 1) \mathbf{a} \odot [\mathbf{A}(\mathbf{b}^* \odot \hat{\mathbf{u}})] &= [\mathbf{A} \odot (\mathbf{a}\mathbf{b}^H)] \hat{\mathbf{u}}; \\ 2) \text{diag}(\mathbf{v})\mathbf{A} \text{diag}(\mathbf{v}) &= (\mathbf{v}\mathbf{v}^T) \odot \mathbf{A}; \\ 3) [\mathbf{v}\mathbf{v}^T \odot (\hat{\mathbf{u}}\hat{\mathbf{u}}^H \mathbf{A})] \hat{\mathbf{u}} &+ [(\mathbf{v}\mathbf{v}^T) \odot (\hat{\mathbf{u}}\hat{\mathbf{u}}^H)] \mathbf{A} \hat{\mathbf{u}} = \\ &2\Re \left\{ (\mathbf{v}^T \odot \hat{\mathbf{u}}^H) \mathbf{A} \hat{\mathbf{u}} \right\} (\mathbf{v} \odot \hat{\mathbf{u}}). \end{aligned}$$

At time  $t$ , each node obtains a signal sample  $r_t(i)$  and a local eigenvector estimate  $\hat{u}_{t-1}(i)$ . The initial state vector to compute the inner product  $\mathbf{r}_t^H \hat{\mathbf{u}}_{t-1}$  is  $\mathbf{z}_0 = \mathbf{r}_t^* \odot \hat{\mathbf{u}}_{t-1}$ . Using (5), the distributed estimate of  $\mathbf{r}_t^H \hat{\mathbf{u}}_{t-1}$ , after  $K$  average consensus iterations, is  $\mathbf{z}_K = \{N\mathbf{W}^K(\mathbf{r}_t^* \odot \hat{\mathbf{u}}_{t-1})\}$ . Hence, (7)

can be written in a vector form as  $\hat{\mathbf{u}}_t = \hat{\mathbf{u}}_{t-1} + \gamma_t \mathbf{d}_{t,K}$ , where  $\mathbf{d}_{t,K} = \mathbf{r}_t \odot \mathbf{z}_K - \hat{\mathbf{u}}_{t-1} \odot \mathbf{z}_K^* \odot \mathbf{z}_K$ . Substituting the expression for  $\mathbf{z}_K$  and applying Property 1.1, the network-wide update is

$$\begin{aligned} \hat{\mathbf{u}}_t &= \hat{\mathbf{u}}_{t-1} + \gamma_t \mathbf{d}_{t,K}, \\ \text{where } \mathbf{d}_{t,K} &= N\mathbf{W}^K \odot \left[ (\mathbf{r}_t \mathbf{r}_t^H) \right. \\ &\quad \left. - \left( N\mathbf{W}^K \odot (\hat{\mathbf{u}}_{t-1} \hat{\mathbf{u}}_{t-1}^H) \right) \mathbf{r}_t \mathbf{r}_t^H \right] \hat{\mathbf{u}}_{t-1}. \quad (8) \end{aligned}$$

##### A. Stability Analysis

This section studies in detail the convergence properties of *Algorithm 1* by deriving the so-called associated *Ordinary Differential Equation* (ODE) [39] of the stochastic approximation in (8). The ODE analysis studies the asymptotic behavior of a system. Hence, we assume the stationarity assumption (Assumption 1) to hold over a long time horizon and  $\gamma_t$  to be small. Under certain conditions, we will show that the magnitude of the estimate  $\hat{\mathbf{u}}$  using *Algorithm 1* converges to a value that is in a neighborhood of 1. The distance between the stable equilibrium point of the ODE  $d\hat{\mathbf{u}}/dt$  and  $\mathbf{u}_1$  the principal eigenvector of  $\mathbf{R}$  decreases as  $K$  increases.

*Assumption 3:* The step-size  $\gamma_t$  is sufficiently small.

*Assumption 4:* The largest eigenvalue of  $\mathbf{R}$  has multiplicity one.

Let Assumption 3 hold. The associated ODE of the stochastic approximation [40] of  $\hat{\mathbf{u}}$  is defined as  $d\hat{\mathbf{u}}/dt = \mathbb{E}\{\mathbf{d}_{t,K}\}$ . Suppose Assumption 1 and 4 are true. By definition, the ODE of (8) is

$$\frac{d\hat{\mathbf{u}}}{dt} = \{N\bar{\mathbf{W}}^K \odot \mathbf{R} - N\bar{\mathbf{W}}^K \odot [(N\bar{\mathbf{W}}^K \odot (\hat{\mathbf{u}}\hat{\mathbf{u}}^H))\mathbf{R}]\} \hat{\mathbf{u}} \quad (9)$$

where  $\bar{\mathbf{W}}^K = \mathbb{E}[\mathbf{W}^K] = \mathbf{W}^K$ , because the network is static and thus  $\mathbf{W}$  matrix is time-invariant.

Let the EVD of  $\mathbf{W}$  be  $\mathbf{W} = \sum_{i=1}^N \lambda_i(\mathbf{W}) \mathbf{v}_i \mathbf{v}_i^T$  with  $1 = \lambda_1(\mathbf{W}) > \lambda_2(\mathbf{W}) > \dots \geq \lambda_N(\mathbf{W})$ , then  $\mathbf{W}^K = \mathbf{J} + \mathbf{X}^K$ , where  $\mathbf{J} = (1/N)\mathbf{1}\mathbf{1}^H$  is a rank one matrix and  $\mathbf{X}^K = \sum_{j=2}^N \lambda_j^K(\mathbf{W}) \mathbf{v}_j \mathbf{v}_j^T$ . Using this decomposition of  $\mathbf{W}^K$  and the relation  $N\mathbf{J} \odot \mathbf{A} = \mathbf{A}$  for any arbitrary  $N \times N$  matrix  $\mathbf{A}$ , the ODE expression in (9) can be written as

$$\frac{d\hat{\mathbf{u}}}{dt} = (\mathbf{I} - \hat{\mathbf{u}}\hat{\mathbf{u}}^H) \mathbf{R} \hat{\mathbf{u}} + \Delta_{\text{ODE}} \hat{\mathbf{u}} \quad (10)$$

where  $\Delta_{\text{ODE}} = N\mathbf{X}^K \odot \mathbf{R} - N\mathbf{X}^K \odot (\hat{\mathbf{u}}\hat{\mathbf{u}}^H \mathbf{R}) - (N\mathbf{X}^K \odot (\hat{\mathbf{u}}\hat{\mathbf{u}}^H)) \mathbf{R} + \mathcal{O}(N^2 \lambda_2^{2K}(\mathbf{W}))$ , for  $K$  sufficiently large. Equation (10) can be interpreted as the decomposition of the ODE into its equivalent form in a centralized network and the ODE deviation influenced by the decentralized network topology. The following part of this section is dedicated to characterize the norm of the eigenvector estimate  $\|\hat{\mathbf{u}}\|_2$  and provide necessary conditions for the decentralized algorithm to converge.

To study the stability of  $\hat{\mathbf{u}}^H \hat{\mathbf{u}}$ , we obtain the following dynamical system for  $\hat{\mathbf{u}}^H \hat{\mathbf{u}}$ :

$$\begin{aligned} \frac{d\hat{\mathbf{u}}^H \hat{\mathbf{u}}}{dt} &= \hat{\mathbf{u}}^H \frac{d\hat{\mathbf{u}}}{dt} + \frac{d\hat{\mathbf{u}}^H}{dt} \hat{\mathbf{u}} \\ &= 2(\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}) \left( 1 - \hat{\mathbf{u}}^H \hat{\mathbf{u}} + \delta_{\text{ODE}} \right) \quad (11) \end{aligned}$$

$$\text{where } \delta_{\text{ODE}} = \frac{\hat{\mathbf{u}}^H \left( \Delta_{\text{ODE}} + \Delta_{\text{ODE}}^H \right) \hat{\mathbf{u}}}{2\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}} = \frac{\Re \left\{ \hat{\mathbf{u}}^H \Delta_{\text{ODE}} \hat{\mathbf{u}} \right\}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}}. \quad (12)$$

$\delta_{\text{ODE}}$  represents the trajectory of the decentralized  $\hat{\mathbf{u}}^H \hat{\mathbf{u}}$  deviating from the trajectory of centralized (i.e.,  $k = \infty$ ) estimate of  $\hat{\mathbf{u}}^H \hat{\mathbf{u}}$ . Furthermore, the equilibrium point of (11) is

$$\hat{\mathbf{u}}_{\text{eq.}}^H \hat{\mathbf{u}}_{\text{eq.}} = 1 + \delta_{\text{ODE}} \quad (13)$$

which, for finite  $K$ , lies in the neighborhood of 1 as stated in the following lemma.

*Lemma 4.1 (Stability of  $\hat{\mathbf{u}}^H \hat{\mathbf{u}}$ ):* Suppose Assumption 1, 2, and 3 hold. Let  $(\lambda_2(\mathbf{W}), \mathbf{v}_2)$  denote the second largest eigenpair of  $\mathbf{W}$ . For any fixed number of average consensus iterations  $K$  satisfying  $\{K \in \mathbb{N}_+ | \lambda_2^K(\mathbf{W}) \ll 1/(2N)\}$ , the equilibrium point of  $\hat{\mathbf{u}}^H \hat{\mathbf{u}}$  is contained in the interval

$$(1 - 3N\lambda_2^K(\mathbf{W}), 1 + 3N\lambda_2^K(\mathbf{W})). \quad (14)$$

*Proof:* See Appendix A-A.  $\blacksquare$

Since  $(\lambda_2(\mathbf{W}), \mathbf{v}_2)$  is also the largest eigenpair of  $\mathbf{X}$ , then the approximation  $\mathbf{X}^K \approx \lambda_2^K(\mathbf{W}) \mathbf{v}_2 \mathbf{v}_2^H$  holds. Using Property 1.2, 1.3, and the approximation on  $\mathbf{X}^K$  yield (34), that is

$$\begin{aligned} \Delta_{\text{ODE}} \hat{\mathbf{u}} &= N\lambda_2^K(\mathbf{W}) \text{diag}(\mathbf{v}_2) \mathbf{R} \text{diag}(\mathbf{v}_2) \hat{\mathbf{u}} \\ &\quad - 2N\lambda_2^K(\mathbf{W}) \Re \left\{ \hat{\mathbf{u}}^H \text{diag}(\mathbf{v}_2) \mathbf{R} \hat{\mathbf{u}} \right\} \text{diag}(\mathbf{v}_2) \hat{\mathbf{u}} \\ &\quad + \mathcal{O}(N^2 \lambda_2^{2K}(\mathbf{W})). \end{aligned}$$

Let  $\Re \left\{ \hat{\mathbf{u}}^H \text{diag}(\mathbf{v}_2) \mathbf{R} \hat{\mathbf{u}} \right\} = \beta(\hat{\mathbf{u}})$ , then the ODE in (10) can be written as

$$\begin{aligned} g(\hat{\mathbf{u}}) &= \frac{d\hat{\mathbf{u}}}{dt} \\ &\approx \mathbf{R} - (\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}) \mathbf{I} \hat{\mathbf{u}} \\ &\quad + N\lambda_2^K(\mathbf{W}) [\text{diag}(\mathbf{v}_2) \mathbf{R} \text{diag}(\mathbf{v}_2) - 2\beta(\hat{\mathbf{u}}) \text{diag}(\mathbf{v}_2)] \hat{\mathbf{u}}. \end{aligned} \quad (15)$$

Define another dynamical system  $f(\hat{\mathbf{u}}) = \left[ \mathbf{R} - (\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}) \mathbf{I} \right] \hat{\mathbf{u}}$  which is called Oja's flow for computing the principal component of  $\mathbf{R}$ , and its convergence properties are studied in [2], [34], [41]. Denote the derivative of  $f$  by  $Df$ . We call  $g(\hat{\mathbf{u}})$  a perturbation of the Oja's flow  $f(\hat{\mathbf{u}})$  if

$$\|f(\hat{\mathbf{u}}) - g(\hat{\mathbf{u}})\| \text{ and } \|Df(\hat{\mathbf{u}}) - Dg(\hat{\mathbf{u}})\|$$

are small for all  $\hat{\mathbf{u}} \in \mathcal{W}$ . Suppose the open set  $\mathcal{W} = \{\hat{\mathbf{u}} \in \mathbb{C}^N \mid \|\hat{\mathbf{u}}\|^2 = 1 \pm 3N\lambda_2^K(\mathbf{W})\}$  and  $\|\hat{\mathbf{u}}_0\|_2 = 1$  is initialized in a neighborhood of its equilibrium point. The next two theorems [42] state general conditions for the equilibrium points of a dynamical system to persist under small perturbations.

*Theorem 4.1 (Perturbation Theorem):* Let  $f : \mathcal{W} \rightarrow \mathcal{E}$  be a  $C^1$  vector field and  $\mathbf{u}^* \in \mathcal{W}$  an equilibrium of  $du/dt = f(\mathbf{u})$  such that  $Df(\mathbf{u}^*) \in L(\mathcal{E})$  is invertible. Then there exists a neighborhood  $\mathcal{U} \subset \mathcal{W}$  of  $\mathbf{u}^*$  and a neighborhood  $\mathcal{V}$  of  $f$  such that for any  $g \in \mathcal{V}$  there is a unique equilibrium  $\hat{\mathbf{u}}^* \in \mathcal{U}$  of  $d\hat{\mathbf{u}}/dt = g(\hat{\mathbf{u}})$ . Moreover, if  $\mathcal{E}$  is normed, for any  $\epsilon > 0$ , we can choose  $\mathcal{V}$  so that  $\|\hat{\mathbf{u}}^* - \mathbf{u}^*\| < \epsilon$ .

A hyperbolic equilibrium  $\mathbf{u}^*$  means the eigenvalues of  $Df(\mathbf{u}^*)$  have nonzero real parts. Moreover,  $\mathbf{u}^*$  is called a sink if  $\text{ind}(\mathbf{u}^*) = \dim \mathcal{E}$ , where  $\text{ind}(\mathbf{u}^*)$  is the number of the eigenvalues of  $Df(\mathbf{u}^*)$  having negative real parts.

*Theorem 4.2:* Suppose that  $\mathbf{u}^*$  is a hyperbolic equilibrium. In Theorem 4.1,  $\mathcal{V}$  and  $\mathcal{U}$  can be chosen so that if  $g \in \mathcal{V}$ , the unique equilibrium  $\hat{\mathbf{u}}^* \in \mathcal{U}$  of  $d\hat{\mathbf{u}}/dt = g(\hat{\mathbf{u}})$  is also hyperbolic and it has the same index as  $\mathbf{u}^*$ .

Using Theorems 4.1 and 4.2, the following lemma states the convergence property of the ODE in (15).

*Lemma 4.2 (Convergence):* Let Assumption 1, 2, 3, and 4 hold. For  $N\lambda_2^K(\mathbf{W})\lambda_1(\mathbf{R})$  small, the ODE  $d\hat{\mathbf{u}}/dt = g(\hat{\mathbf{u}})$  is a perturbation of  $f(\hat{\mathbf{u}})$ . Moreover,  $d\hat{\mathbf{u}}/dt$  has a stable equilibrium point (i.e., a sink)  $\hat{\mathbf{u}}^*$  and it is in a neighborhood of  $\mathbf{u}_1$  (i.e., the principal eigenvector of  $\mathbf{R}$ ) within a radius of

$$\|\hat{\mathbf{u}}^* - \mathbf{u}_1\| \leq 3N\lambda_2^K(\mathbf{W})\lambda_1(\mathbf{R}) + \mathcal{O}(N^2\lambda_2^{2K}(\mathbf{W})).$$

*Proof:* See Appendix A-B.  $\blacksquare$

## V. DISTRIBUTED $p$ -DIMENSIONAL SUBSPACE TRACKING VIA AVERAGE CONSENSUS GOSSIPING

Consider the cost function [11]  $J(\hat{\mathbf{U}}) = \mathbb{E} \left\| \mathbf{r}_t - \hat{\mathbf{U}} \hat{\mathbf{U}}^H \mathbf{r}_t \right\|_2^2$  in which  $\hat{\mathbf{U}} \in \mathbb{C}^{N \times p}$  denotes the  $p$ -dimensional principal subspace estimate of the covariance  $\mathbf{R}$ . [11] proves that there exists a global minimum of  $J(\hat{\mathbf{U}})$ , and  $\hat{\mathbf{U}}$  spans the  $p$  principal eigenvectors of  $\mathbf{R}$  provided that Assumption 5 is true and  $\hat{\mathbf{U}}$  has full rank.

*Assumption 5:* The  $p$ th largest eigenvalue of  $\mathbf{R}$  has multiplicity one, that is,  $\lambda_p(\mathbf{R}) > \lambda_{p+1}(\mathbf{R})$ .

Hence, the principal subspace tracking problem can be cast into an unconstrained minimization problem. The gradient of  $J(\hat{\mathbf{U}})$  with respect to  $\hat{\mathbf{U}}$  is  $\nabla J = (-2\mathbf{R} + \mathbf{R}\hat{\mathbf{U}}\hat{\mathbf{U}}^H + \hat{\mathbf{U}}\hat{\mathbf{U}}^H\mathbf{R})\hat{\mathbf{U}}$ . Let Assumption 3 hold. The steepest descent method yields

$$\hat{\mathbf{U}}_t = \hat{\mathbf{U}}_{t-1} - \gamma_t \left( -2\mathbf{R} + \mathbf{R}\hat{\mathbf{U}}_{t-1}\hat{\mathbf{U}}_{t-1}^H + \hat{\mathbf{U}}_{t-1}\hat{\mathbf{U}}_{t-1}^H\mathbf{R} \right) \hat{\mathbf{U}}_{t-1}.$$

One can estimate the covariance matrix  $\mathbf{R}$  by its instantaneous one  $\mathbf{r}_t \mathbf{r}_t^H$  (Assumption 1). Hence, the subspace update equation using only the received data can be written as

$$\hat{\mathbf{U}}_t = \hat{\mathbf{U}}_{t-1} + \gamma_t \mathbf{D}_t \quad (16)$$

$$\text{where } \mathbf{D}_t = 2\mathbf{r}_t \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1} - \mathbf{r}_t \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1} \hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1}$$

$$- \hat{\mathbf{U}}_{t-1} \hat{\mathbf{U}}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1}. \quad (17)$$

Authors in [8] and [9] refer to this as Normalized Oja (NOja) because if  $\hat{\mathbf{U}}_t^H \hat{\mathbf{U}}_t = \mathbf{I}_p$ , the update (16) is identical to Oja's flow [41] for extracting  $p$ -principal component subspace.

Similarly to what we have done previously to decentralize Oja's learning rule, we want to decentralize the computation of NOja's algorithm by having each node compute only the corresponding row of a subspace estimate  $\hat{\mathbf{U}}_t$ . Consider the NOja's algorithm in (16) updated at each node:

$$\begin{aligned} [\hat{\mathbf{U}}_t]_{i,1:p} &= [\hat{\mathbf{U}}_{t-1}]_{i,1:p} \\ &\quad + \gamma_t \left( 2r_t(i) \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1} - r_t(i) \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1} \hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1} \right. \\ &\quad \left. - [\hat{\mathbf{U}}_{t-1}]_{i,1:p} \left[ \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1} \right]^2 \right) \end{aligned} \quad (18)$$

where  $[\hat{\mathbf{U}}_t]_{i,1:p}$  is the  $i$ th row of  $\hat{\mathbf{U}}_t$ . The NOja's update in a decentralized network requires distributively estimate the vector  $\mathbf{r}_t^H \hat{\mathbf{U}}_{t-1}$  and the Hermitian matrix  $\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1}$ . Both estimates can be achieved using the average consensus protocol described earlier with some simple modifications.

#### A. Average Consensus Protocol for Matrices

Suppose that  $\hat{\mathbf{U}}_{t-1} = [\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_p]$  are the  $p$  columns of the subspace estimate and  $[\hat{\mathbf{U}}_{t-1}]_{i,1:p}$  denotes the  $i$ th row of  $\hat{\mathbf{U}}_{t-1}$ , then  $\mathbf{r}_t^H \hat{\mathbf{U}}_{t-1} = \sum_{i=1}^N r_t^*(i) [\hat{\mathbf{U}}_{t-1}]_{i,1:p}$ . Let  $\mathbf{m}_0^{(i)} = r_t^*(i) [\hat{\mathbf{U}}_{t-1}]_{i,1:p}$  be the initial state vector at node  $i$ , and hence  $\mathbf{Z}_0 = \left[ \begin{matrix} (\mathbf{m}_0^{(1)})^T & \dots & (\mathbf{m}_0^{(N)})^T \end{matrix} \right]^T$  is the initial state matrix. Then the average consensus iteration performed at each node given a vector-valued input is

$$\mathbf{m}_k^{(i)} = [\mathbf{W}]_{i,i} \mathbf{m}_{k-1}^{(i)} + \sum_{(i,j) \in \mathcal{E}} [\mathbf{W}]_{i,j} \mathbf{m}_{k-1}^{(j)}. \quad (19)$$

Let Assumption 2 hold. The output  $\mathbf{Z}_K$  using (19) is

$$\mathbf{Z}_K = \mathbf{W}^K \mathbf{Z}_0 \xrightarrow{k \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N \mathbf{m}_0^{(j)} \mathbf{1}_N \quad (20)$$

where the  $i$ th row of  $\mathbf{Z}_K$  corresponds to the  $i$ th node's estimate of  $(1/N) \mathbf{r}_t^H \hat{\mathbf{U}}_{t-1}$ .

Moreover, of the matrix  $\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1}$ , the  $i$ th node only has access to  $[\hat{\mathbf{U}}_{t-1}]_{i,1:p}$ . Notice that each row of  $\hat{\mathbf{U}}^H \hat{\mathbf{U}}$  is a vector-matrix product of the form  $\hat{\mathbf{u}}_j^H \hat{\mathbf{U}}_{t-1}$ , which can be distributively estimated using (19) by replacing  $\mathbf{r}_t$  with  $\hat{\mathbf{u}}_j$  for  $j = 1, \dots, p$ . Hence, the matrix estimate of  $\hat{\mathbf{U}}^H \hat{\mathbf{U}}$  can be computed in parallel row-by-row using (19) and each row converges to the vector-matrix product  $\hat{\mathbf{u}}_j^H \hat{\mathbf{U}}_{t-1}$ . Or equivalently, since  $\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1} = \sum_{j=1}^N [\hat{\mathbf{U}}_{t-1}]_{j,1:p}^H [\hat{\mathbf{U}}_{t-1}]_{j,1:p}$ , then let  $\mathbf{M}_0^{(i)} = [\hat{\mathbf{U}}_{t-1}]_{i,1:p}^H [\hat{\mathbf{U}}_{t-1}]_{i,1:p}$  be the initial state input matrix at node  $i$ , then

$$\mathbf{M}_k^{(i)} = [\mathbf{W}]_{i,i} \mathbf{M}_{k-1}^{(i)} + \sum_{(i,j) \in \mathcal{E}} [\mathbf{W}]_{i,j} \mathbf{M}_{k-1}^{(j)}. \quad (21)$$

describes one update of the average consensus protocol performed at each node given a matrix-valued input.

#### Algorithm

Consider the problem of distributively computing the principal subspace of  $\mathbf{R}$  based on the NOja's update (16). The key idea is to decentralize the computation of (16) by having each node only updates its local subspace estimate. Upon receiving the sampled vector  $\mathbf{r}_t$  at time  $t$ , the proposed algorithm iteratively estimates  $\mathbf{r}_t^H \hat{\mathbf{U}}_{t-1}$  and  $\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1}$  as described in (19) and (21), respectively. After every node reaches an approximate consensus, it then updates its local estimate of  $\hat{\mathbf{U}}_t$  using (18).

Algorithm 2 summarizes the implementation for distributively tracking the principal subspace of  $\mathbf{R}$ . Here,  $\hat{\mathbf{U}}_0$  is an initial matrix with the property that  $\hat{\mathbf{U}}_0^H \hat{\mathbf{U}}_0 = \mathbf{I}_p$ . One simple construction of  $\hat{\mathbf{U}}_0$  is to set  $\hat{\mathbf{U}}_0$  to be one at its diagonal  $[\hat{\mathbf{U}}_0]_{i,i}$  for  $i = 1, \dots, p$  and zero everywhere else, and  $\text{AC}(\text{argument}, K)$

denotes the output of  $K$  average consensus iterations at each node given the initial state input argument. The argument can be a vector or a matrix, and the output is a vector/matrix of the same size.

---

#### Algorithm 2: p-D Subspace Tracking (at the $i$ th node)

---

**input:** A sequence of  $r_{i,t}$

**output:** The  $i$ th row of the principal subspace  $\hat{\mathbf{U}}$

- 1: **Initialization:**  $t = 0$ ;
  - 2:  $\hat{u}_{i,1:p} = 0, \hat{u}_{i,i} = 1$ ;
  - 3: **Recursion:**
  - 4: **foreach**  $t = 1 : T$  **do**
  - 5:      $\mathbf{a}c_{i,1:p} \leftarrow N \cdot \text{AC}(r_{i,t}^* \hat{u}_{i,1:p}; K)$ ;
  - 6:      $v_{i,1:p} \leftarrow r_{i,t} \cdot \mathbf{a}c_{i,1:p}$ ;
  - 7:      $v_{i,1:p} \leftarrow 2v_{i,1:p} - N v_{i,1:p} \cdot \text{AC}(\hat{u}_{i,1:p}^H \hat{u}_{i,1:p}; K)$ ;
  - 8:      $v_{i,1:p} \leftarrow v_{i,1:p} - \hat{u}_{i,1:p} (\mathbf{a}c_{i,1:p}^H \mathbf{a}c_{i,1:p}; K)$ ;
  - 9:      $\hat{u}_{i,1:p} \leftarrow \hat{u}_{i,1:p} + \gamma_t v_{i,1:p}$ ;
  - 10: **end**
- 

1) *Property 2:* Let  $\mathbf{v} \in \mathbb{R}^N, \hat{\mathbf{U}} \in \mathbb{C}^{N \times p}$ , and denote by  $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{N \times N}$  Hermitian matrices, then [38]:

- 1)  $[\mathbf{A} \circ (\mathbf{B} \hat{\mathbf{U}} \hat{\mathbf{U}}^H)]_{i,1:N} = [\mathbf{B}]_{i,1:N} \hat{\mathbf{U}} \hat{\mathbf{U}}^H \text{diag}([\mathbf{A}]_{i,1:N})$ ;
- 2) if  $\mathbf{A}$  is positive semidefinite,  $\|\mathbf{A} \odot \mathbf{B}\|_F \leq \max[\mathbf{A}]_{i,i} \|\mathbf{B}\|_F$ ;
- 3)  $\|\mathbf{A}^{-1} \text{diag}(\mathbf{v}) \mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^{-2} \text{diag}(\mathbf{v}) \mathbf{A}^2 \text{diag}(\mathbf{v})) \leq \sum_{i=1}^N |v_{i,i}| \leq |v_{\max}| \text{tr}(|\mathbf{v}|)$ ;
- 4)  $\|\hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F \leq \|\hat{\mathbf{U}}\|_F \|\hat{\mathbf{U}}^H\|_F = \text{tr}(\hat{\mathbf{U}}^H \hat{\mathbf{U}})$ .

The remaining part of this section is dedicated to study the convergence properties of Algorithm 2. In between each sampling interval  $[t-1, t)$ , node  $i$  distributively computes a local subspace estimate  $[\hat{\mathbf{U}}_{t-1}]_{i,1:p}$  until at time  $t$  when a new data sample  $r_t(i)$  is received. The initial state matrix to compute the vector-matrix product  $\mathbf{r}_t^H \hat{\mathbf{U}}_{t-1}$  is  $\mathbf{Z}_0 = \text{diag}(\mathbf{r}_t^*) \hat{\mathbf{U}}_{t-1}$ . Using (20), the distributed estimate of  $\mathbf{r}_t^H \hat{\mathbf{U}}_{t-1}$  is

$$N \mathbf{W}^K \mathbf{Z}_0 = N \mathbf{W}^K \text{diag}(\mathbf{r}_t^*) \hat{\mathbf{U}}_{t-1}. \quad (22)$$

Moreover, the distributed estimate for  $\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1}$  at node  $i$  is

$$\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1} \rightarrow \begin{bmatrix} [N \mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{u}_1^*) \hat{\mathbf{U}}_{t-1} \\ \vdots \\ [N \mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{u}_p^*) \hat{\mathbf{U}}_{t-1} \end{bmatrix} \quad (23)$$

(see Appendix B-A for a detailed derivation). Combining the results from (22) and (23) yields

$$\hat{\mathbf{U}}_t = \hat{\mathbf{U}}_{t-1} + \gamma_t \mathbf{D}_{t,K}, \text{ where} \\ \mathbf{D}_{t,K} = \left[ N \mathbf{W}^K \odot \left[ 2 \mathbf{r}_t \mathbf{r}_t^H - (N \mathbf{W}^K \odot (\mathbf{r}_t \mathbf{r}_t^H)) \hat{\mathbf{U}}_{t-1} \hat{\mathbf{U}}_{t-1}^H - (N \mathbf{W}^K \odot (\hat{\mathbf{U}}_{t-1} \hat{\mathbf{U}}_{t-1}^H)) (\mathbf{r}_t \mathbf{r}_t^H) \right] \right] \hat{\mathbf{U}}_{t-1}. \quad (24)$$

(see Appendix B-B).

#### B. Stability Analysis

The asymptotic properties of the proposed algorithm are the properties of the ODE determined by the "mean" dynamics

of the algorithm. Under certain conditions, we will show that the magnitude of  $\|I - \hat{U}^H \hat{U}\|_F$  diminishes. Furthermore, the matrix-valued estimate  $\hat{U}_t$  converges to an equilibrium point, which, for a finite value of  $K$ , stays in the neighborhood of the principal subspace of  $R$ .

Let Assumption 1, 3, 4 hold. The ODE in [40] of (24) is

$$\begin{aligned} \frac{d\hat{U}}{dt} &= \mathbb{E}\{D_{t,k}\} \\ &= \left\{ N\bar{W}^K \odot \left[ 2R - \left( N\bar{W}^K \odot R \right) \hat{U}_{t-1} \hat{U}_{t-1}^H \right. \right. \\ &\quad \left. \left. - \left( N\bar{W}^K \odot \left( \hat{U}_{t-1} \hat{U}_{t-1}^H \right) \right) R \right] \right\} \hat{U}_{t-1} \end{aligned}$$

where  $\bar{W}^K = \mathbb{E}[W^K] = W^K$ , since the network is assumed to be static. Let  $X = W - J$ , then the preceding expression can be written as

$$\begin{aligned} \frac{d\hat{U}}{dt} &= 2R\hat{U} \\ &\quad - R\hat{U}\hat{U}^H\hat{U} - \hat{U}\hat{U}^H R\hat{U} + \tilde{\Delta}_{\text{ODE}}\hat{U} \end{aligned} \quad (25)$$

where

$$\begin{aligned} \tilde{\Delta}_{\text{ODE}} &= 2NX^K \odot R - (NX^K \odot R)\hat{U}\hat{U}^H \\ &\quad - NX^K \odot R\hat{U}\hat{U}^H - NX^K \odot \hat{U}\hat{U}^H R \\ &\quad - (NX^K \odot \hat{U}\hat{U}^H)R + \mathcal{O}(N^2\lambda_2^{2K}(W)) \end{aligned} \quad (26)$$

with  $\tilde{\Delta}_{\text{ODE}}$  being the trajectory deviation from the decentralized ODE to the centralized ODE of  $\hat{U}$ .

Similarly to the principal eigenvector estimation problem, we want to study the trajectory of  $\hat{U}^H \hat{U}$  to learn the stability of (25). The trajectory of  $\hat{U}^H \hat{U}$  is

$$\begin{aligned} \frac{d\hat{U}^H \hat{U}}{dt} &= \hat{U}^H \frac{d\hat{U}}{dt} + \frac{d\hat{U}^H}{dt} \hat{U} \\ &= 2\hat{U}^H R\hat{U} \left( I - \hat{U}^H \hat{U} + \tilde{\delta}_{\text{ODE}} \right) \\ &\quad + 2 \left( I - \hat{U}^H \hat{U} + \tilde{\delta}_{\text{ODE}} \right)^H \hat{U}^H R\hat{U}. \end{aligned}$$

$$\begin{aligned} \text{where } \tilde{\delta}_{\text{ODE}} &= \frac{1}{2}(\hat{U}^H R\hat{U})^{-1} \left( \hat{U}^H \tilde{\Delta}_{\text{ODE}} \hat{U} \right) \\ &= \frac{1}{2}(\hat{U}^H \hat{U})^{-1} \left( \hat{U}^H R^{-1} \tilde{\Delta}_{\text{ODE}} \hat{U} \right). \end{aligned} \quad (27)$$

$\tilde{\delta}_{\text{ODE}}$  represents the deviation of the decentralized approximation of  $\hat{U}^H \hat{U}$  from its centralized approximation. The ODE expression implies that  $(I + \tilde{\delta}_{\text{ODE}})$  is the only equilibrium point and thus  $\hat{U}_{\text{eq}}^H \hat{U}_{\text{eq}} = I + \tilde{\delta}_{\text{ODE}}$ . The size of  $\tilde{\delta}_{\text{ODE}}$  is influenced by the overall network's connectivity. By increasing the number of average consensus iterations in the network, the norm of  $\tilde{\delta}_{\text{ODE}}$  diminishes as stated in the next lemma.

**Lemma 5.1 (Stability of  $\hat{U}^H \hat{U}$ ):** Suppose Assumption 1, 2, 3, and 5 hold. The ODE of  $\hat{U}^H \hat{U}$  converges asymptotically to a neighborhood of  $I_p$  if the number of average consensus iterations  $K$  is sufficiently large. Specifically, the difference between  $\hat{U}^H \hat{U}$  and  $I_p$  decreases exponentially with  $K$ , that is  $\|I - \hat{U}^H \hat{U}\|_F \leq \mathcal{O}(N^{3/2}\lambda_2^K(W))$ .

*Proof:* See Appendix C-A.  $\blacksquare$

We propose to use Lyapunov stability analysis to argue the convergence of *Algorithm 2*. The Lyapunov stability analysis is based on the concept of energy, denoted by a Lyapunov function candidate (LFC)  $V(\mathbf{x}, t)$ . When a system reaches its equilibrium point, the total energy  $V(\mathbf{x}, t)$  is zero or at its minimum. Hence, we choose to define a LFC as  $V(\hat{U}) = \mathbb{E} \left\| \left( I - \hat{U}_t \hat{U}_t^H \right) \mathbf{r}_t \right\|_2^2 = \text{tr} \left( \left( I - \hat{U} \hat{U}^H \right) R \left( I - \hat{U} \hat{U}^H \right) \right)$ , which is the expected power of the received vector  $\mathbf{r}_t$  projected onto the estimated noise space  $I - \hat{U} \hat{U}^H$ . If the time evolution of  $V(\hat{U})$  is negative except at  $\hat{U} = \hat{U}_{\text{eq}}$ , then at the equilibrium the total expected power of  $\mathbf{r}_t$  projected onto the space  $I - \hat{U}_{\text{eq}} \hat{U}_{\text{eq}}^H$  is at its minimum. The time evolution of the Lyapunov function  $V(\hat{U})$  of the proposed algorithm is

$$\frac{dV(\hat{U})}{dt} = \text{tr} \left( \frac{dV^H(\hat{U})}{d\hat{U}} \times \frac{d\hat{U}}{dt} \right),$$

$$\text{where } \frac{dV(\hat{U})}{d\hat{U}} = - \underbrace{\left[ R(I - \hat{U} \hat{U}^H) + (I - \hat{U} \hat{U}^H)R \right]}_{\Phi}$$

$$\hat{U} = -\Phi \hat{U},$$

$$\frac{d\hat{U}}{dt} = \Phi \hat{U} + \tilde{\Delta}_{\text{ODE}} \hat{U}.$$

By a change of variable as in (40), we have  $\tilde{\Delta}_{\text{ODE}} = N\lambda_2^K(W)\tilde{\Delta}(\hat{U})$ . It is expected that for  $K$  sufficiently large, the derivative of the LFC function

$$\begin{aligned} g(V(\hat{U})) &= \frac{dV(\hat{U})}{dt} \\ &= -\text{tr} \left[ (\Phi \hat{U})^H (\Phi \hat{U}) \right] \\ &\quad - N\lambda_2^K(W) \text{tr} \left[ \hat{U}^H \Phi^H \tilde{\Delta} \hat{U} \right] \end{aligned}$$

is a perturbation of the following dynamical system

$$f(V(\hat{U})) = -\text{tr} \left[ (\Phi \hat{U})^H (\Phi \hat{U}) \right] \leq 0.$$

In particular,  $|f(V(\hat{U})) - g(V(\hat{U}))|$  and  $\|Df(V(\hat{U})) - Dg(V(\hat{U}))\|$  are small for any  $V(\hat{U})$  with  $\hat{U}^H \hat{U}$  in a neighborhood of  $I_p$ . Consider  $f(V(\hat{U}))$  as an unperturbed time evolution of the Lyapunov function  $V(\hat{U})$ . Since it is always less or equal to zero, the ‘‘energy’’  $V(\hat{U})$  is a monotonically decreasing function until  $f(V(\hat{U})) = 0$  (i.e.,  $V(\hat{U})$  reaches its minimum), or  $\Phi^H \hat{U} = 0$ . Yang in [11] proves that the subspace estimate  $\hat{U}$  spans the  $p$  largest eigenvalues of  $R$  if and only if when  $V(\hat{U})$  is at its minimum (i.e.,  $\Phi^H \hat{U} = 0$  for  $\hat{U}^H \hat{U} = I_p$ ). We denote this equilibrium point by  $\hat{U}_f$  and thus  $\hat{U}_f \in [\mathbf{U}]$ . Theorem 4.1 says that if  $g(V(\hat{U}))$  is a perturbation of  $f(V(\hat{U}))$ , then there exists a unique equilibrium point  $V^* = V(\hat{U}_{\text{eq}})$  of  $g(V(\hat{U}))$  and it is in a neighborhood of the equilibrium point  $V(\hat{U}_f)$  of  $f(\hat{U})$ . Therefore, for  $K$  large enough, or in other words, when  $g(V(\hat{U}))$  is a perturbation of  $f(V(\hat{U}))$ , the value  $V^*$  is close to  $V(\hat{U}_f)$ , and their difference decreases with an increase in  $K$ . Hence, the time evolution of  $V(\hat{U})$  governed by the perturbed system  $g(V(\hat{U}))$  also decreases until it reaches to  $V^*$ , which sits in a neighborhood of  $V(\hat{U}_f)$ . Using the result from the perturbation theory, there exists an equivalence class

$[\hat{\mathbf{U}}_{\text{eq.}}]$  such that the cost function  $V(\hat{\mathbf{U}}_{\text{e.q.}}) = V^*$  is close to its minimum for  $\hat{\mathbf{U}}_{\text{e.q.}} \in [\hat{\mathbf{U}}_{\text{e.q.}}]$ , and it is anticipated that  $[\hat{\mathbf{U}}_{\text{eq.}}]$  is in a neighborhood of the true principal subspace  $\hat{\mathbf{U}}_f \in [\mathbf{U}]$  of  $\mathbf{R}$  within a radius that decreases as  $K$  increases.

## VI. PRINCIPAL SUBSPACE TRACKING VIA RANDOMIZED GOSSIPING

Up to this point, we have considered the distributed subspace tracking problem in a static network using a distributed average consensus protocol, where the consensus updates are synchronous. This section, instead, considers a randomized gossip algorithm [16] with asynchronous updates for data diffusion. Specifically, data are still synchronously sampled, but in the time that elapses between two sampling times  $[t, t+1)$ , information is processed asynchronously. Hence, there is a separation of time scales. Message exchanges among nodes are triggered by random events that are modeled as a Poisson process at a rate  $\rho/N$ . When a clock ticks, the node randomly selects one of its neighbors to exchange values with. The main objective of such relaxation (i.e., from synchronous to asynchronous update) is to show that we do not need to perform an accurate scheduling of local communication among sensors for the algorithms to converge. Connections to the algorithms based on a deterministic and synchronous gossiping for data fusion are also highlighted.

### A. Randomized Gossip Algorithm Without Collisions or Link Failures

Consider a particular class of gossiping algorithm using an asynchronous time model, in which each node contacts and exchanges information with no more than one of its neighbors when its internal clock ticks. This class of gossiping protocol has received a great attention recently [12], [13], [16]–[18] for numerous reasons. First of all, communication is simple and there is no need to store and maintain routing information. It is resilient to changes in network topology. The overall performance of the network is not compromised by eliminating the fusion center.

Suppose that a clock attached to the  $i$ th node ticks at an arbitrary time  $\tau_k$ . Then node  $i$  picks a neighbor  $j$  to exchange data  $z_{k-1}(i)$  and  $z_{k-1}(j)$  with and then updates  $z_k(i) = z_k(j) = [z_{k-1}(i) + z_{k-1}(j)]/2$ , where  $k$  is the index counting the number of pairwise exchanges occurred. The network repeats the same process at time  $\tau_{k+1}$  when another clock ticks. One can model the network as an undirected graph  $\mathcal{G}_k = (\mathcal{N}, \mathcal{E}_k)$ , where  $\mathcal{E}_k$  represents a connected link at time  $\tau_k$ . Furthermore, denote by  $\mathcal{E}(i)$  a collection of all possible edges that node  $i$  can connect with. If  $\mathcal{E}_k$  describes one pair-wise connection associated with node  $i$  at time  $\tau_k$ , then  $\mathcal{E}_k$  is a subset of  $\mathcal{E}(i)$ . The randomness of a link is modeled by assuming that node  $i$ 's clock ticks at an arbitrary time with a probability  $1/N$  and node  $i$  picks node  $j$  with some probability, that is

$$P_{i,j} = \begin{cases} \text{Prob. of node } i \text{ chooses node } j, & (i,j) \in \mathcal{E}(i) \\ 0, & (i,j) \notin \mathcal{E}(i) \end{cases}. \quad (28)$$

Since the available link  $\mathcal{E}_k$  varies with time  $\tau_k$ , the weight matrix is, hence, time-varying and we denote it by  $\mathbf{W}_k$ . Let  $\overline{\mathbf{W}}$  be

the expected value of  $\mathbf{W}_k$ , then [16] has shown that  $\overline{\mathbf{W}}$  is a symmetric positive-semidefinite doubly stochastic matrix and it can be expressed as  $\overline{\mathbf{W}} = \mathbf{I} - (1/2N)\mathbf{D} + \mathbf{P} + (\mathbf{P}^T/2N)$ , where  $\mathbf{D}$  is a diagonal matrix with entries  $[\mathbf{D}]_{i,i} = \sum_{j=1}^N (P_{i,j} + P_{j,i})$ . In addition, its second largest eigenvalue equals to  $\lambda_2(\overline{\mathbf{W}}) = 1 - (1/N)(1 - \lambda_2(\mathbf{P}))$ . Suppose the network is able to construct a probability matrix  $\mathbf{P}$  with  $|\lambda_2(\mathbf{P})| < 1$ , then  $0 \leq \lambda_2(\overline{\mathbf{W}}) < 1$ . It turns out that the randomized gossip algorithm will converge to the mean value in expectation if the following assumption is satisfied.

*Assumption 6:* For a given undirected network graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E}_k)$  and its associated expected weight matrix  $\overline{\mathbf{W}} \in \mathbb{R}^{N \times N}$ , assume (1) the graph  $\mathcal{G}$  is connected in expectation, (2)  $\overline{\mathbf{W}}$  is doubly stochastic, and (3)  $\lambda_2(\overline{\mathbf{W}}) < 1$ .

### Algorithm

The subspace tracking algorithm via randomized gossiping follows almost the same procedure as in *Algorithm 1* and *Algorithm 2*. The only difference is the way to compute the average at the *consensus* step,  $AC(\text{argument}, K)$ . Specifically, instead of simultaneously taking a weighted average of all the neighbors' data, the randomized gossip algorithm connects two nodes at an arbitrary time, exchanges their values pair-wisely and then updates accordingly.

The remaining part of this section studies the convergence properties of the proposed algorithms via the randomized gossip scheme. Recall that, in a static network, the convergence properties of *Algorithm 1* and *Algorithm 2* are determined by the second largest eigenvalue of the matrix  $\mathbb{E}[\mathbf{W}_1 \cdots \mathbf{W}_K]$ . Since  $\mathbf{W}_k$  for any  $1 \leq k \leq K$  is deterministic and it satisfies Assumption 2, we get

$$\lim_{K \rightarrow \infty} \lambda_2(\mathbb{E}[\mathbf{W}_1 \cdots \mathbf{W}_K]) = \lim_{K \rightarrow \infty} \lambda_2^K(\mathbf{W}) = 0. \quad (29)$$

As  $K$  increases, the error introduced by the decentralized network diminishes. Hence, if we can obtain a similar result for the expected weight matrix of the randomized gossip algorithm, the convergence results will hold.

In the randomized gossiping scheme with asynchronous updates, we have  $\overline{\mathbf{W}}^{K_t}$  equal to

$$\mathbb{E} \left[ \prod_{k=1}^{K_t} \mathbf{W}_k | K_t \right] = \mathbb{E}[\mathbf{W}_{K_t} \mathbf{W}_{K_t-1} \cdots \mathbf{W}_2 \mathbf{W}_1 | K_t] \quad (30)$$

because the weight matrices are independent and identically distributed (i.i.d.). Following the same methodology as in the deterministic case, prior to perform an *update* at time  $t$ , the proposed algorithms need to first reach a *consensus* on the unknown parameters using the randomized gossip algorithm for  $K_t$  number of iterations. In particular,  $K_t$  is a Poisson distributed random variable because data exchanges among nodes are modeled as a Poisson process with a rate parameter  $\rho$ . Let  $T_s$  be the sampling time. Then the probability of  $K_t$  is  $p(K_t | T_s) = e^{-\rho T_s} (\rho T_s)^{K_t} / K_t!$ , because the number of clock tickings (i.e., message exchanges) in a fixed time interval follows a Poisson distribution. Given that  $\mathbf{W}_k$  are i.i.d., and the EVD of the expected weight matrix is  $\overline{\mathbf{W}} = \sum_{i=1}^N \lambda_i(\overline{\mathbf{W}}) \mathbf{v}_i \mathbf{v}_i^T$ , then we have

$$\mathbb{E} \left\{ \prod_{k=1}^{K_t} \mathbf{W}_k | T_s \right\} = \mathbb{E}_{K_t} \left\{ \overline{\mathbf{W}}^{K_t} | K = K_t, T_s \right\}$$



$$\begin{aligned}
&= \sum_{K=0}^{\infty} p(K = K_t | T_s) \overline{\mathbf{W}}^K \\
&= \sum_{i=1}^N \sum_{K=0}^{\infty} \frac{e^{-\rho T_s} (\rho T_s)^K}{K!} \lambda_i^K(\overline{\mathbf{W}}) \overline{\mathbf{v}}_i \overline{\mathbf{v}}_i^T \\
&= \sum_{i=1}^N e^{-\rho T_s (1 - \lambda_i(\overline{\mathbf{W}}))} \overline{\mathbf{v}}_i \overline{\mathbf{v}}_i^T. \quad (31)
\end{aligned}$$

Hence, the eigenvectors of  $\mathbb{E} \left\{ \prod_{\ell=1}^{K_t} \mathbf{W}(\ell) \right\}$  are identical to the eigenvectors of  $\overline{\mathbf{W}}$ , while the eigenvalues are exponential functions of  $\lambda_i(\overline{\mathbf{W}})$ , that is  $\lambda_i \left( \mathbb{E} \left\{ \prod_{k=1}^{K_t} \mathbf{W}_k | T_s \right\} \right) = e^{-\rho T_s (1 - \lambda_i(\overline{\mathbf{W}}))} > 0$ , for  $i = 1, \dots, N$ . Observe that  $\mathbb{E} \left\{ \prod_{k=1}^{K_t} \mathbf{W}_k | T_s \right\}$  is a positive-definite matrix and its eigenvalues monotonically decrease with the eigenvalues of  $\overline{\mathbf{W}}$ . Specifically, its largest eigenvalue equals to 1 and all the remaining eigenvalues are strictly less than 1. Furthermore, suppose the sampling time  $T_s$  is fixed, then

$$\lim_{\rho \rightarrow \infty} \lambda_2 \left( \mathbb{E} \left\{ \prod_{k=1}^{K_t} \mathbf{W}_k | T_s \right\} \right) = \lim_{\rho \rightarrow \infty} (e^{-\rho T_s}) e^{\rho T_s \lambda_2(\overline{\mathbf{W}})} = 0.$$

The above result is analogous to the relation in (29) of the deterministic gossiping scheme. Hence, it is anticipated that for  $\rho$  sufficiently large, the algorithms using the randomized gossip protocol also converge to the true principal component and the principal subspace of  $\mathbf{R}$ .

### B. Relation Between $K$ and $\rho$

From the earlier derivation, we have obtained  $\mathbb{E} \left\{ \prod_{k=1}^{K_t} \mathbf{W}_k | T_s \right\} = \sum_{i=1}^N e^{-\rho T_s (1 - \lambda_i(\overline{\mathbf{W}}))} \overline{\mathbf{v}}_i \overline{\mathbf{v}}_i^T$  using the randomized gossip protocol with asynchronous updates for data fusion. On the other hand, in the deterministic gossip scheme with synchronous updates,  $\mathbb{E} \left\{ \prod_{k=1}^K \mathbf{W} \right\} = \mathbf{W}^K = \sum_{i=1}^N \lambda_i^K(\mathbf{W}) \mathbf{v}_i \mathbf{v}_i^T$ . We recall that the rate of convergence is determined by the second largest eigenvalue of the corresponding (expected) weight matrix. To find a relation between  $K$  and  $\rho$ ,  $\lambda_2 \left( \mathbb{E} \left\{ \prod_{k=1}^{K_t} \mathbf{W}_k | T_s \right\} \right)$  should be approximately equal to  $\lambda_2 \left( \mathbb{E} \left\{ \prod_{k=1}^K \mathbf{W} \right\} \right)$ , that is  $e^{-\rho T_s (1 - \lambda_2(\overline{\mathbf{W}}))} = \lambda_2^K(\mathbf{W})$ . By taking the natural log of both sides of this equality and by rearranging the terms, we get the following relation between  $K$  and  $\rho$ :

$$\rho = \frac{\ln \lambda_2^{-1}(\mathbf{W})}{T_s (1 - \lambda_2(\overline{\mathbf{W}}))} K. \quad (32)$$

*Lemma 6.1:* Suppose Assumption 1, 3, 5, and 6 hold. Let  $T_s$  denote the sampling time, and  $\rho$  represent the rate parameter of the Poisson process in a network of  $N$  sensors, where we assume each clock exhibits a rate  $\rho/N$  Poisson process.  $\hat{\mathbf{U}}_{\text{eq}}^H \hat{\mathbf{U}}_{\text{eq}}$  is in a neighborhood of  $\mathbf{I}_p$ , that is  $\left\| \mathbf{I} - \hat{\mathbf{U}}^H \hat{\mathbf{U}} \right\|_F \leq \mathcal{O}(N^{3/2} e^{-\rho})$ , which decreases with  $\rho$ .

*Proof:* Simply replace the value of  $K$  in Lemma 5.1 with the value of  $\rho$  using the relation in (32). ■

The convergence property of the subspace estimate will still hold provided that Assumption 6 holds.

1) *Randomized Gossip Algorithm With Random Collisions and Link Failures:* Consider now there exists random (temporary) link failures in the network. Collisions are also modeled as random link failures. Assume that any link in  $\mathcal{E}(i)$ , for  $i = 1, \dots, N$ , may fail independently from any other link with a constant probability  $P_f$  across the network. Therefore, the number of pair-wise data exchanges in a time interval  $T_s$  is determined by the actual number of successful links, instead of the number of clock tickings  $K_t$ . Since  $P_f$  is assumed to be a constant for all links, it is equivalent to say that the averaged rate associated with each Poisson clock is decreased by a factor of  $(1 - P_f)$ . Hence, the new rate parameter  $\tilde{\rho}$  equals to the probability of success multiplied by  $\rho$ , that is

$$\tilde{\rho} = (1 - P_f) \rho \leq \rho. \quad (33)$$

Smaller value of  $\tilde{\rho}$  as compared to  $\rho$  implies that the network is less connected in response to random (temporary) link failures.

## VII. NUMERICAL RESULTS

The simulated network includes ten distributed sensors, and the network is generated as a Random Geometric graph [43] over a unit area with radius  $r = 0.6$ . The sensor field is formed as a linear combination of  $p$  narrowband spatial signals corrupted by white Gaussian noise (AWGN) at 20-dB SNR.  $\mathbf{U}^\perp$  denotes the space orthogonal to the signal subspace  $\mathbf{U}$ . All the simulations are compared with the centralized schemes and the step-size  $\gamma_t$  is set to be 0.001 for all  $t$ .

### Algorithms' Performances Using a Deterministic Gossip Algorithm

1) *Principal Eigenvector Estimation:* Fig. 3 illustrates the performance of the proposed eigenvector estimation algorithm by examining the trajectories of  $\left\| \hat{\mathbf{U}}_t^H \hat{\mathbf{u}}_t - 1 \right\|_2$ ,  $\cos(\theta) = \left\| (\mathbf{U}^\perp)^H \hat{\mathbf{u}}_t \right\|_2$ , and  $\text{MSE} = \left\| \hat{\mathbf{u}}_t - \mathbf{u}_1 \right\|_2$ , respectively. From our analysis, we learned that in order to achieve a similar performance, the number of average consensus iterations  $K$  should increase with the size of a network and decrease with a network's connectivity. In our simulation, we set  $K = 3$ . That is, the algorithm performs three sub-iterations of average consensus protocol for data diffusion in between samples  $t$  and  $t + 1$ . Fig. 3(a) shows that the norm of the principal eigenvector estimate asymptotically approaches to  $1 \pm 0.001$ . Fig. 3(b) indicates that  $\hat{\mathbf{u}}_t$  is orthogonal to the noise space  $\mathbf{U}^\perp$  over the time. The MSE as compared with  $\mathbf{u}_1$  diminishes as shown in Fig. 3(b). Overall, the decentralized eigenvector estimation algorithm shows a similar performance as the centralized algorithm.

2) *Principal Subspace Estimation:*  $\hat{\mathbf{u}}_t^{(i)}$  denotes the  $i$ th column of  $\hat{\mathbf{U}}_t$  with dimension 4. The performance of the proposed distributed subspace tracking algorithm is illustrated in Fig. 4 via examining the trajectories of  $\left\| \hat{\mathbf{U}}_t^H \hat{\mathbf{U}}_t - \mathbf{I} \right\|_F$ ,  $\cos(\theta) = \sum_{i=1}^4 \left\| (\mathbf{U}^\perp)^H \hat{\mathbf{u}}_t^{(i)} \right\|_2$ , and  $\text{MSE} = \mathbb{E} \left\| \mathbf{r}_t - \hat{\mathbf{U}}_t \hat{\mathbf{U}}_t^H \mathbf{r}_t \right\|_2$ , respectively, for  $K = \infty$  (centralized network),  $K = 5$  and  $K = 15$ . Fig. 4(a) indicates that the vectors in  $\hat{\mathbf{U}}_t$  are approximately orthonormal. Fig. 4(b) shows that  $\hat{\mathbf{U}}_t$  becomes orthogonal to the noise space

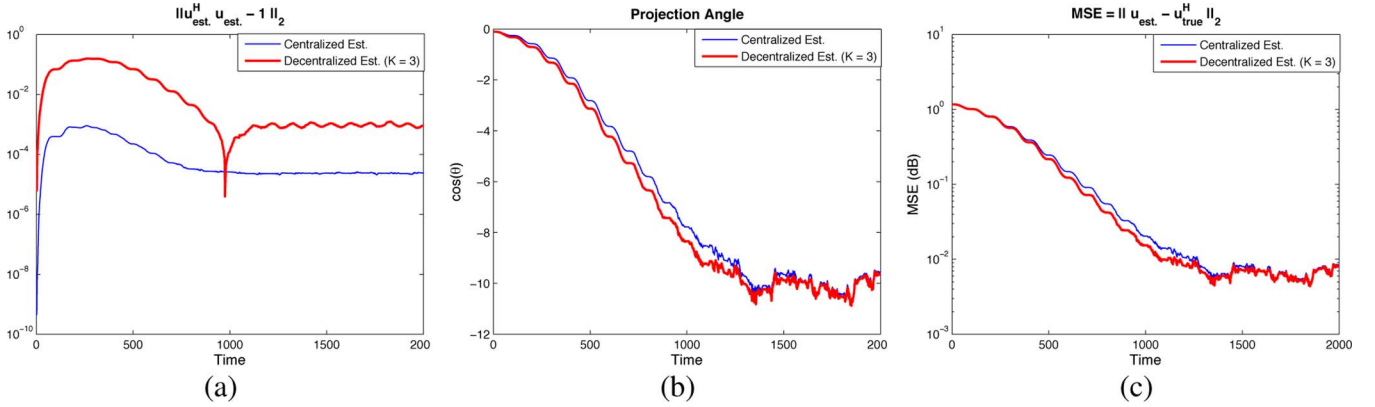


Fig. 3. (a) Trajectory of  $\|\hat{\mathbf{u}}_t^H \mathbf{u}_t - 1\|_2$ . (b) Projection angle between  $\hat{\mathbf{u}}_t$  and  $\mathbf{U}^\perp$ . (c) MSEs.

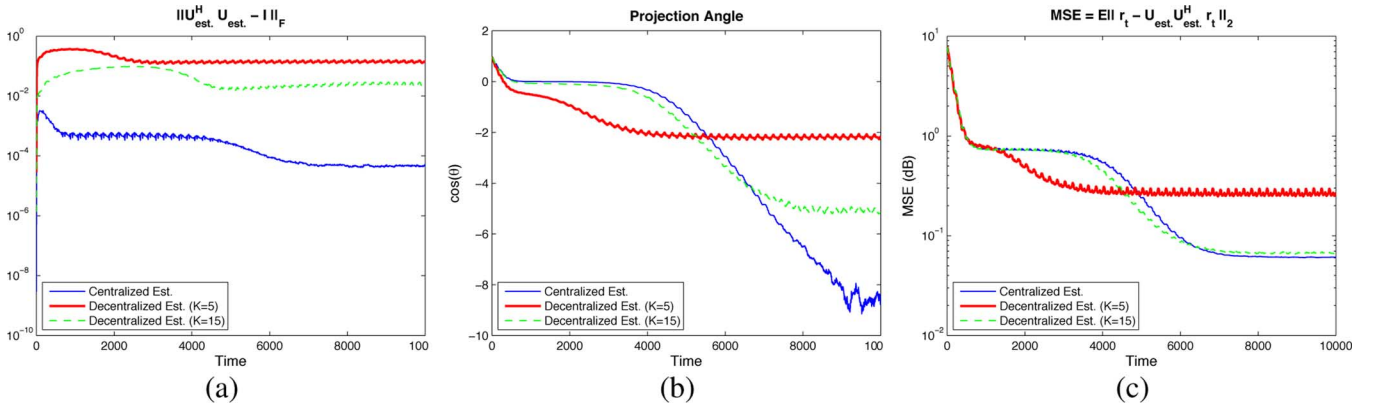


Fig. 4. (a) Trajectories of  $\|\hat{\mathbf{U}}_t^H \hat{\mathbf{U}}_t - \mathbf{I}\|_F$ . (b) Projections between the  $\hat{\mathbf{U}}_t$  and  $\mathbf{U}^\perp$ . (c) MSEs.

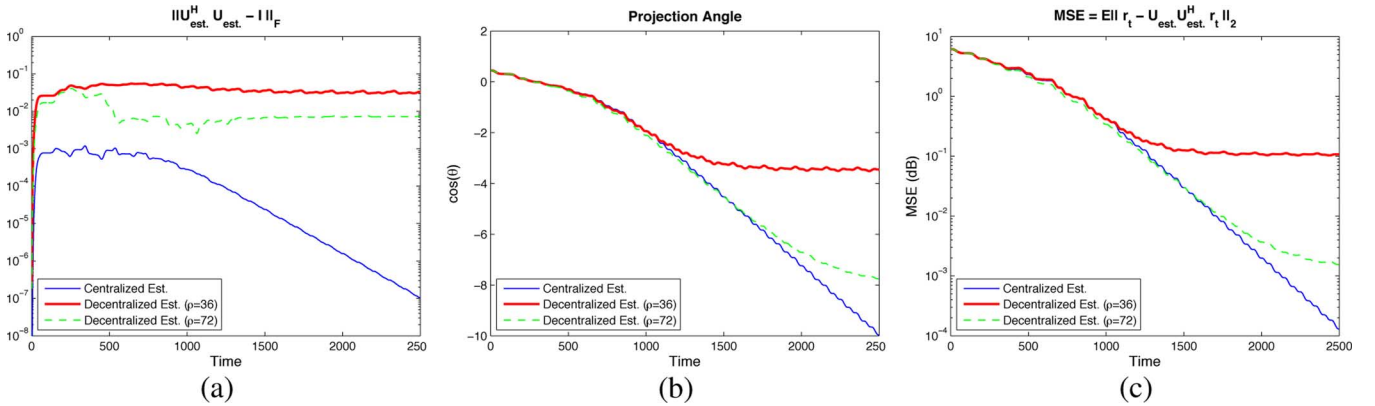


Fig. 5. (a) Trajectories of  $\|\hat{\mathbf{U}}_t^H \hat{\mathbf{U}}_t - \mathbf{I}\|_F$ . (b) Projections of  $\hat{\mathbf{U}}_t$  onto the space  $\mathbf{U}^\perp$ . (c) MSEs.

over the time. The MSE is about 0.1 as shown in Fig. 4(c). The distributed algorithm is observed to converge faster at the beginning and then reaches a plateau because of the error introduced from the distributed computation in the network. The number of consensus iterations performed at each time also increases as compared to the case of estimating a single eigenvector.

#### Algorithm's Performance Using a Randomized Gossip Algorithm

The rate parameter  $\rho$  to perform the randomized gossip protocol is set to be  $\rho = 36$  and  $\rho = 72$ , which correspond to  $K = 5$  and  $K = 10$ , respectively, in a deterministic scheme. Hence, between samples  $t$  and  $t + 1$ , each sensor performs on

average 3.6 and 7.2 sub-iterations of pair-wise data exchanges. Assuming no link or node failure, a similar performance of the algorithm as compared with the deterministic case is observed in Fig. 5.

#### VIII. CONCLUSION

This paper proposes a distributed principal eigenvector and a distributed principal subspace tracking algorithm based on Oja's learning rule [2], [8] and NOja's update equation, respectively. The key idea is to enable an in-network computation algorithm via near-neighbor communication without the need to set routes to forward the data to a fusion center. From the stability analysis, we have showed that given a sufficiently large number of consensus iterations  $K$ , the system is self-stabilizing

( $\mathbf{U}^H \mathbf{U} \rightarrow \mathbf{I} + \mathcal{O}(N^{3/2} \lambda_2^K(\mathbf{W}))$ ) and furthermore, perturbation theorem implies that there exists a stable equilibrium in a neighborhood of the true subspace. Furthermore, we explore the convergence properties of the proposed algorithms using a randomized gossip algorithm for data diffusion. Connections of these algorithms using a randomized gossip scheme to the ones using a deterministic gossip protocol are also highlighted.

#### APPENDIX A

*Proof of Lemma 4.1:*

*Proof:* Since  $(\lambda_2(\mathbf{W}), \mathbf{v}_2)$  is the largest eigenpair of  $\mathbf{X}$ , where  $\mathbf{X} = \mathbf{W} - \mathbf{J}$  and  $\mathbf{J} = (1/N)\mathbf{1}\mathbf{1}^H$ , the approximation  $\mathbf{X} \approx \lambda_2(\mathbf{W})\mathbf{v}_2\mathbf{v}_2^T$  holds. Also,  $\mathbf{v}_2\mathbf{v}_2^T$  is a rank-one matrix with an eigenvalue equals to 1. Hence, its spectral radius is  $\rho(\mathbf{v}_2\mathbf{v}_2^T) = 1$ . Using Property 1.3, we obtain the following approximation:

$$\begin{aligned} \Delta_{\text{ODE}} \hat{\mathbf{u}} &\approx N\lambda_2^K(\mathbf{W}) \left[ (\mathbf{v}_2\mathbf{v}_2^T) \odot \mathbf{R} - (\mathbf{v}_2\mathbf{v}_2^T) \right. \\ &\quad \left. \odot (\mathbf{u}\mathbf{u}^H \mathbf{R}) - \left( (\mathbf{v}_2\mathbf{v}_2^T) \odot (\hat{\mathbf{u}}\hat{\mathbf{u}}^H) \right) \mathbf{R} \right] \hat{\mathbf{u}} \\ &= N\lambda_2^K(\mathbf{W}) \left( (\mathbf{v}_2\mathbf{v}_2^T) \odot \mathbf{R} \right) \hat{\mathbf{u}} \\ &\quad - 2N\lambda_2^K(\mathbf{W}) \Re \left[ \left( \mathbf{v}_2^T \odot \hat{\mathbf{u}}^H \right) \mathbf{R} \hat{\mathbf{u}} \right] \text{diag}(\mathbf{v}_2) \hat{\mathbf{u}}. \end{aligned} \quad (34)$$

Hence, substituting this relation for  $\Delta_{\text{ODE}} \hat{\mathbf{u}}$  in (12) yields

$$\begin{aligned} \delta_{\text{ODE}} &= N\lambda_2^K(\mathbf{W}) \frac{\Re \left\{ \hat{\mathbf{u}}^H \left( (\mathbf{v}_2\mathbf{v}_2^T) \odot \mathbf{R} \right) \hat{\mathbf{u}} \right\}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}} \\ &\quad - N\lambda_2^K(\mathbf{W}) \frac{2\Re \left\{ \hat{\mathbf{u}}^H \Re \left\{ \left( \mathbf{v}_2^T \odot \hat{\mathbf{u}}^H \right) \mathbf{R} \hat{\mathbf{u}} \right\} \text{diag}(\mathbf{v}_2) \hat{\mathbf{u}} \right\}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}}. \end{aligned}$$

Since the spectral radius of a Hadamard product of two non-negative matrices is bounded by the product of the spectral radius of the matrices [44], then  $\rho((\mathbf{v}_2\mathbf{v}_2^T) \odot \mathbf{R}) \leq \rho(\mathbf{v}_2\mathbf{v}_2^T)\rho(\mathbf{R}) = \rho(\mathbf{R})$ . It can be easily derived that

$$\frac{\hat{\mathbf{u}}^H \left( (\mathbf{v}_2\mathbf{v}_2^T) \odot \mathbf{R} \right) \hat{\mathbf{u}}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}} \leq \max_{\hat{\mathbf{u}}} \frac{\hat{\mathbf{u}}^H \left( (\mathbf{v}_2\mathbf{v}_2^T) \odot \mathbf{R} \right) \hat{\mathbf{u}}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}} \leq 1. \quad (35)$$

Using generalized Rayleigh quotient yields

$$\begin{aligned} &\frac{2\Re \left\{ \hat{\mathbf{u}}^H \Re \left\{ \left( \mathbf{v}_2^T \odot \hat{\mathbf{u}}^H \right) \mathbf{R} \hat{\mathbf{u}} \right\} \text{diag}(\mathbf{v}_2) \hat{\mathbf{u}} \right\}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}} \\ &= \frac{2\Re \left\{ \left( \mathbf{v}_2^T \odot \hat{\mathbf{u}}^H \right) \mathbf{R} \hat{\mathbf{u}} \right\}}{\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}}} \Re \left\{ \hat{\mathbf{u}}^H \text{diag}(\mathbf{v}_2) \hat{\mathbf{u}} \right\} \\ &\leq \max_{\hat{\mathbf{u}}} \frac{2\Re \left\{ \hat{\mathbf{u}}^H \mathbf{R}^{-1/2} \text{diag}(\mathbf{v}_2) \mathbf{R}^{1/2} \hat{\mathbf{u}} \right\}}{\hat{\mathbf{u}}^H \hat{\mathbf{u}}} \|\mathbf{v}_2\|_{\infty} \|\hat{\mathbf{u}}\|_2^2 \\ &= 2\|\mathbf{v}_2\|_{\infty} \|\hat{\mathbf{u}}\|_2^2 \lambda_{\max}\{\text{diag}(\mathbf{v}_2)\} \\ &\leq 2\|\hat{\mathbf{u}}\|_2^2. \end{aligned} \quad (36)$$

Combining the results from (35) and (36) yields  $|\delta_{\text{ODE}}| \leq N\lambda_2^K(\mathbf{W}) (1 + 2\|\hat{\mathbf{u}}\|_2^2)$ . From (13), we have

$$1 - N\lambda_2^K(\mathbf{W}) (1 + 2\|\hat{\mathbf{u}}\|_2^2) \leq \|\hat{\mathbf{u}}\|_2^2 \leq 1 + N\lambda_2^K(\mathbf{W}) (1 + 2\|\hat{\mathbf{u}}\|_2^2).$$

For  $\{K \in \mathbb{N}_+ | \lambda_2^K(\mathbf{W}) \ll 1/(2N)\}$ ,  $\|\hat{\mathbf{u}}_{\text{eq}}\|_2^2$  is bounded in the interval

$$\begin{aligned} &\left( \frac{1 - N\lambda_2^K(\mathbf{W})}{1 + 2N\lambda_2^K(\mathbf{W})}, \frac{1 + N\lambda_2^K(\mathbf{W})}{1 - 2N\lambda_2^K(\mathbf{W})} \right) \\ &= \left( 1 - \frac{3}{\frac{1}{(N\lambda_2^K(\mathbf{W}))} + 2}, 1 + \frac{3}{\frac{1}{(N\lambda_2^K(\mathbf{W}))} - 2} \right) \\ &\approx (1 - 3N\lambda_2^K(\mathbf{W}), 1 + 3N\lambda_2^K(\mathbf{W})) \xrightarrow{k \rightarrow \infty} 1. \end{aligned}$$

This completes the proof.  $\blacksquare$

*Proof of Lemma 4.2:*

*Proof:* Since  $f(\hat{\mathbf{u}}) = [\mathbf{R} - (\hat{\mathbf{u}}^H \mathbf{R} \hat{\mathbf{u}})]\hat{\mathbf{u}}$ , the function  $f(\hat{\mathbf{u}})$  has  $N$  equilibrium points, which are the  $N$  eigenvectors of  $\mathbf{R}$ . Let  $\mathbf{u}_i$  be an equilibrium point, then the derivative

$$Df(\mathbf{u}_i) = \mathbf{R} - 2\mathbf{u}_i\mathbf{u}_i^H \mathbf{R} - (\mathbf{u}_i^H \mathbf{R} \mathbf{u}_i) \mathbf{I} = \mathbf{U} \Lambda \mathbf{U}^H$$

with  $\Lambda = [\lambda_1(\mathbf{R}) - \lambda_i(\mathbf{R}), \dots, -2\lambda_i(\mathbf{R}), \dots, \lambda_n(\mathbf{R}) - \lambda_i(\mathbf{R})]$ . By definition,  $\mathbf{u}_i$  is a sink if  $\lambda_j(\mathbf{R}) < \lambda_i(\mathbf{R})$  for  $j = 1, \dots, N$ ,  $j \neq i$  and  $\lambda_i(\mathbf{R}) > 0$ . It is easy to check that  $\mathbf{u}_i$  is a sink if and only if  $i = 1$ . Using Theorem 4.1, if we can show that  $g(\hat{\mathbf{u}})$  is a perturbation of  $f(\hat{\mathbf{u}})$ , then there exists a unique sink in  $g(\hat{\mathbf{u}})$  and it is in a neighborhood of  $\mathbf{u}_1$ . Let  $V = \text{diag} \mathbf{v}_2$ ,  $\epsilon = N\lambda_2^K(\mathbf{W})$  and  $\gamma = 3N\lambda_2^K(\mathbf{W})$ , then

$$\begin{aligned} &|f(\hat{\mathbf{u}}) - g(\hat{\mathbf{u}})| \\ &\leq \epsilon(1 + 2\gamma) \left\| V\mathbf{R}V - \hat{\mathbf{u}}^H V\mathbf{R}\hat{\mathbf{u}}V - \hat{\mathbf{u}}^H \mathbf{R}V\hat{\mathbf{u}}V \right\| \\ &\leq \epsilon\lambda_1(\mathbf{R})[3 + 8\gamma + o(\gamma)] \\ &\|Df(\hat{\mathbf{u}}) - Dg(\hat{\mathbf{u}})\| \\ &\leq \epsilon \left\| V\mathbf{R}V - 2V\hat{\mathbf{u}}\hat{\mathbf{u}}^H(\mathbf{R}V + V\mathbf{R}) \right. \\ &\quad \left. - \hat{\mathbf{u}}^H(\mathbf{R}V + V\mathbf{R})\hat{\mathbf{u}}V \right\| (1 + 2\gamma) \\ &\leq \epsilon\lambda_1(\mathbf{R})[7 + 20\gamma + o(\gamma)]. \end{aligned}$$

for all  $\hat{\mathbf{u}} \in \{\hat{\mathbf{u}} \in \mathbb{R}^N | \|\hat{\mathbf{u}}\|_2^2 = 1 \pm \gamma\}$ . One can make  $\epsilon$  sufficiently small; thus,  $g(\hat{\mathbf{u}})$  is a perturbation of  $f(\hat{\mathbf{u}})$ . Theorem 4.1 implies that there exists an equilibrium point  $\hat{\mathbf{u}}^*$  for the system  $g(\hat{\mathbf{u}}) = d\hat{\mathbf{u}}/dt$  and  $\|\mathbf{u}_1 - \hat{\mathbf{u}}^*\| < 3\epsilon\lambda_1(\mathbf{R}) + o(\epsilon)$ . In addition, Theorem 4.2 indicates that  $\hat{\mathbf{u}}^*$  is hyperbolic and it is a sink to the system  $g(\hat{\mathbf{u}}) = d\hat{\mathbf{u}}/dt$ .  $\blacksquare$

#### APPENDIX B

*Proof of the Claim in (23):*

*Proof:* In order to decentralize any expression of interest, we need to first decompose it into a sum of local variables and then apply (5). Let  $\hat{\mathbf{U}}_{t-1} = [\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_p]$ , then  $\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1}$  can be written as

$$\hat{\mathbf{U}}_{t-1}^H \hat{\mathbf{U}}_{t-1} = \sum_{j=1}^N \underbrace{\begin{bmatrix} \hat{u}_{1,j}^* \hat{u}_{1,j} & \cdots & \hat{u}_{1,j}^* \hat{u}_{p,j} \\ \vdots & \ddots & \vdots \\ \hat{u}_{p,j}^* \hat{u}_{1,j} & \cdots & \hat{u}_{p,j}^* \hat{u}_{p,j} \end{bmatrix}}_{\text{Information known at node } j}$$

where each row represents  $\hat{\mathbf{u}}_r^H \hat{\mathbf{U}}_{t-1}$ . From (20), we obtain a decentralized estimation of  $\hat{\mathbf{u}}_r^H \hat{\mathbf{U}}_{t-1}$  at node  $i$  and it is

$$[\mathbf{N}\mathbf{W}^K \text{diag}(\mathbf{u}_r^H) \mathbf{U}_{t-1}]_{i,1:N} = [\mathbf{N}\mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{u}_r^H) \mathbf{U}_{t-1}.$$

Provided that  $K$  average consensus iterations are performed, the  $i$ th node's approximation of the matrix  $\mathbf{U}_{t-1}^H \mathbf{U}_{t-1}$  is

$$\begin{bmatrix} [\mathbf{N}\mathbf{W}^k]_{i,1:N} \text{diag}(\mathbf{u}_1^*) \mathbf{U}_{t-1,k} \\ \vdots \\ [\mathbf{N}\mathbf{W}^k]_{i,1:N} \text{diag}(\mathbf{u}_p^*) \mathbf{U}_{t-1,k} \end{bmatrix}$$

by stacking the approximations  $\hat{\mathbf{u}}_r^H \hat{\mathbf{U}}_{t-1}$  at node  $i$  for all rows  $r = 1, \dots, N$ . ■

*Proof of the Claim in (24):*

*Proof:* Using Property 2.1 and the expression in (22), the decentralization of  $\mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}$  is

$$\begin{aligned} \text{diag}(\mathbf{r}_t) \mathbf{N}\mathbf{W}^K \text{diag}(\mathbf{r}_t^*) \mathbf{U}_{t-1} \\ = (\mathbf{N}\mathbf{W}^K \odot \mathbf{r}_t \mathbf{r}_t^H) \mathbf{U}_{t-1} \xrightarrow{k \rightarrow \infty} \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}. \end{aligned} \quad (37)$$

Furthermore, given the decentralization of  $\mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1,k}$  in (37) and expression in (23), the  $i$ th node's decentralized approximation of  $\mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \mathbf{U}_{t-1}$  is

$$\underbrace{[(\mathbf{N}\mathbf{W}^K \odot \mathbf{r}_t \mathbf{r}_t^H) \mathbf{U}_{t-1}]_{i,1:N}}_{= [\mathbf{N}\mathbf{W}^K \odot \mathbf{r}_t \mathbf{r}_t^H]_{i,1:N} \mathbf{U}_{t-1}} \begin{bmatrix} [\mathbf{N}\mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{u}_1^*) \mathbf{U}_{t-1} \\ \vdots \\ [\mathbf{N}\mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{u}_p^*) \mathbf{U}_{t-1} \end{bmatrix} \\ = \mathbf{U}_{t-1}^H \text{diag}([\mathbf{N}\mathbf{W}^K]_{i,1:N}) \mathbf{U}_{t-1}$$

Using Property 2.2, the preceding relation can be reduced to

$$(\mathbf{N}\mathbf{W}^K \odot [(\mathbf{N}\mathbf{W}^K \odot \mathbf{r}_t \mathbf{r}_t^H) \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H])_{i,1:N} \mathbf{U}_{t-1}.$$

Therefore, the decentralization of  $\mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \mathbf{U}_{t-1}$  is

$$\begin{aligned} (\mathbf{N}\mathbf{W}^K \odot [(\mathbf{N}\mathbf{W}^K \odot \mathbf{r}_t \mathbf{r}_t^H) \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H]) \mathbf{U}_{t-1} \\ \xrightarrow{k \rightarrow \infty} \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \mathbf{U}_{t-1}. \end{aligned} \quad (38)$$

Using the result in (22), which says that the decentralization of  $\mathbf{r}_t^H \mathbf{U}_{t-1}$  at node  $i$  is  $[\mathbf{N}\mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{r}_t^*) \mathbf{U}_{t-1}$ , the  $i$ th node's decentralized approximation of  $\mathbf{U}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}$  is

$$\begin{aligned} \mathbf{U}_{t-1}^H \underbrace{\text{diag}(\mathbf{r}_t) [\mathbf{N}\mathbf{W}^K]_{i,1:N} [\mathbf{N}\mathbf{W}^K]_{i,1:N} \text{diag}(\mathbf{r}_t^*)}_{= \text{diag}([\mathbf{N}\mathbf{W}^K]_{i,1:N}) \mathbf{r}_t \mathbf{r}_t^H \text{diag}([\mathbf{N}\mathbf{W}^K]_{i,1:N})} \mathbf{U}_{t-1} \\ \stackrel{:= \mathbf{T}}{=} \mathbf{U}_{t-1}^H \text{diag}([\mathbf{N}\mathbf{W}^K]_{i,1:N}) \mathbf{T} \mathbf{U}_{t-1}. \end{aligned}$$

Provided with the above approximation of  $\mathbf{U}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}$  at node  $i$ , the  $i$ th row of the decentralized  $\mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}$  is

$$[\mathbf{I}_N]_{i,1:N} \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \text{diag}([\mathbf{N}\mathbf{W}^K]_{i,1:N}) \mathbf{T} \mathbf{U}_{t-1}.$$

Using Property 2.2, the preceding expression becomes  $[\mathbf{N}\mathbf{W}^K \odot \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H]_{i,1:N} \mathbf{T} \mathbf{U}_{t-1}$ . Substituting the expression for  $\mathbf{T}$  and use Property 2.2, we get

$[\mathbf{N}\mathbf{W}^K \odot ((\mathbf{N}\mathbf{W}^K \odot (\mathbf{U}_{t-1} \mathbf{U}_{t-1}^H)) \mathbf{r}_t \mathbf{r}_t^H)]_{i,1:N} \mathbf{U}_{t-1}$ . Hence, the network-wide decentralization of  $\mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}$  is

$$(\mathbf{N}\mathbf{W}^K \odot [(\mathbf{N}\mathbf{W}^K \odot (\mathbf{U}_{t-1} \mathbf{U}_{t-1}^H)) \mathbf{r}_t \mathbf{r}_t^H]) \mathbf{U}_{t-1} \xrightarrow{k \rightarrow \infty} \mathbf{U}_{t-1} \mathbf{U}_{t-1}^H \mathbf{r}_t \mathbf{r}_t^H \mathbf{U}_{t-1}. \quad (39)$$

Using the results in (37), (38), and (39) in place of the corresponding centralized matrix-products in (16) completes the proof for the network-wide decentralization of the  $p$ -dimensional principal subspace updating algorithm. ■

## APPENDIX C

*Proof of Lemma 5.1:*

*Proof:* Apply the approximation  $\mathbf{X}^K \approx \lambda_2^K(\mathbf{W}) \mathbf{v}_2 \mathbf{v}_2^H$  to (25), we obtain

$$\tilde{\Delta}_{\text{ODE}} \approx N \lambda_2^K(\mathbf{W}) \tilde{\Delta} \quad (40)$$

where  $(\mathbf{v}_2 \mathbf{v}_2^T) \odot [\mathbf{R}(\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H) + (\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H) \mathbf{R}] - [(\mathbf{v}_2 \mathbf{v}_2^T) \odot \mathbf{R}] \hat{\mathbf{U}} \hat{\mathbf{U}}^H - [(\mathbf{v}_2 \mathbf{v}_2^T) \odot \mathbf{R}] \hat{\mathbf{U}} \hat{\mathbf{U}}^H - [(\mathbf{v}_2 \mathbf{v}_2^T) \odot \mathbf{R}] \hat{\mathbf{U}} \hat{\mathbf{U}}^H$ . Hence, over the time, we have

$$\begin{aligned} \|\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F \xrightarrow{t \rightarrow \infty} \|\tilde{\delta}_{\text{ODE}}\|_F, \text{ with} \\ \|\tilde{\delta}_{\text{ODE}}\|_F \leq \frac{N \lambda_2^K(\mathbf{W})}{2} \underbrace{\|\text{tr}(\hat{\mathbf{U}} \hat{\mathbf{U}}^H)^{-1} \text{tr}(\hat{\mathbf{U}} \hat{\mathbf{U}}^H)\|_F}_{\mu_u} \\ \times \|\mathbf{R}^{-1} \tilde{\Delta}\|_F \\ = N \lambda_2^K(\mathbf{W}) \frac{1}{2} \mu_u \|\mathbf{R}^{-1} \tilde{\Delta}\|_F. \end{aligned} \quad (41)$$

Using Property 2.3, 2.4, 2.5 for  $|v_2|_{\max} \leq 1$  and the sub-multiplicative property of Frobenius norm yield

$$\begin{aligned} \|\mathbf{R}^{-1} \tilde{\Delta}\|_F \\ = \|\mathbf{R}^{-1} ((\mathbf{v}_2 \mathbf{v}_2^T) \odot (\mathbf{R}(\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H)))\|_F \\ + \|\mathbf{R}^{-1} ((\mathbf{v}_2 \mathbf{v}_2^T) \odot ((\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H) \mathbf{R}))\|_F \\ + \|\mathbf{R}^{-1} ((\mathbf{v}_2 \mathbf{v}_2^T) \odot \mathbf{R})\|_F \|\hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F \\ + \|\mathbf{R}^{-1} ((\mathbf{v}_2 \mathbf{v}_2^T) \odot (\hat{\mathbf{U}} \hat{\mathbf{U}}^H)) \mathbf{R}\|_F \\ \leq \|\mathbf{R}^{-1} \text{diag}(\mathbf{v}_2) \mathbf{R}\|_F \|\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F \\ + \|\text{vec}(\Lambda^{-1}) \text{vec}^T(\Lambda) \odot (\mathbf{U}^H \text{diag}(\mathbf{v}_2) (\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H) \mathbf{U})\|_F \\ + \|\mathbf{R}^{-1} \text{diag}(\mathbf{v}_2) \mathbf{R}\|_F \|\hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F \\ + \|\text{vec}(\Lambda^{-1}) \text{vec}^T(\Lambda) \odot (\mathbf{U}^H \text{diag}(\mathbf{v}_2) \hat{\mathbf{U}} \hat{\mathbf{U}}^H \text{diag}(\mathbf{v}_2) \mathbf{U})\|_F \\ \leq \sqrt{\text{tr}(|\mathbf{v}_2|)} \|\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F + \|\mathbf{U} \text{diag}(\mathbf{v}_2) (\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H) \mathbf{U}^H\|_F \\ + \sqrt{\text{tr}(|\mathbf{v}_2|)} \|\hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F + \|\mathbf{U} \text{diag}(\mathbf{v}_2) \hat{\mathbf{U}} \hat{\mathbf{U}}^H \text{diag}(\mathbf{v}_2) \mathbf{U}^H\|_F \\ \leq C \left( \|\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F + \|\hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F \right) \\ \leq C \left( \|\mathbf{I} - \hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F + \|\hat{\mathbf{U}} \hat{\mathbf{U}}^H\|_F + \sqrt{N-p} \right) \end{aligned}$$

$$\leq C \left( p + 2 \|\hat{\mathbf{U}}^H \hat{\mathbf{U}}\|_F + \sqrt{N-p} \right)$$

where  $C = 1 + \sqrt{\text{tr}(\mathbf{V}_2)}$ . Combining this bound with (41) and using the triangular inequality  $|p - \|\hat{\mathbf{U}}^H \hat{\mathbf{U}}\|_F| \leq \left\| \mathbf{I} - \hat{\mathbf{U}}^H \hat{\mathbf{U}} \right\|_F$  yield that  $\left\| \hat{\mathbf{U}}^H \hat{\mathbf{U}} \right\|_F$  lies within the interval of  $p \pm (3/2)\mu_u N \lambda_2^K(\mathbf{W}) C(p + \sqrt{N-p}) = p \pm \mathcal{O}(N^{3/2} \lambda_2^K(\mathbf{W}))$ . Hence, the norm  $\|\hat{\mathbf{U}}^H \hat{\mathbf{U}}\|_F$  is bounded around  $p$ , and

$$\begin{aligned} \left\| \mathbf{I} - \hat{\mathbf{U}}^H \hat{\mathbf{U}} \right\|_F &\leq \frac{1}{2} \mu_u N \lambda_2^K(\mathbf{W}) C \left( p + \sqrt{N-p} + 2 \|\hat{\mathbf{U}}^H \hat{\mathbf{U}}\|_F \right) \\ &= \mathcal{O} \left( N^{3/2} \lambda_2^K(\mathbf{W}) \right). \end{aligned}$$

This completes the proof.  $\blacksquare$

## REFERENCES

- [1] G. Stewart, *Matrix Algorithms*. Philadelphia, PA: SIAM, 1998.
- [2] E. Oja, "Simplified neuron model as a principal component analyzer," *J. Math. Biol.*, vol. 15, pp. 267–273, 1982.
- [3] J. Karhunen, "Adaptive algorithms for estimating eigenvectors of correlation type matrices," in *Proc. Int. Conf. Acoust., Speech, Signal Process. ICASSP '84*, Mar. 1984, vol. 9, pp. 592–595.
- [4] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
- [5] N. Owsley, "Adaptive data orthogonalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. ICASSP '78*, Apr. 1978, vol. 3, pp. 109–112.
- [6] K. Sharman, "Adaptive algorithms for estimating the complete covariance eigenstructure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. ICASSP '86*, Apr. 1986, vol. 11, pp. 1401–1404.
- [7] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numerische Math.*, vol. 31, pp. 31–48, 1978.
- [8] J. H. Manton, I. Mareels, and S. Attallah, "An analysis of the fast subspace tracking algorithm NOja," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. ICASSP '93*, Apr. 2002, vol. 2.
- [9] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Process. Lett.*, vol. 8, no. 7, pp. 203–206, Jul. 2001.
- [10] J. H. Manton, U. Helmke, and I. M. Mareels, "A dual purpose principal and minor component flow," *Syst. Control Lett.*, vol. 54, no. 8, pp. 759–769, 2005.
- [11] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.
- [12] A. Dimakis, S. Kar, J. Moura, M. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.
- [13] K. Tsiano and M. Rabbat, "Fast decentralized averaging via multi-scale Gossip," *IEEE Distrib. Comput. Sens. Syst.*, vol. 6131, pp. 320–333, Jun. 2010.
- [14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proc. 42nd IEEE Conf. Decision Control*, Dec. 2003, vol. 5, pp. 4997–5002.
- [15] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Mass. Inst. of Tech., Cambridge, 1984.
- [16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized Gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
- [17] F. Benezit, A. Dimakis, P. Thiran, and M. Vetterli, "Gossip along the way: Order-optimal consensus through randomized path averaging," in *Proc. Allerton Conf. Commun., Control. Comput.*, Urbana-Champaign, IL, Sep. 2007.
- [18] A. Dimakis, A. Sarwate, and M. Wainwright, "Geographic Gossip: Efficient averaging for sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1205–1216, Mar. 2008.
- [19] M. Rabbat, "On spatial Gossip algorithms for distributed averaging," in *Proc. IEEE Statist. Signal Process. Workshop*, Madison, WI, Aug. 2007, pp. 705–709.
- [20] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. 42nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2008, pp. 1722–1726.
- [21] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed consensus algorithms based on advection-diffusion processes," in *Proc. 5th IEEE Sens. Array Multichannel Signal Process. Workshop SAM'08*, Jul. 2008, pp. 266–270.
- [22] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. IPSN'04*, Apr. 2004, pp. 20–27.
- [23] I. Schizas, G. Giannakis, S. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links-part II: Distributed estimation and smoothing of random signals," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.
- [24] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.
- [25] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decision Control and 2005 Eur. Control Conf. CDC-ECC '05*, Dec. 2005, pp. 8179–8184.
- [26] A. Sayed and C. Lopes, "Distributed processing over adaptive networks," in *Proc. 9th Int. Symp. Signal Process. and Its Applicat. ISSPA '07*, Feb. 2007, pp. 1–3.
- [27] F. Cattivelli and A. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
- [28] S. Barbarossa and G. Scutari, "Decentralized maximum-likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3456–3470, Jul. 2007.
- [29] V. Veeravalli, T. Basar, and H. Poor, "Minimax robust decentralized detection," *IEEE Trans. Inf. Theory*, vol. 40, no. 1, pp. 35–40, Jan. 1994.
- [30] L. Li, X. Li, A. Scaglione, and J. Manton, "Decentralized subspace tracking via Gossiping," in *Distributed Computing in Sensor Systems*, ser. Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer, 2010, vol. 6131, pp. 130–143.
- [31] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. 36th Annu. ACM STOC*, Chicago, IL, Jun. 2004, pp. 561–568.
- [32] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip and mixing times of random walks on random graphs," 2004 [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.3609>
- [33] B. Oreshkin, M. Coates, and M. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2850–2865, May 2010.
- [34] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. Anal. Applicat.*, vol. 106, no. 1, pp. 69–84, 1985.
- [35] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sens. Netw. IPSN'05*, Apr. 2005, pp. 63–70.
- [36] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. Berlin, Germany: Springer-Verlag, 1990.
- [37] D. Varagnolo, G. Pilonetto, and L. Schenato, "Distributed statistical estimation of the number of nodes in sensor networks," in *Proc. Decision and Control Conf.*, 2010.
- [38] C. Johnson, *Matrix Theory and Applications*. Providence, RI: American Math. Society, 1990.
- [39] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
- [40] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. New York: Wiley, 1996.
- [41] E. Oja, "Neural networks, principal components, and subspaces," *Int. J. Neural Syst.*, vol. 1, pp. 61–68, 1989.
- [42] M. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*. New York: Academic, 1974.
- [43] M. Penrose, *Random Geometric Graph*. Oxford, U.K.: Oxford Studies in Probability, 2003.
- [44] R. A. Horn and F. Zhang, "Bounds on the spectral radius of a Hadamard product of nonnegative or positive semidefinite matrices," *Electron. J. Linear Algebra*, vol. 20, pp. 90–94, Feb. 2010.



**Lin Li** (S'10) received the B.Sc. degree electrical and computer engineering from the University of California, Davis, in 2008. She is currently pursuing the M.Sc. degree at the University of California, Davis.

Her research interest is in the broad area of wireless communication and signal processing. She is currently working on decentralized network control and adaptive parameter estimation in wireless sensor networks using distributed consensus and gossiping algorithms. She is also interested in modeling how opinions converge in social networks using a gossip

type interaction and applying differential geometry to many problems in signal processing.



**Anna Scaglione** (M'99–SM'09–F'11) received the Laurea (M.Sc. degree) and the Ph.D. degree from the University of Rome, "La Sapienza," Rome, Italy, in 1995 and 1999, respectively.

She is currently a Professor in electrical and computer engineering at the University of California, Davis, where she joined in 2008. She was previously at Cornell University, Ithaca, NY, from 2001 where she became an Associate Professor in 2006; prior to joining Cornell she was an Assistant Professor from 2000 to 2001 at the University of New Mexico. Her

current research focuses on communication and wireless networks, sensors' systems for monitoring, control and energy management and network science.

Dr. Scaglione served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2002 to 2005, and has served the Editorial Board of the IEEE Transactions on Signal Processing since 2008, where she is acting as Area Editor. She has co-organized several special sessions in signal processing and communication conferences; she has served in the past as Guest Editor for the *Communication Magazine* and the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING and she is currently serving as guest editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, for a series of special issues on Smartgrid. She has been on the Signal Processing for Communication Committee from 2004 to 2009. She was general chair of the workshop SPAWC 2005 and keynote speaker in SPAWC 2008 and ISPLC 2010.

She received the 2000 IEEE TRANSACTIONS ON SIGNAL PROCESSING Best Paper Award, the NSF Career Award in 2002, and she is co-recipient of the Ellersick Best Paper Award (MILCOM 2005). Her expertise is in the broad area of signal processing for communication systems and networks.



**Jonathan H. Manton** (M'02–SM'03) received the B.Sc. degree in mathematics and the B.Eng. degree in electrical engineering in 1995 and the Ph.D. degree in 1998, all from the University of Melbourne, Melbourne, Vic., Australia.

He holds a Distinguished Chair at the University of Melbourne with the title Future Generation Professor. He is also an Adjunct Professor in the Mathematical Sciences Institute at the Australian National University. From 1998 to 2004, he was with the Department of Electrical and Electronic Engineering at

the University of Melbourne. During that time, he held a Postdoctoral Research Fellowship then subsequently a Queen Elizabeth II Fellowship, both from the Australian Research Council. In 2005 he became a Full Professor in the Department of Information Engineering, Research School of Information Sciences and Engineering (RSISE), Australian National University. From July 2006 to May 2008, he was on secondment to the Australian Research Council as Executive Director, Mathematics, Information, and Communication Sciences. His traditional research interests range from pure mathematics (e.g., commutative algebra, algebraic geometry, differential geometry) to engineering (e.g., signal processing, wireless communications). Recently though, led by a desire to participate in the convergence of the life sciences and the mathematical sciences, he has started to apply his expertise to research in neuroscience. He also has extensive experience in software development.

Prof. Manton has served recently as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Committee Member for the IEEE Signal Processing for Communications (SPCOM) Technical Committee, and a Committee Member on the Mathematics Panel for the ACT Board of Senior Secondary Studies in Australia. He is currently on the Editorial Board of *Systems and Control Letters*. He has recently received a 2009 Future Summit Australian Leadership Award in recognition of "outstanding achievements and commitment to playing a leading role in shaping the future of Australia."