A Class of Diffusion Proportionate Subband Adaptive Filters for Sparse System Identification over Distributed Networks

Ahmad Pouradabi, Amir Rastegarnia*, Sajad Zandi, Wael M. Bazzi, and Saeid Sanei

Abstract

This paper aims to extend the proportionate adaptation concept to the design of a class of diffusion normalised subband adaptive filter (DNSAF) algorithms. This leads to four extensions of the algorithm associated with different step-size variation, namely, diffusion proportionate normalised subband adaptive filter (DPNSAF), diffusion μ-law PNSAF (DMPNSAF), diffusion improved PNSAF (DIPNSAF) and diffusion improved IPNSAF (DIIPNSAF). Subsequently, steady-state performance, stability conditions and computational complexity of the proposed algorithms are investigated. For each extension the performance has been evaluated using both real and simulated data, where the outcomes demonstrate the accuracy of the theoretical expressions and effectiveness of the proposed algorithms.

Index Terms

Diffusion, distributed estimation, subband adaptive algorithm, sparse system identification, proportionate.

I. INTRODUCTION

IN many networking applications, the ultimate goal of the network can be posed as a solution to distributed estimation problem [1], [2], [3]. Three common strategies for this problem are incremental [4], [5], consensus [6], [7], [8], and diffusion [9], [10], [11], [12], [13], [14], [15], [16]. Among the available solutions, this paper focuses on the diffusion-based adaptive networks, since in comparison with the consensus-based methods, they have lower steady-state estimation error and faster convergence rate [17]. Moreover, compared to the incremental based solutions, they are robust to link and node failures. In diffusion based algorithms, the network nodes communicate with their immediate neighbours, defined by the network topology. Every node exchanges its predictions with its neighbors and fuses the estimated values using a linear combination. Different combination policies have been reported in previous research such as Metropolis or maximum-degree rules [11]. In the diffusion networks, the information diffuses more rapidly in the network.

Manuscript received 2020.

A. Pouradabi, Shahid Rajaee Teacher Training University, Tehran, Iran.

* A. Rastegarnia and S. Zandi, Malayer University, Malayer 65719-95863, Iran (email: rastegarnia@malayeru.ac.ir) (Corresponding Author).

W. M. Bazzi, American University in Dubai, UAE.

S. Sanei, Nottingham Trent University, Clifton Lane, Nottingham, U.K.

A. Related Work

Based on the diffusion strategies and by extending previously proposed adaptive filter algorithms, various distributed estimation algorithms have been derived. These include diffusion least mean square (DLMS) [9], diffusion least mean fourth (DLMF) [18], diffusion affine projection algorithm (DAPA) [19] and diffusion recursive least square (DRLS) [20], [21]. Among these algorithms, DLMS is more common because of its simplicity and ease of implementation. For improving the convergence rate of LMS algorithms, a new family of adaptive filters called subband adaptive filter (SAF) [22], [23], [24] has been proposed. SAF performs well in environments with highly correlated input signals. Fortunately, the diffusion mode of these algorithms, called diffusion normalised subband adaptive filter (DNSAF) has been proposed [25], [26] which has a better convergence rate than the conventional DLMS algorithm.

Diffusion algorithms have been intensively studied for identifying dispersive systems, where the energy is distributed uniformly among all tap-weights [10]. Here, we consider the unknown sparse vector identification problem, whose power is concentrated on only a few taps with other taps having approximately zero power. Sparsity is a characteristic which can be found in both natural and man-made systems [27], [28]. Some of its real-world applications are data compression, acoustic echo cancellation, digital TV transmission, multipath channels and underwater acoustic communications [29], [30], [31], [32], [33]. The design of such algorithms has become a challenging and attractive research agenda recently, where it is shown that through considering the sparsity of a system, the estimation performance can be further improved.

Through some criteria and using the idea that each filter coefficient can be adjusted by assigning different learning rate and even proportional to the magnitude of measured coefficient, in [34], the proportionate normalised least mean square (PNLMS) algorithm has been proposed. PNLMS algorithm exhibits a rapid convergence in the initial iterations, but after the initial phase, its convergence slows down. This concept is called proportionate adaptation and is extended to various adaptive filters. In practice, it is seen that the proportionate adaptive filters have better performance in identifying sparse-type systems than the conventional ones. In [35], an improved PNLMS (IPNLMS) has been presented in which the applied rule switches automatically from one algorithm to another. Another version of PNLMS termed μ-law PNLMS (MPNLMS) has been introduced in [36] to have a faster convergence rate over the entire adaptation procedure until approaching the steady-state condition. This algorithm outperforms the PNLMS algorithm but is computationally slightly more complex. The improved IPNLMS (IIPNLMS) presented in [37] follows one of the two methods depending on the coefficients being active or passive.

B. Contributions

This paper aims to develop efficient and fast algorithms for diffusion-based techniques, which are able to estimate a sparse parameter vector of interest by processing the streaming data. To this end, by extending the method of proportionate adaptation to distributed networks, a family of diffusion proportionate NSAF algorithms, namely diffusion proportionate NSAF (DPNSAF), diffusion μ-law PNSAF (DMPNSAF), diffusion improved PNSAF (DIPNSAF) and diffusion improved IPNSAF (DIIPNSAF) are proposed. In summary, the contributions of this paper are as follows:

- A family of diffusion proportionate NSAF algorithms is proposed which outperform the DNSAF algorithm when the estimated optimum weights are sparse.

TABLE I
SYMBOLS AND THEIR DESCRIPTIONS

Symbol	Description
$(\cdot)^{\intercal}$	Matrix transposition
$\|\cdot\|^2$	Squared Euclidean norm
$\mathrm{vec}\{\mathbf{x}\}$	Makes a diagonal matrix with $\mathbf{x}$
$\|\mathbf{x}\|^2_{\mathbf{\Sigma}}$	Weighted norm, $\mathbf{x}^*\mathbf{\Sigma}\mathbf{x}$
$\mathrm{tr}(\mathbf{X})$	Trace of matrix $\mathbf{X}$
$\mathbb{E}[\cdot]$	Statistical expectation
$\mathrm{bvec}\{\cdot\}$	block vectorization operator
$\mathrm{col}\{a, b\}$	column vector formed by stacking a and b
$\otimes$	Kronecker product
$\odot$	Block Kronecker product
$\mathrm{Blkdiag}\{\mathbf{A}, \mathbf{B}\}$	block diagonal matrix with entries $\mathbf{A}$ and $\mathbf{B}$
$[\mathbf{x}]_t$	tth element of vector $\mathbf{x}$
$\mathbf{I}$	identity matrix

- Different aspects of the proposed algorithms, including the complexity analysis, steady-state performance, stability conditions and transient behavior are analyzed and discussed.
- Through several computer simulations, the performance of each of the proposed algorithms is evaluated for system identification. Real sparse data which is a channel impulse response (FIR model) obtained from a digital microwave radio system is used here.

In this paper, normal lowercase letters are used to denote scalars while bold lowercase letters for used for vectors. A list of symbols is given in Table I.

II. SYSTEM MODEL

Consider a connected network with K nodes denoted by a set $\mathcal{K} = \{1, 2, \cdots, K\}$. Each node k can only collaborate and share information with its immediate neighbors (denoted by $\mathcal{N}_k$). The noisy measurement of each node $k \in \mathcal{K}$ at time instant i is modeled via linear regression as

$$d_k(i) = \mathbf{u}_k^{\intercal}(i)\mathbf{w}^o + \eta_k(i) \tag{1}$$

where $\mathbf{w}^o$ denotes the optimum M-dimensional (unknown) model parameter vector, $\eta_k(i)$ is the zero-mean independent and identically distributed (i.i.d.) measurement noise with variance $\sigma^2_{\eta,k}$, and $\mathbf{u}_k(i) \triangleq [u_k(i), u_k(i-1)\ldots, u_k(i-M+1)]^{\intercal}$ represents an M-dimensional input regressor vector.

To proceed, it assumed that the data satisfy the following assumptions:

Assumption 1.

(i) Both $\{\mathbf{u}_k(i)\}$ and $\{\eta_k(i)\}$ are zero-mean stationary processes.

(ii) $\{\eta_k(i)\}$ is statistically independent of $\mathbf{u}_\ell(j)$ for all $k \neq \ell$ and $i \neq j$.

(iii) The regressor covariance matrix is positive-definite $\mathbf{R}_{\mathbf{u},k} = \mathbb{E}[\mathbf{u}_k^{\intercal}(i)\mathbf{u}_k(i)] > 0$.

TABLE II
SOME OF THE PROPOSED COMBINATION RULES (n_k IS USED TO SHOW THE NUMBER OF NODES NEIGHBORING NODE k, $\ell \neq k$ AND $\ell \in \mathcal{N}_k$)

Method	Employed Rule
Relative degree [20]	$c_{\ell,k} = n_\ell / \sum_{i \in \mathcal{N}_k} n_i$
Maximum degree [40]	$c_{\ell,k} = 1/K$
Uniform [41]	$c_{\ell,k} = 1/n_k$
Metropolis [42]	$c_{\ell,k} = 1/max\,\{n_\ell, n_k\}$
Laplacian [42]	$c_{\ell,k} = 1/n_{max}$

The goal of each node is to calculate an estimate of $\mathbf{w}^o$ by solving the following problem

$$J(\mathbf{w}) = \sum_{k=1}^{K} \mathbb{E}\big[|d_k(i) - \mathbf{u}_k^{\mathsf{T}}(i)\mathbf{w}|^2\big] \tag{2}$$

A fully distributed algorithm, which can be used to solve (2) is the diffusion LMS [38], [39] which involves two adaptation and combination phases. In this algorithm, each node updates its local estimate $\mathbf{w}_k(i)$ at each time step i according to the following iterative scheme:

$$\begin{aligned} \boldsymbol{\phi}_k(i) &= \mathbf{w}_k(i-1) + \mu_k e_k(i)\mathbf{u}_k(i), \quad \text{(adaptation)} \\ \mathbf{w}_k(i) &= \sum_{\ell \in \mathcal{N}_k} c_{\ell,k}\boldsymbol{\phi}_\ell(i), \quad \text{(combination)} \end{aligned} \tag{3}$$

where is a set of non-negative real values to weight the combination of local estimates of the neighboring nodes, $\mu_k > 0$ is a suitably chosen step-size parameter and $e_k(i)$ denotes the estimation error signal defined as

$$e_k(i) = d_k(i) - \mathbf{w}_k^{\mathsf{T}}(i)\mathbf{u}_k(i) \tag{4}$$

Various rules have been proposed ~~There have been proposed various rules~~ for selecting the combination weights, taking into account the degree of each node. In Table II some of these methods have been shown. While other options are possible, i~~I~~n this paper the M~~m~~etropolis rule has been chosen due to ~~considering~~ its promising results in previous studies [20]. A thorough study of the diffusion LMS algorithm has been reported in [39], [11].

A. DNSAF Algorithm

Unlike the update equation in the DLMS algorithm, a DNSAF algorithm employs a subband adaptive filtering structure [24] in the adaptation step. In this algorithm, first the input signal $\mathbf{u}_k(i)$ is split into N subband signals $\hat{u}_k(i)$, by means of N analysis filters $\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_{N-1}$. Then, each subband signal $\hat{u}_k(i)$ is sub-sampled to a lower rate commensurating with the associated bandwidth.

Remark 1. In order to distinguish between the original sequence and sub-sampled signal, the time index for the original sequence is denoted by i, and for the sub-sampled signal by n.

At every node k, the sth decimated input vector is given by

$$\mathbf{u}_{k,s}(n) = [\hat{u}_{k,s}(nN), \hat{u}_{k,s}(nN-1), \cdots, \hat{u}_{k,s}(nN-M+1)]^{\mathsf{T}}$$

The decimated desired signal is expressed as

$$d_{k,s}(n) = \mathbf{u}_{k,s}^{\mathsf{T}}(n)\mathbf{w}^o + \eta_{k,s}(n), \tag{5}$$

where $\eta_{k,s}(n)$ is a sub-sampled additive noise. DNSAF algorithm has an update equation of the form

$$\begin{aligned}\boldsymbol{\phi}_k(n) &= \mathbf{w}_k(n-1) + \mu_k \sum_{s=1}^{N} \frac{\mathbf{u}_{k,s}(n)}{\epsilon + \|\mathbf{u}_{k,s}(n)\|^2} \left\{d_{k,s}(n) - \mathbf{w}_k(n)^\intercal \mathbf{u}_{k,s}(n)\right\}, \quad \text{(adaptation)} \\ \mathbf{w}_k(n) &= \sum_{\ell \in \mathcal{N}_k} c_{\ell,k} \boldsymbol{\phi}_\ell(n), \quad \text{(combination)} \end{aligned} \tag{6}$$

at node k and time i. ϵ is small value to prevent division by zero. Since the nodes exchange data with their neighbors and their current update depends on their previous estimates, performance analysis of the entire network is the ultimate objective. To see the role of inter-node dependence and cooperation and network topology on overall performance, investigating the whole network would be beneficial. If we combine the two equations ~~of~~ in (6) into one and extend it to ~~cover whole~~ the entire network, then , the network update equation can be derived as [DELETED: the *network update equation* for the DNSAF algorithm is given by] [43]

$$\mathbf{w}(n) = \mathbf{G}\mathbf{w}(n-1) + \mathbf{G}\mathbf{D}\boldsymbol{\mathcal{U}}(n)[\epsilon + \boldsymbol{\mathcal{U}}^\intercal(n)\boldsymbol{\mathcal{U}}(n)]^{-1}\mathbf{e}(n) \tag{7}$$

where

$$\mathbf{w}(n) = \text{col}\{\mathbf{w}_1(n), \mathbf{w}_2(n), \cdots, \mathbf{w}_K(n)\} \tag{8}$$

$$\mathbf{G} = \mathbf{C}^\intercal \otimes \mathbf{I}_M \tag{9}$$

$$\mathbf{D} = \text{Blkdiag}\{\mu_1 \mathbf{I}_M, \mu_2 \mathbf{I}_M, \cdots, \mu_K \mathbf{I}_M\} \tag{10}$$

$$\boldsymbol{\mathcal{U}}(n) = \text{Blkdiag}\{\mathbf{U}_1(n), \mathbf{U}_2(n), \cdots, \mathbf{U}_K(n)\}, \tag{11}$$

$$\mathbf{e}(n) = \mathbf{d}(n) - \boldsymbol{\mathcal{U}}^\intercal(n)\mathbf{w}(n-1) \tag{12}$$

where $\mathbf{C}$ is a $K \times K$ whose kth column is $c_{l,k}$, $l = 1, ..., K$ and $\mathbf{U}_k(n)$ and $\mathbf{d}(n)$ are defined as:

$$\mathbf{U}_k(n) = [\mathbf{u}_{k,1}(n), \mathbf{u}_{k,2}(n), \cdots, \mathbf{u}_{k,N}(n)], \quad (M \times N)$$

$$\mathbf{d}(n) = [d_{1,1}(n), \cdots, d_{1,N}(n), \cdots, d_{K,1}(n) \cdots, d_{K,N}(n)]^\intercal.$$

Using the above definitions, (5) can be rewritten as

$$\mathbf{d}(n) = \boldsymbol{\mathcal{U}}^\intercal(n)\mathbf{W}^o + \boldsymbol{\eta}(n) \tag{13}$$

In (13) we have $\mathbf{W}^o = \mathbf{A}\mathbf{w}^o$ and

$$\mathbf{A} = \text{col}\{\mathbf{I}_M, \mathbf{I}_M, \cdots, \mathbf{I}_M\}$$

$$\boldsymbol{\eta}(n) = [\eta_{1,1}(n), \cdots, \eta_{1,N}(n), \cdots, \eta_{K,1}(n) \cdots, \eta_{K,N}(n)]^\intercal$$

A detailed analysis of DNSAF algorithm (including steady-state and convergence analysis) has been reported in [43].

III. PROPOSED ALGORITHMS

In this section, we use the proportionate adaptation framework to exploit the sparseness of unknown vector parameter for obtaining diffusion adaptive networks with improved adaptation rate. The general idea of proportionate adaptation is to adjust the learning step-size according to the magnitude of the filter weight taps. Since the optimum magnitude of the filter taps are unknown, the current estimated magnitude of the weights are used in practice.

Therefore, the adaptation gain is distributed proportionately among filter coefficients, larger gains will be assigned to larger weights and vice vesra, which explains faster convergence rate of this method than the conventional algorithms.

The general form of the proposed algorithms is given by

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{G}\mathbf{w}(n-1)\\ &\quad + \mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)[\epsilon + \boldsymbol{\mathcal{U}}^{\mathsf{T}}(n)\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)]^{-1}\mathbf{e}(n) \qquad (14)\end{aligned}$$

where $\boldsymbol{\Theta}(n)$ is a $KM \times KM$ block diagonal matrix defined as

$$\boldsymbol{\Theta}(n) = \text{Blkdiag}\{\mathbf{B}_1(n), \mathbf{B}_2(n), \cdots, \mathbf{B}_K(n)\} \qquad (15)$$

with

$$\mathbf{B}_k(n) = \text{diag}\{\theta_{k,1}(n), \theta_{k,2}(n), \cdots, \theta_{k,M}(n)\}\,. \qquad (16)$$

$\boldsymbol{\Theta}(n)$ is a gain distributor among individual coefficients of the filter and is proportional to the magnitude of the related impulse-response taps of unknown system $\mathbf{w}^o$. Therefore, larger/smaller taps receive larger/smaller gains which speed up the convergence. In the sequel, we introduce different choices for $\mathbf{B}_k(n)$ which in turn, result in different algorithms.

A. Diffusion Proportionate NSAF Algorithm

Inspired by [44] in the DPNSAF algorithm, the step-size at each node is assigned in a way that the coefficients with larger magnitudes are considered to assign large step-sizes, and vice versa. In this scheme, each $\theta_{k,m}(n)$ is given by

$$\theta_{k,m}(n) = \frac{\alpha_{k,m}(n)}{\frac{1}{M}\sum_{j=1}^{M}\alpha_{k,j}(n)} \qquad (17)$$

where $\alpha_{k,m}(n)$ is defined for every $m = 1, \cdots, M$ as

$$\alpha_{k,m}(n) = \max\{(\rho\max\{\delta, \|\mathbf{w}_k(n)\|_\infty\}), |[\mathbf{w}_k(n)]_m|\} \qquad (18)$$

with $\rho > 0$ and $\delta > 0$. Using the algorithm, the adaptation gains are distributed among the taps. It should be noted that, larger taps are accentuated to accelerate their convergence, leading to a rapid convergence at initial steps.

B. Diffusion μ-law Proportionate NSAF

Subsequent to the initial convergence, the DPNSAF algorithm eases back down significantly. The procedure of calculating the gains to control the step-size distribution among the DPNSAF coefficients is not in accordance with any optimization criterion. In [36] the MPNLMS algorithm has been reported which has faster convergence across the entire adaptation process. Following a similar approach, the diffusion μ-law PNSAF algorithm can be obtained by defining $\theta_{k,m}(n)$ as:

$$\theta_{k,m}(n) = \frac{\beta_{k,m}(n)}{\frac{1}{M}\sum_{j=1}^{M}\beta_{k,j}(n)} \qquad (19)$$

where

$$\beta_{k,m}(n) = \max\{(\rho\cdot\max\{\delta, \|\mathbf{f}_k(n)\|_\infty\}), [\mathbf{f}_k(n)]_m\} \qquad (20)$$

with $\mathbf{f}_k(n) = [f_{k,1}(n), \cdots, f_{k,M}(n)]$ where

$$f_{k,m}(n) = \ln(1 + \nu[\mathbf{w}_k(n)]_m) \tag{21}$$

in which ν is an infinitesimal positive value. The assigned value may be selected according to the additive noise level.

C. Diffusion Improved Proportionate NSAF

In [45], the improved PNLMS algorithm has been provided by employing a rule to use a combination of proportionate and non-proportionate adaptations. It can be considered as a mixture of NLMS and PNLMS algorithms. Therefore, applying it to both sparse and dispersive systems would be favourable. Following the same modification, the DIPNSAF algorithm is proposed and the main diagonal elements of $\theta_{k,m}(n)$ are defined as:

$$\theta_{k,m}(n) = \frac{1-\gamma}{2M} + (1+\gamma)\frac{|[\mathbf{w}_k(n)]_m|}{2\|\mathbf{w}_k(n)\|_1 + \zeta} \tag{22}$$

in which $\|\cdot\|_1$ denotes ℓ_1-norm and ζ is an infinitesimal positive coefficient, considered to prevent any undefined value and $\gamma \in [-1, 1]$. For $\gamma = -1$, the DIPNSAF and DNSAF algorithms have no differences. However, $\gamma = 1$ results in a faster convergence while higher sensitivity regarding sparsity is expected. Practicality speaking, a range of -0.5 to 0 for γ is reasonable.

D. Diffusion Improved IPNSAF

Referring to the IPNLMS algorithm, γ which represents the relative weighting between NLMS and PNLMS algorithms is the same for all the coefficients. In [37], the IIPNLMS has been proposed in which the relative weighting of proportionate and non-proportionate updating is rearranged for every coefficient separately. The corresponding adjustment is attainable by implementing two dissimilar values for γ. For large coefficients, γ is selected in such a way that non-proportional updates are weighed more heavily. On the other hand, for small coefficients, γ is selected to meet the condition in which proportionate updating is desired. Similarly, we propose a DIIPNSAF algorithm with the diagonal entries of $\theta_{k,m}(n)$ defined through a procedure as follows:

$$\theta_{k,m}(n) = \frac{1-\pi_{k,m}(n)}{2M} + (1+\pi_{k,m}(n))\frac{|[\mathbf{w}_k(n)]_m|}{2\|\mathbf{w}_k(n)\|_1 + \zeta} \tag{23}$$

where ζ is again an infinitesimal value and

$$\pi_{k,m}(n) = \begin{cases} \pi_1 & t_{k,m}(n) > g_0 \times \max\{\mathbf{t}_k(n)\} \\ \pi_2 & t_{k,m}(n) \leq g_0 \times \max\{\mathbf{t}_k(n)\} \end{cases} \tag{24}$$

In (24) $\mathbf{t}_k(n) = [t_{k,1}(n), \cdots, t_{k,M}(n)]$ and

$$t_{k,m}(n) = \max\{(\rho \cdot \|\mathbf{w}_k(n)\|_\infty), [\mathbf{w}_k(n)]_m\} \tag{25}$$

A logical value for g_0 is 0.1 [37].

TABLE III
THE COMPUTATIONAL COMPLEXITY OF DNSAF, DPNSAF, DMPNSAF, DIPNSAF AND DIIPNSAF ALGORITHMS.

Algorithm	Multiplications	Divisions	Comparisons
DNSAF	$\sum_{k=1}^{K}(M(3+\lvert\mathcal{N}_k\rvert)+3NL+1)$	K	–
DPNSAF	$\sum_{k=1}^{K}(M(4+\lvert\mathcal{N}_k\rvert)+3NL+2)$	$K(M+1)$	$2KM$
DMPNSAF	$\sum_{k=1}^{K}(M(5+\lvert\mathcal{N}_k\rvert)+3NL+2)$	$K(M+1)$	$2KM$
DIPNSAF	$\sum_{k=1}^{K}(M(5+\lvert\mathcal{N}_k\rvert)+3NL+1)$	$K(M+1)$	–
DIIPNSAF	$\sum_{k=1}^{K}(M(5+\lvert\mathcal{N}_k\rvert)+3NL+1)$	$K(M+1)$	$2KM$

IV. FURTHER INSIGHTS

A. Computational Complexity

The computational complexity of DNSAF algorithm is compared with those of the proposed proportionate algorithms for the number of multiplications, divisions and comparisons for every time iteration and node, as listed in Table III. Here, N, M and L are the number of subbands, the filter length, and the length of analysis and synthesis filters, respectively.

B. Steady-state Analysis

The steady-state performances of the proposed algorithms are analyzed through the time evaluation of $\mathbb{E}\{\|\tilde{\mathbf{w}}(n)\|^2_{\boldsymbol{\Sigma}}\}$, where $\tilde{\mathbf{w}}(n)=\mathbf{w}^o-\mathbf{w}(n)$ represents the weight-error vector and $\boldsymbol{\Sigma}$ refers to any positive-definite and Hermitian matrix. Considering the general form of (14), the update equation of weight-error vector for diffusion proportionate NSAF algorithms is expressed as:

$$\tilde{\mathbf{w}}(n)=\mathbf{G}\tilde{\mathbf{w}}(n-1)-\mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)[\epsilon\mathbf{I}+\boldsymbol{\mathcal{U}}^{\mathsf{T}}(n)\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)]^{-1}\mathbf{e}(n) \tag{26}$$

Replacing $\mathbf{e}(n)=\boldsymbol{\mathcal{U}}^{\mathsf{T}}(n)\tilde{\mathbf{w}}(n-1)+\boldsymbol{\eta}(n)$ in (26) and taking the weighted norm from both sides of the resultant expression gives

$$\|\tilde{\mathbf{w}}(n)\|^2_{\boldsymbol{\Sigma}}=\|\tilde{\mathbf{w}}(n-1)\|^2_{\boldsymbol{\Omega}}+\boldsymbol{\eta}^{\mathsf{T}}(n)\mathbf{Y}(n)\boldsymbol{\eta}(n)+\text{cross terms involving }\boldsymbol{\eta}(n) \tag{27}$$

where

$$\mathbf{Y}(n)=(\mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n))^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n) \tag{28}$$

$$\boldsymbol{\Omega}=\mathbf{G}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{G}-\mathbf{G}^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{G}\mathbf{D}\mathbf{Z}(n)-(\mathbf{G}\mathbf{D}\mathbf{Z}(n))^{\mathsf{T}}\boldsymbol{\Sigma}\mathbf{G} \tag{29}$$

with

$$\Xi(n)=[\epsilon\mathbf{I}+\boldsymbol{\mathcal{U}}^{\mathsf{T}}(n)\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)]^{-1} \tag{30}$$

$$\mathbf{Z}(n)=\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n)\boldsymbol{\mathcal{U}}^{\mathsf{T}}(n) \tag{31}$$

Taking the expectation from both sides of (27) and considering Assumption 1 yields:

$$\mathbb{E}\big[\|\tilde{\mathbf{w}}(n)\|^2_{\boldsymbol{\Sigma}}\big]=\mathbb{E}\big[\|\tilde{\mathbf{w}}(n-1)\|^2_{\boldsymbol{\Sigma}'}\big]+\mathbb{E}[\boldsymbol{\eta}^{\mathsf{T}}(n)\mathbf{Y}(n)\boldsymbol{\eta}(n)] \tag{32}$$

where

$$\boldsymbol{\Sigma}' = \mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G} - \mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G}\mathbf{D}\mathbb{E}[\mathbf{Z}(n)] - \mathbb{E}[\mathbf{Z}^{\intercal}(n)]\,\mathbf{D}^{\intercal}\mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G} + \mathbb{E}[\mathbf{Z}^{\intercal}(n)\mathbf{D}^{\intercal}\mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G}\mathbf{D}\mathbf{Z}(n)] \tag{33}$$

By applying $\operatorname{bvec}\{\cdot\}$ operator to (33) some relations are derived as follow:

$$\operatorname{bvec}\{\mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G}\} = (\mathbf{G}^{\intercal} \odot \mathbf{G}^{\intercal})\boldsymbol{\sigma} \tag{34}$$

$$\begin{aligned}\operatorname{bvec}\{\mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G}\mathbf{D}\mathbb{E}[\mathbf{Z}(n)]\} &= (\mathbb{E}[\mathbf{Z}^{\intercal}(n)] \odot \mathbf{I}_{KM}) \\ &\quad \times (\mathbf{D}^{\intercal} \odot \mathbf{I}_{KM})(\mathbf{G}^{\intercal} \odot \mathbf{G}^{\intercal})\boldsymbol{\sigma}\end{aligned} \tag{35}$$

$$\begin{aligned}\operatorname{bvec}\{\mathbb{E}[\mathbf{Z}^{\intercal}(n)]\,\mathbf{D}^{\intercal}\mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G}\} &= (\mathbf{I}_{KM} \odot \mathbb{E}[\mathbf{Z}^{\intercal}(n)]) \\ &\quad \times (\mathbf{I}_{KM} \odot \mathbf{D}^{\intercal})(\mathbf{G}^{\intercal} \odot \mathbf{G}^{\intercal})\boldsymbol{\sigma}\end{aligned} \tag{36}$$

$$\begin{aligned}&\operatorname{bvec}\{\mathbb{E}[\mathbf{Z}^{\intercal}(n)]\,\mathbf{D}^{\intercal}\mathbf{G}^{\intercal}\boldsymbol{\Sigma}\mathbf{G}\mathbf{D}\mathbf{Z}(n)\} = \\ &\qquad \mathbb{E}[\mathbf{Z}^{\intercal}(n) \odot \mathbf{Z}^{\intercal}(n)]\,(\mathbf{D}^{\intercal} \odot \mathbf{D}^{\intercal})(\mathbf{G}^{\intercal} \odot \mathbf{G}^{\intercal})\boldsymbol{\sigma}\end{aligned} \tag{37}$$

in which $\operatorname{bvec}\{\boldsymbol{\Sigma}\} = \boldsymbol{\sigma}$ and $\operatorname{bvec}\{\boldsymbol{\Sigma}'\} = \boldsymbol{\sigma}'$. Hence, by setting matrix $\mathbf{P}$ as

$$\begin{aligned}\mathbf{P} = \Big(&\mathbf{I}_{K^2M^2} - (\mathbb{E}[\mathbf{Z}^{\intercal}(n)] \odot \mathbf{I}_{KM})(\mathbf{D}^{\intercal} \odot \mathbf{I}_{KM}) \\ &- (\mathbf{I}_{KM} \odot \mathbb{E}[\mathbf{Z}^{\intercal}(n)])(\mathbf{I}_{KM} \odot \mathbf{D}^{\intercal}) \\ &+ (\mathbb{E}[\mathbf{Z}^{\intercal}(n) \odot \mathbf{Z}^{\intercal}(n)])(\mathbf{D}^{\intercal} \odot \mathbf{D}^{\intercal})\Big)(\mathbf{G}^{\intercal} \odot \mathbf{G}^{\intercal})\end{aligned} \tag{38}$$

The following relation is obtained:

$$\boldsymbol{\sigma}' = \mathbf{P}\boldsymbol{\sigma} \tag{39}$$

In addition, the second term in the r.h.s of (32) becomes:

$$\mathbb{E}[\boldsymbol{\eta}^{\intercal}(n)\mathbf{Y}(n)\boldsymbol{\eta}(n)] = \mathbf{g}^{\intercal}\boldsymbol{\sigma} \tag{40}$$

where

$$\begin{aligned}\mathbf{g} = \operatorname{bvec}\Big\{&\mathbb{E}\big[\mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n)\boldsymbol{\eta}(n) \\ &\times (\mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n)\boldsymbol{\eta}(n))^{\intercal}\big]\Big\} \\ = \mathbb{E}\big[&(\mathbf{G} \odot \mathbf{G})(\mathbf{D} \odot \mathbf{D})(\boldsymbol{\Theta}(n) \odot \boldsymbol{\Theta}(n))(\boldsymbol{\mathcal{U}}(n) \odot \boldsymbol{\mathcal{U}}(n)) \\ &\times (\Xi(n) \odot \Xi(n))(\boldsymbol{\eta}(n) \odot \boldsymbol{\eta}(n))\big]\end{aligned} \tag{41}$$

Thus, using the obtained moments, Equation (32) can be expressed in the following iterative form:

$$\mathbb{E}\big[\|\tilde{\mathbf{w}}(n)\|^2_{\boldsymbol{\sigma}}\big] = \mathbb{E}\big[\|\tilde{\mathbf{w}}(n-1)\|^2_{\mathbf{P}\boldsymbol{\sigma}}\big] + \mathbf{g}^{\intercal}\boldsymbol{\sigma} \tag{42}$$

or

$$\mathbb{E}\big[\|\tilde{\mathbf{w}}(n)\|^2_{\boldsymbol{\sigma}}\big] = \mathbb{E}\big[\|\tilde{\mathbf{w}}(0)\|^2_{\mathbf{P}^n\boldsymbol{\sigma}}\big] + \mathbf{g}^{\intercal}(\mathbf{I}_{K^2M^2} + \mathbf{P} + \cdots + \mathbf{P}^{n-1})\boldsymbol{\sigma}. \tag{43}$$

In the recent equation weighting matrix $\mathbf{\Sigma}$ is replaced by $\boldsymbol{\sigma} = \text{bvec}\{\mathbf{\Sigma}\}$ to simplify the notation. This leads to the useful result

$$\begin{aligned}\mathbb{E}\big[\|\tilde{\mathbf{w}}(n)\|^2_{\boldsymbol{\sigma}}\big] = \mathbb{E}\big[\|\tilde{\mathbf{w}}(n-1)\|^2_{\boldsymbol{\sigma}}\big] + \mathbf{g}^{\intercal}\mathbf{P}^n\boldsymbol{\sigma} \\ + \|\tilde{\mathbf{w}}(0)\|^2_{\mathbf{P}^{n-1}(\mathbf{I}-\mathbf{P})\boldsymbol{\sigma}}.\end{aligned} \tag{44}$$

Let's define the following vectors

$$\mathbf{m}_k = \text{vec}\{\text{diag}\{b_{k,K}\} \otimes \mathbf{I}_M\} \tag{45}$$

$$\mathbf{r}_k = \text{vec}\{\text{diag}\{b_{k,K}\} \otimes \mathbf{R}_{\mathbf{u},k}\} \tag{46}$$

where $b_{k,K}$ denotes a column vector of the diagonal matrix $\mathbf{I}_K$ in kth position. Substituting $\boldsymbol{\sigma} = \mathbf{r}_k$ or $\boldsymbol{\sigma} = \mathbf{m}_k$ in (43), the transient response associated with two learning curves, i.e. mean square deviation (MSD) and excess mean square error (EMSE), respectively, is estimated.

According to (43), when n approaches infinity, the final values of EMSE and MSD at each node can be calculated respectively as:

$$\text{EMSE}_k = \mathbf{g}^{\intercal}(\mathbf{I}-\mathbf{P})^{-1}\mathbf{r}_k \tag{47}$$

$$\text{MSD}_k = \mathbf{g}^{\intercal}(\mathbf{I}-\mathbf{P})^{-1}\mathbf{m}_k. \tag{48}$$

C. Transient Expressions

Replacing (45) in (44) leads to local mean-square deviation as follows

$$\text{MSD}_k(n) = \text{MSD}_k(n-1) + \mathbf{g}^{\intercal}\mathbf{P}^n\mathbf{m}_k + \|\tilde{\mathbf{w}}(0)\|^2_{\mathbf{P}^{n-1}(\mathbf{I}-\mathbf{P})\mathbf{m}_k}. \tag{49}$$

with initial condition $\text{MSD}_k(0) = \|\tilde{\mathbf{w}}(0)\|^2_{\text{diag}(b_{k,K}\otimes\mathbf{I}_M)}$. Likewise, by replacing (46) in (44) the local excess mean-square error relation can be derived

$$\text{EMSE}_k(n) = \text{EMSE}_k(n-1) + \mathbf{g}^{\intercal}\mathbf{P}^n\mathbf{r}_k + \|\tilde{\mathbf{w}}(0)\|^2_{\mathbf{P}^{n-1}(\mathbf{I}-\mathbf{P})\mathbf{r}_k}. \tag{50}$$

with initial condition $\text{EMSE}_k(0) = \|\tilde{\mathbf{w}}(0)\|^2_{\text{diag}(b_{k,K}\otimes\mathbf{R}_{\mathbf{u},k})}$.

Besides, EMSE and MSD for the entire network can also be ascertained considering the averages for MSD and EMSE across all the network nodes:

$$\text{MSD}_{\text{net}}(n) = \frac{1}{K}\sum_{k=1}^{K}\text{MSD}_k(n) \tag{51}$$

$$\text{EMSE}_{\text{net}}(n) = \frac{1}{K}\sum_{k=1}^{K}\text{EMSE}_k(n) \tag{52}$$

In the simulation results section, the accuracy of the obtained expressions is examined.

D. Mean Stability

To conduct the stability analysis, we rewrite (26) as

$$\tilde{\mathbf{w}}(n) = \mathbf{G}[\mathbf{I} - \mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n)\boldsymbol{\mathcal{U}}^{\mathsf{T}}(n)]\tilde{\mathbf{w}}(n-1) - \mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\boldsymbol{\mathcal{U}}(n)\Xi(n)\boldsymbol{\eta}(n) \tag{53}$$

Taking expectation of both sides under Assumption 1 yields

$$\mathbb{E}[\tilde{\mathbf{w}}(n)] = \mathbf{G}\boldsymbol{\mathcal{F}}\mathbb{E}[\tilde{\mathbf{w}}(n-1)] \tag{54}$$

where

$$\boldsymbol{\mathcal{F}} \triangleq \mathbf{I} - \mathbf{G}\mathbf{D}\boldsymbol{\Theta}(n)\mathbb{E}\big[\boldsymbol{\mathcal{U}}(n)\Xi(n)\boldsymbol{\mathcal{U}}^{\mathsf{T}}(n)\big] \tag{55}$$

It is obvious that the condition $\lambda_{\max}(\mathbf{G}\boldsymbol{\mathcal{F}}) \leq 1$ guarantees the stability of the proposed algorithms. From the analysis in [11], we can show that

- $\lambda_{\max}(\boldsymbol{\mathcal{F}}) \leq \lambda_{\max}(\mathbf{G}\boldsymbol{\mathcal{F}})$,
- the stability of the block-diagonal matrix $\boldsymbol{\mathcal{F}}$ is equivalent to the stability of its block-diagonal entries $\mathbf{I} - \mathbf{C}^{\mathsf{T}}\mu_k\mathbf{B}_k(n)\mathbb{E}\big[\mathbf{U}_k(n)(\epsilon\mathbf{I} + \mathbf{U}_k^{\mathsf{T}}(n)\mathbf{B}_k(n)\mathbf{U}_k(n))^{-1}\mathbf{U}_k^{\mathsf{T}}(n)\big]$

Hence, in order to guarantee the convergence of $\mathbb{E}[\tilde{\mathbf{w}}(n)]$, the following condition must be met:

$$0 < \mu_k \leq \frac{2}{\lambda_{\max}\left(\displaystyle\sum_{k=1}^{N}\mathbb{E}\big[\mathbf{U}_k(n)(\epsilon\mathbf{I} + \mathbf{U}_k^{\mathsf{T}}(n)\mathbf{B}_k(n)\mathbf{U}_k(n))^{-1}\mathbf{U}_k^{\mathsf{T}}(n)\big]\right)}. \tag{56}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of its matrix argument.

E. Mean-Square Stability

In order to derive the stability condition in the mean-square sense, we rewrite $\mathbf{P}$ in (38) as

$$\mathbf{P} = \mathcal{O}(\mathbf{G}^{\mathsf{T}} \odot \mathbf{G}^{\mathsf{T}}) \tag{57}$$

in which

$$\begin{aligned}\mathcal{O} = \mathbf{I}_{K^2M^2} &- (\mathbb{E}[\mathbf{Z}^{\mathsf{T}}(n)] \odot \mathbf{I}_{KM})(\mathbf{D}^{\mathsf{T}} \odot \mathbf{I}_{KM}) \\ &- (\mathbf{I}_{KM} \odot \mathbb{E}[\mathbf{Z}^{\mathsf{T}}(n)])(\mathbf{I}_{KM} \odot \mathbf{D}^{\mathsf{T}}) \\ &+ (\mathbb{E}[\mathbf{Z}^{\mathsf{T}}(n) \odot \mathbf{Z}^{\mathsf{T}}(n)])(\mathbf{D}^{\mathsf{T}} \odot \mathbf{D}^{\mathsf{T}})\end{aligned} \tag{58}$$

To guarantee the stability, one should ensure that the following condition is satisfied for all eigenvalues of $\mathbf{P}$:

$$|\lambda(\mathbf{P})| < 1 \tag{59}$$

through initialization of μ_k and cooperation protocol ($\mathbf{C}^{\mathsf{T}}$). Through the following procedure global stability can be guaranteed. Using Euclidean norm, we can write

$$\|\mathbf{P}\|_2 = \|\mathcal{O}(\mathbf{G}^{\mathsf{T}} \odot \mathbf{G}^{\mathsf{T}})\|_2 \leq \|\mathcal{O}\|_2 \cdot \|\mathbf{G}^{\mathsf{T}} \odot \mathbf{G}^{\mathsf{T}}\|_2 \tag{60}$$

and this can be rewritten using kronecker product [1] as

$$\begin{aligned}\|\mathcal{O}(\mathbf{G}^{\mathsf{T}} \odot \mathbf{G}^{\mathsf{T}})\|_2 &= \|\mathcal{O}.\mathcal{P}^{\mathsf{T}}(\mathbf{G}^{\mathsf{T}} \otimes \mathbf{G}^{\mathsf{T}})\mathcal{P}\|_2 \\ &\leq \|\mathcal{O}\|_2 \,.\, \|(\mathbf{G}^{\mathsf{T}} \otimes \mathbf{G}^{\mathsf{T}})\|_2 \\ &= \|\mathcal{O}\|_2 \,.\, \|\mathbf{G}^{\mathsf{T}}\|_2 \,.\, \|\mathbf{G}^{\mathsf{T}}\|_2 \end{aligned} \tag{61}$$

In this equation, $\mathcal{P}$ is an orthogonal permutation matrix. Noting (8), we can write

$$|\lambda_{\max}(\mathcal{O}.(\mathbf{G}^{\mathsf{T}} \odot \mathbf{G}^{\mathsf{T}}))| \leq |\lambda_{\max}(\mathcal{O})| \,.\, \|\mathbf{C}^{\mathsf{T}}\|_2^2 \,. \tag{62}$$

For the combination rules that produce matrice $\mathbf{C}^{\mathsf{T}}$ stochasticly, (considering above equation) we can write

$$|\lambda_{\max}(\mathcal{O}.(\mathbf{G}^{\mathsf{T}} \odot \mathbf{G}^{\mathsf{T}}))| \leq |\lambda_{\max}(\mathcal{O})| \,. \tag{63}$$

This condition is generally met, hence to ensure stablity of the network, it is sufficient to ensure $\|\mathbf{C}^{\mathsf{T}}\|_2^2 \leq 1$.

V. SIMULATION RESULTS

In order to evaluate the performances of proposed algorithms and validate the theoretical analysis, we provide different computer simulations in a setup of system identification. To this end, different sparse vectors consisting of $M = 100$ FIR prototype are considered where each one is a short cut model of a digital microwave radio channel[1]. Fig. 1 indicates the topology of a network with $K = 20$ nodes. Two input signals $\mathbf{u}_k(n)$ are considered in the modeling including

1) zero mean white Gaussian with $\mathbf{R}_{\mathbf{u},k} = \sigma_{u_k}^2 \mathbf{I}$,
2) colored Gaussian where the correlated elements are generated at each node by filtering a zero-mean white Gaussian sequence through a first-order system $\frac{1}{1-p_k z^{-1}}$. Consequently, at each studied node, a colored Gaussian signal is produced.

The node profiles of p_k (correlation index), $\sigma_{u_k}^2$ (power of regressor data), and $\sigma_{\eta_k}^2$ (noise variance) are shown in Fig. 2. The combination weights $c_{\ell,k}$ are assigned according to the Metropolis rule. The extended lapped transform (ELT) [46] is used in the simulations as the filter bank with $N = 4$. To compare the algorithm performances, normalized mean square deviation (NMSD) learning curve of the network is used which is defined as [DELETED:

$$\text{NMSD}_{\text{net}} = \frac{1}{K\|\mathbf{w}^o\|^2} \sum_{k=1}^{K} \mathbb{E}\big[\|\tilde{\mathbf{w}}(n)\|^2\big] \tag{64}$$

]

$$\text{NMSD}_{\text{net}} = \frac{1}{K\|\mathbf{w}^o\|^2} \sum_{k=1}^{K} \mathbb{E}\big[\|\tilde{\mathbf{w}}_k(n)\|^2\big] \tag{65}$$

All the modeled learning curves are extracted using ensemble averaging over hundred non-contingent trials. To make it easier to follow the empirical results of this section, all ~~parameter selection of~~ the parameters selected for the simulated algorithms are illustrated in Table IV.

Fig. 3 shows the DNSAF and DPNSAF learning curves for different values of ρ. The step-sizes are adjusted in a way that the steady-state errors have approximately equal values to make their learning curves comparable

[1] The employed radio channel coefficients are available in http://spib.linse.ufsc.br/microwave.html

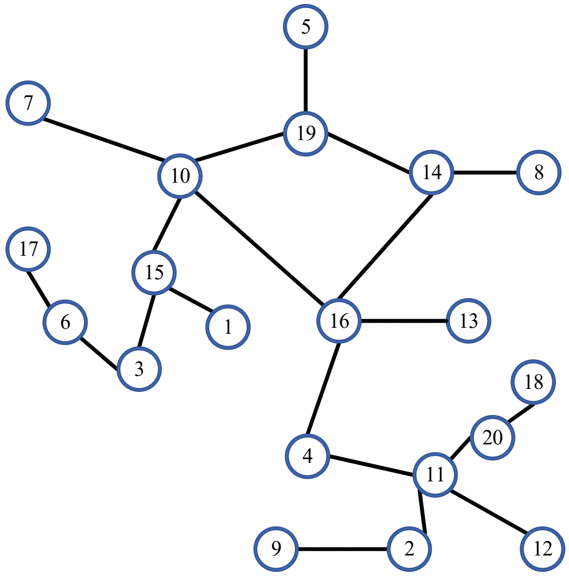

Fig. 1. The network topology with $K = 20$ nodes.

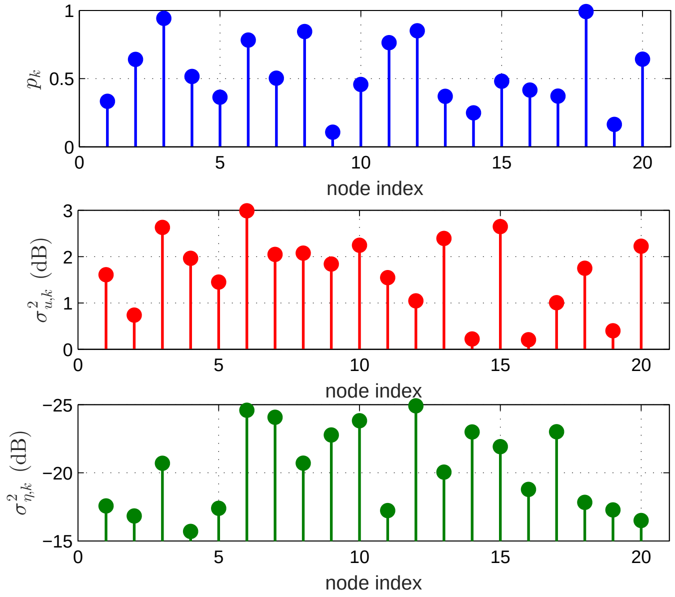

Fig. 2. Correlation index (top), input power (middle), and noise power at each node (bottom).

and we set $\delta = 0.01$. Different values are chosen for ρ, (ρ=1, 0.1, 0.025, and 0.01) in Fig. 3 for identifying the unknown system with given impulse response. It is shown that faster convergence would be achieved by reducing this parameter. However, if the parameter has a small value, slower convergence is expected during the entire adaptation. It should be noted that, a common value for ρ is $\frac{5}{M}$. Besides, relatively fast convergence is seen in the charts for ρ= 0.025, so this value is set in the conducted simulations.

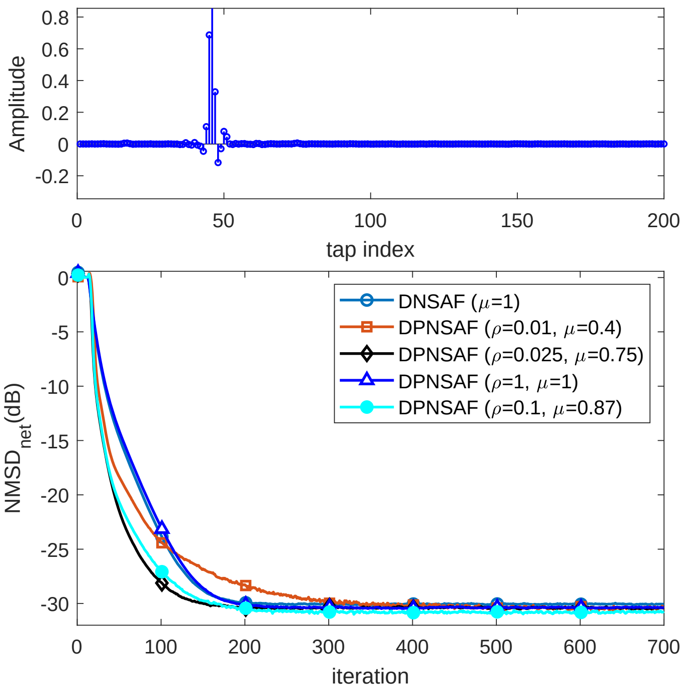

Fig. 3. The NMSD curves of DNSAF and DPNSAF algorithms with various values of ρ: (top) impulse response, (bottom) the NMSD learning curves when input regressors are white Gaussian.

TABLE IV

SUMMARY OF THE PARAMETER SELECTION FOR EMPIRICAL INVESTIGATION

Simmulation Number	Parameter Setup of Each Algorithm	Figure Number
The First	DNSAF ($\mu = 1$) DPNSAF ($\mu = 0.4,\ \rho = 0.01,\ \delta = 0.01$) DPNSAF ($\mu = 0.75,\ \rho = 0.025,\ \delta = 0.01$) DPNSAF ($\mu = 1,\ \rho = 1,\ \delta = 0.01$) DPNSAF ($\mu = 0.87,\ \rho = 0.1,\ \delta = 0.01$)	3
The Second	DPNSAF ($\mu = 0.9,\ \rho = 0.025,\ \delta = 0.01$) DNSAF ($\mu = 1.2$) DMPNSAF ($\mu = 0.9, \rho = 0.025,\ \nu = \sqrt{1000},\ \delta = 0.01$)	4
The Third	DIIPNSAF ($\mu = 0.6,\ \rho = 0.025,\ \pi_1 = -o.8,\ \pi_2 = o.8,\ g_0 = 0.1,\ \zeta = 0.001$) DNSAF ($\mu = 1$) DPNSAF ($\mu = 0.7,\ \rho = 0.025,\ \delta = 0.01$) DIPNSAF ($\mu = 0.87,\ \zeta = 0.001$)	5
The Fourth (theoretical justification)	DIIPNSAF ($\mu = 0.5,\ \rho = 0.025,\ \pi_1 = -o.8,\ \pi_2 = o.8,\ g_0 = 0.1,\ \zeta = 0.001$)	6

The learning curves of DNSAF, DPNSAF and DMPNSAF algorithms are shown in Fig. 4 for both input regression types (when input regressors are white Gaussian and when input regressors are Gaussian AR(1)) $\delta = 0.01$, $\rho = 0.025$, and $\nu = \sqrt{1000}$ are set. The results in Fig. 4 show that the DPNSAF and DMPNSAF algorithms converge faster than DNSAF. On the other hand, the DPNSAF algorithm converges faster initially, while the DMPNSAF converges more rapidly during the entire adaptation. The NMSD learning curves of DNSAF, DPNSAF, DIPNSAF and DIIPNSAF

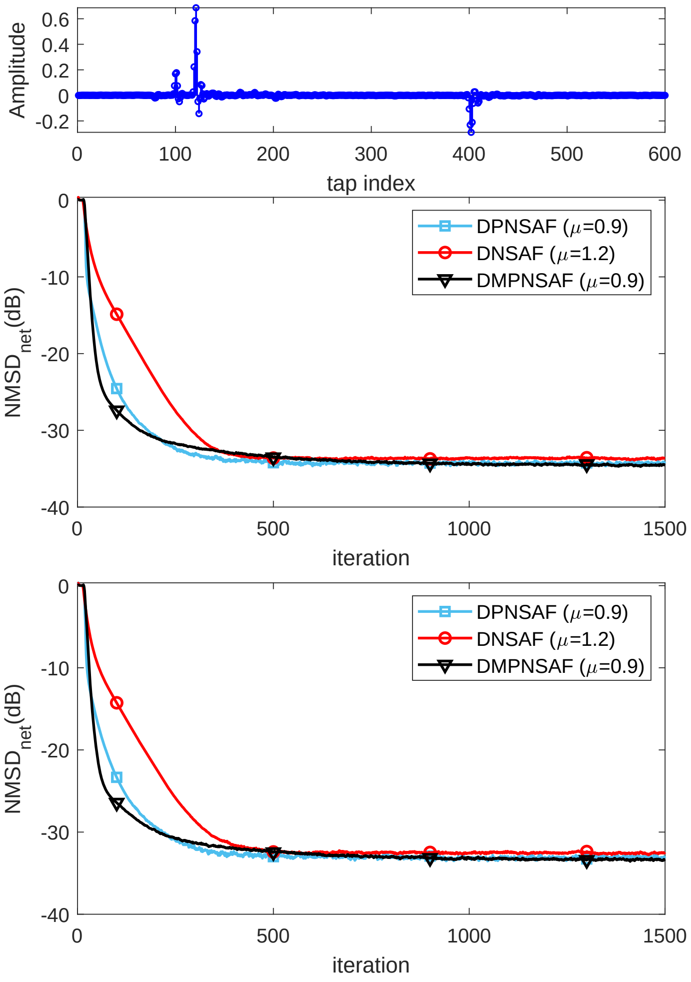

Fig. 4. The NMSD learning curves of DNSAF, DPNSAF, and DMPNSAF algorithms: (top) impulse response of the unknown system, (middle) when input regressors are white Gaussian, (bottom) when input regressors are Gaussian AR(1).

algorithms are shown in Fig. 5 for both input regression types (when input regressors are white Gaussian and when input regressors are Gaussian AR(1)). Here we set, $\delta = 0.01$, $\rho = 0.025$, $\gamma = -0.5$, $\pi_1 = -0.8$, $\pi_2 = 0.8$, $g_0 = 0.1$, and $\zeta = 0.001$ are set. As shown in the figure, the DIPNSAF algorithm concludes faster convergence than the DPNSAF. In addition, the performance of DIIPNSAF algorithm is slightly better than that of the DIPNSAF.

Now, the theoretical values of examined algorithms are confirmed by conducting numerous simulations. In these experiments, the network consists of $K = 6$ nodes and the unknown system is sparse (see Fig. 6). The node profiles of p_k (correlation index), $\sigma^2_{u_k}$ (power of regressor data), and $\sigma^2_{\eta_k}$ (noise variance) are shown in Fig. 7. In Fig. 8 (top), the simulated NMSD learning curves of DPNSAF, DMPNSAF, DIPNSAF and DIIPNSAF and their theoretical steady-state values have been presented. The theoretical values are calculated from (65). It can be observed that there is a decent agreement between the values obtained by simulations and those obtained by theoretical expressions. This can be verified by Fig. 8 (bottom) which presents values of theoretical steady-state NMSD and experimental steady-state NMSD at each individual node.

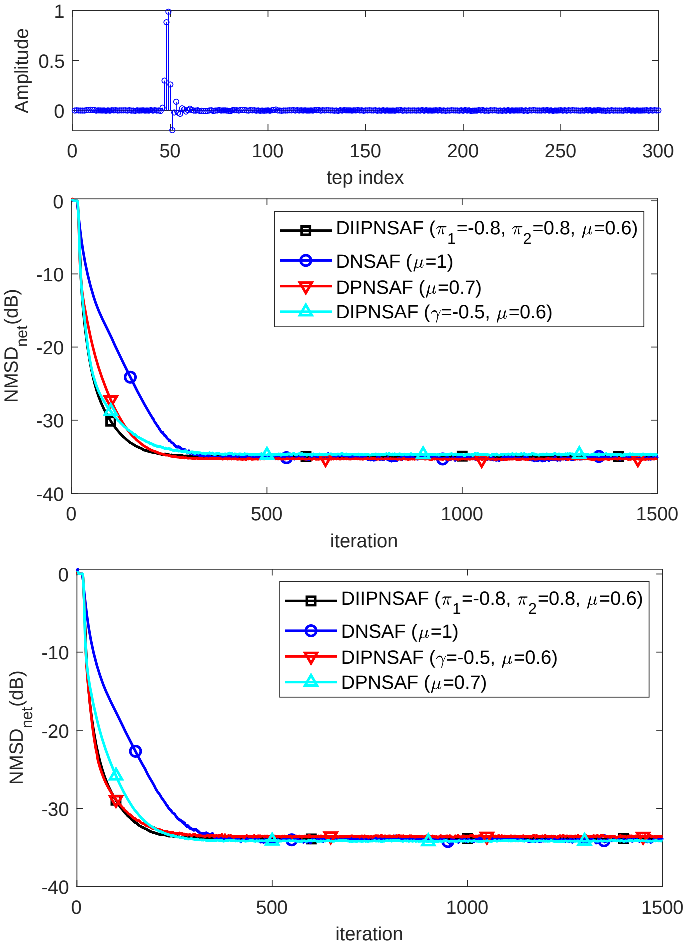

Fig. 5. The NMSD learning curves of DNSAF, DPNSAF, DIPNSAF, and DIIPNSAF algorithms: (top) impulse response of the unknown system, (middle) when input regressors are white Gaussian, (bottom) when input regressors are Gaussian AR(1).

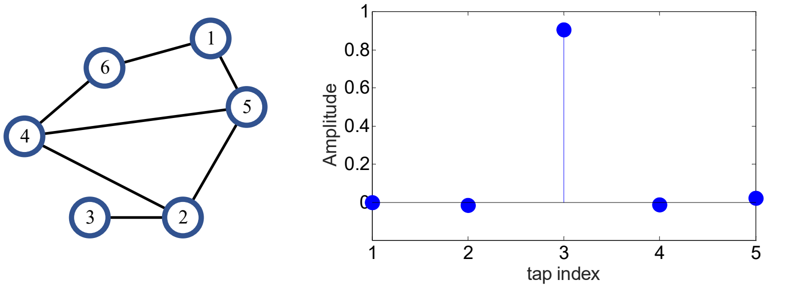

Fig. 6. The sparse impulse response (right) and network topology (left).

VI. CONCLUSIONS

Here, a group of DPSNF algorithms, suitable for sparse systems, based on the proportionate adaptation method has been introduced. The DPNSAF, DMPNSAF, DIPNSAF, and DIIPNSAF algorithms have been proposed for distributed processing over diffusion networks and it is shown that the proposed algorithms are faster than the

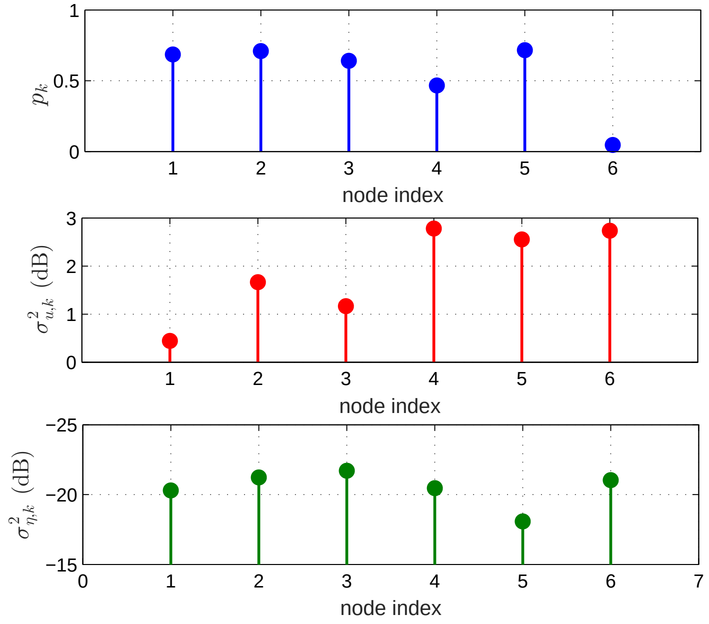

Fig. 7. Correlation index (top), input power (middle), and noise power at each node (bottom).

DNSAF algorithm when they are applied to a system with sparse impulse response. Regarding the weighted energy conservation, the behavior of presented algorithms has been discussed through analysis of the transient and the steady-state performances across the whole network. A decent functionality of the investigated algorithms has been demonstrated by various experiments.

Data Availability Statement

The datasets (radio channel coefficients) analysed during the current study are available in http://spib.linse.ufsc.br/microwave.html

References

[1] A. H. Sayed, “Adaptation, learning, and optimization over networks,” *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[2] X. Liu, X. Gao, and J. Han, “Distributed fault estimation for a class of nonlinear multiagent systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–9, 2018.

[3] Y. Xia, S. Su, H. Su, X. Zhang, W. Luo, and W. Yang, “Detection of data integrity attacks in distributed state estimation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–10, 2020.

[4] C. G. Lopes and A. H. Sayed, “Incremental adaptive strategies over distributed networks,” *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, Aug 2007.

[5] R. Kukde, M. S. Manikandan, and G. Panda, “Incremental learning based adaptive filter for nonlinear distributed active noise control system,” *IEEE Open Journal of Signal Processing*, vol. 1, pp. 1–13, 2020.

[6] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, April 2005, pp. 63–70.

[7] R. Olfati-Saber and J. S. Shamma, “Consensus filters for sensor networks and distributed sensor fusion,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec 2005, pp. 6698–6703.

[8] K. Xie, Q. Cai, Z. Zhang, and M. Fu, “Distributed algorithms for average consensus of input data with fast convergence,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2019.

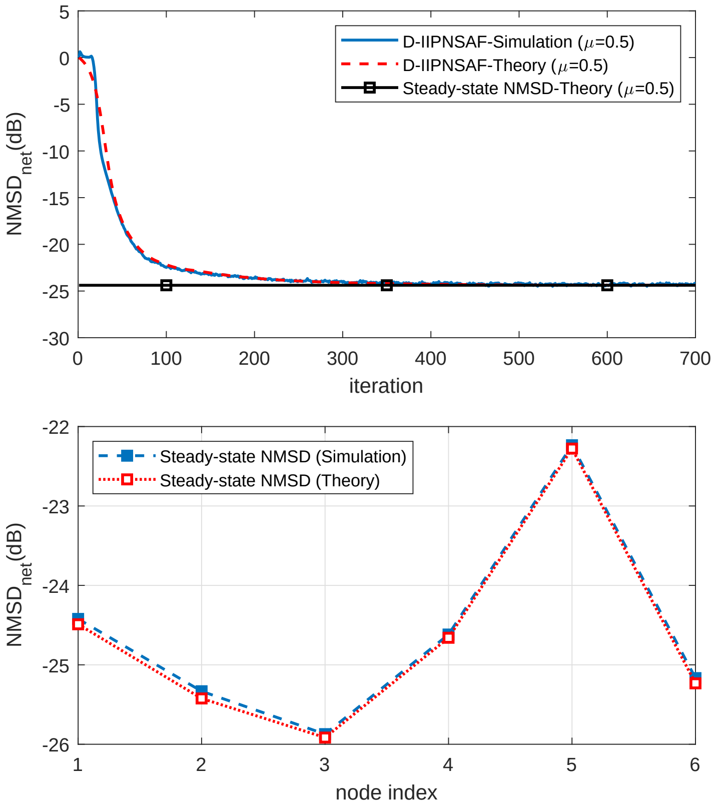

Fig. 8. NMSD curves of DPNSAF, DMPNSAF, DIPNSAF and DIIPNSAF algorithms and their theoretical steady-state values.

[9] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*, vol. 3, April 2007, pp. III–917–III–920.

[10] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, Aug 2012.

[11] X. Zhao and A. H. Sayed, “Performance limits for distributed estimation over LMS adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5107–5124, Oct 2012.

[12] S. Zhang, H. C. So, W. Mi, and H. Han, “A family of adaptive decorrelation NLMS algorithms and its diffusion version over adaptive networks,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 638–649, Feb 2018.

[13] S. Vlaski and A. H. Sayed, “Diffusion learning in non-convex environments,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5262–5266.

[14] V. C. Gogineni and M. Chakraborty, “Improving the performance of multitask diffusion apa via controlled inter-cluster cooperation,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 3, pp. 903–912, March 2020.

[15] L. Hu, F. Chen, S. Duan, L. Wang, and J. Wu, “An improved diffusion affine projection estimation algorithm for wireless sensor networks,” *Circuits, Systems, and Signal Processing*, Dec 2019.

[16] Z. Luo, H. Zhao, and X. Zeng, “A class of diffusion zero attracting stochastic gradient algorithms with exponentiated error cost functions,” *IEEE Access*, vol. 8, pp. 4885–4894, 2020.

[17] S. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, Dec 2012.

[18] W. Wang and H. Zhao, “Performance analysis of diffusion least mean fourth algorithm over network,” *Signal Processing*, vol. 141, pp. 32 – 47, 2017.

[19] W. Xia and Y. Wang, “A variable step-size diffusion LMS algorithm over networks with noisy links,” *Signal Process.*, vol. 148, no. C, p. 205213, Jul. 2018.

[20] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[21] Y. Chu and C. Mak, "A variable forgetting factor diffusion recursive least squares algorithm for distributed estimation," *Signal Processing*, vol. 140, pp. 219 – 225, 2017.

[22] M. De Courville and P. Duhamel, "Adaptive filtering in subbands using a weighted criterion," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2359–2371, Sep. 1998.

[23] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, Jan 1992.

[24] K. A. Lee and W. S. Gan, "Improving convergence of the NLMS algorithm using constrained subband updates," *IEEE Signal Processing Letters*, vol. 11, no. 9, pp. 736–739, Sep. 2004.

[25] J.-H. Seo, S. M. Jung, and P. Park, "A diffusion subband adaptive filtering algorithm for distributed estimation using variable step size and new combination method based on the MSD," *Digital Signal Processing*, vol. 48, pp. 361 – 369, 2016.

[26] J. Ni, "Diffusion sign subband adaptive filtering algorithm for distributed estimation," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2029–2033, Nov 2015.

[27] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1847–1862, 2010.

[28] W. U. Bajwa, J. Haupt, G. Raz, and R. Nowak, "Compressed channel sensing," pp. 5–10, 2008.

[29] C. Paleologu, J. Benesty, and S. Ciochina, *Sparse Adaptive Filters for Echo Cancellation*, 2011.

[30] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing (Signals and Communication Technology)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[31] S. Ozen, W. Hillery, M. Zoltowski, S. M. Nereyanuru, and M. Fimoff, "Structured channel estimation based decision feedback equalizers for sparse multipath channels with applications to digital tv receivers," in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, 2002.*, vol. 1, 2002, pp. 558–564 vol.1.

[32] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed channel sensing: A new approach to estimating sparse multipath channels," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.

[33] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1708–1721, 2010.

[34] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sep. 2000.

[35] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2002, pp. II–1881–II–1884.

ICASSP

[36] Hongyang Deng and M. Doroslovacki, "Proportionate adaptive algorithms for network echo cancellation," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1794–1803, May 2006.

[37] P. A. Naylor, J. Cui, and M. Brookes, "Adaptive algorithms for sparse echo cancellation," *Signal Processing*, vol. 86, no. 6, pp. 1182 – 1192, 2006, applied Speech and Audio Processing.

[35] and [45] are the same

[38] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, July 2008.

[39] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[40] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 2005, pp. 63–70.

[41] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 2996–3000.

[42] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

is this correct?

[43] M. Shams Esfand Abadi and M. S. Shafiee, "A family of diffusion normalized subband adaptive filter algorithms over distributed networks," *International Journal of Communication Systems*, vol. 30, no. 17, p. e3348, 2017.

[44] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, Sep. 2000.

[45] J. Benesty and S. L. Gay, "An improved pnlms algorithm," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2002, pp. II–1881–II–1884.

ICASSP

[46] H. Malvar, "Extended lapped transforms: Properties, applications, and fast algorithms," *IEEE Transactions on Signal Processing*, vol. 40, no. 11, p. 27032714, Nov. 1992.