A Distributed Algorithm for In-Network Adaptive Estimation Using Incremental Aggregated Gradient

Wael Bazzi
Electrical Engineering Department
American University in Dubai
Dubai, UAE
wbazzi@aud.edu

Amir Rastegarnia, Azam Khalili
Department of Electrical Engineering
Malayer University
Malayer, 65719-95863, Iran
{a_rastegar,a.khalili}@ieee.org

Saeid Sanei
Department of Computing
University of Surrey
Surrey, GU2 7XH, U.K.
s.sanei@surrey.ac.uk

***Abstract*—In this paper, we consider the distributed estimation problem where a set of nodes is deployed to estimate a parameter of interest when the statistical model (or information) for the underlying processes is not available, or it varies in time. Such a scenario appears in many real-world applications e.g. in sensor networks. The estimation problem can be expressed mathematically as the minimization of a cost function, which is the sum of continuously differentiable local cost functions. The paper aims to develop an iterative, fully distributed and adaptive solution for the optimization problem. Similar to the existing Incremental Least Mean-Squares (ILMS) algorithm, in the proposed algorithm we use steepest-descent method to generate an approximation of the descent direction at every node. However, unlike the ILMS, the proposed algorithm uses the aggregate gradient at each node which is the average of the previously computed gradients by other nodes. The resultant algorithm which is called Incremental Aggregated Gradient-LMS (IAG-LMS) outperforms the ILMS algorithm in terms of the steady-state error. Moreover, its stability bound (in terms of the step-size parameter) is also wider than the ILMS algorithm. We present numerical simulations to support the mentioned claims and illustrate the results.**

Keywords*— *Adaptive networks; incremental; least mean square; estimation.

I. INTRODUCTION

Consider the following unconstrained optimization problem

$$\underset{\mathbf{w}}{\text{minimize}} \;\; f^{\mathrm{g}}(\mathbf{w}) = \sum_{k=1}^{N} f_k(\mathbf{w}), \quad \mathbf{w} \in \mathbb{R}^M \tag{1}$$

where every $f_k : \mathbb{R}^M \rightarrow \mathbb{R}$ is a continuously differentiable scalar function on $\mathbb{R}^M$ ($M \in \mathbb{Z} \geq 1$). The motivation for considering optimization models of the form in (1) stems from practical problems that arise in a variety of applications such as sensor networks, precision agriculture, environment monitoring, disaster relief management, smart spaces, target localization, neural networks training, and a number of medical applications [1],[2]. As an example, in sensor networks, each $f_k(\mathbf{w})$ corresponds to the data collected by the kth sensor in the network. There are already several useful strategies for estimation over distributed networks such as consensus strategies and adaptive networks. In the consensus strategy [3],[4],[5], each node estimates the source signal (or parameter of interest) and shares the information with its neighbors. As the number of iterations increases, the local estimates, computed by the nodes *may* converge to the same final estimate. The initial implementations suggested for the consensus strategy relied on the use of two time-scales [6]: one for the collection of measurements and another to iterate over the collected data to attain agreement before the process is repeated. The main problem in the mentioned methods is inability of the network to undertake continuous learning and optimization [7]. Therefore, consensus-based algorithms that rely on two time-scales can not be used in the problems that this paper seeks to provide a solution to them.

Single time-scale consensus strategies, on the other hand, have also been developed in the literature [4],[5],[8],[9]. Although this kind of consensus algorithms can deliver continuous learning, as it is shown in [10], they suffer from stability problems. Indeed, consensus networks can become unstable even if all of the nodes in the network are stable. This problem, along with the need for continuous learning in distributed networks, motivated the development of adaptive networks [11],[12]. An adaptive network is a collection of agents (nodes) that collaborate with each other through in-network local processing rules in order to estimate and track the parameters of interest [12]. The two major classes of adaptive networks are incremental strategy [13],[14],[15],[16] and diffusion strategy [17],[18],[19]. In the incremental mode, a cyclic path through the network is required, and nodes communicate with neighbors within this path, while in diffusion mode, each node can communicates with all of its neighboring nodes. Although incremental networks are less robust to node and link failures, they do offer excellent estimation performance since every node uses data from the entire network to update the local estimate of the desired parameter.

In the available incremental-based adaptive networks such as [13],[15],[16],[20],[21],[22],[23],[24] the estimation problem is solved by starting from steepest-descent solution, splitting the global cost function into N local cost functions (N is the number of nodes), and applying a suitable local stochastic approximations (such as LMS-type learning rule) at every node. The resultant algorithms are able to handle the unavailability or variation of statistical information. However, they use only one of the N components (local cost functions) in order to generate an approximate descent direction [25]. Thus, in this paper we employ the aggregate gradient at each node which is the average of the N previously computed gradients. The resultant algorithm outperforms the existing incremental LMS (ILMS) algorithm in terms of the steady-state error. Moreover, the stability bound (in terms of the step-size parameter) is also wider than of the ILMS algorithm. We also present some simulation results to support the mentioned claims.

Notation: We adopt boldface letters for random quantities. The symbol * denotes conjugation for scalars and Hermitian transpose for matrices. $y = \mathrm{floor}(x)$ rounds the elements of x to the nearest integers less than or equal to x. Moreover, $(a)_N$ denotes a modulo N with representative classes $1, 2, \cdots, N$. We use the notation $\|\mathbf{x}\|^2_{\mathbf{A}} =$

 ISBN: 978-1-61208-347-6

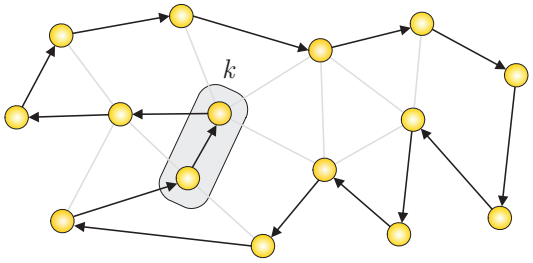

Fig. 1. A distributed network with incremental cooperation among the nodes. The neighbor of node k is defined as its preceding node in Hamilton cycle.

$\mathbf{x}^*\mathbf{A}\mathbf{x}$ for the weighted square norm of $\mathbf{x}$.

II. Problem Statement

To further explain the problem that this paper seeks to provide a solution to it, let us consider a set of nodes $\mathcal{N} = \{1, 2, \cdots, N\}$ that are distributed over a domain in space. Each node $k \in \mathcal{N}$ communicates with its neighbors (denoted by $\mathcal{N}_k$). The set of neighbors for node k for incremental cooperation mode is shown in Fig. 1. At time instant i, node k records the scalar measurement $d_k(i)$ and $1 \times M$ regression data $\mathbf{u}_{k,i}$. The purpose of the network is to solve the following optimization problem at every node in the network

$$\underset{\mathbf{w}}{\text{minimize}} \sum_{k=1}^{N} \mathbb{E}\big[|d_k(i) - \mathbf{u}_{k,i}\mathbf{w}|^2\big] \tag{2}$$

Comparing (1) with (2) reveals that

$$f_k(\mathbf{w}) = \mathbb{E}\big[|d_k(i) - \mathbf{u}_{k,i}\mathbf{w}|^2\big]$$

Remark 1. *In many practical applications, we can assume a linear regression model between the measurements $\{d_k(i)\}$ and $\{\mathbf{u}_{k,i}\}$. According to this model, we have*

$$d_k(i) = \mathbf{u}_{k,i}\mathbf{w}^o + v_k(i) \tag{3}$$

where $\mathbf{w}^o \in \mathbb{R}^M$ is an unknown parameter and $v_k(i)$ is the observation noise term with variance $\sigma^2_{v,k}$. Based on the intended application, the vector $\mathbf{w}^o$ may represent different physical quantities, e.g. location of a target and parameter of an auto-regressive (AR) model.

The optimal solution for (2) is $\mathbf{w}^o$ and can be expressed in terms of the statistics of recorded data $\{d_k(i), \mathbf{u}_{k,i}\}$ via the following normal equation [13],[14]

$$\left(\sum_{k=1}^{N} \mathbf{R}_{u,k}\right)\mathbf{w}^o = \sum_{k=1}^{N} \boldsymbol{p}_{du,k} \tag{4}$$

where

$$\mathbf{R}_{u,k} = \mathbb{E}[\mathbf{u}^*_{k,i}\mathbf{u}_{k,i}], \quad \boldsymbol{p}_{du,k} = \mathbb{E}[d_k(i)\mathbf{u}^*_{k,i}] \tag{5}$$

Remark 2. *The unknown vector $\mathbf{w}^o$ in the linear model* (3) *is the same as the optimal solution in* (4)*.*

This solution needs the statistical information $\{\boldsymbol{p}_{du,k}, \mathbf{R}_{u,k}\}_{k\in\mathcal{N}}$ to be available at every node, which are not available in many applications or they may vary in time. In the next section, we introduce our proposed distributed estimation algorithm which relies on the incremental aggregated gradient method.

III. Proposed Algorithm

To solve the unconstrained problem in (1), we start with the standard steepest-descent method. Let $\mathbf{w}^{\mathrm{g}}_i$ be the global estimate for $\mathbf{w}^o$ at iteration i. Then, using the standard steepest-descent method we have

$$\mathbf{w}^{\mathrm{g}}_i = \mathbf{w}^{\mathrm{g}}_{i-1} - \mu\big(\nabla_{\mathbf{w}} f^{\mathrm{g}}(\mathbf{w}^{\mathrm{g}}_{i-1})\big)^* \tag{6}$$

where $\mu > 0$ is a step-size parameter and $\nabla_{\mathbf{w}} f^{\mathrm{g}}$ represents the complex gradient of the $f^{\mathrm{g}}(\mathbf{w}^{\mathrm{g}})$. The required gradient in (6) is given by

$$\big(\nabla_{\mathbf{w}} f^{\mathrm{g}}(\mathbf{w}^{\mathrm{g}}_{i-1})\big)^* = \sum_{k=1}^{N}(\mathbf{R}_{u,k}\mathbf{w}^{\mathrm{g}}_{i-1} - \boldsymbol{p}_{du,k}) \tag{7}$$

Substituting (7) into (6) yields the steepest-descent solution as

$$\mathbf{w}^{\mathrm{g}}_i = \mathbf{w}^{\mathrm{g}}_{i-1} + \mu\sum_{k=1}^{N}(\boldsymbol{p}_{du,k} - \mathbf{R}_{u,k}\mathbf{w}^{\mathrm{g}}_{i-1}) \tag{8}$$

As we mentioned before, this is not a practical solution as it needs the statistical information $\{\boldsymbol{p}_{du,k}, \mathbf{R}_{u,k}\}_{k\in\mathcal{N}}$. This issue can be addressed by replacing the statistical averages by time averages assuming that the process is ergodic, i.e.

$$\mathbf{R}_{u,k} \approx \mathbf{u}^*_{k,i}\mathbf{u}_{k,i}, \quad \boldsymbol{p}_{du,k} \approx d_k(i)\mathbf{u}^*_{k,i} \tag{9}$$

Replacing (9) in (8) yields

$$\mathbf{w}^{\mathrm{g}}_i = \mathbf{w}^{\mathrm{g}}_{i-1} + \mu\sum_{k=1}^{N}\mathbf{u}^*_{k,i}(d_k(i) - \mathbf{u}_{k,i}\mathbf{w}^{\mathrm{g}}_{i-1}) \tag{10}$$

The above recursive equation can be implemented in a distributed manner by splitting the update equation into N steps and resorting to the incremental cooperation which is established among the nodes. By doing the above steps, the ILMS algorithm is obtained which can be written as follows

$$\begin{cases} \mathbf{w}_{1,i} \leftarrow \mathbf{w}_{N,i-1} \\ \mathbf{w}_{k,i} = \mathbf{w}_{k-1,i} + \mu_k\, \mathbf{u}^*_{k,i}(d_k(i) - \mathbf{u}_{k,i}\mathbf{w}_{k-1,i}) \end{cases} \tag{11}$$

where $\mathbf{w}_{k,i}$ is a local estimate of $\mathbf{w}^o$ at node k and time (iteration) i. Note that we have

$$\mathbf{w}_{N,i} = \mathbf{w}^{\mathrm{g}}_i$$

It is shown in [11], [13] that as $i \to \infty$, we have $\mathbf{w}_{k,i} \to w^o$ in the mean, for every node k, and for an appropriately chosen set of step sizes $\{\mu_k : k = 1, \ldots, N\}$.

It is clear from (6) that the ILMS algorithm uses only the available information at node k in order to generate an approximate for the steepest-descent direction. We can improve the approximate direction if we use the average of the N previously computed gradients [25]. Thus, we use the concept of aggregated gradient to develop our proposed algorithm. To begin with, let us define the sequence of $\{\mathbf{x}_j\},\ j = 1, 2, \cdots$ as

$$\mathbf{x}_j \triangleq \mathbf{w}_{k,i}, \quad j = (i-1) \times N + k \tag{12}$$

Equivalently, we have

$$k = (j)_N, \quad \text{and} \quad i = \text{floor}(j/N) + 1 \tag{13}$$

Using (12), we can define the aggregated gradient at iteration i as

$$\boldsymbol{g}_{k,i} = \Big(\sum_{\ell=0}^{N-1} \nabla_{\mathbf{w}} f_{(k-\ell)_N}(\mathbf{x}_{k-\ell})\Big)^* \tag{14}$$

Note the right hand side of (14) is a function of i since according to (12) $\mathbf{x}_{k-\ell}$ can be expressed in terms of $\mathbf{w}_{k,i}$.

Remark 3. *To calculate the aggregated gradient in* (14) *we need* N *initial points* $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ *which according to* (12) *can be selected as*

$$\mathbf{x}_1 = \mathbf{w}_{1,1}, \quad \mathbf{x}_2 = \mathbf{w}_{2,1}, \quad \cdots \quad \mathbf{x}_N = \mathbf{w}_{N,1}$$

That is why we have defined $\boldsymbol{g}_{k,i}$ *for* $i \geq 2$*. Possible initialization strategies include setting* $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_N$.

Now, we can modify the standard steepest-descent (14) using (6) as follows

$$\mathbf{x}_j = \mathbf{x}_{j-1} - \mu \frac{1}{N} \boldsymbol{g}_{k-1,i}, \quad \text{for } j \geq N+1 \tag{15}$$

or in terms of $\mathbf{w}_{k,i}$ as

$$\mathbf{w}_{k,i} = \mathbf{w}_{k-1,i} - \mu \frac{1}{N} \boldsymbol{g}_{k-1,i} \tag{16}$$

where we add the factor $1/N$ in order to make $\boldsymbol{g}_{k,i}$ comparable to the one used in the standard incremental gradient method [25]. We can also rewrite (14) in an equivalent form as

$$\boldsymbol{g}_{k,i} = \boldsymbol{g}_{k-1,i} - \big(\nabla_{\mathbf{w}} f_{(k)_N}(\mathbf{x}_{k-L})\big)^* + \big(\nabla_{\mathbf{w}} f_{(k)_N}(\mathbf{x}_k)\big)^* \tag{17}$$

As we will discuss later, this form enables us to implement equations (15) and (17) in a distributed manner.

Remark 4. *Note that for* $i \geq 2$ *we can merge the equations* (16) *and* (17) *to obtain an equivalent update equation as*

$$\mathbf{w}_{k,i} = \mathbf{w}_{k-1,i} - \mu \frac{1}{N} \Big(\sum_{\ell=0}^{N-1} \nabla_{\mathbf{w}} f_{(k-\ell)_N}\big(\mathbf{w}_{k',i'}\big) \Big)^* \tag{18}$$

where

$$k' = (k-1-\ell)_N, \quad \text{and} \quad i' = \text{floor}((k-1-\ell)/N) + 1.$$

Although the update equation given by (16) is a distributed solution for (1), however, it is not an *adaptive* solution because we still need to use the second-order moments $\{\boldsymbol{p}_{du,k}, \mathbf{R}_{u,k}\}$ to evaluate (16). To obtain an adaptive solution, we replace the mentioned moments with the local instantaneous approximations as

$$\big(\widetilde{\nabla}_{\mathbf{w}} f_k(\mathbf{w})\big)^* = -(\mathbf{u}_{k,i}^* (d_k(i) - \mathbf{u}_{k,i}\mathbf{w})) \tag{19}$$

where we used $\widetilde{\nabla}_{\mathbf{w}} f_k(\mathbf{w})$ to denote the *approximate* gradient. Using (19), we can rewrite

$$\mathbf{w}_{k,i} = \mathbf{w}_{k-1,i} - \mu \frac{1}{N} \widetilde{\boldsymbol{g}}_{k-1,i} \tag{20a}$$

$$\widetilde{\boldsymbol{g}}_{k,i} = \widetilde{\boldsymbol{g}}_{k-1,i} - \big(\widetilde{\nabla} f_{(k)_N}(\mathbf{x}_{k-L})\big)^* + \big(\widetilde{\nabla} f_{(k)_N}(\mathbf{x}_k)\big)^* \tag{20b}$$

We can implement (20) in a fully distributed manner as follows: at iteration i, upon receiving $\mathbf{x}_{(i-1)\times N+k-1} = \mathbf{w}_{k-1,i}$ and $\widetilde{\boldsymbol{g}}_{k-1,i}$ from node $k-1$, node k updates the local estimate $\mathbf{w}_{k,i}$ and the aggregate estimate $\widetilde{\boldsymbol{g}}_{k,i}$ according to (20) and sends them to the next node $k+1$. The pseudo code of the proposed algorithm (IAG-LMS algorithm) is shown in Algorithm. 1.

Algorithm 1: The pseudo code of the IAG-LMS algorithm.

```
Initialize x_1, x_2, ..., x_N = w_{1,1}, w_{2,1}, ..., w_{N,1} = 0
foreach i = 1, 2, ... do
    foreach k = 1, 2, ..., N ... do
        receive w_{k-1,i} and g̃_{k-1,i} from node k − 1
        update w_{k,i} according to (20a)
        update g̃_{k,i} according to (20b)
        send w_{k,i} and g̃_{k,i} to the next node (node k + 1).
    end
end
```

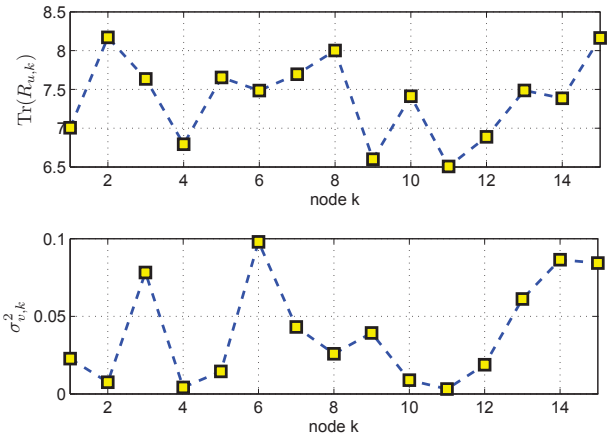

Fig. 2. Node profile $\text{Tr}(R_{u,k})$ (up) and $\sigma_{v,k}^2$ (down).

IV. Simulation Results

In this section, we present some simulation results to evaluate the performance of the proposed algorithm. To this end, we assume a network with $N = 15$ nodes, where the nodes are connected via a ring topology. The regressors $\mathbf{u}_{k,i}$ are generated as independent realizations of a Gaussian distribution with a covariance matrix $\mathbf{R}_{u,k}$ where the eigenvalue spread for every $\mathbf{R}_{u,k}$ is 2. The measurement data $d_k(i)$ at each node k is generated by using the data model (3) where the parameter $\mathbf{w}^o$ is chosen to be $[1\ 1\ 1\ 1\ 1]^T$. The observation noise $v_k(i)$ in (3) is drawn from a Gaussian distribution with variance $\sigma_{v,k}^2 \in (0, 0.1)$. Fig. 2 shows the node profile including the trace of every covariance matrix $\mathbf{R}_{u,k}$ and $\sigma_{v,k}^2$.

We consider the mean-square deviation as the performance metric which is defined at node k as

$$\text{MSD}_k = \mathbb{E}\big[\|\mathbf{w}^o - \mathbf{w}_{k,i}\|^2\big] \tag{21}$$

To evaluate the performance of ILMS and the proposed algorithm, we use *network* MSD that is given by

$$\text{MSD} = \frac{1}{N} \sum_{k=1}^{N} \text{MSD}_k \tag{22}$$

Fig. 3 shows the learning curve, in terms of network MSD, for the ILMS algorithm and the proposed algorithm. For both ILMS and the proposed algorithm we select $\mu = 0.1$. The results are obtained by averaging over 100 independent runs. We observe that the proposed algorithm outperforms the ILMS algorithm in terms of the steady-state error at the expense of a slight increase in the computations (due to the need for updating $\widetilde{\boldsymbol{g}}_{k,i}$) and communications (due to the need for sending $\widetilde{\boldsymbol{g}}_{k,i}$). In Fig. 4, we have plotted the trajectory of the proposed algorithm. The dash straight line corresponds to the $\mathbf{w}^o(1) = 1$. It is seen that the proposed algorithm rapidly converges to the desired parameter.

When we increase the step-size value, we observe that the ILMS algorithm diverges and can not provide an acceptable estimate for $\mathbf{w}^o$. This behavior does not occur for the proposed algorithm as it is shown in Fig. 5, where we have plotted the network MSD for both algorithms for a bigger step-size $\mu = 0.25$). Fig. 6 shows steady-state values of the network MSD for both algorithms in terms of the step-size value. The steady-state values are obtained by averaging over 100 runs 50 time samples after the convergence of algorithms. We observe that the stability range for the proposed algorithm is wider than the ILMS algorithm, which leads to a more robust implementation.

 ISBN: 978-1-61208-347-6

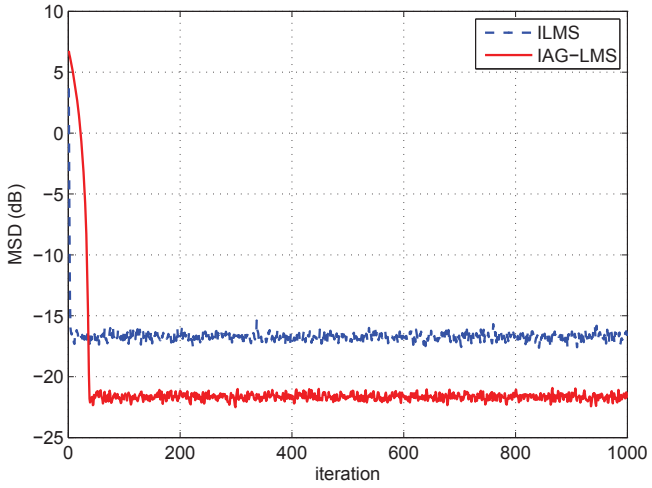

Fig. 3. Learning curve for ILMS and the proposed algorithm in terms of network MSD ($\mu = 0.1$).

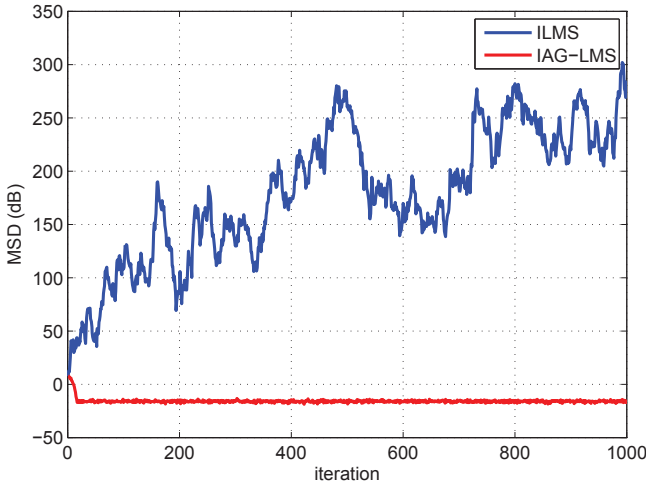

Fig. 5. Learning curve for ILMS and the proposed algorithm in terms of network MSD ($\mu = 0.25$).

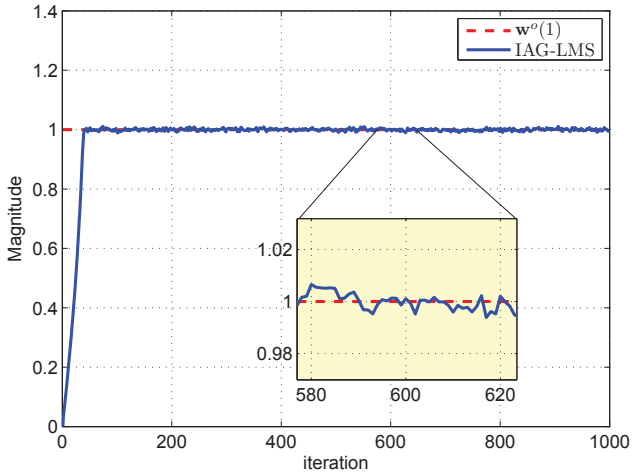

Fig. 4. The trajectory of the proposed algorithm. The dash straight line corresponds to $\mathbf{w}^o(1) = 1$, ($\mu = 0.1$).

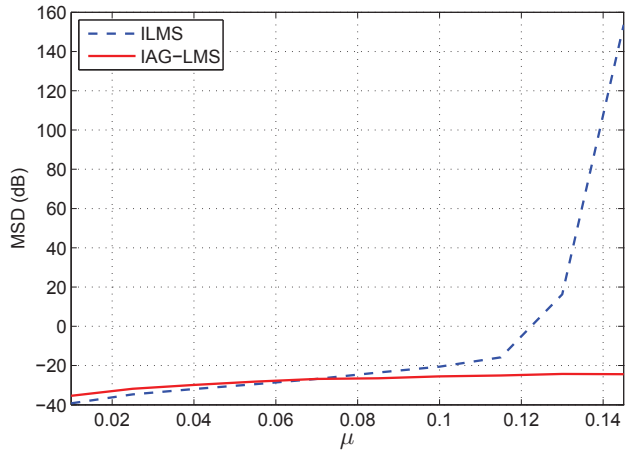

Fig. 6. Steady-state performances of the ILMS algorithm and the proposed algorithm in terms of the step-size parameter.

V. Conclusions and Future Work

In this paper, we proposed an incremental adaptive network based on the aggregated gradient method. The main advantages of the proposed algorithm, (in comparison with the existing ILMS algorithm) are its steady-state performance error and its wider stability range. However, the cost of this improvement is a slight increase in computation and communication per integration per node. In addition, the convergence rate for the proposed algorithm is slower than ILMS algorithm, which in turn affects its tracking performance spatially in fast-changing environment. We presented some simulation results to support our claims in this paper. This work can be extended by considering the effects of noisy links, which is a more practical assumption than the ideal links. Moreover, we can further improve the performance of the proposed algorithm with tuning the step-size parameter at every node according to its measurement quality. Our future work will address the mentioned issues.

References

[1] N. Bulusu and S. Jha, *Wireless Sensor Networks: A Systems Perspective*. Artech House, 2005.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Commun. Mag.*, vol. 14, no. 6, pp. 32–39, 2007.

[3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2508–2530, June 2006.

[4] I. Schizas, G. Mateos, and G. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2365–2382, 2009.

[5] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 674–690, 2011.

[6] L. Xiao, S. Boyd, and S. Lai, "A space-time diffusion scheme for peer-to-peer least-squares estimation," in *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, 2006, pp. 168–176.

[7] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, August 2012.

[8] A. Nedic and A. Ozdaglar, *Cooperative distributed multi-agent optimization*, Y. Eldar and D. Palomar, Eds. Cambridge, U. K.: Cambridge Univ. Press, 2009.

 ISBN: 978-1-61208-347-6

[9] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *Automatic Control, IEEE Transactions on*, vol. 31, no. 9, pp. 803–812, 1986.

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

[11] A. H. Sayed, "Diffusion adaptation over networks," in *E-Reference Signal Processing*, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2013, to appear.

[12] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. Towfic, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.

[13] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4064–4077, August 2007.

[14] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. Asilomar Conf. Signals, Systems, Computers*, Monterey, CA, October 2006, pp. 233–237.

[15] N. Takahashi and I. Yamada, "Incremental adaptive filtering over distributed networks using parallel projection onto hyperslabs," *IEICE Technical Report*, vol. 108, pp. 17–22, 2008.

[16] F. Cattivelli and A. H. Sayed, "Analysis of spatial and incremental LMS processing for distributed estimation," *IEEE Trans. on Signal Process.*, vol. 59, no. 4, pp. 1465–1480, April 2011.

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

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

[19] F. S. Cattivelli and A. H. Sayed, "Multilevel diffusion adaptive networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Taipei, Taiwan, April 2009.

[20] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*. Lexington, MAMIT Lincoln Lab., Lexington, MA: MIT Lincoln Lab, June 2006.

[21] L. Li, J. A. Chambers, C. G. Lopes, and A. H. Sayed, "Distributed estimation over an adaptive incremental network based on the affine projection algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 151–164, January 2010.

[22] C. G. Lopes and A. H. Sayed, "Randomized incremental protocols over adaptive networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Dallas, TX, March 2010, pp. 3514–3517.

[23] A. Khalili, M. A. Tinati, and A. Rastegarnia, "Amplify-and-forward scheme in incremental lms adaptive network with noisy links: Minimum transmission power design," *AEU - International Journal of Electronics and Communications*, vol. 66, no. 3, pp. 262 – 265, 2012.

[24] A. Rastegarnia and A. Khalili, "Incorporating observation quality information into the incremental lms adaptive networks," *Arabian Journal for Science and Engineering*, vol. 39, no. 2, pp. 987–995, 2014.

[25] D. Blatt, A. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 29–51, 2007.

 ISBN: 978-1-61208-347-6