

Communication Efficient Channel Estimation Over Distributed Networks

Muhammed O. Sayin*, N. Denizcan Vanli*, Tolga Göze[†] and Suleyman S. Kozat*

*Department of Electrical and Electronics Engineering

Bilkent University, Ankara, Turkey 06800

Email: {sayin, vanli, kozat}@ee.bilkent.edu.tr

[†]Alcatel-Lucent, Istanbul, Turkey

Email: tolga.goze@alcatel-lucent.com

Abstract—We study diffusion based channel estimation in distributed architectures suitable for various communication applications such as cognitive radios. Although the demand for distributed processing is steadily growing, these architectures require a substantial amount of communication among their nodes (or processing elements) causing significant energy consumption and increase in carbon footprint. Due to growing awareness of telecommunication industry’s impact on the environment, the need to mitigate this problem is indisputable. To this end, we introduce algorithms significantly reducing the communication load between distributed nodes, which is the main cause in energy consumption, while providing outstanding performance. In this framework, after each node produces its local estimate of the communication channel, a single bit or a couple of bits of information is generated using certain random projections. This newly generated data is diffused and then used in neighboring nodes to recover the original full information, i.e., the channel estimate of the desired communication channel. We provide the complete state-space description of these algorithms and demonstrate the substantial gains through our experiments.

I. INTRODUCTION

The demand on distributed networks (or processing units) is steadily growing due to increased efficiency and performance improvements they provide in various different applications [1]. The broadened perspective provided by these architectures significantly enhances channel estimation performance; is used to avoid environmental obstructions; or permit resource sharing and allocation. However, these architectures demand astounding amount of communication between their nodes causing significant energy consumption and increase in carbon footprint. Due to growing awareness of telecommunication industry’s impact on the environment, the need to mitigate this carbon footprint is indisputable. To this end, we introduce novel approaches substantially reducing the communication load for the distributed architectures, which is the main source of energy consumption, without any significant performance degradation [1].

In particular, we investigate “diffusion” based distributed architectures in a channel identification (or estimation) framework, where distributed nodes are used for channel estimation and share their information to improve overall estimation accuracy. The diffusion based distributed algorithms define a strategy in which the nodes from a predefined neighborhood share information with each other [2]–[4]. Such approaches

that diffuse information to their neighbors instead of using a central processing units are stable against time-varying statistical profiles [2], however, entail a high amount of communication load. For example, in a network of N nodes, where M denotes the number of channel coefficient, then the overall communication burden among nodes is given by $N \times M$ at each instant, which can be highly impractical for certain applications.

We propose diffusion based cooperation strategies that have significantly less communication load (e.g., a single bit or a couple of bits of information exchange) and achieve comparable performance to the full information exchange configurations under certain settings. In this framework, each node estimates an unknown communication channel observed through a linear model. After local estimates of the desired channel is produced in each node, a single bit or multiple bits of information is generated using certain random projections of the local estimates. This new information is then diffused and utilized in neighboring nodes instead of the original estimates; significantly reducing the communication load in the network. We only require synchronization of this randomized projection operation, which can be achieved using simple pilot signals [5].

Our approach differs from quantization based diffusion strategies such as [6], where quantized parameter estimates are exchanged among nodes to avoid infinite precision, in terms of the compression of the diffused information. Here, we substantially compress the exchanged information, even to a single bit, and perform local adaptive operations at each node to recover the full channel information. In this sense, our method is more akin to compressive sensing rather than to a quantization framework. To this end, we propose algorithms to significantly reduce the amount of communication between nodes for diffusion based distributed strategies and illustrate the comparable convergence performance of these algorithms in different numerical examples. We also emphasize that our algorithms are generic and can be straightforwardly extended to perform prediction, hypothesis testing or filtering in diffusion based distributed algorithms with significant reduction in communication load.

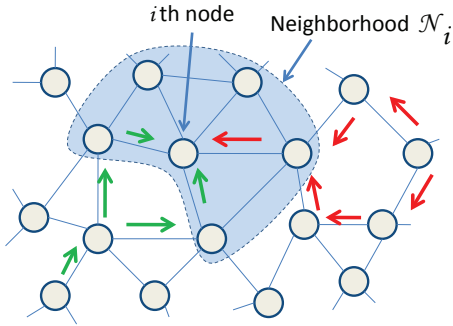


Fig. 1: A distributed network of nodes.

II. PROBLEM DESCRIPTION

Consider a network of nodes distributed spatially as shown in Fig. 1, which models a wide range of applications including spectrum sensing in cognitive radio networks to distributed processing in multi-cores [1]–[4]. Here, we have n distributed nodes and two nodes are considered neighbors if they can exchange information, where we assume that the information exchange is bi-directional. For each node i , we denote the set of its neighbors (including itself) as \mathcal{N}_i . Here, each node estimates an unknown communication channel $\mathbf{h}_o \in \mathbb{R}^m$ observed through a linear model $d_i(t) = \mathbf{u}_i^T(t)\mathbf{h}_o + v_i(t)$, where the observations are corrupted by additive white noise¹ and diffuses its information to neighboring nodes. The observation noise is temporally and spatially white (or independent), i.e., $E[v_i(t)v_j(l)] = \sigma_i^2\delta(i-j)\delta(t-l)$, where $\delta(\cdot)$ is the Kronecker delta and σ_i^2 is the variance of the noise. The underlying \mathbf{h}_o can also represent spectrum parameters or a state vector of an unknown system² in different applications, where our derivations still hold. The linear transformation (or the regressor) $\mathbf{u}_i(t)$ is known by the node \mathcal{N}_i but unknown to the other nodes. We also assume that $\mathbf{u}_i(t)$'s are spatially and temporally uncorrelated with each other and with the observation noise.

At each node an adaptive channel estimation algorithm is used to recover \mathbf{h}_o such as a stochastic gradient approach [7] given as

$$\phi_i(t+1) = (\mathbf{I} - \mu_i\mathbf{u}_i(t)\mathbf{u}_i^T(t))\mathbf{h}_i(t) + \mu_id_i(t)\mathbf{u}_i(t), \quad (1)$$

$\mu_i > 0$, where $\mathbf{h}_i(t)$ represents the current estimate and $\phi_i(t+1)$ is the updated estimate after the new observation. We emphasize that our approach is generic such that one can use different estimation algorithms instead of (1) [7]. As the diffusion strategy, we next use the adapt-then-combine (ATC)

¹We represent vectors (matrices) by bold lower (upper) case letters. For a matrix \mathbf{A} (or a vector \mathbf{a}), \mathbf{A}^T is the transpose. $\|\mathbf{a}\|$ is the Euclidean norm. For notational simplicity we work with real data and all random variables have zero mean. The sign of a is denoted by $\text{sign}(a)$ (0 is considered positive without loss of generality). For a vector \mathbf{a} , $\dim(\mathbf{a})$ denotes the length. The expectation of a vector or a matrix is denoted with an over-line, i.e. $E[\mathbf{a}] = \bar{\mathbf{a}}$. The $\text{diag}(\mathbf{A})$ returns a new matrix with only the main diagonal of \mathbf{A} while $\text{diag}(\mathbf{a})$ puts \mathbf{a} on the main diagonal of the new matrix.

²Although we assume a time invariant desired vector, our derivations can be readily extended to certain non-stationary models [7].

diffusion strategy as an example, however, our derivations also cover other diffusion strategies [2]. In the ATC strategy, at each node i , the final channel estimate is constructed as

$$\mathbf{h}_i(t+1) = \sum_{k \in \mathcal{N}_i} \lambda_{i,k} \phi_k(t+1), \quad (2)$$

after the estimates of the neighboring nodes arrive to i , where $\lambda_{i,k}$'s are the combination weights $\sum_{k \in \mathcal{N}_i} \lambda_{i,k} = 1$ and $\lambda_{i,k} \geq 0$. The combination weights $\lambda_{i,k}$ can also be adapted in time, however, we use weights that are constant in time with the simplex constraint (since the stabilization effect of such weights is demonstrated in [2]).

In the diffusion based distributed networks, the entire channel estimates $\phi_k(t+1)$'s are exchanged within the neighborhood, which requires a substantial amount of communication between the nodes even if efficient quantization methods are used [6]. In the next section, we study algorithms that significantly reduce the amount of information exchange between the neighboring nodes.

III. A SECOND ESTIMATION LEVEL INSTEAD OF DIFFUSION

In the well known formulation (2), each node i receives the entire vector of estimated channel coefficients from all its neighbors. This requires an exchange of $O(m)$ “real” coefficients for each node. Instead of directly transmitting the estimated vector of coefficients, we introduce a different perspective [5]. We first observe that for the node i , the estimated channel coefficients of its neighbors, $\phi_k(t+1)$, $k \in \mathcal{N}_i$, are naturally unknown. Hence, instead of collecting these unknown coefficients from the neighboring nodes, we next formulate another estimation level, where the node i (in addition to its original task) estimates $\phi_k(t+1)$'s of its neighbors. In this case instead of the original $\phi_k(t+1)$'s, each node i constructs estimated vectors, say $\mathbf{a}_k(t+1)$'s, of $\phi_k(t+1)$'s that are directly used in (2) instead of the original $\phi_k(t+1)$'s [5]. We demonstrate through this perspective, we can tremendously reduce the communication load, i.e., from a continuum to a couple of bits, while providing nearly equal performance to the original formulation.

For the node i , $\phi_k(t+1)$ is unknown and suppose both nodes i and k calculates a linear transformation of $\phi_k(t+1)$ through a randomized linear transformation $\mathbf{c}^T(t+1)\phi_k(t+1)$, where construction of $\mathbf{c}(t+1)$ is detailed later in the paper. Then, if $\mathbf{a}_k(t+1)$ accurately represents $\phi_k(t+1)$, then $\mathbf{c}^T(t+1)\mathbf{a}_k(t+1)$ should be close to $\mathbf{c}^T(t+1)\phi_k(t+1)$. In this sense, in this new framework, $\phi_k(t+1)$ is the desired vector of coefficients and $\mathbf{a}_k(t+1)$ is our estimate. Hence, we formulate another, i.e., a second level of, estimation framework at node i to recover $\phi_k(t+1)$ by a stochastic gradient algorithm by minimizing the estimation error as

$$\begin{aligned} \mathbf{a}_k(t+1) &= \mathbf{a}_k(t) + \rho_k \nabla \mathbf{a}_k \epsilon_k^2(t+1) \\ &= \mathbf{a}_k(t) + \rho_k \epsilon_k(t+1) \mathbf{c}(t+1), \end{aligned} \quad (3)$$

where we have $\epsilon_k(t+1) \triangleq \mathbf{c}^T(t+1)\phi_k(t+1) - \mathbf{c}^T(t+1)\mathbf{a}_k(t)$, i.e., the estimation error, and $\rho_k > 0$ is the learning rate. After

we perform the update in (3), we then use $\mathbf{a}_k(t+1)$ at the node i instead of $\phi_k(t+1)$ in (2).

For this formulation, the node k needs to provide only the scalar $\epsilon_k(t+1)$ instead of the whole $\phi_k(t+1)$ to the node i since all the other quantities are known by both nodes i and k . Note that both nodes i and k can synchronously run the same (3) provided that $\epsilon_k(t+1)$ is communicated to i from k , i.e., the node k can also construct $\mathbf{a}_k(t+1)$ in order to construct $\epsilon_k(t+1)$, which is transmitted to the node i .

To accomplish this effectively, we next introduce an adaptive quantization scheme in order to effectively and efficiently construct $\epsilon_k(t+1)$ at the node i for all $k \in \mathcal{N}_i$. If the estimation scheme is successful, then $\epsilon_k(t+1)$ converges to a white noise process due to the linear filtering formulation in (3). Hence adaptive quantizers using linear predictive formulations are usually ineffective since there should be no correlation left in $\epsilon_k(t)$ if estimation is successful. In order to effectively perform quantization, we next introduce a scalar adaptive quantization framework with guaranteed synchronization between nodes. Note that one can straightforwardly extend our formulation to a vector quantization framework, however, we use a scalar quantization framework for notational simplicity.

At time t , we quantize $\epsilon_k(t+1)$ as $\tilde{\epsilon}_k(t+1) = Q_{k,t}(\epsilon_k(t+1))$ at the node k , where $Q_{k,t}(\cdot) : \mathbb{R} \rightarrow \{q_{k,t,1}, \dots, q_{k,t,2^b}\}$ is a time adaptive quantizer using b -bit as explained in the following, and transmit the quantized bits to the node i . We also assume that $\epsilon_k(t+1)$ is Gaussian distributed where this assumption is widely used in the literature [7]. Hence, to construct $Q_{k,t}(\cdot)$, we need the variance and the mean of the process $\epsilon_k(t+1)$. To adaptively estimate the mean and variance of the process, we can only use the previous quantized bits $\tilde{\epsilon}_k(1), \dots, \tilde{\epsilon}_k(t)$ to guarantee synchronization between the nodes i and k . To this end, we use a recursive mean estimator

$$\tilde{m}_k(t+1) = (1 - \eta_k)\tilde{m}_k(t) + \eta_k\tilde{\epsilon}_k(t),$$

and a recursive variance estimator

$$\tilde{\sigma}_k^2(t+1) = (1 - \eta_k)\tilde{\sigma}_k^2(t) + \eta_k\tilde{\epsilon}_k^2(t),$$

using only the quantized error samples [8], where the forgetting factors $\eta_k > 0$ are set equal for notational simplicity. Based on $\tilde{m}_k(t)$, $\tilde{\sigma}_k(t)$, and assuming that the estimation error is Gaussian distributed, we construct

$$\{q_{k,t,1}, \dots, q_{k,t,2^b}\} = \arg \min_{q_1, \dots, q_{2^b}} \int_{-\infty}^{\infty} \frac{(x - Q(x))^2}{\sqrt{2\pi\tilde{\sigma}_k^2(t)}} \exp\left\{-\frac{[x - \tilde{m}_k(t)]^2}{2\tilde{\sigma}_k^2(t)}\right\} dx,$$

where

$$Q(x) \triangleq \arg \min_{q_i \in \{q_1, \dots, q_{2^b}\}} \|x - q_i\|^2,$$

yielding the mean square error optimal quantization algorithm (if the estimated mean and variance converge). We next provide a mean stability analysis of the overall diffusion based algorithm.

IV. MEAN STABILITY ANALYSIS

The diffusion update at the node i with the second layer of adaptation can be written in a compact form as

$$\phi_i(t+1) = \mathbf{h}_i(t) + \mu_i e_i(t) \mathbf{u}_i(t), \quad (4)$$

$$\mathbf{a}_k(t+1) = \mathbf{a}_k(t) + \rho_k Q_{k,t}(\epsilon_k(t+1)) \mathbf{c}(t+1), \quad (5)$$

$$\mathbf{h}_i(t+1) = \lambda_{i,i} \phi_i(t+1) + \sum_{k \in \mathcal{N}_i \setminus i} \lambda_{i,k} \mathbf{a}_k(t+1), \quad (6)$$

where $\mu_i > 0$ and $\rho_k > 0$. Note that (5) is also carried out in the neighboring nodes $k \in \mathcal{N}_i \setminus i$. Since we have two estimation algorithms, we define deviations from the parameter of interests as

$$\Delta \phi_k(t+1) = \mathbf{h}_o - \phi_k(t+1), \quad (7)$$

$$\Delta \mathbf{a}_k(t+1) = \phi_k(t+1) - \mathbf{a}_k(t+1). \quad (8)$$

Substituting (8) and (7) into (6), we get the final estimate as

$$\mathbf{h}_i(t+1) = \sum_{k \in \mathcal{N}_i} \lambda_{i,k} \phi_k(t+1) - \sum_{k \in \mathcal{N}_i \setminus i} \lambda_{i,k} \Delta \mathbf{a}_k(t+1).$$

To continue with the mean stability analysis, we make the following assumptions:

- 1) $\mathbf{c}(t)$ and $\mathbf{u}_k(t)$ are temporally independent.
- 2) The error $\epsilon_k(t)$ and $\mathbf{c}(t)$ are jointly Gaussian and uncorrelated. For sufficiently small step size and long filter length, this assumption is true [7].
- 3) The original parameter estimates $\phi_i(t)$ vary slowly relative to the constructed estimates $\mathbf{a}_i(t)$ through the appropriate step sizes such that

$$\Delta \mathbf{a}_k(t) = \phi_k(t) - \mathbf{a}_k(t) \cong \phi_k(t+1) - \mathbf{a}_k(t) \text{ or}$$

$$\Delta \mathbf{a}_k(t+1) = \phi_k(t+1) - \mathbf{a}_k(t+1) \cong \phi_k(t) - \mathbf{a}_k(t+1).$$

- 4) The quantization error $\Delta \epsilon_k(t+1) = \epsilon_k(t+1) - Q_{k,t}(\epsilon_k(t+1))$ is i.i.d. with zero mean independent from the regressors and observation noise processes [8].

To construct a complete state space recursion, we define the following global variables

$$\mathbf{U}(t) \triangleq \begin{bmatrix} \mathbf{u}_1(t) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{u}_N(t) \end{bmatrix},$$

$$\mathbf{v}(t) \triangleq \begin{bmatrix} v_1(t) \\ \vdots \\ v_N(t) \end{bmatrix}, \quad \Delta \phi(t) \triangleq \begin{bmatrix} \Delta \phi_1(t) \\ \vdots \\ \Delta \phi_N(t) \end{bmatrix},$$

$$\Delta \mathbf{a}(t) \triangleq \begin{bmatrix} \Delta \mathbf{a}_1(t) \\ \vdots \\ \Delta \mathbf{a}_N(t) \end{bmatrix}, \quad \Delta \epsilon(t) \triangleq \begin{bmatrix} \Delta \epsilon_1(t) \\ \vdots \\ \Delta \epsilon_N(t) \end{bmatrix}.$$

Using these global variables, for the first adaptation layer, we get

$$\Delta \phi(t+1) = (\mathbf{I} - \mathbf{D}\mathbf{U}(t)\mathbf{U}(t)^T) \mathbf{G} \Delta \phi(t) - (\mathbf{I} - \mathbf{D}\mathbf{U}(t)\mathbf{U}(t)^T) \tilde{\mathbf{G}} \Delta \mathbf{a}(t) + \mathbf{D}\mathbf{U}(t)\mathbf{v}(t), \quad (9)$$

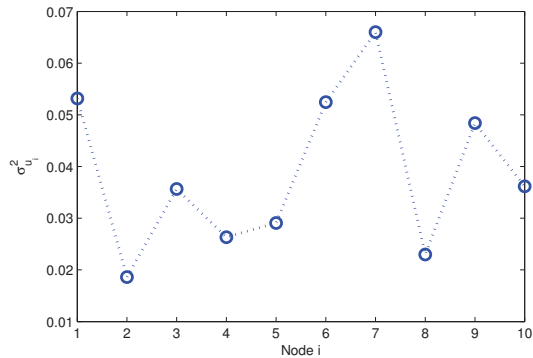


Fig. 2: Statistical profile of the example network ($\sigma_v^2 = 0.01$).

where $\mathbf{G} \triangleq \mathbf{\Lambda} \otimes \mathbf{I}_m$ is the transition matrix (and \otimes is the Kronecker product), $\tilde{\mathbf{G}} \triangleq \mathbf{G} - \text{diag}(\mathbf{G})$, $\mathbf{\Lambda} \triangleq [\lambda_{i,k}]$ is the combination matrix and $\mathbf{D} \triangleq \text{diag}([\mu_1, \mu_2, \dots, \mu_N]) \otimes \mathbf{I}_m$.

The global update for the reconstructed parameters yields

$$\Delta \mathbf{a}(t+1) = (\mathbf{I} - \mathbf{S}\mathbf{M}(t)) \Delta \mathbf{a}(t) + \mathbf{S}\mathbf{L}(t)\Delta \boldsymbol{\epsilon}(t+1), \quad (10)$$

where $\mathbf{S} \triangleq \text{diag}([\rho_1, \rho_2, \dots, \rho_N]) \otimes \mathbf{I}_m$, $\mathbf{L}(t) \triangleq \mathbf{I}_m \otimes \mathbf{c}(t+1)$ and

$$\mathbf{M}(t) = \mathbf{I}_m \otimes \left(\frac{\mathbf{c}(t+1)\mathbf{c}(t+1)^T}{\mathbf{c}(t+1)^T \mathbf{c}(t+1)} \right).$$

If we calculate the expectations of (9) and (10) under our assumptions, then we get

$$\begin{bmatrix} \Delta \bar{\boldsymbol{\phi}}(t+1) \\ \Delta \bar{\mathbf{a}}(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{B}\mathbf{G} & \mathbf{B}\tilde{\mathbf{G}} \\ \mathbf{0} & \mathbf{I} - \mathbf{S}\bar{\mathbf{M}}(t) \end{bmatrix} \begin{bmatrix} \Delta \bar{\boldsymbol{\phi}}(t) \\ \Delta \bar{\mathbf{a}}(t) \end{bmatrix}, \quad (11)$$

where $\mathbf{B} \triangleq \mathbf{I} - \overline{\mathbf{D}\mathbf{U}(t)\mathbf{U}(t)^T}$. From (11) we observe that our algorithms are stable in the mean if $|\lambda(\mathbf{I} - \mathbf{S}\bar{\mathbf{M}})| < 1$ (provided that the full diffusion scheme is stable), where $\lambda(\cdot)$'s are the eigenvalues. Assuming $\mathbf{c}(\cdot)$ are i.i.d. zero mean with unit variance, then $|\lambda(\mathbf{I} - \mathbf{S}\bar{\mathbf{M}})| < 1$ if and only if $|1 - \rho_i| < 1$ for all i .

V. NUMERICAL EXAMPLE

We compare the proposed diffusion algorithms with the scalar, full diffusion and the no-cooperation schemes for the example network with $n = 10$ nodes and $m = 10$. Each node i observes a stationary data $d_i(t) = \mathbf{u}_i^T(t)\mathbf{h}_o + v_i(t)$, where $\mathbf{u}_i(t)$ is i.i.d. zero mean Gaussian vector process with auto-covariance matrix $\mathbf{C} = \sigma_{u_i}^2 \mathbf{I}$ and $\sigma_{u_i}^2$ are seen in Fig. 2. The observation noise $v_i(t)$ is zero-mean i.i.d. Gaussian random process with variance $\sigma_v^2 = 0.01$. We set the true channel coefficients $\mathbf{h}_o \in \mathbb{R}^{10}$ randomly from a normal distribution and normalize to have $\|\mathbf{h}_o\| = 1$.

We combine the estimation parameters through a modified Metropolis rule where

$$\lambda_{i,k} = \begin{cases} \frac{2}{m^2} \frac{1}{\max(n_i, n_k)} & \text{if } i \neq k \text{ are linked,} \\ 0 & \text{for } i \text{ and } k \text{ not linked,} \\ 1 - \sum_{k \in \mathcal{N}_i \setminus i} \lambda_{i,k} & \text{for } i = k \end{cases}$$

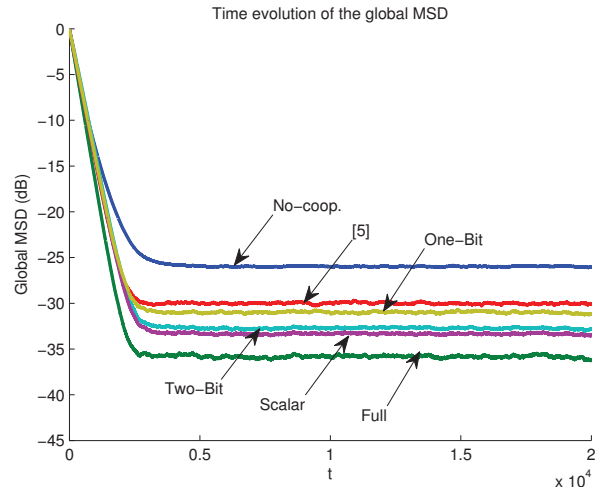


Fig. 3: Global mean-square deviation (MSD) of diffusion and no-cooperation schemes.

and n_i denotes the cardinality of the neighborhood \mathcal{N}_i .

We set the step sizes as 0.05 for the estimation update (4) and 0.01 for the construction update (5) (for the scalar diffusion $\rho_k = 0.1$) so that they converge with the same rate. The forgetting factor η_k is set to 0.02. The randomized projection vector $\mathbf{c}(t)$ is generated i.i.d. Normal random process. In Fig. 3, we compare the global mean-square deviation (MSD) of the diffusion schemes. We observe that introduced schemes enhance the estimation performance of the single-bit diffusion strategy and through the diffusion of two-bit we can achieve almost identical performance with the scalar diffusion strategy.

VI. CONCLUSION

We study diffusion based distributed adaptive channel estimation algorithms that significantly reduce the communication load while providing comparable performance with the full information exchange configurations in our simulations. We achieve this by exchanging either a single bit or a couple of bits of information generated from a second layer of adaptation using random projections. Based on these exchanged information, each node recovers the channel estimates generated by its neighboring nodes (which are then subsequently combined). We also provide a complete state space model and demonstrate the mean stability of the introduced approaches for stationary data. This analysis can also be extended to mean-square and tracking analysis under certain settings.

REFERENCES

- [1] J. J. Xiao, A. Riberio, Z. Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 27–41, 2006.
- [2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [3] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

- [4] ———, “Diffusion detection over adaptive networks using diffusion adaptation,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 1917–1932, 2011.
- [5] M. O. Sayin and S. S. Kozat, “Single-bit and reduced dimension diffusion strategies over distributed networks,” *IEEE Signal Processing Letters*, vol. 20, no. 10, pp. 976–979, 2013.
- [6] S. Xie and H. Li, “Distributed LMS estimation over networks with quantised communications,” *International Journal of Control*, vol. 86, no. 3, pp. 478–492, 2013.
- [7] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley and Sons, 2003.
- [8] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Springer, 1992.