Compressive Diffusion Strategies Over Distributed Networks for Reduced Communication Load

Muhammed O. Sayin and Suleyman Serdar Kozat, *Senior Member, IEEE*

***Abstract*—We study the compressive diffusion strategies over distributed networks based on the diffusion implementation and adaptive extraction of the information from the compressed diffusion data. We demonstrate that one can achieve a comparable performance to the full information exchange configurations, even if the diffused information is compressed into a scalar or a single bit, i.e., a tremendous reduction in the communication load. To this end, we provide a complete performance analysis for the compressive diffusion strategies. We analyze the transient, the steady-state and the tracking performances of the configurations in which the diffused data is compressed into a scalar or a single-bit. We propose a new adaptive combination method improving the convergence performance of the compressive diffusion strategies further. In the new method, we introduce one more freedom-of-dimension in the combination matrix and adapt it by using the conventional mixture approach in order to enhance the convergence performance for any possible combination rule used for the full diffusion configuration. We demonstrate that our theoretical analysis closely follow the ensemble averaged results in our simulations. We provide numerical examples showing the improved convergence performance with the new adaptive combination method while tremendously reducing the communication load.**

***Index Terms*—Compressed diffusion, distributed network, performance analysis.**

I. Introduction

DISTRIBUTED network of nodes provides enhanced performance for several different applications, such as source tracking, environment monitoring and source localization [1]–[4]. In such a distributed network, each node encounters possibly a different statistical profile, which provides broadened perspective on the monitored phenomena. In general, we could reach the best estimate with access to all observation data across the whole network since the observation of each node carries valuable information [5]. In the distributed adaptive estimation framework, we distribute the processing over the network and allow the information exchange among the nodes so that the parameter estimate of each node converges to the best estimate [4], [6].

Manuscript received December 20, 2013; revised April 19, 2014 and July 21, 2014; accepted July 26, 2014. Date of publication August 14, 2014; date of current version September 04, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hing Cheung So. This work was supported in part by the Outstanding Researcher Programme of Turkish Academy of Sciences and TUBITAK projects, Contract no: 112E161 and 113E517.

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, Bilkent, Ankara 06800, Turkey (e-mail: sayin@ee.bilkent.edu.tr; kozat@ee.bilkent.edu.tr).

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.2014.2347917

In the distributed architectures, one can use different approaches to regulate the information exchange among nodes such as the diffusion implementations [6]–[11]. The generic diffusion implementation defines a communication protocol in which only the nodes from a certain neighborhood could exchange information with each other [1], [6]–[11]. In this framework, each node uses a local adaptive algorithm and improves its parameter estimation by fusing its information with the diffused parameter estimations of the neighboring nodes. Via this information sharing, the diffusion approach provides robustness against link failures and changing network topologies [6]. However, diffusion of the parameter vectors within the neighborhoods results in high amount of communication load. For example, in a typical diffusion network of N nodes the overall communication burden is given by $N \times M$ where M is the size of the diffused vector, which implies that the size of the diffused information has a multiplicative impact on the overall communication burden. Additionally, in a wireless network, the neighborhood size also plays a crucial role on the overall communication load since the larger the neighborhood is, the more power is required in the transmission of the information [1]–[4].

We study the compressive diffusion strategies that achieve a better trade-off in terms of the amount of cooperation and the required communication load [12]. Unlike the full diffusion configuration, the compressed diffusion approach diffuses a single-bit of information or a reduced dimensional data vector achieving an impressive reduction in the communication load, i.e., from a full vector to a single bit or to a single scalar. The diffused data is generated through certain random projections of the local parameter estimation vector. Then, the neighboring nodes adaptively construct the original parameter estimations based on the diffused information and fuse their individual estimates for the final estimate. In this sense, this approach reduces the communication load in the spirit of the compressive sensing [12], [13]. The compression is lossy since we do not assume any sparseness or compressibility on the parameter estimation vector [13], [14]. However, the compressive diffusion approach achieves comparable convergence performance to the full diffusion configurations. Since the communication load increases far more in the large networks or the networks where the paths among the nodes are relatively longer, the compressive diffusion strategies play a crucial role in achieving comparable convergence performance with significantly reduced communication load.

There exist several other approaches that seek to reduce the communication load in distributed networks. In [15], [16] and [17], the authors propose the partial diffusion strategies where the nodes diffuse only selected coefficients of the parameter es-

timation vector. In [18], the dimension of the diffused information is reduced through the Krylov subspace projection techniques in the set-theoretic estimation framework. In [19], within a predefined neighborhood, the parameter estimate is quantized before the diffusion in order to avoid unlimited bandwidth requirement. In [20], the nodes transmit the sign of the innovation sequence in the decentralized estimation framework. In [21], in a consensus network, the relative difference between the states of the nodes is exchanged by using a single bit of information. As distinct from the mentioned works, the compressive diffusion strategies substantially compress the whole diffused information and extract the information from the compressed data adaptively [12].

In this paper, we provide a complete performance analysis for the compressive diffusion strategies, which demonstrates comparable convergence performance of the compressed diffusion to the full information exchange configuration. We note that studying the performance of distributed networks with compressive diffusion strategies is not straight-forward since adaptive extraction of information from the diffused data brings in an additional adaptation level. Moreover, such a theoretical analysis is rather challenging for the single-bit diffusion strategy due to the highly nonlinear compression. However, we analyze the transient, steady-state and tracking performance of the configurations in which the diffused data is compressed into a scalar or a single-bit. We also propose a new adaptive combination method improving the performance for any conventional combination rule. In the compressive diffusion framework, we fuse the local estimates with the adaptively extracted information from substantially compressed diffusion data. The extracted information carries relatively less information than the original data. Hence, we introduce "a confidence parameter" concept, which adds one more freedom-of-dimension in the combination matrix. The confidence parameter determines how much we are confident with the local parameter estimation. Through the adaptation of the confidence parameter, we observe enormous enhancement in the convergence performance of the compressive diffusion strategies even for relatively long filter lengths.

Our main contributions include: 1) for Gaussian regressors, we analyze the transient, steady-state and tracking performance of scalar and single-bit diffusion techniques; 2) We demonstrate that our theoretical analysis accurately models the simulated results; 3) We propose a new adaptive combination method for compressive diffusion strategies, which achieves a better trade-off in terms of the transient and steady state performance; 4) We provide numerical examples showing the enhanced convergence performance with the new adaptive combination method in our simulations.

We organize the paper as follows. In Section II, we explain the distributed network and diffusion implementation. In Section III, we introduce the compressive diffusion strategy, i.e., reduced-dimension and single-bit diffusion. In Section IV, we provide a global recursion model for the deviation parameters to facilitate the performance analysis. For Gaussian regressors, we analyze the mean-square convergence performance of the scalar and single-bit diffusion strategies in Sections V and VI, respectively. In Sections VII and VIII we analyze the steady-state and tracking performance of the scalar and single-bit diffusion approaches. In Section IX, we introduce the confidence parameter and propose a new adaptive combination method, improving the convergence performance of the compressive diffusion strategies. In Section X, we provide numerical examples demonstrating the match of theoretical and simulated results, and enhanced convergence performance with the new adaptive combination technique. We conclude the paper in Section XI with several remarks.

Notation: Bold lower (or upper) case letters denote column vectors (or matrices). For a vector $\mathbf{a}$ (or matrix $\mathbf{A}$), $\mathbf{a}^T$ (or $\mathbf{A}^T$) is its ordinary transpose. $\|\cdot\|$ and $\|\cdot\|_{\mathbf{A}}$ denote the L_2 norm and the weighted L_2 norm with the matrix $\mathbf{A}$, respectively (provided that $\mathbf{A}$ is positive-definite). We work with real data for notational simplicity. For a random variable x (or vector $\mathbf{x}$), $E[x]$ (or $E[\mathbf{x}]$) represents its expectation. Here, $\mathrm{Tr}(\mathbf{A})$ denotes the trace of the matrix $\mathbf{A}$. The operator $\mathrm{col}\{\cdot\}$ produces a column vector or a matrix in which the arguments of $\mathrm{col}\{\cdot\}$ are stacked one under the other. For a matrix argument, $\mathrm{diag}\{\mathbf{A}\}$ operator constructs a diagonal matrix with the diagonal entries of $\mathbf{A}$ and for a vector argument, it creates a diagonal matrix whose diagonal entries are elements of the vector. The operator $\otimes$ takes the Kronecker tensor product of two matrices.

II. Distributed Network

Consider a network of N nodes where each node i measures[1] $d_{i,t}$ and $\mathbf{u}_{i,t} \in \mathbb{R}^M$ related via the true parameter vector $\mathbf{w_o} \in \mathbb{R}^M$ through a linear model

$$d_{i,t} = \mathbf{w_o}^T \mathbf{u}_{i,t} + v_{i,t},$$

where $v_{i,t}$ denotes the temporally and spatially white noise. We assume that the regression vector $\mathbf{u}_{i,t}$ is spatially and temporally uncorrelated with the other regressors and the observation noise. If we know the whole temporal and spatial data overall network, then we can obtain the parameter of interest $\mathbf{w_o}$ by minimizing the following global cost with respect to the parameter estimate $\mathbf{w}$ [6]:

$$J_{\mathrm{glob}}(\mathbf{w}) = \sum_{i=1}^{N} E[(d_{i,t} - \mathbf{w}^T \mathbf{u}_{i,t})^2]. \tag{1}$$

The stochastic gradient update for (1) leads to the global least-mean square (LMS) algorithm as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu \sum_{i=1}^{N} \mathbf{u}_{i,t} \left(d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_t\right), \tag{2}$$

where $\mu > 0$ is the step size [7]. Note that (2) brings a significant communication burden by gathering the network-wise information in a central processing unit. Additionally, centralized approach is not robust against link failures and changing network statistics [4], [6]. On the other hand, in the diffusion implementation framework, we utilize a protocol in which each node i can only exchange information with the nodes from its neighborhood $\mathcal{N}_i$ (with the convention $i \in \mathcal{N}_i$) [6], [7]. This protocol distributes the processing to the nodes and provides tracking ability for time-varying statistical profiles [6].

[1]Although we assume a time invariant unknown system vector $\mathbf{w_o}$, we also provide the tracking performance analysis for certain non-stationary models later in the paper.

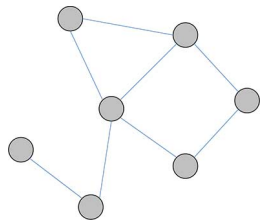

Fig. 1. Distributed network of nodes and the neighborhood $\mathcal{N}_i$.

Assuming the inner-node links are symmetric, we model the distributed network as an undirected graph where the nodes and the communication links correspond to its vertices and edges, respectively (See Fig. 1). In the distributed network, each node employs a local adaptation algorithm and benefits from the information diffused by the neighboring nodes in the construction of the final estimate [6]–[9]. For example, in [6], nodes *diffuse their parameter estimate* to the neighboring nodes and each node i performs the LMS algorithm given as

$$\mathbf{w}_{i,t+1} = \left(\mathbf{I} - \mu_i \mathbf{u}_{i,t}\mathbf{u}_{i,t}^T\right)\boldsymbol{\varphi}_{i,t} + \mu_i d_{i,t}\mathbf{u}_{i,t}, \tag{3}$$

where $\mu_i > 0$ is the local step-size. The intermediate parameter vector $\boldsymbol{\varphi}_{i,t}$ is generated through

$$\boldsymbol{\varphi}_{i,t} = \sum_{j\in\mathcal{N}_i} \gamma_{i,j}\mathbf{w}_{j,t}$$

with $\gamma_{i,j}$'s are the combination weights such that $\sum_{j=1}^{N}\gamma_{i,j} = 1$ for all $i \in \{1,\ldots,N\}$. For a given network topology, the combination weights are determined according to certain combination rules such as uniform [22], the Metropolis [23], [24], relative-degree rules [8] or adaptive combiners [25].

We note that in (3) we could assign $\boldsymbol{\varphi}_{i,t}$ as the final estimate in which we adapt the local estimate through the local observation data and then we fuse with the diffused estimates to generate the final estimate. In [7], authors examine these approaches as combine-than-adapt (CTA) and adapt-than-combine (ATC) diffusion strategies, respectively. In this paper, we study the ATC diffusion strategy, however, the theoretical results hold for both the ATC and the CTA cases for certain parameter changes provided later in the paper.

We emphasize that the diffusion of the parameter estimation vector also brings in high amount of communication load. In the next section, we introduce the compressive diffusion strategies enabling the adaptive construction of the required information from the reduced dimensional diffused information.

III. Compressive Diffusion

We seek to estimate the parameter of interest $\mathbf{w_o}$ through the *reduced dimension information exchange* within the neighborhoods. To this end, in the compressed diffusion approach, unlike the full diffusion scheme, we exchange a significantly reduced amount of information. The diffused information is generated through a certain projection operator, e.g., a time-variant vector $\mathbf{c}_t$, by linearly transforming the parameter vector, e.g., $\mathbf{w}_{i,t}$. In particular, node i diffuses $\mathbf{c}_t^T\mathbf{w}_{i,t}$ instead of the whole parameter vector $\mathbf{w}_{i,t}$ in the scalar diffusion scheme. We might also use a projection matrix, e.g., $\mathcal{C}_t \in \mathbb{R}^{M\times p}$, such that $\dim\{\mathcal{C}_t^T\mathbf{w}_{i,t}\} \ll \dim\{\mathbf{w}_{i,t}\}$ or $p \ll M$. Then the neighboring nodes of i can generate an estimate $\mathbf{a}_{i,t}$ of $\mathbf{w}_{i,t}$ through the exchanged information by using an adaptive estimation algorithm as explained later in this chapter and in [12]. We emphasize that the estimates $\mathbf{a}_{i,t}$'s are the constructed information using the diffused information, not the actual diffused information. Hence, the diffused information might have far smaller dimensions than the parameter estimation vector, which reduces the communication load significantly.

We note that the projection operator plays a crucial role in the construction of $\mathbf{a}_{i,t}$. Hence we constrain the projection operator to span the whole parameter space in order to avoid biased estimate of the original parameters [12]. Based on this constraint, we can construct the projection operator through the pseudo-random number generators (PRNG), which generates a sequence of numbers determined by a seed to approximate the properties of random numbers [26], or through a round-robin fashion in the sequential selection scheme as in [16].

Remark 3.1: We point out that the randomized projection vector could be generated at each node synchronously provided that each node uses the same *seed* for the pseudo-random generator mechanism [26]. Such seed exchanges and the synchronisation can be done periodically by using pilot signals without a serious increase in the communication load [27]. In Section X, we examine the sensitivity of the proposed strategies against the asynchronous events, e.g., complete loss of diffused information, in several scenarios through numerical examples.

Most of the conventional adaptive filtering algorithms can be derived using the following generic update:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}}\{D(\mathbf{w},\mathbf{w}_t) + \mu L(d_t,\mathbf{u}_t,\mathbf{w})\}, \tag{4}$$

where $D(\mathbf{w},\mathbf{w}_t)$ is the divergence, distance or *a priori* knowledge, e.g., the Euclidean distance $\|\mathbf{w}-\mathbf{w}_t\|^2$, and $L(d_t,\mathbf{u}_t,\mathbf{w})$ is the loss function, e.g., the mean square error $E[(d_t-\mathbf{u}_t^T\mathbf{w})]$ [28], [29]. Correspondingly, the diffusion based distributed estimation algorithms can also be generated through the update (4). However, note that the compressive diffusion scheme possesses different side information about the parameter of interest $\mathbf{w_o}$ from the full diffusion configuration, i.e., the constructed estimates instead of the original parameters. Although the constructed estimates $\mathbf{a}_{j,t}$'s track the original parameter estimation vectors, they are also parameter estimates of $\mathbf{w_o}$ as the original parameters. Particularly, in the proposed schemes, each node i has access to the *a priori* knowledge about the true parameter vector $\mathbf{w_o}$ as $\mathbf{w}_{i,t}$ and $\mathbf{a}_{j,t}$'s for $j \in \mathcal{N}_i \setminus i$. Hence, in the compressive diffusion implementation, we update according to

$$\mathbf{w}_{i,t+1} = \arg\min_{\mathbf{w}_i}\left\{\gamma_{ii}\|\mathbf{w}_i-\mathbf{w}_{i,t}\|^2 + \sum_{j\in\mathcal{N}_i\setminus i}\gamma_{ij}\|\mathbf{w}_i-\mathbf{a}_{j,t}\|^2 + \mu_i\left(d_{i,t}-\mathbf{u}_{i,t}^T\mathbf{w}_i\right)^2\right\} \tag{5}$$

such that in the update we also consider the Euclidean distance with the local parameter estimation $\mathbf{w}_{i,t}$ and the constructed estimates $\mathbf{a}_{j,t}$ of the neighboring nodes. In order to simplify the

optimization in (5) and to obtain an LMS update exactly, we can replace the loss term $(d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_i)^2$ with the first order Taylor series expansion around $\mathbf{a}_{j,t}$, i.e.,

$$\left(d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_i\right)^2 = \bar{e}_{i,t}(\mathbf{a}_{j,t})^2 - 2\bar{e}_{i,t}(\mathbf{a}_{j,t})\mathbf{u}_{i,t}^T(\mathbf{w}_i - \mathbf{a}_{j,t}) + O(\|\mathbf{w}_i\|^2), \quad (6)$$

where we denote $\bar{e}_{i,t}(\mathbf{a}_{j,t}) \triangleq d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{a}_{j,t}$. Similarly, the first order Taylor series expansion around $\mathbf{w}_{i,t}$ leads

$$\left(d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_i\right)^2 = e_{i,t}^2 - 2e_{i,t}\mathbf{u}_{i,t}^T(\mathbf{w}_i - \mathbf{w}_{i,t}) + O(\|\mathbf{w}_i\|^2), \quad (7)$$

where $e_{i,t} \triangleq d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_{i,t}$. Since $\sum_{j\in\mathcal{N}_i} \gamma_{ij} = 1$, the approximations (6) and (7) in (5) yields

$$\begin{aligned}\mathbf{w}_{i,t+1} &= \arg\min_{\mathbf{w}_i} \Bigg\{ \gamma_{ii}\|\mathbf{w}_i - \mathbf{w}_{i,t}\|^2 + \sum_{j\in\mathcal{N}_i\setminus i} \gamma_{ij}\|\mathbf{w}_i - \mathbf{a}_{j,t}\|^2 \\ &\quad + \mu_i\gamma_{ii}\left[e_{i,t}^2 - 2e_{i,t}\mathbf{u}_{i,t}^T(\mathbf{w}_i - \mathbf{w}_{i,t})\right] \\ &\quad + \mu_i \sum_{j\in\mathcal{N}_i\setminus i} \gamma_{ij}\left[\bar{e}_{i,t}(\mathbf{a}_{j,t})^2 - 2\bar{e}_{i,t}(\mathbf{a}_{j,t})\mathbf{u}_{i,t}^T(\mathbf{w}_i - \mathbf{a}_{j,t})\right] \Bigg\}. \end{aligned} \quad (8)$$

The minimized term in (8) is a convex function of $\mathbf{w}_i$ and the Hessian matrix $2\mathbf{I}_M \succ \mathbf{0}$ is positive definite. Hence, taking derivative and equating zero, we get the following update

$$\mathbf{w}_{i,t+1} = \boldsymbol{\varphi}_{i,t+1} + \mu_i \mathbf{u}_{i,t}\left(d_{i,t} - \mathbf{u}_{i,t}^T\boldsymbol{\varphi}_{i,t+1}\right), \quad (9)$$

where

$$\boldsymbol{\varphi}_{i,t+1} = \gamma_{ii}\mathbf{w}_{i,t} + \sum_{j\in\mathcal{N}_i\setminus i} \gamma_{ij}\mathbf{a}_{j,t}, \quad (10)$$

which is similar to the distributed LMS algorithm (3). Note that if we interchange $\boldsymbol{\varphi}_{i,t}$ and $\mathbf{w}_{i,t}$, in other words, when we assign the outcome of the combination as the final estimate rather than the outcome of the adaptation, we have the following algorithm:

$$\boldsymbol{\varphi}_{i,t+1} = \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t}\left(d_{i,t} - \mathbf{u}_{i,t}^T\mathbf{w}_{i,t}\right), \quad (11)$$

$$\mathbf{w}_{i,t+1} = \gamma_{ii}\boldsymbol{\varphi}_{i,t+1} + \sum_{j\in\mathcal{N}_i\setminus i} \gamma_{ij}\mathbf{a}_{j,t+1}. \quad (12)$$

We point out that (9) and (10) are the CTA diffusion strategy while (11) and (12) are the ATC diffusion strategy. Figs. 2 and 3 summarize the compressive diffusion strategy for the CTA and ATC strategies where $j_k \in \mathcal{N}_i$. We next introduce different approaches to generate the diffused information (which are used to construct $\mathbf{a}_{j,t+1}$'s).

In the compressive diffusion approach, instead of the full vector and irrespective of the final estimate, we always diffuse the linear transformation of the outcome of the adaptation, e.g., we diffuse $z_{i,t} = \mathbf{c}_t^T\mathbf{w}_{i,t}$ in the CTA strategy and $z_{i,t} = \mathbf{c}_t^T\boldsymbol{\varphi}_{i,t}$ in the ATC strategy. At each node, with the diffused information $z_{i,t}$, we update the constructed estimate $\mathbf{a}_{i,t}$ according to

$$\mathbf{a}_{i,t+1} = \arg\min_{\mathbf{a}_i}\left\{\|\mathbf{a}_i - \mathbf{a}_{i,t}\|^2 + \eta_i\left\|z_{i,t} - \mathbf{c}_t^T\mathbf{a}_i\right\|^2\right\},$$

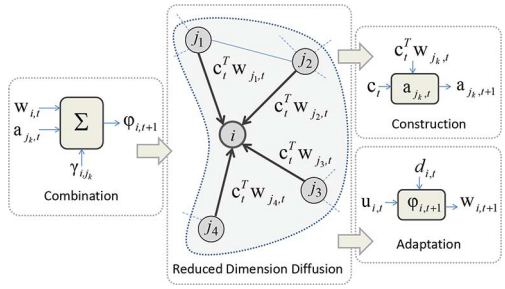

Fig. 2. CTA strategy in the scalar diffusion framework.

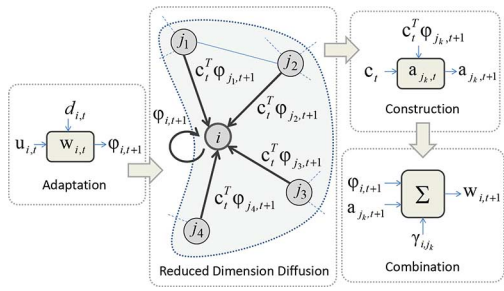

Fig. 3. ATC strategy in the scalar diffusion framework.

where we choose the diffused data as the desired signal and try to minimize the mean-square of the difference between the estimate $\hat{z}_{i,t} = \mathbf{c}_t^T\mathbf{a}_i$ and $z_{i,t}$. Here, $\mathbf{a}_{i,t}$'s are the estimates of the $\mathbf{w}_{i,t}$'s or $\boldsymbol{\varphi}_{i,t}$'s in the CTA and the ATC strategies, respectively. The first order Taylor series approximation of the loss term $\|z_{i,t} - \hat{z}_{i,t}\|^2$ around $\mathbf{a}_{i,t}$ yields the following update

$$\mathbf{a}_{i,t+1} = \mathbf{a}_{i,t} + \eta_i\mathbf{c}_t\left(z_{i,t} - \mathbf{c}_t^T\mathbf{a}_{i,t}\right) \quad (13)$$

where $\eta_i > 0$ is the construction step size. We note that in [12] the reduced dimension diffusion approach constructs $\mathbf{a}_{i,t+1}$'s through the minimum disturbance principle and resulted update involves $[\mathbf{c}_t^T\mathbf{c}_t]^{-1}$ as the normalization term. The constructed estimates $\mathbf{a}_{i,t+1}$'s are combined with the outcome of the local adaptation algorithm through (10) or (12).

We next introduce methods where the information exchange is only a single bit [12]. When we construct $\mathbf{a}_{i,t}$ at node i, assuming $\mathbf{a}_{i,t}$'s are initialized with the same value, node $j \in \mathcal{N}_i$ has knowledge of the constructed estimate $\mathbf{a}_{i,t}$. Hence, we can perform the construction update at each neighboring node via the diffusion of the estimation error, i.e., $\epsilon_{i,t} \triangleq z_{i,t} - \hat{z}_{i,t}$. Note that this does not influence the communication load, however, through the access to the exchange estimate $\mathbf{a}_{i,t+1}$ we can further reduce the communication load. Using the well-known sign algorithm [5], we can construct $\mathbf{a}_{i,t+1}$ as

$$\mathbf{a}_{i,t+1} = \mathbf{a}_{i,t} + \eta_i\mathbf{c}_t\mathrm{sign}(\epsilon_{i,t}). \quad (14)$$

Hence, we can repeat (14) at each neighboring node via the diffusion of $z_{i,t} = \mathrm{sign}(\epsilon_{i,t})$ only and then we combine with the local estimate by using (10) or (12).

In Table I, we tabulate the description of the proposed algorithms. We note that as seen in the Table I, the construction of $\mathbf{a}_{j,t}$ requires additional updates at each neighboring nodes.

TABLE I
THE DESCRIPTION OF THE COMPRESSIVE DIFFUSION SCHEMES WITH THE ATC STRATEGY

Algorithm 1: Scalar Diffusion Strategies - ATC

Initialization:

For $i = 1$ to N do

$\mathbf{u}_{i,0} = \mathbf{c}_0 = \mathbf{w}_{i,0} = \mathbf{a}_{i,0} = [0, \cdots, 0]^T$

End for

Do for $t \geq 0$

For $i = 1$ to N do

Adaptation:

$e_{i,t} = d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_{i,t}$

$\boldsymbol{\varphi}_{i,t+1} = \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} e_{i,t}$

Diffuse $z_{i,t} = \mathbf{c}_t^T \boldsymbol{\varphi}_{i,t}$ to neighboring nodes

Construction:

For all $j \in \mathcal{N}_i \setminus i$ do

$\epsilon_{j,t} = z_{j,t} - \mathbf{c}_t^T \mathbf{a}_{j,t}$

$\mathbf{a}_{j,t+1} = \mathbf{a}_{j,t} + \eta_j \mathbf{c}_t \epsilon_{j,t}$

End for

Combination:

$\mathbf{w}_{i,t+1} = \gamma_{i,i} \boldsymbol{\varphi}_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus i} \gamma_{i,j} \mathbf{a}_{j,t+1}$

End for

Algorithm 2: Single-bit Diffusion Strategies - ATC

Initialization:

For $i = 1$ to N do

$\mathbf{u}_{i,0} = \mathbf{c}_0 = \mathbf{w}_{i,0} = \mathbf{a}_{i,0} = [0, \cdots, 0]^T$

End for

Do for $t \geq 0$

For $i = 1$ to N do

Adaptation:

$e_{i,t} = d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_{i,t}$

$\epsilon_{i,t} = \mathbf{c}_t^T (\boldsymbol{\varphi}_{i,t} - \mathbf{a}_{i,t})$

$\mathbf{a}_{i,t+1} = \mathbf{a}_{i,t} + \eta_i \mathbf{c}_t \text{sign}(\epsilon_{i,t})$

$\boldsymbol{\varphi}_{i,t+1} = \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} e_{i,t}$

Diffuse $z_{i,t} = \text{sign}(\epsilon_{i,t})$ to neighboring nodes

Construction:

For all $j \in \mathcal{N}_i \setminus i$ do

$\mathbf{a}_{j,t+1} = \mathbf{a}_{j,t} + \eta_j \mathbf{c}_t z_{j,t}$

End for

Combination:

$\mathbf{w}_{i,t+1} = \gamma_{i,i} \boldsymbol{\varphi}_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus i} \gamma_{i,j} \mathbf{a}_{j,t+1}$

End for

However, in the following, we propose an approach significantly reducing this computational load provided that all nodes use the same projection operator. We note that (9) and (11) require the linear combination of the constructed estimates. To this end, we define

$$\boldsymbol{\omega}_{i,t} \stackrel{\triangle}{=} \sum_{j \in \mathcal{N}_i \setminus i} \gamma_{ij} \mathbf{a}_{j,t},$$

so that for the same step size, i.e., $\eta = \eta_1 = \cdots = \eta_N$, the following relations

$$\begin{aligned} \mathbf{a}_{1,t+1} &= \mathbf{a}_{1,t} + \eta_1 \mathbf{c}_t \left(z_{1,t} - \mathbf{c}_t^T \mathbf{a}_{1,t} \right), \\ &\vdots \\ \mathbf{a}_{N,t+1} &= \mathbf{a}_{N,t} + \eta_N \mathbf{c}_t \left(z_{N,t} - \mathbf{c}_t^T \mathbf{a}_{N,t} \right) \end{aligned}$$

can be rewritten in a single update as

$$\boldsymbol{\omega}_{i,t+1} = \boldsymbol{\omega}_{i,t} + \eta \, \mathbf{c}_t \left(\sum_{j \in \mathcal{N}_i \setminus i} \gamma_{i,j} z_{j,t} - \mathbf{c}_t^T \boldsymbol{\omega}_{i,t} \right). \quad (15)$$

In that sense, as an example, instead of (12), we can construct the final parameter estimate $\mathbf{w}_{i,t+1}$ through

$$\mathbf{w}_{i,t+1} = \gamma_{i,i} \boldsymbol{\varphi}_{i,t+1} + \boldsymbol{\omega}_{i,t+1}, \quad (16)$$

thanks to the linear error function in the LMS update. Hence, we can significantly reduce the computational load, i.e., to only an additional LMS update, in the scalar diffusion strategies through (15) and (16). On the other hand, if the sign algorithm is used at each node in the construction of $\mathbf{a}_{j,t}$, each node should construct $\mathbf{a}_{j,t}$'s separately since the sign algorithm has a nonlinear error update, i.e., $\text{sign}(\epsilon_{j,t})$. However, the sign algorithm is known for its low complexity implementation and can be implemented through shift-registers provided that the step-size is chosen as a power of 2 [5]. In this sense, the single-bit diffusion strategy significantly reduces the communication load, i.e., from continuum to a single bit, with a relatively small computational complexity increase. We point out that the single-bit diffusion also overcomes the bandwidth related issues especially in the wireless networks due to the significant reduction in the communication load and the inherently quantized diffusion data.

In the sequel, we introduce a global model gathering all network operations into a single update.

IV. GLOBAL MODEL

We can write the scalar (13) and single bit (14) diffusion approaches for the ATC diffusion strategy in a compact form as

$$\begin{aligned} \boldsymbol{\varphi}_{i,t+1} &= \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} e_{i,t}, & (17) \\ \mathbf{a}_{j,t+1} &= \mathbf{a}_{j,t} + \eta_j \mathbf{c}_t h(\epsilon_{j,t}), \\ \mathbf{w}_{i,t+1} &= \gamma_{i,i} \boldsymbol{\varphi}_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus i} \gamma_{i,j} \mathbf{a}_{j,t+1}, & (18) \end{aligned}$$

where $e_{i,t} \stackrel{\triangle}{=} d_{i,t} - \mathbf{u}_{i,t}^T \mathbf{w}_{i,t}$ and $\epsilon_{j,t} \stackrel{\triangle}{=} \mathbf{c}_t^T (\boldsymbol{\varphi}_{j,t} - \mathbf{a}_{j,t})$. For scalar and single bit diffusion approaches, $h(\epsilon_{j,t}) = \epsilon_{j,t}$ and $h(\epsilon_{j,t}) = \text{sign}(\epsilon_{j,t})$, respectively.

For the state-space representation that collects all network operations into a single update, we define $\boldsymbol{\varphi}_t = \text{col}\{\boldsymbol{\varphi}_{1,t}, \ldots, \boldsymbol{\varphi}_{N,t}\}$, $\mathbf{a}_t = \text{col}\{\mathbf{a}_{1,t}, \ldots, \mathbf{a}_{N,t}\}$, $\mathbf{w}_t = \text{col}\{\mathbf{w}_{1,t}, \ldots, \mathbf{w}_{N,t}\}$, $\underline{\mathbf{w}}_{\mathbf{o}} = \text{col}\{\mathbf{w}_{\mathbf{o}}, \ldots, \mathbf{w}_{\mathbf{o}}\}$ with $MN \times 1$ dimensions and $\mathbf{e}_t = \text{col}\{e_{1,t}, \ldots, e_{N,t}\}$, $\boldsymbol{\epsilon}_t = \text{col}\{\epsilon_{1,t}, \ldots, \epsilon_{N,t}\}$, $\mathbf{d}_t = \text{col}\{d_{1,t}, \ldots, d_{N,t}\}$, $\mathbf{v}_t = \text{col}\{v_{1,t}, \ldots, v_{N,t}\}$ with $N \times 1$ dimensions. For a given combination matrix $\boldsymbol{\Gamma} = [\gamma_{i,j}]$, we denote $\mathbf{G} \stackrel{\triangle}{=} \boldsymbol{\Gamma} \otimes \mathbf{I}_M$. Additionally, the regression and projection vectors yields the following $MN \times N$ global matrices

$$\mathbf{U}_t \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{u}_{1,t} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{u}_{N,t} \end{bmatrix}, \quad \mathbf{C}_t \stackrel{\triangle}{=} \begin{bmatrix} \mathbf{c}_{1,t} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{c}_{N,t} \end{bmatrix}.$$

Indeed, we can model the network with compressive diffusion strategy as a larger network in which each node i has an imaginary counterpart which diffuses $\mathbf{a}_{i,t}$ to the neighbors of i, which

is similar to the full diffusion configuration. The real nodes only get information from the imaginary nodes and do not diffuse any information. In that case, the network can be modelled as a directed graph with asymmetric inner node links and the combination matrix is given by

$$\tilde{\mathbf{\Gamma}} = \begin{bmatrix} \mathbf{\Gamma_D} & \mathbf{\Gamma_C} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

where $\mathbf{\Gamma_D} = \mathrm{diag}\{\mathbf{\Gamma}\}$ and $\mathbf{\Gamma_C} = \mathbf{\Gamma} - \mathbf{\Gamma_D}$. Then, we can write $\mathbf{w}_t$ in terms of $\boldsymbol{\varphi}_t$ and $\mathbf{a}_t$ as

$$\mathbf{w}_t = \mathbf{G_D}\boldsymbol{\varphi}_t + \mathbf{G_C}\mathbf{a}_t, \tag{19}$$

where $\mathbf{G_D} \triangleq \mathbf{\Gamma_D} \otimes \mathbf{I}_M$ and $\mathbf{G_C} \triangleq \mathbf{\Gamma_C} \otimes \mathbf{I}_M$. The state-space representation is given by

$$\begin{aligned} \boldsymbol{\varphi}_{t+1} &= \mathbf{w}_t + \mathbf{M}\mathbf{U}_t\mathbf{e}_t, \\ \mathbf{a}_{t+1} &= \mathbf{a}_t + \mathbf{N}\mathbf{C}_t\mathbf{h}(\boldsymbol{\epsilon}_t), \\ \mathbf{w}_{t+1} &= \mathbf{G_D}\boldsymbol{\varphi}_{t+1} + \mathbf{G_C}\mathbf{a}_{t+1}, \end{aligned} \tag{20}$$

where $\mathbf{h}(\boldsymbol{\epsilon}_t) = \mathrm{col}\{h(\epsilon_{1,t}), h(\epsilon_{2,t}), \ldots, h(\epsilon_{N,t})\}$, $\mathbf{M} \triangleq \mathrm{diag}\{[\mu_1, \ldots, \mu_N]\} \otimes \mathbf{I}_M$, and $\mathbf{N} \triangleq \mathrm{diag}\{[\eta_1, \ldots, \eta_N]\} \otimes \mathbf{I}_M$. We obtain the global deviation vectors as

$$\tilde{\boldsymbol{\varphi}}_t \triangleq \underline{\mathbf{w}}_\mathbf{o} - \boldsymbol{\varphi}_t \text{ and } \tilde{\mathbf{a}}_t \triangleq \underline{\mathbf{w}}_\mathbf{o} - \mathbf{a}_t. \tag{21}$$

Since $\mathbf{\Gamma}\underline{\mathbf{1}} = \underline{\mathbf{1}}$,

$$\mathbf{G}\underline{\mathbf{w}}_\mathbf{o} = \underline{\mathbf{w}}_\mathbf{o} \tag{22}$$

then the global deviation update yields

$$\tilde{\boldsymbol{\varphi}}_{t+1} = \mathbf{G_D}\tilde{\boldsymbol{\varphi}}_t + \mathbf{G_C}\tilde{\mathbf{a}}_t - \mathbf{M}\mathbf{U}_t\mathbf{e}_t, \tag{23}$$

$$\tilde{\mathbf{a}}_{t+1} = \tilde{\mathbf{a}}_t - \mathbf{N}\mathbf{C}_t\mathbf{h}(\boldsymbol{\epsilon}_t). \tag{24}$$

In (25),

$$\overbrace{\begin{bmatrix} \tilde{\boldsymbol{\varphi}}_{t+1} \\ \tilde{\mathbf{a}}_{t+1} \end{bmatrix}}^{\tilde{\boldsymbol{\psi}}_{t+1}} = \overbrace{\begin{bmatrix} \mathbf{G_D} & \mathbf{G_C} \\ \mathbf{0} & \mathbf{I}_{MN} \end{bmatrix}}^{\mathbf{X}} \overbrace{\begin{bmatrix} \tilde{\boldsymbol{\varphi}}_t \\ \tilde{\mathbf{a}}_t \end{bmatrix}}^{\tilde{\boldsymbol{\psi}}_t} - \overbrace{\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix}}^{\mathbf{D}} \overbrace{\begin{bmatrix} \mathbf{U}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_t \end{bmatrix}}^{\mathbf{Y}_t} \overbrace{\begin{bmatrix} \mathbf{e}_t \\ \mathbf{h}(\boldsymbol{\epsilon}_t) \end{bmatrix}}^{\underline{\mathbf{h}}(\mathbf{e}_t, \boldsymbol{\epsilon}_t)} \tag{25}$$

we represent the global deviation updates (23) and (24) in a single equation or equivalently

$$\tilde{\boldsymbol{\psi}}_{t+1} = \mathbf{X}\tilde{\boldsymbol{\psi}}_t - \mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t, \boldsymbol{\epsilon}_t), \tag{26}$$

where $\tilde{\boldsymbol{\psi}}_t \triangleq \mathrm{col}\{\tilde{\boldsymbol{\varphi}}_t, \tilde{\mathbf{a}}_t\}$. We next use the following assumptions in the analyses of the weighted-energy recursion of (26):

Assumption 1:
The regressor signal $\mathbf{u}_{i,t}$ is zero-mean independently and identically distributed (i.i.d.) Gaussian random vector process and spatially and temporally independent from the other regressor signals, the randomized projection operator and the observation noise. Each node uses spatially and temporally independent projection vector, i.e., $\mathbf{c}_{i,t}$. The projection operator is zero-mean i.i.d. Gaussian random vector process and the observation noise $v_{i,t}$ is also a zero-mean i.i.d. Gaussian random variable. Note that such assumptions are commonly used in the analysis of traditional adaptive schemes [5], [30].

Assumption 2:
The *a priori* estimation error vector in the construction update (20), i.e., $\boldsymbol{\epsilon}_{a,t} \triangleq \mathbf{C}_t^T(\tilde{\mathbf{a}}_t - \tilde{\boldsymbol{\varphi}}_t)$, has Gaussian distribution and it is jointly Gaussian with the weighted *a priori* estimation error vector, i.e., $\mathbf{C}_t^T\mathbf{\Sigma}(\tilde{\mathbf{a}}_t - \tilde{\boldsymbol{\varphi}}_t)$, for any constant matrix $\mathbf{\Sigma}$. This assumption is reasonable for long filters, i.e., M is large, sufficiently small step size η_i's and by the Assumption 1 [31]. We adopt the Assumption 2 in the analyses of the single-bit diffusion schemes due to the nonlinearity in the corresponding construction update.

We point out that the Assumptions 1 and 2 are impractical in general, however, widely used in the adaptive filtering literature to analyze the performance of the schemes analytically due to the mathematical tractability and the analytical results match closely with the ensemble averaged simulation results. In the next sections, we analyze the mean-square convergence performance of the proposed approaches separately.

V. Scalar Diffusion With Gaussian Regressors

For the one-dimension diffusion approach, (26) yields

$$\tilde{\boldsymbol{\psi}}_{t+1} = \mathbf{X}\tilde{\boldsymbol{\psi}}_t - \mathbf{D}\mathbf{Y}_t\underline{\mathbf{e}}_t, \tag{27}$$

where $\underline{\mathbf{e}}_t \triangleq \mathrm{col}\{\mathbf{e}_t, \boldsymbol{\epsilon}_t\}$. By (19), (21) and (22), we note that $\mathbf{e}_t$ is given by

$$\mathbf{e}_t = \mathbf{U}_t^T(\mathbf{G_D}\tilde{\boldsymbol{\varphi}}_t + \mathbf{G_C}\tilde{\mathbf{a}}_t) + \mathbf{v}_t. \tag{28}$$

Similarly, we have

$$\boldsymbol{\epsilon}_t = \mathbf{C}_t^T(-\tilde{\boldsymbol{\varphi}}_t + \tilde{\mathbf{a}}_t). \tag{29}$$

Hence, through (28) and (29), we obtain the global estimation error $\underline{\mathbf{e}}_t$ as

$$\begin{aligned} \underline{\mathbf{e}}_t &= \begin{bmatrix} \mathbf{U}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_t \end{bmatrix}^T \underbrace{\begin{bmatrix} \mathbf{G_D} & \mathbf{G_C} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}}_{\mathbf{Z}} \begin{bmatrix} \tilde{\boldsymbol{\varphi}}_t \\ \tilde{\mathbf{a}}_t \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{v}_t \\ \mathbf{0} \end{bmatrix}}_{\mathbf{n}_t} \\ &= \mathbf{Y}_t^T\mathbf{Z}\tilde{\boldsymbol{\psi}}_t + \mathbf{n}_t. \end{aligned} \tag{30}$$

Through (30), we rewrite (27) as

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_{t+1} &= \mathbf{X}\tilde{\boldsymbol{\psi}}_t - \mathbf{D}\mathbf{Y}_t\left(\mathbf{Y}_t^T\mathbf{Z}\tilde{\boldsymbol{\psi}}_t + \mathbf{n}_t\right) \\ &= \left(\mathbf{X} - \mathbf{D}\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}\right)\tilde{\boldsymbol{\psi}}_t - \mathbf{D}\mathbf{Y}_t\mathbf{n}_t. \end{aligned} \tag{31}$$

We utilize the weighted-energy relation relating the energy of the error and deviation quantities in the performance analyzes through a weighting matrix $\mathbf{\Sigma}$. Then, we obtain

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_{t+1}^T\mathbf{\Sigma}\tilde{\boldsymbol{\psi}}_{t+1} = \tilde{\boldsymbol{\psi}}_t^T &\left(\mathbf{X} - \mathbf{D}\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}\right)^T\mathbf{\Sigma}\left(\mathbf{X} - \mathbf{D}\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}\right)\tilde{\boldsymbol{\psi}}_t \\ &- 2\mathbf{n}_t^T\mathbf{Y}_t^T\mathbf{D}\mathbf{\Sigma}\left(\mathbf{X} - \mathbf{D}\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}\right)\tilde{\boldsymbol{\psi}}_t \\ &+ \mathbf{n}_t^T\mathbf{Y}_t^T\mathbf{D}\mathbf{\Sigma}\mathbf{D}\mathbf{Y}_t\mathbf{n}_t. \end{aligned}$$

By the Assumption 1, the observation noise $\mathbf{v}_t$ is independent from the network statistics and the weighted energy relation for (31) is given by

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|_{\boldsymbol{\Sigma}}^2 = E\|\tilde{\boldsymbol{\psi}}_t\|_{\boldsymbol{\Sigma}'}^2 + E\left[\mathbf{n}_t^T\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\mathbf{n}_t\right], \quad (32)$$

where

$$\boldsymbol{\Sigma}' \triangleq \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X} - \mathbf{Z}^T\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{X} - \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z} + \mathbf{Z}^T\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}.$$

Apart form the weighting matrix $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}'$ is random due to the data dependence. By the Assumption 1, $\mathbf{Y}_t$ is independent of $\tilde{\boldsymbol{\psi}}_t$ and we can replace $\boldsymbol{\Sigma}'$ by its mean value, i.e., $\boldsymbol{\Sigma}' = E[\boldsymbol{\Sigma}']$ [5], [6]. Hence, the weighting matrix is given by

$$\boldsymbol{\Sigma}' = \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X} - \mathbf{Z}^T E\left[\mathbf{Y}_t\mathbf{Y}_t^T\right]\mathbf{D}\boldsymbol{\Sigma}\mathbf{X} - \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}E\left[\mathbf{Y}_t\mathbf{Y}_t^T\right]\mathbf{Z} + \mathbf{Z}^T\mathbf{D}E\left[\mathbf{Y}_t\mathbf{Y}_t^T\boldsymbol{\Sigma}\mathbf{Y}_t\mathbf{Y}_t^T\right]\mathbf{D}\mathbf{Z}. \quad (33)$$

Note that in the last term of the right hand side (RHS) of (33), we take $\mathbf{D}$'s out of the expectation thanks to the block diagonal structure of $\mathbf{D}$ and $\mathbf{Y}_t\mathbf{Y}_t^T$.

In order to calculate certain data moments in (32) and (33), by the Assumption 1, we obtain

$$\boldsymbol{\Lambda}_{\boldsymbol{u}} \triangleq E\left[\mathbf{U}_t\mathbf{U}_t^T\right] = \mathrm{diag}\left\{\left[\sigma_{u,1}^2, \sigma_{u,2}^2, \ldots, \sigma_{u,N}^2\right]\right\} \otimes \mathbf{I}_M$$
$$\boldsymbol{\Lambda}_{\boldsymbol{c}} \triangleq E\left[\mathbf{C}_t\mathbf{C}_t^T\right] = \mathrm{diag}\left\{\left[\sigma_{c,1}^2, \sigma_{c,2}^2, \ldots, \sigma_{c,N}^2\right]\right\} \otimes \mathbf{I}_M.$$

Then, we obtain

$$\boldsymbol{\Lambda} \triangleq E\left[\mathbf{Y}_t\mathbf{Y}_t^T\right] = \begin{bmatrix} \boldsymbol{\Lambda}_{\boldsymbol{u}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_{\mathbf{c}} \end{bmatrix}.$$

In the performance analysis, convenient vectorisation notation is used to exploit the diagonal structure of matrices [5], [32]. In (32) and (33), matrices have block diagonal structures, thus we use the block vectorisation operator $\mathrm{bvec}\{\cdot\}$ [6] such that given an $NM \times NM$ block matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \ldots & \boldsymbol{\Sigma}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{N1} & \ldots & \boldsymbol{\Sigma}_{NN} \end{bmatrix},$$

where each block $\boldsymbol{\Sigma}_{ij}$ is a $M \times M$ block, $\boldsymbol{\sigma}_{ij} = \mathrm{vec}\{\boldsymbol{\Sigma}_{ij}\}$ with standard $\mathrm{vec}\{\cdot\}$ operator and $\boldsymbol{\sigma}_j = \mathrm{col}\{\boldsymbol{\sigma}_{1j}, \boldsymbol{\sigma}_{2j}, \ldots, \boldsymbol{\sigma}_{Nj}\}$, then

$$\mathrm{bvec}\{\boldsymbol{\Sigma}\} = \boldsymbol{\sigma} = \mathrm{col}\{\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_N\}. \quad (34)$$

We also use the *block Kronecker product* of two block matrices $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} \odot \mathbf{B}$. The ij-block is given by

$$[\mathbf{A} \odot \mathbf{B}]_{ij} = \begin{bmatrix} \mathbf{A}_{ij} \otimes \mathbf{B}_{11} & \ldots & \mathbf{A}_{ij} \otimes \mathbf{B}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{ij} \otimes \mathbf{B}_{N1} & \ldots & \mathbf{A}_{ij} \otimes \mathbf{B}_{NN} \end{bmatrix}. \quad (35)$$

The block vectorisation operator $\mathrm{bvec}\{\cdot\}$ (34) and the block Kronecker product (35) are related by

$$\mathrm{bvec}\{\mathbf{A}\boldsymbol{\Sigma}\mathbf{B}\} = (\mathbf{B}^T \odot \mathbf{A})\boldsymbol{\sigma} \quad (36)$$

and

$$\mathrm{Tr}\{\mathbf{A}^T\mathbf{B}\} = (\mathrm{bvec}\{\mathbf{A}\})^T\mathrm{bvec}\{\mathbf{B}\}. \quad (37)$$

The term in the RHS of (32) yields

$$E\left[\mathbf{n}_t^T\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\mathbf{n}_t\right] = \mathrm{Tr}\left(\boldsymbol{\Lambda}\mathbf{D}^2E\left[\mathbf{n}_t\mathbf{n}_t^T\right]\boldsymbol{\Sigma}\right)$$

and let

$$E\left[\mathbf{n}_t\mathbf{n}_t^T\right] = \mathbf{R}_{\mathbf{n}} = \begin{bmatrix} \mathbf{R}_{\mathbf{v}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{R}_{\mathbf{v}} \triangleq \mathrm{diag}\{\sigma_{v,1}^2, \ldots, \sigma_{v,N}^2\} \otimes \mathbf{I}_M$. Then by (37),

$$E\left[\mathbf{n}_t^T\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\mathbf{n}_t\right] = \mathbf{b}^T\boldsymbol{\sigma},$$

where

$$\mathbf{b} \triangleq \mathrm{bvec}\{\mathbf{R}_{\mathbf{n}}\mathbf{D}^2\boldsymbol{\Lambda}\}. \quad (38)$$

The last term on the RHS of (33) yields $\mathbf{A} = E[\mathbf{Y}_t\mathbf{Y}_t^T\boldsymbol{\Sigma}\mathbf{Y}_t\mathbf{Y}_t^T]$, where the $M \times M$ block is given by

$$[\mathbf{A}]_{ij} = \begin{cases} \boldsymbol{\Lambda}_i\left(\boldsymbol{\Sigma}_{ii} + \boldsymbol{\Sigma}_{ii}^T\right)\boldsymbol{\Lambda}_i + \boldsymbol{\Lambda}_i\mathrm{Tr}(\boldsymbol{\Sigma}_{ii}\boldsymbol{\Lambda}_i) & i = j \\ \boldsymbol{\Lambda}_i\boldsymbol{\Sigma}_{ij}\boldsymbol{\Lambda}_j & i \neq j \end{cases}$$

by the Assumption 1 [5]. The matrix $\boldsymbol{\Lambda}$ could be denoted as $\boldsymbol{\Lambda} = \mathrm{diag}\{\boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_{2N}\}$ where $\boldsymbol{\Lambda}_i$ for $i = \{1, 2, \ldots, 2N\}$ is $M \times M$ block matrix, e.g., $\boldsymbol{\Lambda}_1 = \sigma_{u,1}^2\mathbf{I}_M$ or $\boldsymbol{\Lambda}_{N+1} = \sigma_{c,1}^2\mathbf{I}_M$. The $M \times M$ ijth block of $\boldsymbol{\Sigma}$ is denoted by $\boldsymbol{\Sigma}_{ij}$.

Remark 5.1: We note that if each node used the same projection operator, $\mathbf{c}_{i,t}$'s would be spatially dependent. In that case, $[\mathbf{A}]_{ij}$ is defined as

$$[\mathbf{A}]_{ij} = \begin{cases} \boldsymbol{\Lambda}_i\left(\boldsymbol{\Sigma}_{ii} + \boldsymbol{\Sigma}_{ii}^T\right)\boldsymbol{\Lambda}_i + \boldsymbol{\Lambda}_i\mathrm{Tr}(\boldsymbol{\Sigma}_{ii}\boldsymbol{\Lambda}_i) & i = j, \\ \boldsymbol{\Lambda}_i\left(\boldsymbol{\Sigma}_{ij} + \boldsymbol{\Sigma}_{ij}^T\right)\boldsymbol{\Lambda}_j + \boldsymbol{\Lambda}_i\mathrm{Tr}(\boldsymbol{\Sigma}_{ij}\boldsymbol{\Lambda}_j) & i > N \wedge j > N, \\ \boldsymbol{\Lambda}_i\boldsymbol{\Sigma}_{ij}\boldsymbol{\Lambda}_j & \text{otherwise.} \end{cases}$$

Through (35), (37), we obtain $\mathrm{bvec}\{\mathbf{A}\} = \mathcal{A}\boldsymbol{\sigma}$ with $\mathcal{A} = \mathrm{diag}\{\mathcal{A}_1, \ldots, \mathcal{A}_{2N}\}$, $\mathcal{A}_j = \mathrm{diag}\{\mathcal{A}_{1j}, \ldots, \mathcal{A}_{2Nj}\}$ and

$$\mathcal{A}_{ij} = \begin{cases} 2\boldsymbol{\Lambda}_i \otimes \boldsymbol{\Lambda}_i + \boldsymbol{\lambda}_i\boldsymbol{\lambda}_i^T & i = j \\ \boldsymbol{\Lambda}_i \otimes \boldsymbol{\Lambda}_j & i \neq j \end{cases}$$

where $\boldsymbol{\lambda}_i = \mathrm{vec}\{\boldsymbol{\Lambda}_i\}$.

Hence, the block vectorization of the weighting matrix $\boldsymbol{\Sigma}'$ (33) yields

$$\mathrm{bvec}\{\boldsymbol{\Sigma}'\} = (\mathbf{X}^T \odot \mathbf{X}^T - (\mathbf{X}^T \odot \mathbf{Z}^T)(\mathbf{I}_{2MN} \odot \boldsymbol{\Lambda}\mathbf{D}) - (\mathbf{Z}^T \odot \mathbf{X}^T)(\boldsymbol{\Lambda}\mathbf{D} \odot \mathbf{I}_{2MN}) + (\mathbf{Z}^T \odot \mathbf{Z}^T)(\mathbf{D} \odot \mathbf{D})\mathcal{A})\boldsymbol{\sigma}.$$

For notational simplicity, we change the weighted-norm notation such that $\|\tilde{\boldsymbol{\varphi}}_t\|_{\boldsymbol{\sigma}}^2$ refers to $\|\tilde{\boldsymbol{\varphi}}_t\|_{\boldsymbol{\Sigma}}^2$ where $\boldsymbol{\sigma} = \mathrm{bvec}\{\boldsymbol{\Sigma}\}$. As a result, we obtain the weighted-energy recursion as

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|_{\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{\psi}}_t\|_{\mathbf{F}\boldsymbol{\sigma}}^2 + \mathbf{b}^T\boldsymbol{\sigma} \quad (39)$$

$$\mathbf{F} \triangleq \mathbf{X}^T \odot \mathbf{X}^T + (\mathbf{Z}^T \odot \mathbf{Z}^T)(\mathbf{D} \odot \mathbf{D})\mathcal{A} - (\mathbf{X}^T \odot \mathbf{Z}^T)(\mathbf{I}_{2MN} \odot \boldsymbol{\Lambda}\mathbf{D}) - (\mathbf{Z}^T \odot \mathbf{X}^T)(\boldsymbol{\Lambda}\mathbf{D} \odot \mathbf{I}_{2MN}). \quad (40)$$

TABLE II
INITIAL CONDITIONS AND WEIGHTING MATRICES FOR DIFFERENT CONFIGURATIONS

Framework	$E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\Sigma}}$	$E\|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Sigma}}$	$\boldsymbol{\Sigma}$	$E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\Sigma}}$	$E\|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Sigma}}$	$\boldsymbol{\Sigma}$
CTA	$\frac{1}{N}E\|\tilde{\boldsymbol{\varphi}}_t\|^2$	$\frac{1}{N}\|\underline{\mathbf{w}}_{\mathbf{o}}\|^2$	$\frac{1}{N}\begin{bmatrix}\mathbf{I}_{MN} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$	$\frac{1}{N}E\|\tilde{\boldsymbol{\varphi}}_t\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{u}}}$	$\frac{1}{N}\|\underline{\mathbf{w}}_{\mathbf{o}}\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{u}}}$	$\frac{1}{N}\begin{bmatrix}\boldsymbol{\Lambda}_{\boldsymbol{u}} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$
ATC	$\frac{1}{N}E\|\tilde{\mathbf{w}}_t\|^2$	$\frac{1}{N}\|\underline{\mathbf{w}}_{\mathbf{o}}\|^2$	$\frac{1}{N}\begin{bmatrix}\mathbf{G_D}^T\mathbf{G_D} & \mathbf{G_D}^T\mathbf{G_C}\\ \mathbf{G_C}^T\mathbf{G_D} & \mathbf{G_C}^T\mathbf{G_C}\end{bmatrix}$	$\frac{1}{N}E\|\tilde{\mathbf{w}}_t\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{u}}}$	$\frac{1}{N}\|\underline{\mathbf{w}}_{\mathbf{o}}\|^2_{\boldsymbol{\Lambda}_{\boldsymbol{u}}}$	$\frac{1}{N}\begin{bmatrix}\mathbf{G_D}^T\boldsymbol{\Lambda}_{\boldsymbol{u}}\mathbf{G_D} & \mathbf{G_D}^T\boldsymbol{\Lambda}_{\boldsymbol{u}}\mathbf{G_C}\\ \mathbf{G_C}^T\boldsymbol{\Lambda}_{\boldsymbol{u}}\mathbf{G_D} & \mathbf{G_C}^T\boldsymbol{\Lambda}_{\boldsymbol{u}}\mathbf{G_C}\end{bmatrix}$

Through (39) and (40), we can analyze the learning, convergence and stability behavior of the network. Iterating the weighted-energy recursion, we obtain

$$\begin{aligned}E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\sigma}} &= E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\mathbf{F}\boldsymbol{\sigma}} + \mathbf{b}^T\boldsymbol{\sigma}\\ E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\mathbf{F}\boldsymbol{\sigma}} &= E\|\tilde{\boldsymbol{\psi}}_{t-1}\|^2_{\mathbf{F}^2\boldsymbol{\sigma}} + \mathbf{b}^T\mathbf{F}\boldsymbol{\sigma}\\ &\vdots\\ E\|\tilde{\boldsymbol{\psi}}_1\|^2_{\mathbf{F}^t\boldsymbol{\sigma}} &= E\|\tilde{\boldsymbol{\psi}}_0\|^2_{\mathbf{F}^{t+1}\boldsymbol{\sigma}} + \mathbf{b}^T\mathbf{F}^t\boldsymbol{\sigma}.\end{aligned}$$

Assuming the parameter estimates $\boldsymbol{\varphi}_{i,t}$ and $\mathbf{a}_{i,t}$ are initialized with zeros, $E\|\tilde{\boldsymbol{\psi}}_0\|^2 = \|\underline{\underline{\mathbf{w}}}_{\mathbf{o}}\|^2$ where $\underline{\underline{\mathbf{w}}}_{\mathbf{o}} \triangleq \mathrm{col}\{\underline{\mathbf{w}}_{\mathbf{o}}, \underline{\mathbf{w}}_{\mathbf{o}}\}$. The iterations yield

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\sigma}} = \|\underline{\underline{\mathbf{w}}}_{\mathbf{o}}\|^2_{\mathbf{F}^{t+1}\boldsymbol{\sigma}} + \mathbf{b}^T\left(\sum_{k=0}^{t}\mathbf{F}^k\right)\boldsymbol{\sigma}. \tag{41}$$

By (41), we reach the following final recursion:

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\sigma}} = E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\sigma}} + \mathbf{b}^T\mathbf{F}^t\boldsymbol{\sigma} - \|\underline{\underline{\mathbf{w}}}_{\mathbf{o}}\|^2_{\mathbf{F}^t(\mathbf{I}-\mathbf{F})\boldsymbol{\sigma}}. \tag{42}$$

Remark 5.2: We note that (42) is of essence since through the weighting matrix $\boldsymbol{\Sigma}$ we can extract information about the learning and convergence behavior of the network. In Table II, we tabulate the initial conditions (we assume the initial parameter vectors are set to $\mathbf{0}$) and the weighting matrices corresponding to various conventional performance measures.

Remark 5.3: In this paper, (42) provides a recursion for the weighted deviation parameter where we assign $\boldsymbol{\varphi}_{i,t}$ as the final estimate instead of $\mathbf{w}_{i,t}$, which implies the CTA strategy, however, the recursion also provides the performance of the ATC strategy with appropriate combination matrix $\boldsymbol{\Sigma}$ and the initial condition (See Table II).

Next, we analyze the mean-square convergence performance of the single-bit diffusion approach for Gaussian regressors.

VI. SINGLE-BIT DIFFUSION WITH GAUSSIAN REGRESSORS

The weighted-energy relation of (26) yields

$$\begin{aligned}E\left[\tilde{\boldsymbol{\psi}}_{t+1}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{\psi}}_{t+1}\right] &= E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X}\tilde{\boldsymbol{\psi}}_t\right]\\ &\quad - E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\right]\\ &\quad - E\left[\underline{\mathbf{h}}^T(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{X}\tilde{\boldsymbol{\psi}}_t\right]\\ &\quad + E[\underline{\mathbf{h}}^T(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)].\end{aligned} \tag{43}$$

We evaluate RHS of (43) term by term in order to find the variance relation. We first partition the weighting matrix as follows:

$$\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_2\\ \boldsymbol{\Sigma}_3 & \boldsymbol{\Sigma}_4\end{bmatrix}. \tag{44}$$

Through the partitioning (44), we obtain

$$\begin{aligned}&E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\right]\\ &\quad= E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{u}}^T\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{U}_t^T\mathbf{Z}_{\mathbf{u}}\tilde{\boldsymbol{\psi}}_t\right]\\ &\qquad+ E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{u}}^T\boldsymbol{\Sigma}_2\mathbf{N}\mathbf{C}_t\mathrm{sign}\left(\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right)\right]\\ &\qquad+ E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{d}}^T\boldsymbol{\Sigma}_3\mathbf{M}\mathbf{U}_t\mathbf{U}_t^T\mathbf{Z}_{\mathbf{u}}\tilde{\boldsymbol{\psi}}_t\right]\\ &\qquad+ E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{d}}^T\boldsymbol{\Sigma}_4\mathbf{N}\mathbf{C}_t\mathrm{sign}\left(\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right)\right],\end{aligned} \tag{45}$$

where we partition $\mathbf{X}$ and $\mathbf{Z}$ such that $\mathbf{X} = \mathrm{col}\{\mathbf{X}_{\mathbf{u}}, \mathbf{X}_{\mathbf{d}}\}$ and $\mathbf{Z} = \mathrm{col}\{\mathbf{Z}_{\mathbf{u}}, \mathbf{Z}_{\mathbf{d}}\}$. We note that the second and fourth terms in the RHS of (45) include the nonlinear $\mathrm{sign}(\,\cdot\,)$ function. It is not straight-forward to evaluate the expectations with this nonlinearity, thus we introduce the following lemma.

Lemma 1: Under the Assumption 2, the Price's theorem [5] leads to

$$\begin{aligned}&E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{u}}^T\boldsymbol{\Sigma}_2\mathbf{N}\mathbf{C}_t\mathrm{sign}\left(\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right)\right]\\ &\quad= E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{u}}^T\boldsymbol{\Sigma}_2\mathbf{N}\boldsymbol{\Omega}_t\mathbf{C}_t\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right],\end{aligned} \tag{46}$$

$$\begin{aligned}&E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{d}}^T\boldsymbol{\Sigma}_4\mathbf{N}\mathbf{C}_t\mathrm{sign}\left(\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right)\right]\\ &\quad= E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}_{\mathbf{d}}^T\boldsymbol{\Sigma}_4\mathbf{N}\boldsymbol{\Omega}_t\mathbf{C}_t\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right],\end{aligned} \tag{47}$$

where $\boldsymbol{\Omega}_t$ is defined as

$$\boldsymbol{\Omega}_t \triangleq \begin{bmatrix}\frac{E|\epsilon_{1,t}|}{E\left[\epsilon_{1,t}^2\right]}\mathbf{I}_M & \cdots & \mathbf{0}_M\\ \vdots & \ddots & \vdots\\ \mathbf{0}_M & \cdots & \frac{E|\epsilon_{N,t}|}{E\left[\epsilon_{N,t}^2\right]}\mathbf{I}_M\end{bmatrix}.$$

Proof: The proof is given in Appendix A. □

By (45), (46), (47), the second term on the RHS of (43) is given by

$$E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}\right] = E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\underline{\boldsymbol{\Omega}}_t\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}\tilde{\boldsymbol{\psi}}_t\right], \tag{48}$$

where we drop the arguments of $\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)$ for notational simplicity and $\underline{\boldsymbol{\Omega}}_t$ denotes

$$\underline{\boldsymbol{\Omega}}_t \triangleq \begin{bmatrix}\mathbf{I}_{MN} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Omega}_t\end{bmatrix}.$$

Similarly, the third term on the RHS of (43) is evaluated as

$$E\left[\underline{\mathbf{h}}^T\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{X}\tilde{\boldsymbol{\psi}}_t\right] = E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{Z}^T\mathbf{Y}_t\mathbf{Y}_t^T\underline{\boldsymbol{\Omega}}_t\mathbf{D}\boldsymbol{\Sigma}\mathbf{X}\tilde{\boldsymbol{\psi}}_t\right]. \quad (49)$$

Through partitioning, the last term on the RHS of (43) yields

$$\begin{aligned}E&\left[\underline{\mathbf{h}}^T(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\right]\\ &= E\left[\mathbf{e}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{e}_t\right]\\ &\quad + E\left[\mathbf{e}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_2\mathbf{N}\mathbf{C}_t\mathrm{sign}(\boldsymbol{\epsilon}_t)\right]\\ &\quad + E\left[\mathrm{sign}(\boldsymbol{\epsilon}_t)^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_3\mathbf{M}\mathbf{U}_t\mathbf{e}_t\right]\\ &\quad + E\left[\mathrm{sign}(\boldsymbol{\epsilon}_t)^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_4\mathbf{N}\mathbf{C}_t\mathrm{sign}(\boldsymbol{\epsilon}_t)\right].\end{aligned}$$

Corollary 1: Since $\mathbf{U}_t$ and $\mathbf{C}_t$ are independent from each other, similar to the Lemma 1, we obtain

$$\begin{aligned}E&\left[\underline{\mathbf{h}}^T(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\right]\\ &= E\left[\mathbf{e}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{e}_t\right]\\ &\quad + E\left[\mathbf{e}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_2\mathbf{N}\boldsymbol{\Omega}_t\mathbf{C}_t\boldsymbol{\epsilon}_t\right]\\ &\quad + E\left[\boldsymbol{\epsilon}_t^T\mathbf{C}_t^T\boldsymbol{\Omega}_t\mathbf{N}\boldsymbol{\Sigma}_3\mathbf{M}\mathbf{U}_t\mathbf{e}_t\right]\\ &\quad + E\left[\mathrm{sign}(\boldsymbol{\epsilon}_t)^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_4\mathbf{N}\mathbf{C}_t\mathrm{sign}(\boldsymbol{\epsilon}_t)\right]. \quad (50)\end{aligned}$$

By the Assumption 1, the first term on the RHS of (50) yields

$$\begin{aligned}E\left[\mathbf{e}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{e}_t\right] &= E\left[\mathbf{v}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{v}_t\right]\\ &\quad + E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{Z}_{\mathbf{u}}^T\mathbf{U}_t\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{U}_t^T\mathbf{Z}_u\tilde{\boldsymbol{\psi}}_t\right]. \quad (51)\end{aligned}$$

For the last term on the RHS of (50), we introduce the following lemma.

Lemma 2: Through the Price's theorem, we obtain

$$\begin{aligned}&E\left[\mathrm{sign}(\boldsymbol{\epsilon}_t)^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_4\mathbf{N}\mathbf{C}_t\mathrm{sign}(\boldsymbol{\epsilon}_t)\right]\\ &\quad = E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{Z}_{\mathbf{d}}^T\mathbf{C}_t\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Omega}_t\boldsymbol{\Sigma}_4^{\boldsymbol{C}}\boldsymbol{\Omega}_t\mathbf{N}\mathbf{C}_t\mathbf{C}_t^T\mathbf{Z}_{\mathbf{d}}\tilde{\boldsymbol{\psi}}_t\right]\\ &\quad\quad + E\left[\mathbf{1}^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_4^{\boldsymbol{D}}\mathbf{N}\mathbf{C}_t\mathbf{1}\right], \quad (52)\end{aligned}$$

where $\boldsymbol{\Sigma}_4^{\boldsymbol{D}}$ is the block diagonal matrix of $\boldsymbol{\Sigma}_4$ such that

$$\boldsymbol{\Sigma}_4^{\boldsymbol{D}} = \begin{bmatrix}\boldsymbol{\Theta}_{11} & \cdots & \mathbf{0}_M\\ \vdots & \ddots & \vdots\\ \mathbf{0}_M & \cdots & \boldsymbol{\Theta}_{NN}\end{bmatrix}$$

with $\boldsymbol{\Theta}_{ii}$ is the ii'th $M\times M$ block of $\boldsymbol{\Sigma}_4$ and $\boldsymbol{\Sigma}_4^{\boldsymbol{C}} = \boldsymbol{\Sigma}_4 - \boldsymbol{\Sigma}_4^{\boldsymbol{D}}$.

Proof: The proof is given in Appendix B. □

As a result, by (48), (49), (50), (51) and (52), the relation (43) leads to

$$\begin{aligned}E\|\tilde{\boldsymbol{\psi}}_{t+1}\|_{\boldsymbol{\Sigma}}^2 &= E\|\tilde{\boldsymbol{\psi}}_t\|_{\boldsymbol{\Sigma}'}^2 + E\left[\mathbf{v}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{v}_t\right]\\ &\quad + E\left[\mathbf{1}^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_4^{\boldsymbol{D}}\mathbf{N}\mathbf{C}_t\mathbf{1}\right] \quad (53)\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{\Sigma}' &= \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X} - \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\underline{\boldsymbol{\Omega}}_t\mathbf{Y}_t\mathbf{Y}_t^T\mathbf{Z}\\ &\quad - \mathbf{Z}^T\mathbf{Y}_t\mathbf{Y}_t^T\underline{\boldsymbol{\Omega}}_t\mathbf{D}\boldsymbol{\Sigma}\mathbf{X}\\ &\quad + \mathbf{Z}^T\mathbf{D}\underline{\boldsymbol{\Omega}}_t\mathbf{Y}_t\mathbf{Y}_t^T\tilde{\boldsymbol{\Sigma}}\mathbf{Y}_t\mathbf{Y}_t^T\underline{\boldsymbol{\Omega}}_t\mathbf{D}\mathbf{Z},\end{aligned}$$

where $\tilde{\boldsymbol{\Sigma}}$ denotes

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_2\\ \boldsymbol{\Sigma}_3 & \boldsymbol{\Sigma}_4^{\boldsymbol{C}}\end{bmatrix}.$$

We again note that by the Assumption 1, we get $\boldsymbol{\Sigma}' = E[\boldsymbol{\Sigma}']$ which results

$$\begin{aligned}\boldsymbol{\Sigma}' &= \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X} - \mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\underline{\boldsymbol{\Omega}}_t\boldsymbol{\Lambda}\mathbf{Z} - \mathbf{Z}^T\boldsymbol{\Lambda}\underline{\boldsymbol{\Omega}}_t\mathbf{D}\boldsymbol{\Sigma}\mathbf{X}\\ &\quad + \mathbf{Z}^T\mathbf{D}\underline{\boldsymbol{\Omega}}_t E\left[\mathbf{Y}_t\mathbf{Y}_t^T\tilde{\boldsymbol{\Sigma}}\mathbf{Y}_t\mathbf{Y}_t^T\right]\underline{\boldsymbol{\Omega}}_t\mathbf{D}\mathbf{Z} \quad (54)\end{aligned}$$

and define $\mathbf{B} \triangleq E[\mathbf{Y}_t\mathbf{Y}_t^T\tilde{\boldsymbol{\Sigma}}\mathbf{Y}_t\mathbf{Y}_t^T]$.

In the following, we resort to the vector notation, i.e., the block vectorisation operator $\mathrm{bvec}\{\cdot\}$ and the block Kronecker product. Hence, the block vectorization of the weighting matrix $\boldsymbol{\Sigma}'$ (54) yields

$$\begin{aligned}\mathrm{bvec}\{\boldsymbol{\Sigma}'\} &= (\mathbf{X}^T\odot\mathbf{X}^T - (\mathbf{X}^T\odot\mathbf{Z}^T)(\mathbf{I}_{2MN}\odot\boldsymbol{\Lambda}\mathbf{D}\underline{\boldsymbol{\Omega}}_t)\\ &\quad - (\mathbf{Z}^T\odot\mathbf{X}^T)(\boldsymbol{\Lambda}\mathbf{D}\underline{\boldsymbol{\Omega}}_t\odot\mathbf{I}_{2MN}))\boldsymbol{\sigma}\\ &\quad + (\mathbf{Z}^T\odot\mathbf{Z}^T)(\mathbf{D}\odot\mathbf{D})(\underline{\boldsymbol{\Omega}}_t\odot\underline{\boldsymbol{\Omega}}_t)\mathrm{bvec}\{\mathbf{B}\}. \quad (55)\end{aligned}$$

Block vectorisation of the matrix $\mathbf{B}$ is given by $\mathrm{bvec}\{\mathbf{B}\} = \mathcal{A}\,\mathrm{bvec}\{\tilde{\boldsymbol{\Sigma}}\}$. In order to denote $\mathrm{bvec}\{\tilde{\boldsymbol{\Sigma}}\}$ in terms of $\boldsymbol{\sigma}$, we introduce $\mathbf{K}_1 \triangleq \mathrm{col}\{\mathbf{0}_{MN},\mathbf{I}_{MN}\}$, $\mathbf{K}_2 \triangleq \mathrm{col}\{\mathbf{I}_{MN},\mathbf{0}_{MN}\}$, and $\mathbf{T}_{\mathbf{k}} \triangleq \mathrm{diag}\{\mathbf{0}_{(k-1)M},\mathbf{I}_M,\mathbf{0}_{(N-k)M}\}$. Then, we get

$$\boldsymbol{\Sigma}_4^{\boldsymbol{D}} = \sum_{k=1}^{N}\mathbf{T}_{\mathbf{k}}\mathbf{K}_2^T\boldsymbol{\Sigma}\mathbf{K}_2\mathbf{T}_{\mathbf{k}}, \quad (56)$$

$$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma} - \mathbf{K}_2\boldsymbol{\Sigma}_4^{\boldsymbol{D}}\mathbf{K}_2^T. \quad (57)$$

By (56) and (57), we obtain

$$\begin{aligned}\mathrm{bvec}\{\tilde{\boldsymbol{\Sigma}}\} &= \underbrace{\left(\mathbf{I} - (\mathbf{K}_2\odot\mathbf{K}_2)\sum_{k=1}^{N}(\mathbf{T}_{\mathbf{k}}\odot\mathbf{T}_{\mathbf{k}})(\mathbf{K}_{\mathbf{2}}^T\odot\mathbf{K}_2^T)\right)}_{\mathbf{K}}\boldsymbol{\sigma}\\ &= \mathbf{K}\boldsymbol{\sigma}. \quad (58)\end{aligned}$$

The $\tilde{\boldsymbol{\psi}}$-free terms in (53) are evaluated as

$$E\left[\mathbf{v}_t^T\mathbf{U}_t^T\mathbf{M}\boldsymbol{\Sigma}_1\mathbf{M}\mathbf{U}_t\mathbf{v}_t\right] = \mathbf{b}_1^T(\mathbf{K}_1^T\odot\mathbf{K}_1^T)\boldsymbol{\sigma}, \quad (59)$$

$$E\left[\mathbf{1}^T\mathbf{C}_t^T\mathbf{N}\boldsymbol{\Sigma}_4^{\boldsymbol{D}}\mathbf{N}\mathbf{C}_t\mathbf{1}\right] = \mathbf{b}_2^T(\mathbf{K}_2^T\odot\mathbf{K}_2^T)\boldsymbol{\sigma}, \quad (60)$$

where $\mathbf{b}_1 \triangleq \mathrm{bvec}\{\mathbf{R}_{\mathbf{v}}\mathbf{M}^2\boldsymbol{\Lambda}_{\boldsymbol{u}}\}$ and $\mathbf{b}_2 \triangleq \mathrm{bvec}\{\mathbf{1}\mathbf{1}^T\mathbf{N}^2\boldsymbol{\Lambda}_{\boldsymbol{c}}\}$.

As a result, by (55), (58), (59) and (60), the weighted-energy relation is given by

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|_{\boldsymbol{\sigma}}^2 = E\|\tilde{\boldsymbol{\psi}}_t\|_{\mathbf{F}_t\boldsymbol{\sigma}}^2 + \mathbf{b}^T\boldsymbol{\sigma} \quad (61)$$

$$\begin{aligned}\mathbf{F}_t &= \mathbf{X}^T\odot\mathbf{X}^T - (\mathbf{X}^T\odot\mathbf{Z}^T)(\mathbf{I}_{2MN}\odot\boldsymbol{\Lambda}\mathbf{D}\underline{\boldsymbol{\Omega}}_t)\\ &\quad - (\mathbf{Z}^T\odot\mathbf{X}^T)(\boldsymbol{\Lambda}\mathbf{D}\underline{\boldsymbol{\Omega}}_t\odot\mathbf{I}_{2MN})\\ &\quad + (\mathbf{Z}^T\odot\mathbf{Z}^T)(\mathbf{D}\odot\mathbf{D})(\underline{\boldsymbol{\Omega}}_t\odot\underline{\boldsymbol{\Omega}}_t)\mathcal{A}\mathbf{K} \quad (62)\end{aligned}$$

$$\mathbf{b} = \left(\mathbf{K}_1^T\odot\mathbf{K}_1^T\right)^T\mathbf{b}_1 + \left(\mathbf{K}_2^T\odot\mathbf{K}_2^T\right)^T\mathbf{b}_2. \quad (63)$$

Iterating the weighted-energy recursion (61), (62) and (63), we obtain

$$\begin{aligned}E\|\tilde{\boldsymbol{\psi}}_{t+1}\|_{\boldsymbol{\sigma}}^2 &= E\|\tilde{\boldsymbol{\psi}}_t\|_{\mathbf{F}_t\boldsymbol{\sigma}}^2 + \mathbf{b}^T\boldsymbol{\sigma}\\ E\|\tilde{\boldsymbol{\psi}}_t\|_{\mathbf{F}_t\boldsymbol{\sigma}}^2 &= E\|\tilde{\boldsymbol{\psi}}_{t-1}\|_{\mathbf{F}_{t-1}\mathbf{F}_t\boldsymbol{\sigma}}^2 + \mathbf{b}^T\mathbf{F}_t\boldsymbol{\sigma}\\ &\vdots\\ E\|\tilde{\boldsymbol{\psi}}_1\|_{\mathbf{F}_1\ldots\mathbf{F}_t\boldsymbol{\sigma}}^2 &= E\|\tilde{\boldsymbol{\psi}}_0\|_{\mathbf{F}_0\ldots\mathbf{F}_t\boldsymbol{\sigma}}^2 + \mathbf{b}^T\mathbf{F}_1\ldots\mathbf{F}_t\boldsymbol{\sigma}.\end{aligned}$$

TABLE III
INITIAL CONDITIONS AND WEIGHTING MATRICES FOR THE PERFORMANCE MEASURE OF THE CONSTRUCTION UPDATE FOR THE SINGLE-BIT DIFFUSION APPROACH (FOR THE SCALAR DIFFUSION APPROACH, SET $\zeta = 0$) AND THE GLOBAL MSD OF THE ATC DIFFUSION STRATEGY FOR THE SINGLE-BIT DIFFUSION APPROACH (FOR THE SCALAR DIFFUSION APPROACH, SEE TABLE II)

$E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\Sigma}}$	$E\|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Sigma}}$	$\boldsymbol{\Sigma}$
$\frac{1}{N}E\|\tilde{\mathbf{a}}_t\|^2$	$\frac{1}{N}\|\underline{\mathbf{w}}_{\mathbf{o}} - \zeta\mathbf{1}\|^2$	$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{N}\mathbf{I}_{MN} \end{bmatrix}$
$\sigma^2_{\boldsymbol{\epsilon}_t} = E[\boldsymbol{\epsilon}_t^T\boldsymbol{\epsilon}_t]$	$\zeta\mathbf{1}^T\boldsymbol{\Lambda}_{\boldsymbol{c}}\mathbf{1}$	$\begin{bmatrix} \boldsymbol{\Lambda}_{\boldsymbol{c}} & -\boldsymbol{\Lambda}_{\boldsymbol{c}} \\ -\boldsymbol{\Lambda}_{\boldsymbol{c}} & \boldsymbol{\Lambda}_{\boldsymbol{c}} \end{bmatrix}$
$\frac{1}{N}E\|\tilde{\mathbf{w}}_t\|^2$	$\frac{1}{N}\|\underline{\mathbf{w}}_{\mathbf{o}} - \zeta\mathbf{G}_{\mathbf{C}}\mathbf{1}\|^2$	$\frac{1}{N}\begin{bmatrix} \mathbf{G_D}^T\mathbf{G_D} & \mathbf{G_D}^T\mathbf{G_C} \\ \mathbf{G_C}^T\mathbf{G_D} & \mathbf{G_C}^T\mathbf{G_C} \end{bmatrix}$

In this part of the analyzes, we do not assume that the parameter vectors are initialized with zeros since such an assumption results in infinite terms in the $\boldsymbol{\Omega}_t$ matrix. Hence, we initialize $\mathbf{a}_t$ with $\zeta\,\mathbf{1}_{MN\times 1}$ where ζ has a small value (See Table III).

The iterations yield

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\sigma}} = \|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Pi}_t\boldsymbol{\sigma}} + \mathbf{b}^T\boldsymbol{\Delta}_t\boldsymbol{\sigma}, \tag{64}$$

$$E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\sigma}} = \|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Pi}_{t-1}\boldsymbol{\sigma}} + \mathbf{b}^T\boldsymbol{\Delta}_{t-1}\boldsymbol{\sigma}, \tag{65}$$

where $\boldsymbol{\Pi}_t \triangleq \prod_{i=0}^{t}\mathbf{F}_i$ and $\boldsymbol{\Delta}_t \triangleq \mathbf{I} + \mathbf{F}_t + \mathbf{F}_{t-1}\mathbf{F}_t + \cdots + \mathbf{F}_1\ldots\mathbf{F}_t$. We note that $\boldsymbol{\Pi}_t = \boldsymbol{\Pi}_{t-1}\mathbf{F}_t$ and $\boldsymbol{\Delta}_t = \boldsymbol{\Delta}_{t-1}\mathbf{F}_t + \mathbf{I}$. By (64) and (65), we have the following recursion

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\sigma}} = E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\sigma}} - \|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Pi}_{t-1}(\mathbf{I}-\mathbf{F}_t)\boldsymbol{\sigma}} + \mathbf{b}^T(\mathbf{I} - \boldsymbol{\Delta}_{t-1}(\mathbf{I} - \mathbf{F}_t))\boldsymbol{\sigma}. \tag{66}$$

We point out that $\boldsymbol{\Pi}_{-1} = \mathbf{I}_{(2MN)^2}$ and $\boldsymbol{\Delta}_{-1} = \mathbf{0}_{(2MN)^2}$.

Remark 6.1: The iterations of (66) require the recalculation of $\mathbf{F}_t$ for each time instants since $\mathbf{F}_t$ changes with time because of $\underline{\boldsymbol{\Omega}}_t$ (62). Evaluating the expectations, $\boldsymbol{\Omega}_t$ yields

$$\boldsymbol{\Omega}_t = \sqrt{\frac{2}{\pi}}\begin{bmatrix} \frac{1}{\sigma_{\epsilon_1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_{\epsilon_N}} \end{bmatrix} \otimes \mathbf{I}_M, \tag{67}$$

where $\sigma^2_{\epsilon_i} = E[\epsilon_i^2]$. For analytical reasons, we approximate (67) as

$$\boldsymbol{\Omega}_t \approx \sqrt{\frac{2}{\pi}}\frac{1}{(1/\sqrt{N})\sigma_{\boldsymbol{\epsilon}_t}}\mathbf{I}_{MN} \tag{68}$$

with $\sigma^2_{\boldsymbol{\epsilon}_t} = E[\boldsymbol{\epsilon}_t^T\boldsymbol{\epsilon}_t] = E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\xi}}$ and

$$\boldsymbol{\xi} \triangleq \mathrm{bvec}\left\{\begin{bmatrix} \boldsymbol{\Lambda}_{\boldsymbol{c}} & -\boldsymbol{\Lambda}_{\boldsymbol{c}} \\ -\boldsymbol{\Lambda}_{\boldsymbol{c}} & \boldsymbol{\Lambda}_{\boldsymbol{c}} \end{bmatrix}\right\}.$$

Hence, we can calculate $\mathbf{F}_t$ by iterating the following

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\xi}} = E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\boldsymbol{\xi}} - \|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\Pi}_{t-1}(\mathbf{I}-\mathbf{F}_t)\boldsymbol{\xi}} + \mathbf{b}^T(\mathbf{I} - \boldsymbol{\Delta}_{t-1}(\mathbf{I} - \mathbf{F}_t))\boldsymbol{\xi}, \tag{69}$$

where $E\|\tilde{\boldsymbol{\psi}}_0\|^2_{\boldsymbol{\xi}} = \zeta\mathbf{1}^T\boldsymbol{\Lambda}_{\boldsymbol{c}}\mathbf{1}$. In Table III, we tabulate the initial condition and the weighting matrix necessary for the recursion iterations (69) of $\sigma^2_{\boldsymbol{\epsilon}_t} = E[\boldsymbol{\epsilon}_t^T\boldsymbol{\epsilon}_t]$.

VII. STEADY-STATE ANALYSIS

At the steady-state, (39) yields

$$E\|\tilde{\boldsymbol{\psi}}_{\infty}\|^2_{(\mathbf{I}-\mathbf{F})\boldsymbol{\sigma}} = \mathbf{b}^T\boldsymbol{\sigma}.$$

In order to calculate the steady-state performance measure $E\|\tilde{\boldsymbol{\psi}}_{\infty}\|^2_{\boldsymbol{\sigma}'}$, we choose the weighting matrix as $\boldsymbol{\sigma}' = (\mathbf{I} - \mathbf{F})\boldsymbol{\sigma}$, then the steady-state performance measure is given by

$$E\|\tilde{\boldsymbol{\psi}}_{\infty}\|^2_{\boldsymbol{\sigma}'} = \mathbf{b}^T(\mathbf{I} - \mathbf{F})^{-1}\boldsymbol{\sigma}'. \tag{70}$$

Similar to (70), the steady state mean square error $E[\boldsymbol{\epsilon}_t^T\boldsymbol{\epsilon}_t]$ for the single bit diffusion strategy is given by

$$E\|\tilde{\boldsymbol{\psi}}_{\infty}\|^2_{\boldsymbol{\xi}} = \mathbf{b}^T(\mathbf{I} - \mathbf{F}_{\infty})^{-1}\boldsymbol{\xi}. \tag{71}$$

We point out that $\mathbf{F}_{\infty}$ depends on $E\|\tilde{\boldsymbol{\psi}}_{\infty}\|^2_{\boldsymbol{\xi}}$. Once we calculate $\mathbf{F}_{\infty}$ numerically by (71) or through approximations, we can obtain the steady state performance by (70).

VIII. TRACKING PERFORMANCE

The diffusion implementation improves the ability of the network to track variations in the underlying statistical profiles [6]. In this section, we analyze the tracking performance of the compressive diffusion strategies in a non-stationary environment. We assume a first-order random walk model, which is commonly used in the literature [5], for $\mathbf{w}_{\mathbf{o},t}$ such that

$$\mathbf{w}_{\mathbf{o},t+1} = \mathbf{w}_{\mathbf{o},t} + \mathbf{q}_t,$$

where $\mathbf{q}_t \in \mathbb{R}^M$ denotes a zero-mean vector process independent of the regression data and observation noise with covariance matrix $E[\mathbf{q}_t\mathbf{q}_t^T] = \mathbf{Q}$. We introduce the global time-variant parameter vectors as $\underline{\mathbf{w}}_{\mathbf{o},t} \triangleq \mathrm{col}\{\mathbf{w}_{\mathbf{o},t},\ldots,\mathbf{w}_{\mathbf{o},t}\}$ and we have the global deviation vectors as $\tilde{\boldsymbol{\varphi}}_t \triangleq \underline{\mathbf{w}}_{\mathbf{o},t} - \boldsymbol{\varphi}_t$ and $\tilde{\mathbf{a}}_t \triangleq \underline{\mathbf{w}}_{\mathbf{o},t} - \mathbf{a}_t$. Then, by (26), we obtain

$$\tilde{\boldsymbol{\psi}}_{t+1} = \mathbf{X}\tilde{\boldsymbol{\psi}}_t - \mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t) + \underline{\underline{\mathbf{q}_t}}, \tag{72}$$

where $\underline{\underline{\mathbf{q}_t}} \triangleq \mathrm{col}\{\mathbf{q}_t,\ldots,\mathbf{q}_t\}$ with $2MN\times 1$ dimensions. Since we assume that $\mathbf{q}_t$ is independent from the regression data $\mathbf{u}_{i,t}$, $\mathbf{c}_{i,t}$ and the observation noise $v_{i,t}$ for all $i \in \{1,\ldots,N\}$, (72) yields the following weighted-energy relation

$$\begin{aligned} E\left[\tilde{\boldsymbol{\psi}}_{t+1}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{\psi}}_{t+1}\right] &= E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X}\tilde{\boldsymbol{\psi}}_t\right] \\ &\quad - E\left[\tilde{\boldsymbol{\psi}}_t^T\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\right] \\ &\quad - E\left[\underline{\mathbf{h}}^T(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{X}\tilde{\boldsymbol{\psi}}_t\right] \\ &\quad + E\left[\underline{\mathbf{h}}^T(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\mathbf{Y}_t^T\mathbf{D}\boldsymbol{\Sigma}\mathbf{D}\mathbf{Y}_t\underline{\mathbf{h}}(\mathbf{e}_t,\boldsymbol{\epsilon}_t)\right] \\ &\quad + E\left[\underline{\underline{\mathbf{q}_t}}^T\boldsymbol{\Sigma}\underline{\underline{\mathbf{q}_t}}\right]. \end{aligned} \tag{73}$$

We note that (73) is similar to (43) except for the last term $E[\underline{\underline{\mathbf{q}_t}}^T \boldsymbol{\Sigma} \underline{\underline{\mathbf{q}_t}}]$. We denote $2N \times 2N$ matrix whose terms are 1 as $\underline{\underline{\mathbf{1}}}_{2N} \triangleq [\mathbf{1}, \ldots, \mathbf{1}]$. Then, the last term in (73) is given by $\boldsymbol{\rho}^T \boldsymbol{\sigma}$ where $\boldsymbol{\rho} = \mathrm{bvec}\{\underline{\underline{\mathbf{1}}}_{2N} \otimes \mathbf{Q}\}$. Through (73), we get

$$E\|\tilde{\boldsymbol{\psi}}_{t+1}\|^2_{\boldsymbol{\sigma}} = E\|\tilde{\boldsymbol{\psi}}_t\|^2_{\mathbf{F}_t \boldsymbol{\sigma}} + \mathbf{b}^T \boldsymbol{\sigma} + \boldsymbol{\rho}^T \boldsymbol{\sigma}. \tag{74}$$

We define $\mathbf{F}_t$ in (40) and (62) for scalar and single-bit diffusion strategies, respectively. Similarly, $\mathbf{b}$ is introduced in (38) and (63) for the scalar (time-invariant) and single-bit diffusion strategies. We point out that (74) is different from (39) and (61) only for the term $\boldsymbol{\rho}^T \boldsymbol{\sigma}$. As a result, at steady state, (70) and (74) leads

$$E\|\tilde{\boldsymbol{\psi}}_{\infty}\|^2_{\boldsymbol{\sigma}} = (\mathbf{b} + \boldsymbol{\rho})^T (\mathbf{I} - \mathbf{F}_{\infty})^{-1} \boldsymbol{\sigma}. \tag{75}$$

Through (75) and Table II, we can obtain the tracking performance of the network for the conventional performance measures. We point out that in the full diffusion configuration, $\boldsymbol{\rho} = \mathrm{bvec}\{\underline{\underline{\mathbf{1}}}_{N} \otimes \mathbf{Q}\}$.

In the next section, we introduce the confidence parameter and the adaptive combination method, which provides a better trade-off in terms of the transient and the steady-state performances.

IX. Confidence Parameter and Adaptive Combination

The cooperation among the nodes is not beneficial in general unless the cooperation rule is chosen properly [1]. For example, the uniform [22], the Metropolis [23], the relative-degree rules [8] and the adaptive combiners [25] provide improved convergence performance relative to the no-cooperation configuration in which nodes aim to estimate the parameter of interest $\mathbf{w_o}$ without information exchange. However, the compressive diffusion strategies have a different diffusion protocol than the full diffusion configuration. At each node i, we combine the local estimates $\boldsymbol{\varphi}_{i,t}$ with the constructed estimates $\mathbf{a}_{j,t}$ that track the local estimates $\boldsymbol{\varphi}_{j,t}$ of the neighboring nodes, i.e., $j \in \mathcal{N}_i \backslash i$. Especially at the early stages of the adaptation, the constructed estimates carry far less information than the local estimates since they are not sufficiently close to the original estimates in the mean square sense. Hence, we can consider the constructed estimates as noisy version of the original parameter vectors. Then the overall network operation is akin to the full diffusion scheme with noisy observation. In [11], [33], [34], the authors demonstrate that for imperfect cooperation cases a node should place more weight on the local estimate in the combination step even if the node has worse quality of measurement than its neighbors. To this end, we add one more freedom of dimension to the update by introducing a confidence parameter δ. The confidence parameter determines the weight of the local estimates relative to the constructed estimates such that the new combination matrix $\boldsymbol{\Gamma}'$ is given by

$$\boldsymbol{\Gamma}' = \delta \mathbf{I}_N + (1 - \delta) \boldsymbol{\Gamma} \tag{76}$$

where $0 \leq \delta \leq 1$. We note that $\delta = 1$, in which case we are confident with the local estimates, yields the no-cooperation scheme and $\delta = 0$ is the full diffusion configuration where we thrust the diffused information totally.

For the new combination matrix (76), the combination of the local estimate and the constructed estimates (12) yields

$$\mathbf{w}_{i,t+1} = (1 - \delta) \underbrace{\left[\gamma_{i,i} \boldsymbol{\varphi}_{i,t+1} + \sum_{j \in \mathcal{N}_i \backslash i} \gamma_{i,j} \mathbf{a}_{j,t+1} \right]}_{\hat{\boldsymbol{\varphi}}_{i,t+1}} + \delta \boldsymbol{\varphi}_{i,t+1}. \tag{77}$$

We note that (77) is a convex combination of the parameter vectors $\hat{\boldsymbol{\varphi}}_{i,t+1}$ and $\boldsymbol{\varphi}_{i,t+1}$. Hence, we can adapt the convex combination weight δ using a stochastic gradient update [35]–[38]. Then, (77) yields

$$\mathbf{w}_{i,t+1} = \delta_{i,t+1} \boldsymbol{\varphi}_{i,t+1} + (1 - \delta_{i,t+1}) \hat{\boldsymbol{\varphi}}_{i,t+1}. \tag{78}$$

In [36], authors update all combination weights $\gamma_{i,j}$'s indirectly through a sigmoidal function. Similarly, we re-parameterize the confidence parameter $\delta_{i,t}$ using the sigmoidal function [39] and an unconstrained variable $\alpha_{i,t}$ such that

$$\delta_{i,t} = \frac{1}{1 + e^{-\alpha_{i,t}}}. \tag{79}$$

We train the unconstrained weight $\alpha_{i,t}$ using a stochastic gradient update minimizing $e^2_{i,t} = (d_{i,t} - \mathbf{u}^T_{i,t} \mathbf{w}_{i,t})^2$ as follows

$$\begin{aligned} \alpha_{i,t+1} &= \alpha_{i,t} - \frac{1}{2} \mu_{\mathrm{cvx}} \frac{\partial e^2_{i,t}}{\partial \alpha_{i,t}} \\ &= \alpha_{i,t} + \mu_{\mathrm{cvx}} e_{i,t} \mathbf{u}^T_{i,t} (\boldsymbol{\varphi}_{i,t} - \hat{\boldsymbol{\varphi}}_{i,t}) \delta_{i,t} (1 - \delta_{i,t}). \end{aligned} \tag{80}$$

As a result, we combine the local and constructed estimates via (78), (79) and (80).

In the next section, we provide numerical examples showing the match of the theoretical derivations and simulated results, and the improved convergence performance with the adaptive confidence parameter.

X. Numerical Examples

In this section, we examine two distinct network scenarios where we demonstrate that the theoretical analysis accurately model the simulated results and the confidence parameter provides significantly improved convergence performance. In the first example, we have a network of $N = 5$ nodes where at each node i, we observe a stationary data $d_{i,t} = \mathbf{u}^T_{i,t} \mathbf{w_o} + v_{i,t}$ for $i \in \{1, 2, \ldots, N\}$. The regression data $\mathbf{u}_{i,t}$ is a zero-mean i.i.d. Gaussian with randomly chosen standard deviation σ_{u_i}, i.e., $\sigma_{u_i} = 0.1(\sqrt{10} - 1) b_i + 0.1$ where $b_i \sim U[0, 1]$ is a uniform random variable. The variance of the observation noise is $\sigma^2_{v,i} = 10^{-3}$. Hence, the signal-to-noise ratio over the network varies between 10 to 100. The standard deviation of the projection operator is $\sigma_{c_i} = 1$. The parameter of interest $\mathbf{w_o} \in \mathbb{R}^4$ is randomly chosen. Note that we examine a relatively small network with a short filter length since the computational complexity of the theoretical performance relations (42) and (66) increases exponentially with the filter length M and the network size N. We point out that the overall communication burden $(N \times M)$ in the scalar diffusion strategy is 25% of the full diffusion configuration and the overall communication load in the single-bit diffusion strategy is given by 5-bits per iterations.

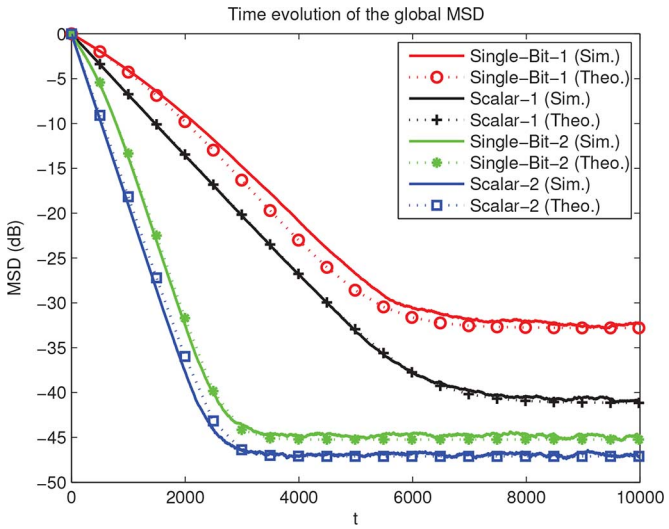

Fig. 4. Comparison of global MSD curves $1/NE\|\tilde{\boldsymbol{\varphi}}_t\|^2$ where the single-bit-1 and the scalar-1 schemes use $\delta = 0$ while the single-bit-2 and the scalar-2 schemes have $\delta = 0.9$.

In the no-cooperation configuration, the combination matrix is given by $\boldsymbol{\Gamma}_0 = \mathbf{I}_N$. We use the Metropolis combination rule [23] for the full diffusion configuration where the adjacency matrix of the network is given by

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

In the Metropolis rule [24], the combination weights are chosen according to

$$\gamma_{i,j} = \begin{cases} \frac{1}{\max\{n_i, n_j\}} & \text{if } j \in \mathcal{N}_i \setminus i, \\ 0 & \text{if } j \notin \mathcal{N}_i, \\ 1 - \sum_{j \in \mathcal{N}_i \setminus i} \gamma_{i,j} & \text{if i = j,} \end{cases}$$

where n_i and n_j denote the number of neighboring nodes for i and j. For the single-bit and the one-dimension diffusion strategies we examine the convergence performance for the confidence parameter $\delta = 0$ and $\delta = 0.9$ in Fig. 4. We choose the step sizes the same for the distributed LMS update (17) of all configurations at all nodes, i.e., $\mu_i = 0.042$. At each node, the step sizes for the construction update (18) are $\eta_i = 0.0015$ (for single-bit approach) and $\eta_i = 0.25$ (for one-dimension diffusion approach). For the single-bit diffusion approach, we set $\zeta = 0.001$ to initialize $\mathbf{a}_{j,t}$. In Fig. 4, we show the global MSD curves, i.e., $E\|\tilde{\boldsymbol{\varphi}}_t\|^2$, of the single-bit and scalar diffusion approaches and compare the performance for different δ values. The confidence parameter $\delta = 0.9$ implies that we give ten times more weight to the local estimate $\boldsymbol{\varphi}_{i,t}$ than the constructed estimates $\mathbf{a}_{j,t}$, where $j \in \mathcal{N}_i \setminus i$. The Fig. 4 demonstrates that the confidence parameter $\delta = 0.9$ improves the convergence performance of the compressive diffusion strategies.

In the same example, Figs. 5 and 6 compare the convergence performance of the single-bit and the scalar diffusion strategies with the no-cooperation and full diffusion configurations for $\delta = 0.9$, which shows the match of the theoretical and ensemble averaged performance results (we perform 200 independent trials). The Fig. 5 shows the time-evolution of the MSD and EMSE curves in the CTA diffusion strategy while the Fig. 6 displays the time-evolution of the MSD curves in the ATC diffusion strategy in which the theoretical curves (42) and (66) are iterated according to the Tables II and III. We note that we obtain similar MSD curves in the CTA and ATC strategies since we set $\delta = 0.9$ and the outcomes of the adaptation and combination operations contain relatively close amount of information.

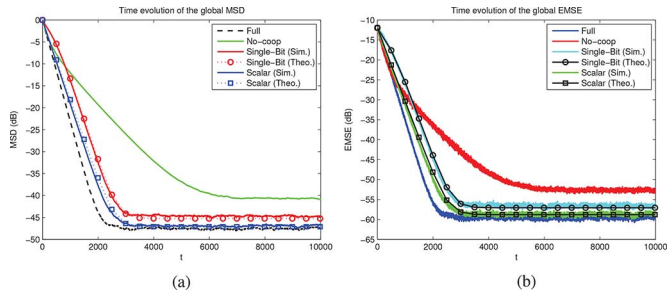

Fig. 5. Comparison of the global MSD and EMSE curves in the CTA strategy. (a) Global MSD curves. (b) Global EMSE curves.

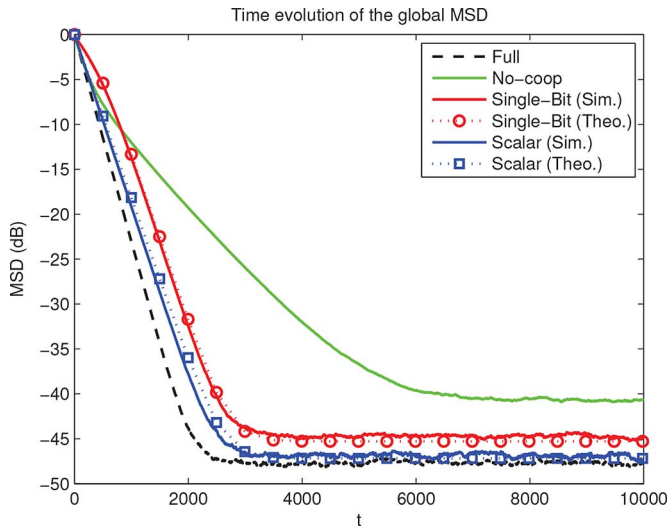

Fig. 6. Comparison of the global MSD curves in the ATC diffusion strategy.

In Fig. 7, we demonstrate the convergence of the constructed estimates $\mathbf{a}_{j,t}$'s to the parameter of interest $\mathbf{w_o}$ in the mean-square sense. We point out that the recursions (42) and (66) also provide the global mean-square deviation of the constructed estimates for the certain combination weight $\boldsymbol{\Sigma}$ in Table II and the theoretical recursion matches with the simulated results.

In Fig. 8, we examine the impact of the synchronization issues on the estimation performance in several different scenarios. As an example, we utilize pilot signals for the re-synchronization at every 10 or 100 samples. In the asynchronous events we assume that the diffused information is completely lost and each neighboring node loses the synchronization of the projection operator until the arrival of the next pilot signal, i.e., a severe synchronization event. We point out that the single-bit diffusion strategy requires the synchronization of the construction updates in addition to the synchronization of the projection operator. Hence, the pilot signals in the single-bit diffusion

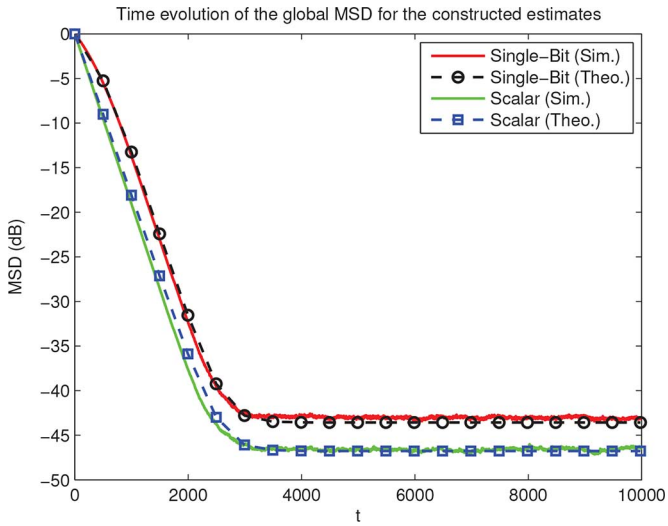

Fig. 7. The MSD curves of the construction estimate $1/NE\|\tilde{\mathbf{a}}_t\|^2$ of the single-bit and scalar diffusion approaches.

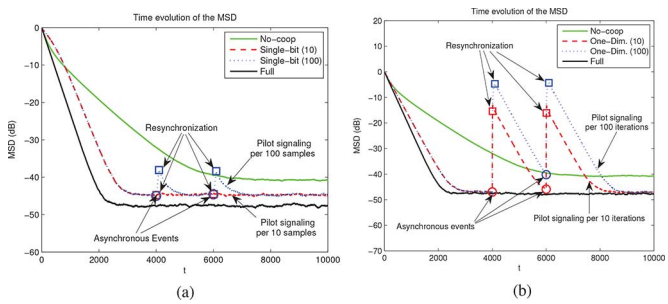

Fig. 8. Impact of asynchronous events on the learning curves. (a) Single-bit diffusion strategy. (b) Scalar diffusion strategy.

scheme also re-synchronize the construction updates in each node within the neighborhood. In the Fig. 8, we observe 2 asynchronous events at 5001st and 6001st time instants, however, through the pilot signals at 5100th and 6100th (pilot signaling per 100 iterations) or at 5010th and 6010th (pilot signaling per 10 iterations) time instants each node can re-synchronize again. We note that single-bit diffusion strategy has performed less sensitive to the asynchronous events thanks to the relatively small learning rate of the construction update.

Fig. 9 shows the time evolution of the global MSD of the proposed schemes, i.e., both ensemble averaged and theoretical results, in a non-stationary environment. We consider a first order random walk model and choose $E[\mathbf{q}_t\mathbf{q}_t^T] = 10^{-5}\mathbf{I}_M$ in the same configuration of the first example. In the Fig. 9, we observe the match of the ensemble averaged and the theoretical results.

In Fig. 10, we compare the time evolution of the proposed schemes with the partial diffusion strategy, where each node diffuses only one coefficient of the parameter vector. For the projection operator, we utilize a sequential selection scheme based on the round robin fashion such that

$$\mathbf{c}_t = \begin{cases} \text{col}\{0,0,1,1\} & \text{if } t \equiv 0 \bmod(4), \\ \text{col}\{1,-1,0,0\} & \text{if } t \equiv 1 \bmod(4), \\ \text{col}\{1,1,0,0\} & \text{if } t \equiv 2 \bmod(4), \\ \text{col}\{0,0,-1,1\} & \text{if } t \equiv 3 \bmod(4). \end{cases}$$

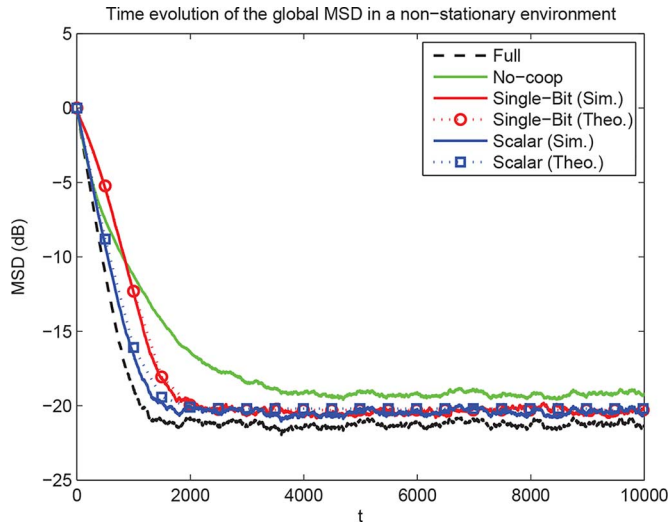

Fig. 9. Tracking performance of the proposed schemes in a non-stationary environment.

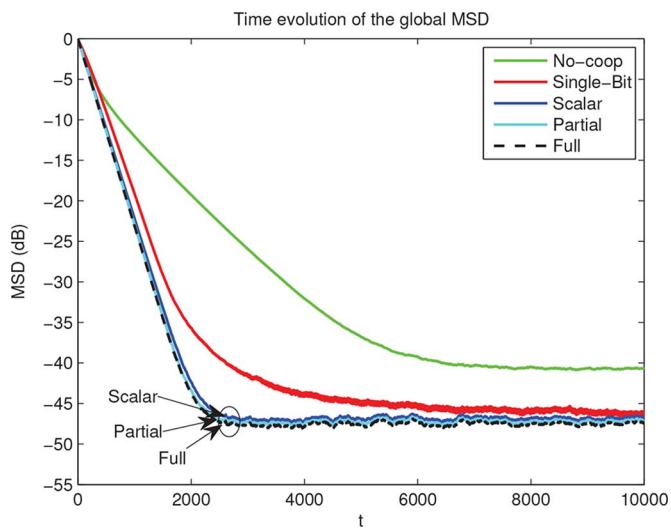

Fig. 10. Comparison of the global MSD curves of the proposed schemes with the partial diffusion configuration.

Note that this scheme satisfies the constraint to span the whole parameter space. Correspondingly, we use a sequential partial-diffusion scheme such that each node shares the same coefficients in order. In the proposed schemes, we choose $\delta = 0.9$ and the step sizes of the all schemes are $\mu_i = 0.042$. For the construction updates, $\eta_i = 0.0035$ and $\eta_i = 0.75$ in the single-bit and scalar diffusion strategies, respectively. The Fig. 10 shows that sequential selection scheme provides enhanced estimation performance also for the compressive diffusion strategies. In the Fig. 10, we also observe that both scalar diffusion and partial diffusion approaches achieve comparable performance.

We can enhance the performance of the scheme through the adaptation of the confidence parameter irrespective of the cooperation rule. As an example, in Fig. 11, we compare the time evolution of the MSD of the scalar diffusion scheme for the adaptive and fixed confidence parameter cases with the Metropolis and uniform combination rules. We use the same configuration with the example 1, initialize $\alpha_{i,t} = 2$, and set $\mu_{cvx} = 10$. Additionally, in Fig. 12, we also plot the time-evolution of the scalar diffusion scheme for $\delta = 0.95$. Note that the

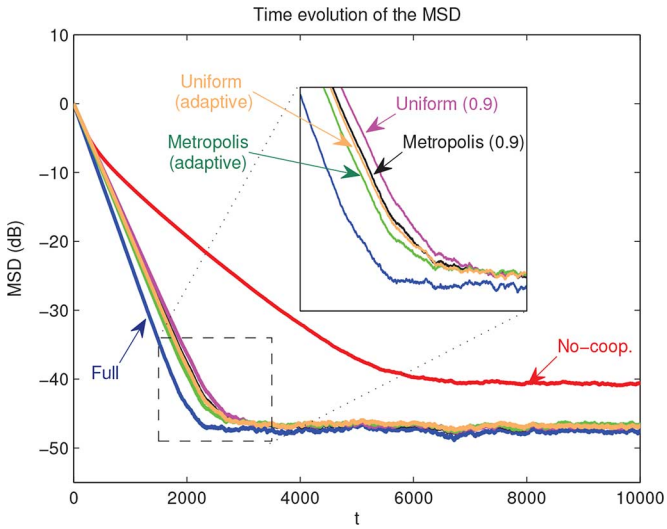

Fig. 11. Comparison of the adaptive and fixed confidence parameter for the Metropolis and uniform combination rules.

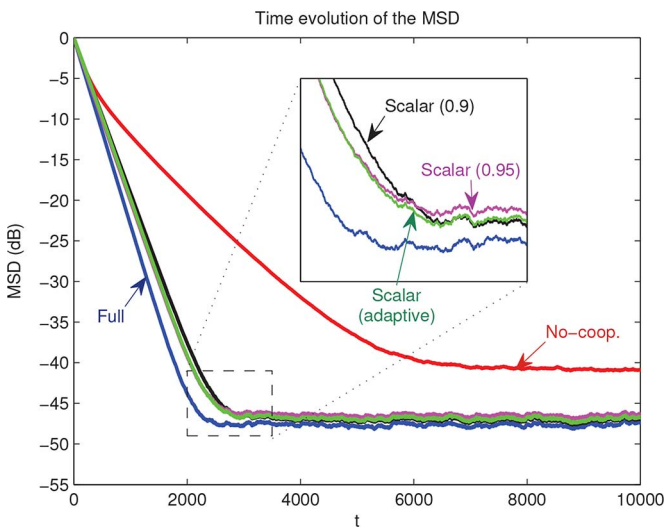

Fig. 12. Comparison of the adaptive and fixed confidence parameter for the scalar diffusion scheme.

adaptive scheme converges as fast as $\delta = 0.95$ scheme while achieving smaller steady-state error similar to $\delta = 0.9$ scheme. Hence, through the confidence parameter we can enhance the performance of the compressive diffusion scheme for certain scenarios.

In the second example, we examine the convergence performance of the adaptive confidence parameter in a relatively large network $N = 20$ with a long filter length $M = 100$. We again observe a stationary data $d_{i,t} = \mathbf{u}_{i,t}^T \mathbf{w_o} + v_{i,t}$ for $i \in \{1, 2, \ldots, N\}$. The regressor data $\mathbf{u}_{i,t}$ is zero-mean i.i.d. Gaussian whose standard deviation is around 0.4. The observation noise $v_{i,t}$ is zero-mean i.i.d. Gaussian whose variance is $\sigma_{n_i}^2 = 10^{-1}$. We note that the signal-to-noise ratio is around 1.55 over the network, which is relatively lower than the signal-to-noise ratio for the example 1. The variance of the projection operator $\mathbf{c}_{i,t}$ is $\sigma_{c_i}^2 = 10^{-2}$ ($\sigma_{c_i}^2 = 10^{-3}$) for the scalar (single-bit) diffusion scheme. The parameter of interest $\mathbf{w_o} \in \mathbb{R}^{100}$ is randomly chosen from a Gaussian distribution and normalized such that $\|\mathbf{w_o}\| = 1$. We point out that in this

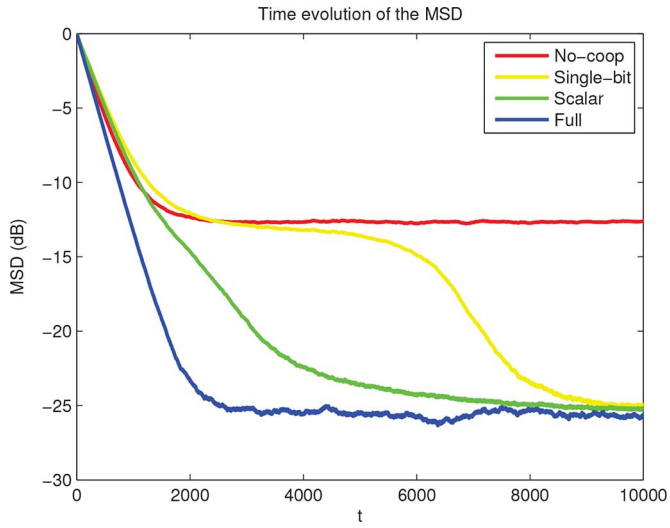

Fig. 13. The global MSD curves in relatively large network and long filter length while the confidence parameter is adapted in time.

example, the overall communication burden in the scalar diffusion strategy is **1%** of the full diffusion configuration while the overall communication load in the single-bit diffusion strategy is given by 20-bits per iterations.

We again use the Metropolis rule as the combination rule, however, in this example, we adapt the confidence parameter through (79) and (80) where we resort to the convex mixture of the adaptive filtering algorithms [35]–[38]. We also choose the step sizes the same for the distributed LMS update (17) of all configurations at all nodes, i.e., $\mu_i = 0.01$. In example 2, the step sizes for the construction update (18) are $\eta_i = 0.01$ (for the single-bit diffusion approach) and $\eta_i = 0.5$ (for the scalar diffusion approach). We set $\mu_{\text{cvx}} = 250$ in (80). The Fig. 13 shows the global MSD curves of the no-cooperation, single-bit, scalar and full diffusion strategies. We observe that the adaptive confidence parameter improves the convergence performance of the compressive diffusion strategies far more such that they achieve comparable performance while the reduction of the communication load is tremendous.

XI. Conclusion

In the diffusion based distributed estimation strategies, the communication load increases far more in the large networks or highly connected network of nodes. Hence, the compressive diffusion approach plays an essential role in achieving comparable convergence performance to the full diffusion configurations while reducing the communication load tremendously. We provide a complete performance analysis for the compressive diffusion strategies. We analyze the mean-square convergence, the steady-state behavior and the tracking performance of the scalar and single-bit diffusion approaches. The numerical examples show that the theoretical analysis model the simulated results accurately. Additionally, we introduce the confidence parameter concept, which adds one more freedom of dimension to the combination rule in order to improve the convergence performance. When we adapt the confidence parameter using the well-known adaptive mixture algorithms, we observe enormous enhancement in the convergence performance of the compressive diffusion strategies even for relatively long filter lengths.

APPENDIX A
PROOF FOR LEMMA 1

We first show the equality of (46) for the two-node case. Then the extension for a larger network is straight forward. We can rewrite the term on the left hand side (LHS) of (46) as

$$E\left[\tilde{\boldsymbol{\psi}}_t^T \mathbf{X}_{\mathbf{u}}^T \boldsymbol{\Sigma}_2 \mathbf{N}\mathbf{C}_t \mathrm{sign}(\boldsymbol{\epsilon}_t)\right] = E\left[\tilde{\boldsymbol{\psi}}_t^T \mathbf{X}_{\mathbf{u}}^T \underbrace{\begin{bmatrix} \boldsymbol{\varsigma}_1 & \boldsymbol{\varsigma}_2 \\ \boldsymbol{\varsigma}_3 & \boldsymbol{\varsigma}_4 \end{bmatrix}}_{\boldsymbol{\Sigma}_2} \mathbf{N}\mathbf{C}_t \mathrm{sign}(\boldsymbol{\epsilon}_t)\right]. \quad (81)$$

After some algebra, (81) yields

$$\begin{aligned} &E\left[\tilde{\boldsymbol{\psi}}_t^T \mathbf{X}_{\mathbf{u}}^T \boldsymbol{\Sigma}_2 \mathbf{N}\mathbf{C}_t \mathrm{sign}(\boldsymbol{\epsilon}_t)\right] \\ &\quad = E\left[\left(\gamma_{11}\tilde{\boldsymbol{\varphi}}_{1,t}^T + \gamma_{12}\tilde{\mathbf{a}}_{2,t}^T\right)\boldsymbol{\varsigma_1}\eta_1 \mathbf{c}_{1,t}\mathrm{sign}(\epsilon_{1,t})\right] \\ &\qquad + E\left[\left(\gamma_{11}\tilde{\boldsymbol{\varphi}}_{1,t}^T + \gamma_{12}\tilde{\mathbf{a}}_{2,t}^T\right)\boldsymbol{\varsigma_2}\eta_2 \mathbf{c}_{2,t}\mathrm{sign}(\epsilon_{2,t})\right] \\ &\qquad + E\left[\left(\gamma_{22}\tilde{\boldsymbol{\varphi}}_{2,t}^T + \gamma_{21}\tilde{\mathbf{a}}_{1,t}^T\right)\boldsymbol{\varsigma_3}\eta_1 \mathbf{c}_{1,t}\mathrm{sign}(\epsilon_{1,t})\right] \\ &\qquad + E\left[\left(\gamma_{22}\tilde{\boldsymbol{\varphi}}_{2,t}^T + \gamma_{21}\tilde{\mathbf{a}}_{1,t}^T\right)\boldsymbol{\varsigma_4}\eta_2 \mathbf{c}_{2,t}\mathrm{sign}(\epsilon_{2,t})\right]. \quad (82) \end{aligned}$$

In order to evaluate the expectations on the RHS of (82), by the Assumption 2 and the Price's result [40]–[42], we obtain

$$\begin{aligned} &E\left[\tilde{\boldsymbol{\psi}}_t^T \mathbf{X}_{\mathbf{u}}^T \boldsymbol{\Sigma}_2 \mathbf{N}\mathbf{C}_t \mathrm{sign}(\boldsymbol{\epsilon}_t)\right] \\ &\quad = E\left[\left(\gamma_{11}\tilde{\boldsymbol{\varphi}}_{1,t}^T + \gamma_{12}\tilde{\mathbf{a}}_{2,t}^T\right)\boldsymbol{\varsigma_1}\eta_1 \mathbf{c}_{1,t}\epsilon_{1,t}\right]\frac{E|\epsilon_{1,t}|}{E\left[\epsilon_{1,t}^2\right]} \\ &\qquad + E\left[\left(\gamma_{11}\tilde{\boldsymbol{\varphi}}_{1,t}^T + \gamma_{12}\tilde{\mathbf{a}}_{2,t}^T\right)\boldsymbol{\varsigma_2}\eta_2 \mathbf{c}_{2,t}\epsilon_{2,t}\right]\frac{E|\epsilon_{2,t}|}{E\left[\epsilon_{2,t}^2\right]} \\ &\qquad + E\left[\left(\gamma_{22}\tilde{\boldsymbol{\varphi}}_{2,t}^T + \gamma_{21}\tilde{\mathbf{a}}_{1,t}^T\right)\boldsymbol{\varsigma_3}\eta_1 \mathbf{c}_{1,t}\epsilon_{1,t}\right]\frac{E|\epsilon_{1,t}|}{E\left[\epsilon_{1,t}^2\right]} \\ &\qquad + E\left[\left(\gamma_{22}\tilde{\boldsymbol{\varphi}}_{2,t}^T + \gamma_{21}\tilde{\mathbf{a}}_{1,t}^T\right)\boldsymbol{\varsigma_4}\eta_2 \mathbf{c}_{2,t}\epsilon_{2,t}\right]\frac{E|\epsilon_{2,t}|}{E\left[\epsilon_{2,t}^2\right]}. \quad (83) \end{aligned}$$

Rearranging (83) into a matrix product form leads (46). Following the same way, we can also get (47) and the proof is concluded. □

APPENDIX B
PROOF FOR LEMMA 2

We derive the RHS of (52) for the two-node case for notational simplicity, however, the derivation holds for any order of network. For the two-node case, the LHS of (52) yields

$$\begin{aligned} &E\left[\mathrm{sign}(\boldsymbol{\epsilon}_t)^T \mathbf{C}_t^T \mathbf{N}\boldsymbol{\Sigma}_4 \mathbf{N}\mathbf{C}_t \mathrm{sign}(\boldsymbol{\epsilon}_t)\right] \\ &\quad = E\left[\mathrm{sign}(\epsilon_{1,t})\mathbf{c}_{1,t}^T \eta_1 \boldsymbol{\varsigma_1} \eta_1 \mathbf{c}_{1,t}\mathrm{sign}(\epsilon_{1,t})\right] \\ &\qquad + E\left[\mathrm{sign}(\epsilon_{1,t})\mathbf{c}_{1,t}^T \eta_1 \boldsymbol{\varsigma_2} \eta_2 \mathbf{c}_{2,t}\mathrm{sign}(\epsilon_{2,t})\right] \\ &\qquad + E\left[\mathrm{sign}(\epsilon_{2,t})\mathbf{c}_{2,t}^T \eta_2 \boldsymbol{\varsigma_3} \eta_1 \mathbf{c}_{1,t}\mathrm{sign}(\epsilon_{1,t})\right] \\ &\qquad + E\left[\mathrm{sign}(\epsilon_{2,t})\mathbf{c}_{2,t}^T \eta_2 \boldsymbol{\varsigma_4} \eta_2 \mathbf{c}_{2,t}\mathrm{sign}(\epsilon_{2,t})\right]. \end{aligned}$$

We re-emphasize that the regressor $\mathbf{c}_{i,t}$ is spatially and temporarily independent. Hence, we obtain

$$\begin{aligned} &E\left[\mathrm{sign}(\boldsymbol{\epsilon}_t)^T \mathbf{C}_t^T \mathbf{N}\boldsymbol{\Sigma}_4 \mathbf{N}\mathbf{C}_t \mathrm{sign}(\boldsymbol{\epsilon}_t)\right] \\ &\quad = E\left[\mathbf{c}_{1,t}^T \eta_1 \boldsymbol{\varsigma_1} \eta_1 \mathbf{c}_{1,t}\right] + E\left[\mathbf{c}_{2,t}^T \eta_2 \boldsymbol{\varsigma_4} \eta_2 \mathbf{c}_{2,t}\right] \\ &\qquad + E\left[\mathbf{c}_{1,t}\mathrm{sign}(\epsilon_{1,t})\right]^T \eta_1 \boldsymbol{\varsigma_2} \eta_2 E\left[\mathbf{c}_{2,t}\mathrm{sign}(\epsilon_{2,t})\right] \\ &\qquad + E\left[\mathbf{c}_{2,t}\mathrm{sign}(\epsilon_{2,t})\right]^T \eta_2 \boldsymbol{\varsigma_3} \eta_1 E[\mathbf{c}_{1,t}\mathrm{sign}(\epsilon_{1,t})]. \quad (84) \end{aligned}$$

Using the Price's result, we can evaluate the last two terms on the RHS of (84) for $i \in \{1, 2\}$ as

$$E[\mathbf{c}_{i,t}\mathrm{sign}(\epsilon_{i,t})] = \frac{E|\epsilon_{i,t}|}{E[\epsilon_{i,t}^2]} E\left[\mathbf{c}_{i,t}\epsilon_{i,t}\right].$$

We point out that the terms involving the diagonal entries of the weighting matrix $\boldsymbol{\Sigma}_4$ in (84) do not include the deviation terms. As a result, rearranging (84) into a compact form results in (52). This concludes the proof. □

REFERENCES

[1] A. H. Sayed, S.-Y. 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, 2013.

[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, 2002.

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

[4] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2001, vol. 4, pp. 2033–2036, vol. 4.

[5] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York, NY, USA: Wiley, 2003.

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

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

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

[9] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[10] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, 2012.

[11] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3460–3475, 2012.

[12] M. O. Sayin and S. S. Kozat, "Single bit and reduced dimension diffusion strategies over distributed networks," *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 976–979, 2013.

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

[14] R. G. Baraniuk, V. Cevher, and M. B. Wakin, "Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective," *Proc. IEEE*, vol. 98, no. 6, pp. 959–971, 2010.

[15] R. Arablouei, S. Werne, and K. Dogancay, "Partial-diffusion recursive least-squares estimation over adaptive networks," in *Proc. IEEE 5th Int. Workshop on Computat. Adv. Multi-Sens. Adapt. Process. (CAMSAP)*, 2013, pp. 89–92.

[16] R. Arablouei, S. Werner, Y. F. Huang, and K. Dogancay, "Distributed least mean square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472–483, 2014.

[17] R. Arablouei, S. Werner, Y. F. Huang, and K. Dogancay, "Adaptive distributed estimation based on recursive least-square and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3510–3522, 2014.

[18] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 257–273, 2013.

[19] S. Xie and H. Li, "Distributed LMS estimation over networks with quantised communications," *Int. J. Contr.*, vol. 86, no. 3, pp. 478–492, 2013.

[20] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4782–4795, 2006.
[21] H. Sayyadi and M. R. Doostmohammadian, "Finite-time consensus in directed switching network topologies and time-delayed communications," *Scientia Iranica*, vol. 18, no. 1, pp. 75–85, Feb. 2011.
[22] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus and flocking," in *Proc. Joint 44th IEEE Conf. Decision Contr. Eur. Contr. Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.
[23] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Contr. Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.* vol. 21, no. 6, pp. 1087–1092, 1953 [Online]. Available: http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114
[25] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, 2010.
[26] E. Barker and J. Kelsey, "Recommendation for random number generation using deterministic random bit generators," in *NIST SP800-90A*, 2012 [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf
[27] J. Joutsensalo and T. Ristaniemi, "Synchronization by pilot signal," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1999, pp. 2663–2666.
[28] F. T. Castoldi and M. L. R. Campos, "Application of a minimum-disturbance description to constrained adaptive filters," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1215–1218, 2013.
[29] M. A. Donmez, H. A. Inan, and S. S. Kozat, "Adaptive mixture methods based on Bregman divergences," *Digit. Signal Process.*, vol. 23, pp. 86–97, 2013.
[30] A. H. Sayed, T. Y. Al-Naffouri, and V. H. Nascimento, "Energy conservation in adaptive filtering," in *Nonlinear Signal and Image Processing: Theory, Methods, and Applications*, K. E. Barner and G. R. Arce, Eds. Boca Raton, FL, USA: CRC, 2003.
[31] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of adaptive filters with error nonlinearities," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 653–663, 2003.
[32] T. Y. Al-Naffouri and A. H. Sayed, "Transient analysis of data-normalized adaptive filters," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 639–652, 2003.
[33] X. Zhao and A. H. Sayed, "Combination weights for diffusion strategies with imperfect information exchange," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 398–402.
[34] S.-Y. Tu and A. H. Sayed, "Adaptive networks with noisy links," in *Proc. IEEE Global Telecommun. Conf.*, 2011, pp. 1–5.
[35] J. Arenas-Garcia, V. Gomez-Verdejo, and A. R. Figueiras-Vidal, "New algorithms for improved adaptive convex combination of LMS transversal filters," *IEEE Trans. Instrum. Meas.*, vol. 54, no. 6, pp. 2239–2249, 2005.
[36] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, 2006.
[37] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3137–3149, 2008.
[38] S. S. Kozat, A. T. Erdogan, A. C. Singer, and A. H. Sayed, "Steady state MSE performance analysis of mixture approaches to adaptive filtering," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4050–4063, Aug. 2010.
[39] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of back propagation learning," in *From Natural to Artificial Neural Computation*, J. Mira and F. Sandoval, Eds. Berlin Heidelberg: Springer, 1995.
[40] R. Price, "A useful theorem for nonlinear devices having gaussian inputs," *IEEE Trans. Inf. Theory*, vol. 4, no. 2, pp. 69–72, 1958.
[41] E. McMahon, "An extension of price's theorem (corresp.)," *IEEE Trans. Inf. Theory*, vol. 10, no. 2, pp. 168–168, 1964.
[42] T. Koh and E. J. Powers, "Efficient methods of estimate correlation functions of Gaussian processes and their performance analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 1032–1035, 1985.

Muhammed O. Sayin was born in Erzincan, Turkey, in 1990. He received the B.S. degree with high honors in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2013.

He is currently working toward the M.S. degree in the Department of Electrical and Electronics Engineering at Bilkent University. His research interests include distributed signal processing, adaptive filtering theory, machine learning, and statistical signal processing.

Suleyman Serdar Kozat (A'10–M'11–SM'11) received the B.S. degree with full scholarship and high honors from Bilkent University, Turkey. He received the M.S. and Ph.D. degrees in electrical and computer engineering from University of Illinois at Urbana Champaign, Urbana. He is a graduate of Ankara Fen Lisesi.

After graduation, he joined IBM Research, T. J. Watson Research Lab, Yorktown, New York, as a Research Staff Member in the Pervasive Speech Technologies Group. While doing his Ph.D., he was also working as a Research Associate at Microsoft Research, Redmond, WA, in the Cryptography and Anti-Piracy Group. He holds several patent inventions due to his research accomplishments at IBM Research and Microsoft Research. After serving as an Assistant Professor at Koc University, he is currently an Assistant Professor (with the Associate Professor degree) at the electrical and electronics department of Bilkent University.

Dr. Kozat is President of the IEEE Signal Processing Society, Turkey Chapter. He has been elected to the IEEE Signal Processing Theory and Methods Technical Committee and IEEE Machine Learning for Signal Processing Technical Committee, 2013. He has been awarded IBM Faculty Award by IBM Research in 2011, Outstanding Faculty Award by Koc University in 2011 (granted the first time in 16 years), Outstanding Young Researcher Award by the Turkish National Academy of Sciences in 2010, ODTU Prof. Dr. Mustafa N. Parlar Research Encouragement Award in 2011, Outstanding Faculty Award by Bilim Kahramanlari, 2013 and holds Career Award by the Scientific Research Council of Turkey, 2009.