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 [3]. 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 [4]. Furthermore, the FC constitutes a single point of failure which compromises the robustness of the network [4], [5]. 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 [6], [7].

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 [8], [9]. 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) [10] and the Karhunen–Loève transform (KLT) [11], which extract 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 [12]–[14], 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

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 802895) and from the Flemish Government under the ”Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme. A conference precursor of this manuscript has been published in [1].

[15], 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 [14]. The MAXVAR formulation historically refers to the objective function introduced by Horst [12]. Carroll [13] 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 and adaptive algorithm for tracking 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 use the later as a basis for extending the algorithm 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 [16], [17], maximal SNR [18], [19] or maximizing correlation between two sets of signals [20], [21]. 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 [22] 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 [22] was on computational efficiency, rather than on bandwidth scaling, as is the case of the present work. Finally, as an alternative to distributed methods, several works investigate the applicability of CCA-like techniques to scenarios involving two sensor arrays and where only a limited number of samples is available at each node [23]–[25].

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 general topologies in Section IV. Section V discusses how a specific subset of 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 $k \in \mathcal{K} = \{1, \dots, K\}$ collects discrete observations of a real-valued M_k-channel sensor signal $\boldsymbol{x}_k = [x_{k,1}, \dots, x_{k,M_k}]^T$. 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:

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

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 [26]. 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

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

$$\text{s.t.} \quad z_{k,i} = \boldsymbol{w}_{k,i}^T\boldsymbol{x}_k,\ z_{l,i} = \boldsymbol{w}_{l,i}^T\boldsymbol{x}_l \tag{2b}$$

$$\mathbb{E}\left\{z_{k,i}^2\right\} = \mathbb{E}\left\{z_{l,i}^2\right\} = 1 \tag{2c}$$

$$\mathbb{E}\left\{z_{k,i}[z_{k,1}, \dots, z_{k,i-1}]\right\} = 0 \tag{2d}$$

$$\mathbb{E}\left\{z_{l,i}[z_{l,1}, \dots, z_{l,i-1}]\right\} = 0 \tag{2e}$$

The three last constraints require the canonical directions to have unit-variance and be orthogonal with respect to each other within each node. 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

$$\boldsymbol{z}_{kl} = [z_{k,1}, \dots, z_{k,M_{kl}}]^T = \boldsymbol{W}_{kl}^T\boldsymbol{x}_k \tag{3}$$

with $M_{kl} = \min(M_k, M_l)$ and $\boldsymbol{W}_{kl} \in \mathbb{R}^{M_k \times M_{kl}}$, it can be shown [27] that the canonical directions and coefficients between two nodes k and l are the solutions of the following generalized eigenvalue problem:

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

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

where $\boldsymbol{R}_{\boldsymbol{x}_k}$ is a shorthand notation for $\boldsymbol{R}_{\boldsymbol{x}_k\boldsymbol{x}_k}$ and $\boldsymbol{\Lambda}_{kl}$ 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 total correlation.

C. Total Squared Correlation

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

$$\Theta_{kl} \triangleq \sum_{i=1}^{M_{kl}} (\rho_i^{kl})^2 = \mathrm{Tr}\left(\boldsymbol{\Lambda}_{kl}^2\right), \tag{5}$$

where $\mathrm{Tr}\left(\cdot\right)$ denotes the trace operator. (5) can be interpreted as a generalization of correlation between random variables to correlation between sets of random variables. Indeed, if we denote $\mathcal{X}_k = \mathrm{Span}\, \boldsymbol{x}_k$, i.e. the signal subspace spanned by the channels of $\boldsymbol{x}_k$, we have:

$$\Theta_{kl} = 0 \Leftrightarrow \mathcal{X}_k \perp \mathcal{X}_l \tag{6}$$

$$\Theta_{kl} = M_{kl} \Leftrightarrow \mathcal{X}_k \subseteq \mathcal{X}_l \tag{7}$$

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 [28]. Still, in the case where the channels of $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$ are linearly independent (which is in practice always true in noisy settings), it can easily be turned into one by defining

$$d_{kl} = \sqrt{\max\{M_k, M_l\} - \Theta_{kl}} \tag{8}$$

which satisfies the triangle equality [29] and corresponds to the so-called Chordal distance in the particular case where $M_k = M_l$ [30]. Considering these above properties, the TSC is an adequate metric for characterizing the degree to which the signals of two nodes correlate, which is one of the goals 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 [31].

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 [13] and defined as follows for K signal subspaces:

$$\min_{\{\boldsymbol{W}_k\}} \min_{\boldsymbol{s}} \sum_{k=1}^{K} \mathbb{E}\left\{ \|\boldsymbol{s} - \boldsymbol{z}_k\|^2 \right\} \tag{9a}$$

$$\text{s.t.} \quad \boldsymbol{z}_k = \boldsymbol{W}_k^T \boldsymbol{x}_k \tag{9b}$$

$$\mathbb{E}\left\{\boldsymbol{s}\boldsymbol{s}^T\right\} = \boldsymbol{I}_Q \tag{9c}$$

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 $\boldsymbol{W}_k \in \mathbb{C}^{M_k \times Q}$ per node such that the filtered observations $\boldsymbol{z}_k$, which we refer to as the per-node *MAXVAR directions*, are as close as possible to some common network-wide Q-dimensional signal $\boldsymbol{s}$, 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 $\boldsymbol{s}$ onto the nodes' signal subspaces $\mathrm{Span}(\boldsymbol{x}_k)$. Hence, $\mathrm{Span}(\boldsymbol{s})$ is the Q-dimensional subspace whose signals have minimal average projection error onto the nodes' individual signal subspaces.

In [32], it is shown that the solution of (9) satisfies

$$\boldsymbol{s} = \frac{1}{K} \sum_k \boldsymbol{z}_k = \boldsymbol{W}^T \boldsymbol{x} \tag{10}$$

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

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

The problem therefore consists in 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 [32] that the $\boldsymbol{W}_k$'s are solutions to the following eigenvalue problem:

$$\boldsymbol{R}_D \boldsymbol{W} = \boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{W} \boldsymbol{\Lambda} \tag{12a}$$

$$\boldsymbol{W}^T \boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{W} = \boldsymbol{I}_Q \tag{12b}$$

where $\boldsymbol{R}_D \triangleq \mathrm{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_1}, \ldots, \boldsymbol{R}_{\boldsymbol{x}_K})$ is the block diagonal matrix containing the node-specific covariance matrices and $\boldsymbol{\Lambda}$ is a diagonal matrix. Solving problem (9) is therefore equivalent to computing the generalized eigenvalue decomposition (GEVD) of the matrix pencil $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}})$, 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 procedure based on MAXVAR. Indeed, we will show in Section V-A that solving a slightly modified version of the MAXVAR problem produces a joint approximation of the TSC of each node pair. As it is useful in its own right and of more general interest, we will first describe a distributed procedure for computing the solutions of the classical MAXVAR problem (12). In order to obtain such a solution, 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 have access to an estimate of the solutions, thereby avoiding the need for a fusion center altogether. In Section V, we describe how the distributed MAXVAR procedure can be modified to produce an approximation of (5).

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). Relying on our interest in a subset of the 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 $\mathcal{N}_k = \mathcal{K} \setminus \{k\}$.

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 $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}})$ 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 [33] 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[2]

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

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

where $\mathrm{Tr}(\cdot)$ 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:

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

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

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

$$\overline{\boldsymbol{x}}_q^i \triangleq \left[\boldsymbol{x}_q^T \quad {\boldsymbol{z}_1^i}^T \quad \cdots \quad {\boldsymbol{z}_{q-1}^i}^T \quad {\boldsymbol{z}_{q+1}^i}^T \cdots \quad {\boldsymbol{z}_K^i}^T\right]^T \tag{16}$$

and

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

[2] Although MAXVAR is also often expressed as a trace maximization problem, we write it as a minimization problem to stay consistent with Carroll's formulation (9).

is a block diagonal matrix with $\boldsymbol{R}^i_{\boldsymbol{z}_k} = \mathbb{E}\left\{\boldsymbol{z}^i_k \boldsymbol{z}^{i\,T}_k\right\}$. Note that the solution of (15) can again be found from a GEVD, this time applied to the pencil $(\boldsymbol{R}^i_{D_q}, \boldsymbol{R}^i_{\tilde{\boldsymbol{x}}_q})$. 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:

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

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

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

where the operator $\mathcal{C}(\cdot)$ denotes the column space of its argument. Indeed, notice that the range constraints in (18c) can be equivalently formulated by introducing the following parametrization of $\boldsymbol{W}$ in (13):

$$\begin{aligned}\boldsymbol{W} = [(\boldsymbol{W}_q)^T \; (\boldsymbol{W}^i_1 \boldsymbol{G}_1)^T \; \cdots \; (\boldsymbol{W}^i_{q-1} \boldsymbol{G}_{q-1})^T \\ (\boldsymbol{W}^i_{q+1} \boldsymbol{G}_{q+1})^T \; \cdots \; (\boldsymbol{W}^i_K \boldsymbol{G}_K)^T]^T\end{aligned} \tag{19}$$

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 $\boldsymbol{W}^i_k$. With this parametrization, (18a) becomes

$$\begin{aligned}&\mathrm{Tr}\left(\boldsymbol{W}^T \boldsymbol{R}_D \boldsymbol{W}\right) \\ &= \mathrm{Tr}\left(\boldsymbol{W}^T_q \boldsymbol{R}_{\boldsymbol{x}_q} \boldsymbol{W}_q\right) + \sum_{k\neq q} \mathrm{Tr}\left(\boldsymbol{G}^T_k \underbrace{\boldsymbol{W}^{iT}_k \boldsymbol{R}_{\boldsymbol{x}_k} \boldsymbol{W}^i_k}_{=\boldsymbol{R}^i_{\boldsymbol{z}_k}} \boldsymbol{G}_k\right).\end{aligned} \tag{20}$$

Finally introducing $\overline{\boldsymbol{W}}$ as

$$\overline{\boldsymbol{W}} = \left[\boldsymbol{W}_q{}^T \; \boldsymbol{G}_1{}^T \; \cdots \; \boldsymbol{G}_{q-1}{}^T \; \boldsymbol{G}_{q+1}{}^T \; \cdots \; \boldsymbol{G}_K{}^T\right]^T \tag{21}$$

and substituting in (20), we indeed obtain the objective of the local problem (15a). A similar derivation can be done for the constraints (18c). The parametrization therefore indeed defines the update rule for the local estimate of $\boldsymbol{W}_k$ at each node:

$$\boldsymbol{W}^{i+1}_k = \begin{cases} \boldsymbol{W}_q & \text{if} k = q \\ \boldsymbol{W}^i_k \boldsymbol{G}_k & \text{if} k \neq q \end{cases} \tag{22}$$

where $\boldsymbol{W}_q$ and $\boldsymbol{G}_k$ are extracted from the solution of (15) based on the partitioning (21). Note that the range constraints make (18) different from a traditional nonlinear Gauss-Seidel approach where all $\boldsymbol{W}_k$ would be fixed to $\boldsymbol{W}^i_k$ for $k \neq q$, in which case (18) could not be solved as a GEVD anymore.

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 $\boldsymbol{z}^i_k$ 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 (21). Due to the sign ambiguities of the GEVCs[3], the solution of (15) is not unique and independent from the sign changes of its columns. Therefore, in order to ensure convergence, we select the solution resulting in the smallest difference $\left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F$. In practice, this amounts to checking whether
$$\left\|\overline{\boldsymbol{w}}^{i+1} - \overline{\boldsymbol{w}}^i\right\|_F < \left\|\overline{\boldsymbol{w}}^{i+1} + \overline{\boldsymbol{w}}^i\right\|_F$$
for each of the columns $\overline{\boldsymbol{w}}^{i+1}$ and $\overline{\boldsymbol{w}}^i$ of $\overline{\boldsymbol{W}}$ and $[\boldsymbol{W}^{iT}_q \; \boldsymbol{I}_Q \; \cdots \boldsymbol{I}_Q]^T$, respectively, and choosing the signs of the columns accordingly.
3) *Update:* Node q sends the update matrices $\{\boldsymbol{G}_k\}_{k\neq q}$ to each node, which update their local estimates $\boldsymbol{W}^i_k$ according to (22). 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:

$$\boldsymbol{s}^i = \frac{1}{K} \sum_k \boldsymbol{z}^i_k \tag{23}$$

It is noted that the updating node q has access to all the data required to compute (23), 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.

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

[3] Multiple solutions could also appear due to a potential (and extremely unlikely) collapse of the eigenspace resulting from GEVLs with multiplicity larger than 1, but the update should in this case be skipped as it violates Assumption III.1 (described in the next subsection).

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 $\boldsymbol{R}^i_{\overline{\boldsymbol{x}}_q}$ and $\boldsymbol{R}^i_{D_q}$ 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
              $(\boldsymbol{R}^i_{D_q}, \boldsymbol{R}^i_{\overline{\boldsymbol{x}}_q})$, scaled to satisfy constraint (15b)
              and minimizing $\left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F$
            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 $k \in \mathcal{K} \setminus \{q\}$ do
                Select $\boldsymbol{G}_k$ as the block of $\overline{\boldsymbol{W}}_q$ (see (21))
                  corresponding to node $k$ and send to node
                  $k$
                At node $k$
                    $\boldsymbol{W}_k^{i+1} \leftarrow \boldsymbol{W}_k^i \boldsymbol{G}_k$
        $i \leftarrow i + 1$
        $q \leftarrow (q \mod K) + 1$
```

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 straightforward. 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* $\left((\boldsymbol{R}^i_{D_q}, \boldsymbol{R}^i_{\overline{\boldsymbol{x}}_q})\right)_{i\in\mathbb{N}}$ *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 stationary 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 infinitesimally 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.

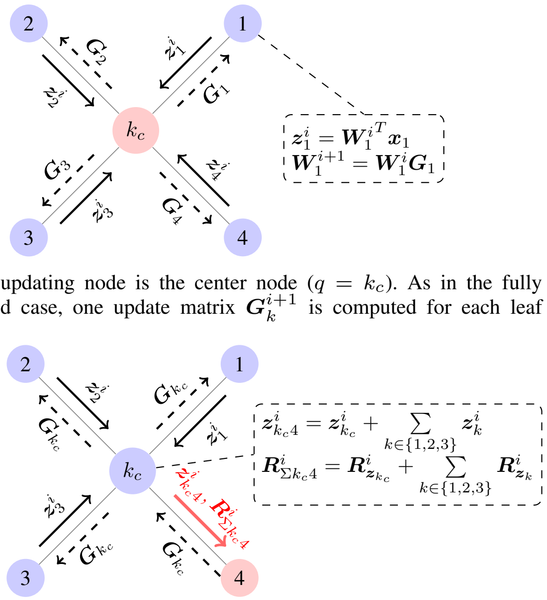
A. Star-Topology Networks

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 $k \in \mathcal{L} = \mathcal{K} \setminus \{k_c\}$, which are exclusively connected to the central node. Therefore,

$$\mathcal{N}_{k_c} = \mathcal{L} \tag{24}$$

$$\mathcal{N}_k = \{k_c\} \qquad \forall k \in \mathcal{L}. \tag{25}$$

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.



(a) The updating node is the center node ($q = k_c$). As in the fully connected case, one update matrix G_k^{i+1} is computed for each leaf node.

(b) The updating node is a leaf node ($q = 4$). A single update matrix $G_{k_c}^{i+1}$ is computed.

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

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 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 G_{k_c} for the central node accordingly. The central node then relays the update matrix 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:

$$z^i_{k_c q} = z^i_{k_c} + \sum_{k \in \mathcal{N}_{k_c} \setminus q} z^i_k \tag{26}$$

$$R^i_{\Sigma k_c q} = R^i_{z_{k_c}} + \sum_{k \in \mathcal{N}_{k_c} \setminus q} R^i_{z_k} \tag{27}$$

It then constructs

$$\overline{x}^i_q \triangleq \left[x_q^T \quad {z^i_{k_c q}}^T \right]^T \tag{28}$$

$$R^i_{D_q} \triangleq \text{Blkdiag}(R_{x_q}, R^i_{\Sigma k_c q}) \tag{29}$$

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

$$\min_{W} \quad \text{Tr}\left(W^T R_D W \right) \tag{30a}$$

$$\text{s.t.} \quad W^T R_{xx} W = I_Q \tag{30b}$$

$$\mathcal{C}(W_{-q}) \subseteq \mathcal{C}(W^i_{-q}) \text{ if } q \in \mathcal{L} \tag{30c}$$

$$\mathcal{C}(W_k) \subseteq \mathcal{C}(W^i_k)\ \forall k \in \mathcal{L} \text{ if } q \notin \mathcal{L} \tag{30d}$$

where W_{-q} is the matrix obtained by removing the rows of W corresponding to W_q. Note that when the updating node is the central node (i.e. $q \notin \mathcal{L}$), problem (30) reduces to the fully connected case problem (18). When $q \in \mathcal{L}$, constraint (30c) results in the equivalent parametrization

$$W = \left[{W_q}^T \quad (W^i_{-q} G_{k_c})^T \right]^T \tag{31}$$

where $G_{k_c} \in \mathbb{R}^{Q \times Q}$.

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 (30c) is active) than for iterations where the updating node q is the center node (constraint (30d) 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 $\mathcal{L}$. Note that the tree concept should not be viewed as an actual topology constraint, but instead as a framework to organize the data streams within the network, i.e., to define in which order nodes have to transmit their data through the network, which will be essential when extending our algorithm to general topologies.

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 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 collects and sums the compressed observations and

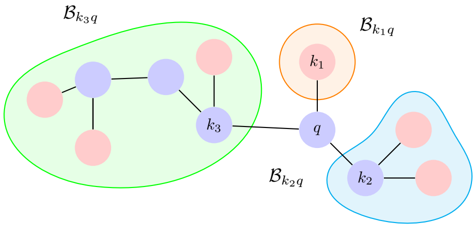


Fig. 2: In this example tree, the subtree $\mathcal{B}_{k_1q}$ is highlighted in orange, $\mathcal{B}_{k_2q}$ in blue and $\mathcal{B}_{k_3q}$ in green. Leaf nodes are colored red.

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

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

related covariance matrices of its respective subtree $\mathcal{B}_{kq}$ and sends them to node q. The updating node q therefore receives

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

$$\boldsymbol{R}_{\Sigma_{kq}}^i \triangleq \sum_{l \in \mathcal{B}_{kq}} \boldsymbol{R}_{\boldsymbol{z}_l}^i = \boldsymbol{R}_{\boldsymbol{z}_k}^i + \sum_{l \in \mathcal{N}_k \setminus q} \boldsymbol{R}_{\Sigma_{lk}}^i \quad \forall\, k \in \mathcal{N}_q \quad (33)$$

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

$$\overline{\boldsymbol{W}}_q \triangleq \left[\boldsymbol{W}_q^T \mid \boldsymbol{G}_{k_1}^T \mid \cdots \mid \boldsymbol{G}_{k_{n_q}}^T\right]^T \quad (34)$$

$$\overline{\boldsymbol{x}}_q^i \triangleq \left[\boldsymbol{x}_q^T \mid \boldsymbol{z}_{k_1q}^{iT} \mid \cdots \mid \boldsymbol{z}_{k_{n_q}q}^{iT}\right]^T \quad (35)$$

$$\boldsymbol{R}_{D_q}^i \triangleq \text{Blkdiag}(\boldsymbol{R}_{\boldsymbol{x}_q}, \boldsymbol{R}_{\Sigma_{k_1}q}^i, \ldots, \boldsymbol{R}_{\Sigma_{k_{n_q}}q}^i) \quad (36)$$

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 (23), the common components in $\boldsymbol{s}$ can be estimated as

$$\boldsymbol{s}^i = \frac{1}{K}\sum_k \boldsymbol{z}_k^i. \quad (37)$$

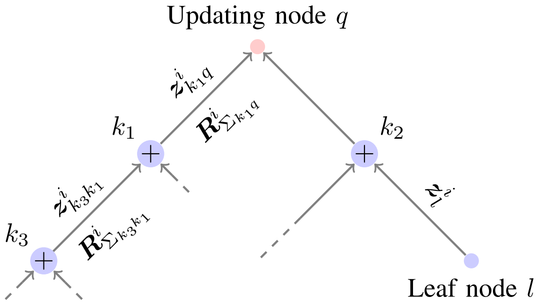

Updating node q
$\boldsymbol{z}_{k_1q}^i$
$\boldsymbol{R}_{\Sigma_{k_1q}}^i$
k_1
k_2
$\boldsymbol{z}_l^i$
$\boldsymbol{z}_{k_3k_1}^i$
$\boldsymbol{R}_{\Sigma_{k_3k_1}}^i$
k_3
Leaf node l

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 $\boldsymbol{W}_k^0$'s
    loop
        for $k \in \mathcal{N}_q$ do
            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 $\left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F$
            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 $k \in \mathcal{N}_q$ do
                Select $\boldsymbol{G}_k^{i+1}$ as the block of $\overline{\boldsymbol{W}}_q$ (see (34))
                  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 both 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 overlaying it with a topology management layer responsible for designing and maintaining a tree-topology at each iteration (assuming the initial graph is connected). Ideally, such a tree should preserve all the neighbors of the updating node q. Indeed, cutting off neighbors would reduce the number of update matrices $\boldsymbol{G}$ in (34), which would reduce the degrees of freedom in the minimization of (15), using the definitions in (34)-(36). Algorithm 4 describes a simple distributed procedure for constructing a tree rooted at node q and preserving the links to its neighbors. It has a per-node message complexity at node k of $\mathcal{O}(\mathcal{N}_k)$ and an overall time complexity of $\mathcal{O}(D)$ where D is

Algorithm 4: Distributed algorithm for computing a spanning tree preserving the neighbors of node q.

```
procedure buildtree($q$)
    for $l \in \mathcal{K}$ do
        Initialize set of children of node $l$: $\mathcal{L}_l \leftarrow \{\}$
        Initialize number of acknowledgment messages
          received: $r_l \leftarrow 0$
    At node $l \in \mathcal{N}_q$
        Set parent to node $q$: $p_l \leftarrow q$
    At node $l \neq q$
        while $p_l$ is not set do
            Wait for message $(\text{ADOPT}, m)$
            Set parent to node $m$: $p_l \leftarrow m$
            Send $(\text{OK}, l)$ to $p_l$ as an acknowledgment
        Send $(\text{ADOPT}, l)$ to $\mathcal{N}_l \setminus \{p_l\}$
        while $r_l < |\mathcal{N}_l|$ do
            Wait for message $(a, m)$
            if $a = ADOPT$ then
                Send $(\text{NOK}, \cdot)$ to node $m$ to indicate that it
                  already has a parent
            else
                if $a = OK$ then
                    Add $m$ to set of children of $l$:
                      $\mathcal{L}_l \leftarrow \mathcal{L}_l + \{m\}$
            $r_l \leftarrow r_l + 1$
```

the diameter of the network (i.e. longest shortest-path between two-nodes). The procedure is as follows. Each neighbor of node q sends a message to each of its neighbors offering to become its parent in the tree. A node accepts to become a child if it does not yet have a parent. Once it has a parent, it does the same with its own neighbors (its parent excluded). One can see that at each step of the algorithm, the nodes which have a parent are part of a tree. The algorithm stops when each node has a parent, producing a tree spanning the entire network graph.

In practice, the data exchange required to construct such a tree is expected to be negligible compared to the exchange of raw sensor data in Algorithm 3, in which a whole batch of T multi-channel signal samples is transmitted by each node. Furthermore, we expect the network topology to be relatively static and change only after many iterations of the algorithm, as the objective function of the algorithm would otherwise not be relevant anymore. In that scenario, it is reasonable to only periodically compute a new spanning tree using either Algorithm 4 or a more elaborate distributed spanning tree algorithm such as those described in [34].

A similar convergence statement as for fully-connected networks can be made for tree networks and their extension to arbitrary networks, and the associated proof can be relatively straightforwardly adapted to these cases, although the notation and definitions become 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	$\propto (M_q + Q(K-1))^3$	$\propto (M_q + Q\lvert\mathcal{N}_q\rvert)^3$

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

The above figures assume the transmission cost of the update matrices $\boldsymbol{G}$ and of the messages involved in the distributed spanning-tree procedure to be negligible compared to a batch of T samples.

V. Correlation Structure Estimation

In this section, we describe how the D-MAXVAR algorithm can be modified 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 by computing a different subset of the per-node MAXVAR directions, instead of the ones corresponding to the Q smallest GEVLs. Finally, we describe how to select an appropriate value for Q.

A. MAXVAR as a low-rank approximation

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

$$\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l} \triangleq \boldsymbol{R}_{\boldsymbol{x}_k}^{-\frac{1}{2}} \boldsymbol{R}_{\boldsymbol{x}_k\boldsymbol{x}_l} \boldsymbol{R}_{\boldsymbol{x}_l}^{-\frac{1}{2}} \tag{38}$$

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

$$\Theta_{kl} = \|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\|_F^2 \,, \tag{39}$$

where $\|\cdot\|_F$ denotes the Frobenius norm (See Appendix D for a proof). We first 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

$$[\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}}]_{kl} = \boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l} \tag{40}$$

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

$$\boldsymbol{U} \triangleq \boldsymbol{R}_D^{\frac{1}{2}} \boldsymbol{W} \boldsymbol{\Lambda}^{-\frac{1}{2}} , \tag{41}$$

allowing us to write (12) as

$$\boldsymbol{U} = \boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{U} \boldsymbol{\Lambda} \tag{42}$$

$$\boldsymbol{U}^T \boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{U} = \boldsymbol{\Lambda}^{-1} \tag{43}$$

which can be reorganized as

$$\boldsymbol{P}_{\boldsymbol{xx}}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1} \tag{44a}$$

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_Q. \tag{44b}$$

The columns of $\boldsymbol{U}$ are therefore orthogonal eigenvectors of $\boldsymbol{P}_{\boldsymbol{xx}}$ and $\boldsymbol{\Lambda}^{-1}$ contains the corresponding eigenvalues. As a result, the Q-dimensional filters computed by MAXVAR are isomorphic to the eigenvectors corresponding to the Q largest eigenvalues of $\boldsymbol{P}_{\boldsymbol{xx}}$ (i.e. the smallest diagonal elements of $\boldsymbol{\Lambda}$) via (41).

B. Naive TSC Approximation using D-MAXVAR

Considering the definition (39) of the TSC, it seems reasonable to approximate the TSC by replacing $\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}$ by $[\hat{\boldsymbol{P}}_{\boldsymbol{xx}}]_{kl}$ in (39). Indeed, if we denote the Q-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}}$ as $\hat{\boldsymbol{P}}_{\boldsymbol{xx}}$, we have

$$\hat{\boldsymbol{P}}_{\boldsymbol{xx}} = \underset{\boldsymbol{A}}{\operatorname{argmin}} \sum_{k,l\in\mathcal{K}} \|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l} - [\boldsymbol{A}]_{kl}\|_F^2 \tag{45}$$

$$\text{s.t.} \quad \operatorname{Rank}(\boldsymbol{A}) = Q. \tag{46}$$

$\hat{\boldsymbol{P}}_{\boldsymbol{xx}}$ can therefore be interpreted as a joint low-rank approximation of the matrices $\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}$. This approach, however, has two issues. Consider the split of (45) between diagonal and off-diagonal blocks:

$$\min_{\boldsymbol{A}} \sum_{k,l\in\mathcal{K},k\neq l} \|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l} - [\boldsymbol{A}]_{kl}\|_F^2 + \sum_{k\in\mathcal{K}} \|\boldsymbol{I}_{M_k} - [\boldsymbol{A}]_{kk}\|_F^2 \tag{47}$$

First, if the off-diagonal blocks of $\boldsymbol{P}_{\boldsymbol{xx}}$ have a very small norm, the joint low-rank approximation will be focused on reconstructing the diagonal blocks, which do not contain any information about the inter-node correlation structure. Second, $\boldsymbol{P}_{\boldsymbol{xx}}$ might have full rank, even though the inter-node correlation structure can be described by a small number of components (e.g. in the case of a completely uncorrelated network, $\boldsymbol{P}_{\boldsymbol{xx}} = \boldsymbol{I}$). This makes the choice of an appropriate value for Q quite challenging as it is seemingly unrelated to the rank of $\boldsymbol{P}_{\boldsymbol{xx}}$. Ideally, our approximation method should be such that increasing the value of Q, and therefore allocating more bandwidth, always results in a better approximation of the TSC. In addition, when the inter-node correlation structure can be described by only a few components, the TSC should be perfectly recovered by selecting $Q \ll M$.

C. TSC Approximation using a Modified D-MAXVAR

In subsection V-A, we have shown that MAXVAR was equivalent to computing the low-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}}$ and in the previous subsection, we have described how using this low-rank approximation to approximate the TSC would result in an approximation lacking multiple desirable properties. Via a slight modification of the original problem, we can obtain a low-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$, from which we can derive a TSC approximation which satisfies the desired properties for Q. For this purpose, we propose to approximate the TSC by using $\hat{\boldsymbol{P}}_{\boldsymbol{x}_k\boldsymbol{x}_l} \triangleq [\hat{\boldsymbol{P}}^D_{\boldsymbol{xx}}]_{kl}$ in (39), where $\hat{\boldsymbol{P}}^D_{\boldsymbol{xx}}$ denotes the low-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$ instead of $\boldsymbol{P}_{\boldsymbol{xx}}$. This would resolve the problems mentioned in the previous subsection, as it removes the influence of the (irrelevant) diagonal entries in (47). In the following, we describe how the D-MAXVAR algorithm can be modified to efficiently compute this new TSC approximation.

We first note that the eigenvectors of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$ and $\boldsymbol{P}_{\boldsymbol{xx}}$ are the same, and that the corresponding eigenvalues can be obtained by substracting 1 from the eigenvalues of $\boldsymbol{P}_{\boldsymbol{xx}}$. Indeed, the full eigenvalue decomposition of $\boldsymbol{P}_{\boldsymbol{xx}}$ can be expressed as

$$\boldsymbol{P}_{\boldsymbol{xx}} = \boldsymbol{U}_M\boldsymbol{\Lambda}_M^{-1}{\boldsymbol{U}_M}^T, \tag{48}$$

where $\boldsymbol{U}_M$ is an orthogonal matrix that contains the full set of M eigenvectors in its columns, and $\boldsymbol{\Lambda}_M^{-1}$ is the diagonal matrix of corresponding eigenvalues. The previously defined matrices $\boldsymbol{U}$ and $\boldsymbol{\Lambda}^{-1}$ are therefore Q-columns submatrices of $\boldsymbol{U}_M$ amd $\boldsymbol{\Lambda}_M^{-1}$, respectively. Substracting $\boldsymbol{I}$ from both sides yields

$$\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I} = \boldsymbol{U}_M(\boldsymbol{\Lambda}_M^{-1} - \boldsymbol{I})\boldsymbol{U}_M^T. \tag{49}$$

One can therefore obtain a Q-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$ by computing the Q eigenvectors of $\boldsymbol{P}_{\boldsymbol{xx}}$ corresponding to the Q largest diagonal elements of $\boldsymbol{\Lambda}_M^{-1} - \boldsymbol{I}$ *in absolute magnitude*[4]. We can link $\boldsymbol{U}_M$ to the full set of GEVCs $\boldsymbol{W}_M$ of $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{xx}})$ by noting that (41) is valid for any subset of eigenvectors. Therefore

$$\boldsymbol{W}_M = \boldsymbol{R}_D^{-\frac{1}{2}}\boldsymbol{U}_M\boldsymbol{\Lambda}_M^{\frac{1}{2}}, \tag{50}$$

and (49) in combination with the orthogonality of $\boldsymbol{U}_M$ yields

$$(\boldsymbol{R}_{\boldsymbol{xx}} - \boldsymbol{R}_D)\boldsymbol{W}_M = \boldsymbol{R}_D\boldsymbol{W}_M\boldsymbol{\Gamma}_M \tag{51a}$$

$$\boldsymbol{W}_M^T\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W}_M = \boldsymbol{I} \tag{51b}$$

with $\boldsymbol{\Gamma}_M \triangleq \boldsymbol{\Lambda}_M^{-1} - \boldsymbol{I}$. $\boldsymbol{W}_M$ thus contains GEVCs of the pencil $(\boldsymbol{R}_{\boldsymbol{xx}} - \boldsymbol{R}_D, \boldsymbol{R}_D)$ in its columns, and $\boldsymbol{\Gamma}_M$ contains the corresponding GEVLs. The problem of finding a Q-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$ is therefore equivalent to finding the Q GEVCs of $(\boldsymbol{R}_{\boldsymbol{xx}} - \boldsymbol{R}_D, \boldsymbol{R}_D)$ associated with its Q-largest (in absolute magnitude) GEVLs. These can be found based on a slightly modified version of the D-MAXVAR algorithm, described hereafter.

Modified D-MAXVAR Algorithm: Let $\boldsymbol{W}_Q$ denote the Q-columns submatrix of $\boldsymbol{W}_M$ associated with the Q-rank approximation of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$. In order to compute $\boldsymbol{W}_Q$ in a distributed fashion, Algorithm 3 must be modified such that, at each iteration i, $\overline{\boldsymbol{W}}^i$ contains the Q GEVCs of $(\boldsymbol{R}^i_{\overline{\boldsymbol{x}}_q} - \boldsymbol{R}^i_{D_q}, \boldsymbol{R}^i_{D_q})$ associated with the largest GEVLs (in absolute magnitude) instead of the Q smallest GEVLs of $(\boldsymbol{R}^i_{D_q}, \boldsymbol{R}^i_{\overline{\boldsymbol{x}}_q})$. This change has no impact on Theorems III.1 and III.2, which remain valid as $\boldsymbol{W}_Q$ is just another stationary point of the original problem (13). A similar result to Theorem III.3 can be obtained by showing that selecting the components as described above is equivalent to

[4] As $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$ is no longer positive semidefinite, the components corresponding to negative eigenvalues also contribute to the low-rank approximation error proportionally to the squared magnitude of their associated eigenvalue.

solving a related optimization problem for which Theorem III.3 applies, that is find the Q largest GEVCs of the pencil $(\boldsymbol{R}_{\boldsymbol{xx}} - \boldsymbol{R}_D, \boldsymbol{R}_D)$.

Remark V.1. We note that the original MAXVAR components in $\boldsymbol{W}$ generally differ from the components in $\boldsymbol{W}_Q$. Additional bandwidth would therefore need to be allocated if *both* the TSC approximation and the solutions to the classical MAXVAR problem would be required. If the goal is only to find an approximate TSC matrix (as in the example of Section VI), D-MAXVAR can be replaced with its modified version, in which case no additional bandwidth is required.

An expression for the approximation of the TSC using $\hat{\boldsymbol{P}}_{\boldsymbol{x}_k\boldsymbol{x}_l} \triangleq [\hat{\boldsymbol{P}}^D_{\boldsymbol{xx}}]_{kl}$ in (39) can be obtained as follows. Let $\boldsymbol{\Gamma}_Q$ be the diagonal submatrix of $\boldsymbol{\Gamma}_M$ corresponding to the GEVCs in $\boldsymbol{W}_Q$ obtained by the modified D-MAXVAR algorithm. We define the approximate TSC as

$$\hat{\Theta}_{kl} \triangleq \left\| [\hat{\boldsymbol{P}}^D_{\boldsymbol{xx}}]_{kl} \right\|_F^2 = \left\| \boldsymbol{U}_{Q,k} \boldsymbol{\Gamma}_Q \boldsymbol{U}_{Q,l}^T \right\|_F^2 \tag{52}$$

where $\boldsymbol{U}_{Q,k} = \boldsymbol{R}_{\boldsymbol{x}_k}^{\frac{1}{2}} \boldsymbol{W}_{Q,k} \boldsymbol{\Lambda}^{-\frac{1}{2}}$ (from (50)). From the definition of the Frobenius norm, we have

$$\left\| \boldsymbol{U}_{Q,k} \boldsymbol{\Gamma}_Q \boldsymbol{U}_{Q,l}^T \right\|_F^2 = \operatorname{Tr}\left(\boldsymbol{U}_{Q,k} \boldsymbol{\Gamma}_Q \boldsymbol{U}_{Q,l}^T \boldsymbol{U}_{Q,l} \boldsymbol{\Gamma}_Q \boldsymbol{U}_{Q,k}^T \right) \tag{53}$$

and from the cyclic property of the trace

$$\left\| \boldsymbol{U}_{Q,k} \boldsymbol{\Gamma}_Q \boldsymbol{U}_{Q,l}^T \right\|_F^2 = \operatorname{Tr}\left(\boldsymbol{U}_{Q,k}^T \boldsymbol{U}_{Q,k} \boldsymbol{\Gamma}_Q \boldsymbol{U}_{Q,l}^T \boldsymbol{U}_{Q,l} \boldsymbol{\Gamma}_Q \right). \tag{54}$$

From (50) and the fact that $\boldsymbol{\Lambda}_M^{\frac{1}{2}} = (\boldsymbol{\Gamma}_M + \boldsymbol{I})^{-\frac{1}{2}}$ (from the definition of $\boldsymbol{\Gamma}_M$), we find that

$$\boldsymbol{U}_{Q,k}^T \boldsymbol{U}_{Q,k} = (\boldsymbol{\Gamma}_Q + \boldsymbol{I}_Q)^{\frac{1}{2}} \boldsymbol{W}_{Q,k}^T \boldsymbol{R}_{\boldsymbol{x}_k} \boldsymbol{W}_{Q,k} (\boldsymbol{\Gamma}_Q + \boldsymbol{I}_Q)^{\frac{1}{2}} \tag{55}$$

$$= (\boldsymbol{\Gamma}_Q + \boldsymbol{I}_Q)^{\frac{1}{2}} \boldsymbol{R}^Q_{\boldsymbol{z}_k} (\boldsymbol{\Gamma}_Q + \boldsymbol{I}_Q)^{\frac{1}{2}} \tag{56}$$

where $\boldsymbol{R}^Q_{\boldsymbol{z}_k} \triangleq \boldsymbol{W}_{Q,k}^T \boldsymbol{R}_{\boldsymbol{x}_k} \boldsymbol{W}_{Q,k}$ is the covariance matrix of the compressed signal $\boldsymbol{z}_k$ associated with $\boldsymbol{W}_{Q,k}$, such that we finally obtain

$$\hat{\Theta}_{kl} = \operatorname{Tr}\left(\boldsymbol{R}^Q_{\boldsymbol{z}_k} (\boldsymbol{\Gamma}_Q^2 + \boldsymbol{\Gamma}_Q) \boldsymbol{R}^Q_{\boldsymbol{z}_l} (\boldsymbol{\Gamma}_Q^2 + \boldsymbol{\Gamma}_Q) \right). \tag{57}$$

Note that $\boldsymbol{R}^Q_{\boldsymbol{z}_k}$ can be locally computed at node k and 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. Finally, we have from (51) that $\boldsymbol{\Gamma}_Q$ can be computed as

$$\boldsymbol{\Gamma}_Q = \boldsymbol{\Lambda}_Q^{-1} - \boldsymbol{I}_Q = \boldsymbol{W}_Q^T \boldsymbol{R}_D \boldsymbol{W}_Q = \left(\sum_k \boldsymbol{R}^Q_{\boldsymbol{z}_k} \right)^{-1} - \boldsymbol{I}_Q, \tag{58}$$

which can be efficiently estimated at each node using (33) without any additional communication.

D. Selecting Q

The following theorem provides a useful bound on the TSC total approximation error:

Theorem V.1. *Let γ denote the Q-largest element of $\boldsymbol{\Gamma}$ in absolute magnitude. Then the total TSC approximation error*

$$\sum_{k,l \in \mathcal{K}, k \neq l} \left(\sqrt{\hat{\Theta}^Q_{kl}} - \sqrt{\Theta_{kl}} \right)^2 \leq (M - Q)\gamma^2. \tag{59}$$

Proof. See Appendix E. □

Any node can compute the above bound to assess the quality of the TSC approximation, as γ also corresponds to the smallest diagonal element of $\boldsymbol{\Gamma}_Q$, locally computable via (58).

Note that if $\operatorname{Rank}(\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I})$ is known a priori, selecting $Q = \operatorname{Rank}(\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I})$ would result in a perfect recovery of $\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}$, and hence of the TSC (by definition (39)). However, the rank is in practice not known in advance, and the node must therefore estimate it by increasing Q until $\gamma = 0$, as then $\operatorname{Rank}(\boldsymbol{P}_{\boldsymbol{xx}} - \boldsymbol{I}) = Q - 1$.

Remark V.2. One can also take a statistical approach and test whether the absolute magnitudes of the eigenvalues are significantly larger than 0, i.e., whether they truly capture correlated components or whether their energy is explained by the estimation noise. The latter can be tested, e.g., via a maximum likelihood ratio test as in [23]–[25], [35], [36].

VI. Simulation Results

In this section, we validate the TSC approximation 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.

$$\boldsymbol{x}_k = \boldsymbol{A}_k \boldsymbol{s} + \alpha \boldsymbol{n}_k \tag{60}$$

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

$$\boldsymbol{a}_k^j = \frac{\boldsymbol{g}_k^j}{\max\{0.1, \|\boldsymbol{m}_k - \boldsymbol{l}_j\|^2\}} \tag{61}$$

where $\boldsymbol{g}_k^j$ is an M_k-dimensional vector of random variables drawn uniformly from $[0.95, 1.05]$, modeling 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 matrix can be computed as

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{A}^H + \alpha^2 \boldsymbol{I} \tag{62}$$

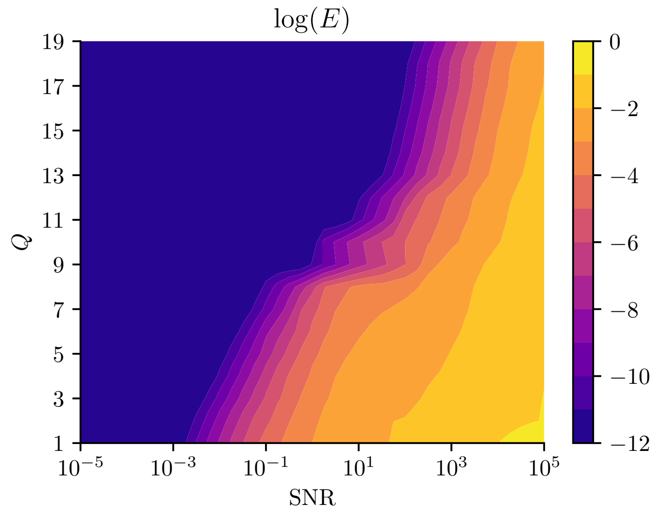


Fig. 4: Monte-Carlo simulations of the TSC approximation error for varying Q and SNR values. The color scale corresponds to the log of the mean value obtained for E after 1000 Monte-Carlo runs. The log-error values are clipped to -12.

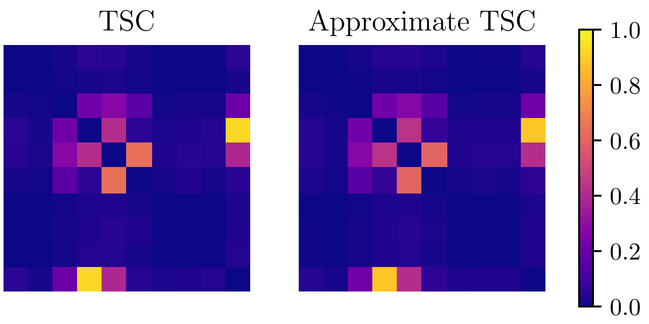


Fig. 5: Example TSC matrices obtained with $K = 10$, $M_k = 8$, $Q = 3$, 3 latent sources and SNR = 1. Each pixel represents the (approximate) TSC of a pair of node. Diagonal values are not shown.

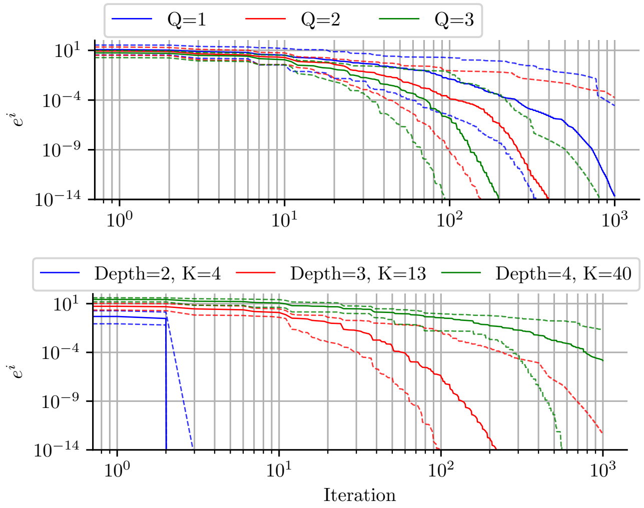


Fig. 6: Monte-Carlo simulations of the convergence of the D-MAXVAR algorithm in tree-topology networks. Solid curves depict the mean values. Shaded areas depict the 5-95% percentile regions. The top plot was generated with $K = 13$. The bottom plots were generated with $K = 4, 13, 40$.

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

For each simulated scenario, we performed 1000 Monte-Carlo runs where the covariance matrices were directly computed via (62). 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

$$E \triangleq \frac{1}{K(K-1)} \sum_{k,l \neq k} |\hat{\Theta}_{kl} - \Theta_{kl}| \tag{63}$$

As expected, the quality of the approximation increases with increasing Q and always eventually reaches 0 for a sufficiently high value of Q. As the SNR grows, so does the value of Q required to obtain a perfect approximation of the TSC. Indeed, a larger SNR causes the eigenvalues of $\boldsymbol{R_{xx}}$, and hence $\boldsymbol{P_{xx}}$, to grow closer to 0. As a consequence, the corresponding eigenvalues of $\boldsymbol{P_{xx}} - \boldsymbol{I}$ become closer to -1, resulting in more power in the off-diagonal blocks which needs to be accounted for. In practice, using our method in such a setting where the signal subspace is perfectly observed by every node does not make much sense as a simple distributed PCA (based for example on [37]) would yield an excellent approximation of $\boldsymbol{R_{xx}}$ with only $Q = d$ components. On the other hand, such a PCA-based method would perform very poorly in low-SNR regimes, as it would first compute the components of highest power, which would in that case correspond to noise and would therefore not yield any information about the correlation structure. Our method is therefore best suited for scenarios where the node-specific subspaces and common subspaces have similar power levels and are hard to separate. Figure 5 showcases such a scenario. We see that the correlation structure is well captured by the approximate TSC using only 3 components when node-specific noise and latent signals have similar power levels.

C. Convergence

Figure 6 shows the convergence of the D-MAXVAR algorithm to the global solution. The metric used is

$$e^i = 1 - \frac{f^*}{f(\boldsymbol{W}^i)} \tag{64}$$

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 expect convergence speed to increase in more densely connected networks, that is networks with more links, as this increases the number of neighbors of each node and hence the number of $\boldsymbol{G}$ matrices possibly used at each iteration (that is assuming that an instance of Algorithm 4 is run at each iteration). In tree-topology networks, the total number of links is $K-1$, while it is K^2 in fully connected networks, the convergence speed is therefore expected to drop much more rapidly with increasing network size K in the case of tree-topology networks as the compensation in terms of added neighbors per node is much less significant than for fully connected networks.

Finally, 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 holds 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. Even though the proposed approximation can be arbitrarily precise, it is not adapted to very high-SNR regimes, as the number of required components can quickly grow to become larger than the number of channels at a given node. Nevertheless, we have demonstrated that the approximation is quite accurate for lower SNR regimes, even with low Q, 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

$$\begin{aligned}L_q(\boldsymbol{W},\boldsymbol{\Lambda}^q,\boldsymbol{\Gamma}_k)&=\operatorname{Tr}\left(\boldsymbol{W}^T\boldsymbol{R}_D\boldsymbol{W}\right)\\&\quad-\operatorname{Tr}\left(\boldsymbol{\Lambda}^q(\boldsymbol{W}^T\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W}-\boldsymbol{I}^Q)\right)\\&\quad-\sum_{k\neq q}\operatorname{Tr}\left(\boldsymbol{\Gamma}_k\boldsymbol{W}_k^T\boldsymbol{N}_k\right)\end{aligned}\tag{65}$$

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

$$\frac{\partial}{\partial\boldsymbol{W}}L_q=0=2\boldsymbol{R}_D\boldsymbol{W}^*-2\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W}^*\boldsymbol{\Lambda}^q-\boldsymbol{N}^q\boldsymbol{\Gamma}^q\tag{66}$$

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

$$\boldsymbol{W}^{*T}\boldsymbol{R}_D\boldsymbol{W}^*=\boldsymbol{\Lambda}^q-\frac{1}{2}\underbrace{\boldsymbol{W}^{*T}\boldsymbol{N}^q}_{0}\boldsymbol{\Gamma}^q\quad\forall q.\tag{67}$$

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 (66), we have

$$\frac{\partial}{\partial\boldsymbol{W}_q}L_q=0=2\boldsymbol{R}_{\boldsymbol{x}_q}\boldsymbol{W}_q^*-2\boldsymbol{R}_{\boldsymbol{x}_q\boldsymbol{x}}\boldsymbol{W}^*\boldsymbol{\Lambda}-\underbrace{\boldsymbol{N}_q\boldsymbol{\Gamma}_q}_{0}\quad\forall q\tag{68}$$

Combining those equations for $q\in\mathcal{K}$ yields $\boldsymbol{R}_D\boldsymbol{W}^*=\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W}^*\boldsymbol{\Lambda}$. As $\boldsymbol{W}^*$ is computed via the GEVD of the local pencil $(\boldsymbol{R}_{D_q}^i,\boldsymbol{R}_{\widetilde{\boldsymbol{x}}_q}^i)$ 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,

$$\overline{\boldsymbol{W}}^T\boldsymbol{R}_{D_q}^i\overline{\boldsymbol{W}}=\boldsymbol{W}^{*T}\boldsymbol{R}_D\boldsymbol{W}^*=\boldsymbol{\Lambda}\tag{69}$$

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

□

B. Proof of Theorem III.2

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

$$\mathcal{D}=\{\boldsymbol{W}\in\mathbb{R}^{M\times Q}\mid\boldsymbol{W}^T\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{W}=\boldsymbol{I}_Q\}.\tag{70}$$

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

$$\mathcal{D}_q(\boldsymbol{V})=\{\boldsymbol{W}\in\mathcal{D}\mid\mathcal{C}(\boldsymbol{W}_k)\subseteq\mathcal{C}(\boldsymbol{V}_k)\quad\forall k\neq q\}\tag{71}$$

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

$$\boldsymbol{W}^{i+1}\in\underset{\boldsymbol{W}\in\mathcal{M}^i}{\operatorname{argmin}}\left\|\boldsymbol{W}-\boldsymbol{W}^i\right\|_F\tag{72a}$$

$$\mathcal{M}^i=\underset{\boldsymbol{W}\in\mathcal{D}_{q_i}(\boldsymbol{W}^i)}{\operatorname{argmin}}\ f(\boldsymbol{W})\tag{72b}$$

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.

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* (72).

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

$$\mathcal{D}_q(\boldsymbol{W}) = \{P^q_{\boldsymbol{W}}\boldsymbol{V} \mid \boldsymbol{V} \in \mathbb{R}^{M\times Q}\} \cap \mathcal{D} \tag{73}$$

where $P^q_{\boldsymbol{W}}$ denotes the orthogonal projection matrix on the linear subspace (71). $m_q(\boldsymbol{W})$ therefore corresponds to the sum of the Q smallest GEVLs of $(P^q_{\boldsymbol{W}}\boldsymbol{R}_D P^{qT}_{\boldsymbol{W}}, P^q_{\boldsymbol{W}}\boldsymbol{R}_{\boldsymbol{xx}}P^{qT}_{\boldsymbol{W}})$ (which is found by substituting the optimization variable in (15) by $P^q_{\boldsymbol{W}}\boldsymbol{V}$). As the GEVLs of a pencil vary continuously with the entries of its constituent matrices [38], and as $P^q_{\boldsymbol{W}}$ varies continuously with $\boldsymbol{W}$ at points where its blocks $\boldsymbol{W}_{k\neq q}$ have linearly independent columns [39] (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. □

Let $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ be any sequence of iterates satisfying the mapping defined by Algorithm 1 and therefore procedure (72). We now show that if $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ has an accumulation point, then it is a fixed point of Algorithm 1 and therefore a stationary point of problem (13).

Lemma B.2. *The accumulation points of $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ are fixed points of Algorithm 1.*

Proof. Let $\boldsymbol{W}^*$ be an accumulation point 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:

$$\lim_{\boldsymbol{W}\to\boldsymbol{W}^*} m_k(\boldsymbol{W}) = m_k(\boldsymbol{W}^*) \quad \forall k \in \mathcal{K}. \tag{74}$$

As $(f(\boldsymbol{W}^i))_{i\in\mathbb{N}}$ 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

$$m_{q_i}(\boldsymbol{W}^i) = f(\boldsymbol{W}^{i+1}) \tag{75}$$

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

$$\lim_{i\to\infty} m_{q_i}(\boldsymbol{W}^i) = f^*. \tag{76}$$

As $\boldsymbol{W}^*$ is an accumulation point of $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$, there is some index set $\mathcal{N}_k \subseteq \mathbb{N}$ such that $(\boldsymbol{W}^i)_{i\in\mathcal{N}_k}$ converges to $\boldsymbol{W}^*$ and $(q_i)_{i\in\mathcal{N}_k} = (k)_i$ for some node k. From (76) and the continuity of m_q we have

$$m_k(\boldsymbol{W}^*) = f^* = f(\boldsymbol{W}^*) \tag{77}$$

As, by definition (71), $\boldsymbol{W}^*$ 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 (72b)). In virtue of (72a), it must be that $F_k(\boldsymbol{W}^*) = \boldsymbol{W}^*$ and $\boldsymbol{W}^*$ is therefore a fixed point of F_k.

We will now establish that $\boldsymbol{W}^*$ is also a fixed point of node $k+1$. As, by hypothesis, the Q-th and $(Q+1)$-th smallest GEVLs of the local pencils $(\boldsymbol{R}^i_{D_k}, \boldsymbol{R}^i_{\overline{\boldsymbol{x}}_k})$ are distinct at the accumulations points of $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$, a small perturbation of the pencil around an accumulation point results in a small perturbation of the generalized eigenspace [9], [38][5]. As a consequence, the convergence of the subsequence $(\boldsymbol{W}^i)_{i\in\mathcal{N}_k}$ to $\boldsymbol{W}^*$ implies the convergence of $(\mathcal{M}^i)_{i\in\mathcal{N}_k}$ to some $\mathcal{M}^*$, which from (72) corresponds to the sequence of sets of generalized eigenvectors of the local pencils $(\boldsymbol{R}^i_{\overline{\boldsymbol{x}}_q}, \boldsymbol{R}^i_{D_q})$, where the convergence of $(\mathcal{M}^i)_{i\in\mathcal{N}_k}$ must be understood in terms of the Haussdorf distance between sets [6].

As $m_k(\boldsymbol{W}^*) = f^*$ and $f(\boldsymbol{W}^*) = f^*$, it must be that $\boldsymbol{W}^* \in \mathcal{M}^*$. Therefore, there exists some convergent sequence $\{\boldsymbol{V}^{i+1}\}_{i\in\mathcal{N}_k}$ with $\boldsymbol{V}^{i+1} \in \mathcal{M}^i$ converging to $\boldsymbol{W}^*$ (As the convergence of $(\mathcal{M}^i)_{i\in\mathcal{N}_k}$ to $\mathcal{M}^*$ implies that for any point $\boldsymbol{W}$ in $\mathcal{M}^*$ we can find a set $\mathcal{M}^i \in (\mathcal{M}^i)_{i\in\mathcal{N}_k}$ containing a point arbitrarily close to $\boldsymbol{W}$). As both sequences converge to the same point,

$$\lim_{i\to\infty, i\in\mathcal{N}_k} \left\|\boldsymbol{W}^i - \boldsymbol{V}^{i+1}\right\|_F = 0, \tag{78}$$

and as $\boldsymbol{V}^{i+1} \in \mathcal{M}^i$, we have from (72a) that

$$\begin{aligned}&\left\|\boldsymbol{V}^{i+1} - \boldsymbol{W}^i\right\|_F \geq \\ &\qquad \min_{\boldsymbol{W}\in\mathcal{M}^i} \left\|\boldsymbol{W} - \boldsymbol{W}^i\right\|_F = \left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F.\end{aligned} \tag{79}$$

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

$$\lim_{i\to\infty, i\in\mathcal{N}_k} \left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F = 0. \tag{80}$$

The convergence of $(\boldsymbol{W}^i)_{i\in\mathcal{N}_k}$ to $\boldsymbol{W}^*$ therefore implies the convergence of $(\boldsymbol{W}^{i+1})_{i\in\mathcal{N}_k}$ to the same point. As $(\boldsymbol{W}^{i+1})_{i\in\mathcal{N}_k}$ =$(\boldsymbol{W}^i)_{i\in\mathcal{N}_k+1}$ with $(q_i)_{i\in\mathcal{N}_k+1} = (k+1)_i$, the argument showing that $\boldsymbol{W}^*$ is a fixed point of node k can be applied to node $k+1$, and inductively to node $k+l$ for any l. $\boldsymbol{W}^*$ is therefore a fixed point of Algorithm 1. □

We can now finally prove Theorem III.2. As $\mathcal{D}$ is compact, $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ must, by definition of an accumulation point, converge to the (possibly infinite) set of its accumulation points. As a consequence, if $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ is not a convergent sequence, it will eventually oscillate between points arbitrarily close to accumulation points. As from Lemma B.2, the accumulation points of $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ are fixed points of the algorithm, and hence from Theorem III.1, stationary points of the problem, the sequence has a finite number of accumulation points, separated by a finite and fixed distance. As a consequence, it cannot be that $\left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F$ converges to 0 and there must be some index set $\mathcal{N}$ such that

$$\lim_{i\to\infty, i\in\mathcal{N}} \left\|\boldsymbol{W}^{i+1} - \boldsymbol{W}^i\right\|_F > 0. \tag{81}$$

Therefore there must also be some other index set $\mathcal{N}_k \subseteq \mathcal{N}$ such that $(q_i)_{i\in\mathcal{N}_k} = (k)_i$. (81) thus contradicts (80) and $(\boldsymbol{W}^i)_{i\in\mathbb{N}}$ must therefore be a convergent sequence.

□

[5] 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 (41) in order to turn the GEVD into an equivalent eigenvalue decomposition.

[6] 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.

C. Proof of Theorem III.3

Since $\boldsymbol{W}^*$ is not a global minimizer and since (13) has no local minima[7], 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. □

D. Proof of Equation (39)

Let $\boldsymbol{U}_{kl} \triangleq \boldsymbol{R}_{\boldsymbol{x}_k}^{\frac{1}{2}} \boldsymbol{W}_{kl}$. (4) becomes

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l} \\ \boldsymbol{P}_{\boldsymbol{x}_l\boldsymbol{x}_k} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{kl} \\ \boldsymbol{U}_{lk} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{kl} \\ \boldsymbol{U}_{lk} \end{bmatrix} \boldsymbol{\Lambda}_{kl} \tag{82}$$

As described in [27], the full eigenvalue decomposition of this matrix consists in the eigenvectors and diagonal matrix

$$\begin{bmatrix} \boldsymbol{U}_{kl} & \boldsymbol{U}_{kl} \\ \boldsymbol{U}_{lk} & -\boldsymbol{U}_{lk} \end{bmatrix} \text{ and } \begin{bmatrix} \boldsymbol{\Lambda}_{kl} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{\Lambda}_{kl} \end{bmatrix} \tag{83}$$

As the eigenvectors matrix is unitary, we have

$$2\left\|\boldsymbol{\Lambda}_{kl}\right\|_F^2 = 2\text{Tr}\left(\boldsymbol{\Lambda}_{kl}^2\right) = \\ \left\|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F^2 + \left\|\boldsymbol{P}_{\boldsymbol{x}_l\boldsymbol{x}_k}\right\|_F^2 = 2\left\|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F^2 \tag{84}$$

which in conjunction with (5) gives $\left\|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F^2 = \Theta_{kl}$. □

E. Proof of Theorem V.1

From the Cauchy-Schwartz inequality, we have

$$\left\|\hat{\boldsymbol{P}}_{\boldsymbol{x}_k\boldsymbol{x}_l} - \boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F^2 \geq \tag{85}$$

$$\left\|\hat{\boldsymbol{P}}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F^2 + \left\|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F^2 - 2\left\|\hat{\boldsymbol{P}}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F \left\|\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{x}_l}\right\|_F \tag{86}$$

$$\geq \Theta_{kl} + \hat{\Theta}_{kl}^Q - 2\sqrt{\hat{\Theta}_{kl}^Q}\sqrt{\Theta_{kl}} \tag{87}$$

$$\geq \left(\sqrt{\hat{\Theta}_{kl}^Q} - \sqrt{\Theta_{kl}}\right)^2 \tag{88}$$

As γ^2 corresponds to the Q-largest squared eigenvalue of $\boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}} - \boldsymbol{I}$, the low-rank approximation error $\left\|\hat{\boldsymbol{P}}_{\boldsymbol{x}\boldsymbol{x}} - \boldsymbol{P}_{\boldsymbol{x}\boldsymbol{x}}\right\|_F^2$ is bounded by $(M-Q)\gamma^2$ and therefore so is

$$\sum_{k,l\in\mathcal{K},k\neq l} \left(\sqrt{\hat{\Theta}_{kl}^Q} - \sqrt{\Theta_{kl}}\right)^2.$$

□

[7] Matrices spanning the generalized eigenspaces of $(\boldsymbol{R}_D, \boldsymbol{R}_{\boldsymbol{x}\boldsymbol{x}})$ 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.

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] A. Bertrand, "Distributed signal processing for wireless eeg sensor networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 923–935, 2015.

[4] S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2010, pp. 1–5.

[5] G. Ing and M. J. Coates, "Parallel particle filters for tracking in wireless sensor networks," in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, 2005*. IEEE, 2005, pp. 935–939.

[6] 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.

[7] 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.

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

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

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

[11] 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.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] ——, "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.

[19] 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.

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

[21] 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.

[22] 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.

[23] P. Stoica, K. M. Wong, and Q. Wu, "On a nonparametric detection method for array signal processing in correlated noise fields," *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 1030–1032, 1996.

[24] Y. Song, P. J. Schreier, D. Ramírez, and T. Hasija, “Canonical correlation analysis of high-dimensional data with very small sample support,” *Signal Processing*, vol. 128, pp. 449–458, 2016.

[25] N. Asendorf and R. R. Nadakuditi, “Improved detection of correlated signals in low-rank-plus-noise type data sets using informative canonical correlation analysis (icca),” *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3451–3467, 2017.

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

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

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

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] G. Pandurangan, P. Robinson, M. Scquizzato *et al.*, “The distributed minimum spanning tree problem,” *Bulletin of EATCS*, vol. 2, no. 125, 2018.

[35] M. S. Bartlett, “The statistical significance of canonical correlations,” *Biometrika*, vol. 32, no. 1, pp. 29–37, 1941.

[36] Z. Bao, J. Hu, G. Pan, and W. Zhou, “Canonical correlation coefficients of high-dimensional normal vectors: finite rank case,” *arXiv preprint arXiv:1407.7194*, 2014.

[37] 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.

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

[39] 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.