A STRATEGY FOR ADJUSTING COMBINATION WEIGHTS OVER ADAPTIVE NETWORKS

Chung-Kai Yu and Ali H. Sayed

Electrical Engineering Department
University of California, Los Angeles, CA 90095

ABSTRACT

This work proposes a strategy to adjust the combination weights of an adaptive network in order to attain both faster convergence during the transient phase and lower mean-square-error during the steady-state phase. Optimal combination weights are designed for both phases, and a procedure for detecting the transition from one phase to the other is also described. Simulation results illustrate the operation of the proposed strategy.

Index Terms— Adaptive Networks, Phase Detection

1. INTRODUCTION AND RELATED WORK

Adaptive networks consist of a collection of agents that interact with each other through in-network local processing rules in order to estimate and track parameters of interest. The individual agents use combination weights to aggregate the information from their neighbors. Among the various static combination rules that have been used before we mention the uniform rule [1], the Laplacian rule [2, 3], and the Metropolis rule [2]. In previous works [4–6], several schemes were proposed for adaptively adjusting the combination weights in diffusion implementations in order to minimize the network mean-square-error performance. These adaptive schemes were shown to improve the steady-state performance at the expense of some deterioration in convergence rate due to the adaptation of the combination coefficients.

In this work, we propose a mixed strategy that offers the advantage of both faster convergence rate and better steady-state performance. We show that the adjustment of the combination coefficients can be split into two phases. During the network transient phase, the coefficients are selected to ensure fastest convergence. The arguments will show that during this phase, the weights should be chosen according to the uniform combination rule. Subsequently, the network should choose the weights in order to minimize its mean-square-error performance. The discussion will further show that possible constructions for the weights during this second phase are according to the relative-variance rule defined later in (19). It is clear, though, that in order to implement this policy, nodes need to know when to switch from operating in the transition phase to operating in the steady-state phase. We therefore formulate a hypothesis testing problem at each node that enables the nodes to detect when the network is switching from one phase to the other. In this manner, the proposed strategy for adjusting the combination weights becomes *fully* adaptive and ends up leading to better mean-square-error performance at a desirable faster convergence rate.

We focus in this article on diffusion strategies for adaptation and learning over networks. These strategies have been shown before to have superior mean-square-error performance than other distributed strategies such as consensus-based solutions [7]. There have of course been useful studies in the literature on accelerating the convergence rate of consensus strategies (e.g., [8–12]). However, these works focus on the traditional consensus implementation for computing averages and do not deal with the broader problem of adaptive implementations with streaming data, which is the problem of interest in our investigation.

2. NETWORK MODEL

2.1. Diffusion Strategy

At each time $i \geq 0$, each node k has access to a scalar measurement $\boldsymbol{d}_k(i) \in \mathbb{C}$ and a $1 \times M$ regression vector $\boldsymbol{u}_{k,i} \in \mathbb{C}^{1\times M}$. The measurements are assumed to be related via the linear regression model:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i} w^o + \boldsymbol{v}_k(i) \tag{1}$$

where $w^o \in \mathbb{C}^{M\times 1}$ is the target vector to be estimated and $\boldsymbol{v}_k(i) \in \mathbb{C}$ is measurement noise.

In adaptive networks, the nodes update their estimates of w^o by communicating with their neighbors. At every time instant i, every node k updates its estimate for w^o in a two-step diffusion process involving adaptation and combination. The adapt-then-combine (ATC) diffusion strategy is described as follows [13, 14]:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} + \mu_k \boldsymbol{u}_{k,i}^* [\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}\boldsymbol{w}_{k,i-1}] \tag{2}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k} \boldsymbol{\psi}_{\ell,i} \tag{3}$$

where the symbol $\mathcal{N}_k$ denotes the set of neighbors of node

This work was supported in part by NSF Grant CCF-1011918. Emails: {ckyuna, sayed}@ee.ucla.edu

k. In the first step (2), an intermediate estimate $\boldsymbol{\psi}_{k,i}$ is determined by adapting the existing estimate $\boldsymbol{w}_{k,i-1}$ using local data. The parameter μ_k is a positive step-size factor. The second step (3) uses non-negative coefficients $\{a_{\ell k}\}$ to combine the estimates from the neighbors. If we collect the coefficients $\{a_{\ell k}\}$ into an $N \times N$ matrix A, then A and its entries are required to satisfy:

$$A^T\mathbb{1} = \mathbb{1},\ a_{lk} \geq 0,\ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{4}$$

in which case A is a left-stochastic matrix (the entries on each of its columns add up to one).

2.2. Mean-Square Performance

Assume the noise process $\boldsymbol{v}_k(i)$ is temporally white and independent over space with variance $\sigma_{v,k}^2$. Assume likewise that the regression data $\boldsymbol{u}_{k,i}$ are also independent over space and temporally white with covariance matrix $R_{u,k} = \mathbb{E}\boldsymbol{u}_{k,i}^*\boldsymbol{u}_{k,i} > 0$. Introduce the weight-error vector $\tilde{\boldsymbol{w}}_{k,i} = w^o - \boldsymbol{w}_{k,i}$ and collect the error vectors from across the network into the column vector, $\tilde{\boldsymbol{w}}_i = \text{col}\{\tilde{\boldsymbol{w}}_{1,i}, \ldots, \tilde{\boldsymbol{w}}_{N,i}\}$. It can be shown that, under expectation and for sufficiently small step-sizes, it holds that [13–15]

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 = \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{B}^*\mathcal{B}} + \text{Tr}(\mathcal{Y}) \tag{5}$$

where

$$\mathcal{B} \triangleq \mathcal{A}^T(I - \mathcal{M}\mathcal{R}_u) \tag{6}$$

$$\mathcal{Y} \triangleq \mathcal{A}^T\mathcal{M}\mathcal{S}\mathcal{M}\mathcal{A} \tag{7}$$

$$\mathcal{R}_u \triangleq \text{diag}\{R_{u,1}, R_{u,2}, ..., R_{u,N}\} \tag{8}$$

$$\mathcal{S} \triangleq \text{diag}\{\sigma_{v,1}^2 R_{u,1}, \sigma_{v,2}^2 R_{u,2}, ..., \sigma_{v,N}^2 R_{u,N}\} \tag{9}$$

$$\mathcal{M} \triangleq \text{diag}\{\mu_1 I_M, \mu_2 I_M, ..., \mu_N I_M\} \tag{10}$$

$$\mathcal{A} \triangleq A \otimes I_M \tag{11}$$

The network mean-square-deviation (MSD) at time i is defined as

$$\xi(i) \triangleq \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\|\tilde{\boldsymbol{w}}_{k,i}\|^2 = \frac{1}{N}\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 \tag{12}$$

Relation (5) can be used to deduce that the steady-state MSD performance of the network is given by the following expression:

$$\text{MSD}_{\text{steady}}^{\text{network}} \triangleq \lim_{i\to\infty}\xi(i) = \frac{1}{N}\sum_{i=0}^{\infty}\text{Tr}(\mathcal{B}^i\mathcal{Y}\mathcal{B}^{*i}) \tag{13}$$

where the step-sizes $\{\mu_k\}$ are chosen small enough to ensure the stability of the matrix $\mathcal{B}$.

Several earlier studies focused on selecting the combination weights $\{a_{\ell k}\}$ in order to minimize the network MSD level given by (13) — see [4, 6]. These schemes help reduce the MSD level albeit at some degradation in convergence rate. In this work, we would like to select the weights in order to attain two objectives: minimize the MSD *and* maximize the convergence rate.

3. TRANSIENT PHASE

It can be deduced from (5) that the convergence rate of the diffusion strategy is determined by $[\rho(\mathcal{B})]^2$, in terms of the square of the spectral radius of $\mathcal{B}$. We can therefore consider the following optimization problem during the transient phase:

$$\min_{A}\ \rho(\mathcal{B}) \tag{14}$$
$$\text{subject to } A^T\mathbb{1} = \mathbb{1},\ a_{\ell k} \geq 0,\ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k$$

Finding the optimal A is generally a non-trivial problem. To proceed, we consider the *max* norm

$$\|A\|_{\text{max}} \triangleq \max_{1\leq \ell,k\leq N}|a_{\ell k}| \tag{15}$$

which is a generalized matrix norm without the submultiplicative property. We further define the norm $\|A\|_{\text{g-max}} \triangleq N \cdot \|A\|_{\text{max}}$, which can be verified to satisfy the submultiplicative property, i.e., $\|AB\| \leq \|A\| \cdot \|B\|$ [16]. Instead of minimizing $\rho(\mathcal{B})$, we shall minimize an upper bound on $\rho(\mathcal{B})$ given by:

$$\begin{aligned}\rho(\mathcal{B}) &= \rho\left[\mathcal{A}^T(I - \mathcal{M}\mathcal{R}_u)\right] \\ &\leq \|\mathcal{A}^T(I - \mathcal{M}\mathcal{R}_u)\|_{\text{g-max}} \\ &\leq \|\mathcal{A}\|_{\text{g-max}} \cdot \|I - \mathcal{M}\mathcal{R}_u\|_{\text{g-max}} \\ &= N \cdot \|\mathcal{A}\|_{\text{max}} \cdot \|I - \mathcal{M}\mathcal{R}_u\|_{\text{g-max}}\end{aligned} \tag{16}$$

Using the structure $\mathcal{A} \triangleq A \otimes I_M$, it holds that $\|\mathcal{A}\|_{\text{max}} = \|A\|_{\text{max}}$. Therefore, problem (14) is replaced by

$$\min_{A}\ \max_{1\leq l,k\leq N} a_{lk} \tag{17}$$
$$\text{subject to } A^T\mathbb{1} = \mathbb{1},\ a_{\ell k} \geq 0,\ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k$$

The solution to problem (17) is the uniform rule:

$$a_{\ell k} = \begin{cases} \frac{1}{|\mathcal{N}_k|}, & \text{if } l \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

which is one of the most widely discussed combination rules in the literature.

Result (18) suggests that during the transient phase, it is best for the nodes to employ uniform combination weights in order to speed up convergence (and, hence, speed up the diffusion of information through the network during the initial learning phase). The result is intuitively appealing. During the transient phase, the agents have not had sufficient time to accumulate enough information about the network and their neighbors to weight these neighbors differently. The most prudent action is therefore to assign equal weights to the neighbors initially.

4. STEADY-STATE PHASE

Although the static uniform combination rule (18) enhances the convergence speed during the initial transient phase, adaptive combination rules can provide superior steady-state mean-square performance in general [4,5,13,17,18]. We will consider the adaptive relative variance rule proposed in [6,14] since it provides good steady-state MSD:

$$\boldsymbol{a}_{\ell k}(i) = \begin{cases} \frac{\gamma_{\ell,k}^{-2}(i)}{\sum_{j\in\mathcal{N}_k}\gamma_{j,k}^{-2}(i)}, & \text{if } \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where the scalar parameter $\boldsymbol{\gamma}_{\ell,k}^2(i)$ is updated according to the following first-order smoothing model:

$$\boldsymbol{\gamma}_{\ell,k}^2(i) = (1-\nu_k)\boldsymbol{\gamma}_{\ell,k}^2(i-1) + \nu_k\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2. \tag{20}$$

where $0 < \nu_k \ll 1$ is a positive coefficient and $\boldsymbol{\gamma}_{\ell,k}^2(-1) = 0$ is set for all k and ℓ. We now need to explain how to select the switching time between the transient and steady-state phases.

5. PHASE TRANSITION

We develop a distributed procedure to implement the task of switching from the transient phase to the steady-state phase (i.e., for switching from the combination weights (18) to the combination weights (19) or other similar adaptive weights)). In the transient phase, the estimation error is generally much larger than the noise variance. Therefore, we can neglect the second term on the right-hand side of (5) and write for the transient phase:

$$\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 \approx \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{B}^*\mathcal{B}} = \mathbb{E}\left(\tilde{\boldsymbol{w}}_{i-1}^*\mathcal{B}^*\mathcal{B}\tilde{\boldsymbol{w}}_{i-1}\right) \tag{21}$$

Since $\mathcal{B}^*\mathcal{B}$ is Hermitian, we apply the Rayleigh-Ritz theorem [16] and obtain that

$$\lambda_{\min}(\mathcal{B}^*\mathcal{B})\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 \le \mathbb{E}\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\mathcal{B}^*\mathcal{B}} \le \rho(\mathcal{B}^*\mathcal{B})\mathbb{E}\|\tilde{\boldsymbol{w}}_i\|^2 \tag{22}$$

Therefore, during the initial learning phase, the network MSD is upper and lower bounded by exponentially decaying functions. Consequently, the evolution of the transient network MSD can be approximated by $\xi(i) \approx \kappa \cdot \xi(i-1)$, where κ is a positive constant less than one because of the condition $\rho(\mathcal{B}) < 1$ for mean-square convergence. At every time i, nodes can estimate the instantaneous network MSD using local data as follows:

$$\boldsymbol{\xi}_k(i) \triangleq \frac{1}{|\mathcal{N}_k|}\sum_{\ell\in\mathcal{N}_k}\|\boldsymbol{\psi}_{\ell,i} - \boldsymbol{w}_{k,i-1}\|^2 \approx \boldsymbol{\xi}(i) \tag{23}$$

Due to measurement and gradient noise, these locally observed network MSD values suffer from fluctuation noise. We model the fluctuation as a temporally independent and zero-mean additive white Gaussian noise, $\epsilon_k(i)$, with variance σ_ϵ^2. Therefore, if we consider a time period from $t = i-L+1$ to i, during the transient phase we can write

$$\boldsymbol{\xi}_k(t) = \alpha_k \cdot \kappa^{t-(i-L+1)} + \boldsymbol{\epsilon}_k(t) \tag{24}$$

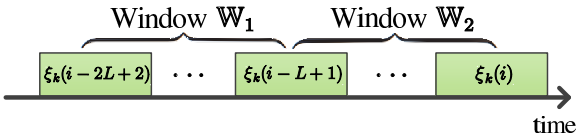

Fig. 1: Two sliding windows are used to detect the phase transition. Data from window $\mathbb{W}_1$ are used to estimate κ and data from window $\mathbb{W}_2$ are used to detect the transition from transient to steady-state operation.

and during the steady-state phase we have

$$\boldsymbol{\xi}_k(t) = \alpha_k + \boldsymbol{\epsilon}_k(t) \tag{25}$$

for some unknown value α_k at the start of the observation period. Let us define

$$\boldsymbol{x}_{k,i} \triangleq [\boldsymbol{\xi}_k(i-L+1), \cdots, \boldsymbol{\xi}_k(i)]^T \tag{26}$$

$$\boldsymbol{z}_{k,i} \triangleq [\boldsymbol{\epsilon}_k(i-L+1), \cdots, \boldsymbol{\epsilon}_k(i)]^T \tag{27}$$

$$s_\kappa \triangleq [1, \cdots, \kappa^{L-1}]^T \tag{28}$$

Then, the MSD window vector $\boldsymbol{x}_{k,i}$ can be modeled as follows during the transient and steady-state phases:

$$\text{transient}: \boldsymbol{x}_{k,i} = \alpha_k \cdot s_\kappa + \boldsymbol{z}_{k,i} \tag{29}$$

$$\text{steady-state}: \boldsymbol{x}_{k,i} = \alpha_k \cdot \mathbb{1}_L + \boldsymbol{z}_{k,i} \tag{30}$$

where $\mathbb{1}_L$ is a $L \times 1$ vector with all entries being one.

We propose a switching method that relies on the use of two overlapping sliding windows, illustrated in Figure 1, to estimate κ and to decide on the transition phase. Each window is of size L with indices $\mathbb{W}_1 = \{i-2L+2, ..., i-L+1\}$ and $\mathbb{W}_2 = \{i-L+1, ..., i\}$. The windows share the MSD value at time $i-L+1$. Then, $\boldsymbol{x}_{k,i-L+1}$ and $\boldsymbol{x}_{k,i}$ refer to the local MSD values within windows $\mathbb{W}_1$ and $\mathbb{W}_2$, respectively. We now describe the operation of the detection scheme for a generic node; we drop the node index k for simplicity. In the algorithm, the first window $\mathbb{W}_1$ is assumed to be at the transient phase and we use its MSD values to estimate κ. We can formulate this estimation problem as a nonlinear least-squares (LS) problem to minimize:

$$J(\kappa,\alpha) = (\boldsymbol{x}_{i-L+1} - \alpha\cdot s_\kappa)^T(\boldsymbol{x}_{i-L+1} - \alpha\cdot s_\kappa) \tag{31}$$

This model is linear in α and nonlinear in κ since s_κ includes exponents of κ. The solution can be obtained by sequentially estimating κ and α [19]. The value of α that minimizes (31) for a given κ is

$$\hat{\boldsymbol{\alpha}} = (s_\kappa^T s_\kappa)^{-1} s_\kappa^T \boldsymbol{x}_{i-L+1} \tag{32}$$

and the resulting LS error is

$$J(\kappa,\hat{\boldsymbol{\alpha}}) = \boldsymbol{x}_{i-L+1}^T[I - s_\kappa(s_\kappa^T s_\kappa)^{-1}s_\kappa^T]\boldsymbol{x}_{i-L+1} \tag{33}$$

Then, the optimal least-squares estimate (LSE) for κ is obtained by solving the following problem:

$$\hat{\boldsymbol{\kappa}}_{\text{LSE}} = \arg\max_\kappa \frac{(\boldsymbol{x}_{i-L+1}^T s_\kappa)^2}{s_\kappa^T s_\kappa} \tag{34}$$

The solution of (34) generally generally lacks a closed-form expression due to the nonlinear exponents of κ in s_κ, and requires numerical computing methods such as the Newton-Raphson iteration [20]. We provide an alternative sub-optimal estimate to reduce the computational complexity. One reasonable way to approximate κ is to use the ratio of two consecutive MSD values, as suggested by (24). Therefore, we can average this ratio through time to obtain a sub-optimal estimate for κ:

$$\hat{\boldsymbol{\kappa}} \triangleq \frac{1}{L-1} \sum_{i_1=i-2L+2}^{i-L+1} \frac{\boldsymbol{\xi}(i_1)}{\boldsymbol{\xi}(i_1-1)} \tag{35}$$

The MSD values in the second window, $\mathbb{W}_2$, are used for hypothesis testing to detect whether we are operating in the transient phase $\mathcal{H}_0$ or the steady-state phase $\mathcal{H}_1$, namely,

$$\begin{aligned} \mathcal{H}_0: \quad & \boldsymbol{x}_i = \alpha_0 \cdot s_\kappa + \boldsymbol{z}_i \\ \mathcal{H}_1: \quad & \boldsymbol{x}_i = \alpha_1 \cdot \mathbb{1}_L + \boldsymbol{z}_i \end{aligned} \tag{36}$$

We can utilize the generalized likelihood ratio test, which estimates unknown parameters under each hypothesis and then decides on the hypotheses. The unknown parameter κ of s_κ is estimated in (35). Therefore, for the transient phase $\mathcal{H}_0$, the estimate for α_0, denoted by $\hat{\alpha}_0$, is given in (32) as

$$\hat{\boldsymbol{\alpha}}_0 = (s_\kappa^T s_\kappa)^{-1} s_\kappa^T \boldsymbol{x}_i \tag{37}$$

Similarly, the estimate for α_1 for the steady-state phase $\mathcal{H}_1$ is obtained by replacing s_κ by $\mathbb{1}_L$:

$$\hat{\boldsymbol{\alpha}}_1 = (\mathbb{1}_L^T \mathbb{1}_L)^{-1} \mathbb{1}_L^T \boldsymbol{x}_i \tag{38}$$

When detecting the hypotheses, we assume equal prior probabilities and seek to maximize the likelihood function where we decide for $\mathcal{H}_0$ if

$$p(\boldsymbol{x}_i|\mathcal{H}_0) > p(\boldsymbol{x}_i|\mathcal{H}_1) \tag{39}$$

and decide for $\mathcal{H}_1$ if

$$p(\boldsymbol{x}_i|\mathcal{H}_0) \leq p(\boldsymbol{x}_i|\mathcal{H}_1) \tag{40}$$

Since the noise process $\boldsymbol{\epsilon}_k(i)$ is assumed to be a zero-mean and temporally independent Gaussian process, $p(\boldsymbol{x}_i|\mathcal{H}_0)$ and $p(\boldsymbol{x}_i|\mathcal{H}_1)$ can be expressed as

$$p(\boldsymbol{x}_i|\mathcal{H}_0) = \frac{1}{(2\pi\sigma_\epsilon^2)^{L/2}} \cdot e^{-\frac{1}{2\sigma_\epsilon^2}\|\boldsymbol{x}_i - \hat{\boldsymbol{\alpha}}_0 s_\kappa\|^2} \tag{41}$$

and

$$p(\boldsymbol{x}_i|\mathcal{H}_1) = \frac{1}{(2\pi\sigma_\epsilon^2)^{L/2}} \cdot e^{-\frac{1}{2\sigma_\epsilon^2}\|\boldsymbol{x}_i - \hat{\boldsymbol{\alpha}}_1 \mathbb{1}_L\|^2} \tag{42}$$

Therefore, each node k makes a decision based on the following decision rule

$$\|\boldsymbol{x}_i - \hat{\boldsymbol{\alpha}}_0 s_\kappa\|^2 \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \|\boldsymbol{x}_i - \hat{\boldsymbol{\alpha}}_1 \mathbb{1}_L\|^2 \tag{43}$$

Once node k detects that steady-state is reached, it employs adaptive methods to adjust the combination weights, such as the one given by (19).

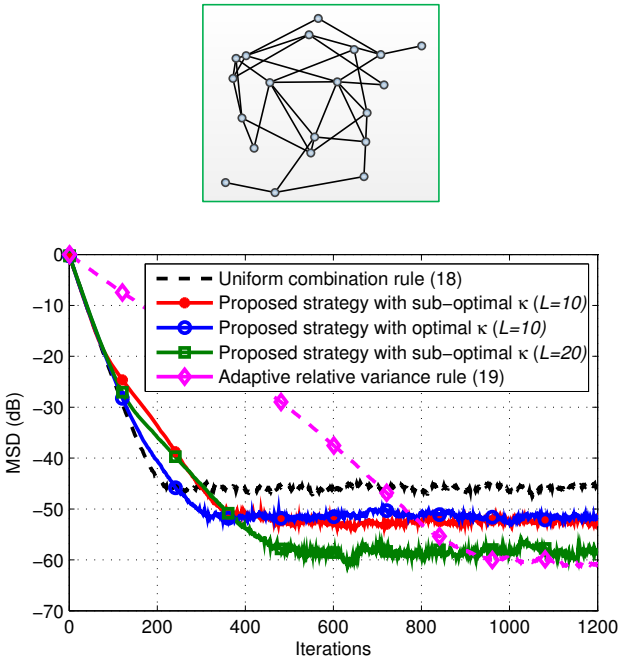

Fig. 2: Topology and performance comparison in terms of network MSD with $\mu = 0.001$.

6. SIMULATION RESULTS

In this section, we simulate the proposed strategy, and compare it with the uniform combination rule and the adaptive relative variance rule. The network has $N = 20$ nodes and its topology is shown in Figure 2. The length of w^o is $M = 8$ and we randomly choose its entries and normalize to $\|w^o\| = 1$. The regressor $\{\boldsymbol{u}_{k,i}\}$ is zero-mean and $R_{u,k}$ is diagonal with entries uniformly generated between $[25, 40]$. The background noise $\boldsymbol{v}_k(i)$ is temporally white and spatially independent Gaussian distributed with zero-mean and $\sigma_{v,k}^2$ uniformly selected between $[-30, 0]$ (dB). We assume a uniform step-size $\mu = 0.001$ for all nodes. When using the adaptive relative variance rule in (19), we set $\nu_k = 0.2$. The network MSD performance is simulated in Figure 2. It is observed that our proposed strategies achieve faster convergence rate than the adaptive relative variance rule and nearly minimal steady-state MSD. It should be noted that a larger window size L gives a lower detection error and thus lower steady-state MSD in general. Furthermore, the sub-optimal estimate of κ in (35) with lower complexity provides good performance, compared with the optimal LSE estimate in (34).

7. CONCLUSION

We proposed an operating strategy to adapt the combination weights over adaptive networks. The proposed procedure can achieve both optimal transient and steady-state network performance.

8. REFERENCES

[1] 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. on Decision and Control and European Control Conf.*, Seville, Spain, Dec. 2005, pp. 2996–3000.

[2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2003.

[3] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. Information Processing Sensor Networks (IPSN)*, Berkeley, CA, Apr. 2004, pp. 11–19.

[4] 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, Sept. 2010.

[5] X. Zhao and A. H. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5107–5124, Oct. 2012.

[6] S-Y. Tu and A. H. Sayed, "Optimal combination rules for adaptation and learning over networks," in *Proc. IEEE CAMSAP*, San Juan, Puerto Rico, Dec. 2011, pp. 317–320.

[7] 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, Dec. 2012.

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

[9] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, "Accelerated distributed average consensus via localized node state prediction," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1563–1576, Apr. 2009.

[10] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[11] M. Zhu and S. Martnez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[12] V. Schwarz and G. Matz, "Nonlinear average consensus based on weight morphing," in *Proc. IEEE ICASSP*, Kyoto, Japan, Mar. 2012, pp. 3129–3132.

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

[14] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing,* R. Chellapa and S. Theodoridis, Eds., Elsevier, 2013. Also available online as manuscript arXiv:1205.4220v1 [cs.MA], May 2012.

[15] 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, July 2008.

[16] R. Horn and C. R. Johnson, *Matrix Analysis*, New York: Cambridge Univ. Press, 1990.

[17] 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, July 2012.

[18] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Transactions on Fundamentals of Electronics,Communications and Computer Sciences*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.

[19] S. M. Kay, *Fundamentals of Statistical Signal Processing:Estimation Theory*, vol. I, Prentice-Hall, NJ, 1993.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, 2004.