Trading off Complexity With Communication Costs in Distributed Adaptive Learning via Krylov Subspaces for Dimensionality Reduction

Symeon Chouvardas, *Student Member, IEEE*, Konstantinos Slavakis, *Senior Member, IEEE*, and Sergios Theodoridis, *Fellow, IEEE*

***Abstract*—In this paper, the problem of dimensionality reduction in adaptive distributed learning is studied. We consider a network obeying the ad-hoc topology, in which the nodes sense an amount of data and cooperate with each other, by exchanging information, in order to estimate an unknown, common, parameter vector. The algorithm, to be presented here, follows the set-theoretic estimation rationale; i.e., at each time instant and at each node of the network, a closed convex set is constructed based on the received measurements, and this defines the region in which the solution is searched for. In this paper, these closed convex sets, known as property sets, take the form of hyperslabs. Moreover, in order to reduce the number of transmitted coefficients, which is dictated by the dimension of the unknown vector, we seek for possible solutions in a subspace of lower dimension; the technique will be developed around the Krylov subspace rationale. Our goal is to find a point that belongs to the intersection of this infinite number of hyperslabs and the respective Krylov subspaces. This is achieved via a sequence of projections onto the property sets and the Krylov subspaces. The case of highly correlated inputs that degrades the performance of the algorithm is also considered. This is overcome via a transformation which whitens the input. The proposed schemes are brought in a decentralized form by adopting the combine-adapt cooperation strategy among the nodes. Full convergence analysis is carried out and numerical tests verify the validity of the proposed schemes in different scenarios in the context of the adaptive distributed system identification task.**

***Index Terms*—Adaptive distributed learning, diffusion, Krylov subspaces, Projections.**

I. Introduction

WIRELESS sensor networks (WSNs) comprise of a number of nodes, which sense an amount of data and cooperate with each other in order to estimate an unknown and common parameter vector. Sensor networks have attracted a considerable interest over the recent years due to a plethora of applications in which they contribute. Some typical examples are: environmental monitoring, acoustic source localization and life sciences, just to name a few, e.g., [1]–[3]. A first approach to the problem of estimating the unknown vector is the centralized one. In such a scenario, the sensors transmit the sensed information to a central node, also known as fusion center, and this one carries out the full amount of computations. The existence of a fusion center is not always feasible due to power and/or position constraints. Moreover, such a philosophy lacks robustness, since if the fusion center fails, then the network collapses. On the contrary, by following a fully decentralized philosophy, the previously mentioned limitations can be overstepped. Depending on the way with which the sensors are deployed over the field, the following topologies are defined.

- The incremental, in which each node is able to communicate with only one neighboring node and the nodes lie in a cyclic pattern, e.g., [4], [5]. Despite the fact that this topology requires small bandwidth, a Hamiltonian Cycle has to be constructed and maintained, which is an NP hard task, e.g., [6], [7]. Furthermore, if one node is malfunctioning, the network collapses.
- The diffusion, where each node transmits information to a subset of the node set. This subset is also known as the neighborhood of a node. The diffusion ad-hoc topology requires larger bandwidth, compared to the incremental one. Nevertheless, its implementation turns out to be easier when large networks are involved, and it is robust against node failures [8]–[12].

In this paper, we study the problem of *adaptive* distributed learning, where the estimate of the unknown vector is updated dynamically based on measurements that become available to each one of the nodes sequentially, one per time instant; these updates are then diffused throughout the network. It has been verified that if the nodes cooperate with each other and embed in their local update mechanisms the estimates received from their neighborhood, then the overall performance of the algorithms is enhanced compared to the case where each node operates individually [9].

Obviously, this cooperation demands that at every time instant each node will transmit a number of coefficients, which equals to the dimension of the vector to be estimated. In applications where this dimension is large, the exchange of information among the nodes can be a burden. In the current study, in order to achieve *dimensionality reduction* and consequently to

Manuscript received August 07, 2012; revised December 01, 2012; accepted January 26, 2013. Date of publication February 21, 2013; date of current version March 09, 2013. This work was supported in part by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ali Sayed.

S. Chouvardas and S. Theodoridis are with the Department of Informatics and Telecommunications, University of Athens, 157 84 Athens, Greece (e-mail: schouv@di.uoa.gr; stheodor@di.uoa.gr).

K. Slavakis is with the Department of Telecommunications Science and Technology, University of Peloponnese, Tripolis 22100, Greece (e-mail: slavakis@uop.gr).

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

Digital Object Identifier 10.1109/JSTSP.2013.2246762

reduce the number of transmitted coefficients, the reduced rank adaptive filtering rationale is adopted. Algorithms whose goal is to reduce the amount of transmitted information, by performing dimensionality reduction, have been proposed in the context of distributed quantized Kalman Filtering [13], [14], and quantized consensus algorithms, e.g., [15]. However, to the best of our knowledge, this is the first time that a reduced rank algorithm able for adaptive operation in diffusion networks is developed. The basic concept of our reduced rank adaptive filtering task can be summarized as follows: instead of seeking for the unknown vector in the original space, one seeks for the projection of it onto a lower dimension subspace. Via this procedure, the obtained estimates are optimally forced in a lower dimension space, and each node transmits fewer coefficients than the ones originally needed, in the case where the full dimensionality of the unknown vector was exploited. Here, the associated subspaces are the so-called Krylov subspaces, constructed by exploiting the statistics of the sensed information. The Krylov subspaces have been used in several applications, as for example in the reduced rank adaptive filtering [16], [17], in the Multistage Nested Wiener Filter [18], in the auxiliary vector filtering, [19], etc. It has been verified, e.g., [20], that the performance of the algorithms, which employ the Krylov subspace rationale, depends highly on the statistics of the input. More specifically, if the input signal is highly correlated, then the performance of the algorithms is degraded. In order to overstep this problem we propose a whitening technique, which is based on the Discrete Cosine Transformation (DCT) and it has been employed in the context of non-distributed adaptive learning [20], [21]. This strategy is properly reformed, in order to be suitable for operation in distributive learning.

The reported algorithms, follow the set-theoretic estimation rationale [22]; i.e., instead of seeking for a unique optimum vector, that minimizes a certain cost function, we search for a set of points that are *in agreement* with the received set of measurements. More specifically, we seek for solutions within the intersection of the Krylov subspaces and the property sets, namely hyperslabs, formed by the received measurements. We assume that *any* point that lies within this set is in agreement with the current measurements. The goal becomes that of finding a point that lies in the intersection of this *infinite* number of hyperslabs, which are constructed sequentially one per time instant, with the respective Krylov subspaces. This can be achieved via a sequence of corresponding projections, as dictated by the set-theoretic estimation, e.g., [23]–[26]. Furthermore, the algorithmic scheme is brought to a distributed fashion, by adopting a cooperation strategy among the nodes. Summarizing, the main contributions of the paper are the following:

- A novel reduced rank adaptive algorithm, which achieves dimensionality reduction, suitable for operating in networks operating under the diffusion ad-hoc topology, is developed for the first time. The algorithm is built around the Krylov subspace rationale.
- The case where the input is highly correlated, which leads to performance degradation of the Krylov based algorithms, is separately considered. To this end, a modification of the algorithm is derived by employing a whitening transformation (see [20], [21]).

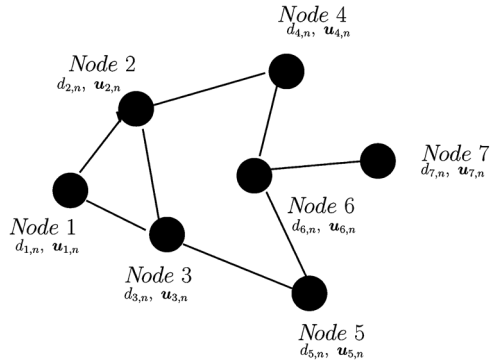


Fig. 1. Illustration of a diffusion network with $K = 7$ nodes.

The paper is organized as follows. In Section II, the problem formulation is described, and in Section III, the set-theoretic rationale is given. In Section IV, we shed light on basic concepts regarding the adaptive distributed learning, and in the Section V, the algorithm, as well as its theoretical analysis are provided. The algorithmic scheme, which is appropriate for highly correlated environments, is described in Section VI. Finally, in Section VII the performance of the proposed algorithmic schemes is validated and in the Appendices the theoretical background is discussed, and full proofs of the theorems are given.

Notation: The set of all non-negative integers and the set of all real numbers will be denoted by $\mathbb{Z}_{\geq 0}$ and $\mathbb{R}$ respectively. Given two integers j_1, j_2, with $j_1 \leq j_2$, we define $\overline{j_1, j_2} = \{j_1, j_1 + 1, \ldots, j_2\}$. Vectors will be denoted by boldface letters and matrices will be denoted by uppercase boldface letters. Moreover, $\|\cdot\|$ will stand for the Euclidean norm, whereas $\langle \cdot, \cdot \rangle_{\boldsymbol{W}}$ and $\|\cdot\|_{\boldsymbol{W}}$ will stand for the weighted inner product and the weighted norm respectively, with definition $\langle \boldsymbol{w}_1, \boldsymbol{w}_2 \rangle_{\boldsymbol{W}} = \boldsymbol{w}_1^T \boldsymbol{W} \boldsymbol{w}_2, \forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^m$ and $\|\boldsymbol{w}\|_{\boldsymbol{W}} = \sqrt{\boldsymbol{w}^T \boldsymbol{W} \boldsymbol{w}}, \forall \boldsymbol{w} \in \mathbb{R}^m$, where the $m \times m$ matrix $\boldsymbol{W}$ is positive definite. $\mathbb{E}[\cdot]$ stands for the expectation operator. The 2-norm of a matrix, say $\boldsymbol{A}$, will be denoted by $\|\boldsymbol{A}\|$. Finally, given a set $\mathcal{S}$, $|\mathcal{S}|$ will stand for its cardinality.

II. Problem Statement

Following the philosophy of a diffusion network, we consider a network consisting of K spatially distributed nodes. Our task is to estimate an unknown parameter vector of interest, $\boldsymbol{w}_* \in \mathbb{R}^m$, through measurements $(d_{k,n}, \boldsymbol{u}_{k,n}) \in \mathbb{R} \times \mathbb{R}^m$, which are related according to the linear model

$$d_{k,n} = \boldsymbol{u}_{k,n}^T \boldsymbol{w}_* + v_{k,n}, \forall n \in \mathbb{Z}_{\geq 0}, \forall k \in \mathcal{N}, \tag{1}$$

where $\mathcal{N}$ denotes the node set: $\mathcal{N} = \{1, \ldots, K\}$ and $v_{k,n}$ is the additive noise process with variance equal to $\sigma_k^2, \forall k \in \mathcal{N}$. An example of such a network is illustrated in Fig. 1. We assume that each node is able to communicate with a subset of $\mathcal{N}$, namely $\mathcal{N}_k$, which is the so-called *neighborhood* of k. In distributed adaptive learning, the estimates, at each node, are generated by exploiting: a) the sensed information, i.e., the measurement pair, and b) the information received by the neighborhood, whereas in the classical adaptive learning, only the measurements are taken into consideration. At each node, say

$k \in \mathcal{N}$, and at every time instant, this extra information comprises the estimates of the unknown vector, occurring from the nodes with which communication is possible, i.e., $\forall l \in \mathcal{N}_k$. It is by now well established, e.g., [9], [11], that this information-exchange results in a faster convergence speed, as well as a lower steady state error floor, compared to the case where the nodes operate individually. Furthermore, if the nodes cooperate with each other and under a properly chosen algorithmic scheme, asymptotic consensus can be achieved; that is, the nodes converge to the *same* estimate, e.g., [10], [12], [27]. Further details on the diffusion methodology will be presented in Section IV.

A. Krylov Subspaces and the Reduced Rank Wiener Solution

Our kickoff point is the Wiener filtering task. Throughout this section, the notational dependence on the nodes is suppressed for simplicity purposes, since the results hold true for all nodes. It can been shown, e.g., [21], that the solution that minimizes the mean-square error (MSE), i.e., $\mathbb{E}[(d_n - \boldsymbol{u}_n^T\boldsymbol{w})^2]$, where $d_n, \boldsymbol{u}_n$ are related via (1), satisfies the celebrated Wiener-Hopf equation, given by

$$\boldsymbol{p} = \boldsymbol{R}\boldsymbol{w}, \tag{2}$$

where the $m \times m$ matrix $\boldsymbol{R} = \mathbb{E}[\boldsymbol{u}_n\boldsymbol{u}_n^T]$ is the so-called input autocorrelation matrix, and the vector $\boldsymbol{p} = \mathbb{E}[d_n\boldsymbol{u}_n]$ is the cross-correlation vector between the input and the desired response. If the matrix $\boldsymbol{R}$ is invertible, which is usually the case, then the solution of (2) is the unknown vector $\boldsymbol{w}_*$, e.g., [28]. Throughout this paper, we will assume that $\boldsymbol{R}$ is invertible. Our main goal in this paper is to use the Wiener MSE solution in its constrained form. Since our objective is to reduce dimensionality, we are going to search for the filter that minimizes the MSE and at the same time lies in a lower dimension subspace. This brings Krylov spaces into the scene.

Given an $m \times m$ matrix $\boldsymbol{A}$ and a vector $\boldsymbol{w} \in \mathbb{R}^m$, the Krylov subspace of dimension $D < m$ is defined as $K_D(\boldsymbol{A}, \boldsymbol{w}) := \text{span}\{\boldsymbol{w}, \boldsymbol{A}\boldsymbol{w}, \ldots, \boldsymbol{A}^{D-1}\boldsymbol{w}\}$. The Krylov subspaces play a central role and they have been employed in the reduced rank adaptive filtering task, e.g., [16], [29], and it has been observed that they provide a good trade-off between the dimensionality reduction and the performance of the developed algorithms, due to their strong connection with the Wiener solution. In the sequel, we will comment on the physical reasoning of these subspaces. Following a similar rationale as in [30] and in [29], we denote by $\boldsymbol{w}_{WF}^{(D)} \in \mathbb{R}^m$ the solution of the Wiener-Hopf equation in the Krylov subspace, $K_D(\boldsymbol{R}, \boldsymbol{p})$. In words, $\boldsymbol{w}_{WF}^{(D)}$ is the vector we obtain if we solve the Wiener-Hopf equation and constraint the solution to lie inside $K_D(\boldsymbol{R}, \boldsymbol{p})$. This vector is the optimum one, in the MSE sense, which belongs to this subspace, e.g., [29]. Moreover, it has an elegant geometrical property; it is the projection of $\boldsymbol{w}_*$ in the $\boldsymbol{R}$-norm sense (see Appendix A) onto $K_D(\boldsymbol{R}, \boldsymbol{p})$, i.e., $\boldsymbol{w}_{WF}^{(D)} = P^{(\boldsymbol{R})}_{K_D(\boldsymbol{R},\boldsymbol{p})}(\boldsymbol{w}_*)$, where the operator $P^{(\boldsymbol{R})}_{K_D(\boldsymbol{R},\boldsymbol{p})}(\boldsymbol{w}_*)$ stands for the previously mentioned projection. Analytically, it is given by [16]:

$$\boldsymbol{w}_{WF}^{(D)} = \boldsymbol{T}(\boldsymbol{T}^T\boldsymbol{R}\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{p} = \boldsymbol{T}(\boldsymbol{T}^T\boldsymbol{R}\boldsymbol{T})^{-1}\boldsymbol{T}^T\boldsymbol{R}\boldsymbol{w}_*,$$

where $\boldsymbol{T} \in \mathbb{R}^{m\times D}$ is a matrix whose columns form an orthonormal basis for the subspace $K_D(\boldsymbol{R}, \boldsymbol{p})$.

Now, let us examine one more viewpoint which clarifies the connection between $\boldsymbol{w}_{WF}^{(D)}$ and $\boldsymbol{w}_*$. Our starting point will be the MultiStage Nested Wiener Filter (MSNWF), proposed in [31]. Put in general terms, the MSNWF solves the Wiener-Hopf equation, without inversion of the matrix $\boldsymbol{R}$. The MSNWF consists of m filters, $\boldsymbol{t}_i \in \mathbb{R}^m$, $i = 1, \ldots, m$, which produce m outputs $d_i[n] = \boldsymbol{t}_i^T\boldsymbol{u}_n$, $i = 1, \ldots, m$, and they are computed via the following optimization

$$\begin{aligned} i = 2, \ldots, m, \quad \boldsymbol{t}_i &= \arg\max_{\boldsymbol{t}}\{\boldsymbol{t}^T\boldsymbol{R}\boldsymbol{t}_{i-1}\} \\ &= \arg\max_{\boldsymbol{t}}\mathbb{E}\left\{d_i[n]d_{i-1}[n]\right\} \\ &\text{s.t.} \quad \boldsymbol{t}^T\boldsymbol{t} = 1 \\ &\quad\ \boldsymbol{t}^T\boldsymbol{t}_r = 0, \quad r = 1, \ldots, i-1, \end{aligned} \tag{3}$$

and $\boldsymbol{t}_1$ occurs by maximization of $\boldsymbol{t}_1 = \arg\max_{\boldsymbol{t}}\mathbb{E}\{\boldsymbol{t}^T\boldsymbol{u}_n d_n\} = \arg\max_{\boldsymbol{t}}\mathbb{E}\{d_1[n]d_n\}$, s.t. $\boldsymbol{t}^T\boldsymbol{t} = 1$. The physical reasoning of the previous optimization problem can be summarized as follows. The first filter $\boldsymbol{t}_1$ is obtained so as to maximize the correlation of the output $d_1[n]$ and the desired one d_n. The i-th filter is computed in a similar notion, which is the maximization of the correlation between the current and the previous outputs, i.e., $d_i[n]$ and $d_{i-1}[n]$. Furthermore, as it can be seen by (3), we restrict the filters to be orthonormal. It has been proved, e.g., in [18], that the m-th output response occurring by the MSNWF, equals to the one occurring by the unknown vector, i.e., $\hat{d}_n = \boldsymbol{u}_n^T\boldsymbol{w}_*$.

It is very interesting to see what happens if one stops the iterations in (3), at step D. It turns out that the obtained solution corresponds to the reduced rank Wiener Filter (WF), $\boldsymbol{w}_{\text{WF}}^{(D)}$. Moreover, as it has been proved, e.g., [30], the filters $\boldsymbol{t}_i, i = 1, \ldots, D$, form a basis in the Krylov subspace; in other words, if we group them in a matrix, we obtain the matrix $\boldsymbol{T}$.

Now, let us see how the previous arguments can be employed in the adaptive filtering task. As we have already mentioned, in the reduced rank adaptive filtering, instead of seeking for the unknown solution, which in our case is $\boldsymbol{w}_*$, one seeks for the projection of it onto a subspace of reduced dimension; in our case this is the projection, in the $\boldsymbol{R}$ norm sense, onto $K_D(\boldsymbol{R}, \boldsymbol{p})$. Obviously, the fact that instead of tracking $\boldsymbol{w}_*$, one tracks for its projection in a subspace of lower dimension, results at an increased error floor in the steady state, which depends on the distance between the true solution, and the reduced rank one. These issues will be clarified in the sequel.

A natural question rising is how accurately can $\boldsymbol{w}_*$ be identified by employing the Krylov subspace rationale. It has been proved, e.g., [16], that

$$\left\|\boldsymbol{w}_* - \boldsymbol{w}_{WF}^{(D)}\right\| \leq 2\tau_{\min}^{-1/2}\|\boldsymbol{w}_*\|_{\boldsymbol{R}}\alpha_\kappa^D, \tag{4}$$

where $\tau_{\min}$ is the smallest eigenvalue of the matrix $\boldsymbol{R}$, $\alpha_\kappa := (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$ with $\kappa := \|\boldsymbol{R}\|\|\boldsymbol{R}^{-1}\| \geq 1$. From the previous findings, it can be readily observed that the input statistics play a central role in the performance of the algorithms built around the Krylov subspaces. More specifically, if the eigenvalue spread of the matrix $\boldsymbol{R}$ is large, which yields a large value of α_κ, the upper bound in the previous inequality is larger, and it has also been experimentally verified that the performance of the respective algorithm is degraded.

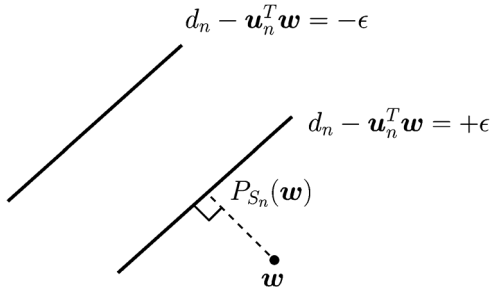


Fig. 2. Illustration of a hyperslab, as well as the projection of an arbitrary vector onto it.

III. Set-Theoretic Estimation

A. The Full Rank Case

In this paper, the set-theoretic estimation rationale will be adopted, e.g., [22], [25]. At each time instant, a property (closed convex) set is constructed, based on the received set of measurements, $(d_n, \boldsymbol{u}_n)$, and the noise statistics, such that the unknown vector lies within this set, with high probability. The goal is to find a point that lies in the intersection of an infinite number of sets (with the possible exception of a finite number of them).

The adopted methodology was presented in [23] and generalized in [24], [26], and comprises a sequence of projections. It has been shown that under certain assumptions, the algorithm converges to a point that lies arbitrarily close to the intersection of these sets. This algorithmic scheme, can be seen as a generalization of the classical Projections Onto Convex Sets (POCS) algorithm, e.g., [22], [32], [33]. The difference lies in the fact that in the POCS, the number of involved sets is finite, whereas in its adaptive setting, an infinite number of sets is considered.

In the current study, the adopted property sets take the form of hyperslabs, e.g., [12]. [25], [34]. The definition of a hyperslab is given by: $S_n = \{\boldsymbol{w} \in \mathbb{R}^m : |d_n - \boldsymbol{u}_n^T\boldsymbol{w}| \leq \epsilon\}$, where $\epsilon > 0$ is a user-defined threshold, which takes into consideration the statistics of the noise. In words, a vector is in agreement with the measurements $(d_n, \boldsymbol{u}_n)$ if the distance between the desired response, d_n, and the response to the input, $\boldsymbol{u}_n^T\boldsymbol{w}$, is smaller or equal than ϵ. Such criteria have been proposed in the context of the robust statistics rationale and successfully used in the support vector machine framework, e.g., [35], [36]. The projection operator onto a hyperslab takes the following simple analytic form

$$\forall \boldsymbol{w} \in \mathbb{R}^m, \quad P_{S_n}(\boldsymbol{w}) = \boldsymbol{w} + \beta_n \boldsymbol{u}_n, \tag{5}$$

where

$$\beta_n \begin{cases} \frac{d_n - \boldsymbol{u}_n^T\boldsymbol{w} + \epsilon}{\|\boldsymbol{u}_n\|^2}, & \text{if } d_n - \boldsymbol{u}_n^T\boldsymbol{w} < -\epsilon, \\ 0, & \text{if } \left|d_n - \boldsymbol{u}_n^T\boldsymbol{w}\right| \leq \epsilon, \\ \frac{d_n - \boldsymbol{u}_n^T\boldsymbol{w} - \epsilon}{\|\boldsymbol{u}_n\|^2}, & \text{if } d_n - \boldsymbol{u}_n^T\boldsymbol{w} > \epsilon. \end{cases}$$

Finally, the geometry of a hyperslab is illustrated in Fig. 2.

B. The Reduced Rank Case

According to the discussion in the previous subsection, the property sets are constructed so as to contain the unknown vector $\boldsymbol{w}_*$ with a high probability. The question now is which strategy to follow in the case of reduced rank scenarios. Our kick off point will be the reduced rank Wiener solution. More specifically, the property sets will be constructed so as to contain the vector $w_{\mathrm{WF}}^{(D)}$ with a high probability. As it will become clear later on, this can be guaranteed by seeking for points that lie in the intersection of the hyperslabs and the Krylov subspace, i.e., $K_D(\boldsymbol{R}, \boldsymbol{p})$. Let us define the set $\overline{S}_n := S_n \cap K_D(\boldsymbol{R}, \boldsymbol{p}) = \{\boldsymbol{w} \in K_D(\boldsymbol{R}, \boldsymbol{p}) : |d_n - \boldsymbol{u}_n^T\boldsymbol{w}| \leq \epsilon\}$. Recall from the discussion in Section II that $w_{\mathrm{WF}}^{(D)} \in K_D(\boldsymbol{R}, \boldsymbol{p})$. In order to have $w_{\mathrm{WF}}^{(D)} \in \overline{S}_n$, the following must hold true

$$\begin{aligned} \left|d_n - \boldsymbol{u}_n^T w_{\mathrm{WF}}^{(D)}\right| \leq \epsilon \Leftrightarrow \left|\boldsymbol{u}_n^T\boldsymbol{w}_* + v_n - \boldsymbol{u}_n^T w_{\mathrm{WF}}^{(D)}\right| \\ \leq \epsilon \Leftrightarrow \left|\boldsymbol{u}_n^T\left(\boldsymbol{w}_* - \boldsymbol{w}_{\mathrm{WF}}^{(D)}\right) + v_n\right| \leq \epsilon. \end{aligned} \tag{6}$$

From (6), it can be seen that the parameter ϵ, which determines the width of the hyperslab, determines the probability that $w_{\mathrm{WF}}^{(D)} \in \overline{S}_n$, in the sense that the larger the ϵ, the larger the possibility that the previously mentioned condition will hold. Obviously, in the full rank case, in which the condition to be satisfied is $\boldsymbol{w}_* \in S_n$, the only term, which dictates the choice of ϵ, is v_n. Hence, the width of the hyperslabs is chosen with respect to the statistics of the noise. In the reduced rank case, besides the noise, one has to take into consideration the term $\boldsymbol{u}_n^T(\boldsymbol{w}_* - w_{\mathrm{WF}}^{(D)})$. However, in practice, as it has been documented in [16], in cases where the eigenvalue spread of $\boldsymbol{R}$ is close to 1, which implies that the distance between $\boldsymbol{w}_*$ and $w_{\mathrm{WF}}^{(D)}$ is small (see also (4)), the noise term is the dominant one. Hence, if the user-controlled parameter, ϵ, is defined according to the noise statistics, the condition of having $w_{\mathrm{WF}}^{(D)} \in \overline{S}_n$, holds with a high probability. In the sequel, a technique appropriate for the case where the eigenvalue spread is large, will be proposed in order to overstep this limitation.

In order to construct the subspace, knowledge on the statistics of the input and the desired response, i.e., $\boldsymbol{R}, \boldsymbol{p}$, is required. A reasonable strategy is to rely on estimates of the previously mentioned quantities. To this end, the autocorrelation is estimated via $\hat{\boldsymbol{R}}_n := \sum_{j=0}^{n-1} \zeta^{n-1-j}\boldsymbol{u}_j\boldsymbol{u}_j^T$ and the crosscorrelation via $\hat{\boldsymbol{p}}_n := \sum_{j=0}^{n-1} \zeta^{n-1-j} d_j \boldsymbol{u}_j$, where $\zeta \in (0, 1]$ is the so-called forgetting factor, employed in order to "forget" past values in time varying scenarios. The estimates, $\hat{\boldsymbol{R}}_n, \hat{\boldsymbol{p}}_n$ are updated $\forall n \in \mathbb{Z}_{\geq 0}$, according to the following formulas: $\hat{\boldsymbol{R}}_n = \zeta\hat{\boldsymbol{R}}_{n-1} + \boldsymbol{u}_{n-1}\boldsymbol{u}_{n-1}^T$ and $\hat{\boldsymbol{p}}_n = \zeta\hat{\boldsymbol{p}}_{n-1} + d_{n-1}\boldsymbol{u}_{n-1}$. Having obtained the estimates of $\hat{\boldsymbol{R}}_n$ and $\hat{\boldsymbol{p}}_n$, our goal now is to develop the projection operator that projects an estimate to the intersection of the corresponding hyperslab and the current estimate of the Krylov subspace, i.e., $S_n \cap K_n$, where $K_n := K_D(\hat{\boldsymbol{R}}_n, \hat{\boldsymbol{p}}_n)$.

Claim 1: The projection of a vector lying in K_n onto $S_n \cap K_n$ is given by

$$\forall \boldsymbol{w} \in K_n: \quad P_{S_n \cap K_n}(\boldsymbol{w}) = \boldsymbol{w} + \tilde{\beta}\hat{\boldsymbol{T}}_n\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n, \tag{7}$$

where $\hat{\boldsymbol{T}}_n$ is an $m \times D$ matrix, whose columns form an orthonormal basis of K_n and

$$\tilde{\beta} = \begin{cases} \frac{d_n - \boldsymbol{w}^T\hat{\boldsymbol{T}}_n\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n + \epsilon}{\left\|\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n\right\|^2}, & \text{if } d_n - \boldsymbol{w}^T\hat{\boldsymbol{T}}_n\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n < -\epsilon, \\ 0, & \text{if } \left|d_n - \boldsymbol{w}^T\hat{\boldsymbol{T}}_n\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n\right| \leq \epsilon \\ \frac{d_n - \boldsymbol{w}^T\hat{\boldsymbol{T}}_n\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n - \epsilon}{\left\|\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n\right\|^2}, & \text{if } d_n - \boldsymbol{w}^T\hat{\boldsymbol{T}}_n\hat{\boldsymbol{T}}_n^T\boldsymbol{u}_n > \epsilon. \end{cases}$$

Proof: The proof is given in Appendix B. ■

Now, let us see how the case where the denominator in the previous equation equals to zero is treated. First of all, recall that the columns of $\hat{\boldsymbol{T}}_n$ form a basis for the Krylov subspace. If $\hat{\boldsymbol{T}}_n^T \boldsymbol{u}_n = \boldsymbol{0}$, this means that the vector $\boldsymbol{u}_n$ is perpendicular to the Krylov subspace. Moreover, it holds, e.g., [25], that the vector $\boldsymbol{u}_n$ is perpendicular to the hyperplanes $H_{1,n} = \{\boldsymbol{w} \in \mathbb{R}^m : d_n - \boldsymbol{u}_n^T \boldsymbol{w} = -\epsilon\}$ and $H_{2,n} = \{\boldsymbol{w} \in \mathbb{R}^m : d_n - \boldsymbol{u}_n^T \boldsymbol{w} = +\epsilon\}$, which constitute the hyperplanes that define the hyperslab. These two facts imply that $\hat{\boldsymbol{T}}_n^T \boldsymbol{u}_n = \boldsymbol{0}$ in the case where the subspace is "parallel" to the hyperslab. This case is treated as in the full rank case, i.e., when the input vector is $\boldsymbol{0}$, e.g., [25]. To be more specific, if such an input vector occurs, it is not taken into consideration in the algorithmic flow.

IV. Cooperation Strategies in Diffusion Networks and the Consensus Matrix

In this section, we will describe how the nodes cooperate with each other in order to exploit the spatially received estimates. First of all, depending on the strategy with which the received estimates from the neighboring nodes are embedded in the adaptation, the following cooperation directions are defined:

- Combine-Adapt, in which, at each node, the estimates received from the neighborhood are combined in a particular way, and then the aggregate is put into the adaptation step, e.g., [11], [12], [17].
- Adapt-Combine, where before the combination step, the adaptation takes place, e.g., [9], [27].
- Consensus based, where the computations are made in parallel and there is no clear distinction between the combine and the adaptation step [10], [37].

In the current paper, the combine-adapt cooperation strategy will be followed. We assume that the following statements, regarding the network, hold true: $l \in \mathcal{N}_k \Leftrightarrow k \in \mathcal{N}_l$ and $k \in \mathcal{N}_k, \forall k \in \mathcal{N}$ and the network is assumed to be strongly connected, i.e., there exists a possibly multihop path, connecting every two nodes of the network. These assumptions are very common in adaptive distributed learning (see for example [6], [7]). At each node and at each time instant, the estimates received from the neighborhood are fused under a certain protocol. To this end, we define the combination coefficients, such that $c_{k,l}(n) > 0$, if $l \in \mathcal{N}_k$, $c_{k,l}(n) = 0$, if $l \notin \mathcal{N}_k$ and $\sum_{l \in \mathcal{N}_k} c_{k,l}(n) = 1, \forall k \in \mathcal{N}$. In words, every node assigns a weight to each one of the estimates, which are received from the neighborhood, and a convex combination of them is computed; this aggregate takes part in the adaptation step. The steps of the combine-adapt cooperation strategy are given in detail in Section V-B. Two well known examples of combination coefficients are: the Metropolis rule, where

$$c_{k,l}(n) = \begin{cases} \frac{1}{\max\{|\mathcal{N}_k|, |\mathcal{N}_l|\}}, & \text{if } l \in \mathcal{N}_k \text{ and } l \neq k, \\ 1 - \sum_{l \in \mathcal{N}_k \setminus k} c_{k,l}(n), & \text{if } l = k, \\ 0, & \text{otherwise,} \end{cases}$$

and the uniform rule, in which the coefficients are defined as

$$c_{k,l}(n) = \begin{cases} \frac{1}{|\mathcal{N}_k|}, & \text{if } l \in \mathcal{N}_k, \\ 0, & \text{otherwise.} \end{cases}$$

Gathering all the coefficients in a matrix, we define the combination matrix $\boldsymbol{C}_n$, in which the k,l-th component equals to $c_{k,l}(n)$. Now, let us give the definition of the consensus subspace $\mathcal{O}$. This linear subspace is defined: $\mathcal{O} := \{\underline{\boldsymbol{w}} \in \mathbb{R}^{Km} : \underline{\boldsymbol{w}} = [\boldsymbol{w}^T, \ldots, \boldsymbol{w}^T]^T, \boldsymbol{w} \in \mathbb{R}^m\}$, and its dimension equals to m. The $Km \times Km$ consensus matrix is given by $\boldsymbol{P}_n = \boldsymbol{C}_n \otimes \boldsymbol{I}_m$, where the symbol $\otimes$ stands for the Kronecker product and $\boldsymbol{I}_m$ stands for the $m \times m$ identity matrix. Some very useful properties of the consensus matrix as well as the consensus subspace are [27]:

1) $\boldsymbol{P}_n \boldsymbol{1}_{Km} = \boldsymbol{1}_{Km}$, where $\boldsymbol{1}_{Km} \in \mathbb{R}^{Km}$ is the vector of ones. Throughout the paper, we assume that the involved consensus matrices are doubly-stochastic, i.e., $\boldsymbol{P}_n^T \boldsymbol{1}_{Km} = \boldsymbol{1}_{Km}$
2) $\|\boldsymbol{P}_n\| = 1$.
3) Any consensus matrix $\boldsymbol{P}_n$ can be decomposed as
$$\boldsymbol{P}_n = \boldsymbol{X}_n + \boldsymbol{B}\boldsymbol{B}^T,$$
where $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m]$ is an $Km \times m$ matrix, and $\boldsymbol{b}_k = (\boldsymbol{1}_K \otimes \boldsymbol{e}_k)/\sqrt{K}$, $\boldsymbol{e}_k$ is an $m \times 1$ vector of zeros except the k-th entry, which is one and $\boldsymbol{X}_n$ is an $Km \times Km$ matrix for which it holds that $\|\boldsymbol{X}_n\| < 1$.
4) $\boldsymbol{P}_n \breve{\underline{\boldsymbol{w}}} = \breve{\underline{\boldsymbol{w}}}, \forall \breve{\underline{\boldsymbol{w}}} \in \mathcal{O}$.
5) The vectors $\boldsymbol{b}_k, k = 1, \ldots, m$ constitute a basis for $\mathcal{O}$. The projection of a vector, $\underline{\boldsymbol{w}} \in \mathbb{R}^{Km}$, onto this linear subspace is given by $P_{\mathcal{O}}(\underline{\boldsymbol{w}}) := \boldsymbol{B}\boldsymbol{B}^T \underline{\boldsymbol{w}}, \forall \underline{\boldsymbol{w}} \in \mathbb{R}^{Km}$.

V. Proposed Scheme

First of all, it has to be pointed out, that despite the fact that the nodes seek for the same unknown vector, the input as well as the noise statistics differ from node to node. Hence, in contrast to the non-distributed scenario, here, we should take into consideration the statistics from all the nodes. Let us define the mean square error loss function $\mathcal{L} : \mathbb{R}^m \to [0, +\infty)$, for the whole network

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}) &= \frac{1}{K} \sum_{k \in \mathcal{N}} \mathbb{E}\left\{(d_{k,n} - \boldsymbol{u}_{k,n}^T \boldsymbol{w})^2\right\} \\ &= \frac{1}{K} \sum_{k \in \mathcal{N}} \left(\boldsymbol{w}^T \boldsymbol{R}_k \boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{p}_k + \sigma_{d_k}^2\right) \\ &= \boldsymbol{w}^T \boldsymbol{R}' \boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{p}' + \frac{1}{K} \sum_{k \in \mathcal{N}} \sigma_{d_k}^2, \end{aligned} \tag{8}$$

where $\sigma_{d_k}^2 = E\{d_{k,n}^2\}$, $\boldsymbol{R}' = (1/K)\sum_{k \in \mathcal{N}} E\{\boldsymbol{u}_{k,n}\boldsymbol{u}_{k,n}^T\} = (1/K)\sum_{k \in \mathcal{N}} \boldsymbol{R}_k$ and $\boldsymbol{p}' = (1/K)\sum_{k \in \mathcal{N}} E\{d_{k,n}\boldsymbol{u}_{k,n}\} = (1/K)\sum_{k \in \mathcal{N}} \boldsymbol{p}_k$. It can be readily shown following similar steps as in [16], that the solution minimizing (8) is given by $\boldsymbol{w}_* = \boldsymbol{R}'^{-1}\boldsymbol{p}'$. This argument indicates that a reasonable strategy in order to achieve dimensionality reduction is to construct the Krylov subspace relying on $\boldsymbol{R}'$ and $\boldsymbol{p}'$; i.e., the average values relying on approximations of the previously mentioned quantities. To this end, at each node, the following approximations are computed: $\hat{\boldsymbol{R}}_n' = (1/K)\sum_{k \in \mathcal{N}} \hat{\boldsymbol{R}}_{k,n}'$, where $\hat{\boldsymbol{R}}_{k,n}' = \zeta \hat{\boldsymbol{R}}_{k,n-1} + \boldsymbol{u}_{k,n-1}\boldsymbol{u}_{k,n-1}^T$ and $\hat{\boldsymbol{p}}_n' = (1/K)\sum_{k \in \mathcal{N}} \hat{\boldsymbol{p}}_{k,n}'$, with $\hat{\boldsymbol{p}}_{k,n} = \zeta \hat{\boldsymbol{p}}_{k,n-1} + d_{k,n-1}\boldsymbol{u}_{k,n-1}$ and ζ is the forgetting factor. From the previous relations, it can be observed that in order to construct the respective subspace, every node must have access

to measurements coming out from the other nodes of the network, i.e., $\boldsymbol{u}_{k,n}, d_{k,n}$; however, this is, in general, infeasible in distributed networks. In the sequel, we present two techniques which will help us overstep this obstacle.

A. Enhancing the Information Flow

First of all, it should be stressed out that in system identification problems the input is defined as follows: $\boldsymbol{u}_{k,n} = [u_{k,n} \; u_{k,n-1} \ldots u_{k,n-m+1}]^T$. Hence, the novel information at each time instant comprises of two numbers: $u_{k,n}$ and $d_{k,n}$. In order to enhance the information flow, the following strategies are adopted.

1) We assume that $\hat{\boldsymbol{R}}_n^{'}$ and $\hat{\boldsymbol{p}}_n^{'}$ will not be updated every time instant but every L time instants instead. Thus the coefficients $u_{k,n}$ and $d_{k,n}$ will be delivered to the other nodes of the network within a time window of size L. This parameter is chosen with respect to the size of the network as well as the maximum distance between two nodes. As it will become clear in the simulations section, the larger the L the worse the performance of the algorithm; this behavior is due to the fact that for a large time window, $\hat{\boldsymbol{R}}_n^{'}$ and $\hat{\boldsymbol{p}}_n^{'}$ are updated less frequently and their convergence to a good approximation is slowed down. Nevertheless, as it will become apparent in the simulations section, provided that L does not take too large values, the algorithm turns out to be relatively insensitive to its choice.
2) We adopt a multi-cluster architecture (see for example [38]) for the network in order to improve the flow of transmitted information. In principle, nodes which are connected to a large number of neighbors are "equipped" with better transmission capabilities. Despite the fact that the issue of clustering the nodes according to predefined protocols has been extensively discussed in the literature, see [38] and references therein, complex protocols are beyond the scope of this paper. So, we adopt a simple hierarchical protocol, which has been employed in the context of adaptive distributed learning in [39]. More specifically, we classify the nodes, according to the number of their neighbors, into two subclasses: the hierarchical and the non-hierarchical ones. The former are able to communicate over three nodes, whereas the latter are not, and every non-hierarchical node is connected to a hierarchical one. The rationale is to assign enhanced transmission capabilities to the nodes which have many neighbors; through this procedure the information is delivered faster throughout the network, e.g., [38].

Now, let us see how the information needed to construct the Krylov subspace is distributed over such a network, which is illustrated in Fig. 3. Notice that the network comprises of $K = 14$ nodes and the number of the hierarchical nodes equals to 3. At each time instant, nodes have to transmit D coefficients to their neighborhood; these are the updated components lying in the reduced space $\mathbb{R}^D$. At time instant 1, node 1 transmits to node 2, $u_{1,1}, d_{1,1}$, at time instant 2, $u_{4,1}, d_{4,1}$, at 3, $u_{5,1}, d_{5,1}$ and at time instant 4, $u_{6,1}, d_{6,1}$. Node 2, at time instant $n = 1$, transmits to 3, $u_{2,1}, d_{2,1}, u_{7,1}, d_{7,1}$. At $n = 2$, $u_{1,1}, d_{1,1}, u_{8,1}, d_{8,1}$, at $n = 3$, $u_{4,1}, d_{4,1}, u_{10,1}, d_{10,1}$, at $n = 4$, $u_{5,1}, d_{5,1}, u_{9,1}, d_{9,1}$ and at $n = 5$, $u_{6,1}, d_{6,1}$. The rest of the exchanges follow a similar

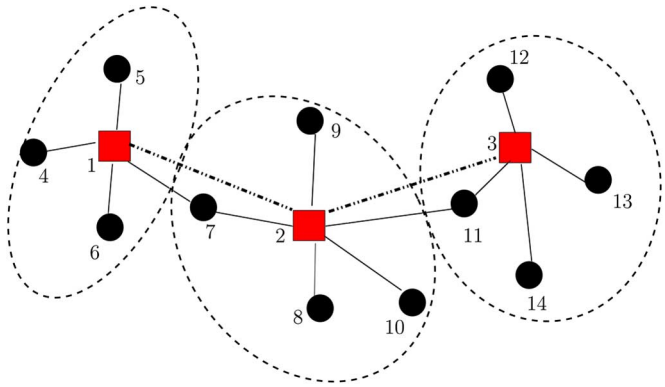


Fig. 3. Illustration of a hierarchical network with $L = 5$. The solid lines denote the simple communication links, whereas the dashed-dotted ones the hierarchical communication links.

philosophy. The largest number of coefficients is transmitted by node 2 and amounts to $D+4$, where D comes from the D coefficients of the estimate and the other four from the information needed to construct the subspace. In the full rank scenario, every node has to transmit m coefficients to each neighboring node. Hence, if D is much smaller than m, which is the case of our interest, then the nodes transmit fewer coefficients, if they seek for a reduced rank solution.

Unfortunately, in networks with a large number of nodes and/or in scenarios where the longest path, among the nodes of the network, is large, the previously mentioned techniques may fail. Nevertheless, as it will become apparent in the simulations section, another route can be followed. Indeed, the Krylov subspace can be constructed by exploiting information coming from a single node, e.g., a master node, without significant degradation of the performance of the algorithm. It can be readily obtained that, if the information of a single node has to be delivered throughout the network, each node transmits $D+2$ coefficients at most. Hence in this scenario, the only limitation is to use a large enough L, which depends on the longest path among the nodes of the network, and then distribute the two coefficients, which are used to construct the subspace. A possible criterion in order to chose the master node is to find the node with the smallest eigenvalue spread, as (4) suggests. Techniques for finding this "optimum" node are beyond the scope of this paper and will be presented in a future work. In the simulations section, the case of choosing the "worst" node is also adopted in order to study the sensitivity of this scenario in failing to choose the "best" node.

Finally, if the statistics are the same for the nodes of the network, then the Krylov subspaces can be constructed locally, and then the information transmitted by each node drops to D coefficients, i.e., the length of the reduced rank estimate.

B. The Algorithmic Scheme

As it has been already mentioned, our goal has now become to search for estimates that lie in the reduced D-dimensional Krylov subspace. However, in general, any vector in such a subspace is expressed in terms of m components, since it is a subset of $\mathbb{R}^m$. Our next goal becomes to map the respective estimates in the $\mathbb{R}^D$ subspace; this mapping will result in the description of the estimates in terms of D components. Nevertheless, the mapping which leads vectors from the Krylov subspace to $\mathbb{R}^D$,

is known. Moreover, the inverse mapping leading vectors from $\mathbb{R}^D$ to the subspace is also known. This correspondence between vectors of the Krylov subspace with vectors lying in $\mathbb{R}^D$ will be the kick-off point in order to reduce the communication load. More specifically, at each node, vectors which belong in $\mathbb{R}^D$ will be computed and transmitted, reducing the communication load; these vectors can be readily mapped, locally at each node, back to the original Krylov subspace where they belong.

Let us define the $m \times D$ matrix $\hat{\boldsymbol{T}}_n$ the columns of which form a basis for $K_n = K_D(\hat{\boldsymbol{R}}_n^{'}, \hat{\boldsymbol{p}}_n^{'})$. The following holds: $\forall \tilde{\boldsymbol{w}} \in \mathbb{R}^D, \exists \boldsymbol{w} \in K_n : \boldsymbol{w} = \hat{\boldsymbol{T}}_n \tilde{\boldsymbol{w}}$ and $\tilde{\boldsymbol{w}} = \hat{\boldsymbol{T}}_n^T \boldsymbol{w}$[1], [16]. According to the previous discussion, the matrix $\hat{\boldsymbol{T}}_n^T$ maps vectors, of dimension m which belong in K_n, to the reduced dimension space, i.e., $\mathbb{R}^D$, whereas $\hat{\boldsymbol{T}}_n$ maps vectors lying in $\mathbb{R}^D$ to $K_n \subset \mathbb{R}^m$.

The steps of the algorithm for each node k and at time instant n, can be summarized as follows:

- The estimates, of reduced dimension, from the neighborhood, i.e., $\tilde{\boldsymbol{w}}_{l,n} \in \mathbb{R}^D, \forall l \in \mathcal{N}_k$, are received and convexly combined, with respect to the adopted combination strategy in order to produce $\tilde{\boldsymbol{\phi}}_{k,n} := \sum_{l \in \mathcal{N}_k} c_{k,l}(n) \tilde{\boldsymbol{w}}_{l,n}$. As already said, these estimates are related to their counterparts in the Krylov subspace in $\mathbb{R}^m$ ones, according to: $\forall n \in \mathbb{Z}_{\geq 0}, \forall k \in \mathcal{N}, \tilde{\boldsymbol{w}}_{k,n} = \hat{\boldsymbol{T}}_n^T \boldsymbol{w}_{k,n}$ (see also Appendix B).
- Taking into consideration the newly received information, i.e., $(d_{k,n}, \boldsymbol{u}_{k,n})$ the following hyperslab is defined in $\mathbb{R}^D$: $\tilde{S}_{k,n} := \{\tilde{\boldsymbol{w}} \in \mathbb{R}^D : |d_{k,n} - \boldsymbol{u}_{k,n}^T \hat{\boldsymbol{T}}_n \tilde{\boldsymbol{w}}| \leq \epsilon_k\}$, where $\epsilon_k > 0$ can vary from node to node, depending on the noise statistics. The aggregate $\tilde{\boldsymbol{\phi}}_{k,n}$, which was computed at the previous step, is projected onto the q most recent hyperslabs, and then a convex combination of the resulting projections is computed. It has been verified, that by projecting onto a $q > 1$ number of hyperslabs the convergence speed is accelerated [34], [40].
- The information needed in order to update the subspace is distributed over the network, using one of the techniques described in Section V-A. If $\mathrm{mod}(n, L) = 0$, then $\hat{\boldsymbol{R}}_{k,n}, \hat{\boldsymbol{p}}_{k,n}$ are updated, and the matrix $\hat{\boldsymbol{T}}_{n+1}$ is computed.

[1]From now on, the tilded vectors will stand for vectors lying in $\mathbb{R}^D$.

The previous can be encoded in the following formula.

$$\tilde{\boldsymbol{w}}_{k,n+1} = \tilde{\boldsymbol{\phi}}_{k,n} + \tilde{\mu}_{k,n} \left(\sum_{j \in \mathcal{J}} \omega_{k,j} P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_{k,n}) - \tilde{\boldsymbol{\phi}}_{k,n} \right), \quad (9)$$

where $\mathcal{J} = \overline{\max\{0, n-q+1\}, n}$, $\sum_{j \in \mathcal{J}} \omega_{k,j} = 1, \forall k \in \mathcal{N}$ and $\tilde{\mu}_{k,n} \in (0, 2\tilde{\mathcal{M}}_{k,n})$ where [25]: (see equation at the bottom of the page).

In the previously described scheme, the obtained estimates lie in $\mathbb{R}^D$, which implies that each sensor will transmit D coefficients at each time instant. The following claim clarifies the connection between the algorithm in (9) and the Krylov subspaces, discussed in the previous section.

Claim 2: Eq. (9) is equivalent to

$$\boldsymbol{w}_{k,n+1} = \hat{\boldsymbol{T}}_{n+1} \hat{\boldsymbol{T}}_n^T \Bigg(\boldsymbol{\phi}_{k,n} + \mu_{k,n} \left(\sum_{j \in \mathcal{J}} \omega_{k,j} P_{S_{k,j} \cap K_n}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n} \right) \Bigg), \quad (11)$$

where $\mu_{k,n} \in (0, 2\mathcal{M}_{k,n})$, $\boldsymbol{\phi}_{k,n} = \hat{\boldsymbol{T}}_n \tilde{\boldsymbol{\phi}}_{k,n}$, that is, the corresponding aggregate in the respective Krylov space, and (see equation at bottom of page).

Proof: The proof is given in Appendix C. ∎

The geometrical interpretation of the algorithm is given in Fig. 4. The complexity of the algorithm is of order $O(qD)$ coming from (9), $O(Km/L)$ from the update of $\hat{\boldsymbol{R}}_n^{'}$, and $O(Dm^2/L)$ due to the computation of $\hat{\boldsymbol{T}}_n$, e.g., [16]. It is important to notice that the dominant complexity-contributing terms, which are involved in the subspace computation, depend also on the frequency with which the subspace is constructed. Hence, if one is to reduce the computational load, a larger L must be chosen. Obviously, this results to a performance degradation; however as it will become clear in the simulations section, the algorithms turn out to be relatively insensitive to this parameter.

As it will become clear shortly, the algorithm enjoys a number of nice convergence properties. Despite the fact that at each node the recursion given in (9) is employed, the theoretical properties for (11) will be studied since the estimates computed

$$\tilde{\mathcal{M}}_{k,n} = \begin{cases} \dfrac{\sum_{j \in \mathcal{J}} \omega_{k,j} \left\| P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_{k,n}) - \tilde{\boldsymbol{\phi}}_{k,n} \right\|^2}{\left\| \sum_{j \in \mathcal{J}} \omega_{k,j} P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_{k,n}) - \tilde{\boldsymbol{\phi}}_{k,n} \right\|^2}, & \text{if } \left\| \sum_{j \in \mathcal{J}} \omega_{k,j} P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_{k,n}) - \tilde{\boldsymbol{\phi}}_{k,n} \right\| \neq 0, \\ 1, & \text{otherwise.} \end{cases} \quad (10)$$

$$\mathcal{M}_{k,n} = \begin{cases} \dfrac{\sum_{j \in \mathcal{J}_n} \omega_{k,j} \left\| P_{S_{k,j} \cap K_n}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n} \right\|^2}{\left\| \sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S_{k,j} \cap K_n}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n} \right\|^2}, & \text{if } \left\| \sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S_{k,j} \cap K_n}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n} \right\| \neq 0 \\ 1, & \text{otherwise.} \end{cases}$$

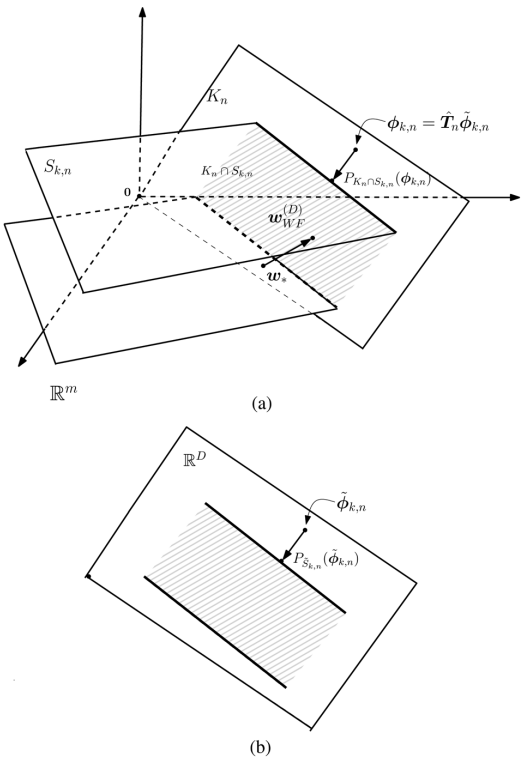


Fig. 4. (a) Geometrical illustration of the algorithm for $q = 1$. The aggregate $\boldsymbol{\phi}_{k,n}$, which belongs in the subspace, is projected onto the intersection of the subspace and the hyperslab, generated by the measurement data. (b) The algorithmic scheme in the reduced dimension space, i.e., $\mathbb{R}^D$.

by this scheme belong to the same subspace with $\boldsymbol{w}_{\mathrm{WF}}^{(D)}$ and from the fact that the two schemes are equivalent. For the algorithm in (11), we prove a number of nice convergence properties such as: monotonicity, asymptotic optimality and strong convergence to a point which lies in the consensus subspace. Moreover, we prove that the estimates at each node converge to a vector which belongs to the Krylov subspace. It is important to notice that asymptotic optimality implies that the distance of the computed estimates from the intersection of the hyperslabs with the Krylov subspace will tend asymptotically to zero. Moreover, recalling the discussion in Section III-B, these sets contain $\boldsymbol{w}_{\mathrm{WF}}^{(D)}$ with a high probability.

Assumptions 1:

(a) There exists a non-negative integer, say n_0, for which $\Omega = \bigcap_{n \geq n_0} \Omega_n \neq \emptyset$ where $\Omega_n = K_n \cap \Omega'_n$ with $\Omega'_n := \bigcap_{k \in \mathcal{N}} \bigcap_{j \in \mathcal{J}_n} S_{k,j}$. In words, the hyperslabs together with the Krylov subspaces share a non-empty intersection.

(b) There exists n_1 such that $\hat{\boldsymbol{T}}_n = \hat{\boldsymbol{T}}_{n_1}, \forall n \geq n_1$. In other words, after a finite number of iterations, the subspace remains fixed[2].

(c) Let some sufficient small $\varepsilon_1 > 0$ such that $\mu_{k,n} \in (\varepsilon_1 \mathcal{M}_{k,n}, \mathcal{M}_{k,n}(2 - \varepsilon_1))$, $k \in \mathcal{N}$.

(d) Let us define $\mathfrak{C} := \tilde{\boldsymbol{\Omega}} \cap \tilde{\mathcal{O}}$, where the cartesian product space $\tilde{\boldsymbol{\Omega}} := \underbrace{\tilde{\Omega} \times \ldots \times \tilde{\Omega}}_{K}$, $\tilde{\Omega} := \bigcap_{n \geq n_0} \bigcap_{k \in \mathcal{N}} \bigcap_{j \in \mathcal{J}_n} \tilde{S}_{k,j}$ and $\tilde{\mathcal{O}} := \{\underline{\tilde{\boldsymbol{w}}} \in \mathbb{R}^{KD} : \underline{\tilde{\boldsymbol{w}}} = [\tilde{\boldsymbol{w}}^T, \ldots, \tilde{\boldsymbol{w}}^T]^T, \tilde{\boldsymbol{w}} \in \mathbb{R}^D\}$. We assume that $\mathrm{ri}_{\tilde{\mathcal{O}}} \tilde{\boldsymbol{\Omega}} \neq \emptyset$, where this term stands for the relative interior of $\mathfrak{C}$ with respect to $\tilde{\mathcal{O}}$ (see Appendix A).

[2]For a large choice of n_1 the approximations of the quantities used in order to construct the subspace are good and, consequently, this assumption does not lead to performance degradation.

Theorem 1: Under the previously adopted assumptions, the following properties can be proved.

- **Monotonicity**. Under assumptions (a), (b), (c) for the recursion given in (11) it holds that

$$\|\underline{\boldsymbol{w}}_{n+1} - \underline{\hat{\boldsymbol{w}}}_*\| \leq \|\underline{\boldsymbol{w}}_n - \underline{\hat{\boldsymbol{w}}}_*\|, \ \forall n \geq n'_0$$

where $n'_0 = \max\{n_0, n_1\}$, $\underline{\hat{\boldsymbol{w}}} = \begin{bmatrix} \hat{\boldsymbol{w}} \\ \vdots \\ \hat{\boldsymbol{w}} \end{bmatrix} \in \mathbb{R}^{Km}, \forall \hat{\boldsymbol{w}} \in \Omega$ and $\underline{\boldsymbol{w}}_n = \begin{bmatrix} \boldsymbol{w}_{1,n} \\ \vdots \\ \boldsymbol{w}_{K,n} \end{bmatrix}$.

- **Asymptotic Optimality**. If assumptions (a), (b), (c) hold true, we have that

$$\lim_{n \to \infty} \mathrm{d}(\boldsymbol{w}_{k,n+1}, \Omega_n) = 0, \ \forall k \in \mathcal{N},$$

where $\mathrm{d}(\cdot, \Omega_n)$ denotes the distance of a vector from Ω_n. In other words, the distance of the estimates from the intersection set Ω_n, tends asymptotically to zero.

- **Asymptotic Consensus**. Consider that assumptions (a), (b), (c), hold. Then $\lim_{n \to \infty} \|\boldsymbol{w}_{k,n} - \boldsymbol{w}_{l,n}\| = 0, \forall k, l \in \mathcal{N}$.

- **Strong Convergence**. Under assumptions (a), (b), (c), (d), it holds that $\lim_{n \to \infty} \underline{\boldsymbol{w}}_n = \underline{\boldsymbol{w}}_O, \underline{\boldsymbol{w}}_O \in \mathcal{O}$. Moreover, if we define $\underline{\boldsymbol{w}}_O := [\boldsymbol{w}_O^T, \ldots, \boldsymbol{w}_O^T]^T$, it holds that $\boldsymbol{w}_O \in K_{n_1}$. The previous relation yields that the estimates for the whole network converge to a point that lies in the consensus subspace and the estimate at each node converges to a point which lies in the estimated Krylov subspace.

Proof: The proof is provided in Appendix D. ■

VI. Whitening the Input

Recall the discussion in Section II regarding (4). As it was documented there, the performance of the Krylov based reduced rank algorithm is dictated, mainly, by the input statistics. In other words, in cases where the input is highly correlated and, henceforth, the eigenvalue spread of the autocorrelation matrix takes a large value, then the upper bound of the distance between the unknown vector and the one, which is tracked inside the Krylov subspace, is large and as it has been experimentally verified, the performance of the algorithms built around the Krylov subspaces is degraded. This results to an increased error floor in the steady state, as we will see in the Numerical Examples section. Hence, a reasonable strategy, which will be adopted here, is to employ a transformation that "whitens" the input. To this end, at each time instant the input vectors are multiplied with a properly chosen matrix, such that the autocorrelation matrix of the "new" input to be as close as possible to the identity matrix. A first approach could be to employ the celebrated Karhunen Loeve transform in order to produce a transformed input for which the eigenvalue spread of the autocorrelation matrix would be equal to 1. Nevertheless, as it has been also documented in [21], this approach requires a-priori knowledge of the input statistics, which is in general infeasible.

Hence, an alternative route has to be followed. In the non-distributed scenario, the following transformation has been proposed [21]: $\boldsymbol{\psi}_n = \boldsymbol{Z}^{1/2}\boldsymbol{Y}\boldsymbol{u}_n \in \mathbb{R}^m$, where $\boldsymbol{Y}$ is the $m \times m$ Discrete Cosine Transformation (DCT) transformation matrix[3], and $\boldsymbol{Z} = \text{diag}\{(1/\hat{\sigma}_1^2)\dots(1/\hat{\sigma}_m^2)\}$, where $\hat{\sigma}_i$, $i = 1, \dots, m$ is the i-th element in the diagonal of the matrix $E\{\boldsymbol{Y}\boldsymbol{u}_n\boldsymbol{u}_n^T\boldsymbol{Y}^T\}$. The physical reasoning of this transformation can be summarized as follows[4]. The left and right multiplication with the DCT matrix, approximately diagonalizes the matrix $\boldsymbol{R}$ (see also [21]) so as to produce

$$E\left\{\boldsymbol{Y}\boldsymbol{u}_n\boldsymbol{u}_n^T\boldsymbol{Y}^T\right\} \approx \begin{bmatrix} \hat{\sigma}_1^2 & & 0 \\ & \ddots & \\ 0 & & \hat{\sigma}_m^2 \end{bmatrix}. \tag{12}$$

Now, it is not difficult to see that the multiplication with the matrix $\boldsymbol{Z}^{1/2}$, normalizes the diagonal entries of the matrix in (12) so that the resulting autocorrelation matrix approximates the identity matrix. In practice, since the coefficients $\hat{\sigma}_i$, $i = 1, \dots, m$ are unknown, one relies on the following recursive approximation of them: $\hat{\sigma}_{i,n}^2 = \gamma\hat{\sigma}_{i,n-1}^2 + [\boldsymbol{Y}\boldsymbol{u}_n]_i^2$, where $\gamma \in (0, 1]$, and with $[\cdot]_i$ we denote the i-th component of a vector, e.g., [20], [21]. Obviously the performance of the previously mentioned transformation, i.e., how "close" will be the final matrix to the identity one, depends on $\boldsymbol{R}$. However, in practice it has been observed that the previously mentioned transformation results in autocorrelation matrices, which are reasonably close to diagonal.

In the distributed scenario, our goal is to impose a transformation, which is common to all the nodes of the network, and which whitens the autocorelation matrix used for the construction of the subspace, i.e., $\boldsymbol{R}'$. Assuming that the input vectors between any two different nodes of the network are independent and have zero mean, which is usually the case, e.g., [9], [10], we have that $\boldsymbol{R}' = (1/K)E[\boldsymbol{u}_n'\boldsymbol{u}_n'^T]$, where $\boldsymbol{u}_n' = \sum_{k\in\mathcal{N}}\boldsymbol{u}_{k,n}$. The transformed input takes the form ([21]): $\boldsymbol{\psi}_{k,n} = \boldsymbol{Z}'^{(1/2)}\boldsymbol{Y}\boldsymbol{u}_{k,n}$ where $\boldsymbol{Z}' = \text{diag}\{(1/\hat{\sigma}_1'^2)\dots(1/\hat{\sigma}_m'^2)\}$, and σ_i', $i = 1, \dots, m$ is the i-th element of the diagonal of the matrix $\boldsymbol{Y}\boldsymbol{R}'\boldsymbol{Y}^T$. Employing the transformed input in the linear model, we get that $d_{k,n} = \boldsymbol{u}_{k,n}^T\boldsymbol{w}_* + v_{k,n} = \boldsymbol{\psi}_{k,n}^T\boldsymbol{h}_* + v_{k,n}$, where

$$\boldsymbol{h}_* = \boldsymbol{Z}'^{-\frac{1}{2}}\boldsymbol{Y}\boldsymbol{w}_* \Leftrightarrow \boldsymbol{w}_* = \boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{h}_*. \tag{13}$$

It should be pointed out that, by employing a transformed input, the generated estimates do not track the original unknown vector, but the transformed one, i.e., $\boldsymbol{h}_*$. Nevertheless, by multiplying them with the inverse transformation, which is in our case $\boldsymbol{Y}^T\boldsymbol{Z}'^{(1/2)}$, one obtains estimates tracking the original unknown vector (see also [41]).

It is obvious that the definition of the corresponding Krylov subspace changes, since the input changes. Let us define

$$\mathcal{R}' = \frac{1}{K}\sum_{k\in\mathcal{N}}\mathbb{E}\left\{\boldsymbol{\psi}_{k,n}\boldsymbol{\psi}_{k,n}^T\right\}$$

[3]For the DCT transformation matrix holds that $\boldsymbol{Y}\boldsymbol{Y}^T = \boldsymbol{Y}^T\boldsymbol{Y} = \boldsymbol{I}_m$. It should be pointed out that a variety of transformations could be employed, e.g., the Fourier Transformation. However, the DCT one is usually adopted [21].

[4]For a more detailed analysis the reader is referenced to [21].

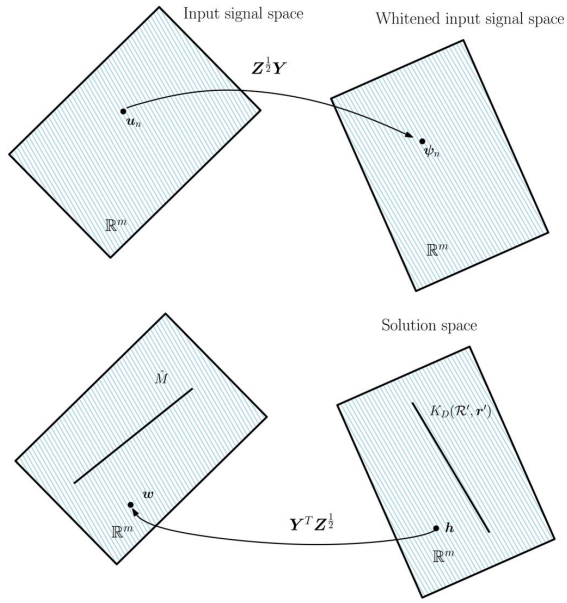


Fig. 5. Illustration of $K_D(\mathcal{R}', \boldsymbol{r}')$, $\hat{M}$ and the connection between points that belong to them.

$$= \frac{1}{K}\sum_{k\in\mathcal{N}}\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{Y}E\left\{\boldsymbol{u}_{k,n}\boldsymbol{u}_{k,n}^T\right\}\boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}$$
$$= \boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{Y}\boldsymbol{R}'\boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}, \tag{14}$$

and $\boldsymbol{r}' = (1/K)\sum_{k\in\mathcal{N}}E\{d_{k,n}\boldsymbol{\psi}_{k,n}\}$. Using a similar rationale as in Section III-B, the algorithm, after employing the transformed input, tracks the following vector

$$\hat{\boldsymbol{h}} = \boldsymbol{T}'(\boldsymbol{T}'^T\mathcal{R}'\boldsymbol{T}')^{-1}\boldsymbol{T}'^T\mathcal{R}'\boldsymbol{h}_* = P_{K_D(\mathcal{R}',\boldsymbol{r}')}^{(\mathcal{R}')}(\boldsymbol{h}_*), \tag{15}$$

where $\boldsymbol{T}'$ is an $m \times D$ matrix whose column form an orthonormal basis of $K_D(\mathcal{R}', \boldsymbol{r}')$. Now, let us shed some light on the connection between the estimates generated exploiting the transformed input, and the estimates, which are produced relying on the original input. If we substitute (13) and (14) into (15) we obtain

$$\hat{\boldsymbol{h}} = \boldsymbol{T}'\left(\boldsymbol{T}'^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{Y}\boldsymbol{R}'\boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{T}'\right)^{-1}\boldsymbol{T}'^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{Y}\boldsymbol{R}'\boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{h}_*$$
$$= \boldsymbol{T}'\left(\boldsymbol{T}'^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{Y}\boldsymbol{R}'\boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{T}'\right)^{-1}\boldsymbol{T}'^T\boldsymbol{Z}'^{\frac{1}{2}}\boldsymbol{Y}\boldsymbol{R}\boldsymbol{w}_*. \tag{16}$$

Notice that $\boldsymbol{S}_{\hat{M}} := \boldsymbol{Y}^T\boldsymbol{Z}'^{(1/2)}\boldsymbol{T}'$ is an $m \times D$ matrix, of rank D ([42]), hence its columns form a basis for a new subspace $\hat{M} := \text{range}\{\boldsymbol{Y}^T\boldsymbol{Z}'^{(1/2)}\boldsymbol{T}'\}$. Fig. 5 illustrates the connection between the points of the two subspaces. Now, according to the previous discussion, if we left multiply (16) by $\boldsymbol{Y}^T\boldsymbol{Z}'^{(1/2)}$, in order to employ the inverse transformation, we get

$$\boldsymbol{Y}^T\boldsymbol{Z}'^{\frac{1}{2}}\hat{\boldsymbol{h}} = \boldsymbol{S}_{\hat{M}}\left(\boldsymbol{S}_{\hat{M}}^T\boldsymbol{R}'\boldsymbol{S}_{\hat{M}}\right)^{-1}\boldsymbol{S}_{\hat{M}}^T\boldsymbol{R}'\boldsymbol{w}_* = \hat{\boldsymbol{w}} = P_{\hat{M}}^{(\boldsymbol{R}')}(\boldsymbol{w}_*). \tag{17}$$

From (17) we conclude that

$$\boldsymbol{Y}^T\boldsymbol{Z}^{\frac{1}{2}}\hat{\boldsymbol{h}} = \hat{\boldsymbol{w}} \Leftrightarrow \hat{\boldsymbol{h}} = \boldsymbol{Z}'^{-\frac{1}{2}}\boldsymbol{Y}\hat{\boldsymbol{w}} \tag{18}$$

Equation (18) establishes the connection among estimates occurring in the case where the input is $\boldsymbol{\psi}_{k,n}, \forall k \in \mathcal{N}, \forall n \in \mathbb{N}$, with the ones produced by the input $\boldsymbol{u}_{k,n}, \forall k \in \mathcal{N}, \forall n \in \mathbb{N}$.

It should be pointed out that, if we employ the whitening transformation, the estimates obtained from the original input (which are produced by employing the inverse transformation), lie in $\hat{M}$, which is also a subspace of dimension equal to D, instead of $K_D(\boldsymbol{R}', \boldsymbol{p}')$, which would be the case if the original input $\boldsymbol{u}_{k,n}$ were employed. Despite the fact that the reduced rank Wiener Filter $\boldsymbol{w}_{WF}^{(D)}$ does not belong to $\hat{M}$, in general, as it will become apparent in the Numerical Examples section, if the input is highly correlated it is better to seek for $\hat{\boldsymbol{w}}$ instead of $\boldsymbol{w}_{WF}^{(D)}$, since the misadjustment between $\boldsymbol{w}_{WF}^{(D)}$ and $\boldsymbol{w}_*$ is large.

Obviously, in order to construct the matrix $\boldsymbol{Z}'$, knowledge on the statistic has to be available. As in the previous section, we rely on estimates of the unknown statistics in order to construct $\boldsymbol{Z}'$. More specifically, the approximated matrix is given by $\hat{\boldsymbol{Z}}_n^{'} = \mathrm{diag}\{(1/\hat{\sigma}_{1,n}'^2)\ldots(1/\hat{\sigma}_{m,n}'^2)\}$, where $\hat{\sigma}_{i,n}'^2 = \gamma\hat{\sigma}_{i,n-1}'^2 + (1/K)\boldsymbol{Y}[\boldsymbol{u}_n']_i^2$, $\gamma \in (0,1]$.

The algorithm is similar to the one developed in the previous section and its mathematical formula is given by

$$\boldsymbol{h}_{k,n+1} = \hat{\boldsymbol{T}}_{n+1}^{'}\hat{\boldsymbol{T}}_n^{'T}\Bigg(\boldsymbol{\varphi}_{k,n} + \mu_{k,n}\left(\sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_{k,j}'\cap\hat{K}_n}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n}\right)\Bigg), \quad (19)$$

with $\boldsymbol{\varphi}_{k,n} = \sum_{l\in\mathcal{N}_k} c_{k,l}(n)\boldsymbol{h}_{l,n}$ and $S_{k,n}' := \{\boldsymbol{w}\in\mathbb{R}^m : |d_{k,n} - \boldsymbol{w}^T\boldsymbol{\psi}_{k,n}| \le \epsilon_k\}$. Furthermore, the $m\times D$ matrix $\hat{\boldsymbol{T}}_n^{'}$ is defined similarly to $\hat{\boldsymbol{T}}_n$, and its columns form an orthonormal basis of $\hat{K}_n := K_D(\hat{\mathcal{R}}_n', \hat{\boldsymbol{r}}_n')$, where $\hat{\mathcal{R}}_n', \hat{\boldsymbol{r}}_n'$, are approximations of the $\mathcal{R}'$ and $\boldsymbol{r}'$ respectively, and they are computed recursively in a similar way as in the previous section.

Recall the assumptions of Theorem 1. In order to derive the convergence analysis of the algorithm in (19), we consider that the assumptions of Theorem 1 hold true, with the following slight modifications:

- The intersection Ω becomes $\hat{\Omega} = \bigcap_{n\ge n_0}\hat{\Omega}_n \neq \emptyset$, where $\hat{\Omega}_n = \hat{K}_n\cap\hat{\Omega}_n'$, and $\hat{\Omega}_n' := \bigcap_{k\in\mathcal{N}}\bigcap_{j\in\mathcal{J}_n} S_{k,j}'$ (Assumption (a')).
- There exists n_1 such that $\hat{\boldsymbol{T}}_n^{'} = \hat{\boldsymbol{T}}_{n_1}^{'}, \forall n\ge n_1$ (Assumption (b')).
- After a finite number of iterations, say n_2, $\boldsymbol{Z}_n' = \boldsymbol{Z}_{n_2}', \forall n \ge n_2$ and, for compact notations, we define $n_0' = \max\{n_0, n_1, n_2\}$ (Assumption (c')).
- The upper bound of the step size equals to $2\mathcal{M}_{k,n}'$, where (see equation at bottom of page). (Assumption (d')).
- The set $\mathfrak{C}$, now becomes $\mathfrak{C}'$, with $\mathfrak{C}' := \hat{\hat{\boldsymbol{\Omega}}}\cap\tilde{\mathcal{O}}$, where $\hat{\hat{\boldsymbol{\Omega}}} := \underbrace{\hat{\hat{\Omega}}\times\ldots\times\hat{\hat{\Omega}}}_{K}$ and $\hat{\hat{\Omega}} := \bigcap_{n\ge n_0}\bigcap_{k\in\mathcal{N}}\bigcap_{j\in\mathcal{J}_n}\tilde{S}_{k,j}'$ employing the modified input (Assumption (e')).

Theorem 2:

- **Monotonicity**: Assume that assumptions $(\mathrm{a}'), (\mathrm{b}'), (\mathrm{c}'), (\mathrm{d}')$, hold true. It holds that
$$\|\underline{\boldsymbol{w}}_{n+1} - \underline{\hat{\boldsymbol{w}}}_*'\|_{\boldsymbol{G}} \le \|\underline{\boldsymbol{w}}_n - \underline{\hat{\boldsymbol{w}}}_*'\|_{\boldsymbol{G}}, \forall n\ge n_0',$$
where $\boldsymbol{G} = \mathrm{diag}\underbrace{\{\boldsymbol{A},\ldots,\boldsymbol{A}\}}_{K}, (Km\times Km)$, $\boldsymbol{A} = \boldsymbol{Y}^T\boldsymbol{Z}_{n_2}^{-1}\boldsymbol{Y}, (m\times m)$, $\underline{\hat{\boldsymbol{w}}}_*' = [\hat{\boldsymbol{w}}_*'^T,\ldots,\hat{\boldsymbol{w}}_*'^T]^T$, where $\hat{\boldsymbol{w}}_*' \in \overline{\Omega}$, $\overline{\Omega} = \bigcap_{n\ge n_0'}\overline{\Omega}_n$, $\overline{\Omega}_n = \overline{M}\cap\Omega_n'$ and $\overline{M} := \mathrm{range}\{\boldsymbol{Y}^T\boldsymbol{Z}_{n_2}'^{(1/2)}\boldsymbol{T}_{n_1}'\}$ is an approximation of $\hat{M}$. The last equation states that the algorithm enjoys monotonicity, *in the $\boldsymbol{G}$ norm sense*.
- **Asymptotic Optimality**: Under Assumptions $(\mathrm{a}'), (\mathrm{b}'), (\mathrm{c}'), (\mathrm{d}')$, it holds that
$$\lim_{n\to\infty}\mathrm{d}(\boldsymbol{w}_{k,n+1}, \overline{\Omega}_n) = 0, \forall k\in\mathcal{N}.$$
- **Strong Convergence to a point that lies in the Consensus subspace**: Consider that $(\mathrm{a}'), (\mathrm{b}'), (\mathrm{c}'), (\mathrm{d}'), (\mathrm{e}')$, hold true it holds that $\lim_{n\to\infty}\underline{\boldsymbol{w}}_n = \underline{\boldsymbol{w}}_O', \underline{\boldsymbol{w}}_O' \in \mathcal{O}$. As in Theorem 1, if we define
- $\underline{\boldsymbol{w}}_O' := [\boldsymbol{w}_O'^T,\ldots,\boldsymbol{w}_O'^T]^T$, it holds that $\boldsymbol{w}_O' \in \overline{M}$. In other words, as in Theorem 1, the estimates for the whole network converge to a point that lies in the consensus subspace and the estimate at each node converges to a point which lies in $\overline{M}$.

Proof: The proof is given in Appendix D. ■

VII. Numerical Examples

In this section, the performance of the proposed algorithms is validated within the system identification framework. To the best of our knowledge, in the literature, there has not been proposed a reduced rank adaptive algorithm, suitable for operation in diffusion networks. To this end, in order to evaluate the performance of the proposed algorithms, we compare it with a modified version of the proposed scheme, denoted as subsampled Adaptive Projected Subgradient Method (sAPSM), where each node, instead of transmitting the whole estimate vector, at every time instant, transmits a subset of D coefficients of it. More specifically, at time instant 1, the first D coefficients are transmitted, at time instant 2, the coefficients $\#D+1,\ldots,\#2D$ and so on. Moreover, the proposed algorithms are compared with

$$\mathcal{M}_{k,n}' = \begin{cases} \dfrac{\sum_{j\in\mathcal{J}_n}\omega_{k,j}\left\|P_{S_{k,j}'\cap\hat{K}_n}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n}\right\|^2}{\left\|\sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_{k,j}'\cap\hat{K}_n}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n}\right\|^2}, & \text{if } \left\|\sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_{k,j}'\cap\hat{K}_n}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n}\right\| \neq 0 \\ 1, & \text{otherwise,} \end{cases}$$

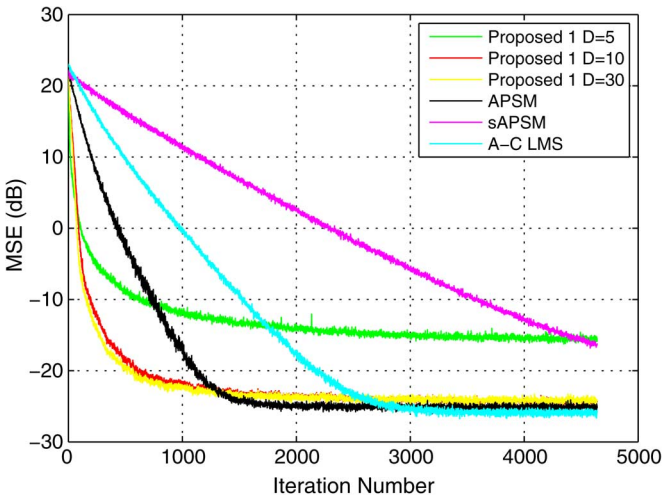


Fig. 6. Average MSE for the first experiment.

TABLE I
STEADY STATE DISTANCES

D	$\|\boldsymbol{w}_{\rm av}-\boldsymbol{w}_{\rm WF}^{(D)}\|^2$	$\|\boldsymbol{w}_{\rm av}-\boldsymbol{w}^*\|^2$
5	$1.21*10^{-4}$	$3.49*10^{-4}$
10	$1.87*10^{-4}$	$1.26*10^{-5}$
20	$2.28*10^{-4}$	$1.23*10^{-5}$
30	$2.40*10^{-4}$	$1.22*10^{-5}$

TABLE II
SQUARED DISTANCE FROM THE CONSENSUS SUBSPACE

D	SDCS
5	$2.6*10^{-4}$
10	$2.1*10^{-4}$
20	$1.9*10^{-4}$
30	$1.9*10^{-4}$

the full rank APSM, i.e., the proposed where the full vector estimate is transmitted and with the diffusion based Adapt-Combine LMS (A-C LMS)[9].

In the first experiment, we consider a diffusion network, in which the number of nodes equals to $K = 20$. The unknown vector is of dimension $m = 160$. We consider that the input samples, $\boldsymbol{u}_n = [u_n, \ldots, u_{n-m+1}]^T$, obey the following model $u_{k,n} = \theta_k u_{k,n-1} + \sqrt{1-\theta_k^2}\chi_{k,n}$, where θ_k is a parameter, which we will alter throughout the experiments, so as to validate the proposed schemes in weakly or strongly correlated environments, and $\chi_{k,n}$ is drawn from the Gaussian distribution with unit variance. The variance of the noise, at each node, equals to $\sigma_k = 0.01 \times \xi_k$, where $\xi_k \in (0.5, 1]$, under the uniform distribution. Furthermore, the combination coefficients are chosen with respect to the Metropolis rule. Finally, the adopted performance metric, which will be used, is the average Mean Square Error (MSE), given by $\text{MSE}(n) = 1/K \sum_{k\in\mathcal{N}}(d_{k,n} - \boldsymbol{u}_{k,n}^T \boldsymbol{w}_{k,n})^2$, and the curves are the result of averaging 100 realizations for smoothing purposes.

The number of hyperslabs used per time update equals to $q = 4$, the step-size is chosen $\mu_{k,n} = 1/2 \times \mathcal{M}_{k,n}$ and the width of the hyperslabs equals to $\epsilon_k = 1.3 \times \sigma_k$. The weights are set $\omega_{k,n} = 1/q$. The step-size in the A-C LMS equals to 3×10^{-3}, so that the algorithm converges to a similar error floor with the full rank APSM. In the first experiment, we study the performance of the proposed scheme (denoted as Proposed 1), with respect to the dimension of the subspace, within which we seek for a solution. For this reason, we consider a weakly correlated environment, so the parameter $\theta_k \in (0, 0.5), \forall k \in \mathcal{N}$, with respect to the Uniform distribution. Moreover, since the unknown vector does not undergo changes, the forgetting factor is chosen $\zeta = 1$. Finally, we assume that $L = 1$, i.e., the subspace is updated at each time instant and for the sAPSM $D = 30$. From Fig. 6 it can be seen that even if the dimension of the subspace takes small values, compared to m, the Proposed 1 performs significantly well. Analytically, the Proposed 1, converges fast and for the specific choices $D = 10$, $D = 30$ the steady state error floor is only slightly increased compared to the full rank APSM and the A-C LMS. If $D = 5$, then the steady state error floor increases significantly. Moreover, the Krylov-based algorithms outperform the sAPSM. Finally, it should be pointed out that the complexity of the LMS is of order $O(m)$ and the complexity of the APSM is of order $O(qm)$.

In Table I we present the steady state Mean Square Deviation, i.e., $\|\boldsymbol{w}_{\rm av} - \boldsymbol{w}^*\|^2$, as well as the distance of the steady state estimate from $\boldsymbol{w}_{\rm WF}^{(D)}$, i.e., $\|\boldsymbol{w}_{\rm av} - \boldsymbol{w}_{\rm WF}^{(D)}\|^2$, where $\boldsymbol{w}_{\rm av} = 1/K \sum_{k\in\mathcal{N}} \boldsymbol{w}_{k,n}$, for a large n. It can be observed, that the smaller the dimension of the Krylov subspace, the smaller the distance of the estimate from $\boldsymbol{w}_{\rm WF}^{(D)}$, whereas the mean square deviation is larger. This is a direct consequence of (4) since, as one can see in this equation, a smaller D leads to a larger upper bound of the distance between $\boldsymbol{w}_{\rm WF}^{(D)}$ and $\boldsymbol{w}_*$. Finally, in Table II, we present the steady state squared distance from the consensus subspace (SDCS), i.e., $\|(\boldsymbol{I}_m - \boldsymbol{B}\boldsymbol{B}^T)\underline{\boldsymbol{w}}_n\|^2$, $n \to \infty$. It can be readily seen, that in the steady state every choice of D leads to a small distance from the consensus subspace.

In the second experiment, Fig. 7, the parameters remain the same as in the previous one. Nevertheless, here we examine the Average Excess Mean Square Error (EMSE) instead of the MSE. The Average EMSE is given by $\text{EMSE} := 1/K \sum_{k\in\mathcal{N}}(\boldsymbol{u}_{k,n}^T \boldsymbol{w}_* - \boldsymbol{u}_{k,n}^T \boldsymbol{w}_{k,n})^2$. From Fig. 7 it can be seen that the full rank LMS and the APSM converge to a lower steady state error floor, compared to the algorithms built around the Krylov subspace rationale. This fact is expected since in the Krylov based algorithms we seek for a vector lying in a subspace of lower dimension, and not the unknown one. However, the Krylov based algorithms converge significantly faster and, moreover, compared to the full rank algorithms, the difference in the steady state error is relatively small.

In the third experiment, we consider that the parameters remain the same as in the previous experiment, albeit a fixed dimension for the subspace, namely $D = 10$, is chosen. Our goal is to study the sensitivity of the algorithm, to the parameter L. To this end, we set different values to L, or in other words, to the frequency with which the subspace is updated. From Fig. 8 it can be readily observed that the smaller the update window, the faster the convergence, due to the fact that for a small window we update the estimate of the subspace more often, and we reach sooner a good approximation of it, compared to the case of a larger window. Moreover, as in the previous experiment, the Krylov-based algorithms outperform the sAPSM. Finally, we should note that the Proposed 1 performs well even for large

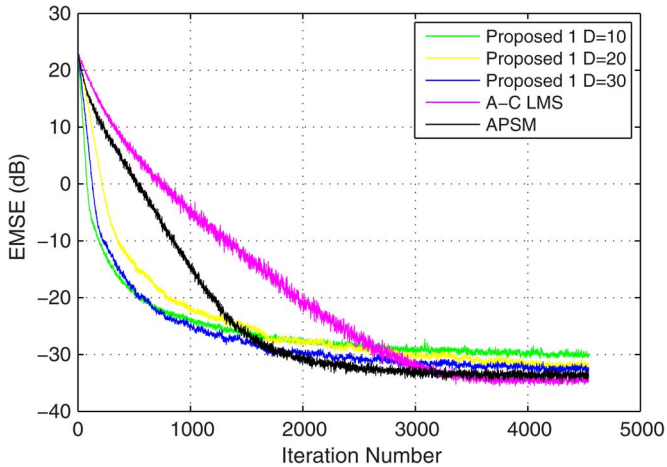


Fig. 7. Average EMSE for the second experiment.

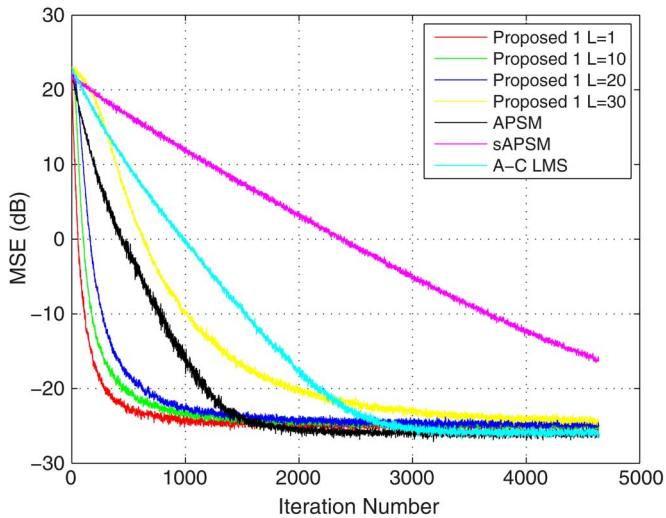


Fig. 8. Average MSE for the third experiment.

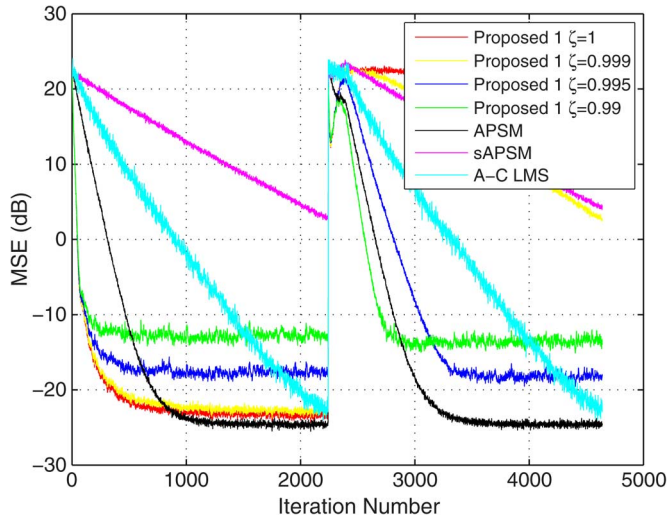


Fig. 9. Average MSE for the fourth experiment.

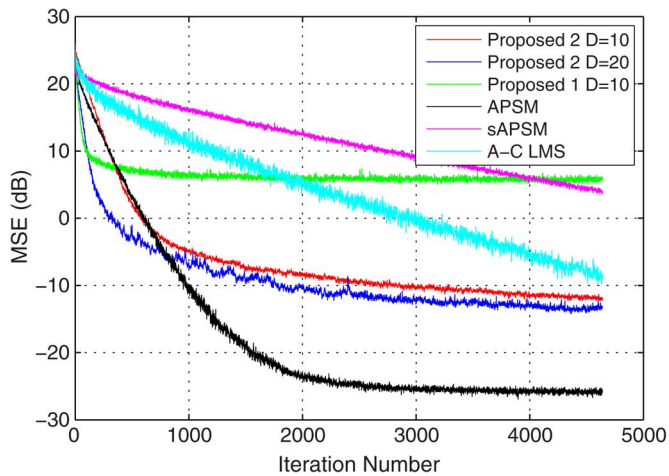


Fig. 10. Average MSE for the fifth experiment.

values of L, which makes it appropriate to be adopted in distributed learning.

In the fourth experiment, we consider a non-stationary environment, since, as it is by now well established, a fast convergence speed does not necessarily coincide with a good tracking ability [28]. To be more specific, we consider that a sudden change in the unknown parameter vector takes place. So, in this experiment, we fix $L = 1$ and $D = 10$ and we alter the forgetting factor. From Fig. 9 it can be seen that until the system undergoes the change, the best performance is achieved for $\zeta = 1$, whereas for smaller ζ the steady state error floor is increased. Nevertheless, if $\zeta = 1$, the algorithm has a long memory of the old statistics, through which the subspace is constructed, that have to change and its tracking ability is not good. On the contrary, the other choices of ζ provide a good tracking ability. Obviously, for large L the tracking ability may be affected, since apart from the forgetting factor, one has to take into consideration the fact that at time instant n the quantities sensed at a past time instant are delivered through the network; this is a direct consequence of the strategy adopted in Section V-A, in order to enhance the information flow. In this case, we consider that the algorithm is able to monitor abrupt changes of the orbit $(\boldsymbol{w}_{k,n})_{n\in\mathbb{Z}_{\geq 0}}$, in order to restart transmitting the input and the desire response. In order to "sense" the previously mentioned abrupt changes, we employ the following metric: $\|\boldsymbol{w}_{k,n+1} - \boldsymbol{w}_{k,n}\|/\|\boldsymbol{w}_{k,n} - \boldsymbol{w}_{k,n-1}\|, \forall k \in \mathcal{N}$, or, more specifically, we restart the transmission of the input coefficients and the desired responses, if this ratio is greater than a threshold, which is chosen, here, to be equal to 10.

In the fifth experiment, we validate the performance of the whitening version (denoted as Proposed 2), in a strongly correlated environment. To this end, the parameter θ_k takes values inside the interval (0.8,1). We compare the Proposed 1 for $D = 10$, the Proposed 2, for the following choices $D = 10$, $D = 20$, the sAPSM, the full rank APSM and the A-C LMS. In the A-C LMS, we choose the largest step-size for which the algorithm converges, and it equals to 10^{-3}. The rest of the parameters remain the same as in the previous experiments, and the forgetting factor which corresponds to the computation of $\hat{\sigma}'_{i,n}$ equals to $\gamma = 1$. Fig. 10 illustrates that the performance of the Proposed 1 is degraded due to the highly correlated input. However, by employing the transformation, which whitens the input (Proposed 2), the performance is significantly enhanced, even if the dimension is relatively low, compared to the case where we employ the original input.

Finally, in the sixth experiment, we examine how the performance of the Proposed 1 is affected when the Krylov subspace

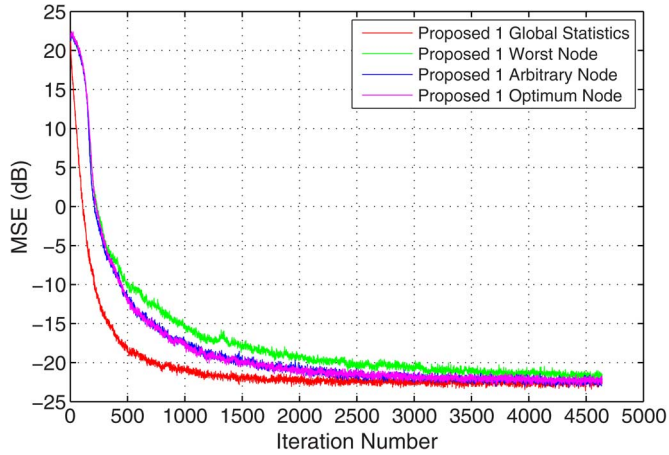


Fig. 11. Average MSE for the sixth experiment.

is constructed based on information coming from a single node (see also Section V-A). To this end, we compare the Proposed 1 in the case where the subspace is constructed using information from every node, with the same algorithm in the cases where: a) the optimum node, b) the worst node and c) an arbitrary node, provide information in order to construct the subspace. D equals to 20 and the rest parameters are the same as in the first experiment. The optimum node is the one with the less correlated input and the worst node is the one with the most correlated input. Fig. 11 shows that by using global information the algorithm converges faster. Nevertheless, the proposed scheme performs well even in the worst case scenario, where the node with the most correlated input is used in order to compute the subspaces. This results is very useful, in large networks, where using global information may be prohibited.

VIII. Conclusions

In this paper, the task of distributed reduced rank adaptive filtering was studied. The algorithms follow the set-theoretic estimation rationale. At each time instant and at each node, a closed convex set, which takes the form of hyperslab, is defined and a possible solution is searched within the intersection of these with a corresponding set of reduced dimension Krylov subspaces. Thus, a significant reduction of the transmitted number of coefficients to the network is achieved, at only small performance degradation. Furthermore, since the performance may degrade when the input samples are highly correlated, a scheme employing a whitening preprocessing has been proposed. Full convergence results are presented, and numerical examples verify the robustness of the proposed algorithms in different scenarios, in the context of the system identification task.

Appendix A
Basic Tools Of Convex Analysis

A set $\mathcal{C} \subseteq \mathbb{R}^m$, for which it holds that $\forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{C}$ and $\forall t \in [0,1]$, $t\boldsymbol{w}_1 + (1-t)\boldsymbol{w}_2 \in \mathcal{C}$, is called convex. A function $\Theta : \mathbb{R}^m \rightarrow \mathbb{R}$ will be called convex if $\forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^m$ and $\forall t \in [0,1]$ it holds that $\Theta(t\boldsymbol{w}_1 + (1-t)\boldsymbol{w}_2) \leq t\Theta(\boldsymbol{w}_1) + (1-t)\Theta(\boldsymbol{w}_2)$. The subdifferential of Θ at an arbitrary point, $\boldsymbol{w}$, is defined as the set of all subgradients of Θ at $\boldsymbol{w}$ ([43], [44]), i.e., $\partial\Theta(\boldsymbol{w}) := \{\boldsymbol{s} \in \mathbb{R}^m : \Theta(\boldsymbol{w}) + \langle \boldsymbol{x} - \boldsymbol{w}, \boldsymbol{s} \rangle \leq \Theta(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^m\}$.

Let us define the distance of an arbitrary point $\boldsymbol{w}$ from a closed non-empty convex set $\mathcal{C}$. It is given by the distance function

$$\begin{aligned} \mathrm{d}(\cdot, \mathcal{C}) : \mathbb{R}^m &\rightarrow [0, +\infty) \\ : \boldsymbol{w} &\mapsto \inf\{\|\boldsymbol{w} - \boldsymbol{x}\| : \boldsymbol{x} \in \mathcal{C}\}, \end{aligned}$$

The projection mapping, $P_{\mathcal{C}}$ onto $\mathcal{C}$, is defined as $P_{\mathcal{C}}(\boldsymbol{w}) := \mathrm{argmin}_{\boldsymbol{x}\in\mathcal{C}}\|\boldsymbol{w} - \boldsymbol{x}\|$, whereas the projection in the $\boldsymbol{W}$-norm sense, where $\boldsymbol{W}$ is an $m \times m$ positive definite matrix, is given via the following optimization $P_{\mathcal{C}}^{(\boldsymbol{W})}(\boldsymbol{w}) := \arg\min_{\boldsymbol{x}\in\mathcal{C}} \|\boldsymbol{w} - \boldsymbol{x}\|_{\boldsymbol{W}}$. The projection operator is related to the distance function, as follows: $d(\boldsymbol{w}, \mathcal{C}) = \|\boldsymbol{w} - P_{\mathcal{C}}(\boldsymbol{w})\|$. Moreover, the projection of a point, say $\boldsymbol{w}$, onto a subspace, say V, is given by $P_V(\boldsymbol{w}) = \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{w}$, where $\boldsymbol{Q}$ is a matrix whose columns form a basis for V, whereas the projection of $\boldsymbol{w}$ onto V in the $\boldsymbol{W}$-norm sense equals to $P_V^{(\boldsymbol{W})}(\boldsymbol{w}) = \boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{Q})\boldsymbol{Q}^T\boldsymbol{W}\boldsymbol{w}$. Finally, the relative interior of a nonempty set, $\mathcal{C}$, with respect to another one, $\mathcal{S}$, is defined as $\mathrm{ri}_{\mathcal{S}}(\mathcal{C}) = \{\boldsymbol{w} \in \mathcal{C} : \exists \varepsilon_0 > 0$ with $\emptyset \neq (B_{(\boldsymbol{w}_0,\varepsilon_0)} \cap \mathcal{S}) \subset \mathcal{C}\}$, where $B_{(\boldsymbol{w}_0,\varepsilon_0)}$ is the open ball defined as $B_{(\boldsymbol{w}_0,\varepsilon_0)} := \{\boldsymbol{w} \in \mathbb{R}^m : \|\boldsymbol{w} - \boldsymbol{w}_0\| < \varepsilon_0\}$, e.g., [45], with center $\boldsymbol{w}_0$ and radius equal to ε_0.

Appendix B
Proof of Claim 1

Proof: By basic linear algebraic arguments [46], it can be verified that the subspace K_n is isomorphic and isometric to $\mathbb{R}^D$ via the mapping $\hat{\boldsymbol{T}}_n : \mathbb{R}^D \rightarrow K_n$ where $\hat{\boldsymbol{T}}_n$ is the matrix whose columns form an orthonormal basis for K_n.

Take the previously mentioned argument into consideration and fix a $\boldsymbol{\phi} \in K_n$. Then notice that

$$\begin{aligned} \|\boldsymbol{\phi} - P_{S_n\cap K_n}(\boldsymbol{\phi})\| &= \min_{\boldsymbol{w}\in S_n\cap K_n} \|\boldsymbol{\phi} - \boldsymbol{w}\| \\ &= \min_{\tilde{\boldsymbol{w}}\in \tilde{S}_n} \|\tilde{\boldsymbol{\phi}} - \tilde{\boldsymbol{w}}\| \\ &= \left\|\tilde{\boldsymbol{\phi}} - P_{\tilde{S}_n}(\tilde{\boldsymbol{\phi}})\right\| \\ &= \left\|\boldsymbol{\phi} - \hat{\boldsymbol{T}}_n P_{\tilde{S}_n}(\tilde{\boldsymbol{\phi}})\right\|, \end{aligned} \tag{20}$$

where $\tilde{S}_n = \{\tilde{\boldsymbol{w}} \in \mathbb{R}^D : |d_n - \boldsymbol{u}^T\hat{\boldsymbol{T}}_n\tilde{\boldsymbol{w}}| \leq \epsilon\}$.

By (21) and the uniqueness of the projection, we obtain

$$P_{S_n\cap K_n}(\boldsymbol{\phi}) = \hat{\boldsymbol{T}}_n P_{\tilde{S}_n}(\tilde{\boldsymbol{\phi}}) = \hat{\boldsymbol{T}}_n P_{\tilde{S}_n}\left(\hat{\boldsymbol{T}}_n^T\boldsymbol{\phi}\right), \tag{21}$$

which completes our proof.

Appendix C
Proof of Claim 2

Recalling the arguments of Claim 1, it can be verified that

$$\tilde{\mathcal{M}}_{k,n} = \mathcal{M}_{k,n} \tag{22}$$

Moreover, it holds that $\boldsymbol{w}_{k,n+1} = \hat{\boldsymbol{T}}_{n+1}\tilde{\boldsymbol{w}}_{k,n+1}$. Going back to (9) and substituting $\tilde{\boldsymbol{\phi}}_{k,n} = \hat{\boldsymbol{T}}_n^T\boldsymbol{\phi}_{k,n}$, $P_{S_{k,n}\cap K_n}(\boldsymbol{\phi}_{k,n}) = \hat{\boldsymbol{T}}_n P_{\tilde{S}_{k,n}}(\tilde{\boldsymbol{\phi}}_{k,n}) \Rightarrow P_{\tilde{S}_{k,n}}(\tilde{\boldsymbol{\phi}}_{k,n}) = \hat{\boldsymbol{T}}_n^T P_{S_{k,n}\cap K_n}(\boldsymbol{\phi}_{k,n})$, and if left multiply with $\hat{\boldsymbol{T}}_{n+1}$ and (23) we obtain the desired result.

Appendix D
Proof of Theorems 1 and 2

We will prove Theorem 2, since Theorem 1 is a special case of it. To be more specific, the properties which will be proved

in this appendix hold also for Theorem 1, if we substitute the matrix $\boldsymbol{Z}'^{-(1/2)}_{n_2}\boldsymbol{Y}$ by $\boldsymbol{I}_m$.

A. Monotonicity

First of all let us define $\underline{\hat{\boldsymbol{h}}_*} = \begin{bmatrix} \hat{\boldsymbol{h}}_* \\ \vdots \\ \hat{\boldsymbol{h}}_* \end{bmatrix} \in \mathbb{R}^{Km}, \forall \hat{\boldsymbol{h}}_* \in \hat{\Omega}$ and $\underline{\boldsymbol{h}}_n = \begin{bmatrix} \boldsymbol{h}_{1,n} \\ \vdots \\ \boldsymbol{h}_{K,n} \end{bmatrix}$. Since $\forall n \geq n_1$, we have that $\hat{\boldsymbol{T}}'_n = \hat{\boldsymbol{T}}_{n+1'} = \hat{\boldsymbol{T}}'_{n_1}$, it holds that (see also [16]) $\hat{\boldsymbol{T}}'_{n+1}\hat{\boldsymbol{T}}'^{T}_{n} = P_{\hat{K}_{n_1}}$. Fix a node, say $k \in \mathcal{N}$. We have that, $\forall n \geq n'_0$

$$\|\boldsymbol{h}_{k,n+1} - \tilde{\boldsymbol{h}}_*\| = \left\| P_{\hat{K}_{n_1}} \left(\boldsymbol{\varphi}_{k,n} + \mu_{k,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S'_{k,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n} \right) \right) - \tilde{\boldsymbol{h}}_* \right\|$$

However, from the definition of $\hat{\Omega}$ and since $\hat{\boldsymbol{h}}_* \in \hat{\Omega} \Rightarrow \hat{\boldsymbol{h}}_* \in \hat{K}_{n_1} \Leftrightarrow \hat{\boldsymbol{h}}_* = P_{\hat{K}_{n_1}}(\hat{\boldsymbol{h}}_*), \forall n \geq n'_0$ by definition. Hence

$$\|\boldsymbol{h}_{k,n+1} - \tilde{\boldsymbol{h}}_*\| = \left\| P_{\hat{K}_{n_1}} \left(\boldsymbol{\varphi}_{k,n} + \mu_{k,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S'_{k,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n} \right) \right) - P_{\hat{K}_{n_1}}(\tilde{\boldsymbol{h}}_*) \right\|. \quad (23)$$

A well known property of the projection operator, e.g., [23], is the non-expansivity, i.e., given a non-empty convex set, say $\mathcal{C}$, $\|P_{\mathcal{C}}(\boldsymbol{w}_1) - P_{\mathcal{C}}(\boldsymbol{w}_2)\| \leq \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|, \forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathbb{R}^m$. Combining the previous with (24) we obtain

$$\|\boldsymbol{h}_{k,n+1} - \tilde{\boldsymbol{h}}_*\| \leq \left\| \boldsymbol{\varphi}_{k,n} + \mu_{k,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S'_{k,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n} \right) - \tilde{\boldsymbol{h}}_* \right\|$$

Gathering the inequalities for every node, we obtain that (see equation at bottom of page). Following similar steps as in [12, Theorem 1] and under assumptions (a')–(d') it can be proved that

$$\left\| \begin{bmatrix} \boldsymbol{\varphi}_{1,n} + \mu_{1,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S'_{1,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{1,n}) - \boldsymbol{\varphi}_{1,n} \right) \\ \vdots \\ \boldsymbol{\varphi}_{K,n} + \mu_{K,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{k,j} P_{S'_{K,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{K,n}) - \boldsymbol{\varphi}_{K,n} \right) \end{bmatrix} - \underline{\tilde{\boldsymbol{h}}_*} \right\| \leq \left\| \begin{bmatrix} \boldsymbol{h}_{1,n} \\ \vdots \\ \boldsymbol{h}_{K,n} \end{bmatrix} - \underline{\tilde{\boldsymbol{h}}_*} \right\| \quad (25)$$

Combining (25), (26) obtain

$$\|\underline{\boldsymbol{h}}_{n+1} - \underline{\tilde{\boldsymbol{h}}}_*\| \leq \|\underline{\boldsymbol{h}}_n - \underline{\tilde{\boldsymbol{h}}}_*\|, \forall n \geq n'_0 \quad (26)$$

From (27) we have

$$\left\| \begin{bmatrix} \boldsymbol{h}_{1,n+1} - \tilde{\boldsymbol{h}}_* \\ \vdots \\ \boldsymbol{h}_{K,n+1} - \tilde{\boldsymbol{h}}_* \end{bmatrix} \right\| \leq \left\| \begin{bmatrix} \boldsymbol{h}_{1,n} - \tilde{\boldsymbol{h}}_* \\ \vdots \\ \boldsymbol{h}_{K,n} - \tilde{\boldsymbol{h}}_* \end{bmatrix} \right\|, \forall n \geq n'_0 \quad (27)$$

However, if we take into consideration (18) and $\forall n \geq n'_0$ $\begin{bmatrix} \boldsymbol{h}_{1,n+1} - (\hat{\boldsymbol{h}}_* \\ \vdots \\ \boldsymbol{h}_{K,n+1} - (\hat{\boldsymbol{h}}_* \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}_{n_2}^{-(1/2)'} \boldsymbol{Y}(\boldsymbol{w}_{1,n+1} - \hat{\boldsymbol{w}}'_*) \\ \vdots \\ \boldsymbol{Z}_{n_2}^{-(1/2)'} \boldsymbol{Y}(\boldsymbol{w}_{K,n+1} - \hat{\boldsymbol{w}}'_*) \end{bmatrix}$. Having as kick off point the previous equation, it is not difficult to obtain that $\forall n \geq n'_0$

$$\left\| \begin{bmatrix} \boldsymbol{h}_{1,n+1} - \tilde{\boldsymbol{h}}_* \\ \vdots \\ \boldsymbol{h}_{K,n+1} - \tilde{\boldsymbol{h}}_* \end{bmatrix} \right\| = \left\| \begin{bmatrix} \boldsymbol{w}_{1,n+1} - \hat{\boldsymbol{w}}'_* \\ \vdots \\ \boldsymbol{w}_{K,n+1} - \hat{\boldsymbol{w}}'_* \end{bmatrix} \right\|_{\boldsymbol{G}}. \quad (28)$$

Since $\boldsymbol{A}$ is a positive definite matrix ([16]), it is not difficult to obtain that $\boldsymbol{G}$ is also positive definite. Let us take a closer look on $\hat{\boldsymbol{w}}'_*$. Since by assumption $\hat{\boldsymbol{h}}_* \in \hat{K}_n \cap S'_{j,n}, \forall k \in \mathcal{N}, \forall j \in \mathcal{J}, \forall n \geq n'_0$ we have that there exists $\tilde{\boldsymbol{h}}_* \in \mathbb{R}^D$ such that $\hat{\boldsymbol{h}}_* = \hat{\boldsymbol{T}}'_{n_1}\tilde{\boldsymbol{h}}_*$ and $|\boldsymbol{\psi}_{k,j}^T \hat{\boldsymbol{h}}_* - d_{k,j}| \leq \epsilon_k, \forall k \in \mathcal{N}, j \in \mathcal{J}, n \geq n'_0$. The previous equations yield that $\hat{\boldsymbol{w}}'_* = \boldsymbol{Y}^T \boldsymbol{Z}_{n_2}^{(1/2)'} \hat{\boldsymbol{T}}'_{n_1} \tilde{\boldsymbol{h}}_* \Rightarrow \hat{\boldsymbol{w}}'_* \in \overline{M}$, and since $\boldsymbol{\psi}_{k,n}^T \hat{\boldsymbol{h}}_* = \boldsymbol{u}_{k,n}^T \hat{\boldsymbol{w}}'_*$, it holds that $\hat{\boldsymbol{w}}'_* \in S_{k,j}, \forall k \in \mathcal{N}, n \geq n'_0$.

Now, combining (28) and (29) implies

$$\left\| \underline{\boldsymbol{w}}_{n+1} - \underline{\hat{\boldsymbol{w}}}'_* \right\|_{\boldsymbol{G}} \leq \left\| \underline{\boldsymbol{w}}_n - \underline{\hat{\boldsymbol{w}}}'_* \right\|_{\boldsymbol{G}}, \forall n \geq n'_0, \quad (29)$$

$$\left\| \begin{bmatrix} \boldsymbol{h}_{1,n+1} \\ \vdots \\ \boldsymbol{h}_{K,n+1} \end{bmatrix} - \underline{\tilde{\boldsymbol{h}}}_* \right\| \leq \left\| \begin{bmatrix} \boldsymbol{\varphi}_{1,n} + \mu_{1,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{1,j} P_{S'_{1,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{1,n}) - \boldsymbol{\varphi}_{1,n} \right) \\ \vdots \\ \boldsymbol{\varphi}_{K,n} + \mu_{K,n} \left(\sum_{j \in \mathcal{J}_n} \omega_{K,j} P_{S'_{K,j} \cap \hat{K}_{n_1}}(\boldsymbol{\varphi}_{K,n}) - \boldsymbol{\varphi}_{K,n} \right) \end{bmatrix} - \underline{\tilde{\boldsymbol{h}}}_* \right\| \quad (24)$$

B. Asymptotic Optimality

Let us define the following non-negative cost function $\forall k \in \mathcal{N}$

$$\Theta_{k,n}(\boldsymbol{h}) = \begin{cases} \frac{1}{L_{k,n}} \sum_{j \in \mathcal{I}_{k,n}} \omega_{k,j} \mathrm{d}\left(\boldsymbol{\varphi}_{k,n}, S'_{k,j} \cap \hat{K}_n\right) \\ \quad \mathrm{d}\left(\boldsymbol{h}, S'_{k,j} \cap \hat{K}_n\right), & \text{if } \mathcal{I}_{k,n} \neq \emptyset \\ 0, & \text{if } \mathcal{I}_{k,n} = \emptyset, \end{cases} \tag{30}$$

where $\mathcal{I} := \{j \in \mathcal{J}\boldsymbol{\varphi}_{k,n} \notin S_{k,j}\}$ and $L_{k,n} = \sum_{j \in \mathcal{J}_n} \omega_{k,j} \mathrm{d}(\boldsymbol{\varphi}_{k,n}, S'_{k,j} \cap \hat{K}_n)$, It can be readily seen that since $\hat{\boldsymbol{h}}_* \in \Omega$, $\Theta_{k,n}(\hat{\boldsymbol{h}}_*) = 0, \forall k \in \mathcal{N}, \forall n \geq n_0$. Following similar steps as in [12] and [34], we have that (11) can be equivalently written (See equation at bottom of page) where $\lambda_{k,n} = (\mu_{k,n}/\mathcal{M}'_{k,n}) \in (\varepsilon_1, 2 - \varepsilon_1)$ and with $\Theta'_{k,n}(\boldsymbol{\varphi}_{k,n})$ we denote the subgradient of $\Theta_{k,n}(\cdot)$ at the point $\boldsymbol{\varphi}_{k,n}$. Following similar steps as in [12], [47] and if assumptions (a′)–(d′) hold true, it can be proved that

$$\lim_{n \to \infty} \Theta_{k,n}(\boldsymbol{\varphi}_{k,n}) = 0, \forall k \in \mathcal{N} \tag{32}$$

which in turn implies that ([12])

$$\lim_{n \to \infty} \mathrm{d}(\boldsymbol{h}_{k,n+1}, \hat{\Omega}_n) = 0, \forall k \in \mathcal{N}. \tag{33}$$

This yields that

$$\lim_{n \to \infty} \left\| \boldsymbol{h}_{k,n+1} - P_{\hat{\Omega}_n}(\boldsymbol{h}_{k,n+1}) \right\| = 0. \tag{34}$$

However, using similar arguments as before with $\hat{\boldsymbol{w}}'_*$, for every point $\overset{\circ}{\boldsymbol{h}}$ that lies in $\hat{\Omega}_n$ it holds that $\overset{\circ}{\boldsymbol{w}} := \boldsymbol{Y} \boldsymbol{Z}_{n_2}^{-(1/2)'} \overset{\circ}{\boldsymbol{h}} \in \overline{\Omega}_n$. Hence, if we define $\breve{\boldsymbol{w}} = \boldsymbol{Y}^T \boldsymbol{Z}_{n_2}^{(1/2)'} P_{\hat{\Omega}_n}(\boldsymbol{h}_{k,n+1})$, we have

$$\begin{aligned} &\left\| \boldsymbol{w}_{k,n+1} - P_{\overline{\Omega}_n}(\boldsymbol{w}_{k,n+1}) \right\| \\ &\quad \leq \left\| \boldsymbol{w}_{k,n+1} - \breve{\boldsymbol{w}} \right\| \\ &\quad = \left\| \boldsymbol{Y}^T \boldsymbol{Z}_{n_2}^{\frac{1}{2}'} \boldsymbol{h}_{k,n+1} - \boldsymbol{Y}^T \boldsymbol{Z}_{n_2}^{\frac{1}{2}'} P_{\hat{\Omega}_n}(\boldsymbol{h}_{k,n+1}) \right\| \\ &\quad \leq \left\| \boldsymbol{Y}^T \boldsymbol{Z}_{n_2}^{\frac{1}{2}'} \right\| \left\| \boldsymbol{h}_{k,n+1} - P_{\hat{\Omega}_n}(\boldsymbol{h}_{k,n+1}) \right\| \end{aligned} \tag{35}$$

where the first inequality holds from the definition of the distance function, as the vectors $\breve{\boldsymbol{w}}$, $P_{\overline{\Omega}_n}(\boldsymbol{w}_{k,n+1}) \in \overline{\Omega}_n$. Taking limits in (36) and recalling (35), we conclude that $\lim_{n \to \infty} \mathrm{d}(\boldsymbol{w}_{k,n+1}, \overline{\Omega}_n) = 0$.

C. Asymptotic Consensus

Under assumptions (a′)–(d′), the algorithmic scheme achieves asymptotic consensus, i.e., [27]

$$\lim_{n \to \infty} \|\boldsymbol{h}_{k,n} - \boldsymbol{h}_{l,n}\| = 0, \forall k, l \in \mathcal{N}.$$

It has been proved [27], that the algorithmic scheme achieves asymptotic consensus, i.e., $\|\boldsymbol{h}_{k,n} - \boldsymbol{h}_{l,n}\| \to 0, n \to \infty, \forall k, l \in \mathcal{N}$ if and only if

$$\lim_{n \to \infty} \|\underline{\boldsymbol{h}}_n - P_{\mathcal{O}}(\underline{\boldsymbol{h}}_n)\| = 0. \tag{36}$$

First of all, notice that $\boldsymbol{h}_{k,n} \in \hat{K}_n, \forall n \in \mathbb{Z}_{\geq 0}$. Since $\boldsymbol{\varphi}_{k,n} = \sum_{l \in \mathcal{N}_k} c_{k,l}(n) \boldsymbol{h}_{l,n}$ is a convex combination of vectors which belong to $\hat{K}_n$, which is a convex set [45], then $\boldsymbol{\varphi}_{k,n} \in \hat{K}_n$. Hence $P_{\hat{K}_{n_1}}(\boldsymbol{\varphi}_{k,n}) = \boldsymbol{\varphi}_{k,n}, \forall k \in \mathcal{N}, \forall n \geq n'_0$. So,

$$\begin{aligned} &\|\boldsymbol{h}_{k,n+1} - \boldsymbol{\varphi}_{k,n}\| \\ &\quad = \left\| P_{\hat{K}_{n_1}} \left(\boldsymbol{\varphi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})}{\left\| \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right\|^2} \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right) \right. \\ &\qquad \left. - \boldsymbol{\varphi}_{k,n} \right\| \\ &\quad = \left\| P_{\hat{K}_{n_1}} \left(\boldsymbol{\varphi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})}{\left\| \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right\|^2} \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right) \right. \\ &\qquad \left. - P_{\hat{K}_{n_1}}(\boldsymbol{\varphi}_{k,n}) \right\| \\ &\quad \leq \left\| \boldsymbol{\varphi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})}{\left\| \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right\|^2} \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) - \boldsymbol{\varphi}_{k,n} \right\| \\ &\quad = \left\| \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})}{\left\| \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right\|^2} \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right\| \to 0, \end{aligned}$$

where in the inequality we have used the nonexpansivity of the projection operator onto a closed convex set and the limit on the last equality holds true as $\|\lambda_{k,n}(\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})/\|\Theta'_{k,n}(\boldsymbol{\varphi}_{k,n})\|^2)\Theta'_{k,n}(\boldsymbol{\varphi}_{k,n})\| \leq (2 - \varepsilon_1)(\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})/\|\Theta'_{k,n}(\boldsymbol{\varphi}_{k,n})\|) \to 0$ which holds from (33) and [47]. Now, it can be readily seen that

$$\lim_{n \to \infty} \|\boldsymbol{h}_{k,n+1} - \boldsymbol{\varphi}_{k,n}\| = 0, \forall k \in \mathcal{N} \tag{37}$$

$$\boldsymbol{h}_{k,n+1} = \begin{cases} P_{\hat{K}_n} \left(\boldsymbol{\varphi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\varphi}_{k,n})}{\left\| \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right\|^2} \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \right), & \text{if } \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) \neq 0, \\ P_{\hat{K}_n}(\boldsymbol{\varphi}_{k,n}), & \text{if } \Theta'_{k,n}(\boldsymbol{\varphi}_{k,n}) = 0, \end{cases} \tag{31}$$

If we generalize (38) for the whole network, we have

$$\lim_{n\to\infty} \|\underline{\boldsymbol{h}}_{n+1} - \boldsymbol{P}_n \underline{\boldsymbol{h}}_n\| = 0. \quad (38)$$

Having as kick off point (39) and if we follow similar steps as in [47] it can be verified that $\lim_{n\to\infty}(\boldsymbol{I}_{Km} - \boldsymbol{B}\boldsymbol{B}^T)\underline{\boldsymbol{h}}_{n+1} = 0$. The previous relation implies that

$$\lim_{n\to\infty} \|\boldsymbol{h}_{k,n} - \boldsymbol{h}_{l,n}\| = 0, \forall k,l \in \mathcal{N}. \quad (39)$$

Hence, $\forall n \geq n_1 \|\boldsymbol{w}_{k,n} - \boldsymbol{w}_{l,n}\| \leq \|\boldsymbol{Y}^T \hat{\boldsymbol{Z}}'^{(1/2)}_{n_1}\| \|\boldsymbol{h}_{k,n} - \boldsymbol{h}_{l,n}\|$. Taking limits and recalling (40) completes our proof.

D. Strong Convergence

Following similar steps as in Claim 2, it can be proved that the algorithm in (19) can be equivalently written:

$$\tilde{\boldsymbol{h}}_{k,n+1} = \tilde{\boldsymbol{\varphi}}_{k,n} + \tilde{\mu}_{k,n}\left(\sum_{j\in\mathcal{J}} \omega_{k,j} P_{\tilde{S}'_{k,j}}(\tilde{\boldsymbol{\varphi}}_{k,n}) - \tilde{\boldsymbol{\varphi}}_{k,n}\right), \quad (40)$$

where $\tilde{S}'_{k,j} := \{\tilde{\boldsymbol{w}} \in \mathbb{R}^D : |d_{k,n} - \boldsymbol{\psi}^T_{k,n} \hat{\boldsymbol{T}}'_n \tilde{\boldsymbol{w}}| \leq \epsilon_k\}$. Notice that the algorithm in (41) is a special case of the algorithm proposed in [12]. The difference is that in the latter, the combined information coming from the nodes of the neighborhood is projected onto a convex set, before the adaptation step. So, the convergence analysis, which took place in [12] holds for the scheme presented in (9). In order to verify this, we have to examine if the assumptions, under which the scheme converges, hold here too. First of all notice that under Assumption (a), $\exists \boldsymbol{h}_0 \in \hat{\Omega}$. From the previous we have that $\exists \tilde{\boldsymbol{h}}_0 \in \mathbb{R}^D$ such that $\tilde{\boldsymbol{h}}_0 = \hat{\boldsymbol{T}}'^T_{n_1} \boldsymbol{h}_0$. Moreover, since $\boldsymbol{h}_0$ belongs to the intersection of the hyperslabs, it satisfies $|d_{k,n} - \boldsymbol{\psi}^T_{k,n}\boldsymbol{h}_0| \leq \epsilon_k, \forall k \in \mathcal{N}, \forall n \geq n'_0$. So, we have that

$$\left|d_{k,j} - \boldsymbol{\psi}^T_{k,j}\boldsymbol{h}_0\right| \leq \epsilon_k \Leftrightarrow \left|d_{k,j} - \boldsymbol{\psi}^T_{k,j}\hat{\boldsymbol{T}}'_{n_1}\tilde{\boldsymbol{h}}_0\right| \leq \epsilon_k, \forall k \in \mathcal{N}, \forall j \in \mathcal{J}, \forall n \geq n'_0. \quad (41)$$

From the previous we have that there exists $\tilde{\boldsymbol{h}}_0 \in \mathbb{R}^D$, such that $\tilde{\boldsymbol{h}}_0 \in \tilde{S}'_{k,j}, \forall k \in \mathcal{N}, \forall j \in \mathcal{J}, \forall n \geq n'_0$. Thus, $\hat{\tilde{\Omega}} \neq \emptyset$. Moreover, $\tilde{\mu}_{k,n} \in (0, 2\tilde{\mathcal{M}}'_{k,n})$. In [12, Theorem 1.1] it has been proved, that these two facts, together with Assumption (d) are the assumptions under which the algorithm converges to a point, i.e.,

$$\lim_{n\to\infty} \underline{\tilde{\boldsymbol{h}}}_n = \underline{\tilde{\boldsymbol{h}}}_O, \quad (42)$$

where $\underline{\tilde{\boldsymbol{h}}}_O := [\tilde{\boldsymbol{h}}^T_O, \ldots, \tilde{\boldsymbol{h}}^T_O]^T \in \tilde{\mathcal{O}}$. Taking into consideration (43) it follows that $\lim_{n\to\infty} \tilde{\boldsymbol{h}}_{k,n} = \tilde{\boldsymbol{h}}_O, \forall k \in \mathcal{N}$. Our proof is complete, since from the previous equation we have that

$$\lim_{n\to\infty} \boldsymbol{h}_{k,n} = \lim_{n\to\infty} \hat{\boldsymbol{T}}'_{n_1} \tilde{\boldsymbol{h}}_{k,n} = \hat{\boldsymbol{T}}_{n_1} \tilde{\boldsymbol{h}}_O.$$

If we write the previous relation for all the nodes of the network we obtain

$$\lim_{n\to\infty} \underline{\boldsymbol{h}}_n = \underline{\boldsymbol{h}}_O,$$

where $\underline{\boldsymbol{h}}_O = [(\hat{\boldsymbol{T}}'_{n_1}\tilde{\boldsymbol{h}}_O)^T, \ldots, (\hat{\boldsymbol{T}}'_{n_1}\tilde{\boldsymbol{h}}_O)^T]^T$. In words, the algorithm converges to a point, which lies in the consensus subspace.

According to the previous discussion we have that $\underline{\boldsymbol{w}}_n = \overline{\boldsymbol{G}}\underline{\boldsymbol{h}}_n, \forall n \geq n'_0$, with $\overline{\boldsymbol{G}} = \operatorname{diag}\underbrace{\{\boldsymbol{Y}^T \boldsymbol{Z}^{(1/2)'}_{n_2}, \ldots, \boldsymbol{Y}^T \boldsymbol{Z}^{(1/2)'}_{n_2}\}}_{K}$. Recall that from Theorem 1, we have $\lim_{n\to\infty} \underline{\boldsymbol{h}}_n = \underline{\boldsymbol{h}}_O$, where $\underline{\boldsymbol{h}}_O \in \mathcal{O}$. Hence,

$$\lim_{n\to\infty} \underline{\boldsymbol{w}}_n = \lim_{n\to\infty} \overline{\boldsymbol{G}}\underline{\boldsymbol{h}}_n = \overline{\boldsymbol{G}}\underline{\boldsymbol{h}}_O = \underline{\boldsymbol{w}}'_O. \quad (43)$$

Since consensus holds for $\underline{\boldsymbol{h}}_O$, i.e., $\underline{\boldsymbol{h}}_O = \begin{bmatrix} \boldsymbol{h}_O \\ \vdots \\ \boldsymbol{h}_O \end{bmatrix}$, it can be readily obtained that $\underline{\boldsymbol{w}}'_O = \begin{bmatrix} \boldsymbol{w}'_O \\ \vdots \\ \boldsymbol{w}'_O \end{bmatrix}$, hence $\underline{\boldsymbol{w}}'_O \in \mathcal{O}$. Finally, since $\boldsymbol{w}'_O = \boldsymbol{Y}^T \boldsymbol{Z}^{(1/2)'}_{n_2} \hat{\boldsymbol{T}}'_{n_1} \tilde{\boldsymbol{h}}_O \Rightarrow \boldsymbol{w}'_O \in \overline{M}$, which finishes our proof.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] F. Cattivelli and A. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[3] D. Blatt and A. Hero, "Energy based sensor network source localization via projection onto convex sets (POCS)," *IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3614–3619, Sep. 2006.

[4] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[5] L. Li, J. Chambers, C. Lopes, and A. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 151–164, Jan. 2009.

[6] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, Jun. 2009.

[7] C. Papadimitriou, *Computational Complexity*. New York, NY, USA: Wiley, 2003.

[8] N. Takahashi, I. Yamada, and A. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.

[9] F. Cattivelli and A. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[10] G. Mateos, I. Schizas, and G. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP J. Adv. Signal Process.*, vol. 2009, 2010, Article ID 981030.

[11] C. Lopes and A. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[12] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[13] E. Msechu, S. Roumeliotis, A. Ribeiro, and G. Giannakis, "Decentralized quantized kalman filtering with scalable communication cost," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3727–3741, Oct. 2008.

[14] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, Jul. 2006.

[15] S. Pereira and A. Pages-Zamora, "Distributed consensus in wireless sensor networks with quantized information exchange," in *Proc. IEEE 9th Workshop Signal Process. Adv. Wireless Commun.. SPAWC*, Jul. 2008, pp. 241–245.

[16] M. Yukawa, R. C. de Lamare, and I. Yamada, "Robust reduced-rank adaptive algorithm based on parallel subgradient projection and Krylov subspace," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4660–4674, Dec. 2009.

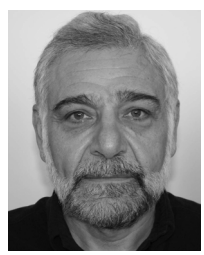
[17] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off communications bandwidth with accuracy in adaptive diffusion networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 2048–2051.
[18] M. Joham and M. D. Zoltowski, "Interpretation of the multi-stage nested wiener filter in the krylov subspace framework," Munich Univ. of Technol., Munich, Germany, Tech. Rep. TUMLNS-TR-00-6, 2000.
[19] A. Kansal, S. Batalama, and D. Pados, "Adaptive maximum sinr rake filtering for ds-cdma multipath fading channels," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 9, pp. 1765–1773, Dec. 1998.
[20] M. Yukawa, "Krylov-proportionate adaptive filtering techniques not limited to sparse systems," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 927–943, Mar. 2009.
[21] A. Sayed, *Fundamentals of Adaptive Filtering*. New York, NY, USA: Wiley, 2003.
[22] P. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, no. 2, pp. 182–208, Feb. 1993.
[23] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Funct. Anal. Optimiz.*, vol. 25, no. 7&8, pp. 593–617, 2004.
[24] K. Slavakis, I. Yamada, and N. Ogura, "The adaptive projected subgradient method over the fixed point set of strongly attracting nonexpansive mappings," *Numer. Funct. Anal. Optimiz.*, vol. 7, no. 8, pp. 905–930, 2006.
[25] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 97–123, Feb. 2011.
[26] K. Slavakis and I. Yamada, "The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning," *SIAM J. Optimiz.* 2011 [Online]. Available: http://arxiv.org/abs/1008.5231, accepted for publication
[27] R. Cavalcante, I. Yamada, and B. Mulgrew, "An adaptive projected subgradient approach to learning in diffusion networks," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2762–2774, Jul. 2009.
[28] S. Haykin, *Adaptive filter theory*. Delhi, India: Pearson Education, 2008.
[29] M. Honig and J. Goldstein, "Adaptive reduced-rank interference suppression based on the multistage wiener filter," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 986–994, Jun. 2002.
[30] J. Goldstein, I. Reed, and L. Scharf, "A multistage representation of the wiener filter based on orthogonal projections," *IEEE Trans. Inf. Theory*, vol. IT-44, no. 7, pp. 2943–2959, Nov. 1980.
[31] J. Goldstein and I. Reed, "Reduced-rank adaptive filtering," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 492–496, Mar. 1997.
[32] L. Gubin, B. Polyak, and E. Raik, "The method of projections for finding the common point of convex sets* 1," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 6, pp. 1–24, 1967.
[33] L. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, 1967.
[34] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.
[35] P. Huber and E. Ronchetti, *Robust Statistics*. New York, NY, USA: Wiley, 2009.
[36] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. : Academic Press, 2009.
[37] G. Mateos, I. Schizas, and G. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, Nov. 2009.
[38] M. Gerla and J. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Netw.*, vol. 1, no. 3, pp. 255–265, 1995.
[39] F. Cattivelli and A. Sayed, "Hierarchical diffusion algorithms for distributed estimation," in *IEEE Workshop Statist. Signal Process.*, 2009, pp. 537–540.
[40] I. Yamada, K. Slavakis, and K. Yamada, *An Efficient Robust Adaptive Filtering Scheme Based on Parallel Subgradient Projection Techniques*, vol. 6, pp. 3725–3728, 2001, unpublished.
[41] P. Diniz, *Adaptive Filtering: ALGORITHms and Practical Implementation*. Berlin, Germany: Springer-Verlag, 2008.
[42] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[43] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA: Athena Scientific, 2003.
[44] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms: Fundamentals*. New York, NY, USA: Springer, 1993.
[45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[46] G. Strang, *Introduction to Linear Algebra*. Cambridge, U.K.: Wellesley Cambridge Press, 2003.
[47] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

Symeon Chouvardas (S'11) received the B.Sc. degree from the Department of Informatics and Telecommunications, University of Athens, Athens, Greece, in 2008. He is currently pursuing the Ph.D. degree in the Department of Informatics and Telecommunications, University of Athens. His research interests are in adaptive theory, machine learning and distributed learning

Konstantinos Slavakis (M'08–SM'12) received the M.E. and Ph.D. degrees in electrical and electronic engineering from the Tokyo Institute of Technology (TokyoTech), Tokyo, Japan, in 1999 and 2002, respectively.

From 2004 to 2006, he was with TokyoTech as a Japan Society for the Promotion of Science Post-Doctoral Fellow and from 2006 to 2007, he was a Post-Doctoral Fellow with the Department of Informatics and Telecommunications, University of Athens, Athens, Greece, under the ENTER Program. Since September 2007, he has been an Assistant Professor with the Department of Telecommunications Science and Technology, University of Peloponnese, Tripolis, Greece. His current research interests include applications of convex analysis and computational algebraic geometry to signal processing, machine learning, arrays, and multidimensional systems problems.

Dr. Slavakis serves as an Associate and Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He received the Tejima Memorial Award of TokyoTech for his Ph.D. dissertation. He was a recipient of the Japanese Government (MEXT) Scholarship from 1996 to 2002.



Sergios Theodoridis (F'10) is currently Professor of Signal Processing and Communications in the Department of Informatics and Telecommunications of the University of Athens. His research interests lie in the areas of Adaptive Algorithms and Communications, Machine Learning and Pattern Recognition, Signal Processing for Audio Processing and Retrieval. He is the co-editor of the book "Efficient Algorithms for Signal Processing and System Identification," Prentice Hall 1993, the co-author of the best selling book "Pattern Recognition," Academic Press, 4th ed. 2008, the co-author of the book "Introduction to Pattern Recognition: A MATLAB Approach," Academic Press, 2009, and the co-author of three books in Greek, two of them for the Greek Open University.

He is the co-author of six papers that have received best paper awards including the 2009 IEEE Computational Intelligence Society Transactions on Neural Networks Outstanding paper Award. He has served as an IEEE Signal Processing Society Distinguished Lecturer.

He was the general chairman of EUSIPCO-98, the Technical Program co-chair for ISCAS-2006 and co-chairman and co-founder of CIP-2008 and co-chairman of CIP-2010. He has served as President of the European Association for Signal Processing (EURASIP) and as member of the Board of Governors for the IEEE CAS Society. He currently serves as member of the Board of Governors (Member-at-Large) of the IEEE SP Society.

He has served as a member of the Greek National Council for Research and Technology and he was Chairman of the SP advisory committee for the Edinburgh Research Partnership (ERP). He has served as vice chairman of the Greek Pedagogical Institute and he was for four years member of the Board of Directors of COSMOTE (the Greek mobile phone operating company). He is Fellow of IET, a Corresponding Fellow of RSE, a Fellow of EURASIP and a Fellow of IEEE.