sensors

MDPI

Article

Coordinate-Descent Adaptation over Hamiltonian Multi-Agent Networks

Azam Khalili [1], Vahid Vahidpour [1], Amir Rastegarnia [1], Ali Farzamnia [2,*], Kenneth Teo Tze Kin [2] and Saeid Sanei [3]

[1] Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran; matiin2441@yahoo.com (A.K.); v.vahidpour@gmail.com (V.V.); amir_rastegar2003@yahoo.com (A.R.)
[2] Faculty of Engineering, Universiti Malaysia Sabah, Kota Kinabalu 88400, Malaysia; kenteo@ums.edu.my
[3] Science and Technology, Nottingham Trent University, Clifton Lane, Nottingham NG11 8NS, UK; saeid.sanei@ntu.ac.uk
* Correspondence: alifarzamnia@ums.edu.my

Abstract: The incremental least-mean-square (ILMS) algorithm is a useful method to perform distributed adaptation and learning in Hamiltonian networks. To implement the ILMS algorithm, each node needs to receive the local estimate of the previous node on the cycle path to update its own local estimate. However, in some practical situations, perfect data exchange may not be possible among the nodes. In this paper, we develop a new version of ILMS algorithm, wherein in its adaptation step, only a random subset of the coordinates of update vector is available. We draw a comparison between the proposed coordinate-descent incremental LMS (CD-ILMS) algorithm and the ILMS algorithm in terms of convergence rate and computational complexity. Employing the energy conservation relation approach, we derive closed-form expressions to describe the learning curves in terms of excess mean-square-error (EMSE) and mean-square deviation (MSD). We show that, the CD-ILMS algorithm has the same steady-state error performance compared with the ILMS algorithm. However, the CD-ILMS algorithm has a faster convergence rate. Numerical examples are given to verify the efficiency of the CD-ILMS algorithm and the accuracy of theoretical analysis.

Keywords: adaptive estimation; coordinate-descent; distributed networks; incremental algorithm

check for updates

Citation: Khalili, A.; Vahidpour, V.; Rastegarnia, A.; Farzamnia, A.; Teo Tze Kin, K.; Sanei, S. Coordinate-Descent Adaptation over Hamiltonian Multi-Agent Networks. *Sensors* **2021**, *21*, 7732. https://doi.org/10.3390/s21227732

Academic Editor: Charalampos Konstantopoulos

Received: 10 September 2021
Accepted: 12 November 2021
Published: 20 November 2021

1. Introduction

Distributed processing over networks refers to the information extraction from streaming data collected at a group of spatially-dispersed and interconnected nodes. Distributed optimization problems of this kind appear in many applications, such as wireless sensor networks [1–3], multi-robot systems [4,5], smart grid programs [6–9], statistical learning over netwoks [10–12], and clustering [13,14]. Several classes of distributed optimization and estimation algorithms over multi-agent networks have been introduced in the literature. The consensus methods [15–18], incremental adaptive strategies [19–24], and diffusion networks [25–29] are among the most notable approaches.

In typical consensus strategies, each individual node collects noisy measurements over a period of time and performs a local processing task (e.g., calculates a local estimate). Then, the nodes collaborate through several iterations and share their information to achieve agreement. In the incremental strategies, a Hamilton cycle is established through the nodes where each node receives data from previous node on cycle path and sends the updated data to the next node. Such a mode of cooperation decreases the inter-node communication across the network and modifies the network independence. In diffusion strategies, information is spread through the network simultaneously and locally at all nodes and the task of distributed processing is performed. As in this mode of cooperation each node collaborates with its connected neighbors, the communication burden is higher than that of an incremental based strategy.

In this paper, we propose an incremental-based algorithm for solving the distributed estimation problem over a Hamilton network. One of the primary methods for solving this problem is the incremental least-mean-square (ILMS) algorithm reported in [19]. In this algorithm, the local calculated estimates are updated at each node by employing the LMS adaptation rule, without requiring any prior information about the data statistics. In order to implement the ILMS algorithm, it is assumed that at each time instant, the adaptation step has access to all entries of the approximate gradient vector—see expression (3) further ahead. In some practical situations (e.g., limited communication resources or missing data due to imperfect communication links) only a fraction of the approximate gradient vector is available for adaption. Thus, developing a new version of ILMS algorithm for networks with mentioned scenarios is of practical importance.

So far, different algorithms have been reported in the literature which solve the problem of distributed learning with partial information. In [30–32], a Bernoulli variable is used to model the step-size parameter, so that when the step-size is zero, not all entries of intermediate estimates are updated and the adaptation step is skipped for some. In [33–38], a different notion of partial diffusion of information is employed which relies on the exchange of a subset of entries of the weight estimate vectors themselves. In other words, it is assumed that each node shares only a fraction of its local information with its neighbors. Likewise, in [39], a situation is assumed in which some entries in the regression vector are missing at random due to incomplete information or censoring. In order to undo these changes, authors in [39] proposed an estimation scheme to estimate the underlying model parameters when the data are missing. Some other criteria have been proposed in the literature to motivate partial updating scheme. For instance, in [40], the periodic and sequential partial LMS updates are proposed, where the former scheme updates all filter coefficients every P-th iteration, with $P > 1$, instead of every iteration. The later updates only a portion of coefficients, which are predetermined, at each iteration. In [41], the stochastic partial LMS algorithm is proposed that is a randomized version of sequential scheme where the coefficient subsets are selected in random instead of deterministic fashion. The works [42,43] utilize the concept of set-membership filtering to partially update the weight vector, where the updates occur only when the innovation obtained from the data exceeds a predetermined threshold. The works [12,44,45] employ Krylov subspace concept to partially update the weight vector that is based on dimensionality reduction policies. Some other techniques based on energy considerations limit the updates, e.g., [46]. In [47], a reduced complexity augmented complex least-mean-square algorithm is proposed by employing partial-update for improper complex signals. The algorithm involves selection of a fraction of coefficients at every iteration. In [48], the authors focus on diffusion learning mechanisms and study its mean-square error performance under a generalized coordinate-descent scheme where the adaptation step by each node involves only a random subset of the coordinate-descent gradient vector. None of the aforementioned works; however, rely on the adaptive incremental strategy. To address this issue, we combine two techniques (incremental mode of cooperation and coordinate-descent updates) to consolidate a new method, termed coordinate-descent ILMS (CD-ILMS), for distributed estimation problem.

The main contributions of this manuscript can be stated as follows:

(a) A CD-ILMS algorithm is proposed to simulate the situation where some entries in approximate gradient vectors are missing or to control the computational burden of the update estimate purposely;

   1. We examine the effect of random partial gradient information on the learning performance and convergence rate of ILMS algorithm for MSE cost function;
   2. A theoretical analysis of the performance of CD-ILMS algorithm is concluded under some typical simplifying assumptions and approximations, typical in adaptive systems and tend to achieve a performance level that matches well with practice;

3. Stability conditions for CD-ILMS algorithm are derived both in mean and mean-square senses under certain statistical conditions. We find the necessary condition on step-size μ to guarantee the stability of CD-ILMS algorithms;
4. Employing the energy conservation paradigm, we derive closed-form expressions to describe the learning curves in terms of excess mean-square-error (EMSE) and mean-square-deviation (MSD);
5. We compare the CD-ILMS algorithm and regular full-update ILMS algorithm in terms of convergence rate and computational complexity. We find that although the CD-ILMS algorithm convergence to steady-state region takes longer, the overall computational complexity, i.e., savings in computation per iteration, does remain invariant.

The paper layout is organized as follows: In Section 2, we briefly explain the ILMS algorithm and formulate the CD-ILMS algorithm. In Section 3.1, we describe the system model and assumptions. The stability and performance analysis of CD-ILMS algorithm are analyzed both in mean and mean-square sense in Sections 3.2 and 3.3, respectively. Sections 3.4 and 3.5 investigate the transient and steady-state performance of CD-ILMS algorithm, respectively. In Section 4, we draw a comparison between ILMS and CD-ILMS algorithms on the basis of convergence rate and computational complexity. Performance evaluations are illustrated in Section 5. The paper is finally concluded in Section 6.

Notation: Throughout the paper, we adopt normal lowercase letters for scalars, bold lowercase letters for column vectors and bold uppercase letters for matrices, while $\mathbf{I}$ denotes an identity matrix of appropriate size. Superscript $(.)^{\mathsf{T}}$ denotes transposition for real-valued vectors and matrices. The symbol $\mathbb{E}[.]$ is the expectation operator, $\mathsf{tr}\{.\}$ represents the trace of its matrix argument, $\mathsf{vec}\{.\}$ vectorizes a matrix by stacking its columns on top of each other, and $\otimes$ is the standard Kronecker product.

2. Algorithm Description

We consider a network with K nodes with incremental cooperation topology, as shown in Figure 1. The network is used to estimate the parameter $\boldsymbol{w}^o \in \mathbb{R}^{\mathrm{L}\times 1}$, that minimize the following aggregate global cost function:

$$\min_{\boldsymbol{w}} f^{\mathrm{g}}(\boldsymbol{w}) \triangleq \sum_{k=1}^{\mathrm{K}} f_k(\boldsymbol{w}) \tag{1}$$

where it is assumed that each individual cost function, $f_k(\boldsymbol{w}) : \mathbb{R}^{\mathrm{L}\times 1} \to \mathbb{R}$ is ν-strongly convex and its gradients satisfy the δ-Lipschitz condition [25]. These conditions are equivalent to requiring $f_k(\boldsymbol{w})$ to be twice-differentiable and Hessian matrices of the individual costs, $\nabla^2_{\boldsymbol{w}} f_k(\boldsymbol{w}) \in \mathbb{R}^{\mathrm{L}\times\mathrm{L}}$, to be bounded as follows

$$0 < \nu \mathbf{I}_{\mathrm{L}} \leq \nabla^2_{w} f_k(w) \leq \delta \mathbf{I}_{\mathrm{L}} \tag{2}$$

for some positive parameters $\{\nu, \delta\}$, where $\nu < \delta$. Some cases of interest, e.g., Log-Loss regression or MSE [25,26], automatically satisfy the condition in (2).

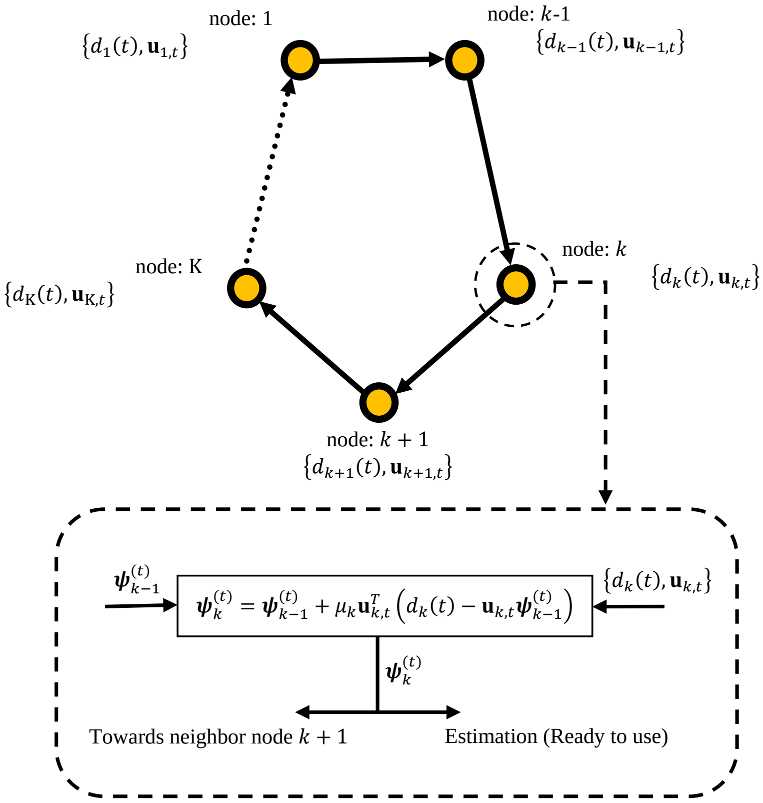
node: 1
$\{d_1(t), \mathbf{u}_{1,t}\}$
node: k-1
$\{d_{k-1}(t), \mathbf{u}_{k-1,t}\}$
node: K
$\{d_\mathrm{K}(t), \mathbf{u}_{\mathrm{K},t}\}$
node: k
$\{d_k(t), \mathbf{u}_{k,t}\}$
node: $k+1$
$\{d_{k+1}(t), \mathbf{u}_{k+1,t}\}$
$\boldsymbol{\psi}_{k-1}^{(t)}$
$\boldsymbol{\psi}_k^{(t)} = \boldsymbol{\psi}_{k-1}^{(t)} + \mu_k \mathbf{u}_{k,t}^T \left(d_k(t) - \mathbf{u}_{k,t}\boldsymbol{\psi}_{k-1}^{(t)}\right)$
$\{d_k(t), \mathbf{u}_{k,t}\}$
$\boldsymbol{\psi}_k^{(t)}$
Towards neighbor node $k+1$
Estimation (Ready to use)

Figure 1. Data processing in an incremental network.

2.1. Incremental Steepest-Descent Solution

By applying the steepest-descent method in conjunction with the incremental strategy to (1) gives the Incremental Steepest-Descent Solution as lists in Algorithm 1. Observe that in this implementation, each node k needs to receive the calculated estimate of w^o at node $k-1$ (which is denoted by $\boldsymbol{\psi}_{k-1}^{(t)}$). In (3), w_t denotes a global estimation of w^o at time instant t. Moreover, since the true gradient vectors, $\{\nabla_{w^\top} f_k(\cdot)\}$, are not available beforehand in many practical applications, they have been replaced by approximates, $\{\widehat{\nabla_{w^\top} f_k(\cdot)}\}$.

Algorithm 1. Incremental Strategy [19].

```
**Initialization:** start with $w_0$ = initial condition.
**for** every time $t \geq 1$ **do**
    set $\boldsymbol{\psi}_0^{(t)} = w_{t-1}$
    **for** nodes $k = 1, \ldots, \mathrm{K}$ **do**
        receive $\boldsymbol{\psi}_{k-1}^{(t)}$ from node $k-1$
            $$\boldsymbol{\psi}_k^{(t)} = \boldsymbol{\psi}_{k-1}^{(t)} - \mu_k \widehat{\nabla_{w^\top} f_k}(\boldsymbol{\psi}_{k-1}^{(t)})$$
    **end for**
    set $w_t = \boldsymbol{\psi}_\mathrm{K}^{(t)}$
**end for**
**End**
```

2.2. Coordinate-Descent Incremental Adaptation

Consider a case in which, due to some practical issues (e.g., missing data or limited computational and communication burden) only a subset of the approximate gradient vector can be updated as follows:

$$\boldsymbol{\psi}_k^{(t)} = \boldsymbol{\psi}_{k-1}^{(t)} - \mu_k \widehat{\nabla_{\boldsymbol{w}^\mathsf{T}} f_k}(\boldsymbol{\psi}_{k-1}^{(t)}) \tag{3}$$

In order to handle such a situation, first a suitable model should be adopted. To this end, we follow a similar approach to [48] and define a diagonal random matrix $\boldsymbol{\Pi}_{k,t} \in \mathbb{R}^{\mathrm{L}\times\mathrm{L}}$ of node k at iteration t as

$$\boldsymbol{\Pi}_{k,t} \triangleq \mathrm{diag}\{\pi_{k,t}(1), \pi_{k,t}(2), \ldots, \pi_{k,t}(\mathrm{L})\} \tag{4}$$

where $\{\pi_{k,t}(l)\}$ are some Bernoulli random variables, that the randomness in the update varies across space, k, and over time, t. We have $\pi_{k,t}(l) = 0$ or $\pi_{k,t}(l) = 1$ with the following probability

$$\mathrm{Prob}(\pi_{k,t}(l) = 0) = \pi_k; \quad 0 \leq \pi_k < 1 \tag{5}$$

where $\pi_{k,t}(l) = 0$ means that l-th entry of $\widehat{\nabla_{\boldsymbol{w}^\mathsf{T}} f_k}(\boldsymbol{\psi}_{k-1}^{(t)})$ is missing and l-th entry of $\boldsymbol{\psi}_{k,t}$ in (3) is not updated. Multiplying the approximate gradient vector by $\boldsymbol{\Pi}_{k,t}$, we replace $\widehat{\nabla_{\boldsymbol{w}^\mathsf{T}} f_k}\left(\boldsymbol{\psi}_{k-1}^{(t)}\right)$ by $\widehat{\nabla_{\boldsymbol{w}^\mathsf{T}} f_k}^{\mathrm{miss}}\left(\boldsymbol{\psi}_{k-1}^{(t)}\right)$ as

$$\widehat{\nabla_{\boldsymbol{w}^\mathsf{T}} f_k}^{\mathrm{miss}}\left(\boldsymbol{\psi}_{k-1}^{(t)}\right) = \boldsymbol{\Pi}_{k,t} \times \widehat{\nabla_{\boldsymbol{w}^\mathsf{T}} f_k}\left(\boldsymbol{\psi}_{k-1}^{(t)}\right) \tag{6}$$

3. Performance Analysis

3.1. Data Model and Assumptions

Consider MSE networks where at each time instant t, each node k is assumed to observe a scalar measurement $d_k(t) \in \mathbb{R}$ and a $1 \times \mathrm{L}$ row regression vector $\mathbf{u}_{k,t} \in \mathbb{R}^{1\times\mathrm{L}}$. A linear regression model is used to describe the collected data at each individual node as follows:

$$d_k(t) = \mathbf{u}_{k,t}\boldsymbol{w}^{\mathrm{o}} + v_k(t) \tag{7}$$

where $v_k(t)$ is the noise process. The individual cost function, $f_k(\boldsymbol{w})$, that is associated with each node k is the MSE (quadratic) cost

$$f_k(\boldsymbol{w}) = \mathsf{E}\left[\left|d_k(t) - \mathbf{u}_{k,t}\boldsymbol{w}\right|^2\right] \tag{8}$$

To proceed with analysis, we introduce some notation and assumptions. In our analysis, we assume that:

Assumption 1. *(i)* *For all nodes k, and $t \geq 1$, the measurement noises $\{v_k(t)\}$ are all zero-mean, white, and independent from the input and desired signals, with variances $\sigma_{v,k}^2$;*

1. *The regression data $\mathbf{u}_{k,t}$ for all nodes k, and all observation times $t \geq 1$, are zero-mean, white overt time and space with*

$$\mathsf{E}\left[\mathbf{u}_{k,t}^\mathsf{T}\mathbf{u}_{\ell,t}\right] \triangleq \mathbf{R}_{u,k}\delta_{k\ell}\delta_{st}$$

where $0 < \mathbf{R}_{u,k} \in \mathbb{R}^{\mathrm{L}\times\mathrm{L}}$, and $\delta_{k\ell}$ denotes the Kronecker delta, i.e., it is equal to one when $k = \ell$ and zero otherwise;

2. *The step-sizes, μ_k, are small enough so as to ignore the quadratic term in μ_k.*

Let the notation $\mathcal{K}_{t-1}$ represent the available information about the random processes $\boldsymbol{\psi}_k^{(\tau)}$ and $\boldsymbol{\Pi}_{k,\tau}$ at all agents $k = 1, 2, \ldots, \mathrm{K}$ for $\tau \leq t-1$:

$$\mathcal{K}_{t-1} \triangleq \text{filtration}\left\{\boldsymbol{\psi}_k^{(\tau)}, \boldsymbol{\Pi}_{k,\tau}\right\} \tag{9}$$

Assumption 2. *For all k, ℓ, m, n, the indicator variables $\pi_{k,t}(m)$ and $\pi_{k,t}(n)$ are assumed to be independent of each other. Moreover, the random variables $\pi_{k,t}(m)$ and $\mathcal{K}_{t-1}$ are independent of each other, for any iteration $w \in \mathcal{K}_{t-1}$ and for all nodes k.*

Based on the adaptive implementation of incremental strategy [19], the adaptive distributed coordinate-descent incremental algorithm or distributed coordinate-descent incremental LMS (CD-ILMS) algorithm can be summarized in Algorithm 2. In this algorithm, each node updates the local estimate as

$$\boldsymbol{\psi}_k^{(t)} = \boldsymbol{\psi}_{k-1}^{(t)} + \mu_k \boldsymbol{\Pi}_{k,t} \mathbf{u}_{k,t}^{\mathsf{T}} \left(d_k(t) - \mathbf{u}_{k,t} \boldsymbol{\psi}_{k-1}^{(t)}\right) \tag{10}$$

Algorithm 2. Coordinate-Descent Incremental LMS (CD-ILMS) Strategy.

```
Initialization: start with $w_{-1}$ = initial condition.
for every time $t \geq 0$ do
    set the fictitious boundary condition at $\boldsymbol{\psi}_0^{(t)} \leftarrow w_{t-1}$.
    for nodes $k = 1, \ldots, \mathrm{K}$ do
        node $k$ receives $\boldsymbol{\psi}_{k-1}^{(t)}$ from its preceding neighbor
        $k-1$,
        node $k$ performs:
            $\boldsymbol{\psi}_k^{(t)} = \boldsymbol{\psi}_{k-1}^{(t)} + \mu_k \boldsymbol{\Pi}_{k,t} \mathbf{u}_{k,t}^{\mathsf{T}} \left(d_k(t) - \mathbf{u}_{k,t} \boldsymbol{\psi}_{k-1}^{(t)}\right)$
    end for
    set $w_t \leftarrow \boldsymbol{\psi}_{\mathrm{K}}^{(t)}$
end for
End
```

3.2. Mean Stability Analysis

Now, the mean behavior of the proposed algorithm is analyzed. More specifically, we seek conditions that for sufficiently large t and for all k, the proposed algorithm is asymptotically unbiased. To proceed, the following local error signals are defined:

$$\tilde{\boldsymbol{\psi}}_k^{(t)} \triangleq w^{\mathrm{o}} - \boldsymbol{\psi}_k^{(t)}, \quad \tilde{w}_t \triangleq w^{\mathrm{o}} - w_t$$

By replacing d_k from (7) into (10) and subtracting w^o from both sides, gives

$$\tilde{\boldsymbol{\psi}}_k^{(t)} = \left(\mathbf{I}_{\mathrm{L}} - \mu_k \boldsymbol{\Pi}_{k,t} \mathbf{u}_{k,t}^{\mathsf{T}} \mathbf{u}_{k,t}\right) \tilde{\boldsymbol{\psi}}_{k-1}^{(t)} - \mu_k \boldsymbol{\Pi}_{k,t} \mathbf{u}_{k,t}^{\mathsf{T}} v_k(t) \tag{11}$$

Taking statistical expectations of both sides of (11) together with using the items on Assumption 1, we obtain

$$\mathsf{E}\left[\tilde{\boldsymbol{\psi}}_k^{(t)}\right] = (\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_k)\mathbf{R}_{u,k}) \mathsf{E}\left[\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right] \tag{12}$$

Through iterating recursion (12) over t and setting $k = \mathrm{K}$, we deduce that $\mathsf{E}\Big[\tilde{\boldsymbol{\psi}}_k^{(t)}\Big]$ evolves according to

$$\begin{aligned}\mathsf{E}[\tilde{\boldsymbol{w}}_t] &= \left[\prod_{k=1}^{\mathrm{K}}(\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_k)\mathbf{R}_{u,k})\right]\mathsf{E}[\tilde{\boldsymbol{w}}_{t-1}] \\ &= \left[\prod_{k=1}^{\mathrm{K}}\mathbf{Q}_k\right]\mathsf{E}[\tilde{\boldsymbol{w}}_{t-1}] = \mathbf{Q}\mathsf{E}[\tilde{\boldsymbol{w}}_{t-1}] \end{aligned} \tag{13}$$

where $\mathbf{Q}_k \triangleq (\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_k)\mathbf{R}_{u,k})$. Clearly, the mean weight-error vector for the CD-ILMS depends on the spectral radius of $\mathbf{Q}$. The following proposition summarizes the required mean stability condition for the proposed algorithm.

Proposition 1 (Mean Stability). *Assume the data model (7) and Assumption 1 hold. Then, the CD-ILMS algorithm is asymptotically unbiased if, and only if, the step-size parameters $\{\mu_k\}$ satisfy the following condition:*

$$0 < \mu_k < \frac{2}{(1-\pi_k)\lambda_{\max}\{\mathbf{R}_{u,k}\}} \tag{14}$$

where $\lambda_{\max}$ denotes the largest eigenvalue of its argument.

Proof. The asymptotic unbiasedness of the CD-ILMS algorithm is guaranteed if, and only if, the matrix $\mathbf{Q}$ be stable, or equivalently, all its eigenvalues lie inside the unit disc, namely,

$$\rho(\mathbf{Q}) < 1 \tag{15}$$

As the spectral radius of a matrix is always smaller than any induced norm of the same matrix [49] we have

$$\begin{aligned}\rho(\mathbf{Q}) \le \|\mathbf{Q}\| &\le \|\mathbf{Q}_1\|\|\mathbf{Q}_2\|\dots\|\mathbf{Q}_{\mathrm{K}}\| \\ &\overset{*}{=} \rho(\mathbf{Q}_1)\rho(\mathbf{Q}_2)\dots\rho(\mathbf{Q}_{\mathrm{K}})\end{aligned} \tag{16}$$

where step $(*)$ is because every $\mathbf{Q}_k$ is a symmetric matrix. This means that its spectral radius agrees with its 2-induced norm, so that $(*)$ is justified. Accordingly, to guarantee the constraint (15) for all k, it is enough to have $\rho(\mathbf{Q}_k) \le 1$, which is equivalent to

$$\big|1 - \mu_k(1-\pi_k)\lambda_{\max}\{\mathbf{R}_{u,k}\}\big| < 1 \quad \forall\, k. \tag{17}$$

Thus, the conclusion in (14) is verified. □

3.3. Mean-Square Performance

In this section, the mean-square performance of the error recursion (11) is examined. We start by equating the squared-weighted Euclidean norms of both sides of (11) and taking the expectations together with employing Assumption 1. After a straightforward calculation, we find the following weighted-variance relation:

$$\begin{aligned}\mathsf{E}\Big[\Big\|\tilde{\boldsymbol{\psi}}_k^{(t)}\Big\|_{\boldsymbol{\Sigma}}^2\Big] &= \mathsf{E}\Big[\Big\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\Big\|_{\boldsymbol{\Sigma}'}^2\Big] \\ &+ \mu_k^2\sigma_{v,k}^2\mathsf{E}\Big[\mathbf{u}_{k,t}\boldsymbol{\Pi}_{k,t}\boldsymbol{\Sigma}\boldsymbol{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\Big]\end{aligned} \tag{18}$$

$$\boldsymbol{\Sigma}' = \Big(\mathbf{I}_{\mathrm{L}} - \mu_k\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\boldsymbol{\Pi}_{k,t}\Big)\boldsymbol{\Sigma}\Big(\mathbf{I}_{\mathrm{L}} - \mu_k\boldsymbol{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\Big) \tag{19}$$

for any arbitrary positive-definite weighting matrix $\mathbf{\Sigma}$. Due to the assumption on $v_k(t)$, the expectations of the cross-terms involving $v_k(t)$ evaluate to zero. Note that items in Assumption 1 guarantee that $\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}$ is independent of $\mathbf{\Sigma}'$ and $\mathbf{u}_{k,t}$. Thus, the expectation $\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{\Sigma}'}^2\right]$ can be rewritten as

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{\Sigma}'}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathsf{E}[\mathbf{\Sigma}']}^2\right] \tag{20}$$

By defining

$$\boldsymbol{\sigma} \triangleq \mathsf{vec}\{\mathbf{\Sigma}\} \tag{21}$$

$$\boldsymbol{\sigma}' \triangleq \mathsf{vec}\{\mathsf{E}[\mathbf{\Sigma}']\} \tag{22}$$

and employing (20)–(22), we can modify (18) to

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k}^{(t)}\right\|_{\boldsymbol{\sigma}}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\boldsymbol{\sigma}'}^2\right] + \mu_k^2\sigma_{v,k}^2\mathsf{E}\left[\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\mathbf{\Sigma}\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\right] \tag{23}$$

where $\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k}^{(t)}\right\|_{\boldsymbol{\sigma}}^2\right]$ and $\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\boldsymbol{\sigma}'}^2\right]$ refer to the same quantities as $\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{\Sigma}}^2\right]$ and $\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{\Sigma}'}^2\right]$, respectively.

Taking the vectorization operator of both sides of (19), and using (21), (22), Assumptions 1 and 2, together with employing the relationship between the Kronecker product and the vectorization operator (For any matrices $\{\mathbf{A}, \mathbf{\Sigma}, \mathbf{B}\}$ of compatible dimensions we have $\mathsf{vec}\{\mathbf{A}\mathbf{\Sigma}\mathbf{B}\} = (\mathbf{B}^{\mathsf{T}} \otimes \mathbf{A})\mathsf{vec}\{\mathbf{\Sigma}\}$) we find that the weighting vectors $\{\boldsymbol{\sigma}, \boldsymbol{\sigma}'\}$ satisfy the following relation

$$\boldsymbol{\sigma}' = \mathbf{F}_k\boldsymbol{\sigma} \tag{24}$$

where $\mathbf{F}_k \in \mathbb{R}^{\mathrm{L}^2\times\mathrm{L}^2}$ and given by

$$\mathbf{F}_k \triangleq \mathsf{E}\left[\left(\mathbf{I}_{\mathrm{L}} - \mu_k\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\right) \otimes \left(\mathbf{I}_{\mathrm{L}} - \mu_k\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\right)\right] \tag{25}$$

Note that the vectorization commutes through expectation. Considering Assumption 1, we can approximate $\mathbf{F}_k$ as follows

$$\mathbf{F}_k \approx \left[(\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_k)\mathbf{R}_{u,k}) \otimes \left(\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_k)\mathbf{R}_{u,k}^{\mathsf{T}}\right)\right] \tag{26}$$

To evaluate $\mathsf{E}\left[\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\mathbf{\Sigma}\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\right]$ we employ a useful property from matrix algebra (which relates the vectorization operator matrix to trace operator (For real matrices $\{\mathbf{A}, \mathbf{B}\}$, the trace of a product can be written as $\mathsf{tr}\{\mathbf{A}^{\mathsf{T}}\mathbf{B}\} = \mathsf{vec}^{\mathsf{T}}\{\mathbf{B}\}\mathsf{vec}\{\mathbf{A}\}$) [50]) together with the fact that $\mathbf{\Sigma}$ is symmetric and deterministic, to obtain

$$\begin{aligned}\mathsf{E}\left[\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\mathbf{\Sigma}\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\right] &= \mathsf{E}\left[\mathsf{tr}\left\{\mathbf{\Sigma}\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\right\}\right] \\ &= \mathsf{vec}^{\mathsf{T}}\left\{\mathsf{E}\left[\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\right]\right\}\boldsymbol{\sigma} \\ &= \mathsf{vec}^{\mathsf{T}}\{\mathbf{H}_k\}\boldsymbol{\sigma}\end{aligned} \tag{27}$$

where

$$\mathbf{H}_k \triangleq \mathsf{E}\Big[\mathbf{\Pi}_{k,t}\mathbf{u}_{k,t}^{\mathsf{T}}\mathbf{u}_{k,t}\mathbf{\Pi}_{k,t}\Big] \tag{28}$$

which, in the light of Assumptions 1 and 2, can be evaluated as

$$\mathbf{H}_k = \mathsf{E}[\mathbf{\Pi}_{k,t}\mathbf{R}_{u,k}\mathbf{\Pi}_{k,t}] \tag{29}$$

It follows by direct inspection that the entries of $\mathbf{H}_k$ are given by

$$\mathbf{H}_k(p,q) = \begin{cases} (1-\pi_k)^2\mathbf{R}_{u,k}(\mathrm{p},\mathrm{q}), & \mathrm{p} \neq \mathrm{q} \\ (1-\pi_k)\mathbf{R}_{u,k}(\mathrm{p},\mathrm{p}), & \mathrm{p} = \mathrm{q} \end{cases} \tag{30}$$

where

$$\mathsf{E}[\mathbf{\Pi}_{k,t}\mathbf{\Pi}_{\ell,t}] = \begin{cases} (1-\pi_k)(1-\pi_\ell), & k \neq \ell \\ (1-\pi_k), & k = \ell \end{cases}$$

Substitution of (24) and (27) into (23) gives

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_k^{(t)}\right\|_{\boldsymbol{\sigma}}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{F}_k\boldsymbol{\sigma}}^2\right] + \mu_k^2\sigma_{v,k}^2\mathsf{vec}^{\mathsf{T}}\{\mathbf{H}_k\}\boldsymbol{\sigma} \tag{31}$$

In the following proposition, we summarize the required conditions that guarantee the mean-square convergence for the DC-ILMS algorithm.

Proposition 2 (Mean-square Stability). *Assume the data $\{d_k(t), \mathbf{u}_{k,t}\}$ satisfy the model* (7)*. Under Assumptions 1 and 2, the recursion of type* (31) *is stable and convergent if, and only if, $\mathbf{F}_k$ given by* (25) *is stable. On the other hand, the stability of $\mathbf{F}_k$ is achieved when $\{\mu_k\}$ are chosen, such that they satisfy* (14).

Proof. Let $\mathbf{A}$ and $\mathbf{B}$ be two arbitrary matrices of compatible dimensions. Then, any eigenvalue of $\mathbf{A} \otimes \mathbf{B}$ is a product $\lambda_i(\mathbf{A}) \times \lambda_j(\mathbf{B})$, where $\lambda_i(\mathbf{A})$ is an eigenvalue of $\mathbf{A}$ and $\lambda_j(\mathbf{B})$ is an eigenvalue of $\mathbf{B}$ [50]. Moreover, the sets of eigenvalues of $\mathbf{A}$ and $\mathbf{A}^{\mathsf{T}}$ are equal. Accordingly, using the expression (26), we have that $\rho(\mathbf{F}_k) = [\rho(\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_l)\mathbf{R}_{u,k})]^2$. It follows that $\mathbf{F}_k$ is stable if, and only if, $(\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_l)\mathbf{R}_{u,k})$ is stable. Therefore, $\{\mu_k\}$ that guarantee the mean stability, ensure mean-square stability as well. Thus, the result (14) follows. □

3.4. Learning Curves

In this section, the variance relation (31) is employed to obtain a recursive equation that explains the transient behavior of the CD-ILMS algorithm. Since the weighting matrices, $\{\mathbf{\Sigma}, \mathbf{\Sigma}'\}$, can be node-dependent, we can replace $\{\boldsymbol{\sigma}, \boldsymbol{\sigma}'\}$ by $\{\boldsymbol{\sigma}_k, \boldsymbol{\sigma}'_k\}$, so that (31) can be rewritten in the following form

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_k^{(t)}\right\|_{\boldsymbol{\sigma}_k}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{F}_k\boldsymbol{\sigma}_k}^2\right] + \mathbf{g}_k\boldsymbol{\sigma}_k \tag{32}$$

with $\mathbf{g}_k \triangleq \mu_k^2\sigma_{v,k}^2\mathsf{vec}^{\mathsf{T}}\{\mathbf{H}_k\}$. Observe that the expression (32) relates $\tilde{\boldsymbol{\psi}}_k^{(t)}$ to $\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}$ (not $\tilde{\boldsymbol{\psi}}_k^{(t-1)}$). To resolve this issue, the incremental topology along with the weighting matrices should be employed [19] as follows: first, by iterating (32) a set of K coupled equalities are obtained

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_1^{(t)}\right\|_{\boldsymbol{\sigma}_1}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{\mathrm{K}}^{(t-1)}\right\|_{\mathbf{F}_1\boldsymbol{\sigma}_1}^2\right] + \mathbf{g}_1\boldsymbol{\sigma}_1$$

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_k^{(t)}\right\|_{\boldsymbol{\sigma}_2}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_1^{(t)}\right\|_{\mathbf{F}_2\boldsymbol{\sigma}_2}^2\right] + \mathbf{g}_2\boldsymbol{\sigma}_2$$

$$\vdots$$

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\boldsymbol{\sigma}_{k-1}}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-2}^{(t)}\right\|_{\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1}}^2\right] + \mathbf{g}_{k-1}\boldsymbol{\sigma}_{k-1} \tag{33}$$

$$[+2ex]\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_k^{(t)}\right\|_{\boldsymbol{\sigma}_k}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{F}_k\boldsymbol{\sigma}_k}^2\right] + \mathbf{g}_k\boldsymbol{\sigma}_k \tag{34}$$

$$\vdots$$

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{\mathrm{K}}^{(t)}\right\|_{\boldsymbol{\sigma}_{\mathrm{K}}}^2\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{\mathrm{K}-1}^{(t)}\right\|_{\mathbf{F}_{\mathrm{K}}\boldsymbol{\sigma}_{\mathrm{K}}}^2\right] + \mathbf{g}_{\mathrm{K}}\boldsymbol{\sigma}_{\mathrm{K}}$$

In order to explain the flow of energy through the agents it is required to make connection between the free parameters $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$. If we choose the weighting vector $\boldsymbol{\sigma}_{k-1} = \mathbf{F}_k\boldsymbol{\sigma}_k$ and combine (33) and (34) we get

$$\begin{aligned}\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_k^{(t)}\right\|_{\boldsymbol{\sigma}_k}^2\right] &= \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{F}_k\boldsymbol{\sigma}'_k}^2\right] + \mathbf{g}_k\boldsymbol{\sigma}_k \\ &= \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-2}^{(t)}\right\|_{\mathbf{F}_{k-1}\mathbf{F}_k\boldsymbol{\sigma}_k}^2\right] + \mathbf{g}_{k-1}\mathbf{F}_k\boldsymbol{\sigma}_k + \mathbf{g}_k\boldsymbol{\sigma}_k\end{aligned} \tag{35}$$

Iterating in this manner, a recursive expression is obtained which relates $\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}$ to $\tilde{\boldsymbol{\psi}}_{k-1}^{(t-1)}$, namely

$$\begin{aligned}\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\boldsymbol{\sigma}_{k-1}}^2\right] &= \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t-1)}\right\|_{\mathbf{F}_k\ldots\mathbf{F}_{\mathrm{K}}\mathbf{F}_1\ldots\mathbf{F}_{k-1}\boldsymbol{\sigma}_k}^2\right] \\ &\quad + \mathbf{g}_k\mathbf{F}_{k+1}\ldots\mathbf{F}_{\mathrm{K}}\mathbf{F}_1\ldots\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1} \\ &\quad + \mathbf{g}_{k+1}\mathbf{F}_{k+2}\ldots\mathbf{F}_{\mathrm{K}}\mathbf{F}_1\ldots\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1}. \\ &\quad \ldots + \mathbf{g}_{k-2}\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1} + \mathbf{g}_{k-1}\boldsymbol{\sigma}_{k-1}\end{aligned} \tag{36}$$

By choosing $\boldsymbol{\sigma} = \mathbf{q} = \mathrm{vec}\{\mathbf{I}_{\mathrm{L}^2}\}$ and substituting it in (36), gives the following expression that explains how MSD of node k evolves over time:

$$\begin{aligned}\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\boldsymbol{\sigma}_{k-1}}^2\right] &= \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\right\|_{\left(\boldsymbol{\Gamma}_{k-1,1}\right)^t\mathbf{q}}^2\right] \\ &\quad + \mathbf{a}_{k-1}\left(\mathbf{I}_{\mathrm{L}^2} + \ldots + \left(\boldsymbol{\Gamma}_{k-1,1}\right)^{t-1}\right)\mathbf{q}\end{aligned} \tag{37}$$

where $\boldsymbol{\Gamma}_{k-1,\ell} \in \mathbb{R}^{\mathrm{L}^2\times\mathrm{L}^2}$, the product of $\mathbf{F}_k$ matrices for each node, and $\mathbf{a}_{k-1} \in \mathbb{R}^{1\times\mathrm{L}^2}$ are defined by

$$\boldsymbol{\Gamma}_{k-1,\ell} \triangleq \mathbf{F}_{k+\ell-1}\mathbf{F}_{k+1}\ldots\mathbf{F}_{\mathrm{L}}\mathbf{F}_1\ldots\mathbf{F}_{k-1},\ \ell = 1,2,\ldots,\mathrm{L} \tag{38}$$

and

$$\mathbf{a}_{k-1} \triangleq \mathbf{g}_k\boldsymbol{\Gamma}_{k-1,2} + \mathbf{g}_{k+1}\boldsymbol{\Gamma}_{k-1,3}\ldots + \mathbf{g}_{k-2}\boldsymbol{\Gamma}_{k-1,\mathrm{K}} + \mathbf{g}_{k-1} \tag{39}$$

Therefore, the theoretical expression for MSD of node k is given by the compact form

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{q}}^{2}\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\right\|_{\gamma_{t-1}}^{2}\right] + \mathbf{a}_{k-1}\mathbf{b}_{t-1} \tag{40}$$

where the vectors γ_{t-1} and $\mathbf{b}_{t-1}$ are formulated by

$$\gamma_{t-1} = \boldsymbol{\Gamma}_{k-1,1}\gamma_{t-2},\ \gamma_{-1} = \mathbf{q}$$
$$\mathbf{b}_{t-1} = \mathbf{b}_{t-2} + \gamma_{t-1},\ \mathbf{b}_{-1} = \text{null vector of size } \mathrm{L}^2 \times 1$$

Accordingly, the selection of $\boldsymbol{\sigma}_{k-1} = \mathbf{r}_k = \mathsf{vec}\{\mathbf{R}_{u,k}\}$ leads to a an expression that explains how EMSE of node k evolves over time:

$$\mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(t)}\right\|_{\mathbf{r}_k}^{2}\right] = \mathsf{E}\left[\left\|\tilde{\boldsymbol{\psi}}_{k-1}^{(-1)}\right\|_{\gamma'_{t-1}}^{2}\right] + \mathbf{a}_{k-1}\mathbf{b}'_{t-1} \tag{41}$$

where the vectors γ'_{t-1} and $\mathbf{b}'_{t-1}$ are given, respectively, by

$$\gamma'_{t-1} = \boldsymbol{\Gamma}_{k-1,1}\gamma'_{t-2},\ \gamma'_{-1} = \mathbf{r}_k$$
$$\mathbf{b}'_{t-1} = \mathbf{b}'_{t-2} + \gamma'_{t-1},\ \mathbf{b}'_{-1} = \text{null vector of size } \mathrm{L}^2 \times 1$$

3.5. Steady-State Behavior

The variance relation (32) can also be used to evaluate the steady-state performance of CD-ILMS algorithm. Let $\mathbf{p}_k \triangleq \tilde{\boldsymbol{\psi}}_k^{(\infty)}$. At steady-state, i.e., when $t \to \infty$, the variance relation (32) can be written as

$$\mathsf{E}\left[\|\mathbf{p}_k\|_{\boldsymbol{\sigma}_k}^2\right] = \mathsf{E}\left[\|\mathbf{p}_{k-1}\|_{\mathbf{F}_k\boldsymbol{\sigma}_k}^2\right] + \mathbf{g}_k\boldsymbol{\sigma}_k \tag{42}$$

Iterating the Equation (42) for an incremental network topology and selecting appropriate weighting vectors $\boldsymbol{\sigma}_k$ for $k = 1, 2, \ldots, \mathrm{K}$, we obtain an equality only involving $\mathbf{p}_{k-1}$, given by

$$\begin{aligned}\mathsf{E}\left[\|\mathbf{p}_{k-1}\|_{\boldsymbol{\sigma}_{k-1}}^2\right] = \mathsf{E}&\left[\|\mathbf{p}_{k-1}\|_{\mathbf{F}_k\ldots\mathbf{F}_\mathrm{K}\mathbf{F}_1\ldots\mathbf{F}_{k-1}\boldsymbol{\sigma}_k}^2\right] \\ &+ \mathbf{g}_k\mathbf{F}_{k+1}\ldots\mathbf{F}_\mathrm{K}\mathbf{F}_1\ldots\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1} \\ &+ \mathbf{g}_{k+1}\mathbf{F}_{k+2}\ldots\mathbf{F}_\mathrm{K}\mathbf{F}_1\ldots\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1}. \\ &\ldots + \mathbf{g}_{k-2}\mathbf{F}_{k-1}\boldsymbol{\sigma}_{k-1} + \mathbf{g}_{k-1}\boldsymbol{\sigma}_{k-1}\end{aligned} \tag{43}$$

Using (38) and (39) to simplify the expression (43), we obtain

$$\mathsf{E}\left[\|\mathbf{p}_k\|_{(\mathbf{I}_{\mathrm{L}^2} - \boldsymbol{\Gamma}_{k-1,1})\boldsymbol{\sigma}_{k-1}}^2\right] = \mathbf{a}_{k-1}\boldsymbol{\sigma}_{k-1} \tag{44}$$

Equation (44) can be employed to evaluate the performance measures at node k, defined as follows:

$$\eta_k \triangleq \mathsf{E}\left[\|\mathbf{p}_{k-1}\|_{\mathbf{q}}^2\right],\quad \mathbf{q} = \mathsf{vec}\{\mathbf{I}_{\mathrm{L}^2}\} \qquad \text{(MSD)} \tag{45}$$

$$\zeta_k \triangleq \mathsf{E}\left[\|\mathbf{p}_{k-1}\|_{\mathbf{r}_k}^2\right],\quad \mathbf{r}_k = \mathsf{vec}\{\mathbf{R}_{u,k}\} \qquad \text{(EMSE)} \tag{46}$$

Proper selection of the weighting vector $\boldsymbol{\sigma}_{k-1}$ in (44) gives the required mean-square values. Selecting the weighting vector $\boldsymbol{\sigma}_k$ as the solution of the linear equation $(\mathbf{I}_{\mathrm{L}^2} - \boldsymbol{\Gamma}_{k-1,1})\boldsymbol{\sigma}_{k-1} = \mathbf{q}$ or $(\mathbf{I}_{\mathrm{L}^2} - \boldsymbol{\Gamma}_{k-1,1})\boldsymbol{\sigma}_{k-1} = \mathbf{r}_k$, the desired MSD and EMSE at node k can be obtained as follows:

$$\eta_k = \mathbf{a}_{k-1}\left(\mathbf{I}_{\mathrm{L}^2} - \mathbf{\Gamma}_{k-1,1}\right)^{-1}\mathbf{q}, \qquad \text{(MSD)} \tag{47}$$

$$\zeta_k = \mathbf{a}_{k-1}\left(\mathbf{I}_{\mathrm{L}^2} - \mathbf{\Gamma}_{k-1,1}\right)^{-1}\mathbf{r}_k, \qquad \text{(EMSE)} \tag{48}$$

The vector $\mathbf{a}_{k-1}$ can be regarded as the effect of aggregating the transformed noise and the local data statistics from all the nodes in the incremental network topology and the matrix $\mathbf{\Gamma}_{k-1,\ell}$ can be interpreted as the transition matrix for the weighting vector $\boldsymbol{\sigma}_{k-1}$.

Remark 1. *It should be noted that* (47) *and* (48) *are exactly the theoretical MSD and EMSE for full-update ILMS algorithm, which means that the CD-ILMS algorithm has the same steady-state error performance compared with the incremental LMS algorithm.*

4. Further Insight into the Proposed CD-ILMS

In order to gain more insight into the performance of CD-ILMS algorithm, in this section we compare the convergence rate and computational complexity of CD-ILMS algorithm with the regular full-update ILMS algorithm [19].

4.1. Convergence Rate

To make the analysis more tractable, we consider the following assumption.

Assumption 3. *The missing probabilities* $\{\pi_k\}$*, the covariance matrices* $\{\mathbf{R}_{u,k}\}$*, and the step-size* $\{\mu_k\}$ *are identical, i.e.,*

$$\pi_k = \pi, \;\; \mu_k = \mu, \; \mathbf{R}_{u,k} = \mathbf{R}_u \quad \forall k$$

It can be seen from (13) that the evolution of weight-error vector for both CD-ILMS and ILMS algorithms are controlled by the modes of the following matrices

$$\mathbf{Q}_{\mathrm{coor}} = \left(\mathbf{I}_{\mathrm{L}} - \mu_k(1-\pi_k)\mathbf{R}_{u,k}\right), \quad \forall k \tag{49}$$

$$\mathbf{Q}_{\mathrm{full}} = \left(\mathbf{I}_{\mathrm{L}} - \mu_k\mathbf{R}_{u,k}\right), \quad \forall k \tag{50}$$

where the subscript 'coor' and 'full' denote, respectively, the stochastic coordinate-descent and full-update incremental implementation. Thus, under Assumption 3, we have

$$\mathbf{Q}_{\mathrm{coor}} = \left(\mathbf{I}_{\mathrm{L}} - \mu(1-\pi)\mathbf{R}_u\right)^{\mathrm{K}} \tag{51}$$

$$\mathbf{Q}_{\mathrm{full}} = \left(\mathbf{I}_{\mathrm{L}} - \mu\mathbf{R}_u\right)^{\mathrm{K}} \tag{52}$$

From (51) and (52), we find that the modes of convergence are given by

$$\kappa_{\mathrm{coor},l} = \left\{(1-\mu(1-\pi)\lambda_l)^{\mathrm{K}}\right\}_{l=1,\ldots,\mathrm{L}} \tag{53}$$

$$\kappa_{\mathrm{full},l} = \left\{(1-\mu\lambda_l)^{\mathrm{K}}\right\}_{l=1,\ldots,\mathrm{L}} \tag{54}$$

where λ_l denotes the eigenvalues of $\mathbf{R}_u$. Letting τ_{coor} and τ_{full} represent the largest number of time iterations that are required for the mean error vector, $\mathrm{E}\left[\tilde{\boldsymbol{\psi}}_k^{(t)}\right]$, to converge to zero. Then, it holds that

$$\begin{aligned}\frac{\tau_{\text{coor}}}{\tau_{\text{full}}} &= \frac{\ln(\kappa_{\text{full},l})}{\ln(\kappa_{\text{coor},l})} \\ &= \frac{\ln\Big((1-\mu\lambda_l)^{\text{K}}\Big)}{\ln\Big((1-\mu(1-\pi)\lambda_l)^{\text{K}}\Big)} \\ &\overset{*}{\approx} \frac{-\mu\lambda_l}{-\mu(1-\pi)\lambda_l} = \frac{1}{1-\pi}\end{aligned} \tag{55}$$

where in step $(*)$ we considered $\ln(1-z) \approx -z$ as $z \to 0$.

Remark 2. *Expression (55) illustrates the increase in the convergence time in the CD-ILMS algorithm. Since the convergence time is longer, the CD-ILMS algorithm may need more quantities to be exchanged through the network compared to the ILMS algorithm.*

4.2. Computational Complexity

We now provide a discussion on computational complexity of full-update ILMS and its coordinate-descent implementations. Let $\theta_{\text{add}} \geq 0$ and $\theta_{\text{mul}} \geq 0$ represent the number of real additions and multiplications, respectively, that are required for every entry of the gradient vector. In CD-ILMS algorithm, every node requires $(1-\pi)(\theta_{\text{mul}}\text{L}+\text{L})$ real multiplication and $(1-\pi)(\theta_{\text{add}}\text{L}+\text{L})$ real additions per iteration in adaptation step (10), while in the full implementation, every node requires $\theta_{\text{mul}}\text{L}+\text{L}$ multiplications and $\theta_{\text{add}}\text{L}+\text{L}$ additions per iteration. Moreover, Let β_{coor} and β_{full} denote the number of multiplications needed by the adaptation steps per iteration at each agent k in the CD-ILMS and full-update cases. Then,

$$\beta_{\text{full},k} = (\theta_{\text{mul}}+1)\text{L} \tag{56}$$

$$\beta_{\text{coor},k} = \beta_{\text{full},k} - (\theta_{\text{mul}}+1)\text{L}\pi \tag{57}$$

If these algorithms require τ_{coor} and τ_{full} iterations to attain their steady-state values, then the total number of real multiplications at node k, represented by $\alpha_{\text{coor},k}$ and $\alpha_{\text{full},k}$, are obtained by

$$\alpha_{\text{coor},k} = \beta_{\text{coor},k} \times \tau_{\text{coor}} \tag{58}$$

$$\alpha_{\text{full},k} = \beta_{\text{full},k} \times \tau_{\text{full}} \tag{59}$$

so that employing (55) results in

$$\frac{\alpha_{\text{coor},k}}{\alpha_{\text{full},k}} = \frac{\beta_{\text{coor},k}}{\beta_{\text{full},k}} \times \frac{1}{1-\pi} \tag{60}$$

Substituting (56) and (57) into the first item in (60), we obtain

$$\frac{\beta_{\text{coor},k}}{\beta_{\text{full},k}} = 1-\pi \tag{61}$$

Thus, from (60) and (61)

$$\frac{\alpha_{\text{coor},k}}{\alpha_{\text{full},k}} = 1 \tag{62}$$

Remark 3. *It is obvious that, $\alpha_{\text{coor},k}$ and $\alpha_{\text{full},k}$ are essentially identical. This means that although the convergence of CD-ILMS algorithm to steady-state region takes longer, the overall computational complexity, i.e., savings in computation per iteration, does remain invariant. Generally speaking, in situations where the computational complexity per iteration require to be minimal, the coordinate-*

descent scheme is recommended. Moreover, the coordinate-descent scheme requires more iterations to achieve the same steady-state performance.

Remark 4 (Communication Costs)**.** *In coordinate-descent incremental and regular incremental schemes, each node k receives weight estimate, $\boldsymbol{\psi}_{k,t}$, from its predecessor node $k-1$ in the cycle. In this manner the overall number of communications needed at each node per iteration is* L. *Moreover, the nodes also need to send their global estimate, $\boldsymbol{w}_{t-1}$, thus the communications requirement per iteration per node is* 2L.

5. Simulation Results

This section provides some computer simulations to evaluate the performance of CD-ILMS algorithm and verify the theoretical analysis. We assume a distributed network composed of $\mathrm{K}=20$ agents with ring topology (see Figure 2). The $\mathrm{L}=8$ vector $\boldsymbol{w}^o$ is generated randomly with $\|\boldsymbol{w}^o\|^2=1$ and step-sizes are $\mu_k=0.02$. To obtain more accurate results, we conducted 100 independent simulations. Although the analysis depends on the independence assumptions, all simulations are accomplished by regressors with shift structure to deal with realistic scenarios. To this end, the regressors are generated by the following first-order auto-regressive model

$$\mathbf{u}_{k,t}(l)=z_k\mathbf{u}_{k,t}(l-1)+\sqrt{1-z_k^2}\mathbf{h}_{k,t}(l),\ 1\leq l<\mathrm{L}$$

where the parameters $z_k\in(-1,\ 1)$ and $\mathbf{h}_{k,t}$ denotes a are zero-mean, unit-variance Gaussian sequences. For each node, the measurement noise $v_k(t)$ is a zero mean white Gaussian sequence with $\sigma_v^2\in(0.001,\ 0.1)$. Figure 3 illustrates the network statistical settings.

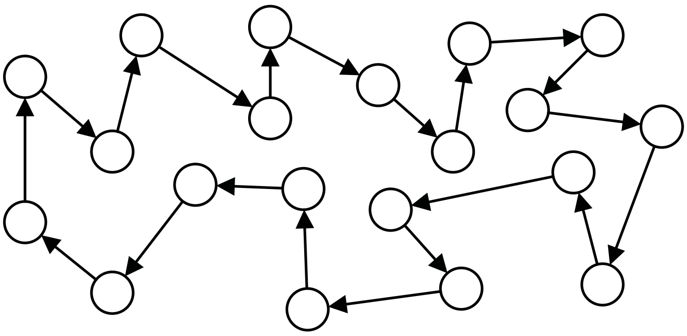

Figure 2. Network topology consisting of $\mathrm{K}=20$ agents with ring topology.

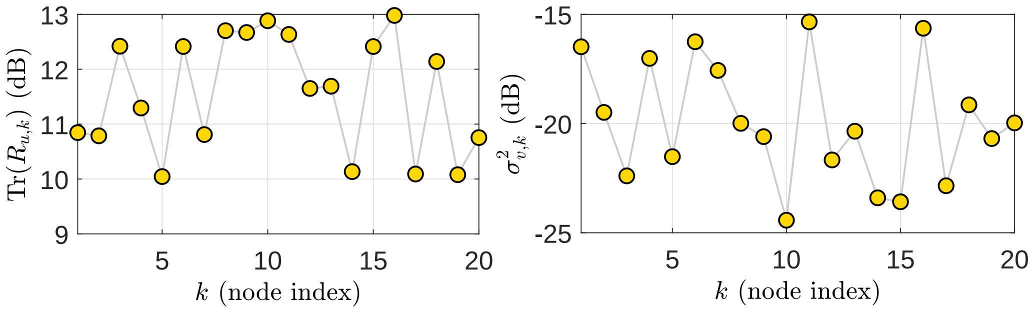

Figure 3. The network statistical settings: regressor power profile (**Left**) and observation noise power profile (**Right**).

In Figure 4, we plot the experimental and theoretical MSD and EMSE learning curves using both full and partial updates for different values of $\pi_k\in\{0.4,0.6,0.9\}$. It can be observed that the CDILS algorithm has the same steady-state error performance compared with the ILMS algorithm. Moreover, as we expected, as $\pi_k\to 0$ the performance of the coordinate-descent approaches to that of the full-gradient incremental algorithm. Generally speaking, the speed of convergence reduces proportionally for coordinate-descent schemes as the missing probability is decreased.

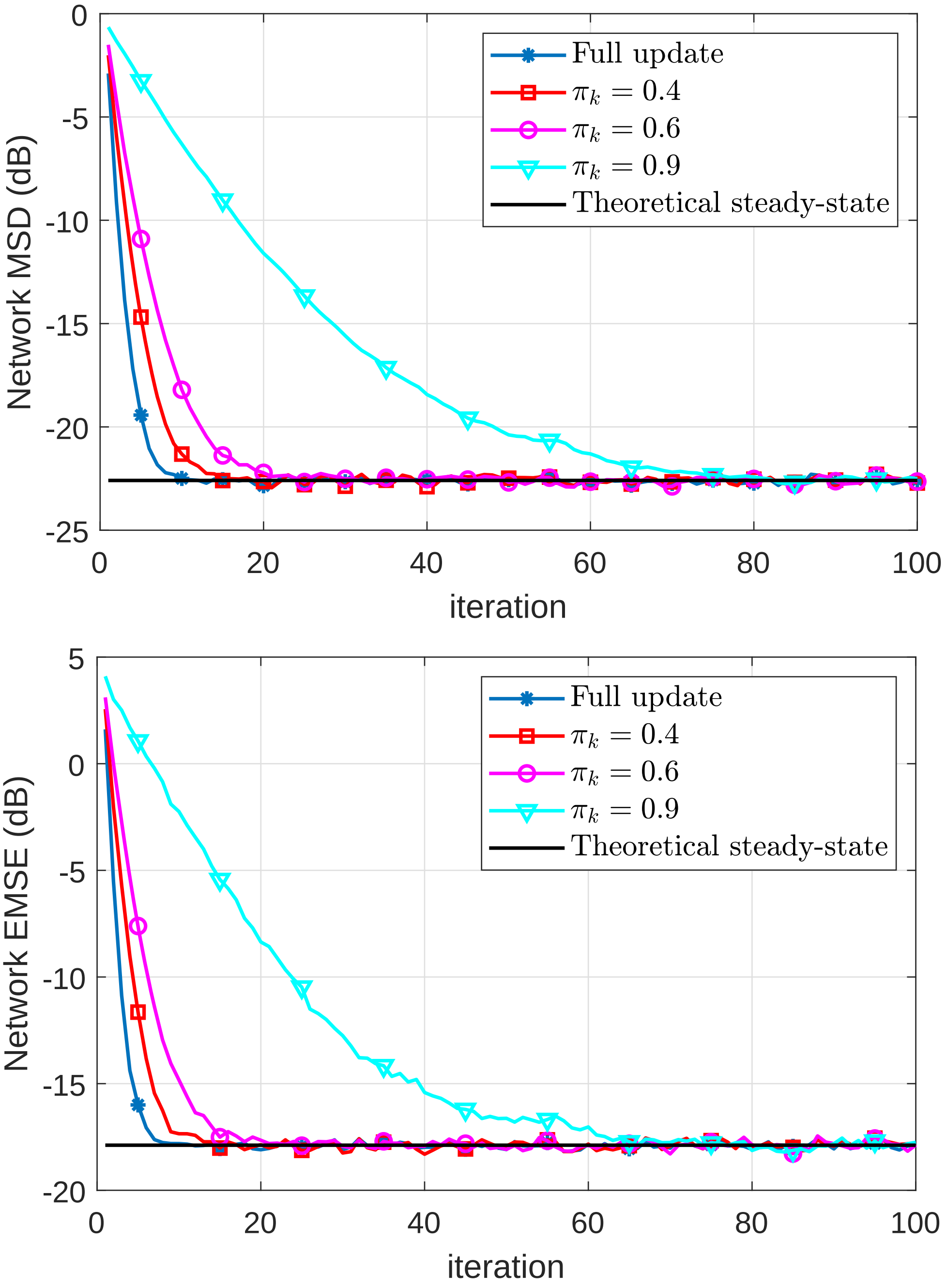

Figure 4. Simulated network MSD and EMSE curves and their theoretical results.

In Figure 5, we demonstrate the experimental and theoretical steady-state MSDs and EMSEs for $\pi_k = 0.6$. It can be seen that, the calculated steady-state MSD and EMSE values using the theoretical expressions in (47) and (48) have a good agreement between with those obtained by the simulation results.

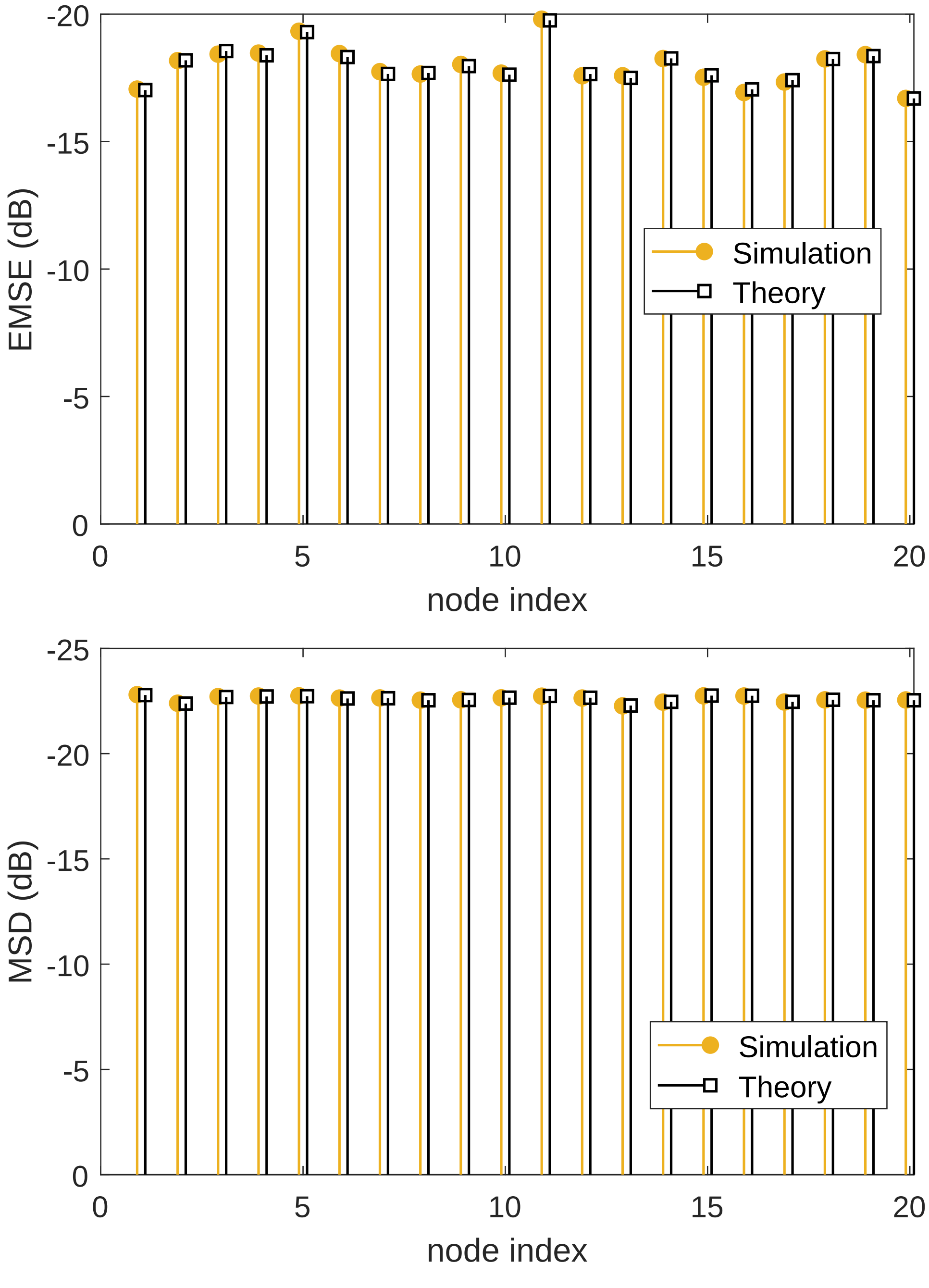

Figure 5. Theoretical and experimental steady-state MSD and EMSE at each node.

Figure 6 shows the modes of convergence for the case $\mathbf{R_u} = \mathbf{I}_L$ and $K = 10$, for $\pi_k \in \{0.5, 0.8\}$, as a function of $\bar{\mu} = \mu\lambda$, both for ILMS and CD-ILMS algorithms. From Figure 6, it can be observed that, regardless of the values of step-size parameter, the full-update ILMS algorithm is always faster than the CD-ILMS algorithm.

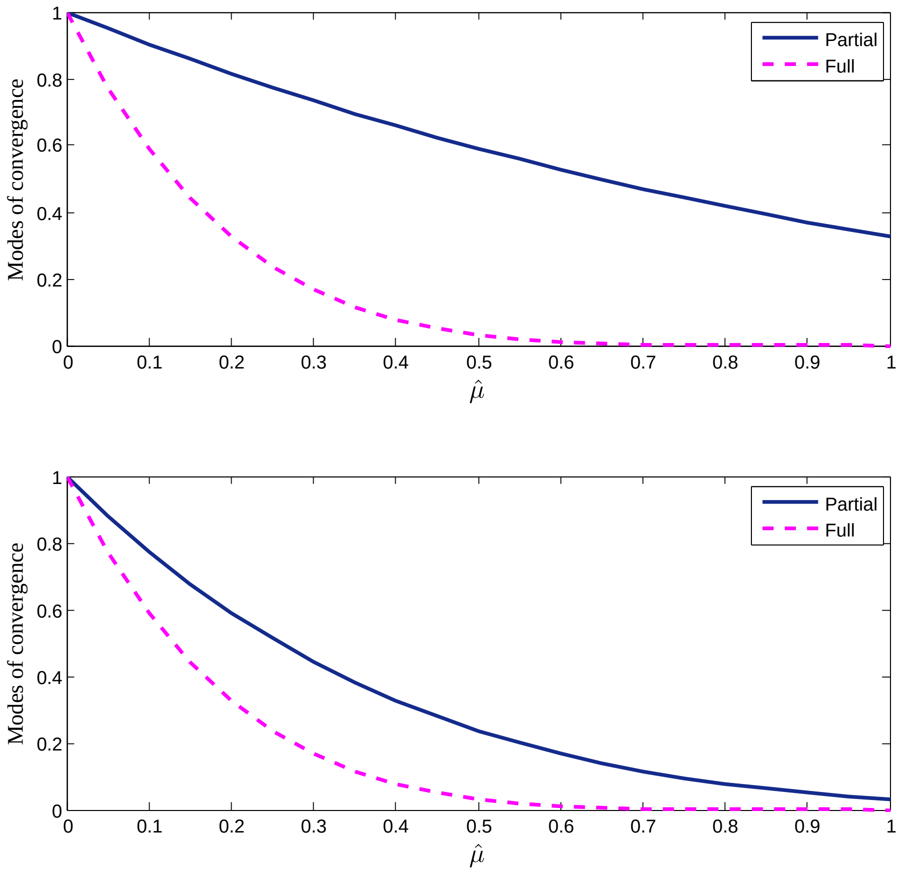

Figure 6. Modes of mean-convergence, for ILMS and CD-ILMS algorithms using K $= 10$, $\pi = 0.8$ (**top**) and $\pi = 0.5$ (**bottom**).

6. Conclusions

In this paper, we have derived the CD-ILMS algorithm for adaptive estimation over incremental networks. Moreover, its detailed performance analysis based on the weighted energy-conservation approach under some assumptions and approximations has been discussed. More specifically, we have derived mean and mean-square stability conditions and theoretical expressions for steady-state and learning curves of MSD and EMSE. To gain further insight into the performance of CD-ILMS algorithm, its convergence and computational complexity have been compared with those of the full-update ILMS algorithm. It has been shown that the full-update ILMS algorithm and CD-ILMS algorithm provide the same steady-state performance, while full-update ILMS is always faster than the CD-ILMS algorithm. Finally, some numerical examples have been provided to support the theoretical derivations.

Author Contributions: Conceptualization, A.K., V.V. and A.R.; methodology, A.K., V.V. and A.R.; software, V.V. and A.R.; validation, A.K., V.V., A.R. and A.F.; formal analysis, X.X.; investigation, A.K., V.V. and A.R.; resources, A.F. and K.T.T.K.; data curation, A.F. and K.T.T.K.; writing—original draft preparation, A.K., V.V. and A.R.; writing—review and editing, A.K., V.V., A.R., A.F. and K.T.T.K.; visualization, A.R.; supervision, S.S.; project administration, S.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rossi, L.A.; Krishnamachari, B.; Kuo, C.C. Distributed parameter estimation for monitoring diffusion phenomena using physical models. In Proceedings of the 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, USA, 4–7 October 2004; pp. 460–469.
2. Li, D.; Wong, K.D.; Hu, Y.H.; Sayeed, A.M. Detection, classification, and tracking of targets. *IEEE Signal Process. Mag.* **2002**, *19*, 17–29.
3. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A survey on sensor networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114. [CrossRef]
4. Ren, W.; Beard, R.W.; Atkins, E.M. Information consensus in multivehicle cooperative control. *IEEE Control Syst. Mag.* **2007**, *27*, 71–82.
5. Zhou, K.; Roumeliotis, S.I. Multirobot active target tracking with combinations of relative observations. *IEEE Trans. Robot.* **2011**, *27*, 678–695. [CrossRef]
6. Amin, S.M.; Wollenberg, B.F. Toward a smart grid: Power delivery for the 21st century. *IEEE Power Energy Mag.* **2005**, *3*, 34–41. [CrossRef]
7. Ibars, C.; Navarro, M.; Giupponi, L. Distributed demand management in smart grid with a congestion game. In Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010; pp. 495–500.
8. Kim, H.; Kim, Y.J.; Yang, K.; Thottan, M. Cloud-based demand response for smart grid: Architecture and distributed algorithms. In Proceedings of the 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), Brussels, Belgium, 17–20 October 2011; pp. 398–403.
9. Giannakis, G.B.; Kekatos, V.; Gatsis, N.; Kim, S.J.; Zhu, H.; Wollenberg, B.F. Monitoring and optimization for power grids: A signal processing perspective. *IEEE Signal Process. Mag.* **2013**, *30*, 107–128. [CrossRef]
10. Duchi, J.C.; Agarwal, A.; Wainwright, M.J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Autom. Control* **2011**, *57*, 592–606. [CrossRef]
11. Chen, J.; Towfic, Z.J.; Sayed, A.H. Dictionary learning over distributed models. *IEEE Trans. Signal Process.* **2014**, *63*, 1001–1016. [CrossRef]
12. Chouvardas, S.; Slavakis, K.; Kopsinis, Y.; Theodoridis, S. A sparsity promoting adaptive algorithm for distributed learning. *IEEE Trans. Signal Process.* **2012**, *60*, 5412–5425. [CrossRef]
13. Zhao, X.; Sayed, A.H. Distributed clustering and learning over networks. *IEEE Trans. Signal Process.* **2015**, *63*, 3285–3300. [CrossRef]
14. Chen, J.; Richard, C.; Sayed, A.H. Diffusion LMS over multitask networks. *IEEE Trans. Signal Process.* **2015**, *63*, 2733–2748. [CrossRef]
15. Kar, S.; Moura, J.M. Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 674–690. [CrossRef]
16. Barbarossa, S.; Scutari, G. Bio-inspired sensor network design. *IEEE Signal Process. Mag.* **2007**, *24*, 26–35. [CrossRef]
17. Khan, U.A.; Moura, J.M.F. Distributing the Kalman filter for large-scale systems. *IEEE Trans. Signal Process.* **2008**, *56*, 4919–4935. [CrossRef]
18. Schizas, I.D.; Mateos, G.; Giannakis, G.B. Distributed LMS for consensus-based in-network adaptive processing. *IEEE Trans. Signal Process.* **2009**, *57*, 2365–2382. [CrossRef]
19. Lopes, C.G.; Sayed, A.H. Incremental adaptive strategies over distributed networks. *IEEE Trans. Signal Process.* **2007**, *55*, 4064–4077. [CrossRef]
20. Rabbat, M.G.; Nowak, R.D. Quantized incremental algorithms for distributed optimization. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 798–808. [CrossRef]
21. Li, L.; Chambers, J.A.; Lopes, C.G.; Sayed, A.H. Distributed estimation over an adaptive incremental network based on the affine projection algorithm. *IEEE Trans. Signal Process.* **2009**, *58*, 151–164. [CrossRef]
22. Cattivelli, F.S.; Sayed, A.H. Analysis of spatial and incremental LMS processing for distributed estimation. *IEEE Trans. Signal Process.* **2010**, *59*, 1465–1480. [CrossRef]
23. Khalili, A.; Tinati, M.A.; Rastegarnia, A. Steady-state analysis of incremental LMS adaptive networks with noisy links. *IEEE Trans. Signal Process.* **2011**, *59*, 2416–2421. [CrossRef]
24. Khalili, A.; Rastegarnia, A.; Sanei, S. Performance analysis of incremental LMS over flat fading channels. *IEEE Trans. Control Netw. Syst.* **2016**, *4*, 489–498. [CrossRef]
25. Sayed, A.H. Adaptive networks. *Proc. IEEE* **2014**, *102*, 460–497. [CrossRef]
26. Sayed, A.H. Adaptation, learning, and optimization over networks. *Found. Trends Mach. Learn.* **2014**, *7*, 311–801. [CrossRef]
27. Chen, J.; Sayed, A.H. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process.* **2012**, *60*, 4289–4305. [CrossRef]
28. Chen, J.; Sayed, A.H. On the learning behavior of adaptive networks—Part I: Transient analysis. *IEEE Trans. Inf. Theory* **2015**, *61*, 3487–3517. [CrossRef]
29. Chen, J.; Sayed, A.H. On the learning behavior of adaptive networks—Part II: Performance analysis. *IEEE Trans. Inf. Theory* **2015**, *61*, 3518–3548. [CrossRef]

30. Zhao, X.; Sayed, A.H. Asynchronous adaptation and learning over networks—Part I: Modeling and stability analysis. *IEEE Trans. Signal Process.* **2014**, *63*, 811–826. [CrossRef]
31. Zhao, X.; Sayed, A.H. Asynchronous adaptation and learning over networks—Part II: Performance analysis. *IEEE Trans. Signal Process.* **2014**, *63*, 827–842. [CrossRef]
32. Zhao, X.; Sayed, A.H. Asynchronous adaptation and learning over networks—Part III: Comparison analysis. *IEEE Trans. Signal Process.* **2014**, *63*, 843–858. [CrossRef]
33. Arablouei, R.; Dogancay, K.; Werner, S.; Huang, Y.F. Adaptive distributed estimation based on recursive least-squares and partial diffusion. *IEEE Trans. Signal Process.* **2014**, *62*, 3510–3522. [CrossRef]
34. Arablouei, R.; Werner, S.; Huang, Y.F.; Dogancay, K. Distributed least mean-square estimation with partial diffusion. *IEEE Trans. Signal Process.* **2014**, *62*, 472–484. [CrossRef]
35. Vahidpour, V.; Rastegarnia, A.; Khalili, A.; Bazzi, W.M.; Sanei, S. Analysis of partial diffusion LMS for adaptive estimation over networks with noisy links. *IEEE Trans. Netw. Sci. Eng.* **2018**, *5*, 101–112. [CrossRef]
36. Vahidpour, V.; Rastegarnia, A.; Khalili, A.; Sanei, S. Analysis of partial diffusion recursive least squares adaptation over noisy links. *IET Signal Process.* **2017**, *11*, 749–757. [CrossRef]
37. Vahidpour, V.; Rastegarnia, A.; Khalili, A.; Sanei, S. Partial Diffusion Kalman Filtering for Distributed State Estimation in Multiagent Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3839–3846. [CrossRef] [PubMed]
38. Vahidpour, V.; Rastegarnia, A.; Latifi, M.; Khalili, A.; Sanei, S. Performance Analysis of Distributed Kalman Filtering with Partial Diffusion Over Noisy Network. *IEEE Trans. Aerosp. Netw. Electron. Syst.* **2019**, *56*, 1767–1782. [CrossRef]
39. Gholami, M.R.; Ström, E.G.; Sayed, A.H. Diffusion estimation over cooperative networks with missing data. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing, Austin, TX, USA, 3–5 December 2013; pp. 411–414.
40. Douglas, S.C. Adaptive filters employing partial updates. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process.* **1997**, *44*, 209–216. [CrossRef]
41. Godavarti, M.; Hero, A.O. Partial update LMS algorithms. *IEEE Trans. Signal Process.* **2005**, *53*, 2382–2399. [CrossRef]
42. Werner, S.; Mohammed, M.; Huang, Y.F.; Koivunen, V. Decentralized set-membership adaptive estimation for clustered sensor networks. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 31 March–4 April 2008; pp. 3573–3576.
43. Werner, S.; Huang, Y.F. Time-and coefficient-selective diffusion strategies for distributed parameter estimation. In Proceedings of the 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 7–10 November 2010; pp. 696–697.
44. Chouvardas, S.; Slavakis, K.; Theodoridis, S. Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 257–273. [CrossRef]
45. Theodoridis, S.; Slavakis, K.; Yamada, I. Adaptive learning in a world of projections. *IEEE Signal Process. Mag.* **2010**, *28*, 97–123. [CrossRef]
46. Gharehshiran, O.N.; Krishnamurthy, V.; Yin, G. Distributed energy-aware diffusion least mean squares: Game-theoretic learning. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 821–836. [CrossRef]
47. Vahidpour, V.; Rastegarnia, A.; Khalili, A.; Bazzi, W.M.; Sanei, S. Variants of Partial Update Augmented CLMS Algorithm and Their Performance Analysis. *arXiv Preprint* **2019**, arXiv:2001.08981.
48. Wang, C.; Zhang, Y.; Ying, B.; Sayed, A.H. Coordinate-descent diffusion learning by networked agents. *IEEE Trans. Signal Process.* **2017**, *66*, 352–367. [CrossRef]
49. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 2012.
50. Abadir, K.M.; Magnus, J.R. *Matrix Algebra*; Cambridge University Press: Cambridge, UK, 2005; Volume 1.