Diffusion Partitioned-Block Frequency-Domain Adaptive Filter Over Multitask Networks

Hongyu Han, Yishu Peng, Fenglian Zhang, Dinh Cong Le, and Sheng Zhang, *Senior Member, IEEE*

***Abstract*—As a generalization of the frequency-domain adaptive filter (FDAF) algorithm, partitioned-block frequency-domain adaptive filter (PBFDAF) results in minimal signal path delay. In this paper, we propose the diffusion normalized PBFDAF algorithm based on an unsupervised clustering strategy to address the low complexity implementation issues in multitask networks. Each node adaptively adjusts the combination weight coefficients by minimizing the instantaneous mean square deviation (MSD) in frequency-domain. The simulation results demonstrate that the proposed algorithm achieves superior performance.**

***Index Terms*—Adaptive networks, frequency-domain, diffusion strategy, adaptive combination weight.**

I. Introduction

DISTRIBUTED adaptive estimation problem has been widely studied during the past decades, and applied to many fields, such as target tracking and localization, environmental monitorings, and active noise control [1]–[3]. In the adaptive single-task networks, all agents estimate a single parameter vector from streaming data by various collaborative manners, such as consensus strategies, incremental strategies, and diffusion strategies. Due to the superior robustness and adaptation ability of diffusion strategy, many algorithms based that have been proposed, including diffusion least-mean square (D-LMS), diffusion decorrelation normalized LMS, and their improved variants [4]–[6].

Different from the single-task scenarios, multitask networks need to infer multiple parameter vectors cooperatively. Each cluster has different unknown parameter vector but similarities exist in connected clusters. In general, the multitask networks algorithms are divided into two types. First, multitask networks have been clustered and all extra information is assumed to be obtained. In this scenario, the multitask D-LMS algorithms developed in [7], [8] promote inter-cluster cooperation by incorporating regularizer terms in cost functions. Based on this, many improved versions have been presented, for example, multitask affine projection [9], multitask recursive least squares [10], robust multitask adaptive filtering algorithms [11], and so on. In the case where there is no knowledge of the cluster structure, the D-LMS with adaptive clustering minimizes the instantaneous mean square deviation (MSD) to dynamically adjust the combination weight matrix [12], [13]. Later, the new diffusion clustering schemes [14] were proposed, which compare the weight estimates of neighboring nodes to determine whether they belong to the same cluster. In addition, the federated multitask learning and graph federated multitask learning have been studied recently in [15], [16].

In practical applications with multi-tap filters, such as acoustic echo cancellation and channel equalization, the full-band filters will cause a very large amount of computation burden in the time-domain. With the resort to fast fourier transform (FFT), the frequency-domain implementations based on overlapped-block convolution are efficient methods to solve the problem [17]–[19]. There are mainly two kinds of frequency-domain algorithms, namely, unconstrained and constrained versions. The performance analysis for full modeling and under-modeling scenarios can be found in [20], [21]. Unfortunately, due to the collection of input and output block data, the traditional frequency-domain algorithms with large FFT points suffer from serious signal path delay. To overcome this drawback, the partitioned-block frequency-domain adaptive filter (PBFDAF) was designed [22]–[26].

In order to reduce the computational burden of each node in multitask networks, this paper designs the diffusion partitioned-block frequency-domain algorithms without prior information about clustering. First, we derive the diffusion normalized PBFDAF (D-NPBFDAF) algorithm and analyze its stability. Then, we propose the improved D-NPBFDAF algorithm with adaptive combination weight matrix, which is determined by minimizing instantaneous MSD in frequency-domain. Simulations are carried out to illustrate the superior performance of the proposed algorithms.

II. Multitask Problem and D-NPBFDAF

We consider a strongly-connected network with N agents, and each agent k has access to measurement data $d_k(n)$ and $L \times 1$ input regression vector $\mathbf{x}_k(n)$, related via a linear regression model of the form

$$d_k(n) = \mathbf{x}_k^{\mathsf{T}}(n)\mathbf{w}_k^o + v_k(n), k = 1, 2, \cdots, N \tag{1}$$

where n is the time index, $(\cdot)^{\mathsf{T}}$ represents the transpose operation, $\mathbf{w}_k^o$ is the $L \times 1$ unknown parameter vector to be estimated at node k, and $v_k(n)$ is additive zero-mean white noise with variance $\sigma_{v,k}^2$. In the D-NPBFDAF algorithm, the filter coefficients $\mathbf{w}_k(n)$ are partitioned to P subfilters of length $M = L/P$, namely, $\mathbf{w}_{k,p}(n) = [w_{k,pM}(n), w_{k,pM+1}(n), \cdots, w_{k,(p+1)M-1}(n)]^{\mathsf{T}}$,

Manuscript received **, 2022. This work was supported in part by **

Hongyu Han is with the School of Computer Science, Sichuan Normal University, Chengdu 610068, China (hyhan@my.swjtu.edu.cn).

Yishu Peng, Fenglian Zhang, Sheng Zhang and are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan 611756, China (e-mail: yishupeng.22@gmail.com; 1037028412@qq.com; dr.s.zhang@ieee.org).

Dinh Cong Le is with the School of Engineering and Technology, Vinh University, 182 Le Duan str., Vinh City, Nghean province, Vietnam (congdhv@gmail.com).

where p denotes the p-th subfilter. The output of the filter is given by

$$\hat{d}_k(n) = \sum_{p=0}^{P-1} \overline{\mathbf{x}}_{k,p}^{\mathsf{T}}(n)\mathbf{w}_{k,p}(n), \tag{2}$$

where $\overline{\mathbf{x}}_{k,p}(n) = [x_k(n-pM), x_k(n-pM-1), \cdots, x_k(n-(p+1)M+1)]^{\mathsf{T}}$. The equation (2) can be implemented by using FFT for each sub-block. The frequency-domain matrix $\mathbf{X}_{k,p}(j)$ of the p-th partition at node k is $\mathsf{diag}(\mathbf{x}_{k,p}^F(j))$, where $\mathbf{x}_{k,p}^F(j) = \boldsymbol{F}[x_k((j-p)M), x_k((j-p)M-1), \cdots, x_k((j-p-2)M+1)]^{\mathsf{T}}$, j is the block index, and $\mathsf{diag}(\cdot)$ denotes the diagonal matrix consisting of diagonal entries $(\cdot)$. $\boldsymbol{F}$ is the fourier transform matrix with the (ℓ, n) entry $\frac{1}{\sqrt{2M}}e^{\frac{-i2\pi\ell n}{2M}}, \ell, n = 0, 1, \cdots, 2M-1$, and $(\cdot)^F$ denotes the vector in frequency-domain. The later delay block input vectors are obtained via block index shifting without involving any computation as

$$\mathbf{X}_{k,p+1}(j) = \mathbf{X}_{k,p}(j-1), p = 0, \cdots, P-2 \tag{3}$$

With the fast implementation of linear convolution, the block vector $\mathbf{d}_k(j) = [d_k(jM), d_k(jM-1), \cdots, d_k(jM-M+1)]^{\mathsf{T}}$ can be computed in frequency-domain as

$$\begin{aligned}\mathbf{d}_k(j) &= \begin{bmatrix}\mathbf{I}_M & \mathbf{0}_M\end{bmatrix} \boldsymbol{F}^* \sum_{p=0}^{P-1} \mathbf{X}_{k,p}(j)\mathbf{w}_{k,p}^{o,F} + \mathbf{v}_k(j) \\ &= \begin{bmatrix}\mathbf{I}_M & \mathbf{0}_M\end{bmatrix} \boldsymbol{F}^* \mathbf{X}_k(j)\mathbf{w}_k^{o,F} + \mathbf{v}_k(j),\end{aligned} \tag{4}$$

where $(\cdot)^*$ denotes the complex conjugate transpose operator, $\mathbf{X}_k(j) = \begin{bmatrix}\mathbf{X}_{k,0}(j) & \mathbf{X}_{k,1}(j) & \cdots & \mathbf{X}_{k,P-1}(j)\end{bmatrix}$ is a $2M \times 2L$ input matrix, $\mathbf{w}_k^{o,F} = \begin{bmatrix}(\mathbf{w}_{k,0}^{o,F})^{\mathsf{T}} & (\mathbf{w}_{k,1}^{o,F})^{\mathsf{T}} & \cdots & (\mathbf{w}_{k,P-1}^{o,F})^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ is a $2L \times 1$ vector with $\mathbf{w}_{k,p}^{o,F} = \sqrt{2M}\boldsymbol{F}^* \begin{bmatrix}(\mathbf{w}_{k,p}^o)^{\mathsf{T}} & \mathbf{0}_{1\times M}\end{bmatrix}^{\mathsf{T}}$, and $\mathbf{v}_k(j) = [v_k(jM), v_k(jM-1), \cdots, v_k(jM-M+1)]^{\mathsf{T}}$ is the noise vector.

To obtain robust estimation for $\mathbf{w}_k^{o,F}$, the following global cost function is built

$$\begin{aligned}&\min_{\mathbf{w}_k^F} \mathbb{E}\{\|\mathbf{d}_k(j) - \bar{\mathbf{X}}_k(j)\mathbf{w}_k^F\|_{c_k}^2\} + \sum_{\ell\in\mathcal{N}_k\backslash k} b_{k,\ell}\|\mathbf{w}_k^F - \boldsymbol{\varphi}_l^F(j)\|^2, \\ &\text{s.t.}\, \boldsymbol{F}\mathbf{w}_{k,p}^F = \begin{bmatrix}\times \\ \mathbf{0}_{M\times 1}\end{bmatrix}, \quad p = 0, \cdots, P-1\end{aligned} \tag{5}$$

where $\overline{\mathbf{X}}_k(j) = \begin{bmatrix}\mathbf{I}_M & \mathbf{0}_M\end{bmatrix} \boldsymbol{F}^*\mathbf{X}_k(j)$, $\mathbf{w}_k^F = \begin{bmatrix}(\mathbf{w}_{k,0}^F)^{\mathsf{T}} & (\mathbf{w}_{k,1}^F)^{\mathsf{T}} & \cdots & (\mathbf{w}_{k,P-1}^F)^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$ with $\mathbf{w}_{k,p}^F = \sqrt{2M}\boldsymbol{F}^* \begin{bmatrix}(\mathbf{w}_{k,p})^{\mathsf{T}} & \mathbf{0}_{1\times M}\end{bmatrix}^{\mathsf{T}}$, $E\{\cdot\}$ and $\|\cdot\|$ denote mathematical expectation and Euclidean norm, respectively. $\boldsymbol{\varphi}_l^F(j)$, $\ell \in \mathcal{N}_k\backslash k$ is the intermediate estimate available from a neighbor. $b_{k,\ell}$ and c_k represent nonnegative combination coefficients and positive constant, respectively.

Following the analogous procedure used in [18], the update of the D-NPBFDAF is

$$\begin{aligned}\boldsymbol{\varphi}_k^F(j) =& \mathbf{Q}[\mathbf{w}_k^F(j-1) + \mu_k\overline{\boldsymbol{\Lambda}}_k^{-1}(j)\overline{\mathbf{X}}_k^*(j)(\mathbf{d}_k(j) \\ &- \overline{\mathbf{X}}_k(j)\mathbf{w}_k^F(j-1))],\end{aligned} \tag{6}$$

$$\mathbf{w}_k^F(j) = \sum_{\ell\in\mathcal{N}_k} c_{k,\ell}\boldsymbol{\varphi}_l^F(j), \tag{7}$$

where the constrained matrix $\mathbf{Q} = \mathbf{I}_P \otimes \overline{\mathbf{Q}}$ with $\overline{\mathbf{Q}} \triangleq \boldsymbol{F}^*\boldsymbol{D}\boldsymbol{F}$ and $\boldsymbol{D} = \boldsymbol{I}_M \oplus \mathbf{0}_M$, $\otimes$ is the Kronecker product, and $\oplus$ constructs a block diagonal matrix by two matrices. μ_k is the step-size, $c_{k,\ell} = \mu_k b_{k,\ell}$ if $\ell \in \mathcal{N}_k\backslash k$, $c_{k,\ell} = 0$ if $\ell \notin \mathcal{N}_k$, $c_{k,k} = c_k$ and $c_{k,k} = 1 - \mu_k \sum_{\ell\in\mathcal{N}_k\backslash k} b_{k,\ell}$. For sufficiently large M, the normalized matrix $\overline{\boldsymbol{\Lambda}}_k(j) = \mathbb{E}\{\overline{\mathbf{X}}_k^*(j)\overline{\mathbf{X}}_k(j)\}$ can be approximated by a diagonal matrix and calculated as

$$\overline{\boldsymbol{\Lambda}}_k(j) = \mathbf{I}_P \otimes \boldsymbol{\Lambda}_k(j), \tag{8}$$

$$\boldsymbol{\Lambda}_k(j) = \beta\boldsymbol{\Lambda}_k(j-1) + \frac{(1-\beta)}{2}\mathbf{X}_{k,0}^*(j)\mathbf{X}_{k,0}(j), \tag{9}$$

where $0 \ll \beta < 1$ is the forgetting factor. Note that the unconstrained version sets $\mathbf{Q} = \mathbf{I}_{2L}$, while the unnormalized version sets $\overline{\boldsymbol{\Lambda}}_k(j) = \mathbf{I}_{2L}$. The constrained version has a smaller steady-state error than the unconstrained version [20], [22]. In the case of colored inputs, the normalized version can accelerate convergence.

III. Stability Analysis

We start with a global weight-error relation for the D-NPBFDAF algorithm. The frequency-domain weight error vectors $\tilde{\boldsymbol{\varphi}}_k^F(j)$ and $\tilde{\mathbf{w}}_k^F(j)$ at each node k are introduced:

$$\tilde{\boldsymbol{\varphi}}_k^F(j) = \mathbf{w}_k^{o,F} - \boldsymbol{\varphi}_k^F(j), \tilde{\mathbf{w}}_k^F(j) = \mathbf{w}_k^{o,F} - \mathbf{w}_k^F(j) \tag{10}$$

The equation (6) can be written as

$$\boldsymbol{\varphi}_k^F(j) = \mathbf{Q}[\mathbf{w}_k^F(j-1) + \mu_k\overline{\boldsymbol{\Lambda}}_k^{-1}(j)\mathbf{X}_k^*(j)\mathbf{e}_k^F(j)], \tag{11}$$

where $\mathbf{e}_k^F(j)$ is the frequency-domain error vector defined by

$$\mathbf{e}_k^F(j) = \mathbf{H}\mathbf{X}_k(j)\tilde{\mathbf{w}}_k^F(j-1) + \mathbf{v}_k^F(j), \tag{12}$$

with $\mathbf{H} = \boldsymbol{F}\boldsymbol{D}\boldsymbol{F}^*$ and the frequency-domain noise vector $\mathbf{v}_k^F(j) = \boldsymbol{F}[v_k(jM), \cdots, v_k(jM-M+1), \quad \mathbf{0}_{1\times M}]^{\mathsf{T}}$.

From (7) and (11), $\tilde{\boldsymbol{\varphi}}_k^F(j)$ and $\tilde{\mathbf{w}}_k^F(j)$ are recursively expressed as

$$\tilde{\boldsymbol{\varphi}}_k^F(j) = \mathbf{Q}[\tilde{\mathbf{w}}_k^F(j-1) - \mu_k\overline{\boldsymbol{\Lambda}}_k^{-1}(j)\mathbf{X}_k^*(j)\mathbf{e}_k^F(j)], \tag{13}$$

$$\tilde{\mathbf{w}}_k^F(j) = \sum_{\ell\in\mathcal{N}_k} c_{k,\ell}\tilde{\boldsymbol{\varphi}}_l^F(j) + \mathbf{w}_k^{o,F} - \sum_{\ell\in\mathcal{N}_k} c_{k,\ell}\mathbf{w}_l^{o,F}, \tag{14}$$

where $\mathbf{w}_k^{o,F} = \mathbf{Q}\mathbf{w}_k^{o,F}$.

Replacing $\mathbf{e}_k^F(j)$ in (13) by (12), we get

$$\begin{aligned}\tilde{\boldsymbol{\varphi}}_k^F(j) =& \mathbf{Q}[\tilde{\mathbf{w}}_k^F(j-1) - \mu_k\overline{\boldsymbol{\Lambda}}_k^{-1}(j)\mathbf{X}_k^*(j)\mathbf{H}\mathbf{X}_k(j)\tilde{\mathbf{w}}_k^F(j-1)] \\ &- \mu_k\mathbf{Q}\overline{\boldsymbol{\Lambda}}_k^{-1}(j)\mathbf{X}_k^*(j)\mathbf{v}_k^F(j).\end{aligned} \tag{15}$$

Define the global weight-error vectors $\tilde{\mathbf{w}}^F(j)$, $\tilde{\boldsymbol{\varphi}}^F(j)$, the optimal weight vector $\mathbf{w}^{o,F}$ and $\mathbf{g}^F(j)$:

$$\tilde{\mathbf{w}}^F(j) = \begin{bmatrix}\tilde{\mathbf{w}}_1^F(j) \\ \tilde{\mathbf{w}}_2^F(j) \\ \vdots \\ \tilde{\mathbf{w}}_N^F(j)\end{bmatrix}, \tilde{\boldsymbol{\varphi}}^F(j) = \begin{bmatrix}\tilde{\boldsymbol{\varphi}}_1^F(j) \\ \tilde{\boldsymbol{\varphi}}_2^F(j) \\ \vdots \\ \tilde{\boldsymbol{\varphi}}_N^F(j)\end{bmatrix}, \mathbf{w}^{o,F} = \begin{bmatrix}\mathbf{w}_1^{o,F} \\ \mathbf{w}_2^{o,F} \\ \vdots \\ \mathbf{w}_N^{o,F}\end{bmatrix}, \tag{16}$$

$$\mathbf{g}^F(j) = \mathrm{col}\{\overline{\boldsymbol{\Lambda}}_1^{-1}(j)\mathbf{X}_1^*(j)\mathbf{v}_1^F(j), \cdots, \overline{\boldsymbol{\Lambda}}_N^{-1}(j)\mathbf{X}_N^*(j)\mathbf{v}_N^F(j)\}, \tag{17}$$

and the matrices:

$$\mathcal{Q} = \mathbf{I}_N \otimes \mathbf{Q}, \mathcal{C} = \mathbf{C} \otimes \mathbf{I}_{2L}, \tag{18}$$

$$\mathcal{M} = \mathsf{diag}\{\mu_1 \mathbf{I}_{2L}, \cdots, \mu_N \mathbf{I}_{2L}\}, \tag{19}$$

$$\mathbf{X}(j) = \mathsf{diag}\{\overline{\mathbf{\Lambda}}_1^{-1}(j)\mathbf{X}_1^*(j)\mathbf{H}\mathbf{X}_1(j), \cdots, \overline{\mathbf{\Lambda}}_N^{-1}(j)\mathbf{X}_N^*(j)\mathbf{H}\mathbf{X}_N(j)\}, \tag{20}$$

where $\mathrm{col}\{\cdot\}$ constructs a column vector of its vector arguments, $c_{k,\ell}$ is the (k,ℓ) entry of the matrix $\mathbf{C}$, and $\mathbf{C}\mathbf{1}_{N\times 1} = \mathbf{1}_{N\times 1}$.

Then, we have

$$\tilde{\boldsymbol{\varphi}}^F(j) = \mathcal{Q}\tilde{\mathbf{w}}^F(j-1) - \mathcal{Q}\mathcal{M}\mathbf{X}(j)\tilde{\mathbf{w}}^F(j-1) - \mathcal{Q}\mathcal{M}\mathbf{g}^F(j), \tag{21}$$

$$\tilde{\mathbf{w}}^F(j) = \mathcal{C}\tilde{\boldsymbol{\varphi}}^F(j) + (\mathbf{I}_{2NL} - \mathcal{C})\mathbf{w}^{o,F}. \tag{22}$$

Based on (21) and (22), we get

$$\begin{aligned}\tilde{\mathbf{w}}^F(j) =& \mathcal{C}\mathcal{Q}(\mathbf{I}_{2NL} - \mathcal{M}\mathbf{X}(j))\tilde{\mathbf{w}}^F(j-1) - \mathcal{C}\mathcal{Q}\mathcal{M}\mathbf{g}^F(j) \\ &+ (\mathbf{I}_{2NL} - \mathcal{C})\mathbf{w}^{o,F}.\end{aligned} \tag{23}$$

To facilitate the convergence analysis, we introduce the following statistical assumptions, which are widely adopted in the literature [20]–[23].

Assumption 1: The frequency-domain input signals $\mathbf{x}_{k,p}^F(j)$ for $p = 0, 1, \cdots, P-1$ are spatially and temporally independent zero-mean stationary random sequences for $k = 1, \cdots, N$.

Assumption 2: The frequency-domain noise signal vectors $\mathbf{v}_{k,j}^F$ are spatially and temporally independent zero-mean stationary random variables for $k = 1, \cdots, N$ and are independent of $\mathbf{x}_{k,p}^F(j)$ for $p = 0, 1, \cdots, P-1$.

Taking expectation of both sides of (23) with Assumptions 1 and 2, we get the mean weight error recursion as

$$\begin{aligned}\mathbb{E}\{\tilde{\mathbf{w}}^F(j)\} =& \mathcal{C}\mathcal{Q}(\mathbf{I}_{2NL} - \mathcal{M}\mathbb{E}\{\mathbf{X}(j)\})\mathbb{E}\{\tilde{\mathbf{w}}^F(j-1)\} \\ &+ (\mathbf{I}_{2NL} - \mathcal{C})\mathbf{w}^{o,F}.\end{aligned} \tag{24}$$

Theorem 1: (Mean stability) The D-NPBFDAF algorithm is mean stable if the step-size meets the condition:

$$\rho(\mathcal{Q}(\mathbf{I}_{2NL} - \mathcal{M}\mathbb{E}\{\mathbf{X}(j)\})) < 1. \tag{25}$$

Taking the limits of both sides of (24), we attain the steady-state $\mathbb{E}\{\tilde{\mathbf{w}}^F(\infty)\}$ as

$$\begin{aligned}&\mathbb{E}\{\tilde{\mathbf{w}}^F(\infty)\} \\ &= (\mathbf{I}_{2NL} - \mathcal{C}\mathcal{Q}(\mathbf{I}_{2NL} - \mathcal{M}\mathbb{E}\{\mathbf{X}(j)\}))^{-1}(\mathbf{I}_{2NL} - \mathcal{C})\mathbf{w}^{o,F}.\end{aligned} \tag{26}$$

Obviously, $\mathbf{w}_k^F(\infty)$ will converge to a biased solution, which depends on the combination coefficients $c_{k,\ell}$. Thus, selecting appropriate combination coefficients can reduce the steady-state bias and improve the convergence performance.

IV. Adaptive Combanition Weight Matrix

In this section, we extend the adaptive clustering strategy [13] to frequency-domain to further improve the steady-state performance of the D-NPBFDAF algorithm.

Algorithm 1: ACD-NPBFDAF

Initialization: $\mathbf{w}_k^F(0) = \mathbf{0}_{2L\times 1}$, μ_k, $\mathbf{Q} = \mathbf{I}_P \otimes \overline{\mathbf{Q}}$, $\overline{\mathbf{Q}} \triangleq \boldsymbol{F}^*\boldsymbol{D}\boldsymbol{F}$
$\boldsymbol{D} = \boldsymbol{I}_M \oplus \mathbf{0}_M$

For $j = 1, 2, 3 \cdots$

At node k, $k = 1, 2, \cdots, N$

$\mathbf{x}_{k,0}(j) = [x_k(jM), x_k(jM-1), \cdots, x_k(jM-2M+1)]^\mathsf{T}$

$\mathbf{d}_k(j) = [d_k(jM), d_k(jM-1), \cdots, d_k(jM-M+1)]^\mathsf{T}$

Signal FFT transform step:

$\mathbf{x}_{k,0}^F(j) = \boldsymbol{F}\mathbf{x}_{k,0}(j)$

$\mathbf{x}_{k,p+1}^F(j) = \mathbf{x}_{k,p}^F(j-1), p = 0, 1, \cdots, P-2$

$\mathbf{X}_{k,p}(j) = \mathsf{diag}(\mathbf{x}_{k,p}^F(j)), p = 0, 1, \cdots, P-1$

$\mathbf{X}_k(j) = \begin{bmatrix}\mathbf{X}_{k,0}(j) & \mathbf{X}_{k,1}(j) & \cdots & \mathbf{X}_{k,P-1}(j)\end{bmatrix}$

$\mathbf{y}_k^F(j) = \mathbf{X}_k(j)\mathbf{w}_k^F(j-1)$

$\mathbf{y}_k(j) = \begin{bmatrix}\mathbf{I}_M & \mathbf{0}_M\end{bmatrix}\boldsymbol{F}^*\mathbf{y}_k^F(j)$

$\mathbf{e}_k(j) = \mathbf{d}_k(j) - \mathbf{y}_k(j)$

$\mathbf{e}_k^F(j) = \boldsymbol{F}\begin{bmatrix}\mathbf{I}_M \\ \mathbf{0}_M\end{bmatrix}\mathbf{e}_k(j)$

Normalized matrix update step:

Update of $\overline{\mathbf{\Lambda}}_k(j)$ using (8) and (9)

Weight update step:

$\boldsymbol{\varphi}_k^F(j) = \mathbf{Q}[\mathbf{w}_k^F(j-1) + \mu_k\overline{\mathbf{\Lambda}}_k^{-1}(j)\mathbf{X}_k^*(j)\mathbf{e}_k^F(j)]$

Calculating the combination coefficients step:

$\mathbf{z}_k^F(j) = \mathbf{X}_k(j)\boldsymbol{\varphi}_k^F(j)$

$\mathbf{z}_k(j) = \begin{bmatrix}\mathbf{I}_M & \mathbf{0}_M\end{bmatrix}\boldsymbol{F}^*\mathbf{z}_k^F(j)$

$\mathbf{r}_k(j) = \mathbf{d}_k(j) - \mathbf{z}_k(j)$

$\mathbf{r}_k^F(j) = \boldsymbol{F}\begin{bmatrix}\mathbf{I}_M \\ \mathbf{0}_M\end{bmatrix}\mathbf{r}_k(j)$

$\mathbf{g}_k(j) = \overline{\mathbf{\Lambda}}_k^{-1}(j)\mathbf{X}_k^*(j)\mathbf{r}_k^F(j)$

$\overline{\mathbf{g}}_k(j) = \mathbf{g}_k(j)/(\|\mathbf{g}_k(j)\| + \varepsilon)$

$c_{k,\ell}(j) = \frac{\|\boldsymbol{\varphi}_k^F(j) + \mu_k\overline{\mathbf{g}}_k(j) - \boldsymbol{\varphi}_\ell^F(j)\|^{-2}}{\sum_{n\in\mathcal{N}_k}\|\boldsymbol{\varphi}_k^F(j) + \mu_k\overline{\mathbf{g}}_k(j) - \boldsymbol{\varphi}_n^F(j)\|^{-2}}$

Combination step:

Update of $\mathbf{w}_k^F(j)$ using (7)

End

The adaptive combination weight matrix $\mathbf{C}$ is obtained by minimizing the instantaneous MSD at each node k as

$$\begin{aligned}&\min_{\mathbf{c}_k} \mathbb{E}\{\|\hat{\mathbf{w}}_k^{o,F} - \sum_{\ell\in\mathcal{N}_k} c_{k,\ell}\boldsymbol{\varphi}_\ell^F(j)\|^2\} \\ &\text{subject to} \quad \mathbf{c}_k^\mathsf{T}\mathbf{1}_{N\times 1} = 1, \quad c_{k,\ell} \geqslant 0 \\ &\qquad\qquad\qquad c_{k,\ell} = 0, \qquad \text{if } \ell \notin \mathcal{N}_k\end{aligned} \tag{27}$$

where $\mathbf{c}_k = [c_{k,1}, c_{k,2}, \cdots, c_{k,N}]^\mathsf{T}$, and $\hat{\mathbf{w}}_k^{o,F}$ is a estimator for $\mathbf{w}_k^{o,F}$.

The expression (27) can be written as

$$\begin{aligned}&\mathbb{E}\{\|\hat{\mathbf{w}}_k^{o,F} - \sum_{\ell\in\mathcal{N}_k} c_{k,\ell}\boldsymbol{\varphi}_\ell^F(j)\|^2\} \\ &= \sum_{\ell\in\mathcal{N}_k}\sum_{m\in\mathcal{N}_k} c_{k,\ell}c_{k,m}\mathbb{E}\{(\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_\ell^F(j))^*(\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_m^F(j))\}.\end{aligned} \tag{28}$$

Let $\mathbf{\Phi}_k(j)$ be a $N \times N$ matrix at each node k, and the (ℓ, m)th entry is formed by

$$[\mathbf{\Phi}_k(j)]_{\ell,m} = \begin{cases}\mathbb{E}\{(\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_\ell^F(j))^*(\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_m^F(j))\}, & \ell, m \in \mathcal{N}_k \\ 0, & \text{otherwise}\end{cases} \tag{29}$$

where $\mathbf{\Phi}_k(j)$ is a positive semi-definite matrix. Then, the cost function (27) can be written as

$$\begin{aligned}&\min_{\mathbf{c}_k} \ \mathbf{c}_k^\mathsf{T}\mathbf{\Phi}_k(j)\mathbf{c}_k \\ &\text{subject to} \quad \mathbf{c}_k^\mathsf{T}\mathbf{1}_{N\times 1} = 1, \quad c_{k,\ell} \geqslant 0 \\ &\qquad\qquad\qquad c_{k,\ell} = 0, \qquad \text{if } \ell \notin \mathcal{N}_k.\end{aligned} \tag{30}$$

We replace $\mathbf{\Phi}_k(j)$ by its instantaneous value and diagonal matrix in order to make the problem more tractable. The problem can be formulated as

$$\begin{aligned} \min_{\mathbf{c}_k} & \sum_{\ell=1}^{N} c_{k,\ell}^2 \|\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_\ell^F(j)\|^2 \\ \text{subject to} \quad & \mathbf{c}_k^\mathsf{T} \mathbf{1}_{N\times 1} = 1, \quad c_{k,\ell} \geqslant 0 \\ & c_{k,\ell} = 0, \qquad \text{if} \ \ \ell \notin \mathcal{N}_k. \end{aligned} \tag{31}$$

The solution is given by

$$c_{k,\ell}(j) = \frac{\|\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_\ell^F(j)\|^{-2}}{\sum\limits_{m\in\mathcal{N}_k} \|\hat{\mathbf{w}}_k^{o,F} - \boldsymbol{\varphi}_m^F(j)\|^{-2}}, \text{for } \ell \in \mathcal{N}_k. \tag{32}$$

The solution (32) has an intuitive interpretation for the obtained combination parameter $c_{k,\ell}(j)$, which is inversely proportional to the square of the Euclidean distance between the weight estimator of node k and $\boldsymbol{\varphi}_\ell^F(j)$.

The expression (32) requires the knowledge of the weight vector $\hat{\mathbf{w}}_k^{o,F}$, which is generally not available beforehand. Hence, it is necessary to consider the estimator of $\mathbf{w}_k^{o,F}$ in expression (32). Using the similar scheme in [13], we obtain $\hat{\mathbf{w}}_k^{o,F}$ as

$$\hat{\mathbf{w}}_k^{o,F}(j) = \boldsymbol{\varphi}_k^F(j) - \mu_k \nabla J_k(\mathbf{w}) \mid_{\mathbf{w}=\boldsymbol{\varphi}_k^F(j)}, \tag{33}$$

where $J_k(\mathbf{w}) = E\{\|\mathbf{d}_k(j) - \overline{\mathbf{X}}_k(j)\mathbf{w}\|^2\}$.

Using the instantaneous value $\mathbf{g}_k(j) \triangleq \overline{\mathbf{X}}_k^*(j)(\mathbf{d}_k(j) - \overline{\mathbf{X}}_k(j)\boldsymbol{\varphi}_k^F(j))$ to approximate the true gradient $\nabla J_k(\mathbf{w})$, we attain the following estimator:

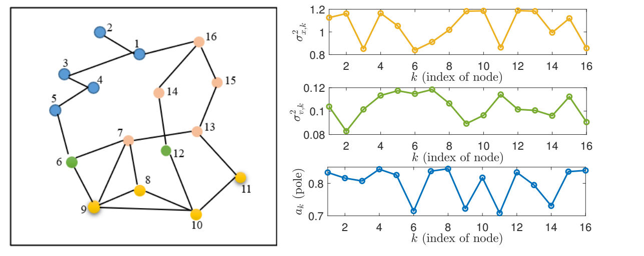
$$\hat{\mathbf{w}}_k^{o,F}(j) = \boldsymbol{\varphi}_k^F(j) + \mu_k \mathbf{g}_k(j). \tag{34}$$

Substituting this expression into (32) yields

$$c_{k,\ell}(j) = \frac{\|\boldsymbol{\varphi}_k^F(j) + \mu_k \mathbf{g}_k(j) - \boldsymbol{\varphi}_\ell^F(j)\|^{-2}}{\sum\limits_{m\in\mathcal{N}_k} \|\boldsymbol{\varphi}_k^F(j) + \mu_k \mathbf{g}_k(j) - \boldsymbol{\varphi}_m^F(j)\|^{-2}}, \quad \text{for } \ell \in \mathcal{N}_k. \tag{35}$$

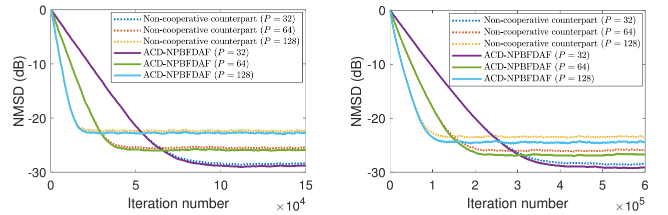
With similar analysis manipulations in [13], we can also provide an intuitive interpretation of (35), namely, the combination coefficients determined by (35) consider the similarity between intermediate and neighborhood estimates, and the slope of $J_k(\mathbf{w})$ to reduce MSD. The D-NPBFDAF with adaptive clustering (ACD-NPBFDAF) is summarized in Algorithm 1, where parameter ε in the normalized gradient $\overline{\mathbf{g}}_k(j) = \mathbf{g}_k(j)/(\|\mathbf{g}_k(j)\|^2 + \varepsilon)$ is a small positive number to avoid division by zero.

As we known, the D-LMS has a complexity of $O(MP)$ for per input sample while the ACD-NPBFDAF requires $O(log_2(2M))$ computation load [22], [24], [25]. Thus, for a large value of M, the computational efficiency of the proposed algorithm is obvious. In addition, the signal delay is reduced as compared to the original diffusion FDAF (D-FDAF) algorithm (i.e., $P = 1$) [18].



(a) (b)

Fig. 1. Node profile. (a) Network topology with $N = 16$; (b) Network signal, noise powers, and correlation index.



(a) (b)

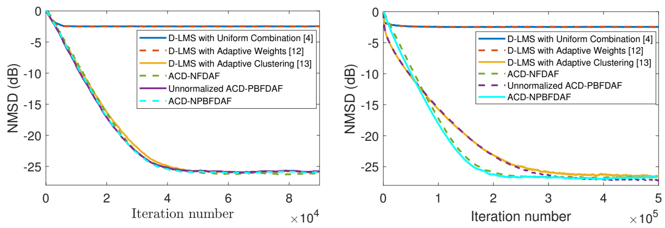
Fig. 2. Convergence performance of different P. (a) Using white input signals; (b) Using colored input signals.

V. Numerical Simulations

In this section, we use Monte-Carlo simulations to validate the effectiveness of the proposed algorithm. We consider a multitask network with 16 nodes. The network is divided into four clusters : $C_1 = \{1, 2, 3, 4, 5\}, C_2 = \{7, 13, 14, 15, 16\}, C_3 = \{8, 9, 10, 11\}, C_4 = \{6, 12\}$, where nodes in the same cluster estimate the common parameter vector without any prior information about clusters. The length of filter is $L = 512$. The coefficient vectors of the form $\mathbf{w}_{C_i}^o = \mathbf{w}_o + \delta\mathbf{w}_{C_i}$ are chosen as $\mathbf{w}_o = 0.5 + \mathbf{z}_k$, where $\mathbf{z}_k$ and $\mathbf{w}_{C_i}$ are zero-mean $L \times 1$ random vectors generated by Gaussian distribution with covariance matrices $0.04\boldsymbol{I}_L$ and $0.25\boldsymbol{I}_L$, respectively. δ is 1 in cluster C_1, and $-1, 2, -5$ for cluster C_2, C_3, C_4, separately. The settings of the network are showed in Fig. 1. The input regression vector $\mathbf{x}_k(n)$ and observation noise $v_k(n)$ are zero-mean $i.i.d$ Gaussian distributed with covariance matrix $\sigma_{x,k}^2\boldsymbol{I}_L$ and variance $\sigma_{v,k}^2$, respectively. All simulated curves are investigated with the network MSD (NMSD), defined as

$$\text{NMSD}(j) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\{\|\mathbf{w}_k^o - \mathbf{w}_k(j)\|^2\}. \tag{36}$$

In the first experiment, we study the effect of P on convergence rate in the case of white and colored input signals. The colored input signal is generated by $x_k(n) = a_k x_k(n-1) + \sqrt{1-a_k^2}s(n)$, where a_k denotes the correlation index at node k given in Fig. 1, $s(n)$ is a zero-mean white Gaussian signal. The algorithmic parameters in Fig. 2(a) are $\mu_k = 0.00085$ and $\epsilon = 0.01$. In Fig. 2(b), the step-size is selected as $\mu_k = 0.0002$. From Fig. 2, we observe that as the number of partitions increases, the ACD-NPBFDAF algorithm exhibits a faster convergence rate but larger steady-state deviation. It is because the block size M is the update period of the algorithm in essence. A large parameter P results in a small update period and provides fast convergence rate,



(a) (b)

Fig. 3. Convergence performance of different diffusion algorithms. (a) Using white input signals; (b) Using colored input signals.

and vice versa. Meanwhile, the proposed ACD-NPBFDAF outperforms the non-cooperative counterpart for both white and colored input signals.

In the second experiment, the proposed ACD-NPBFDAF algorithm is compared with the D-LMS with uniform combination policy [4], D-LMS with adaptive weights [12], D-LMS with adaptive clustering [13], normalized D-FDAF with adaptive clustering (i.e., ACD-NPBFDAF with $P = 1$), and unnormalized ACD-PBFDAF. In Fig. 3(a), the parameters of the ACD-NPBFDAF are $\mu_k = 0.00085$ and $P = 64$. To achieve the same steady-state error, other algorithmic parameters are $\mu_k = 0.0017$ and $P = 64$ for the unnormalized ACD-PBFDAF, $\mu_k = 0.032$ for the normalized D-FDAF with adaptive clustering (ACD-NFDAF), $\mu_k = 0.0003$ for the D-LMS with uniform combination policy, $\mu_k = 0.0004$ and $\nu_k = 0.1$ for the D-LMS with adaptive weights, and $\mu_k = 0.0001$ for the D-LMS with adaptive clustering. In Fig. 3(b), the changed parameter settings are $\mu_k = 0.0001$ for the D-LMS with adaptive weights and adaptive clustering, $\mu_k = 0.007$ for the ACD-NFDAF, $\mu_k = 0.0016$ for the unnormalized ACD-PBFDAF, and $\mu_k = 0.0002$ for the ACD-NPBFDAF. Note that the D-LMS with uniform combination policy has a large mean-square deviation due to the indiscriminate cooperation among neighboring nodes. The D-LMS with adaptive weights also performs poorly in this scenario where the tasks to be estimated are not well-separated. It is clear that the ACD-NPBFDAF algorithm achieves faster convergence speed or less signal path delay than competing algorithms. The fast convergence speed is achieved by utilizing the normalization matrix $\overline{\mathbf{\Lambda}}_k(j)$, which assigns an individual step-size to each frequency bin.

VI. Conclusion

In this paper, we have proposed the ACD-NPBFDAF algorithm over multitask networks. It mainly includes five steps: signal FFT transform, normalized matrix update, weight update, calculating the combination coefficients and combination. Compared with time domain diffusion algorithms, the proposed algorithm has lower computation complexity, especially for the multi-tap filter. Furthermore, the ACD-NPBFDAF algorithm provides less signal path delay than the D-FDAF algorithm. The simulation results verify the effectiveness.

References

[1] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4/5, pp. 311–801, 2014.

[2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, no.2, pp. 17–29, Mar. 2002.

[3] A. H. Sayed, S. Tu, J. Chen, X. Zhao and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155-171, May. 2013.

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

[5] 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 Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 638–649, Aug. 2017.

[6] S. Zhang and W. X. Zheng, "Normalized least mean-square algorithm with variable step size based on diffusion strategy," in *Proc. IEEE Int. Symp. Circuits Syst.*, Florence, Italy, May 2018, pp. 1–4.

[7] J. Chen, C. Richard and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129-4144, Aug. 2014.

[8] R. Nassif, C. Richard, A. Ferrari and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6329-6344, Dec. 2016.

[9] V. C. Gogineni and M. Chakraborty, "Improving the performance of multitask diffusion APA via controlled inter-cluster cooperation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 903–912, Mar. 2020.

[10] X. Cao and K. J. R. Liu, "Decentralized sparse multitask RLS over networks," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6217–6232, Dec. 2017.

[11] F. Chen, X. Li, S. Duan, L. Wang, and J. Wu, "Diffusion generalized maximum correntropy criterion algorithm for distributed estimation over multitask network," *Digit. Signal Process.*, vol. 81, pp. 16–25, Oct. 2018.

[12] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," *Proc. Int. Workshop Cognit. Inf. Process. (CIP)*, Paradorde Baiona, Spain, May 2012, pp. 1–6.

[13] J. Chen, C. Richard and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733-2748, June. 2015.

[14] Q. Shi, F. Chen, X. Li, and S. Duan, "Distributed adaptive clustering learning over time-varying multitask networks," *Inf. Sci.*, vol. 567, pp. 278–297, Aug. 2021.

[15] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Conf. Neural Inf. Process. Sys.*, Long Beach, CA, USA, 2017.

[16] V. C. Gogineni, S. Werner, Y. -F. Huang, and A. Kuh, "Decentralized graph federated multitask learning for streaming data," *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, USA, Mar. 2022, pp. 101-106.

[17] E. Ferrara, "Fast implementations of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 474-475, Aug. 1980.

[18] S. Zhang, F. Zhang, H. Chen, R. Merched, and A. H. Sayed, "Frequency-domain diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 69, pp. 5419–5430, Aug. 2021.

[19] S. Zhang, J. Zhang, Y. Xia and H. C. So, "Adaptive frequency-domain normalized implementations of widely-linear complex-valued filter," *IEEE Trans. Signal Process.*, vol. 69, pp. 5801-5814, Oct. 2021.

[20] F. Yang, G. Enzner, and J. Yang, "A unified approach to the statistical convergence analysis of frequency-domain adaptive filters," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1785-1796, Apr. 2019.

[21] F. Yang and J. Yang, "Convergence analysis of deficient-length frequency-domain adaptive filters," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4242-4255, Nov. 2019.

[22] K. P. J. Soo, "Multidelay block frequency domain adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing.*, vol. 38, pp. 373-376, 1990.

[23] K. S. Chan and B. Farhang-Boroujeny, "Analysis of the partitioned frequency-domain block LMS (PFBLMS) algorithm," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 1860-1874, Sept. 2001.

[24] J. M. P. Borrallo and M. G. Otero, "On the implementation of a partitioned block frequency domain adaptive filter (PBFDAF) for long acoustic echo cancellation," *Signal Process.*, vol. 27, pp. 301–315, Jun. 1992.

[25] Y. Zhou, J. Chen, and X. Li, "A time/frequency-domain unified delayless partitioned block frequency-domain adaptive filter," *IEEE Signal Process. Lett.*, vol. 14, no. 12, pp. 976–979, Dec. 2007.

[26] F. Yang, G. Enzner, and J. Yang, "Transient analysis of partitioned-block frequency-domain adaptive filters," in *Proc. Eur. Signal Process. Conf.*, 2019, pp. 1–5.