A Sparsity Promoting Adaptive Algorithm for Distributed Learning

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

***Abstract*—In this paper, a sparsity promoting adaptive algorithm for distributed learning in diffusion networks is developed. The algorithm follows the set-theoretic estimation rationale. At each time instance and at each node of the network, a closed convex set, known as property set, is constructed based on the received measurements; this defines the region in which the solution is searched for. In this paper, the property sets take the form of hyperslabs. The goal is to find a point that belongs to the intersection of these hyperslabs. To this end, sparsity encouraging variable metric projections onto the hyperslabs have been adopted. In addition, sparsity is also imposed by employing variable metric projections onto weighted ℓ_1 balls. A combine adapt cooperation strategy is adopted. Under some mild assumptions, the scheme enjoys monotonicity, asymptotic optimality and strong convergence to a point that lies in the consensus subspace. Finally, numerical examples verify the validity of the proposed scheme compared to other algorithms, which have been developed in the context of sparse adaptive learning.**

***Index Terms*—Adaptive distributed learning, diffusion networks, projections, sparsity.**

I. Introduction

SPARSITY, i.e., the presence of a small number of nonzero coefficients in an unknown signal/parameter vector, which is to be estimated, has been recently attracting an overwhelming interest under the Compressed Sensing (CS) framework [1], [2]. Such techniques have been applied in a wide range of applications such as data compression, echo cancellation, spectrum cartography, medical signal processing, to name but a few, e.g., [3]–[5]. However, most of the efforts, so far, have been invested in developing algorithms which are appropriate for batch mode of operation. Accordingly, the estimation of the signal parameters can be achieved only after a fixed number of measurements has been collected and stored. If a new measurement becomes available, the whole estimation process has to be repeated from scratch. As the number of measurements increases, the computational burden becomes prohibitive for real time applications. On the contrary, time-adaptive/online updating improves the current estimate dynamically as new measurements are obtained. Moreover, batch methods are not directly suited for time varying scenarios, where the parameter vector changes, as time evolves. Online learning techniques overcome the previously mentioned limitations. Online techniques for sparsity-aware learning have recently become the focus of intense research activity, e.g., [6]–[8].

In this paper, the task of sparsity-aware learning is treated in the context of distributed processing [9]–[12]. To be more specific, we consider the typical setup of a wireless sensor network, in which the estimate of the unknown parameter vector is based on noisy measurements sensed by a number of spatially distributed nodes. This task can be fulfilled following several approaches. In the so called centralized scenario, the nodes transmit the measured information to a central node, called fusion center, which carries out the full amount of computations. Nevertheless, the existence of a fusion center is not always feasible due to power or geographical constraints. Furthermore, this approach lacks robustness, since if the fusion center is malfunctioning then the network collapses. Hence, in many applications, a decentralized philosophy has to be followed, in which the nodes themselves take part in the computation task. The most celebrated examples of such networks are:

- The incremental one, in which each node is able to communicate with only one neighbouring node and, henceforth, the nodes are part of a cyclic pattern, e.g., [13], [14]. This topology requires small bandwidth, albeit it is not robust to cope with malfunctioning nodes; once a node fails, the network collapses.
- The diffusion, where each node shares information with a subset of nodes. Despite the fact that the diffusion topology requires larger bandwidth, compared to the incremental one, it is robust to cope with node failures, and its implementation turns out to be easier, compared to the implementation of an incremental one, when large networks are involved [9], [10], [12], [15].

Although there are a few sparsity promoting methods for batch processing in distributed learning, e.g., [3], [16], to the best of our knowledge there is no algorithm, yet, capable for time-adaptive/online processing to operate in diffusion networks. The algorithm to be presented here addresses both needs: a) that of the sparsity promotion and b) that of operation in diffusion networks. Its development follows the set-theoretic estimation rationale [17]; that is, instead of seeking for a (unique) optimum

Manuscript received November 08, 2011; revised March 17, 2012 and May 28, 2012; accepted May 30, 2012. Date of publication June 15, 2012; date of current version September 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Huaiyu Dai. This work was supported by the European Union (European Social Fund—ESF) and by the 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.

S. Chouvardas, Y. Kopsinis, and S. Theodoridis are with the Department of Informatics and Telecommunications, University of Athens, Panepistimioupolis, Ilissia, 157 84 Athens, Greece (e-mail: schouv@di.uoa.gr; kospinis@ieee.org; 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/TSP.2012.2204987

vector, we search for a set of points that are *in agreement* with the received set of measurements. To this end, at each time instance, a closed convex set is defined by the currently received input-output training data pair, and *any* point that lies within this set is considered to be in agreement with the current measurements. In this paper, these convex sets are chosen to be hyperslabs. Moreover, following similar philosophy as in [6], in order to exploit the *a-priori* knowledge concerning the sparsity of the unknown vector, we constrain the search for a solution within sparsity-promoting weighted ℓ_1 balls. Thus, the goal becomes that of finding a point that lies in the intersection of the *infinite* number of hyperslabs with the previously mentioned constraint sets; this is successfully solved (see for example [18]–[20]) by employing a sequence of projections onto the hyperslabs and the weighted ℓ_1 balls. In the current study, the previous scheme is enhanced with respect to its sparsity promoting potential; this is achieved by reformulating the projection operators appropriately so as to exploit further the *a-priori* information concerning the sparsity of the unknown vector. To this end, the *variable metric projections* rationale is adopted, e.g., [21]. As a consequence, the variable metric projections improve the convergence speed, when seeking for a sparse vector, since different weights are assigned at each coefficient of the updated vector, and through this procedure, small coefficients are forced to diminish faster. The reasoning of assigning different weights at each coefficient is also met in the so called proportionate algorithms [22], [23]. The main contributions of this paper are the following.

- An adaptive distributed algorithm suitable for estimating sparse unknown vectors in diffusion networks is developed. As we have already mentioned, this is the first time that a sparsity promoting distributed adaptive algorithm is presented.
- The algorithm builds upon variable metric projections and the variable metric projection on a weighted ℓ_1 ball is derived.
- The previous choice of projections leads to a time varying induced inner product and at the same time the constraints on the weighted ℓ_1 balls are time varying as well. Convergence for such a task is also provided, for a first time. Moreover, since our focus is on diffusion learning strategies, consensus is also established .

The paper is organized as follows. In Section II the general problem is described and in the next section, adaptive strategies for estimating sparse signals are provided. In Section IV, we shed light on basic concepts regarding adaptive distributed learning and in Section V the proposed algorithm, together with its theoretical analysis, is discussed. Finally, in Section VI the performance of the proposed algorithm is validated and in the Appendices the theoretical background is discussed, and full proofs of the theorems are given.

The notation which will be used throughout the paper is the following. The set of all real numbers and the set of all nonnegative integers are denoted by $\mathbb{R}$ and $\mathbb{Z}_{\geq 0}$, respectively. Given two integers j_1, j_2, with $j_1 \leq j_2$, we define $\overline{j_1, j_2} = \{j_1, \ldots, j_2\}$. The stage of discussion will be the Euclidean space $\mathbb{R}^m$, where m is a positive integer. We denote vectors by boldface letters, e.g., $\boldsymbol{h}$, and matrices with upper-case boldfaced letters. Furthermore, we define the weighted inner product as follows: $\forall \boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^m$, $\langle \boldsymbol{h}_1, \boldsymbol{h}_2 \rangle_{\boldsymbol{V}} := \boldsymbol{h}_1^T \boldsymbol{V} \boldsymbol{h}_2$, and the weighted norm $\forall \boldsymbol{h} \in \mathbb{R}^m$, $\|\boldsymbol{h}\|_{\boldsymbol{V}} = \sqrt{\langle \boldsymbol{h}, \boldsymbol{h} \rangle_{\boldsymbol{V}}}$, where the $m \times m$ matrix, $\boldsymbol{V}$, is positive definite, and the notation $(\cdot)^T$ stands for the transposition operator. The Euclidean norm, i.e., $\|\cdot\|$, is a special case of the previously mentioned norm, and occurs if $\boldsymbol{V} = \boldsymbol{I}_m$, where $\boldsymbol{I}_m$ is the $m \times m$ identity matrix. Moreover, the 2-norm of a matrix, say $\boldsymbol{A}$, is denoted by $\|\boldsymbol{A}\|$. Given a vector $\boldsymbol{h} = [h_1, \ldots, h_m]^T \in \mathbb{R}^m$, the ℓ_1 norm is defined $\|\boldsymbol{h}\|_1 := \sum_{i=1}^m |h_i|$, and the support set, $\mathrm{supp}(\boldsymbol{h}) := \{i \in \overline{1, m} : h_i \neq 0\}$. Finally, the ℓ_0 "norm" is the cardinality of the support set, i.e., $\|\boldsymbol{h}\|_0 := |\mathrm{supp}(\boldsymbol{h})|$, where given a set, say $\mathcal{S}$, the notation $|\mathcal{S}|$ stands for its cardinality.

II. Problem Formulation

Consider the problem of estimating an unknown parameter vector $\boldsymbol{h}_* \in \mathbb{R}^m$, exploiting measurements $(d_n, \boldsymbol{u}_n)_{n \in \mathbb{Z}_{\geq 0}} \in \mathbb{R} \times \mathbb{R}^m$, which are related via the linear system

$$d_n = \boldsymbol{u}_n^T \boldsymbol{h}_* + v_n, \ \forall n \in \mathbb{Z}_{\geq 0} \tag{1}$$

where v_n is the noise process and $n \in \mathbb{Z}_{\geq 0}$ denotes the (discrete) time index. We assume that $\boldsymbol{h}_*$ is sparse, i.e., $\|\boldsymbol{h}_*\|_0 \ll m$, or, in other words, it has a few number of nonzero coefficients. Suppose that a finite number of measurements, say N, is available. In that case, (1) can be written as

$$\boldsymbol{d} = \boldsymbol{U} \boldsymbol{h}_* + \boldsymbol{v},$$

where the regression matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]^T \in \mathbb{R}^{N \times m}$, $\boldsymbol{d} = [d_1, \ldots, d_N]^T \in \mathbb{R}^N$, $\boldsymbol{v} = [v_1, \ldots, v_N]^T \in \mathbb{R}^N$. When $N < m$, classical techniques, as for example the celebrated least-squares method, fail to obtain a good estimate of the unknown parameters, since the sparsity of $\boldsymbol{h}_*$ is not taken into consideration and, consequently, there is no guarantee that the estimate will predict the support, i.e., the set of nonzero components, while forcing the rest to become zero. This results to an increased misadjustment between the true and the estimated values, [24]. Nevertheless, one can resort to a sparsity promoting technique, namely Least Absolute Shrinkage and Selection Operator (Lasso), and overstep the previously mentioned problem. Analytically, the Lasso estimator promotes sparsity by solving the following optimization task

$$\hat{\boldsymbol{h}} = \mathrm{argmin}_{\|\boldsymbol{h}\|_1 \leq \delta} \|\boldsymbol{d} - \boldsymbol{U}\boldsymbol{h}\|^2,$$

where the term $\|\boldsymbol{d} - \boldsymbol{U}\boldsymbol{h}\|$ accounts for the error residual in the estimation process; the ℓ_1 norm promotes sparsity by shrinking small coefficient values towards zero, e.g., [4]. Most of the emphasis in solving the Lasso problem has been given on batch techniques, see, e.g., [25]. However, such techniques are inappropriate for online learning, where data arrive sequentially and/or the environment is not stationary but it undergoes changes as time evolves.

III. Sparsity Promoting Adaptive Algorithms

Although sparsity promoting adaptive algorithms have drawn the attention of the signal processing community for many years, see, e.g., [22], [23], it is only recently that the topic is being treated in a more theoretically sound framework,

within the spirit of ℓ_1 regularization, e.g., [6]–[8], [26], [27]. The *a-priori* information concerning the underlying sparsity is exploited by constraining the ℓ_1 norm. Providing this *a-priori* information, the convergence rate is improved significantly, and the associated error floor in the steady state is also reduced.

As it is often the case, most of these efforts evolve along the three main axes in adaptive filtering. One is along the gradient descend rationale, as this is represented in the adaptive learning by the LMS [7], [27]. The other direction follows Newton-type arguments, as represented by the RLS [8]. The other route is more recent and builds upon recent extensions of the classical Projections Onto Convex Sets (POCS) theory, which allow for applications in the online time-adaptive setting, e.g., [18]–[20], [28]. Our new algorithm belongs to this last category and it exploits its potential to allow for convex constraints to be efficiently incorporated within the algorithmic flow.

A. Set-Theoretic Estimation Approach and Variable Metric Projections

In this paper, the set-theoretic estimation rationale e.g., [17], [20], [29], will be adopted. The philosophy behind this family of algorithms is that instead of adopting a loss function to be optimized, one obtains an estimate that lies in the intersection of an infinite number of convex sets. Each one of these (convex) property sets is constructed using information which is provided by the respective measurement pair $(d_n, \boldsymbol{u}_n)$; based on this information as well as that of the noise source, a convex set/region is constructed where the unknown vector lies with a high probability. We say that such a convex set is "in agreement" with the received measurement pair. Moreover, when convex constraints are present, each one of them defines a convex region and the solution is searched in the intersection of all the involved sets; those associated with the measurements as well as those with the constraints.

A strategy to find a point that lies in the intersection of all these convex sets was developed in [18]. This algorithmic scheme can be seen as a generalization of the POCS theory [17], [30], [31]. The difference lies in the fact that in the classical POCS theory a finite number of convex sets is involved. On the contrary, in its adaptive version, an infinite number of sets are involved. In the adaptive setting, the task of identifying a point in the intersection of convex sets is accomplished by projecting, in parallel, the currently available estimate over the q (user-defined) most recently "received" sets. This provides the new estimate. If convex constraints are present, e.g., [19], further projections are performed, one for each of the constraint sets (the definition of the projection is given in Appendix A). Under some mild assumptions, the estimates converge to a point that lies in the intersection of all the involved convex sets.

It has been pointed out (see, for example, [21]), that the sparsity-related *a-priori* knowledge can be "embedded" in the projection operators to the benefit of the algorithm's performance. To this end, the notion of the *variable metric projection* was introduced. The result of a variable metric projection of a vector onto a closed convex set (see also Appendix A) is determined by: a) a positive definite matrix, which defines the induced inner product; b) the specific convex set, onto which the projection takes place; and c) the vector itself. As it will become clear later on, for a properly chosen matrix, which is time-dependent and it is constructed via the current estimate at each time instance, the variable metric projection pushes small coefficients to diminish faster. In other words, by employing at each time instance a different inner product in our Euclidean space, we manage to change the topology of the space in order to favor sparse solution vectors.

In the current paper, the adopted property sets, which define our solution set take the form of hyperslabs, i.e.,

$$S_n := \{\boldsymbol{h} \in \mathbb{R}^m : |d_n - \boldsymbol{u}_n^T \boldsymbol{h}| \leq \epsilon\} \tag{2}$$

where $\epsilon \geq 0$ is a user-defined parameter. The parameter ϵ serves as a threshold and it takes into consideration the noise, as well as possible inaccuracies in the adopted model. In this setting, any point that lies within this hyperslab is considered to be in agreement with the current measurement. The choice of a hyperslab is in line with criteria that have been proposed in the context of the robust statistics rationale, e.g., [20], [32]. The variable metric projection onto the respective hyperslabs is defined as [33]:

$$\forall \boldsymbol{h} \in \mathbb{R}^m, \quad P_{S_n}^{(\boldsymbol{G}_n)}(\boldsymbol{h}) := \boldsymbol{h} + \beta_n \boldsymbol{G}_n^{-1} \boldsymbol{u}_n \tag{3}$$

where

$$\beta_n = \begin{cases} \frac{d_n - \boldsymbol{u}_n^T \boldsymbol{h} + \epsilon}{\|\boldsymbol{u}_n\|^2_{\boldsymbol{G}_n^{-1}}}, & \text{if } d_n - \boldsymbol{u}_n^T \boldsymbol{h} < -\epsilon, \\ 0, & \text{if } |d_n - \boldsymbol{u}_n^T \boldsymbol{h}| \leq \epsilon, \\ \frac{d_n - \boldsymbol{u}_n^T \boldsymbol{h} - \epsilon}{\|\boldsymbol{u}_n\|^2_{\boldsymbol{G}_n^{-1}}}, & \text{if } d_n - \boldsymbol{u}_n^T \boldsymbol{h} > \epsilon. \end{cases}$$

Note that if $\boldsymbol{G}_n = \boldsymbol{I}_m$, then (3) is the standard metric projection onto a hyperslab. The positive definite diagonal matrix $\boldsymbol{G}_n^{-1}$ is constructed following similar philosophy as in [21], [23]. The ith coefficient of its diagonal equals to $g_{i,n}^{-1} = \frac{1-\alpha}{m} + \alpha \frac{|h_i^{(n)}|}{\|\boldsymbol{h}_n\|_1}$, where $\alpha \in [0, 1)$ is a parameter, that determines to which extend the sparsity level of the unknown vector will be taken into consideration, and $h_i^{(n)}$ denotes the ith component of $\boldsymbol{h}_n$. Now, in order to grasp the reasoning of the variable metric projections, consider the ideal situation, in which $\boldsymbol{G}_n^{-1}$ is generated by the unknown vector $\boldsymbol{h}_*$. It is easy to verify that $g_{i,n}^{-1} > g_{i',n}^{-1}$, if $i \in \text{supp}(\boldsymbol{h}_*)$, and $i' \notin \text{supp}(\boldsymbol{h}_*)$. Hence, employing the variable metric projection, the amplitude of each coefficient of the vector used to construct $\boldsymbol{G}_n^{-1}$ determines the weight that will be assigned to the corresponding coefficient of the second term of the right hand side in (3). That is, components with smaller magnitude are multiplied with small coefficients of $\boldsymbol{G}_n^{-1}$. Loosely speaking, the variable metric projections accelerate the convergence speed when tracking a sparse vector, since by assigning different weights pushes the coefficients of the estimates with small amplitude to diminish faster. The geometric implication of it is that the projection is made to "lean" towards the direction of the more significant components of the currently available estimate. Another viewpoint, as documented in [23], is the following. The coefficients of the matrix $\boldsymbol{G}_n^{-1}$, which are multiplied with the second term of the right hand side in (3), can be seen as m individual step-sizes, one for each coefficient. As it is by now well established in adaptive filter community (see for example [34]), the larger the step-size the faster the convergence. Hence, coefficients of large amplitude are assigned a large "step-size", whereas the rest are multiplied by a smaller

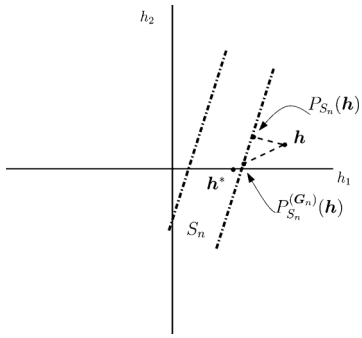


Fig. 1. Illustration of a hyperslab, the standard metric projection of a vector $\boldsymbol{h}$ onto it, denoted by $P_{S_n}(\boldsymbol{h})$, and the variable metric projection onto it.

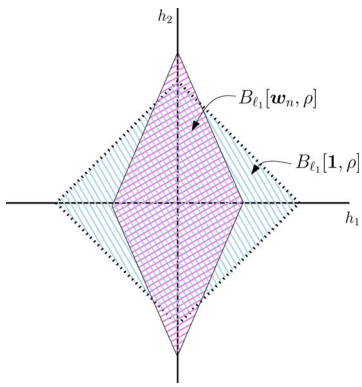


Fig. 2. Illustration of a weighted ℓ_1 ball (solid line magenta) and an unweighted ℓ_1 ball (dashed line blue).

"step-size". This results to a faster convergence speed compared to the case where the same step-size is adopted to each one of coefficients; the latter case results in $\boldsymbol{G}_n = \boldsymbol{I}_m$, as it has been shown in [22], [23]. Obviously, since $\boldsymbol{h}_*$ is unknown, the weights rely on the available estimates, i.e., $\boldsymbol{h}_n$, at each time instance. These concepts are depicted in Fig. 1.

Remark 1: The variable metric projections rationale is in line with the so called *proportionate* algorithms [22], [23], [35]. At the heart of these algorithms lies the fact that at every time instance different weights are assigned to the coordinates of the vector, which produces the next estimate. ■

In this paper, we go one step further, as far as sparsity is concerned. In a second stage, additional sparsity-related constraints, which are built around the weighted ℓ_1 ball, are employed, [2]. A sparsity promoting adaptive scheme, based on set-theoretic estimation arguments, in which the constraints are weighted ℓ_1 balls, was presented in [6]. Given a vector of weights $\boldsymbol{w}_n = [w_1^{(n)}, \ldots, w_m^{(n)}]^T$, where $w_i^{(n)} > 0, \forall i = 1, \ldots, m$, and a positive radius, ρ, the weighted ℓ_1 ball is defined as: $B_{\ell_1}[\boldsymbol{w}_n, \rho] := \{\boldsymbol{h} \in \mathbb{R}^m : \sum_{i=1}^m w_i^{(n)}|h_i| \leq \rho\}$. Notice, that the classical ℓ_1 ball occurs if $\boldsymbol{w}_n = \mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^m$ is the vector of ones. The projection onto $B_{\ell_1}[\boldsymbol{w}_n, \rho]$, is given in [6, Theorem 1], and the geometry of these sets is illustrated in Fig. 2.

It was shown that the estimates of the algorithm proposed in [6] converge asymptotically to a point, which lies arbitrarily close to the intersection of the hyperslabs with the weighted ℓ_1 balls, with the possible exception of a finite number of outliers. In this paper, a generalized version of the algorithm presented in [6] will be developed in the next section.

Remark 2: The weighted ℓ_1 ball is determined by the vector of weights, and the radius. Strategies of constructing the weights have been proposed in [2], [6]. More specifically, the rule $w_i^{(n)} = \frac{1}{(|h_i^{(n)}| + \tilde{\epsilon}_n)}$, $i = 1, \ldots, m$, where $\tilde{\epsilon}_n$ is a sequence of positive numbers, was used in order to avoid divisions by zero. It has been shown, e.g., [6], that by choosing the weights according to the previously mentioned strategy, a necessary condition that guarantees convergence of the algorithm to the unknown parameter is to set $\rho \geq \|\boldsymbol{h}_*\|_0$, since then it holds that $\boldsymbol{h}_* \in B_{\ell_1}[\boldsymbol{w}_n, \rho]$. ■

Here we should note that in [6], standard metric projections onto the hyperslabs and the weighted ℓ_1 balls take place. However, since we employ variable metric projections onto the hyperslabs, the induced inner product is time varying and it is determined by the matrix $\boldsymbol{G}_n$ (see also Appendix D). This fact forces us to employ variable metric projections onto the respective ℓ_1 balls too.

1) Claim 1: Recall the definition of the diagonal matrix $\boldsymbol{G}_n$. The variable metric projection onto $B_{\ell_1}[\boldsymbol{w}_n, \rho]$ is given by $P_{B_{\ell_1}[\boldsymbol{w}_n,\rho]}^{(\boldsymbol{G}_n)} = \boldsymbol{G}_n^{-\frac{1}{2}} P_{B_{\ell_1}[\boldsymbol{G}_n^{-\frac{1}{2}}\boldsymbol{w}_n,\rho]} \boldsymbol{G}_n^{\frac{1}{2}}$.

Proof: The proof is given in Appendix B. ■

IV. Adaptive Distributed Learning

We now come to the main point of this paper. Our task is to derive an algorithm, for distributed learning, to estimate the sparse unknown parameter vector $\boldsymbol{h}_* \in \mathbb{R}^m$, by exploiting measurements collected at the K nodes of a network which comply with the diffusion topology. An example of such a network is illustrated in Fig. 3. The node set is denoted by $\mathcal{N} = \{1, \ldots, K\}$ and we assume that each node is able to communicate, i.e., to exchange information, with a subset of $\mathcal{N}$, namely $\mathcal{N}_k$, $k = 1, \ldots, K$. This set, hereafter, will be called the *neighbourhood of* k. Moreover, each node has access to the measurement pair $(d_{k,n}, \boldsymbol{u}_{k,n})_{n \in \mathbb{Z}_{\geq 0}}$, $k \in \mathcal{N}$, where $\boldsymbol{u}_{k,n} \in \mathbb{R}^m$ and $d_{k,n} \in \mathbb{R}$, and the measurements are related according to $d_{k,n} = \boldsymbol{u}_{k,n}^T \boldsymbol{h}_* + v_{k,n}$, where $v_{k,n}$ stands for the additive noise at each node. In a nutshell, what differentiates the adaptive distributed learning from its classical adaptive counterpart is the fact that in the former case, each node, besides the locally received measurement pair, also exploits information received by its neighboring nodes. For a fixed node, say k and at every time instance, this extra information comprises the estimates of the unknown vector, which have been obtained at the previous time instance and from the nodes with which communication is possible, i.e., $\forall l \in \mathcal{N}_k$. The use of this extra information results in a faster convergence speed, as well as a lower steady state error floor, compared to the case where the measurement pair is solely used, e.g., [9], [12]. One more objective, which makes the exchange of the estimates crucial, is that the distributed "nature" of our problem imposes the need for *consensus*; this means that the nodes will have to converge to the same estimate. It has been shown that, if the nodes exchange their estimates and exploit this information properly, asymptotic consensus can be achieved [10], [11], [36], [37].

Depending on the way with which the estimates are exploited, the following cooperation strategies have been proposed:

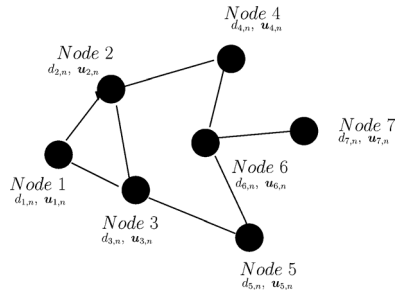


Fig. 3. Illustration of a diffusion network with $K = 7$ nodes.

- Combine -Adapt, in which, at every node the estimates from the neighborhood are fused under a certain protocol, and then the aggregate is put into the adaptation step [9], [11], [38].
- Adapt—Combine, where the adaptation step takes place prior to the combine one, [12], [37]. More specifically, at each node a temporary estimate is computed based on the received measurements. In the sequel, this is combined with the temporary estimates, which are received from the neighborhood in order to produce the final estimate, at each time instance.
- Consensus-based, where the computations are made in parallel and there is no clear distinction between the combine and the adapt steps [10], [36]. The general philosophy of these algorithms is to impose constraints, which force the nodes of the network to converge to the same estimate.

Now, let us shed light on the combination of the estimates, which are received from the neighborhood of each node. Recall the previous discussion; an arbitrary node, k, is able to communicate with every node that belongs to $\mathcal{N}_k$. We assume that the following hold true: $k \in \mathcal{N}_k, \forall k \in \mathcal{N}$ and $l \in \mathcal{N}_k \iff k \in \mathcal{N}_l,\ \forall k, l \in \mathcal{N}$. Moreover, we consider that the network is strongly connected, i.e., there is a, possibly multihop, path that connects every two nodes of the network. These assumptions are very common in adaptive distributed learning (see for example [9], [10]). As stated earlier, the received from the neighborhood estimates are fused under a certain protocol. The most common strategy is to take a linear combination of the estimates. To be more specific, we define the combination coefficients, for which we have that $c_{k,l}(n) > 0$, if $l \in \mathcal{N}_k$, $c_{k,l}(n) = 0$, if $l \notin \mathcal{N}_k$ and $\sum_{l\in\mathcal{N}_k} c_{k,l}(n) = 1$. From the previous definition, it can be readily seen that every node assigns a *weight* to each one of the estimates, which are received from the neighborhood. 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

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

Collecting all the coefficients for a network, we define the combination matrix $\boldsymbol{C}_n$, in which the k,lth component is $c_{k,l}(n)$. This matrix gives us information about the network's topology; that is, if the k,lth entry is equal to zero, this implies that the nodes $k,\ l$ are not connected. The opposite also holds true, since a positive coefficient implies that the nodes are connected. Finally, we define the $Km \times Km$ consensus matrix, $\boldsymbol{P}_n = \boldsymbol{C}_n \otimes \boldsymbol{I}_m$, where the symbol $\otimes$ stands for the Kronecker product. Some very useful properties of this matrix can be found in Appendix C.

V. Proposed Algorithmic Scheme

The goal is to bring together the sparsity promoting "tools", which where discussed in Section III, and to reformulate them in a fashion that is appropriate for distributed learning. We will proceed by adopting the combine-adapt strategy, which was presented in the previous section. The main steps of the algorithm, for node k and at time instance n, can be summarized as follows:

Algorithm:

1) The estimates from the neighborhood are received and combined with respect to the adopted combination strategy, in order to produce $\boldsymbol{\phi}_{k,n} = \sum_{l\in\mathcal{N}_k} c_{k,l}(n)\boldsymbol{h}_{l,n},\ \forall k \in \mathcal{N}$.
2) Exploiting the newly received measurements $d_{k,n}, \boldsymbol{u}_{k,n}$ the following hyperslab is defined: $S_{k,n} = \{\boldsymbol{h} \in \mathbb{R}^m : |d_{k,n} - \boldsymbol{u}_{k,n}^T\boldsymbol{h}| \leq \epsilon_k\}$, where the parameter ϵ_k is allowed to vary from node to node. The aggregate $\boldsymbol{\phi}_{k,n}$ is projected, using variable metric projections, onto the q most recent hyperslabs, constructed locally; in the sequel, their convex combination is computed. Analytically, the sliding window $\mathcal{J}_n := \overline{\max\{0, n-q+1\}, n}$ is defined, and it determines the hyperslabs that will be considered at time instance n. Given the set of weights $\forall j \in \mathcal{J}_n$, $\omega_{k,j}$, where $\sum_{j\in\mathcal{J}_n}\omega_{k,j} = 1, \forall k \in \mathcal{N}$, the convex combination of the projections onto the hyperslabs, i.e., $\sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_{k,j}}^{(\boldsymbol{G}_n)}(\boldsymbol{\phi}_{k,n})$ is computed. The effect of projecting onto a $q > 1$ number of hyperslabs is to speed up convergence [6].
3) The result of the previous step is projected onto the sparsity constraint set, i.e., the weighted ℓ_1 ball.

The previous steps can be encoded in the following mathematical formula:

$$\boldsymbol{h}_{k,n+1} \tag{4}$$

$$= P_{B_{\ell_1}[\boldsymbol{w}_n,\rho]}^{(\boldsymbol{G}_n)}\left(\boldsymbol{\phi}_{k,n} + \mu_{k,n} \times \left(\sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_{k,j}}^{(\boldsymbol{G}_n)}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n}\right)\right), \tag{5}$$

where $\mu_{k,n} \in (0, 2\mathcal{M}_{k,n})$, and

$$\mathcal{M}_{k,n} := \begin{cases} \dfrac{\sum_{j\in\mathcal{J}_n}\omega_{k,j}\left\|P_{S_{k,j}}^{(\boldsymbol{G}_n)}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n}\right\|^2}{\left\|\sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_j}^{(\boldsymbol{G}_n)}(\boldsymbol{\phi}_{k,n}) - \boldsymbol{\phi}_{k,n}\right\|^2}, \\ \quad \text{if } \sum_{j\in\mathcal{J}_n}\omega_{k,j}P_{S_{k,j}}^{(\boldsymbol{G}_n)}(\boldsymbol{\phi}_{k,n}) \neq \boldsymbol{\phi}_{k,n} \\ 1, \quad \text{otherwise.} \end{cases} \tag{6}$$

The algorithm has an elegant geometrical interpretation which can be seen in Fig. 4.

From the theoretical analysis concerning convergence, it turns out that the weighted ℓ_1 ball, as well as $\boldsymbol{G}_n$, have to be the same for every node of the network, which implies that this

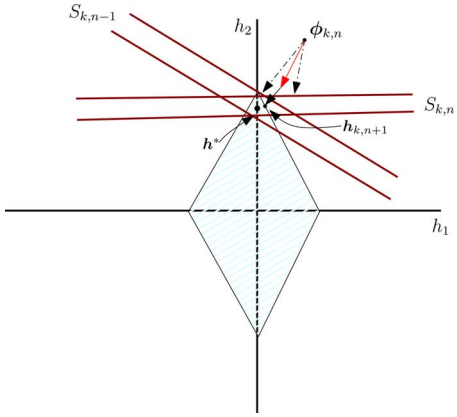


Fig. 4. Geometrical interpretation of the algorithm. The number of hyperslabs onto which $\boldsymbol{\phi}_{k,n}$ is projected, using variable metric projections, is $q = 2$. The result of these two projections, which are illustrated by the dash dotted black line, is combined (red line) and the result is projected (solid black line) onto the sparsity promoting weighted ℓ_1 ball, in order to produce the next estimate.

information cannot be constructed locally. This fact, as it will be established in the theoretical analysis of the algorithm, is essential in order to guarantee consensus. Hence, a reasonable strategy is to construct $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$, using the methodology described in Section III, via $\boldsymbol{h}_{k_{opt},n}$, where k_{opt} is the node with the smallest noise variance. It is obvious that this requires knowledge, in every node, of $\boldsymbol{h}_{k_{opt},n}$, something that is in general infeasible. However, it is not essential to update the parameters at every time instance. Instead, $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$ can be updated at every, $n' \geq 1$, time instances, where n' are the time steps required for $\boldsymbol{h}_{k_{opt},n}$ to be distributed over the network. Experiments regarding the robustness of the proposed algorithm with respect to n' are given in the Numerical Examples section. Moreover, as it will become clear in the Numerical Examples section, it turns out that the algorithm is robust in cases where the knowledge of the less noisy node is not available, and/or in cases where the assumption that these quantities must be common to all nodes is violated and each node uses the locally available values. It should be pointed out that such discrepancies in the adaptive filtering between theory and practices are common. The most celebrated example is the so called *independence assumption* adopted in the LMS, which is commonly employed to prove convergence, although in practice it does not hold, e.g., [34].

Regarding the complexity of the algorithm, it has been shown in [6], that if standard metric projections take place, then the complexity of the respective algorithm is $O(qm)$ coming from the projection operators and $O(m\log_2 m)$ occurring from the projection onto the weighted ℓ_1 ball. If we employ the variable metric projections, at each node, it is obvious that the term $\boldsymbol{G}_n^{-1}\boldsymbol{u}_{k,j},\ j \in \mathcal{J}_n$ has to be computed, and this adds qm multiplication operations.

Remark 3: The algorithm presented in [6] is a special case of the scheme in (5), if $K = 1$ and $\boldsymbol{G}_n = \boldsymbol{I}_m$. The same also holds for the IPNLMS [23] if we let $K = 1$, $q = 1$, $\epsilon_k = 0$ and $P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]} = I$, where I stands for the identity operator. ■

As it will be verified in Appendix D, the algorithm in (5) enjoys monotonicity, asymptotic optimality and strong convergence to a point that lies in the consensus subspace. The assumptions under which the previous hold are the following.

Assumptions:

(a) Define $\forall n \in \mathbb{Z}_{\geq 0}$, $\Omega_n = B_{\ell_1}[\boldsymbol{w}_n, \rho] \cap \left(\bigcap_{j\in\mathcal{J}_n}\bigcap_{k\in\mathcal{N}} S_{k,j}\right)$. Assume that there exists $n_0 \in \mathbb{Z}_{\geq 0}$, such that $\Omega := \bigcap_{n\geq n_0}\Omega_n \neq \emptyset$.

(b) There exists $n_1 \in \mathbb{Z}_{\geq 0}$, such that $\boldsymbol{G}_n = \boldsymbol{G}_{n_1} =: \boldsymbol{G}, \forall n \geq n_1$. In other words, the update of the matrix $\boldsymbol{G}_n$ pauses after a finite number of iterations[1].

(c) Assume a sufficiently small ε_1, such that $\forall k \in \mathcal{N}$, $\frac{\mu_{k,n}}{\mathcal{M}_{k,n}} \in [\varepsilon_1, 2-\varepsilon_1]$.

(d) Assume $\forall k \in \mathcal{N}\ \tilde{\omega}_k := \inf\{\omega_{k,j} : j \in \mathcal{J}_n, n \in \mathbb{Z}_{\geq 0}\} > 0$.

(e) Define $\mathfrak{C} := \boldsymbol{\Omega} \cap \mathcal{O}$, where the cartesian product space $\boldsymbol{\Omega} := \underbrace{\Omega \times \cdots \times \Omega}_{K}$. We assume that $\mathrm{ri}_{\mathcal{O}}\boldsymbol{\Omega} \neq \emptyset$, where this term stands for the relative interior of $\mathfrak{C}$ with respect to $\mathcal{O}$ (see Appendix A).

Theorem 1: Under the previous assumptions, the following hold:

(1) **Monotonicity**. Under assumptions (a), (b), (c), it holds that $\forall n \geq z_0,\ \forall \underline{\hat{\boldsymbol{h}}} \in \mathfrak{C}, \|\underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}} \leq \|\underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}}$, where $z_0 := \max\{n_0, n_1\}$, $\underline{\boldsymbol{G}}$ is the $Km \times Km$ block-diagonal matrix, with definition $\underline{\boldsymbol{G}} := \mathrm{diag}\{\underbrace{\boldsymbol{G}, \ldots, \boldsymbol{G}}_{K}\}$, and $\underline{\boldsymbol{h}}_n = [\boldsymbol{h}_{1,n}^T, \ldots, \boldsymbol{h}_{K,n}^T]^T \in \mathbb{R}^{Km}, \forall n \in \mathbb{Z}_{\geq 0}$.

(2) **Asymptotic Optimality**. If assumptions (a), (b), (c), (d) hold true then $\lim_{n\longrightarrow\infty} \max\{\mathrm{d}(\boldsymbol{h}_{k,n+1}, S_{k,j}) : j \in \mathcal{J}_n\} = 0,\ \forall k \in \mathcal{N}$, where $\mathrm{d}(\cdot, S_{k,j})$ denotes the distance of $\boldsymbol{h}_{k,n+1}$ from $S_{k,j}$ (see Appendix A). The previous implies that the distance of the estimates from the respective hyperslabs will tend asymptotically to zero.

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

(4) **Strong Convergence**. Under assumptions (a), (b), (c), (d), (e), it holds that $\lim_{n\longrightarrow\infty}\underline{\boldsymbol{h}}_n = \hat{\underline{\boldsymbol{h}}}_*, \hat{\underline{\boldsymbol{h}}}_* \in \mathcal{O}$. So, the estimates for the whole network, converge to a point that lies in the consensus subspace.

Proof: The proof is given in Appendix D. ■

VI. Numerical Examples

In this section, the performance of the proposed algorithm is validated within the system identification framework. Due to the fact that the online algorithmic schemes, which have been proposed in the literature, cover nondistributed learning scenarios, in the first experiment we compare the proposed algorithm against others in the context of a nondistributed system identification task. This essentially allow us to evaluate the use of the variable metric projections scheme, since this is one of the contributions of this paper. More specifically, we compare the proposed algorithm with the Adaptive Projection based algorithm using Weighted ℓ_1 Balls (APWL1) [6], with the Online Cyclic Coordinate Descent Time Weighted Lasso (OCCD-TWL), the Online Cyclic Coordinate Descent Time and Norm Weighted LASSO (OCCD-TNWL), both proposed in [8], and with the LMS-based, Sparse Adaptive Orthogonal Matching

[1]Notice that the matrix $\boldsymbol{G}_n$ is constructed via $\boldsymbol{h}_{k_{opt},n}$, hence $\forall n \geq n_1$, the variable metric projections is determined by $\boldsymbol{h}_{k_{opt},n_1}$. In practice, for sufficiently large n_1, the algorithm has converged and the fact that $\boldsymbol{G}_n$ is not updated does not affect the performance of the algorithm.

Pursuit (Spadomp) [27]. The unknown vector is of dimension $m = 512$ and the number of nonzero coefficients, equals to 20. Moreover, the input samples $\boldsymbol{u}_n = [u_n, \ldots, u_{n-m+1}]^T$ are drawn from a Gaussian distribution, with zero mean and standard deviation equal to 1. The noise process is Gaussian with variance equal to $\sigma^2 = 0.01$. Finally, the adopted performance metric, which will be used, is the average Mean Square Deviation (MSD), given by $\mathrm{MSD}(n) = \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{h}_{k,n} - \boldsymbol{h}_*\|^2$, and the curves occur from an averaging of 100 realizations for smoothing purposes.

In the projection-based algorithms, i.e., the proposed and the APWL1, the number of hyperslabs used per time update equals to $q = 55$, the width of the hyperslabs equals to $\epsilon = 1.3 \times \sigma$, and the step-size equals to $\mu_n = 0.2 \times \mathcal{M}_n$, where $\mathcal{M}_n$ is given in (6), where the node subscript is omitted. It should be pointed out that the performance of the algorithm turns out to be relatively insensitive to different choices of the parameter ϵ. A detailed experimental analysis on how different choices of ϵ affect the projection-based algorithms, has taken place in [6]. Moreover, for the weights we choose $\omega_n = \frac{1}{q}$. These choices are not necessarily optimal, albeit they lead to a good trade-off between the convergence speed and the steady state error floor. Regarding the choice of q, the larger the q the faster the convergence. This behaviour is also met in the Affine Projection Algorithm (APA), where the larger the number of affine sets, employed at each time instance, the faster the convergence. The parameter q is not a critical parameter, and one can choose it depending on the complexity load that can be afforded by the algorithm in real time operations. The radius of the weighted ℓ_1 ball equals to $\rho = \|\boldsymbol{h}_*\|_0$ and the weights are constructed according to the discussion in Section III. It should be stressed out that we experimentally observed that the proposed algorithm is rather insensitive to overestimated values of the sparsity level, which implies that even if we do not know the exact value of $\|\boldsymbol{h}_*\|_0$, if we set $\rho \geq \|\boldsymbol{h}_*\|_0$ the proposed algorithm exhibits a good performance; this behavior was also observed in [6]. Furthermore, we set $\tilde{\epsilon}_n = 10^{-2}$. The weighting matrix $\boldsymbol{G}_n$ is defined according to the strategy presented in Section III. Regarding the parameter α, we observed that a value close to 1 leads to a fast convergence speed but it increases the steady state error floor, and vice versa. So, at the beginning of the adaptation, we choose $\alpha = 0.99$ and at every 250 time instances, we set $\alpha = \frac{\alpha}{2}$. Finally, $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$ are updated at every time instance, i.e., $n' = 1$. In the OCCD-TWL and the OCCD-TNWL, the regularization parameter is chosen to be $\lambda_{\mathrm{TWL}} = \sqrt{2\sigma^2 n \log m}$, $\lambda_{\mathrm{TNWL}} = \sqrt{2\sigma^2 n^{\frac{4}{3}} \log m}$, respectively, as adviced in [8]. The step size, adopted in the Spadomp, equals to 0.2, due to the fact that this choice gives similar steady state error floor with the projection-based algorithms[2]. The forgetting factor of OCCD-TWN, OCCD-TNWL and Spadomp equals to 1 since, in the specific example, the system under consideration does not change with time. From Fig. 5, it can be seen that the proposed algorithm exhibits faster convergence speed compared to the APWL1 to the common error floor. Moreover, the proposed algorithm outperforms the Spadomp, since it converges faster and the steady state error floor is slightly better. We should point out, that the

[2]Extensive experiments have shown that a choice of a smaller step-size, results in a slower convergence speed, without significant improvement in the steady state error floor.

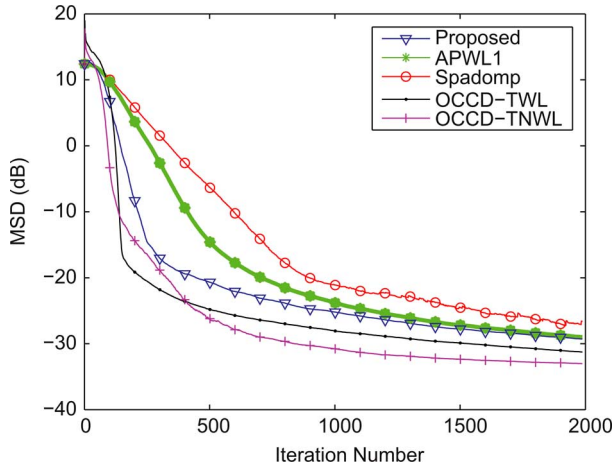


Fig. 5. MSD for the experiment 1.

complexity of the Spadomp is $O(m)$, which implies that for the previously mentioned choice of q, the proposed algorithm is of larger complexity. Compared to the OCCD-TWL, we observe that its performance is slightly better, compared to the proposed one, albeit the complexity of the algorithm is $O(m^2)$. Finally, the OCCD-TNWL outerforms the rest of the algorithms, at the expense of a higher complexity, which is approximately twice that of OCCD-TWL.

In the second experiment, we consider a network consisted of $K = 10$ nodes, in which the nodes are tasked to estimate an unknown parameter $\boldsymbol{h}_*$ of dimension $m = 256$. The number of nonzero coefficients, of the unknown parameter equals to 20 and each node has access to the measurements $(d_{k,n}, \boldsymbol{u}_{k,n})$, where the regressors are defined as in the previous experiment. The variance of the noise at each node is $\sigma_k^2 = 0.01\varsigma_k$, where $\varsigma_k \in [0.5, 1]$, following the uniform distribution. We compare the proposed algorithm with the distributed APWL1, i.e., if we let $\boldsymbol{G}_n = \boldsymbol{I}_m$, and the distributed Lasso (Dlasso) [3]. The Dlasso is a batch algorithm, which implies that the data have to be available prior to start the processing. So, here we assume that at every time instance, in which a new pair of data samples becomes available, the algorithm is re-initialized so as to solve a new optimization problem. For the projection-based algorithms, $q = 20$ and the rest of the parameters are chosen as in the previous experiment. Moreover, the combiners $c_{k,l}(n)$ are chosen with respect to the Metropolis rule. Finally, the regularization parameter in the Dlasso is set via the distributed cross-validation procedure, which is proposed in [3]. From Fig. 6, we observe that the Dlasso outperforms the projection-based algorithms and that the proposed algorithm converges faster than APWL1. However, the complexity of the proposed algorithm is significantly lower than that of the Dlasso. Dlasso, at every time instance, requires the inversion of a $m \times m$ matrix.

In the third experiment, we study the sensitivity of the proposed algorithm to choice of the parameter n', i.e., the frequency at which $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$ are updated. To this end, the parameters are the same as in the previous experiment, but we set different values to n'. Fig. 7 illustrates that the algorithm is relatively insensitive to the frequency of the updates, since even in the case where $n' = 20$ the algorithm exhibits fast convergence speed. This is important, since the robustness of the proposed scheme

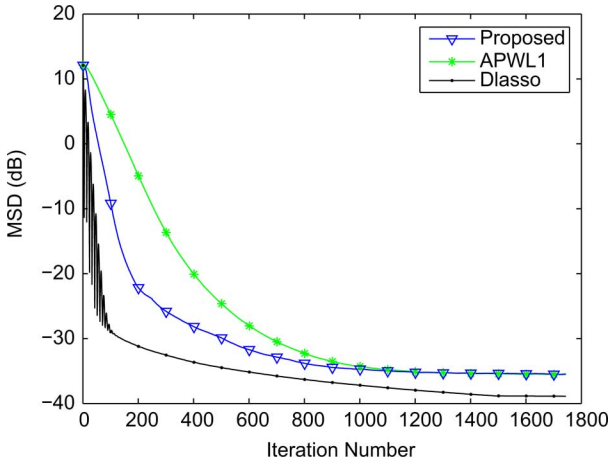


Fig. 6. MSD for the experiment 2.

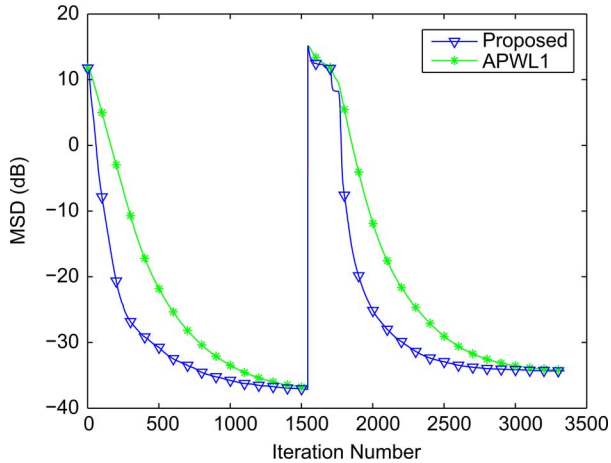


Fig. 8. MSD for the experiment 4.

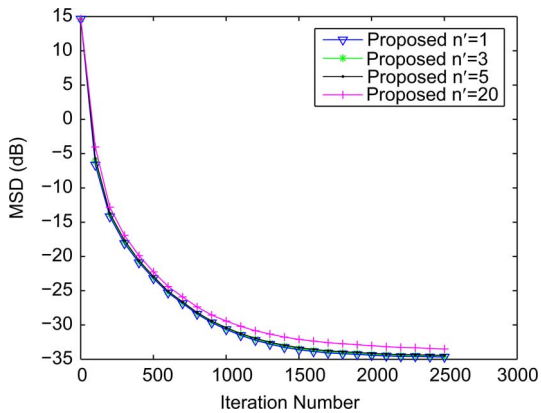


Fig. 7. MSD for the experiment 3.

to choice of the parameter n' makes it suitable to be adopted in distributed learning.

In the fourth experiment, we validate the performance of the algorithm in a nonstationary environment. It is by now well established that a fast convergence speed does not necessarily imply a good tracking ability [39]. More specifically, we consider that a sudden change in the unknown parameter takes place. So, until $\boldsymbol{h}_*$ changes, the parameters remain the same as in the second experiment, and after the sudden change, we have that $\|\boldsymbol{h}_*\|_0 = 15$. The radius of the weighted ℓ_1 ball is set equal to 23, since we have observed that the performance is relative insensitive to choices of ρ, as long as it remains larger than $\|\boldsymbol{h}_*\|_0$. Furthermore, we assume the algorithm is able to monitor sudden changes of the orbit $(\boldsymbol{h}_{k,n})_{n\in\mathbb{Z}_{\geq 0}}$, in order to reset the value of α when the channel changes. To be more specific, we reset the value of α, if the ratio $\frac{\|\boldsymbol{h}_{k,n+1}-\boldsymbol{h}_{k,n}\|}{\|\boldsymbol{h}_{k,n}-\boldsymbol{h}_{k,n-1}\|}$, $\forall k \in \mathcal{N}$, is greater than a threshold, which is chosen, here, to be equal to 10. This strategy is adopted since we observed that if the algorithm has converged, the previously mentioned ratio takes values close to 1, whereas if an abrupt change takes place in the unknown parameter, then the value of the ratio increases significantly. From Fig. 8, it can be observed that both the projection-based algorithms enjoy good tracking ability, when a sudden change occurs. Moreover, as in the previous experiments, the proposed algorithm converges faster than the APWL1 to a similar error floor.

Finally, in the fifth experiment, we study the robustness of the proposed scheme, with respect to adopting different strategies in order to construct $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$. To this end, we consider the following strategies: a) the previously mentioned quantities are constructed using the node with the smallest noise variance (Proposed a), b) $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$ are generated via the node with the largest variance (Proposed b) and c) $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$ are constructed locally at every node (Proposed c). Obviously, the latter one violates the theoretical assumption of having common weights to all nodes. In order to verify whether the nodes reach consensus, we plot the squared distance of $\underline{\boldsymbol{h}}_n$ from the consensus subspace, i.e., $\|\underline{\boldsymbol{h}}_n - P_{\mathcal{O}}(\underline{\boldsymbol{h}}_n)\|^2$. As in the previous experiments, the curves occur from an averaging of 100 independent experiments. From Fig. 9, it can be readily seen that the distance of $\underline{\boldsymbol{h}}_n$ from the consensus subspace, is decreasing as time steps increase. It is interesting that even in the Proposed c, where the assumption for achieving asymptotic consensus is violated, the estimates for the whole network tend asymptotically to the consensus subspace. Loosely speaking, even if there cannot be theoretical guarantees that the nodes will achieve asymptotic consensus in the case where each node constructs $\boldsymbol{w}_n$ and $\boldsymbol{G}_n$ using local information, the fact that the estimates received from the neighbourhood are combined at each step, leads the nodes to asymptotic consensus.

VII. Conclusions

A sparsity promoting adaptive algorithm for distributed learning has been proposed. The algorithm builds upon set-theoretic estimation arguments. In order to exploit the sparsity of the unknown vector, variable metric projections onto the hyperslabs, which define the region in which we search for the solution. Moreover, extra projections onto sparsity promoting weighted ℓ_1 balls are employed in order to further enhance the performance of the proposed scheme. Full convergence analysis has been derived. Numerical examples, within the system identification task, demonstrate the comparative performance of the proposed algorithm against other recently published algorithms.

APPENDIX A
BASIC CONCEPTS OF CONVEX ANALYSIS

The stage of discussion will be $\mathbb{R}^m$ and the induced inner product, given a positive definite $m \times m$ matrix $\boldsymbol{V}$, is $\langle \boldsymbol{h}_1, \boldsymbol{h}_2 \rangle_{\boldsymbol{V}} = \boldsymbol{h}_1^T \boldsymbol{V} \boldsymbol{h}_2$. A set $\mathcal{C} \subseteq \mathbb{R}^m$, for which it holds that $\forall \boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathcal{C}$ and $\forall t \in [0,1]$, $t\boldsymbol{h}_1 + (1-t)\boldsymbol{h}_2 \in \mathcal{C}$, is called convex. Moreover, a function $\Theta : \mathbb{R}^m \longrightarrow \mathbb{R}$ will be called convex if $\forall \boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^m$ and $\forall t \in [0,1]$ the inequality $\Theta(t\boldsymbol{h}_1 + (1-t)\boldsymbol{h}_2) \leq t\Theta(\boldsymbol{h}_1) + (1-t)\Theta(\boldsymbol{h}_2)$ is satisfied. Finally, the subdifferential of Θ at an arbitrary point, $\boldsymbol{h}$, is defined as the set of all subgradients of Θ at $\boldsymbol{h}$ ([40], [41]), i.e.,

$$\partial_{(\boldsymbol{V})}\Theta(\boldsymbol{h}) := \{\boldsymbol{s} \in \mathbb{R}^m : \Theta(\boldsymbol{h}) + \langle \boldsymbol{x} - \boldsymbol{h}, \boldsymbol{s} \rangle_{\boldsymbol{V}} \leq \Theta(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \mathbb{R}^m\}.$$

The distance of an arbitrary point $\boldsymbol{h}$ from a closed nonempty convex set $\mathcal{C}$, with respect to $\boldsymbol{V}$, is given by the distance function

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

and if we let $\boldsymbol{V}$ be the identity matrix, the Euclidean distance is given. This function is continuous, convex, nonnegative and is equal to zero for every point that lies in $\mathcal{C}$[41]. Moreover, the projection mapping, $P_{\mathcal{C}}^{(\boldsymbol{V})}$ onto $\mathcal{C}$, is defined as $P_{\mathcal{C}}^{(\boldsymbol{V})}(\boldsymbol{h}) := \mathrm{argmin}_{\boldsymbol{x} \in \mathcal{C}} \|\boldsymbol{h} - \boldsymbol{x}\|_{\boldsymbol{V}}$, and as in the distance function, if $\boldsymbol{V} = \boldsymbol{I}_m$ the standard metric projection is obtained.

Finally, the relative interior of a nonempty set, $\mathcal{C}$, with respect to another one, $\mathcal{S}$, is defined as

$$\mathrm{ri}_{\mathcal{S}}(\mathcal{C}) = \{\boldsymbol{h} \in \mathcal{C} : \exists \varepsilon_0 > 0 \text{ with } \emptyset \neq (B_{(\boldsymbol{h}_0, \varepsilon_0)} \cap \mathcal{S}) \subset \mathcal{C}\},$$

where $B_{(\boldsymbol{h}_0, \varepsilon_0)}$ is the open ball with definition $B_{(\boldsymbol{h}_0, \varepsilon_0)} := \{\boldsymbol{h} \in \mathbb{R}^m : \|\boldsymbol{h} - \boldsymbol{h}_0\| < \varepsilon_0\}$ (see for example [42]), with center $\boldsymbol{h}_0$ and radius equal to ε_0.

APPENDIX B
VARIABLE METRIC PROJECTION ONTO THE WEIGHTED ℓ_1 BALL

The variable metric projection of $\boldsymbol{h}$, onto $B_{\ell_1}[\boldsymbol{w}_n, \rho]$, is given by

$$\begin{aligned} \min_{\boldsymbol{x} \in B_{\ell_1}[\boldsymbol{w}_n, \rho]} \quad & \|\boldsymbol{h} - \boldsymbol{x}\|_{\boldsymbol{G}_n}^2 \\ \text{s.t.} \quad & \sum_{i=1}^{m} w_i^{(n)} |x_i| \leq \rho, \end{aligned}$$

where $\boldsymbol{x} := [x_1, \ldots, x_m]^T$. However, $\|\boldsymbol{h} - \boldsymbol{x}\|_{\boldsymbol{G}_n}^2 = \|\boldsymbol{G}_n^{\frac{1}{2}}(\boldsymbol{h} - \boldsymbol{x})\|^2 = \|\boldsymbol{G}_n^{\frac{1}{2}}\boldsymbol{h} - \boldsymbol{\xi}\|^2$, where $\boldsymbol{\xi} := \boldsymbol{G}_n^{\frac{1}{2}}\boldsymbol{x}$. Moreover, $\boldsymbol{x} = \boldsymbol{G}_n^{-\frac{1}{2}}\boldsymbol{\xi} \iff x_i = \sqrt{g_{i,n}^{-1}}\xi_i, i = 1, \ldots, m$, where ξ_i are the coefficients of $\boldsymbol{\xi}$. From the previous, it holds that $\sum_{i=1}^m w_i^{(n)}|x_i| = \sum_{i=1}^m \sqrt{g_{i,n}^{-1}} w_i^{(n)} |\xi_i|$. Hence the initial optimization problem, is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & \|\boldsymbol{G}_n^{\frac{1}{2}}\boldsymbol{h} - \boldsymbol{\xi}\|^2 \\ \text{s.t.} \quad & \sum_{i=1}^{m} \sqrt{g_{i,n}^{-1}} w_i^{(n)} |\xi_i| \leq \rho. \end{aligned}$$

The solution of the previous optimization, is the standard metric projection of $\boldsymbol{G}_n^{\frac{1}{2}}\boldsymbol{h}$ onto $B_{\ell_1}[\boldsymbol{G}_n^{-\frac{1}{2}}\boldsymbol{w}_n, \rho]$ and it can be found in [6]. So, from the previous $\boldsymbol{\xi}_{opt} = P_{B_{\ell_1}[\boldsymbol{G}_n^{-\frac{1}{2}}\boldsymbol{w}_n, \rho]}(\boldsymbol{G}_n^{\frac{1}{2}}\boldsymbol{h}) \iff P^{(\boldsymbol{G}_n)}_{B_{\ell_1}[\boldsymbol{G}_n^{-\frac{1}{2}}\boldsymbol{w}_n, \rho]}(\boldsymbol{h}) = \boldsymbol{G}_n^{-\frac{1}{2}} P_{B_{\ell_1}[\boldsymbol{G}_n^{-\frac{1}{2}}\boldsymbol{w}_n, \rho]}(\boldsymbol{G}_n^{\frac{1}{2}}\boldsymbol{h})$.

APPENDIX C
PROPERTIES OF THE CONSENSUS MATRIX

Consider a $Nm \times Nm$ consensus matrix $\boldsymbol{P}_n$. Then the following hold [37]:

(1) $\|\boldsymbol{P}_n\| = 1$.

(2) 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 $\boldsymbol{B}_k = \frac{(\boldsymbol{1}_K \otimes \boldsymbol{e}_k)}{\sqrt{K}}$, $\boldsymbol{e}_k$ is a $m \times 1$ vector of zeros except the kth entry, which is one and $\boldsymbol{X}_n$ is an $Km \times Km$ matrix for which it holds that $\|\boldsymbol{X}_n\| < 1$.

(3) $\boldsymbol{P}_n \underline{\check{\boldsymbol{h}}} = \underline{\check{\boldsymbol{h}}},\ \forall \underline{\check{\boldsymbol{h}}} \in \mathcal{O} := \{\underline{\boldsymbol{h}} \in \mathbb{R}^{Km} : \underline{\boldsymbol{h}} = [\boldsymbol{h}^T, \ldots, \boldsymbol{h}^T]^T,\ \boldsymbol{h} \in \mathbb{R}^m\}$. The subspace $\mathcal{O}$ is the so called consensus subspace of dimension m, and $\boldsymbol{B}_k,\ k = 1, \ldots, m$, constitute a basis for this set. Hence, the orthogonal projection of a vector, $\underline{\boldsymbol{h}}$, onto this linear subspace is given by $P_{\mathcal{O}}(\underline{\boldsymbol{h}}) := \boldsymbol{B}\boldsymbol{B}^T\underline{\boldsymbol{h}},\ \forall \underline{\boldsymbol{h}} \in \mathbb{R}^{Km}$.

APPENDIX D
PROOF OF THEOREM 1

Monotonicity:

Lemma 1: Define the following nonnegative loss functions, $\forall k \in \mathcal{N}$:

$$\forall n \in \mathbb{Z}_{\geq 0},\ \forall \boldsymbol{h} \in \mathbb{R}^m, \quad \Theta_{k,n}(\boldsymbol{h}) := \begin{cases} \sum_{j \in \mathcal{I}_{k,n}} \frac{\omega_{k,j}\, \mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j})}{L_{k,n}} \mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{h}, S_{k,j}), & \text{if } \mathcal{I}_{k,n} \neq \emptyset, \\ 0, & \text{if } \mathcal{I}_{k,n} = \emptyset, \end{cases} \tag{7}$$

where $\mathcal{I}_{k,n} := \{j \in \mathcal{J}_n : \boldsymbol{\phi}_{k,n} \notin S_{k,j}\}$ and $L_{k,n} := \sum_{j \in \mathcal{J}_n} \omega_{k,j} \mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j})$. Then (5) is equivalent to[3]

$$\forall n \in \mathbb{Z}_{\geq 0},\ \forall k \in \mathcal{N}, \quad \boldsymbol{h}_{k,n+1} = \begin{cases} P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n, \rho]} \left(\boldsymbol{\phi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|_{\boldsymbol{G}}^2} \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) \right), & \text{if } \mathcal{I}_{k,n} \neq \emptyset, \\ P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n, \rho]} \left(\boldsymbol{\phi}_{k,n} \right), & \text{if } \mathcal{I}_{k,n} = \emptyset, \end{cases} \tag{8}$$

where $\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})$ is the subgradient of the function and $\lambda_{k,n} \in (0, 2)$.

Proof: First of all, notice that if $\mathcal{I}_{k,n} \neq \emptyset$, then there exists $j_0 \in \mathcal{J}_n$ such that $\boldsymbol{\phi}_{k,n} \notin S_{k,j_0} \iff \mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j_0}) > 0$. Hence, $L_{k,n} \geq \omega_{k,j_0} \mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j_0}) > 0$, which implies that the denominator in (8) is positive and the cost function is

[3]The time dependence on $\boldsymbol{G}_n$ is omitted for simplicity in notation.

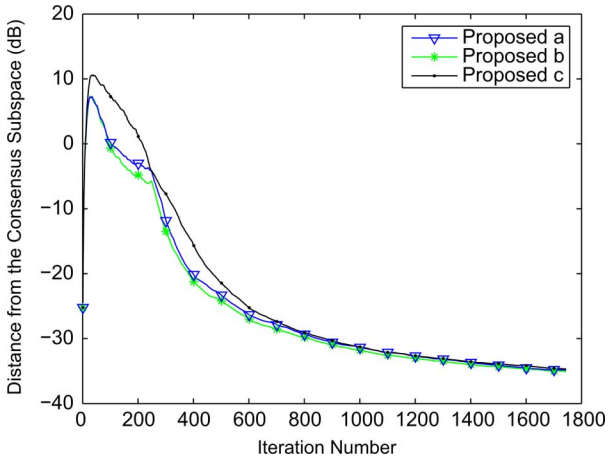


Fig. 9. Squared distance from the consensus subspace, for experiment 5.

well defined. Now, a subgradient of the distance function, i.e., $\mathrm{d}_{(\boldsymbol{G})}(\cdot, S_{k,j})$, is the following [43]:

$$\mathrm{d}'_{(\boldsymbol{G})}(\boldsymbol{h}, S_{k,j}) = \begin{cases} \frac{\boldsymbol{h} - P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{h})}{\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{h}, S_{k,j})}, & \text{if } \boldsymbol{h} \notin S_{k,j} \\ \mathbf{o}, & \text{otherwise.} \end{cases} \tag{9}$$

Recalling basic properties of the subdifferential (see for example [41]), we have that

$$\partial\Theta_{k,n}(\boldsymbol{h}) = \begin{cases} \sum_{j\in\mathcal{I}_{k,n}} \frac{\omega_{k,j}\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j})}{L_{k,n}} \partial \mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{h}, S_{k,j}), \\ \qquad\qquad \text{if } \mathcal{I}_{k,n} \neq \emptyset, \\ \{\mathbf{o}\}, \qquad\qquad \text{if } \mathcal{I}_{k,n} = \emptyset. \end{cases} \tag{10}$$

So, combining (9), (10) and if $\mathcal{I}_{k,n} \neq \emptyset$ we have

$$\begin{aligned} \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) &= \sum_{j\in\mathcal{I}_{k,n}} \frac{\omega_{k,j}\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j})}{L_{k,n}} \\ &\quad \times \frac{\boldsymbol{\phi}_{k,n} - P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{\phi}_{k,n})}{\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n}, S_{k,j})} \\ &= \frac{1}{L_{k,n}} \sum_{j\in\mathcal{I}_{k,n}} \omega_{k,j} \left(\boldsymbol{\phi}_{k,n} - P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{\phi}_{k,n})\right) \\ &= \frac{1}{L_{k,n}} \sum_{j\in\mathcal{J}_{n}} \omega_{k,j} \left(\boldsymbol{\phi}_{k,n} - P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{\phi}_{k,n})\right). \end{aligned} \tag{11}$$

Nevertheless, since $\mathcal{I}_{k,n} \neq \emptyset$, then there exists $j_0 \in \mathcal{J}_n$ such that $\boldsymbol{\phi}_{k,n} \notin S_{k,j_0} \iff P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{\phi}_{k,n}) \neq \boldsymbol{\phi}_{k,n}$. So, if $\mathcal{I}_{k,n} \neq \emptyset$ then $\Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) \neq \mathbf{0}$. Following similar steps as in [6], it can be proved that $\forall n \geq z_0$, $\forall j \in \mathcal{J}_n$, $\forall k \in \mathcal{N}$, $\Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) = \mathbf{0} \iff \boldsymbol{\phi}_{k,n} = \sum_{j\in\mathcal{J}_n} \omega_{k,j} P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{\phi}_{k,n})$. From this fact, if we define $\mu_{k,n} := \mathcal{M}_{k,n}\lambda_{k,n}$, and if we substitute (11) in (8) the lemma is proved. ■

Claim 2: It holds that $\|\boldsymbol{P}\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}} \leq \|\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}}$, $\forall \breve{\underline{\boldsymbol{h}}} \in \mathcal{O}, \forall \underline{\boldsymbol{h}} \in \mathbb{R}^{Km}$, where $\boldsymbol{P}$ is a $Km \times Km$ consensus matrix with $\|\boldsymbol{P}\| = 1$.

Proof: From the definition of $\|\cdot\|_{\underline{\boldsymbol{G}}}$, it can be readily seen that $\|\boldsymbol{P}\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}} = \|\underline{\boldsymbol{G}}^{\frac{1}{2}}\left(\boldsymbol{P}\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\right)\| = \|\underline{\boldsymbol{G}}^{\frac{1}{2}}\boldsymbol{P}\left(\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\right)\|$, where this holds since $\breve{\underline{\boldsymbol{h}}} \in \mathcal{O}$. Moreover, $\underline{\boldsymbol{h}} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_K \end{bmatrix}, \boldsymbol{h}_k \in \mathbb{R}^m, k \in \mathcal{N}$ and $\breve{\underline{\boldsymbol{h}}} \in \mathcal{O} \iff \breve{\underline{\boldsymbol{h}}} = \begin{bmatrix} \breve{\boldsymbol{h}} \\ \vdots \\ \breve{\boldsymbol{h}} \end{bmatrix}, \breve{\boldsymbol{h}} \in \mathbb{R}^m$. Recalling the definition of the consensus matrix, with coefficients $c_{k,l}, k, l \in \mathcal{N}$, we have the following

$$\begin{aligned} \|\underline{\boldsymbol{G}}^{\frac{1}{2}}\boldsymbol{P}\left(\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\right)\| &= \left\| \begin{bmatrix} \boldsymbol{G}^{\frac{1}{2}} \sum_{l\in\mathcal{N}_1} c_{1,l}\left(\boldsymbol{h}_l - \breve{\boldsymbol{h}}\right) \\ \vdots \\ \boldsymbol{G}^{\frac{1}{2}} \sum_{l\in\mathcal{N}_K} c_{K,l}\left(\boldsymbol{h}_l - \breve{\boldsymbol{h}}\right) \end{bmatrix} \right\| \\ &= \left\| \begin{bmatrix} \sum_{l\in\mathcal{N}_1} c_{1,l}\boldsymbol{G}^{\frac{1}{2}}\left(\boldsymbol{h}_l - \breve{\boldsymbol{h}}\right) \\ \vdots \\ \sum_{l\in\mathcal{N}_K} c_{K,l}\boldsymbol{G}^{\frac{1}{2}}\left(\boldsymbol{h}_l - \breve{\boldsymbol{h}}\right) \end{bmatrix} \right\| \\ &= \left\| \boldsymbol{P} \begin{bmatrix} \boldsymbol{G}^{\frac{1}{2}}\left(\boldsymbol{h}_1 - \breve{\boldsymbol{h}}\right) \\ \vdots \\ \boldsymbol{G}^{\frac{1}{2}}\left(\boldsymbol{h}_K - \breve{\boldsymbol{h}}\right) \end{bmatrix} \right\| \\ &\leq \|\boldsymbol{P}\| \left\| \begin{bmatrix} \boldsymbol{G}^{\frac{1}{2}}\left(\boldsymbol{h}_1 - \breve{\boldsymbol{h}}\right) \\ \vdots \\ \boldsymbol{G}^{\frac{1}{2}}\left(\boldsymbol{h}_K - \breve{\boldsymbol{h}}\right) \end{bmatrix} \right\| \\ &= \left\|\underline{\boldsymbol{h}} - \breve{\underline{\boldsymbol{h}}}\right\|_{\underline{\boldsymbol{G}}}. \end{aligned} \tag{12}$$

From (12), our claim is proved. ■

First of all, given a convex function $\Theta : \mathbb{R}^m \to \mathbb{R}$, with nonempty level set, where the level set is defined $\mathrm{lev}_{\leq 0}\Theta := \{\boldsymbol{h} \in \mathbb{R}^m : \Theta(\boldsymbol{h}) \leq 0\}$, let us define the subgradient projection mapping, as follows $T^{(\boldsymbol{G})}_{\Theta} : \mathbb{R}^m \to \mathbb{R}^m$[43]:

$$T^{(\boldsymbol{G})}_{\Theta}(\boldsymbol{h}) := \begin{cases} \boldsymbol{h} - \frac{\Theta(\boldsymbol{h})}{\|\Theta'(\boldsymbol{h})\|^2_{\boldsymbol{G}}}\Theta'(\boldsymbol{h}), & \boldsymbol{h} \notin \mathrm{lev}_{\leq 0}\Theta \\ \boldsymbol{h}, & \boldsymbol{h} \in \mathrm{lev}_{\leq 0}\Theta, \end{cases}$$

where $\Theta'(\boldsymbol{h})$ is any subgradient of Θ, at $\boldsymbol{h}$. Similarly, we define the relaxed subgradient projection mapping, $T^{(\boldsymbol{G})}_{\Theta,\lambda}(\boldsymbol{h}) := I + \lambda(T^{(\boldsymbol{G})}_{\Theta}(\boldsymbol{h}) - I), \lambda \in (0, 2)$, where I is the identity mapping.

Now, given a nonempty closed convex set, say $\mathcal{C} \subset \mathbb{R}^m$, and a convex function $\Theta : \mathbb{R}^m \to \mathbb{R}$, such that $\mathcal{C} \cap \mathrm{lev}_{\leq 0}\Theta \neq \emptyset$ it holds that [43]:

$$\begin{aligned} &\forall\, \boldsymbol{h} \in \mathbb{R}^m, \forall \hat{\boldsymbol{h}} \in \mathcal{C} \cap \mathrm{lev}_{\leq 0}\Theta \\ &: \frac{2-\lambda}{2}\|\boldsymbol{h} - P_{\mathcal{C}}T^{(\boldsymbol{G})}_{\Theta,\lambda}(\boldsymbol{h})\|^2_{\boldsymbol{G}} \\ &\quad \leq \|\boldsymbol{h} - \hat{\boldsymbol{h}}\|^2_{\boldsymbol{G}} - \|P_{\mathcal{C}}T^{(\boldsymbol{G})}_{\Theta,\lambda}(\boldsymbol{h}) - \hat{\boldsymbol{h}}\|^2_{\boldsymbol{G}}. \end{aligned} \tag{13}$$

Following similar steps as in [6], it can be proved that $\forall n \geq z_0$, $\boldsymbol{\phi}_{k,n} \in \mathrm{lev}_{\leq 0}\Theta_{k,n} \iff \mathcal{I}_{k,n} = \emptyset$ and $\forall n \geq z_0, \boldsymbol{\phi}_{k,n} \notin \mathrm{lev}_{\leq 0}\Theta_{k,n} \iff \mathcal{I}_{k,n} \neq \emptyset$. Moreover, $\mathrm{lev}_{\leq 0}\Theta_{k,n} = \bigcap_{j\in\mathcal{I}_{k,n}} S_{k,j} \supset \Omega_n \supset \Omega$. Recall the definition of the relaxed projection mapping; it can be readily seen that

$\boldsymbol{h}_{k,n+1} = P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]} T^{(\boldsymbol{G})}_{\Theta_{k,n},\lambda_{k,n}}(\boldsymbol{\phi}_{k,n})$. Exploiting this fact, under Assumptions (a), (b), and (13) we have that

$$\begin{aligned}
&\forall n \geq z_0, \forall k \in \mathcal{N}, \forall \hat{\boldsymbol{h}} \in \Omega \\
&: 0 \leq \frac{2-\lambda_{k,n}}{2} \|\boldsymbol{\phi}_{k,n} - \boldsymbol{h}_{k,n+1}\|^2_{\boldsymbol{G}} \\
&= \frac{2-\lambda_{k,n}}{2} \|\boldsymbol{\phi}_{k,n} - P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]} T^{(\boldsymbol{G})}_{\Theta_{k,n},\lambda_{k,n}}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}} \\
&\leq \|\boldsymbol{\phi}_{k,n} - \hat{\boldsymbol{h}}\|^2_{\boldsymbol{G}} - \|P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]} T^{(\boldsymbol{G})}_{\Theta_{k,n},\lambda_{k,n}}(\boldsymbol{\phi}_{k,n}) \\
&\quad - \hat{\boldsymbol{h}}\|^2_{\boldsymbol{G}}. \qquad (14)
\end{aligned}$$

Recalling the definitions $\underline{\boldsymbol{h}}_n = [\boldsymbol{h}^T_{1,n}, \ldots, \boldsymbol{h}^T_{K,n}]^T \in \mathbb{R}^{Km}$, $\boldsymbol{P}_n \underline{\boldsymbol{h}}_n = [\boldsymbol{\phi}^T_{1,n}, \ldots, \boldsymbol{\phi}^T_{K,n}]^T \in \mathbb{R}^{Km}$, and (14), we have

$$\begin{aligned}
&\forall n \geq z_0, \ \forall \underline{\hat{\boldsymbol{h}}} \in \mathfrak{C} : \\
&0 \leq \min_k \left\{ \frac{2-\lambda_{k,n}}{2} \right\} \|\boldsymbol{P}_n \underline{\boldsymbol{h}}_n - \underline{\boldsymbol{h}}_{n+1}\|^2_{\underline{\boldsymbol{G}}} \\
&\leq \|\boldsymbol{P}_n \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}} - \|\underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}}. \qquad (15)
\end{aligned}$$

Nevertheless, from Claim 2, the previous inequality can be rewritten

$$\begin{aligned}
0 &\leq \|\boldsymbol{P}_n \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}} - \|\underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}} \\
&\leq \|\underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}} - \|\underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}}.
\end{aligned}$$

Hence,

$$\forall n \geq z_0, \ \forall \underline{\hat{\boldsymbol{h}}} \in \mathfrak{C} : \|\underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}} \leq \|\underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}} \qquad (16)$$

which completes our proof. ■

Asymptotic Optimality: A well known property of the projection operator (see for example [43]), is the nonexpansivity, i.e., given a nonempty set $\mathcal{C}$, $\|P^{(\boldsymbol{G})}_{\mathcal{C}}(\boldsymbol{h}_1) - P^{(\boldsymbol{G})}_{\mathcal{C}}(\boldsymbol{h}_2)\|_{\boldsymbol{G}} \leq \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_{\boldsymbol{G}}, \forall \boldsymbol{h}_1, \boldsymbol{h}_2 \in \mathbb{R}^m$. Recall the definition of the algorithm given in (8). Then, $\forall k \in \mathcal{N}, \forall n \geq z_0, \forall \hat{\boldsymbol{h}} \in \Omega$, we have

$$\begin{aligned}
&\|\boldsymbol{h}_{k,n+1} - \hat{\boldsymbol{h}}\|_{\boldsymbol{G}} \\
&= \left\| P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]} \left(\boldsymbol{\phi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}} \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) \right) - \hat{\boldsymbol{h}} \right\|_{\boldsymbol{G}} \\
&= \left\| P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]} \left(\boldsymbol{\phi}_{k,n} \right.\right. \\
&\quad \left.\left. - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}} \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) \right) - P^{(\boldsymbol{G})}_{B_{\ell_1}[\boldsymbol{w}_n,\rho]}(\hat{\boldsymbol{h}}) \right\|_{\boldsymbol{G}} \\
&\leq \left\| \boldsymbol{\phi}_{k,n} - \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}} \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) - \hat{\boldsymbol{h}} \right\|_{\boldsymbol{G}}, \qquad (17)
\end{aligned}$$

where the equality in the second line holds since, by definition, $\hat{\boldsymbol{h}} \in \Omega \subset B_{\ell_1}[\boldsymbol{w}_n, \rho]$ and the inequality, from the nonexpansivity of the projection operator. Assuming that $\Theta'_{k,n}(\boldsymbol{\phi}_{k,n}) \neq \boldsymbol{0}, \forall k \in \mathcal{N}$, and rewriting (17) for all the nodes of the network we have

$$\begin{aligned}
&\left\| \underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}} \right\|^2_{\underline{\boldsymbol{G}}} \\
&\leq \left\| \begin{bmatrix} \boldsymbol{\phi}_{1,n} - \lambda_{1,n} \frac{\Theta_{1,n}(\boldsymbol{\phi}_{1,n})}{\|\Theta'_{1,n}(\boldsymbol{\phi}_{1,n})\|^2_{\boldsymbol{G}}} \Theta'_{1,n}(\boldsymbol{\phi}_{1,n}) \\ \vdots \\ \boldsymbol{\phi}_{K,n} - \lambda_{K,n} \frac{\Theta_{K,n}(\boldsymbol{\phi}_{K,n})}{\|\Theta'_{K,n}(\boldsymbol{\phi}_{K,n})\|^2_{\boldsymbol{G}}} \Theta'_{K,n}(\boldsymbol{\phi}_{K,n}) \end{bmatrix} - \underline{\hat{\boldsymbol{h}}} \right\|^2_{\underline{\boldsymbol{G}}} \\
&= \left\| \begin{bmatrix} \boldsymbol{\phi}_{1,n} - \hat{\boldsymbol{h}} \\ \vdots \\ \boldsymbol{\phi}_{K,n} - \hat{\boldsymbol{h}} \end{bmatrix} \right\|^2_{\underline{\boldsymbol{G}}} + \sum_{k \in \mathcal{N}} \lambda^2_{k,n} \frac{\left(\Theta_{k,n}(\boldsymbol{\phi}_{k,n})\right)^2}{\left\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\right\|^2_{\boldsymbol{G}}} \\
&\quad - 2 \sum_{k \in \mathcal{N}} \lambda_{k,n} \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n}) \langle \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}), \left(\boldsymbol{\phi}_{k,n} - \hat{\boldsymbol{h}}\right) \rangle_{\boldsymbol{G}}}{\left\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\right\|^2_{\boldsymbol{G}}}. \qquad (18)
\end{aligned}$$

Nevertheless,

$$\left\| \begin{bmatrix} \boldsymbol{\phi}_{1,n} - \hat{\boldsymbol{h}} \\ \vdots \\ \boldsymbol{\phi}_{K,n} - \hat{\boldsymbol{h}} \end{bmatrix} \right\|_{\underline{\boldsymbol{G}}} = \|\boldsymbol{P}_n \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}} \leq \|\underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}}\|_{\underline{\boldsymbol{G}}}. \qquad (19)$$

From the definition of the subgradient, we have

$$\begin{aligned}
\langle \Theta'_{k,n}(\boldsymbol{\phi}_{k,n}), \left(\boldsymbol{\phi}_{k,n} - \hat{\boldsymbol{h}}\right) \rangle_{\boldsymbol{G}} &\geq \Theta_{k,n}(\boldsymbol{\phi}_{k,n}) \\
&\quad - \Theta_{k,n}(\hat{\boldsymbol{h}}) = \Theta_{k,n}(\boldsymbol{\phi}_{k,n}), \qquad (20)
\end{aligned}$$

where the last equation, holds due to the fact that $\hat{\boldsymbol{h}} \in \Omega \iff \Theta'_{k,n}(\hat{\boldsymbol{h}}) = \boldsymbol{0}$. Taking (19) and (20) into consideration, we obtain

$$\begin{aligned}
&\left\| \underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}} \right\|^2_{\underline{\boldsymbol{G}}} \\
&\leq \left\| \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}} \right\|^2_{\underline{\boldsymbol{G}}} - \sum_{k \in \mathcal{N}} \lambda_{k,n}(2 - \lambda_{k,n}) \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}}. \qquad (21)
\end{aligned}$$

Here, notice that the sequence $\left\| \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}} \right\|_{\underline{\boldsymbol{G}}}$ is bounded and monotone decreasing, hence it converges. The latter fact implies that

$$\lim_{n \to \infty} \left(\left\| \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}} \right\|_{\underline{\boldsymbol{G}}} - \left\| \underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}} \right\|_{\underline{\boldsymbol{G}}} \right) = 0. \qquad (22)$$

Under Assumption (c), (21) can be rewritten

$$\begin{aligned}
&\sum_{k \in \mathcal{N}} \varepsilon^2_1 \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}} \\
&\leq \sum_{k \in \mathcal{N}} \lambda_{k,n}(2 - \lambda_{k,n}) \frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}} \\
&\leq \left\| \underline{\boldsymbol{h}}_n - \underline{\hat{\boldsymbol{h}}} \right\|^2_{\underline{\boldsymbol{G}}} - \left\| \underline{\boldsymbol{h}}_{n+1} - \underline{\hat{\boldsymbol{h}}} \right\|^2_{\underline{\boldsymbol{G}}}. \qquad (23)
\end{aligned}$$

Taking limits in (23) and recalling (22) we have that

$$\lim_{n\longrightarrow\infty}\frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}}=0,\quad \forall k\in\mathcal{N}.$$

If we follow similar steps as in [6], it can be verified that $\forall n\in\mathbb{Z}_{\geq 0},\forall k\in\mathcal{N},\forall \boldsymbol{h}\in\mathbb{R}^m:\|\Theta'_{k,n}(\boldsymbol{h})\|_{\boldsymbol{G}}\leq 1$. So, if $\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\neq\mathbf{0}$

$$\Theta_{k,n}(\boldsymbol{\phi}_{k,n})\leq\frac{\Theta_{k,n}(\boldsymbol{\phi}_{k,n})}{\|\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})\|^2_{\boldsymbol{G}}}\rightarrow 0,\ n\rightarrow\infty. \tag{24}$$

Obviously, recalling the previous discussion, $\Theta'_{k,n}(\boldsymbol{\phi}_{k,n})=\mathbf{0}\iff\Theta_{k,n}(\boldsymbol{\phi}_{k,n})=0,\ \forall n\geq z_0$. Combining this fact together with (24), we have that

$$\forall k\in\mathcal{N},\ \lim_{n\longrightarrow\infty}\Theta_{k,n}(\boldsymbol{\phi}_{k,n})=0. \tag{25}$$

Now, following similar steps as in [6], it can be shown that there exists $D>0$ such that $L_{k,n}\leq D,\forall k\in\mathcal{N},\ \forall n\in\mathbb{Z}_{\geq 0}$. From the definition of $\Theta_{k,n}$, and under Assumption (d), we have $\forall k\in\mathcal{N}$

$$\begin{aligned}\frac{D}{\tilde{\omega}_k}\Theta_{k,n}(\boldsymbol{\phi}_{k,n})&\geq\frac{D}{\tilde{\omega}_k}\sum_{j\in\mathcal{J}_n}\omega_{k,j}\frac{\mathrm{d}^2_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n},S_{k,j})}{L_{k,n}}\\&\geq\frac{D}{\tilde{\omega}_k}\frac{\tilde{\omega}_k}{D}\sum_{j\in\mathcal{J}_n}\mathrm{d}^2_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n},S_{k,j})\\&\geq\max\{\mathrm{d}^2_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n},S_{k,j}):j\in\mathcal{J}_n\}.\end{aligned}$$

Taking limits in the previous inequality, we obtain that

$$\lim_{n\longrightarrow\infty}\max\{\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n},S_{k,j}):j\in\mathcal{J}_n\}=0. \tag{26}$$

Combining (15) with the result of Claim 2, we have

$$\begin{aligned}&\forall n\geq z_0,\ \forall\hat{\underline{\boldsymbol{h}}}\in\mathfrak{C}:\\&0\leq\min_k\left\{\frac{2-\lambda_{k,n}}{2}\right\}\|\boldsymbol{P}_n\underline{\boldsymbol{h}}_n-\underline{\boldsymbol{h}}_{n+1}\|^2_{\underline{\boldsymbol{G}}}\\&\leq\|\underline{\boldsymbol{h}}_n-\hat{\underline{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}}-\|\underline{\boldsymbol{h}}_{n+1}-\hat{\underline{\boldsymbol{h}}}\|^2_{\underline{\boldsymbol{G}}}.\end{aligned} \tag{27}$$

Taking limits in (27) and recalling (22) gives us

$$\begin{aligned}&\lim_{n\longrightarrow\infty}\|\boldsymbol{P}_n\underline{\boldsymbol{h}}_n-\underline{\boldsymbol{h}}_{n+1}\|^2_{\underline{\boldsymbol{G}}}=0\\&\qquad\iff\lim_{n\longrightarrow\infty}\sum_{k\in\mathcal{N}}\|\boldsymbol{\phi}_{k,n}-\boldsymbol{h}_{k,n+1}\|^2_{\boldsymbol{G}}=0.\end{aligned} \tag{28}$$

Fix an arbitrary point $\boldsymbol{v}\in S_{k,j},\forall k\in\mathcal{N},\ \forall j\in\mathcal{J}_n$. Then from the triangle inequality we have

$$\begin{aligned}\|\boldsymbol{h}_{k,n+1}-\boldsymbol{v}\|_{\boldsymbol{G}}&\leq\|\boldsymbol{h}_{k,n+1}-\boldsymbol{\phi}_{k,n}\|_{\boldsymbol{G}}\\&\quad+\|\boldsymbol{\phi}_{k,n}-\boldsymbol{v}\|_{\boldsymbol{G}}\Rightarrow\\\inf_{\boldsymbol{v}\in S_{k,j}}\|\boldsymbol{h}_{k,n+1}-\boldsymbol{v}\|_{\boldsymbol{G}}&\leq\|\boldsymbol{h}_{k,n+1}-\boldsymbol{\phi}_{k,n}\|_{\boldsymbol{G}}\\&\quad+\inf_{\boldsymbol{v}\in S_{k,j}}\|\boldsymbol{\phi}_{k,n}-\boldsymbol{v}\|_{\boldsymbol{G}}\Rightarrow\\\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{h}_{k,n+1},S_{k,j})&\leq\|\boldsymbol{h}_{k,n+1}-\boldsymbol{\phi}_{k,n}\|_{\boldsymbol{G}}\\&\quad+\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{\phi}_{k,n},S_{k,j}).\end{aligned} \tag{29}$$

If we take limits in (29), from (26) and (28), it can be seen that

$$\begin{aligned}&\lim_{n\longrightarrow\infty}\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{h}_{k,n+1},S_{k,j})=0,\ \forall k\in\mathcal{N},\forall j\in\mathcal{J}_n\iff\\&\lim_{n\longrightarrow\infty}\sum_{j\in\mathcal{J}_n}\mathrm{d}_{(\boldsymbol{G})}(\boldsymbol{h}_{k,n+1},S_{k,j})=0,\ \forall k\in\mathcal{N}.\end{aligned} \tag{30}$$

The definitions of the distance function and the projection operator, yield

$$\begin{aligned}\mathrm{d}(\boldsymbol{h}_{k,n+1},S_{k,j})&=\|\boldsymbol{h}_{k,n+1}-P_{S_{k,j}}(\boldsymbol{h}_{k,n+1})\|\\&\leq\|\boldsymbol{h}_{k,n+1}-P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{h}_{k,n+1})\|.\end{aligned} \tag{31}$$

Nevertheless, the Rayleigh-Ritz theorem implies [44]$\forall\boldsymbol{h}\in\mathbb{R}^m:\|\boldsymbol{h}\|\leq\tau^{-\frac{1}{2}}_{\min}\|\boldsymbol{h}\|_{\boldsymbol{G}}$, where $\tau_{\min}$ is the smallest eigenvalue of $\boldsymbol{G}$. Combining this fact as well as (31) we obtain

$$\begin{aligned}\mathrm{d}(\boldsymbol{h}_{k,n+1},S_{k,j})&\leq\|\boldsymbol{h}_{k,n+1}-P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{h}_{k,n+1})\|\\&\leq\tau^{-\frac{1}{2}}_{min}\|\boldsymbol{h}_{k,n+1}-P^{(\boldsymbol{G})}_{S_{k,j}}(\boldsymbol{h}_{k,n+1})\|_{\boldsymbol{G}}\\&\rightarrow 0,\ n\rightarrow\infty,\ \forall k\in\mathcal{N},\end{aligned} \tag{32}$$

where the limit holds from (30). From the previous, it is not difficult to obtain that

$$\lim_{n\longrightarrow\infty}\max\{\mathrm{d}(\boldsymbol{h}_{k,n+1},S_{k,j}):j\in\mathcal{J}_n\}=0,$$

which completes our proof. ■

Asymptotic Consensus: In [37] it has been proved, that the algorithmic scheme achieves asymptotic consensus, i.e., $\|\boldsymbol{h}_{k,n}-\boldsymbol{h}_{l,n}\|\rightarrow 0,\ n\rightarrow\infty,\ \forall k,l\in\mathcal{N}$ if and only if

$$\lim_{n\longrightarrow\infty}\|\underline{\boldsymbol{h}}_n-P_{\mathcal{O}}(\underline{\boldsymbol{h}}_n)\|=0. \tag{33}$$

Let Assumptions (a), (b), (c), (d), hold true. We define the following quantity

$$\underline{\boldsymbol{\epsilon}}_n:=\underline{\boldsymbol{h}}_{n+1}-\boldsymbol{P}_n\underline{\boldsymbol{h}}_n. \tag{34}$$

Obviously from (28)

$$\lim_{n\longrightarrow\infty}\underline{\boldsymbol{\epsilon}}_n=\mathbf{0}. \tag{35}$$

Now, if we rearrange the terms in (34) and if we iterate the resulting equation, we have:

$$\begin{aligned}\underline{\boldsymbol{h}}_{n+1}&=\boldsymbol{P}_n\underline{\boldsymbol{h}}_n+\underline{\boldsymbol{\epsilon}}_n\\&=\boldsymbol{P}_n\boldsymbol{P}_{n-1}\underline{\boldsymbol{h}}_{n-1}+\boldsymbol{P}_n\underline{\boldsymbol{\epsilon}}_{n-1}+\underline{\boldsymbol{\epsilon}}_n=\cdots\\&=\prod_{i=1}^{n}\boldsymbol{P}_i\underline{\boldsymbol{h}}_0+\sum_{j=1}^{n}\prod_{l=0}^{n-j}\boldsymbol{P}_{n-l}\underline{\boldsymbol{\epsilon}}_{j-1}+\underline{\boldsymbol{\epsilon}}_n.\end{aligned}$$

If we left-multiply the previous equation by $(\boldsymbol{I}_{Km}-\boldsymbol{B}\boldsymbol{B}^T)$, where $\boldsymbol{I}_{Km}$ is the $Km\times Km$ identity matrix, and follow similar steps as in [37, Lemma 2], it can be verified that $\lim_{n\longrightarrow\infty}\|\left(\boldsymbol{I}_{Km}-\boldsymbol{B}\boldsymbol{B}^T\right)\underline{\boldsymbol{h}}_{n+1}\|=0$ which completes our proof. ■

Strong Convergence: We will prove, that under assumptions (a), (b), (c), (d), (e), $\lim_{n\rightarrow\infty}\underline{\boldsymbol{h}}_n=\hat{\underline{\boldsymbol{h}}}_*, \hat{\underline{\boldsymbol{h}}}_*\in\mathcal{O}$. Recall that the projection operator, of an arbitrary vector $\underline{\boldsymbol{h}}\in\mathbb{R}^{Km}$ onto the consensus subspace equals to $P_{\mathcal{O}}(\underline{\boldsymbol{h}})=\boldsymbol{B}\boldsymbol{B}^T\underline{\boldsymbol{h}}, \forall\underline{\boldsymbol{h}}\in\mathbb{R}^{Km}$. Taking into consideration Assumption (e) together with (16), from [18, Lemma 1], we have that there exists $\hat{\underline{\boldsymbol{h}}}_*\in\mathcal{O}$ such that

$$\lim_{n\rightarrow\infty}P_{\mathcal{O}}(\underline{\boldsymbol{h}}_n)=\hat{\underline{\boldsymbol{h}}}_*. \tag{36}$$

Now, exploiting the triangle inequality we have that

$$\|\underline{\boldsymbol{h}}_n-\hat{\underline{\boldsymbol{h}}}_*\|\leq\|\underline{\boldsymbol{h}}_n-P_{\mathcal{O}}(\underline{\boldsymbol{h}}_n)\|+\|\hat{\underline{\boldsymbol{h}}}_*-P_{\mathcal{O}}(\underline{\boldsymbol{h}}_n)\|\rightarrow 0,\ n\rightarrow\infty \tag{37}$$

where this limit holds from (33) and (36). The proof is complete since (37) implies that $\lim_{n\rightarrow\infty}\underline{\boldsymbol{h}}_n=\hat{\underline{\boldsymbol{h}}}_*$. ■

REFERENCES

[1] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[3] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, 2010.

[4] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[5] E. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[6] 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, 2011.

[7] Y. Chen, Y. Gu, and A. Hero, "Sparse LMS for system identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 3125–3128.

[8] D. Angelosante, J. Bazerque, and G. Giannakis, "Online adaptive estimation of sparse signals: where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, 2010.

[9] 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, 2008.

[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.

[11] 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, 2011.

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

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

[14] 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, 2009.

[15] 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, 2010.

[16] J. Mota, J. Xavier, P. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, 2012.

[17] P. L. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, no. 2, pp. 182–208, 1993.

[18] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Function. Anal. Optimiz.*, vol. 25, no. 7&8, pp. 593–617, 2004.

[19] K. Slavakis, I. Yamada, and N. Ogura, "The adaptive projected subgradient method over the fixed point set of strongly attracting nonexpansive mappings," *Numer. Function. Anal. Optimiz.*, vol. 7, no. 8, pp. 905–930, 2006.

[20] 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, 2011.

[21] M. Yukawa and I. Yamada, "A unified view of adaptive variable-metric projection algorithms," *EURASIP J. Adv. Signal Process.*, vol. 2009, pp. 2–2, 2009.

[22] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508–518, Sep. 2000.

[23] J. Benesty and S. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. Acoust, Speech, Signal Process.*, 2002, pp. 1881–1884.

[24] R. Baraniuk, "Compressive sensing," *Lecture Notes in IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 118–120, 2007.

[25] A. Maleki and D. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330–341, 2010.

[26] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *IEEE Int. Conf. Acoust. Speech Signal Process.*, Dallas, TX, 2010, pp. 3734–3737.

[27] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An adaptive greedy algorithm with application to nonlinear communications," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 2998–3007, 2010.

[28] K. Slavakis and I. Yamada, "The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning," 2011 [Online]. Available: http://arxiv.org/abs/1008.5231, submitted for publication

[29] P. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. New York: Springer-Verlag, 2008.

[30] L. Gubin, B. Polyak, and E. Raik, "The method of projections for finding the common point of convex sets* 1," *USSR Computat. Math. Math. Phys.*, vol. 7, no. 6, pp. 1–24, 1967.

[31] 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 Computat. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, 1967.

[32] P. Huber and E. Ronchetti, *Robust Statistics*. New York: Wiley, 2009.

[33] M. Yukawa and I. Yamada, "Set-theoretic adaptive filtering based on data-driven sparsification," *Int. J. Adapt. Contr. Signal Process.* , vol. 25, pp. 707–722.

[34] A. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ: Wiley, 2003.

[35] M. Yukawa, "Krylov-proportionate adaptive filtering techniques not limited to sparse systems," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 927–943, 2009.

[36] 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, 2009.

[37] 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, 2009.

[38] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off communications bandwidth with accuracy in adaptive diffusion networks," *Proc. IEEE ICASSP*, pp. 2048–2051, May 2011.

[39] S. Haykin, *Adaptive Filter Theory*. India: Pearson Education, 2008.

[40] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. New York: Athena Scientific, 2003.

[41] J. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms: Fundamentals*. New York: Springer, 1993.

[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[43] M. Yukawa, K. Slavakis, and I. Yamada, "Adaptive parallel quadratic-metric projection algorithms," *IEEE Trans. Audio, Speech, Language Process.*, vol. 15, no. 5, pp. 1665–1680, 2007.

[44] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

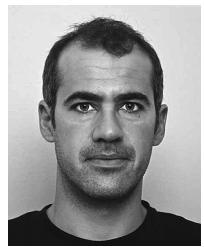
Symeon Chouvardas (S'11) received the B.Sc. degree from the Department of Informatics and Telecommunications, University of Athens, Athens, Greece, in 2008. He is currently working towards the Ph.D. degree in the Department of Informatics and Telecommunications, University of Athens.

His research interests are in adaptive filter 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.

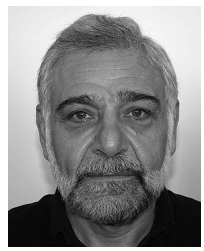
From 2004 to 2006, he was with TokyoTech as a Japan Society for the Promotion of Science postdoctoral fellow, and from 2006 to 2007, he was a postdoctoral fellow with the Department of Informatics and Telecommunications, University of Athens, Athens, Greece, under the ENTER Program. Since September 2007, 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.



Yannis Kopsinis (M'04) received the B.Sc. degree from the Department of Informatics and Telecommunications, University of Athens, Greece, in 1998 and the Ph.D. degree from the same department in 2003.

From January 2004 to December 2005, he was a research fellow with the Institute for Digital Communications, School of Engineering and Electronics, the University of Edinburgh, U.K. From January 2006 to September 2009, he was a senior researcher in the same department. He is currently a Ramon Y. Cajal fellow in the University of Granada, Spain. His current research interests include adaptive signal processing, time-frequency analysis, and online compressed sensing.



Sergios Theodoridis (F'10) is currently Professor of Signal Processing and Communications in the Department of Informatics and Telecommunications of the University of Athens, Greece. 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 coauthor of the best selling book *Pattern Recognition* (Academic Press, 4th ed., 2008), the coauthor of the book *Introduction to Pattern Recognition: A MATLAB Approach* (Academic Press, 2009), and the coauthor of three books in Greek, two of them for the Greek Open University.

Dr. Theodoridis is the coauthor 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 (SP) Society Distinguished Lecturer. He was the General Chairman of EUSIPCO 1998, 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 member of the Board of Directors of COSMOTE (the Greek mobile phone operating company) for four years. He is Fellow of IET, a Corresponding Fellow of RSE, and a Fellow of EURASIP.