MAXVAR-based Distributed Correlation Estimation in a Wireless Sensor Network

MAXVAR-based Distributed Correlation Estimation in a Wireless Sensor Network

Charles Hovine ⬡, Alexander Bertrand ⬡

KU Leuven, Department of Electrical Engineering (ESAT)
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
KU Leuven Institute for Artificial Intelligence (Leuven.AI)
Leuven, Belgium
{charles.hovine, alexander.bertrand}@esat.kuleuven.be

Abstract—The performance of most array signal processing tasks relies on the presence of correlation between sensor signals. In a wireless sensor network, where sensor nodes are spread out over a relatively large area, it is useful to identify nodes observing similar sensor signals and hence common phenomenons, for example to partition the network according to the observed latent signals and corresponding correlation structure. This can be achieved via the so-called MAXVAR formulation of generalized canonical correlation analysis, which finds a low-dimensional subspace that highlights correlated signal components between multiple nodes' observed signal subspaces. The classical procedure for computing the solutions of MAXVAR consists in performing a generalized eigenvalue decomposition after collecting all the sensors' signals at a fusion center. However, this typically incurs high communication and computational costs. In this paper, we describe a low communication and computational cost distributed algorithm that computes the solutions of MAXVAR without aggregating the nodes' observations at a central location. We show how a subset of those solutions can be used locally by each node to estimate the global correlation structure across all nodes in the network, thereby allowing any node to evaluate the presence of correlated signals at any other node, even if no direct link is shared. We prove the convergence of the algorithm and validate our method for estimating the correlation structure via simulations.

Index Terms—Wireless sensor networks, distributed array processing, generalized canonical correlation analysis, MAXVAR, network pruning.

I. INTRODUCTION

A Wireless Sensor Network (WSN) consists of a collection of nodes that are equipped with one or more sensors, wireless communication capabilities, and a processing unit. The sensor observations collected in a WSN can either be forwarded to a "fusion center" (FC) where all the data is collected and centrally processed, or the data processing task can be collaboratively performed in a decentralized fashion by the sensing nodes themselves [2]. Centralized computation has the advantage of allowing the use of off-the-shelf algorithms and techniques, but comes at the cost of greater bandwidth requirements, since all nodes have to send their raw data to

the FC, as well as a large computational cost at the FC, which scales poorly with respect to the number of sensor nodes. The aforementioned increase in bandwidth is even more severe in the case of networks relying on multi-hop routing towards the FC, where some nodes act as data relays for other nodes. Furthermore, the FC constitutes a single point of failure which compromises the robustness of the network. In comparison, distributed processing only requires each node to solve a local low-complexity task to collaboratively complete a global task of higher complexity.

Array signal processing tasks such as signal estimation, filtering, or subspace estimation, generally boil down to the extraction of specific signal components, often split across many sensor signals. Due to the presence of sensing noise and the spatial distribution of the underlying signal sources, different nodes typically observe different but correlated signals, some sharing common components and others not. It is therefore of great importance to identify which node pairs share a common latent signal subspace and which do not, as this knowledge can be used to prune the network or cluster the nodes according to the similarity between their signal subspaces, resulting in further bandwidth and computational complexity reduction at each node [3], [4].

In this paper, we leverage the fact that the so-called principal angles derived from the solutions of the canonical correlation analysis (CCA) problem can be used to quantify the similarity between two nodes' sensor signal subspaces [5], [6]. Indeed, CCA can be used to estimate the signal components that are maximally correlated between two different nodes. It is closely related to principal component analysis (PCA) [7] and the Karhunen–Loève transform (KLT) [8], which extracts the highest power components, yet not necessarily observed by both nodes. As CCA is only applicable to identify the most correlated signal components within the signal subspaces of two nodes, applying it to a network of more than two nodes would result in a combinatorial complexity scaling. In this paper, we show that one of its multi-set generalizations, the so-called "Maximum Variance" (MAXVAR) generalization [9]–[11], can be used to approximate the solutions of the pairwise CCA problems and therefore the complete correlation structure of the network with a complexity that scales linearly with the network's size. MAXVAR has indeed been shown to provide a description of the intersection between multiple subspaces

[12], which fits with the general purpose of finding "shared" signals subspaces across multiple nodes of a WSN.

There exist multiple multi-set generalizations of CCA, each characterized by a specific objective function and set of constraints. The MAXVAR formulation historically refers to the objective function introduced by Horst [9]. Carroll [10] later introduced a new set of constraints for this objective which turns the formulation into an easily interpretable subspace decomposition and on which we focus in this paper.

As using MAXVAR in a centralized fashion would hinder our progress towards bandwidth and complexity reduction, we present a distributed algorithm for obtaining the solutions of MAXVAR, relying on the exchange of compressed signals between the nodes. We describe variants of this distributed MAXVAR (D-MAXVAR) algorithm for fully connected, star-topology and tree-topology networks, and describe how it can be extended to arbitrary topologies. We then explain how its solution can be used to evaluate the pair-wise correlation between any pair of nodes in the network, even if they are not connected via a direct link.

Previous works have investigated similar distributed subspace decompositions, targeting the union of the nodes' subspaces by extracting the components of greatest power [13], [14], maximal SNR [15], [16] or maximizing correlation between two sets of signals [17], [18]. In this work, we focus on finding the components that most adequately describe the inter-node correlation coefficient, also known as Pearson's correlation coefficient. Distributed algorithms already exist for the so-called SUMCORR generalization of CCA [19] but, to our knowledge, no distributed algorithm has been developed for MAXVAR, which, contrarily to SUMCORR, admits an analytical solution. In addition, the focus of [19] was on computational efficiency, rather than on bandwidth scaling, as is the case of the present work. In our previous work [1], we presented a condensed description of the algorithm for star- and tree-topology networks. In this paper, we provide a proof of convergence and explain how the algorithm's outputs can be used to estimate the network's correlation structure. We also provide numerical simulations to demonstrate the algorithm's effectiveness and its convergence properties.

The paper is organized as follows. In Section II, we cover the preliminaries and algebraic concepts related to MAXVAR and explain how it can be used in the context of WSNs. In Section III, we describe a distributed MAXVAR algorithm for fully connected networks and discuss its convergence properties. The algorithm is extended to arbitrary topologies in Section IV. Section V discusses how the solutions of MAXVAR can be used by each node to estimate the inter-node correlation structure, and in particular evaluate the degree to which each node's signals correlate with any other node, even several hops away. In Section VI, we assess the algorithm's performance through various simulations. We conclude by a brief discussion in Section VII.

II. PRELIMINARIES

A. WSN Setting and Notation

We consider a WSN consisting of K nodes in which each node $k \in \mathcal{K} = \{1, \ldots, K\}$ collects discrete observations of a real-valued M_k-channel sensor signal $\boldsymbol{x}_k = [x_{k,1}, \ldots, x_{k,M_k}]^T$. We model $\boldsymbol{x}_k \in \mathbb{R}^{M_k}$ as a stochastic process and denote $\boldsymbol{x}_k[t]$ its value at time t. We assume that $\boldsymbol{x}_k$ is zero-mean, ergodic and short-time stationary, allowing us to estimate the slowly varying covariance matrices from sample averages over finite segments of data:

$$\mathbb{E}\left\{\boldsymbol{x}_k[t]\boldsymbol{x}_l^T[t]\right\} = \boldsymbol{R}_{\boldsymbol{x}_k\boldsymbol{x}_l} \approx \frac{1}{T}\boldsymbol{X}_k[t]\boldsymbol{X}_l^T[t] \qquad (1)$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation operator and $\boldsymbol{X}_k[t]$ denotes the $M_k \times T$ observation matrix containing T consecutive observations of $\boldsymbol{x}_k$ centered around t in its columns. Finally, we define the network-wide observation vector as the M-channel vector $\boldsymbol{x}$ obtained by stacking the $\boldsymbol{x}_k$'s and where $M = \sum_k M_k$.

B. Canonical Correlation Analysis (CCA)

We wish to characterize the intensity of correlations between the signals of any two nodes. The so-called *canonical correlation coefficients* provide such a characterization, which can be found by means of canonical correlation analysis [20]. Considering two multi-channel signals $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$ associated with two nodes $k, l \in \mathcal{K}$, CCA computes spatial filters $\boldsymbol{w}_k$ and $\boldsymbol{w}_l$ that maximize the correlation coefficient ρ^{kl} between their output signals $z_k = \boldsymbol{w}_k^T\boldsymbol{x}_k$ and $z_l = \boldsymbol{w}_l^T\boldsymbol{x}_l$. The output signals z_l and z_k are referred to as the first canonical directions, and ρ^{kl} is referred to as the first canonical correlation coefficient. Additional canonical directions and coefficients can be found by computing additional pairs of spatial filters that maximize the correlation between their outputs while having their outputs remain uncorrelated (orthogonal) to the previous canonical directions. Formally, the i-th canonical correlation coefficient ρ_i^{kl} and *canonical directions* $z_{k,i}, z_{l,i}$ are defined as

$$\rho_i^{kl} = \mathbb{E}\{z_{k,i}z_{l,i}\} = \max_{\boldsymbol{w}_k, \boldsymbol{w}_l} \mathbb{E}\left\{\boldsymbol{w}_k^T\boldsymbol{x}_k\boldsymbol{x}_l^T\boldsymbol{w}_l\right\} \qquad (2a)$$

$$\text{s.t.} \quad z_k = \boldsymbol{w}_k^T\boldsymbol{x}_k, \; z_l = \boldsymbol{w}_l^T\boldsymbol{x}_l \qquad (2b)$$

$$\mathbb{E}\left\{z_k^2\right\} = \mathbb{E}\left\{z_l^2\right\} = 1 \qquad (2c)$$

$$\mathbb{E}\{z_k[z_{k,1}, \ldots, z_{k,i-1}]\} = 0 \qquad (2d)$$

$$\mathbb{E}\{z_l[z_{l,1}, \ldots, z_{l,i-1}]\} = 0 \qquad (2e)$$

The three last conditions require the canonical directions to have unit-variance and be orthogonal with respect to each other. Note that $\rho_i^{kl} = 1$ implies that the pairs $(z_{k,i}, z_{l,i})$ span the exact intersection between the sensor signal subspaces of nodes k and l.

If we now express the set of canonical directions of node k with respect to node l as

$$\boldsymbol{z}_{kl} = [z_{kl,1}, \ldots, z_{kl,M_{kl}}]^T = \boldsymbol{W}_{kl}^T\boldsymbol{x}_k \qquad (3)$$

with $M_{kl} = \min(M_k, M_l)$ and $\boldsymbol{W}_{kl} \in \mathbb{R}^{M_k \times M_{kl}}$, it can be shown [21] that the canonical directions and coefficients between two nodes k and l are the solutions of the following

generalized eigenvalue problem:

$$
\begin{bmatrix} \mathbf{0} & \mathbf{R}_{\mathbf{x}_k \mathbf{x}_l} \\ \mathbf{R}_{\mathbf{x}_l \mathbf{x}_k} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{W}_{kl} \\ \mathbf{W}_{lk} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\mathbf{x}_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathbf{x}_l} \end{bmatrix} \begin{bmatrix} \mathbf{W}_{kl} \\ \mathbf{W}_{lk} \end{bmatrix} \mathbf{\Lambda}_{kl}
$$
(4a)

$$
\begin{bmatrix} \mathbf{W}_{kl}^T & \mathbf{W}_{lk}^T \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathbf{x}_k} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{\mathbf{x}_l} \end{bmatrix} \begin{bmatrix} \mathbf{W}_{kl} \\ \mathbf{W}_{lk} \end{bmatrix} = \mathbf{I} \quad \text{(4b)}
$$

where $\mathbf{R}_{\mathbf{x}_k}$ is a shorthand notation for $\mathbf{R}_{\mathbf{x}_k \mathbf{x}_k}$ and $\mathbf{\Lambda}_{kl}$ is a diagonal matrix whose diagonal entries are non-negative and correspond to the canonical correlation coefficients. As a result, the optimal solution of (2) is given by the generalized eigenvectors corresponding to the largest generalized eigenvalues. By expressing the canonical directions via the parametrization introduced by (3), CCA can indeed be seen as the problem of finding two sets of spatial filters whose corresponding outputs have maximal correlation.

Remark II.1. Although we have thus far assumed real signals, the developments in this paper can be straighforwardly extended to the complex case by replacing transposes by conjugate transposes where appropriate. This would turn problem (2) into a maximization of the *real part* of $\mathbb{E}\{z_{k,i}z_{l,i}\}$ (which would enforce the phases of the two signals $z_{k,i}$ and $z_{l,i}$ to be perfectly aligned). Note that, as the pencil in (4) is symmetric, ρ_i^{kl}, i.e. the i-th largest diagonal element of $\mathbf{\Lambda}_{kl}$, is guaranteed to have no imaginary part in the complex or real case [22].

C. Total Squared Correlation

From the definition of canonical correlation coefficients, we can define the *total squared correlation* (TSC)

$$
\Theta_{kl} \triangleq \sum_{i=1}^{M_{kl}} (\rho_i^{kl})^2 = \text{Tr}\left(\mathbf{\Lambda}_{kl}^2\right) \quad (5)
$$

which can be interpreted as a generalization of correlation between random variables to correlation between sets of random variables. Indeed, if we denote $\mathcal{X}_k = \text{Span}\,\mathbf{x}_k$, i.e. the signal subspace spanned by the channels of $\mathbf{x}_k$, we have:

$$
\Theta_{kl} = 0 \Leftrightarrow \mathcal{X}_k \perp \mathcal{X}_l \quad (6)
$$
$$
\Theta_{kl} = M_{kl} \Leftrightarrow \mathcal{X}_k \subseteq \mathcal{X}_l \quad (7)
$$

assuming $\dim(\mathcal{X}_k) \leq \dim(\mathcal{X}_l)$. Note that in the one-dimensional case ($M_k = M_l = 1$), the TSC reduces to the usual squared correlation coefficient. Contrarily to correlation, the TSC is not concerned with the direction of the relationship (i.e. positively or negatively correlated), only the absolute magnitude of the relationship is captured. The TSC is positive and symmetric but does not satisfy the triangle inequality and is therefore not a distance [23]. Still, in the case where the channels of $\mathbf{x}_k$ and $\mathbf{x}_l$ are linearly independent (which is in practice always true in noisy settings), it can easily be turned into one by defining

$$
d_{kl} = \sqrt{\max\{M_k, M_l\} - \Theta_{kl}} \quad (8)
$$

which satisfies the triangle equality [24] and corresponds to the so-called Chordal distance in the particular case where $M_k = M_l$ [25]. Considering these above properties, the TSC is an adequate metric for characterizing the degree to which the signals of two nodes correlate, which is the main goal of this paper. In a WSN context, the TSC could for example be used as the key ingredient to an adaptive network where links between low-TSC node pairs are pruned, while more bandwidth is allocated to links between high-TSC node pairs. Similarly, the TSC could be used to define a weighted graph describing the correlation structure of the network, and on which spectral clustering techniques could be applied [26].

D. MAXVAR: Extension to more than two Subspaces

Problem (2) can be generalized to more than two nodes or subspaces in various fashions. As it admits a closed form solution[1], we consider here the so-called *Maximum-Variance* (MAXVAR) generalization of the CCA problem, as described by Carroll [10] and defined as follows for K signal subspaces:

$$
\min_{\{\mathbf{W}_k\}} \min_{\mathbf{s}} \sum_{k=1}^{K} \mathbb{E}\left\{\|\mathbf{s} - \mathbf{z}_k\|^2\right\} \quad (9a)
$$
$$
\text{s.t.} \quad \mathbf{z}_k = \mathbf{W}_k^T \mathbf{x}_k \quad (9b)
$$
$$
\mathbb{E}\left\{\mathbf{s}\mathbf{s}^T\right\} = \mathbf{I}_Q \quad (9c)
$$

where Q is the number of desired components. From the parametrization introduced by constraint (9b), the problem becomes equivalent to finding a Q-outputs filter $\mathbf{W}_k \in \mathbb{C}^{M_k \times Q}$ per node such that the filtered observations $\mathbf{z}_k$, which we refer to as the per-node *MAXVAR directions*, are as close as possible to some common network-wide Q-dimensional signal $\mathbf{s}$, and hence as close as possible to each other. From (9), it is clear that the per-node MAXVAR directions are the orthogonal projections of $\mathbf{s}$ onto the nodes' signal subspaces $\text{Span}(\mathbf{x}_k)$. Hence, $\text{Span}(\mathbf{s})$ is the Q-dimensional subspace whose signals have minimal average projection error onto the nodes' individual signal subspaces.

In [27], it is shown that the solution of (9) satisfies

$$
\mathbf{s} = \frac{1}{K}\sum_k \mathbf{z}_k = \mathbf{W}^T \mathbf{x} \quad (10)
$$

where $\mathbf{W} \triangleq [\mathbf{W}_1^T \cdots \mathbf{W}_K^T]^T$ is the matrix obtained by stacking the $\mathbf{W}_k$'s. Substituting (10) in (9a), the objective can be reformulated as

$$
\min_{\{\mathbf{W}_k\}} \sum_{k,l=1}^{K} \mathbb{E}\left\{\|\mathbf{W}_l^T \mathbf{x}_l - \mathbf{W}_k^T \mathbf{x}_k\|^2\right\}. \quad (11)
$$

The problem therefore consists in finding finding the set of node-specific filters whose outputs are as close to each other as possible in a minimum squared error sense. Furthermore, it is also shown in [27] that the $\mathbf{W}_k$'s are solutions to the following eigenvalue problem:

$$
\mathbf{R}_D \mathbf{W} = \mathbf{R}_{\mathbf{x}\mathbf{x}} \mathbf{W} \mathbf{\Lambda} \quad (12a)
$$
$$
\mathbf{W}^T \mathbf{R}_{\mathbf{x}\mathbf{x}} \mathbf{W} = \mathbf{I}_Q \quad (12b)
$$

where $\mathbf{R}_D \triangleq \text{Blkdiag}(\mathbf{R}_{\mathbf{x}_1}, \ldots, \mathbf{R}_{\mathbf{x}_K})$ is the block diagonal matrix containing the node-specific covariance matrices and $\mathbf{\Lambda}$ is a diagonal matrix. Solving problem (9) is therefore equivalent to computing the generalized eigenvalue decomposition (GEVD) of the matrix pencil $(\mathbf{R}_D, \mathbf{R}_{\mathbf{x}\mathbf{x}})$, keeping only the generalized eigenvectors (GEVC) corresponding to the Q smallest generalized eigenvalues (GEVL).

---

[1]Which is not the case for SUMCORR, discussed in Section I.

E. Problem Statement

For each pair of nodes, we wish to assess how "close" their observed signal subspaces are to each other. This can be achieved by computing the TSC as described by (5). However, this approach has two major drawbacks. Firstly, it requires the computation of $\boldsymbol{R}_{\boldsymbol{x}_k \boldsymbol{x}_l}$ which is only possible if the full observations $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$ are co-located at a single node (or an FC), therefore incurring high communication cost. Secondly, a CCA solution needs to be computed for each of the $K(K-1)/2$ pairs of nodes, further increasing the communication and computational cost of the procedure. As an efficient alternative, we propose to jointly approximate the TSCs between all node pairs using a distributed MAXVAR procedure. Indeed, we will show in Section V that solving the MAXVAR problem produces a joint approximation of the TSC of each node pair. In order to find the solution of (12), all nodes would typically need to share their observations to an FC where the full covariance matrix could be estimated. This requires a large communication bandwidth between the nodes and the FC, in particular if all nodes are not directly connected to the FC, which increases the stress on the communication links of the nodes that are close to the fusion center. In addition, such a centralized processing does not leverage the fact that the signal subspace in which inter-node correlation is present typically manifests a low dimension, and that its signals can therefore be efficiently described by only a few components.

In Section III, we present a distributed algorithm for solving (12) in a distributed fashion and relying on the transmission of low-dimensional compressed views of the data between neighboring nodes, thus lifting the need to transfer the raw M_k-channel sensor observations of all nodes to an FC through a possibly multi-hop network. Instead, the signal observations are directly fused with other observations within the network, and such that each node eventually has access to an estimate of (5) (which is established in Section V), thereby avoiding the need for a fusion center altogether.

III. Distributed MAXVAR in Fully-Connected Networks

In this section, we derive a distributed iterative algorithm for computing the Q first per-node MAXVAR directions associated with each node, and which is such that each node k eventually has access to the network-wide estimate of $\boldsymbol{s}$ and its own per-node MAXVAR direction $\boldsymbol{z}_k$ as defined in (9b). In Section V, we explain how the TSC between any node pair can be approximated based on only those quantities (even for those nodes that do not share a direct communication link). Relying on our interest in the first components only (which is motivated in Section V) and the particular problem structure, we show that neighboring nodes only need to share Q-dimensional compressed views of their observed signal subspaces at each iteration, which eventually converge to their first Q per-node MAXVAR directions $\boldsymbol{z}_k$. In order to facilitate the reader's understanding and intuition in the algorithm development, we first derive the algorithm for the simpler case of fully-connected networks and later extend it to more general topologies.

A. Algorithm Derivation

In a fully-connected network, any node can communicate with any other node via a single hop. By denoting the set of neighbors of node k as $\mathcal{N}_k$, we have for such networks $\mathcal{N}_k = \mathcal{K} \smallsetminus \{k\}$.

Considering the GEVD formulation (12), the problem of finding the Q first per-node MAXVAR directions is equivalent to finding the Q GEVCs of the matrix pencil $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{xx}})$ associated with its smallest GEVLs and corresponding to the columns of $\boldsymbol{W}$. The algorithm iteratively updates the $M_k \times Q$ matrices $\boldsymbol{W}_k^i$ (where the superscript i denotes the iteration index), which act as the local estimates of $\boldsymbol{W}_k$ and, as will be shown, as a node-specific compressor of the nodes' signals. It can be shown [28] that the GEVCs (corresponding to columns of $\boldsymbol{W}$) associated with the Q smallest GEVLs from the GEVD in (12) and hence the MAXVAR solutions of (9) coincide with the solution of the following trace minimization problem

$$\min_{\boldsymbol{W} \in \mathbb{R}^{M \times Q}} \quad \mathrm{Tr}\left(\boldsymbol{W}^T \boldsymbol{R}_D \boldsymbol{W}\right) \tag{13a}$$

$$\text{s.t.} \quad \boldsymbol{W}^T \boldsymbol{R}_{\boldsymbol{xx}} \boldsymbol{W} = \boldsymbol{I}_Q \tag{13b}$$

where $\mathrm{Tr}\left(\cdot\right)$ denotes the trace operator. The core idea of the algorithm is to have the nodes solve a local version of (13) in turns, and expressed in terms of the node's own raw observations and the other nodes' compressed observations $\boldsymbol{z}_k^i = \boldsymbol{W}_k^{iH} \boldsymbol{x}_k$ ($\boldsymbol{W}_k^{iH}$ denotes the compression matrix associated with the signals of node k at iteration i of the algorithm). The notation $\boldsymbol{z}_k^i, \boldsymbol{W}_k^{iH}$ for the compressed observations and compression matrices is chosen deliberately, as they also correspond to node k's local estimate of its part of the solution of (9). We indeed expect that eventually

$$\lim_{i \to \infty} \boldsymbol{W}_k^i = \boldsymbol{W}_k \text{ and } \lim_{i \to \infty} \boldsymbol{z}_k^i = \boldsymbol{z}_k. \tag{14}$$

The algorithm in fully connected networks is as follows. At the beginning of each iteration, an updating node q is selected. Every other node transmits a batch of its compressed observations $\boldsymbol{z}_k^i \ \forall k \in \mathcal{N}_q$ to the updating node, such that it can locally solve the following problem:

$$\min_{\overline{\boldsymbol{W}}} \quad \mathrm{Tr}\left(\overline{\boldsymbol{W}}^H \boldsymbol{R}_{D_q}^i \overline{\boldsymbol{W}}\right) \tag{15a}$$

$$\text{s.t.} \quad \overline{\boldsymbol{W}}^H \boldsymbol{R}_{\overline{\boldsymbol{x}}_q \overline{\boldsymbol{x}}_q}^i \overline{\boldsymbol{W}} = \boldsymbol{I}_Q \tag{15b}$$

where $\boldsymbol{R}_{\overline{\boldsymbol{x}}_q \overline{\boldsymbol{x}}_q}^i$ is the covariance matrix of

$$\overline{\boldsymbol{x}}_q^i \triangleq \begin{bmatrix} \boldsymbol{x}_q^T & \boldsymbol{z}_1^{i\,T} & \cdots & \boldsymbol{z}_{q-1}^{i\,T} & \boldsymbol{z}_{q+1}^{i\,T} \cdots & \boldsymbol{z}_K^{i\,T} \end{bmatrix}^T \tag{16}$$

and

$$\boldsymbol{R}_{D_q}^i \triangleq \mathrm{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_q}, \boldsymbol{R}_{\boldsymbol{z}_1}^i, \ldots,$$
$$\boldsymbol{R}_{\boldsymbol{z}_{q-1}}^i, \boldsymbol{R}_{\boldsymbol{z}_{q+1}}^i, \ldots, \boldsymbol{R}_{\boldsymbol{z}_K}^i) \tag{17}$$

is a block diagonal matrix with $\boldsymbol{R}_{\boldsymbol{z}_k}^i = \mathbb{E}\left\{\boldsymbol{z}_k^i \boldsymbol{z}_k^{i\,T}\right\}$. Note that the solution of (15) can again be found from a GEVD, this time applied to the pencil $(\boldsymbol{R}_{D_q}^i, \boldsymbol{R}_{\overline{\boldsymbol{x}}_q}^i)$. An update rule for the network-wide $\boldsymbol{W}^{i+1}$ emerges naturally by noticing that solving the local problem (15) is equivalent to solving the original centralized problem (13) with additional constraints:

$$\min_{\boldsymbol{W}} \quad \mathrm{Tr}\left(\boldsymbol{W}^H \boldsymbol{R}_D \boldsymbol{W}\right) \tag{18a}$$

$$\text{s.t.} \quad \boldsymbol{W}^H \boldsymbol{R}_{\boldsymbol{xx}} \boldsymbol{W} = \boldsymbol{I}_Q \tag{18b}$$

$$\mathcal{C}(\boldsymbol{W}_k) \subseteq \mathcal{C}(\boldsymbol{W}_k^i) \quad \forall k \neq q \tag{18c}$$

where the operator $\mathcal{C}(\cdot)$ denotes the column space of its argument. These range constraints make (18) different from a traditional nonlinear Gauss-Seidel approach where all $\boldsymbol{W}_k$ would be fixed to $\boldsymbol{W}_k^i$ for $k \neq q$, in which case (18) can not be solved as a GEVD anymore.

The range constraints in (18c) result in the following parametrization of $\boldsymbol{W}$ in (13):

$$\boldsymbol{W} = [(\boldsymbol{W}_q)^T \ (\boldsymbol{W}_1^i \boldsymbol{G}_1)^T \ \cdots \ (\boldsymbol{W}_{q-1}^i \boldsymbol{G}_{q-1})^T$$
$$(\boldsymbol{W}_{q+1}^i \boldsymbol{G}_{q+1})^T \ \cdots \ (\boldsymbol{W}_K^i \boldsymbol{G}_K)^T]^T \quad (19)$$

where the multiplication with $\boldsymbol{G}_k \in \mathbb{R}^{Q \times Q}$ allows to select a new $\boldsymbol{W}_k$ within the column space of $\boldsymbol{W}_k^i$. The elements of this parametrization are therefore related to the solution of (15) through the partitioning

$$\overline{\boldsymbol{W}} = \begin{bmatrix} \boldsymbol{W}_q^T & \boldsymbol{G}_1^T & \cdots & \boldsymbol{G}_{q-1}^T & \boldsymbol{G}_{q+1}^T & \cdots & \boldsymbol{G}_K^T \end{bmatrix}^T \quad (20)$$

The parametrization also defines the update rule for the local estimate of $\boldsymbol{W}_k$ at each node:

$$\boldsymbol{W}_k^{i+1} = \begin{cases} \boldsymbol{W}_q & \text{if } k = q \\ \boldsymbol{W}_k^i \boldsymbol{G}_k & \text{if } k \neq q \end{cases} \quad (21)$$

where $\boldsymbol{W}_q$ and $\boldsymbol{G}_k$ are extracted from the solution of (15) based on the partitioning (20).

Concretely, the procedure at each iteration can be divided into three steps (the detailed procedure is formally described in Algorithm 1):

1) *Aggregation:* Each node $k \in \mathcal{N}_q$ sends a new batch of T observations of the locally compressed signals $\boldsymbol{z}_k^i$ to the updating node q.
2) *Local solution:* The updating node q solves problem (15), expressed exclusively in terms of locally available data. From the solution, it extracts the update of its own local estimate of the solution $\boldsymbol{W}_q$ as well as the update matrices $\boldsymbol{G}_k$ corresponding to each other node, as defined by the partitioning (20). Due to the sign ambiguities of the GEVCs and a potential collapse of the eigenspace in case of GEVLs with multiplicity larger than 1, the solution of (15) is not unique. Therefore, in order to ensure convergence, we select the solution resulting in the smallest difference $\|\boldsymbol{W}_q^{i+1} - \boldsymbol{W}_q^i\|_F$.
3) *Update:* Node q sends the update matrices $\{\boldsymbol{G}_k\}_{k \neq q}$ to each node, which update their local estimates $\boldsymbol{W}_k^i$ according to (21). Note that the transmission cost of these $Q \times Q$ update matrices is negligible compared to the transmission costs in the aggregation step (assuming $T \gg Q^2$). The role of the updating node is finally passed on to node $(q \mod K) + 1$.

If required, the common components $\boldsymbol{s}$ can be estimated at any iteration by performing an in-network summation:

$$\boldsymbol{s}^i = \frac{1}{K} \sum_k \boldsymbol{z}_k^i \quad (22)$$

It is noted that the updating node q has access to all the data required to compute (22), such that no additional bandwidth is required to compute $\boldsymbol{s}$ at node q.

Remark III.1. The batch of T (compressed) samples that is transmitted by each node during the aggregation step would typically consist of different samples than the ones used in the previous iteration, in order to avoid retransmitting the same data multiple times which would substantially increase the bandwidth. Therefore, the time index t corresponding to the first sample of a batch is typically updated as $t + iT'$ at each iteration (with T' possibly smaller than T). This implies that the iterations are spread out over time, and that the algorithm behaves as an adaptive filter tracking the signal statistics over time. In order for the algorithm to converge, we therefore have to assume that the signal statistics change sufficiently slowly compared to the convergence dynamics of the algorithm. To make the convergence analysis mathematically tractable, all convergence proofs in the remaining of this paper implicitly assume that the signal statistics remain stationary during convergence of the algorithm.

Algorithm 1: D-MAXVAR algorithm in a fully connected network.

begin
  $i \leftarrow 0$
  $q \leftarrow 1$
  Randomly initialize $\boldsymbol{W}^0$
  loop
    for $k \in \mathcal{K} \setminus \{q\}$ **do**
      At node k
        Send a new batch of T samples
        $\boldsymbol{z}_k^i[t] = \boldsymbol{W}_k^{iT} \boldsymbol{x}_k[t]$ to node q
    At node q
      Compute $\boldsymbol{R}_{\overline{\boldsymbol{x}}_q}^i$ and $\boldsymbol{R}_{D_q}^i$ based on the received samples and according to (16) and (17)
      Compute the Q GEVCs corresponding to the Q smallest GEVLs of the matrix pencil $(\boldsymbol{R}_{D_q}^i, \boldsymbol{R}_{\overline{\boldsymbol{x}}_q}^i)$, scaled to satisfy constraint (15b) and minimizing $\|\boldsymbol{W}_q^{i+1} - \boldsymbol{W}_q^i\|_F$
      Put the Q resulting GEVCs in the columns of $\overline{\boldsymbol{W}}_q$
      $\boldsymbol{W}_q^{i+1} \leftarrow [\boldsymbol{I}_{M_q} \boldsymbol{0}] \overline{\boldsymbol{W}}_q$
      for $k \in \mathcal{K} \setminus \{q\}$ **do**
        Select $\boldsymbol{G}_k$ as the block of $\overline{\boldsymbol{W}}_q$ (see (20)) corresponding to node k and send to node k
        At node k
          $\boldsymbol{W}_k^{i+1} \leftarrow \boldsymbol{W}_k^i \boldsymbol{G}_k$
    $i \leftarrow i + 1$
    $q \leftarrow (q \mod K) + 1$

B. Convergence and Optimality

In this subsection, we provide some insight in the convergence and optimality properties of the D-MAXVAR algorithm, and establish formal convergence proofs.

A first important observation is that the solution of (18) at iteration i is by definition in the constraint set of the problem at iteration $i+1$ (corresponding to selecting $\boldsymbol{G}_k = \boldsymbol{I}_Q$). As a result, the objective function (18a) decreases monotonically and must therefore converge, as the objective function is bounded in the constraint set. However, the convergence of the sequence of optimization variables $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$ is less straighforward. Indeed, monotonic convergence of the objective function does not imply convergence of its arguments, nor that the global minimum is eventually attained. Nonetheless, by making an

assumption which is always satisfied in practice, we can show several interesting properties about the convergence behavior and limit points of the algorithm.

Assumption III.1. *The accumulation points of the sequence of local pencils $(\boldsymbol{R}_{D_q}^i, \boldsymbol{R}_{\overline{\boldsymbol{x}}_q}^i)_{i \in \mathbb{N}}$ are non singular and have distinct Q-th and $(Q+1)$-th smallest GEVLs.*

Due to the presence of uncorrelated sensor noise, the non-singularity assumption is always verified. The assumption about distinct eigenvalues is merely technical, as we can show that such points corresponding to degenerate local problems would be unstable in practice, unless they correspond to the global solution of (13) (see Theorem III.3 below), and the algorithm would therefore eventually diverge from such points.

In what follows, a fixed point $\boldsymbol{W}^*$ is defined as a point which is invariant under the updates of Algorithm 1, i.e., $(\boldsymbol{W}^i)_{i \in \mathbb{N}} = (\boldsymbol{W}^*)_{i \in \mathbb{N}}$ if $\boldsymbol{W}^0 = \boldsymbol{W}^*$. The following theorem gives an important characterization of the algorithm's fixed points:

Theorem III.1. *The columns of matrices which are fixed points of Algorithm 1 are stationnary points of problem (13) and therefore GEVCs of the pencil $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{xx}})$.*

Proof. See Appendix A. □

Assumption III.1 guarantees that the local problems have well-defined solutions at accumulation points of the algorithm, allowing us to state our main convergence result:

Theorem III.2 (Convergence). *If Assumption III.1 holds, $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$ converges to a fixed point, and hence stationary point, of problem (13).*

Proof. See Appendix B. □

Finally, we show that the global minimizers of (13) are the only stable fixed points. In other words, all convergence trajectories to limit cycles or stationary points where (13) is not minimized are unstable in the sense that the algorithm can be kicked out of such trajectories by infintesimally small perturbations. This is formalized in the following theorem:

Theorem III.3 (Unstable Accumulation Points). *Let $\boldsymbol{W}^*$ be an accumulation point of Algorithm 1. Then $\boldsymbol{W}^*$ is an unstable accumulation point if and only if it is not a global minimizer of problem (13).*

Proof. See Appendix C. □

Therefore, in the presence of numerical noise, we expect the algorithm to converge to a global minimizer of problem (13), as demonstrated by the simulations performed in Section VI.

IV. DISTRIBUTED MAXVAR IN GENERAL NETWORK TOPOLOGIES

Before describing the D-MAXVAR algorithm for more general topologies, we first explain how it can be established in a star topology, thereby introducing some important insights towards further extensions.
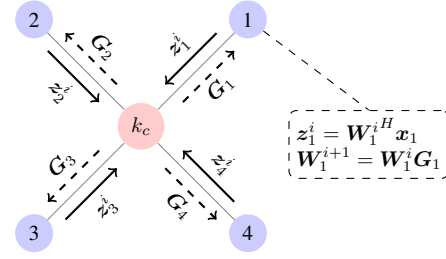
(a) The updating node is the center node ($q = k_c$). As in the fully connected case, one update matrix $\boldsymbol{G}_k^{i+1}$ is computed for each leaf node.
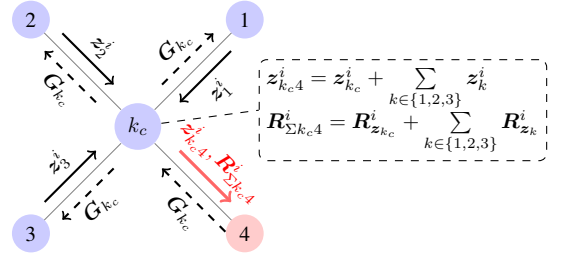
(b) The updating node is a leaf node ($q = 4$). A single update matrix $\boldsymbol{G}_{k_c}^{i+1}$ is computed.

Fig. 1: D-MAXVAR data flow in a star-topology network with $K = 5$

A. Star-Topology Networks

In a star-topology network, we can distinguish two kinds of nodes: the central node k_c, which shares a link with all other nodes in the network, and the leaf nodes $k \in \mathcal{L} = \mathcal{K} \smallsetminus \{k_c\}$, which are exclusively connected to the central node. Therefore,

$$\mathcal{N}_{k_c} = \mathcal{L} \tag{23}$$

$$\mathcal{N}_k = \{k_c\} \qquad \forall k \in \mathcal{L}. \tag{24}$$

A naive strategy to apply the algorithm presented above for fully connected networks to a star-topology network would be to let the central node act as a relay between the leaf nodes. We discard this solution for two reasons: Firstly, the bandwidth required at the central node would grow linearly with the number of nodes in the case of a broadcast communication protocol or quadratically in the case of one-to-one communication. The maximum network size would therefore largely depend on the bandwidth available at the central node. Secondly, further bandwidth savings can be achieved by allowing the central node to compress and fuse the signals it receives from the leaf nodes.

We will apply a separate treatment to the iterations where the updating node is the central node and those where it is a leaf node, of which we give a brief overview hereafter:

a) The updating node is the central node ($q = k_c$)

As all nodes share a link with the updating node, the network proceeds as in the fully-connected case. All nodes send their compressed observations $\boldsymbol{z}_k^i$ to the central node which then solves (15). The leaf nodes update their local estimates of the solution as in the fully-connected case.

b) The updating node is a leaf node ($q \neq k_c$)

As only the central node shares a link with the updating node, it collects the compressed observations of the other leaf nodes. It fuses (i.e. adds) them together with its own observations and sends them to the updating node. The updating node now acts *as if the data received from the central node were*

the compressed observations of a single node and proceeds to compute its local solution estimate and the *single* update matrix $\boldsymbol{G}_{k_c}$ for the central node accordingly. The central node then relays the update matrix $\boldsymbol{G}_{k_c}$ to the other leaf nodes, which then all update their local estimate with this single update matrix. The data flow for star-topology networks is illustrated in Figure 1. Note that the computation of the local solution at node q also requires aggregating and fusing the second order signal statistics through the network similarly to the signal observations, as will be explained below.

Formally, for a star-topology network, when the updating node is a leaf node $q \in \mathcal{L}$, node q receives the following from the central node:

$$\boldsymbol{z}_{k_c q}^i = \boldsymbol{z}_{k_c}^i + \sum_{k \in \mathcal{N}_{k_c} \smallsetminus q} \boldsymbol{z}_k^i \tag{25}$$

$$\boldsymbol{R}_{\Sigma k_c q}^i = \boldsymbol{R}_{\boldsymbol{z}_{k_c}}^i + \sum_{k \in \mathcal{N}_{k_c} \smallsetminus q} \boldsymbol{R}_{\boldsymbol{z}_k}^i \tag{26}$$

It then constructs

$$\overline{\boldsymbol{x}}_q^i \triangleq \begin{bmatrix} \boldsymbol{x}_q^T & \boldsymbol{z}_{k_c q}^{i\,T} \end{bmatrix}^T \tag{27}$$

$$\boldsymbol{R}_{D_q}^i \triangleq \mathrm{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_q}, \boldsymbol{R}_{\Sigma k_c q}^i) \tag{28}$$

as it did in the fully-connected case according to (16)-(17), and solves problem (15). With those new definitions, we can show that solving the local problem (15) is equivalent to solving

$$\min_{\boldsymbol{W}} \quad \mathrm{Tr}\left(\boldsymbol{W}^H \boldsymbol{R}_D \boldsymbol{W}\right) \tag{29a}$$

$$\text{s.t.} \quad \boldsymbol{W}^H \boldsymbol{R}_{\boldsymbol{xx}} \boldsymbol{W} = \boldsymbol{I}_Q \tag{29b}$$

$$\mathcal{C}(\boldsymbol{W}_{-q}) \subseteq \mathcal{C}(\boldsymbol{W}_{-q}^i) \text{ if } q \in \mathcal{L} \tag{29c}$$

$$\mathcal{C}(\boldsymbol{W}_k) \subseteq \mathcal{C}(\boldsymbol{W}_k^i) \,\forall k \in \mathcal{L} \text{ if } q \notin \mathcal{L} \tag{29d}$$

where $\boldsymbol{W}_{-q}$ is the matrix obtained by removing the rows of $\boldsymbol{W}$ corresponding to $\boldsymbol{W}_q$. Note that when the updating node is the central node (i.e. $q \notin \mathcal{L}$), problem (29) reduces to the fully connected case problem (18). When $q \in \mathcal{L}$, constraint (29c) results in the equivalent parametrization

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{W}_q^T & (\boldsymbol{W}_{-q}^i \boldsymbol{G}_{k_c})^T \end{bmatrix}^T \tag{30}$$

where $\boldsymbol{G}_{k_c} \in \mathbb{R}^{Q \times Q}$.

Remark IV.1. Note that the number of degrees of freedom is lower for iterations where the updating node q is a leaf node (constraint (29c) is active) than for iterations where the updating node q is the center node (constraint (29d) is active). We therefore generally expect a lower decrease of the objective function (on average) for iterations in which a leaf node is the updating node.

B. Tree-Topology Networks

A tree-topology network has an acyclic graph, which implies that there is a unique path between any two nodes. The nodes with a single neighbor are also referred to as leaf nodes and constitute the end points of the branches in the tree. As we did for star-topology networks, we denote the set of leaf nodes $\mathcal{L}$.

The procedure in tree-topology networks is conceptually similar to the star-topology case, where the updating node behaves as if it were the center node of a star-topology network, as described in Section IV-A. Consider the two isolated subtrees obtained by disconnecting the updating node

---

Algorithm 2: Recursive procedure for aggregating observations in the branch $\mathcal{B}_{kp}$ to obtain $\boldsymbol{z}_{kp}$

procedure aggregate(k, p)
  for $l \in \mathcal{N}_k \smallsetminus \{p\}$ **do**
    aggregate(l, k)
  At node k
    if $k \notin \mathcal{L}$ **then**
      Compute $\boldsymbol{z}_{kp}[t] = \boldsymbol{z}_k[t] + \sum_{l \in \mathcal{N}_k \smallsetminus p} \boldsymbol{z}_{lk}[t]$ from a new batch of T samples $\boldsymbol{z}_k^i[t] = \boldsymbol{W}_k^{iT} \boldsymbol{x}_k[t]$
      $\boldsymbol{R}_{\Sigma_{kp}} = \boldsymbol{R}_{\boldsymbol{y}_k} + \sum_{l \in \mathcal{N}_k \smallsetminus p} \boldsymbol{R}_{\Sigma_{lk}}$
      Send $(\boldsymbol{z}_{kp}, \boldsymbol{R}_{\Sigma_{kp}})$ to node p
    else
      Send $\boldsymbol{z}_{kp} = \boldsymbol{z}_k$ to node p

---

q from one of its neighbors $k \in \mathcal{N}_q$. We denote $\mathcal{B}_{kq}$ the set of nodes in the subtree containing k (see Figure 2). At each iteration, each of the neighboring nodes $k \in \mathcal{N}_q$ of node q recursively collect and sum the compressed observations and related covariance matrices of their respective subtree $\mathcal{B}_{kq}$ and sends them to node q. The updating node q therefore receives

$$\boldsymbol{z}_{kq}^i \triangleq \sum_{l \in \mathcal{B}_{kq}} \boldsymbol{z}_l^i = \boldsymbol{z}_k^i + \sum_{l \in \mathcal{N}_k \smallsetminus q} \boldsymbol{z}_{lk}^i \quad \forall\, k \in \mathcal{N}_q \tag{31}$$

$$\boldsymbol{R}_{\Sigma kq}^i \triangleq \sum_{l \in \mathcal{B}_{kq}} \boldsymbol{R}_{\boldsymbol{z}_l}^i = \boldsymbol{R}_{\boldsymbol{z}_k}^i + \sum_{l \in \mathcal{N}_k \smallsetminus q} \boldsymbol{R}_{\Sigma lk}^i \quad \forall\, k \in \mathcal{N}_q \tag{32}$$

Note that these definitions are recursive and that those values can be efficiently computed by performing an in-network summation in a recursive fashion. This recursive aggregation procedure is described by Algorithm 2 and illustrated in Figure 3. Note that even in nodes where $Q > M_k$ (e.g. in case of single-channel sensor nodes) this aggregation process realizes an overall bandwidth reduction due to the in-network fusion of data (as opposed to straightforwardly relaying the raw data). Similarly to the other topologies, we define

$$\overline{\boldsymbol{W}}_q \triangleq \begin{bmatrix} \boldsymbol{W}_q^T \mid \boldsymbol{G}_{k_1}^T \mid \cdots \mid \boldsymbol{G}_{k_{n_q}}^T \end{bmatrix}^T \tag{33}$$

$$\overline{\boldsymbol{x}}_q^i \triangleq \begin{bmatrix} \boldsymbol{x}_q^T \mid \boldsymbol{z}_{k_1 q}^{iT} \mid \cdots \mid \boldsymbol{z}_{k_{n_q} q}^{iT} \end{bmatrix}^T \tag{34}$$

$$\boldsymbol{R}_{D_q}^i \triangleq \mathrm{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_q}, \boldsymbol{R}_{\Sigma_{k_1} q}^i, \ldots, \boldsymbol{R}_{\Sigma_{k_{n_q}} q}^i) \tag{35}$$

with $n_q = |\mathcal{N}_q|$ and $\{k_1, \ldots, k_{n_q}\} = \mathcal{N}_q$. This allows us to solve the local problem (15) which is again equivalent to the global problem (13) equipped with additional range constraints, this time for each subtree $\mathcal{B}_{(\cdot)}$. A complete description of the procedure is given by Algorithm 3.

Finally, note that similarly to (22), the common components in $\boldsymbol{s}$ can be estimated as

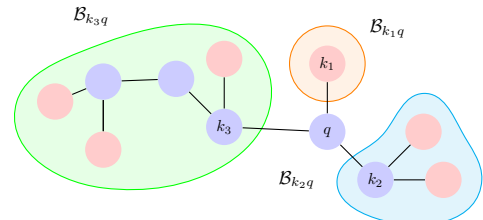$$\boldsymbol{s}^i = \frac{1}{K} \sum_k \boldsymbol{z}_{kq}^i. \tag{36}$$

Fig. 2: In this example tree, the subtree $\mathcal{B}_{k_1 q}$ is highlighted in orange, $\mathcal{B}_{k_2 q}$ in blue and $\mathcal{B}_{k_3 q}$ in green. Leaf nodes are colored red.
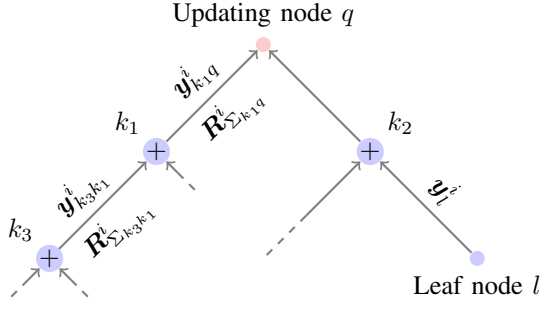
Fig. 3: In-network summing and aggregation at the updating node. Covariance matrices and compressed observations are recursively propagated from the leafs towards the updating node.

Algorithm 3: D-MAXVAR algorithm in a tree-topology network.

begin
  $i \leftarrow 0$
  Initialize updating node as $q \leftarrow 1$
  Randomly initialize the the $\boldsymbol{W}_k^0$'s
  loop
    for $k \in \mathcal{N}_q$ **do**
      $\texttt{aggregate}(k, q)$ (see Algorithm 2)
    At node q
      Compute the Q GEVCs corresponding to the Q smallest GEVLs of the matrix pencil $(\boldsymbol{R}_{D_q}^i, \boldsymbol{R}_{\overline{\boldsymbol{x}}_q}^i)$, scaled to satisfy constraint (15b) and minimizing $\left\| \boldsymbol{W}_q^{i+1} - \boldsymbol{W}_q^i \right\|_F$
      Put the Q resulting GEVCs in the columns of $\overline{\boldsymbol{W}}_q$
      $\boldsymbol{W}_q^{i+1} \leftarrow \left[\boldsymbol{I}_{M_q} \boldsymbol{0} \right] \overline{\boldsymbol{W}}_q$
      for $k \in \mathcal{N}_q$ **do**
        Select $\boldsymbol{G}_k^{i+1}$ as the block of $\overline{\boldsymbol{W}}_q$ (see (33)) corresponding to node k and disseminate within subtree $\mathcal{B}_{kq}$
        for $l \in \mathcal{B}_{kq}$ **do**
          At node l
            $\boldsymbol{W}_l^{i+1} \leftarrow \boldsymbol{W}_l^i \boldsymbol{G}_k$
    $i \leftarrow i + 1$
    $q \leftarrow (q \mod K) + 1$

C. Arbitrary Networks

It is in tree-topology networks that our algorithm offers the greatest benefits in terms of complexity and communication costs. Even though tree-topology networks are quite specific, the D-MAXVAR algorithm can be generalized to networks with arbitrary topologies by pruning the network to a (possibly different) tree in each iteration. Ideally, the tree at iteration i should preserve all the neighbors of the updating node q. Indeed, cutting off neighbors would reduce the number of update matrices $\boldsymbol{G}$ in (33), which would reduce the degrees of freedom in the minimization of (15), using the definitions in (33)-(35).

A similar convergence statement as for fully-connected networks can be made for arbitrary networks, and the associated proof can be relatively straightforwardly adapted to these cases, although the notation and definitions becomes substantially more convoluted.

D. Complexity and Communication Cost

Table I summarizes the communication cost and complexity of our algorithm. The major benefit of the arbitrary-topology variant is that it scales well, i.e. the per-node communication cost and transmission cost is independent of the network size (as opposed to a naive multi-hop relay procedure, which would grow with the depth of the tree). This is because the sensor observations of the different nodes are fused along the way when the data travels through the network, as described in the aggregation step of Algorithm 2.

	Fully-connected	**Arbitrary Topology**
Transmission cost per node	$\propto Q$	$\propto Q$
Complexity at updating node q	$\propto (M_q + Q(K-1))^3$	$\propto (M_q + Q\|\mathcal{N}_q\|)^3$

TABLE I: Communication cost and complexity of the D-MAXVAR algorithm

The above figures assume the transmission cost of the update matrices $\boldsymbol{G}$ to be negligible compared to a batch of T samples.

V. CORRELATION STRUCTURE ESTIMATION

After convergence of the D-MAXVAR algorithm, each node has access to estimates of its own per-node MAXVAR direction $\boldsymbol{z}_k$, of its own spatial filter $\boldsymbol{W}_k$, and of the common components $\boldsymbol{s}$. In this section, we describe how these variables can be used to efficiently approximate the TSC of all node pairs, even for a pair of nodes that do not directly exchange signals with each other. We first describe how the TSC relates to a low-rank approximation of a particular block-whitened correlation matrix (defined below). We then show how this low-rank approximation can be computed from the per-node MAXVAR directions. Finally, we describe a practical solution for selecting an adequate value for Q, the number of computed directions.

A. MAXVAR as a low-rank approximation

The covariance matrix between node k and l of the (per-node) whitened signals is

$$\boldsymbol{P}_{\boldsymbol{x}_k \boldsymbol{x}_l} \triangleq \boldsymbol{R}_{\boldsymbol{x}_k}^{-\frac{1}{2}} \boldsymbol{R}_{\boldsymbol{x}_k \boldsymbol{x}_l} \boldsymbol{R}_{\boldsymbol{x}_l}^{-\frac{1}{2}}. \tag{37}$$

The TSC between nodes k and l can be shown to be equal

$$\Theta_{kl} = \left\| \boldsymbol{P}_{\boldsymbol{x}_k \boldsymbol{x}_l} \right\|_F^2 \tag{38}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. See supplementary materials for a proof.

We now show how $\boldsymbol{W}$, i.e. the MAXVAR solution of (9), relates to the eigenvalue decomposition of $\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} = \boldsymbol{R}_D^{-\frac{1}{2}} \boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{R}_D^{-\frac{1}{2}}$, the network-wide covariance matrix of the (per-node) whitened signals. Note that $\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}}$ is a block matrix such that

$$[\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}}]_{kl} = \boldsymbol{P}_{\boldsymbol{x}_k \boldsymbol{x}_l} \tag{39}$$

where $[\cdot]_{kl}$ denotes the $M_k \times M_l$ block corresponding to node k and l, and where in particular the diagonal blocks $[\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}}]_{kk} = \boldsymbol{I}$. We stated earlier that the spatial filters $\boldsymbol{W}$ were solutions of the GEVD (12). If we assume that $\boldsymbol{\Lambda}$ only contains non-zero GEVLs (which is always the case in practice), we can define

$$\boldsymbol{U} \triangleq \boldsymbol{R}_D^{\frac{1}{2}} \boldsymbol{W} \boldsymbol{\Lambda}^{-\frac{1}{2}}, \tag{40}$$

allowing us to write (12) as

$$U = P_{xx}U\Lambda \qquad (41)$$

$$U^H P_{xx}U = \Lambda^{-1} \qquad (42)$$

which can be reorganized as

$$P_{xx}U = U\Lambda^{-1} \qquad (43a)$$

$$U^H U = I_Q. \qquad (43b)$$

The columns of U are therefore orthogonal eigenvectors of P_{xx} and Λ^{-1} contains the corresponding eigenvalues. As a result, the Q-dimensional filters computed by MAXVAR relate to the eigenvectors corresponding to the Q largest eigenvalues of P_{xx} (i.e. the smallest diagonal elements of Λ) via (40). If we denote the Q-rank approximation of P_{xx} as $\hat{P}_{xx}$, we have

$$\hat{P}_{xx} = \underset{A}{\operatorname{argmin}} \sum_{k,l\in\mathcal{K}} \|P_{x_k x_l} - [A]_{kl}\|_F^2 \qquad (44)$$

$$\text{s.t.} \quad \operatorname{Rank}(A) = Q \qquad (45)$$

$\hat{P}_{xx}$ can therefore be interpreted as a joint low-rank approximation of the matrices $P_{x_k x_l}$.

B. TSC Approximation

We now propose to approximate the TSC by using $\hat{P}_{x_k x_l} \triangleq [\hat{P}_{xx}]_{kl}$ instead of P_{xx} in (38). We define the approximate TSC as

$$\hat{\Theta}_{kl} \triangleq \left\|\hat{P}_{x_k x_l}\right\|_F^2 = \left\|U_k^Q(\Lambda^Q)^{-1}U_l^{QH}\right\|_F^2 \qquad (46)$$

where $U_k = R_{x_k}^{\frac{1}{2}}W_k\Lambda^{-\frac{1}{2}}$ (from the definition of U in (40)), and the superscript Q indicates that we only consider the columns of U and elements of Λ corresponding to the Q largest eigenvalues of P_{xx}. From the definition of the Frobenius norm, we have

$$\left\|U_k^Q(\Lambda^Q)^{-1}U_l^{QH}\right\|_F^2$$
$$= \operatorname{Tr}\left(U_k^Q(\Lambda^Q)^{-1}U_l^{QH}U_l^Q(\Lambda^Q)^{-1}U_k^{QH}\right) \qquad (47)$$

and from the cyclic property of the trace

$$\left\|U_k^Q(\Lambda^Q)^{-1}U_l^{QH}\right\|_F^2$$
$$= \operatorname{Tr}\left(U_k^{QH}U_k^Q(\Lambda^Q)^{-1}U_l^{QH}U_l^Q(\Lambda^Q)^{-1}\right) \qquad (48)$$

From (40), we find that

$$U_k^{QH}U_k^Q = (\Lambda^Q)^{-\frac{1}{2}}W_k^{QH}R_{x_k}^{\frac{1}{2}}R_{x_k}^{\frac{1}{2}}W_k^Q(\Lambda^Q)^{-\frac{1}{2}} \qquad (49)$$

$$= (\Lambda^Q)^{-\frac{1}{2}}R_{z_k}(\Lambda^Q)^{-\frac{1}{2}} \qquad (50)$$

such that we finally obtain

$$\hat{\Theta}_{kl} = \operatorname{Tr}\left(R_{z_k}(\Lambda^Q)^{-2}R_{z_l}(\Lambda^Q)^{-2}\right) \qquad (51)$$

The TSC approximations can therefore be computed from D-MAXVAR's compressed observations' covariance matrices R_{z_k}, which can locally computed at node k and be shared with negligible communication cost compared to the compressed signal observations themselves. Indeed, instead of sharing $M_k T$ samples (T being the window length considered) samples, the nodes only need to share $Q \times Q$ matrices, which is cheap even for nodes several hops away.

We can show that the approximation error in each off-diagonal block of P_{xx} constitutes an upper bound to the TSC approximation error, as stated in the following theorem:

Theorem V.1. *Let $\hat{P}_{x_k x_l}^Q$ denote the block of the best Q-rank approximation of P_{xx} corresponding to nodes k and l. We have*

$$\left(\sqrt{\hat{\Theta}_{kl}^Q} - \sqrt{\Theta_{kl}}\right)^2 \leq \left\|\hat{P}_{x_k x_l}^Q - P_{x_k x_l}\right\|_F^2 \quad \forall Q,k,l \quad (52)$$

Proof. See supplementary materials. $\square$

Note that we are indeed only interested in the off-diagonal blocks, as the diagonal blocks (i.e. $k = l$) are always equal to the identity matrix and do not carry any useful information regarding the network's correlation structure. Theorem V.1 implies that if the network-wide covariance matrix R_{xx} has rank Q, the TSC can be perfectly recovered. This occurs when $\operatorname{Span} s = \operatorname{Span} x$, i.e. x is in the Q-dimensional subspace spanned by s. Furthermore, note that any component in x which is observed by a single node does not contribute to the off-diagonal blocks of P_{xx}. Therefore, if $\operatorname{Span} x_k = \operatorname{Span} s \oplus \operatorname{Span} n_k$, where n_k are node specific signals (which are considered noise for our purposes), we expect $\dim(\operatorname{Span} s) < \dim(\operatorname{Span} x)$ and the off-diagonal blocks of P_{xx} can be described by only $\dim(\operatorname{Span} s)$ components, even if R_{xx} has full rank.

C. A Heuristic for Selecting Q

Note that the TSC estimator defined by (46) relies on the blocks of the low-rank approximation of the full matrix P_{xx} and not on the low-rank approximation of the blocks themselves[2]. What we are effectively minimizing is the total error for all blocks as defined by the joint low-rank approximations (44). From Eckart-Young Theorem [29], we know that this error decreases with increasing Q, but this unfortunately also includes the approximation error of the diagonal blocks, in which we have no interest (as they are always equal to the identity matrix). A pragmatic strategy could be to increase Q as long as the total compound error of the off-diagonal block decreases. The question that then arises is whether one can easily check whether adding a component will result in a decrease of the low-rank approximation error of the off-diagonal blocks. This is stated in the following theorem:

Theorem V.2. *The average approximation error of the off-diagonal blocks of the $(Q + 1)$-rank approximation of P_{xx} is smaller than the average approximation error of the off-diagonal blocks of the Q-rank approximation, i.e.*

$$\sum_{k,l\neq k}\left\|\hat{P}_{x_k x_l}^{Q+1} - P_{x_k x_l}\right\|_F^2 < \sum_{k,l\neq k}\left\|\hat{P}_{x_k x_l}^Q - P_{x_k x_l}\right\|_F^2, \quad (53)$$

if and only if

$$\sum_k\left(\hat{\Theta}_{kk}^{Q+1} - \hat{\Theta}_{kk}^Q\right) > 2\lambda_{Q+1} - \lambda_{Q+1}^2 \qquad (54)$$

where λ_Q denotes the Q largest eigenvalue of P_{xx}[3].

Proof. See supplementary materials. $\square$

Q should therefore be incremented as long as (54) holds and as long as the available bandwidth permits it. Note that this does

---

[2]We remind the reader that this follows from the fact that we wished to avoid the combinatorial complexity scaling which would have resulted from solving all pair-wise CCA problems (each yielding a low-rank approximation of the blocks of P_{xx}).

[3]i.e. the diagonal elements of Λ^{-1}.

not guarantee that the TSC approximation error will decrease in *each* (k, l) pair, only that the total upper bound derived from Theorem V.1 on the total approximation error of TSC will decrease. The advantage of this heuristic is that each node can locally evaluate $\hat{\Theta}_{kk}^Q$ using (51) and the nodes can therefore efficiently verify wether (54) holds.

VI. SIMULATION RESULTS

In this section, we validate the TSC approximation and heuristic described in Section V and demonstrate the convergence properties of the D-MAXVAR algorithm in tree-topology networks.

A. Simulation Settings

In the context of WSNs, it is appropriate to model the observed signals as noisy observations of a mixture of uniformly spatially distributed latent sources, i.e.

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{s} + \alpha \boldsymbol{n}_k \tag{55}$$

where $\boldsymbol{n}_k$ is M_k-dimensional spatially white sensing noise at node k and uncorrelated with the noise at other nodes, $\boldsymbol{s}$ is a d-dimensional spatially white latent signal and $\boldsymbol{A}_k$ is the mixing matrix associated with node k. Finally, α is a network-wide parameter allowing us to modulate the signal-to-noise ratio (SNR). Let $\boldsymbol{A}_k = [\boldsymbol{a}_k^1, \ldots, \boldsymbol{a}_k^d]$, where $\boldsymbol{a}_k^j$ denote the steering vector at node k associated with source s^j. We model it as

$$\boldsymbol{a}_k^j = \frac{\boldsymbol{g}_k^j}{\max\{0.1, \|\boldsymbol{m}_k - \boldsymbol{l}_j\|^2\}} \tag{56}$$

where $\boldsymbol{g}_k^j$ is an M_k-dimensional vector of random variables drawn uniformly from $[0.95, 1.05]$, modelling the slight discrepancies in the channel gains, and $\boldsymbol{m}_k$ and $\boldsymbol{l}_j$ are the random coordinates of node k and source j, uniformly drawn from a 10 by 10 square. With this model the sources can be seen as point sources radiating energy uniformly in all directions. The associated covariance matrice can be computed as

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{A}^H + \alpha^2 \boldsymbol{I} \tag{57}$$

where $\boldsymbol{A}$ is the matrix obtained by stacking the $\boldsymbol{A}_k$'s. The total power of the latent sources picked up by the nodes is $P_s = \|\boldsymbol{A}\|^2$ and the total noise power is $\alpha^2 M$. We set $\alpha^2 = P_s (M\,\text{SNR})^{-1}$ to obtain the desired SNR.

For each simulated scenario, we perfomed 1000 Montecarlo runs where the covariance matrices were directly computed via (57). Additionally, we set $M_k = 8$ and $d = 3$.

B. TSC Approximation

Using the above model and settings with $K = 10$, we obtained the results depicted by Figure 4 with the average absolute error E defined as

$$E \triangleq \frac{1}{K(K-1)} \sum_{k,l \neq k} |\hat{\Theta}_{kl} - \Theta_{kl}| \tag{58}$$

In low SNR regimes the latent subspace is completely shadowed by noise. The average TSC is close to zero and correctly approximated as such. In medium SNR regimes, the latent subspace is observed by most nodes, but in a noisy fashion, such that the subspace in which correlation occurs has a possibly large dimension. Indeed, both the latent subspace and per-node noise components are needed to correctly evaluate the degree of correlation between two nodes. The error is
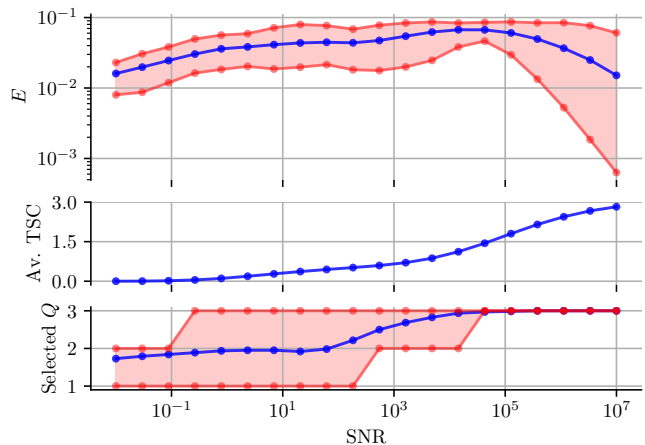

Fig. 4: Montecarlo simulations of the TSC approximation error and selected Q according to the method described in Section V-C for varying SNR values. Blue curves depict the median values. Red shaded areas depict the 5-95% percentile regions. The top plot depicts the average pairwise absolute TSC approximation error E. The middle plot depicts the average pairwise TSC in the network. The bottom plot depicts the number of components Q as selected by the heuristic described in Section V-C.
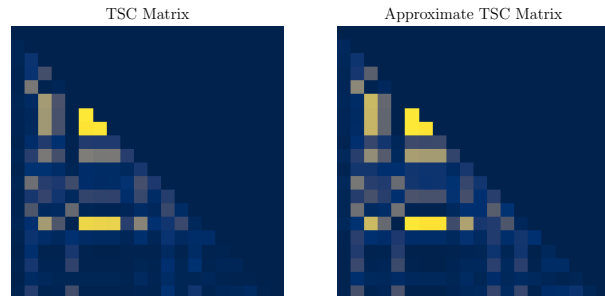

Fig. 5: Example TSC matrices obtained with $K = 20$, $M_k = 8$, 3 latent sources and SNR $= 1$. Each pixel represents the (approximate) TSC of a pair of node. Only the lower triangular half is shown.

therefore largest in this region, but stays reasonable when compared to the true average TSC with a relative error of approximately 3%. In high SNR regimes, the latent subspace is perfectly observed and dominates the noise, which results in an almost perfect approximation of the TSC with a relative error of approximately 1% and decreasing towards 0% with increasing SNR. As expected, the number of components used (Q) increases with the SNR and never over-estimates the dimension of the latent subspace ($\dim \text{Span}\, \boldsymbol{s} = d$). Thanks to this last fact, the algorithm can adaptively set its overall bandwidth to match the SNR. Figure 5 depicts an example matrix of pairwise TSCs for a unit SNR. In this example, we see that nodes sharing common components are correctly identified.

C. Convergence

Figure 6 shows the convergence of the D-MAXVAR algorithm to the global solution. The metric used is
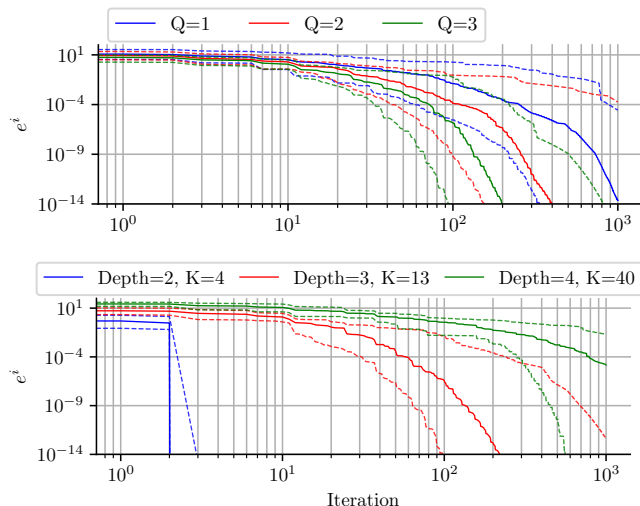
$$e^i = 1 - \frac{f^*}{f(\boldsymbol{W}^i)} \tag{59}$$

Fig. 6: Montecarlo simulations of the convergence of the D-MAXVAR algorithm in tree-topology networks. Solid curves depict the mean values. Shaded areas depict the 5-95% percentile regions. The top plot was generated with $K = 13$. The bottom plots were generated with $K = 4, 13, 40$.

where $f(\boldsymbol{W}) = \boldsymbol{W}^H \boldsymbol{R}_D \boldsymbol{W}$ is the objective function of (13) and f^* is its value at a global minimizer. In the top plot, we see that convergence is indeed faster with a larger number of components Q. In the bottom plot, we see the impact of the tree depth: the deeper the tree, the higher the compression (due to the recursive summing resulting from Algorithm 2). We see that, in practice, convergence to a global minima is always achieved due to the instability of other fixed points (see Theorem III.3).

VII. CONCLUSION AND DISCUSSION

In this paper, we have presented an algorithm which computes the solution of the so-called MAXVAR problem in a distributed setting. Our algorithm displays significant savings in computational and communication requirements compared to a centralized procedure where signal observations are collected at a single location. In particular, in arbitrary-topology networks, the communication cost is independent of the network size and only depends on the degree of each node and the chosen compression factor Q. We have proven the convergence properties of the algorithm, and shown via simulations that the condition for global convergence hold in practice.

We have also shown how the networks' correlation structure can be efficiently estimated from the algorithm's outputs and for negligible additional cost, via an approximation of the total squared cosine. However, this TSC approximation suffers from one drawback: we cannot guarantee that the signal components used for its computation result in the lowest possible approximation error, only that they correspond to the minimum of an upper-bound on the error. Nevertheless, we have demonstrated that the approximation is quite accurate and allows significant computational savings compared to the exact computation of all pair-wise TSCs. Finally, we note that, in most cases, there will be no interest in perfectly estimating the TSC. Indeed, for most applications, the TSC will be thresholded in order to determine whether the link

between two nodes should be kept alive, or simply used as a distance input to some clustering algorithm.

APPENDIX

A. Proof of Theorem III.1

Proof. Let us assume that $\boldsymbol{W}^*$ is a fixed point of Algorithm 1. Then $\boldsymbol{W}^*$ is a solution of problem (18) when $\boldsymbol{W}^i = \boldsymbol{W}^*$ for any $q \in \mathcal{K}$. The Lagrangian of this problem can be expressed as

$$
\begin{aligned}
L_q(\boldsymbol{W}, \boldsymbol{\Lambda}^q, \boldsymbol{\Gamma}_k) = {} & \mathrm{Tr}\left(\boldsymbol{W}^T \boldsymbol{R}_D \boldsymbol{W}\right) \\
& - \mathrm{Tr}\left(\boldsymbol{\Lambda}^q (\boldsymbol{W}^T \boldsymbol{R}_{xx} \boldsymbol{W} - \boldsymbol{I}^Q)\right) \\
& - \sum_{k \neq q} \mathrm{Tr}\left(\boldsymbol{\Gamma}_k \boldsymbol{W}_k^T \boldsymbol{N}_k\right) \quad (60)
\end{aligned}
$$

where $\boldsymbol{\Lambda}^q$ and $\boldsymbol{\Gamma}_k \, \forall k \neq q$ are matrices of proper dimensions containing the Lagrange multipliers and $\boldsymbol{N}_k$ is an $M_k \times (M_k - Q)$ whose columns span the left null space of $\boldsymbol{W}_k^*$. As $\boldsymbol{W}^*$ is a solution of (18), it must satisfy

$$
\frac{\partial}{\partial \boldsymbol{W}} L_q = 0 = 2\boldsymbol{R}_D \boldsymbol{W}^* - 2\boldsymbol{R}_{xx} \boldsymbol{W}^* \boldsymbol{\Lambda}^q - \boldsymbol{N}^q \boldsymbol{\Gamma}^q \quad (61)
$$

where $\boldsymbol{\Gamma}^q$ is the matrix obtained by vertically stacking all the rows of $\boldsymbol{\Gamma}_k$ and $\boldsymbol{N}^q$ is the block diagonal matrix whose blocks are $\boldsymbol{N}_k$, and where the entries of the blocks corresponding to q are set to zero for both matrices. Left-multiplying by $\boldsymbol{W}^{*T}$ and using constraint (18b), we obtain

$$
\boldsymbol{W}^{*T} \boldsymbol{R}_D \boldsymbol{W}^* = \boldsymbol{\Lambda}^q - \frac{1}{2} \underbrace{\boldsymbol{W}^{*T} \boldsymbol{N}^q}_{0} \boldsymbol{\Gamma}^q \quad \forall q. \quad (62)
$$

Since the left-hand side is independent of q, we can conclude that $\boldsymbol{\Lambda}^q = \boldsymbol{\Lambda}$ is the same for every choice of q. From this and (61), we have

$$
\frac{\partial}{\partial \boldsymbol{W}_q} L_q = 0 = 2\boldsymbol{R}_{x_q} \boldsymbol{W}_q^* - 2\boldsymbol{R}_{x_q x} \boldsymbol{W}^* \boldsymbol{\Lambda} - \underbrace{\boldsymbol{N}_q \boldsymbol{\Gamma}_q}_{0} \quad \forall q \quad (63)
$$

Combining those equations for $q \in \mathcal{K}$ yields $\boldsymbol{R}_D \boldsymbol{W}^* = \boldsymbol{R}_{xx} \boldsymbol{W}^* \boldsymbol{\Lambda}$. As $\boldsymbol{W}^*$ is computed via the GEVD of the local pencil $(\boldsymbol{R}_{D_q}^i, \boldsymbol{R}_{\overline{x}_q}^i)$ corresponding to $\boldsymbol{W}^i = \boldsymbol{W}^*$, it must be such that the local $\overline{\boldsymbol{W}}$ in (15) diagonalizes $\boldsymbol{R}_{D_q}^i$ at every updating node q. As a consequence, and considering that the update matrices $\boldsymbol{G}_k$ are identity matrices due to the fact that $\boldsymbol{W}^*$ is a fixed point,

$$
\overline{\boldsymbol{W}}^T \boldsymbol{R}_{D_q}^i \overline{\boldsymbol{W}} = \boldsymbol{W}^{*T} \boldsymbol{R}_D \boldsymbol{W}^* = \boldsymbol{\Lambda} \quad (64)
$$

is a diagonal matrix and the columns of $\boldsymbol{W}^*$ are therefore GEVCs of the pencil $(\boldsymbol{R}_D, \boldsymbol{R}_{xx})$.

$\square$

B. Proof of Theorem III.2

In order to prove Theorem III.2, we first prove two intermediate results. In the following, let $f : \mathbb{R}^M \to \mathbb{R}$ denote the objective function (13a) and let the constraint set (13b) be

$$
\mathcal{D} = \{\boldsymbol{W} \in \mathbb{R}^M \mid \boldsymbol{W}^T \boldsymbol{R}_{xx} \boldsymbol{W} = \boldsymbol{I}_Q\}. \quad (65)
$$

Furthermore, let the constraint set of the local problems (18) be

$$
\mathcal{D}_q(\boldsymbol{V}) = \{\boldsymbol{W} \in \mathcal{D} \mid \mathcal{C}(\boldsymbol{W}_k) \subseteq \mathcal{C}(\boldsymbol{V}_k) \quad \forall k \neq q\} \quad (66)
$$

where $\boldsymbol{V} = \boldsymbol{W}^i$ in (18). For convenience, we define the following equivalent procedure to an update of Algorithm 1:

$$\boldsymbol{W}^{i+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \mathcal{M}^i} \left\| \boldsymbol{W}_{q_i} - \boldsymbol{W}_{q_i}^i \right\|_F \qquad (67a)$$

$$\mathcal{M}^i = \operatorname*{argmin}_{\boldsymbol{W} \in \mathcal{D}_{q_i}(\boldsymbol{W}^i)} f(\boldsymbol{W}) \qquad (67b)$$

where $q_i = i \mod K$. We denote F_q the mapping producing a new iterate $\boldsymbol{W}^{i+1}$ from $\boldsymbol{W}^i$ at the updating node q at iteration i. For simplicity, we assume that (67a) is uniquely defined. If this is not the case, we select the matrix $\boldsymbol{W}^{i+1}$ associated with the update matrices $\boldsymbol{G}_k$ closest to identity matrices with respect to the Frobenius norm.

Lemma B.1. *Let $m_q : \mathbb{R}^{M \times Q} \to \mathbb{R}$ be such that $\{m_q(\boldsymbol{W})\} = f(F_q(\boldsymbol{W}))$, i.e. mapping $\boldsymbol{W}^i$ to $f(\boldsymbol{W}^{i+1})$ in procedure (67). Then m_q is a continuous function for any q at points where the blocks $\boldsymbol{W}_{k \neq q}$ have full column rank.*

Proof. The constraint set of the local problem (15) can be expressed as

$$\mathcal{D}_q(\boldsymbol{W}) = \{P_{\boldsymbol{W}}^q \boldsymbol{V} \mid \boldsymbol{V} \in \mathbb{R}^M\} \cap \mathcal{D} \qquad (68)$$

where $P_{\boldsymbol{W}}^q$ denotes the orthogonal projection matrix on the linear subspace (66). $m_q(\boldsymbol{W})$ therefore corresponds to the sum of the Q smallest GEVLs of $(P_{\boldsymbol{W}}^q \boldsymbol{R}_D P_{\boldsymbol{W}}^{qT}, P_{\boldsymbol{W}}^q \boldsymbol{R}_{\boldsymbol{xx}} P_{\boldsymbol{W}}^{qT})$ (which is found by substituting the optimization variable in (15) by $P_{\boldsymbol{W}}^q \boldsymbol{V}$). As the GEVLs of a pencil vary continuously with the entries of its constituent matrices [30], and as $P_{\boldsymbol{W}}^q$ varies continuously with $\boldsymbol{W}$ at points where its blocks $\boldsymbol{W}_{k \neq q}$ have linearly independent columns [31] (which is true under the assumption that the local problems are non-singular), m_q is a continuous function at points where the blocks $\boldsymbol{W}_{k \neq q}$ have full column rank. $\square$

Let $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$ be any sequence of iterates satisfying the mapping defined by Algorithm 1 and therefore procedure (67). We now show that if $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$ is a convergent sequence, then its limit $\lim_{i \to \infty} \boldsymbol{W}^i = \boldsymbol{W}^*$, is a fixed point of Algorithm 1 and therefore a stationary points of problem (13).

Lemma B.2. *If $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$ is a convergent sequence, it converges to a fixed point of Algorithm 1.*

Proof. Let $\boldsymbol{W}^*$ be the unique limit of $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$. From the continuity of m_q (see Lemma B.1), and under the assumption of non-singular local pencils, we have:

$$\lim_{\boldsymbol{W} \to \boldsymbol{W}^*} m_k(\boldsymbol{W}) = m_k(\boldsymbol{W}^*) \quad \forall k \in \mathcal{K}. \qquad (69)$$

As $(f(\boldsymbol{W}^i))_{i \in \mathcal{N}}$ is bounded and decreases monotonically, it converges to some f^* and therefore, from the continuity of f, $f(\boldsymbol{W}^*) = f^*$. By definition of m_q we have

$$m_{q_i}(\boldsymbol{W}^i) = f(\boldsymbol{W}^{i+1}) \qquad (70)$$

with $q_i = i \mod K$. Therefore,

$$\lim_{i \to \infty} m_{q_i}(\boldsymbol{W}^i) = f^* \qquad (71)$$

and, in particular, every subsequence such that $q_i = k$ for some node k converges to f^* and

$$m_k(\boldsymbol{W}^*) = f^* = f(\boldsymbol{W}^*) \quad \forall k \in \mathcal{K}. \qquad (72)$$

As, by definition (66), $\boldsymbol{W}^*$ is in $\mathcal{D}_k(\boldsymbol{W}^*)$, and by the definition of m_q, the minimum value of f in $\mathcal{D}_k(\boldsymbol{W}^*)$ is $m_k(\boldsymbol{W}^*) = f^*$, $\boldsymbol{W}^*$ must be in $\mathcal{M}$ (as defined in (67b)).

In virtue of (67a), it must be that $F_k(\boldsymbol{W}^*) = \boldsymbol{W}^*$ and $\boldsymbol{W}^*$ is therefore a fixed point of F_k for any node k. $\square$

We can now finally prove Theorem III.2.
As, by hypothesis, the Q-th and $(Q+1)$-th smallest GEVLs of the local pencils $(\boldsymbol{R}_{D_k}^i, \boldsymbol{R}_{\overline{\boldsymbol{x}}_k}^i)$ are distinct at the accumulations points of $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$, a small pertubartion of the pencil around an accumulation point results in a small perturbation of the generalized eigenspace [6], [30][4]. As a consequence, the convergence of some subsequence $(\boldsymbol{W}^i)_{i \in \mathcal{N}}$ to $\boldsymbol{W}^*$ (which is a stationnary point) implies the convergence of $(\mathcal{M}^i)_{i \in \mathcal{N}}$ to some $\mathcal{M}^*$, which from (67) corresponds to the sequence of sets of generalized eigenvectors of the local pencils $(\boldsymbol{R}_{\overline{\boldsymbol{x}}_q}^i, \boldsymbol{R}_{D_q}^i)$, where the convergence of $(\mathcal{M}^i)_{i \in \mathcal{N}}$ must be understood in terms of the Haussdorf distance between sets [5].

As the sublevel sets of f in $\mathcal{D}$ are compact and f decreases monotonically, there indeed exists a convergent subsequence $(\boldsymbol{W}^i)_{i \in \mathcal{N}}$ converging to some $\boldsymbol{W}^*$, which defines a corresponding $\mathcal{M}^*$ according to (67b). As $m_k(\boldsymbol{W}^*) = f^* \, \forall k \in \mathcal{K}$, it must be that $\boldsymbol{W}^* \in \mathcal{M}^*$. Therefore, there exists some convergent sequence $\{\boldsymbol{V}^{i+1}\}_{i \in \mathcal{N}}$ with $\boldsymbol{V}^{i+1} \in \mathcal{M}^i$ converging to $\boldsymbol{W}^*$ (As the convergence of $(\mathcal{M}^i)_{i \in \mathcal{N}}$ to $\mathcal{M}^*$ implies that for any point $\boldsymbol{W}$ in $\mathcal{M}^*$ we can find a set $\mathcal{M}^i \in (\mathcal{M}^i)_{i \in \mathcal{N}}$ containing a point arbitrarily close to $\boldsymbol{W}$). As both sequences converge to the same point,

$$\lim_{i \to \infty, i \in \mathcal{N}} \left\| \boldsymbol{W}^i - \boldsymbol{V}^{i+1} \right\|_F = 0, \qquad (73)$$

and as $\boldsymbol{V}^{i+1} \in \mathcal{M}^i$, we have from (67a) that

$$\left\| \boldsymbol{V}_{q_i}^{i+1} - \boldsymbol{W}_{q_i}^i \right\|_F \geq$$
$$\min_{\boldsymbol{W} \in \mathcal{M}^i} \left\| \boldsymbol{W}_{q_i} - \boldsymbol{W}_{q_i}^i \right\|_F = \left\| \boldsymbol{W}_{q_i}^{i+1} - \boldsymbol{W}_{q_i}^i \right\|_F. \quad (74)$$

(73) in combination with the squeeze theorem therefore implies that

$$\lim_{i \to \infty, i \in \mathcal{N}} \left\| \boldsymbol{W}_{q_i}^{i+1} - \boldsymbol{W}_{q_i}^i \right\|_F = 0. \qquad (75)$$

Without loss of generality, we could have chosen $\mathcal{N}$ such that $(q_i)_{i \in \mathcal{N}} = (k)_{i \in \mathcal{N}}$ for any abritrary node k. Indeed, because the sublevel sets of f in $\mathcal{D}$ are compact any sequence $(\boldsymbol{W}^i)_i$ is guaranteed to have a convergent subsequence, and in particular sequences such that $(q_i)_{i \in \mathcal{N}}$ is a constant sequence. As a consequence of this and (75), for every node k, there exists some set of indices $\mathcal{N}_k$ such that $(\boldsymbol{W}_k^i)_{i \in \mathcal{N}_k}$ converges to some $\boldsymbol{W}_k^*$ and

$$\lim_{i \to \infty, i \in \mathcal{N}_k} \left\| \boldsymbol{W}_k^{i+1} - \boldsymbol{W}_k^i \right\|_F = 0. \qquad (76)$$

The convergence of $(\boldsymbol{W}_k^i)_{i \in \mathcal{N}_k}$ to $\boldsymbol{W}_k^*$ therefore implies the convergence of $(\boldsymbol{W}_k^{i+1})_{i \in \mathcal{N}_k}$ to the same point. By induction, it must be that $(\boldsymbol{W}_k^{i+l})_{i \in \mathcal{N}_k}$ converges to $\boldsymbol{W}_k^*$ for any offset l and $(\boldsymbol{W}_k^i)_{i \in \mathbb{N}}$ is therefore a convergent sequence (as $\cup_{l \in \mathbb{N}} \{\mathcal{N} + l\} = \mathbb{N}$). As this is true for every node k, every block of the solution converges and $(\boldsymbol{W}^i)_{i \in \mathbb{N}}$ is a convergent

---

[4] The result is established for non-generalized eigenspaces, but it can be straightforwardly extended to generalized eigenspaces by performing a change of variables similar to (40) in order to turn the GEVD into an equivalent eigenvalue decomposition.

[5] The Haussdorf distance $d_H(\mathcal{V}, \mathcal{W})$ between two sets $\mathcal{V}$ and $\mathcal{W}$ is defined as $\max\{\sup_{X \in \mathcal{V}} \inf_{Y \in \mathcal{W}} \|X - Y\|, \sup_{Y \in \mathcal{V}} \inf_{X \in \mathcal{W}} \|X - Y\|\}$. $d_H(\mathcal{V}, \mathcal{W}) = \varepsilon$ implies that for any point in $\mathcal{V}$ we can find a point in $\mathcal{W}$ at a distance at most ε and vice-versa.

sequence which, from Lemma B.2, converges to a fixed point, and hence to a stationary point (see Theorem III.1). $\qquad\square$

C. Proof of Theorem III.3

Since $\boldsymbol{W}^*$ is not a global minimizer and since (13) has no local minima[6], every neighborhood $\mathcal{V} \subseteq \mathcal{D}$ around $\boldsymbol{W}^*$ contains a continuum of points $\boldsymbol{U} \in \mathcal{V}$ for which $f(\boldsymbol{U}) < f(\boldsymbol{W}^*)$. Now take any point $\boldsymbol{U}$ in $\mathcal{U}$. Since $(f(\boldsymbol{W}^i))_{i\in\mathbb{N}}$ is monotonically decreasing, setting $\boldsymbol{W}^0 = \boldsymbol{U}$ will result in a sequence $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ that remains at a finite distance from $\boldsymbol{W}^*$. Therefore, $\boldsymbol{W}^*$ cannot be a stable accumulation point. $\qquad\square$

REFERENCES

[1] C. Hovine and A. Bertrand, "Distributed MAXVAR: Identifying common signal components across the nodes of a sensor network," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 2159–2163.

[2] S. Haykin and K. R. Liu, *Handbook on array processing and sensor networks*. John Wiley & Sons, 2010, vol. 63.

[3] N. Vlajic and D. Xia, "Wireless sensor networks: to cluster or not to cluster?" in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*. IEEE, 2006, pp. 9–pp.

[4] D. Xia and N. Vlajic, "Near-optimal node clustering in wireless sensor networks for environment monitoring," in *21st International Conference on Advanced Information Networking and Applications (AINA'07)*. IEEE, 2007, pp. 632–641.

[5] X. Sun and Q. Cheng, "On subspace distance," in *International Conference Image Analysis and Recognition*. Springer, 2006, pp. 81–89.

[6] G. H. Golub and C. F. Van Loan, *Matrix computations*, 4th ed. John Hopkins University press, 2013.

[7] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[8] K. Fukunaga and W. L. Koontz, "Representation of random processes using the finite karhunen-loeve expansion," *Information and Control*, vol. 16, no. 1, pp. 85–101, 1970.

[9] P. Horst, *Generalized canonical correlations and their application to experimental data*. Journal of clinical psychology, 1961, no. 14.

[10] J. D. Carroll, "Generalization of canonical correlation analysis to three or more sets of variables," in *Proceedings of the 76th annual convention of the American Psychological Association*, vol. 3. Washington, DC, 1968, pp. 227–228.

[11] J. R. Kettenring, "Canonical analysis of several sets of variables," *Biometrika*, vol. 58, no. 3, pp. 433–451, 1971.

[12] M. Sørensen, C. I. Kanatsoulis, and N. D. Sidiropoulos, "Generalized canonical correlation analysis: A subspace intersection approach," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2452–2467, 2021.

[13] L. Li, A. Scaglione, and J. H. Manton, "Distributed principal subspace estimation in wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 725–738, 2011.

[14] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed PCA," *Signal Processing*, vol. 104, pp. 120–135, 2014.

[15] ——, "Distributed adaptive generalized eigenvector estimation of a sensor signal covariance matrix pair in a fully connected sensor network," *Signal Processing*, vol. 106, pp. 209–214, 2015.

[16] A. Hassani, A. Bertrand, and M. Moonen, "GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2557–2572, 2015.

[17] J. Chen and I. D. Schizas, "Distributed information-based clustering of heterogeneous sensor data," *Signal Processing*, vol. 126, pp. 35–51, 2016.

[18] A. Bertrand and M. Moonen, "Distributed canonical correlation analysis in wireless sensor networks with application to distributed blind source separation," *IEEE Transactions on Signal Processing*, vol. 63, no. 18, pp. 4800–4813, 2015.

[19] X. Fu, K. Huang, E. E. Papalexakis, H. A. Song, P. Talukdar, N. D. Sidiropoulos, C. Faloutsos, and T. Mitchell, "Efficient and distributed generalized canonical correlation analysis for big multiview data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2304–2318, 2018.

[20] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in statistics*. Springer, 1992, pp. 162–190.

[21] T. De Bie, N. Cristianini, and R. Rosipal, "Eigenproblems in pattern recognition," in *Handbook of Geometric Computing*. Springer, 2005, pp. 129–167.

[22] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[23] M. O'Searcoid, *Metric spaces*. Springer Science & Business Media, 2006.

[24] G. Zuccon, L. A. Azzopardi, and C. Van Rijsbergen, "Semantic spaces: Measuring the distance between different subspaces," in *International Symposium on Quantum Interaction*. Springer, 2009, pp. 225–236.

[25] K. Ye and L.-H. Lim, "Schubert varieties and distances between subspaces of different dimensions," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 1176–1197, 2016.

[26] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.

[27] J. Vía, I. Santamaría, and J. Pérez, "A learning algorithm for adaptive canonical correlation analysis of several data sets," *Neural Networks*, vol. 20, no. 1, pp. 139–152, 2007.

[28] A. H. Sameh and J. A. Wisniewski, "A trace minimization algorithm for the generalized eigenvalue problem," *SIAM Journal on Numerical Analysis*, vol. 19, no. 6, pp. 1243–1259, 1982.

[29] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[30] T. Kato, *Perturbation theory for linear operators*. Springer Science & Business Media, 2013, vol. 132.

[31] G. Forchini, "A note on the continuity of projection matrices with application to the asymptotic distribution of quadratic forms," *Applicationes Mathematicae*, vol. 1, no. 32, pp. 51–55, 2005.

---

[6]Matrices spanning the generalized eigenspaces of $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{xx}})$ corresponding to the Q largest or smallest GEVLs, are global maximizers and minimizers of (13). Matrices spanning the generalized eigengspaces corresponding to any other combination of GEVLs are saddle points.