A Distributed Target Localization Algorithm for Mobile Adaptive Networks

Amin Lotfzad Pak[1], **Amir Rastegarnia**[2*], **Azam Khalili**[3], and **Md Kafiul Islam**[4], Non-members

ABSTRACT

Adaptive networks with mobile nodes possess the distributed adaptation abilities in addition to the collective patterns of motion. Thus, mobile adaptive networks have been used in new applications such as modeling the biological networks and source localization. The original mobile adaptive networks needs full cooperation among the neighbor nodes meaning that each node gathers the information from all of its neighbour nodes. This strategy requires large amount of communications per iteration per node. To address this problem, in this paper we propose a mobile diffusion adaptive network with selective cooperation. In the proposed algorithm each node selects a subset of its neighbors so that its steady-state performance be as close as possible to the traditional mobile diffusion network. Since the selective cooperation reduces the learning rate we also use affine projection algorithm (APA) as the learning rules at the nodes. Our simulation results reveal that the proposed algorithm is able to achieve the same mean-square deviation (MSD) as the original mobile adaptive network but with a lower communication per iteration per node.

Keywords: Mobile adaptive network, Distributed estimation, Selective cooperation, Target localization.

1. INTRODUCTION

One of the important trends of the current technical developments is to connect smart devices to each other and create highly complex systems. Such development can be observed in many areas such as sensor networks [1–3], peer-to-peer (P2P) networks [4], fog computing systems [5], robot swarms [6–8], and smart grids [9, 10]. Most of the these complex systems have a commonality: they consist of a multitude of separate agents (nodes) that are connected only with their neighbors, perform decentralized processing to achieve a global objective by relying on local information. The global objective, in most application is to estimate a desired parameter, namely $\mathbf{w}^o \in \mathbb{R}^M$, $(M \geq 1)$. 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. Different methods/algorithms have been reported in the literature to solve the distributed learning problems including consensus strategies [11, 12] and adaptive networks. The main problem of consensus solutions is that do not allow the network to undertake a continuous learning and optimization [13] which motivated the development of adaptive networks.

An adaptive network is a collection of nodes with adaptation and learning abilities that interact with each other to solve estimation and inference tasks in real-time [14]. Two major classes of adaptive networks are incremental strategy [15–26] and diffusion strategy [27–35]. Incremental networks need a Hamiltonian cycle among the nodes, i.e., a cyclic path that covers all nodes in the network, which is generally difficult to enforce since determining a Hamiltonian cycle is an NP-hard problem. In addition, cyclic trajectories are not robust to node or link failure [36]. In comparison, diffusion strategies are scalable, robust, and able to match well the performance of incremental networks. It is shown in [37] that diffusion-based adaptive network leads to faster convergence and lower mean-square deviation than consensus strategies. Moreover, diffusion least-mean-squares (LMS) strategies for distributed estimation can deliver lower excess-mean-square-error (EMSE) than a centralized solution [38]. Thus, in this work we design our proposed algorithm based on the diffusion strategy.

Diffusion LMS algorithms consist of two steps including the adaptation step, where the node updates the weight estimate using local measurement data, and the combination step where the information from the neighbors are aggregated. Based on the order of these two steps, diffusion algorithms can be categorized into two classes known as the Combine-then-Adapt and Adapt-then-Combine (ATC). It is observed that the ATC version of diffusion LMS outperforms the Combine-then-Adapt (CTA) algorithm

Manuscript received on May 23, 2016 ; revised on June 3, 2016.

[1] The author is with Faculty of Electrical Enginering, University of Tabriz, Tabriz, 51664, Iran, Email: lotfzad89@ms.tabrizu.ac.ir

[2] The author is with Department of Electrical Engineering, Malayer University, Malayer, 65719-95863, Iran, Email: a_rastegar@ieee.org

[3] The author is with Department of Electrical Engineering, Malayer University, Malayer, 65719-95863, Iran, Email: a.khalili@ieee.org

[4] The author is with Department of Electrical and Computer Engineering, National University of Singapore, Singapore-117583, Email: kafiul_islam@nus.edu.sg

[39]. The ATC algorithm in its general form can be written as

$$\begin{cases} \boldsymbol{\psi}_{k,i} = f_a\big(\mathbf{w}_{k,i-1}\big) & \text{(Adaptation)},\\ \mathbf{w}_{k,i} = f_d\big(\boldsymbol{\psi}_{l,i}, l \in \mathcal{N}_k\big) & \text{(Combination)} \end{cases} \tag{1}$$

for some functions f_a and f_d. In (1) $\mathbf{w}_{k,i-1}$ denotes local estimate of node k at time $i-1$. Furthermore, $\mathcal{N}_k$ denotes the number of nodes directly connected to node k at time i, including itself. As we can see from (1) in adaptation step, node k uses its own data to update the weight estimate $\mathbf{w}_{k,i-1}$ to intermediate value $\boldsymbol{\psi}_{k,i}$. In the combination step, each node gathers the intermediate estimates $\boldsymbol{\psi}_{l,i},\ l \in \mathcal{N}_k$ combines them to obtain the updated weight estimate $\mathbf{w}_{k,i}$.

1.1 Motivations and contributions

In some applications such as gas leak detection, search and rescue operations, cleaning leftover mine fields, and robotic odor source localization we face with distributed target (source) localization with mobile nodes (mobile robots). The original adaptive networks in [27–35] can not be used to solve such problems as they did not incorporate the node mobility. In [40, 41] another dimension of complexity which is node mobility has been added to the diffusion networks. The resultant mobile adaptive networks perform two diffusion-based estimation tasks: one for estimating the location of a target and the other one for tracking the center of mass of the network. Incorporating the node mobility enables the resulting diffusion networks to use them in new applications such as modeling the various forms of sophisticated behavior exhibited by biological networks [42–44] and source localization [45–47].

We can conclude from (1) that in combination step, each node needs to gather the information (intermediate estimates $\boldsymbol{\psi}_{k,i}$) from all of its neighbour nodes. Specifically, required communication cost per iteration per node is $|\mathcal{N}_k - 1|M$ for real data and $2|\mathcal{N}N_k - 1|M$ for complex data. Thus, this step may require large amount of communications at each iteration for dense networks or for large M. In some applications, however, networks cannot afford large communication overhead. To address this problem, in this paper we propose a mobile diffusion adaptive network with selective cooperation. More specifically, in the proposed algorithm each node selects a subset of its neighbors so that its steady-state performance be as close as possible to the traditional mobile diffusion network. On the other hand, the selective cooperation reduces the learning rate of the resultant network. To improve the convergence rate of mobile diffusion network with selective cooperation we use affine projection algorithm at the nodes as the learning rules. Our simulation results reveal that the proposed algorithm is able to achieve the same mean-square deviation (MSD) as the original mobile adaptive network but with a lower communication per iteration per node. We believe that the proposed algorithm can be successfully applied to solve decentralized source localization with mobile nodes such as robotic odor source localization.

1.2 Paper organization and notation

The outline of the paper is as follows: In Section 2.we provide a brief overview on modeling of coordinated motion behavior of collective mobile nodes. The proposed algorithm is introduced in Section 3. . The numerical simulation results are provide in Section 4. Finally, concluding remarks are made in Section 5.

Throughout the paper we use boldface letters for matrices and vectors and small letters for scalars. The notation * is used to denote complex conjugation for scalars and complex-conjugate transposition for matrices. We use E to denote the expectation operator. Finally, $\|\mathbf{x}\|^2_{\boldsymbol{\Sigma}}$ represents the weighted norm notation defined as $\mathbf{x}^*\boldsymbol{\Sigma}\mathbf{x}$.

2. MOBILE ADAPTIVE NETWORKS WITH SELF-ORGANIZATION ABILITIES

To begin with, consider a collection of N nodes distributed over a 3-dimensional space. Let $\mathbf{x}_{k,i}$ denote the location vector of node k at time i relative to a global coordinate. The location of the center of mass at time i is denoted by $\mathbf{x}^g_i$ and is given by[1]

$$\mathbf{x}^g_i = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_{k,i} \tag{2}$$

Each node can update its location vector from $\mathbf{x}_{k,i}$ to $\mathbf{x}_{k,i+1}$ as

$$\mathbf{x}_{k,i+1} = \mathbf{x}_{k,i} + \Delta t\mathbf{v}_{k,i+1} \tag{3}$$

where Δt shows the time step and $\mathbf{v}_{k,i+1}$ is the velocity vector of the node. As (3) reveals, the velocity vector is an important factor in determining the updated location of the node. Other factors, in turn, influence the velocity vector such as [40]

- distance of node to the target,
- the desire to move in coordination with the other nodes,
- the desire to keep the certain distance from neighbors to avoid collisions.

As we continue we will see how to use these factors to adjust the velocity vectors at the nodes.

Remarks 1. *Since the objective of every node is to move toward the position of the target, then we can conclude that the center of gravity of the network also will approach the target as time evolves, i.e.*

$$\lim_{i\to\infty}\mathbf{x}^g_i \to \mathbf{w}^o$$

More details on mobile adaptive networks can be found in [40]

One way to adjust the velocity vectors is to use the position vectors as

$$\mathbf{v}_{k,i} = \alpha \frac{\mathbf{w}^o - \mathbf{x}_{k,i}}{\|\mathbf{w}^o - \mathbf{x}_{k,i}\|} \tag{4}$$

In (4), α is a positive scaling factor used to bound the speed in pursuing the target. Obviously to use (4) each node needs to know $\mathbf{w}^o$ beforehand which is not possible in practice. Instead, nodes can replace $\mathbf{w}^o$ by local estimates $\mathbf{w}_{k,i}$, which will be shown further ahead. Another factor considered in determining the motion of the nodes is to avoid collisions during the moving toward the target by keeping the certain distance say, r from neighbors i.e.

$$r - \varepsilon \le \|\mathbf{x}_{k,i} - \mathbf{x}_{l,i}\| \le r + \varepsilon, \quad l \in \mathcal{N}_k \backslash \{k\}$$

where ε is positive and small enough quantity. In order to maintain this certain distance by each node we define a cost function as

$$J_k(\mathbf{v}_{k,i}) = \sum_{l \in \mathcal{N}_k \backslash \{k\}} \Big[\|(\mathbf{x}_{k,i} + \Delta t\ \mathbf{v}_{k,i}) - (\mathbf{x}_{l,i} + \Delta t\ \mathbf{v}_{l,i})\| - r\Big]^2 \tag{5}$$

where $\mathbf{x}_{k,i} + \Delta t\ \mathbf{v}_{k,i}$ is the updated location of the node k, and $\mathbf{x}_{l,i} + \Delta t\ \mathbf{v}_{l,i}$ shows the updated location of the neighbors of the node k. Minimizing (5) ensures that the safe distance between node k and its neighbor will be kept after updating the location of nodes. So to get the optimal $\mathbf{v}_{k,i}$ which minimizes (5), we differentiate $J_k(v_k)$ with respect to $\mathbf{v}_{k,i}$.

$$\frac{\partial J_k(\mathbf{v}_{k,i})}{\partial \mathbf{v}_{k,i}} = 2 \sum_{l \in \mathcal{N}_k \backslash \{k\}} \Big(\Delta t\ \mathbf{v}_k - \Delta t\ \mathbf{v}_{l,i} - (\mathbf{x}_{l,i} - \mathbf{x}_{k,i}) + r \frac{\mathbf{x}_{l,i} - \mathbf{x}_{k,i} + \Delta t\ (\mathbf{v}_{l,i} - \mathbf{v}_{k,i})}{\|\mathbf{x}_{l,i} - \mathbf{x}_{k,i} + \Delta t\ (\mathbf{v}_{l,i} - \mathbf{v}_{k,i})\|}\Big) \tag{6}$$

To derive $\mathbf{v}_{k,i}$ from (6) we consider the last term in (4). Fig. 1 depicts the current location of nodes k and l and their updated locations. The term $\Delta t\ (\mathbf{v}_{l,i} - \mathbf{v}_{k,i})$ denotes the misaligned distance of nodes k and l after the update relative to their displacement before the update. As the velocity of node k and its neighbor are close together the term $\Delta t(\mathbf{v}_{l,i} - \mathbf{v}_{k,i})$ is negligible. So we have

$$\frac{\mathbf{x}_{l,i} - \mathbf{x}_{k,i} + \Delta t\ (\mathbf{v}_{l,i} - \mathbf{v}_{k,i})}{\|\mathbf{x}_{l,i} - \mathbf{x}_{k,i} + \Delta t\ (\mathbf{v}_{l,i} - \mathbf{v}_{k,i})\|} \approx \frac{\mathbf{x}_{l,i} - \mathbf{x}_{k,i}}{\|\mathbf{x}_{l,i} - \mathbf{x}_{k,i}\|} \tag{7}$$

By substituting (7) in (6) and putting it equal to zero $\mathbf{v}_{k,i}$ is obtained as follows:

$$\mathbf{v}_{k,i} = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \backslash \{k\}} \Big[\mathbf{v}_{l,i} + (1 - \frac{r}{\|\mathbf{x}_{l,i} - \mathbf{x}_{k,i}\|}) \frac{\mathbf{x}_{l,i} - \mathbf{x}_{k,i}}{\Delta t}\Big] \tag{8}$$

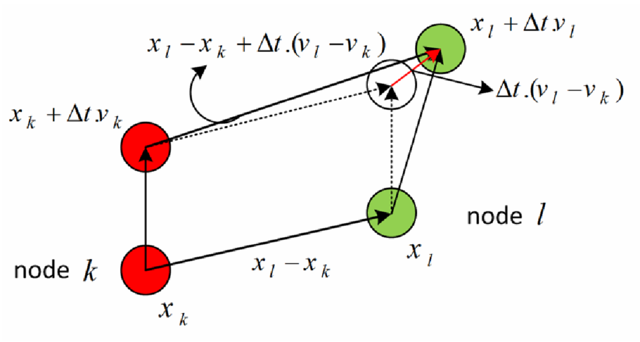

Fig.1: *Location of node l relative to node k before and after the update.*

First term of (8) denotes the average velocity of the neighbors of the node k. The second term represents a linear combination of the displacement vectors $\{\mathbf{x}_{l,i} - \mathbf{x}_{k,i}\}$ which suggests that the nodes should adjust their velocities to be consistent with average displacement vector in neighborhood while maintaining a distance r from their neighbors [40].

As we mentioned earlier, another factor influenced the velocity vector of nodes is the desire of the nodes to have a harmonic motion with other nodes. It means that the speed of each node is relevant with other nodes. The average velocity of nodes, which is defined as follows reflects the harmonic motion:

$$\mathbf{v}^g = \frac{1}{N} \sum_{l=1}^{N} \mathbf{v}_{l,i} \tag{9}$$

Obviously nodes cannot use this term directly to adjust their velocity vectors because they do not have access to all velocities $\{\mathbf{v}_{l,i}\}$ across the network. Instead, the nodes replace by local estimates $\mathbf{v}^g_{k,i}$. Thus, using (4), (8) and (9) we modify the velocity vectors as:

$$\mathbf{v}_{k,i+1} = \alpha \frac{\mathbf{w}^o - \mathbf{x}_{k,i}}{\|\mathbf{w}_{k,i} - \mathbf{x}_{k,i}\|} + \beta \mathbf{v}^g_{k,i} + \gamma \boldsymbol{\delta}_{k,i} \tag{10}$$

where $\{\beta, \gamma\}$ are non-negative weighting factors. We have

$$\boldsymbol{\delta}_{k,i} = \frac{1}{|\mathcal{N}_k| - 1} \sum_{l \in \mathcal{N}_k \backslash \{k\}} \left(1 - \frac{r}{\|\mathbf{x}_{l,i} - \mathbf{x}_{k,i}\|}\right) (\mathbf{x}_{l,i} - \mathbf{x}_{k,i}) \tag{11}$$

Note that in (10) the local estimation of $\{\mathbf{w}^o, \mathbf{v}^g\}$ is used instead of their global quantity. It means we will use $\{\mathbf{w}_{k,i}, \mathbf{v}^g_{k,i}\}$ which will be shown later.

Finally to estimate $\mathbf{w}^o$ which is the main objective of the network, each node needs to know the distance and direction relative to the target. Each node by

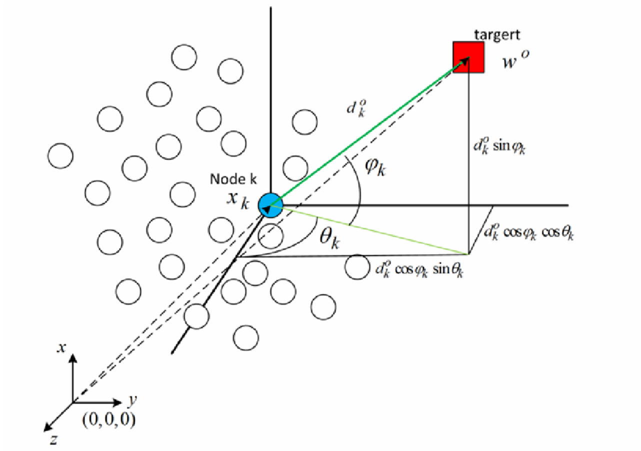

Fig.2: *The model of system in $\mathbb{R}^3$ denotes the location of a node and its distance from target, d_k^o.*

having this data and observations, and considering a origin coordinate for all nodes, can estimate the location of the target relative to the origin coordinate. Also, we assume that every node knows its location relative to the defined global coordinate system. For better understanding and to obtain a mathematical model for this process we consider the Fig. 2. As this figure shows, the distance between a target located at $\mathbf{w}^o$ and a node k located at $\mathbf{x}_{k,i}$ at time i can be defined as the following inner product

$$d_k^o(i) = \mathbf{u}_{k,i}(\mathbf{w}^o - \mathbf{x}_{k,i}) \tag{12}$$

where $\mathbf{u}_{k,i}$ denotes the unit direction vector pointing to $\mathbf{w}^o$, and by using the azimuth angle, $\theta_k(i)$, and the elevation angle, $\varphi_k(i)$, this vector can be defined as follows

$$\mathbf{u}_{k,i} = [\cos\theta_k(i)\cos\varphi_k(i) \quad \sin\theta_k(i)\cos\varphi_k(i) \quad \sin\varphi_k(i)] \tag{13}$$

It is assumed that the measurement of distance to target is noisy, so we have

$$d_k(i) = \mathbf{u}_{k,i}(\mathbf{w}^o - \mathbf{x}_{k,i}) + n_k(i) \tag{14}$$

In (14), $n_k(i)$ denotes the additive noise. Using Fig. 2, we can derive a linear regression model by rearranging the $n_k(i)$ as

$$\hat{d}_k(i) = d_k(i) + \mathbf{u}_{k,i}\mathbf{x}_{k,i} = \mathbf{u}_{k,i}\mathbf{w}^o + n_k(i) \tag{15}$$

Equation (15) represents the linear relation between $\hat{d}_k(i)$ and $\mathbf{u}_{k,i}$. The local estimation of $\mathbf{w}^o$ by diffusion mechanisms and in a distributed manner by each node will be described in following. First we assume that every node k at each time instant i has access to $\{\hat{d}_k(i), \mathbf{u}_{k,i}\}$. Using these local data and the data given by other nodes in its adjacent, to estimate the target position in a distributed manner, each node desires to minimize the following cost function

$$J_w^{glob}(\omega) = \sum_{k=1}^{N} \mathsf{E}[\hat{d}_k(i) - \mathbf{u}_{k,i}\mathbf{w}]^2 \tag{16}$$

Individual nodes cannot optimize (16) because they do not have access to the data across all nodes. To solve this problem, we can apply the ATC diffusion algorithm that was developed in [39]. The adaptation and combination steps for the ATC algorithm is given by

$$\begin{aligned} \boldsymbol{\psi}_{k,i} &= \mathbf{w}_{k,i-1} + \mu_k \sum_{l\in\mathcal{N}_k} c_{l,k}^w \mathbf{u}_{l,i}^{\mathsf{T}}[\hat{d}_l(i) - \mathbf{u}_{l,i}\mathbf{w}_{k,i-1}] \\ \mathbf{w}_{k,i} &= \sum_{l\in\mathcal{N}_k} \boldsymbol{\psi}_{l,i} \end{aligned} \tag{17}$$

where μ_k is the step-size used by node k and $\{a_{l,k}^w, c_{l,k}^w\}$ are set of non-negative real coefficients which satisfy the following properties

$$\begin{aligned} \sum_{l=1}^{N} c_{l,k}^w &= \sum_{l=1}^{N} a_{l,k}^w = 1 \\ c_{l,k}^w &= a_{l,k}^w = 0 \quad \text{if} \quad l \notin \mathcal{N}_k \end{aligned} \tag{18}$$

To use (3) and (10), we need to estimate the velocity of the center gravity, $\mathbf{v}^g$, in a distributed manner. Since the velocities of nodes are changing in time, we need to keep track of $\mathbf{v}^g$ over time. So we introduce the global cost function as follows:

$$J_v^{glob}(\mathbf{v}^g) = \sum_{k=1}^{N} \mathsf{E}\,\|\mathbf{v}_{k,i} - \mathbf{v}^g\|^2 \tag{19}$$

Following the derivation in [40], one can arrive at the following diffusion algorithm for estimating $\mathbf{v}^g$

$$\begin{aligned} \boldsymbol{\phi}_{k,i} &= \mathbf{v}_{k,i-1}^g + \lambda_k \sum_{l\in\mathcal{N}_k} c_{l,k}^v(\mathbf{v}_{l,i} - \mathbf{v}_{k,i-1}^g) \\ \mathbf{v}_{k,i}^g &= \sum_{l\in\mathcal{N}_k} a_{l,k}^v \boldsymbol{\phi}_{l,i} \end{aligned} \tag{20}$$

where λ_k is a positive step size used by node k and $\{a_{l,k}^w, c_{l,k}^w\}$ are set of non-negative real coefficients satisfying the same properties as (19).

3. PROPOSED ALGORITHM

In this section, we investigate the case where nodes cooperate only with subset of their neighbors. We present an algorithm in which the goal is to reduce the required communication cost per iteration per, while at the same time maintain the performance of the network.

3.1 Adaptive estimation with selective cooperation

As mentioned earlier, in the conventional diffusion strategies each node cooperates with all of its neighbors. In some applications, networks cannot afford large communication overhead due to energy consumption cost or bandwidth restrictions. In order to minimize the communication overhead for such applications, we need to introduce a new decision metric that selects only a subset of neighbours to consult at node k. To this end, we firstly consider a *stand alone* adaptive filter. In such a filter, the objective is to minimize the following function at every iteration i

$$\arg\min_{\mathbf{w}_i} \|\mathbf{w}_i - \mathbf{w}^o\| \tag{21}$$

where $\mathbf{w}_i$ is an estimation of $\mathbf{w}^o$ at time i. Since $\mathbf{w}^o$ is unknown we can use its most available estimate i.e. $\mathbf{w}_{i-1}$. In mobile diffusion networks (See (17)), each node requires the intermediate estimates of its neighbors. To reduce the amount of communications, we can select neighbors of node k that have small estimated variance product measure and ignore the other neighbors. Therefore for diffusion networks we define the decision metric as follows

$$t_k(i) = \left\|\boldsymbol{\psi}_{k,i} - \mathbf{w}_{k,i-1}\right\|^2 \tag{22}$$

Thus, based on (24) node k selects node l to update the weight vector if we have

$$t_l(i) \leq t_k(i), \quad l \in \mathcal{N}_k$$

The estimation processes for a diffusion mobile adaptive network with decision metric $t_k(i)$ is shown in Fig. 3. Note that we can further reduce the required communication if we define a decision metric for estimation of $\mathbf{v}^g$ which is required in the combination step of equation (20). So we define

$$s_k(i) = \left\|\boldsymbol{\phi}_{k,i} - \mathbf{v}^g_{k,i-1}\right\|^2 \tag{23}$$

By using this metric, some steps of the flowchart shown in Fig. 3 needs to be changed as we have shown in Fig. 4.

Remarks 2. *As we will show in the simulations, the selective cooperation causes a penalty on convergence speed of mean square deviation (MSD) curve of the network. This issue can be addressed by replacing the LMS learning rules by the affine projection algorithm (APA).*

3.2 Convergence rate improvement

To describe the mobile diffusion networks with selective cooperation and affine projection learning, we consider a stand alone filter with APA. In this algorithm the unknown vector $\mathbf{w}^o$ is estimated (in a stand alone filter) via the following update equation

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu\mathbf{U}_i^*(\varepsilon\mathbf{I} + \mathbf{U}_i\mathbf{U}_i^*)^{-1}\left[\mathbf{d}_i - \mathbf{U}_i\mathbf{w}_{i-1}\right] \tag{24}$$

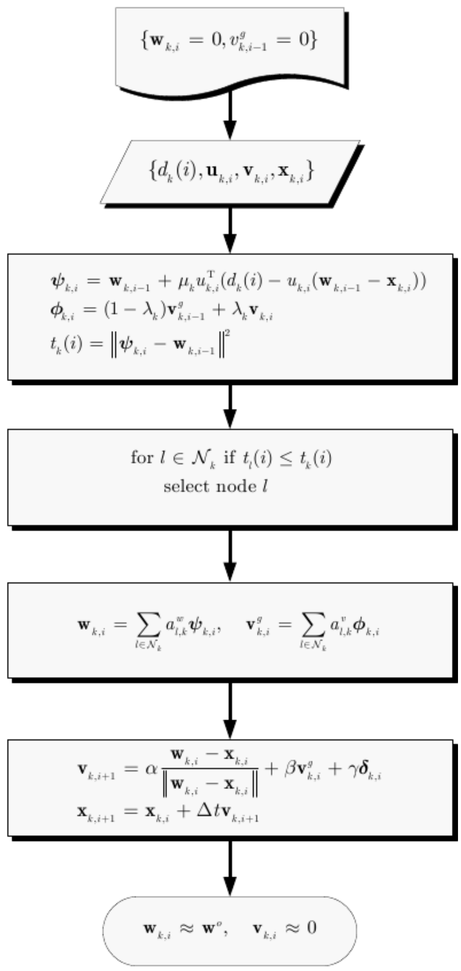

Fig.3: *Proposed algorithm for selective cooperation to estimate the position of a target in mobile adaptive network.*

where ε is small positive constant. The $\mathbf{U}_i$ and $\mathbf{d}_i$ are expressed as follows

$$\mathbf{U}_i \triangleq \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_{i-1} \\ \vdots \\ \mathbf{u}_{i-K+1} \end{bmatrix} \quad \mathbf{d}_i = \begin{bmatrix} d(i) \\ d(i-1) \\ \vdots \\ d(i-K+1) \end{bmatrix} \tag{25}$$

Specifically, we choose a positive integer K (usually $K \leq M$) and M is the size of the weight vector of the input signal.

In estimation the position of a target in mobile adaptive network when nodes get closer to the target the input signals of each node get colored so this cause the lower convergence rate in estimation process. To improve the convergence rate we are interested in using affine projection algorithm. Note that, since the computational complexity of APA is more than LMS, we just use the APA when nodes get closer enough to the target. In (17), in the intermediate estimation step we can apply APA instead of LMS algorithm so

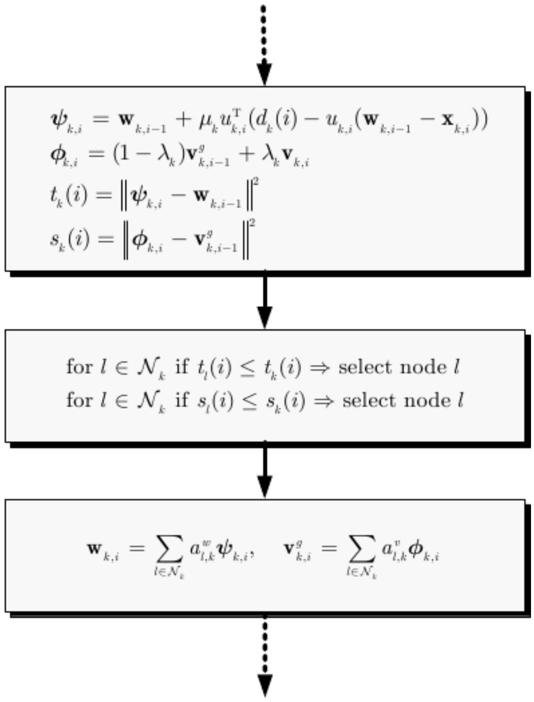

Fig.4: *Proposed algorithm while applied for both estimation processes.*

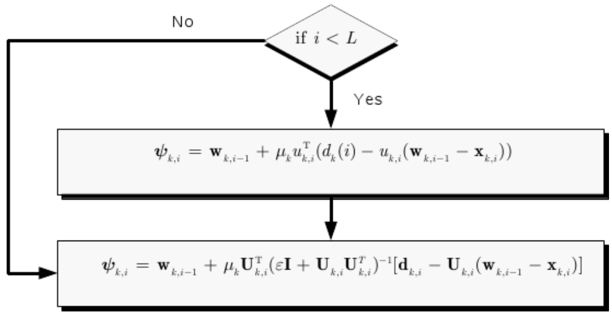

Fig.5: *Improved algorithm after using APA.*

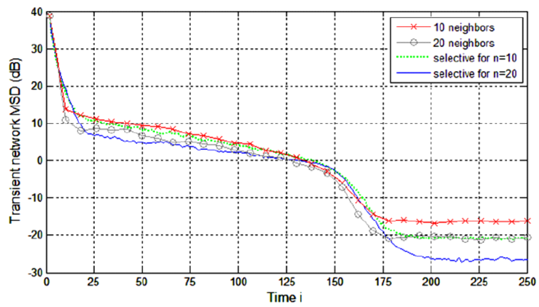

Fig.6: *Transient network MSD of the target position.*

the $\psi_{k,i}$ can be compute by APA as follows:

$$\psi_{k,i} = \mathbf{w}_{k,i-1} + \mu_k \mathbf{U}_{k,i}^{\mathsf{T}}(\varepsilon\mathbf{I} + \mathbf{U}_{k,i}\mathbf{U}_{k,i}^{\mathsf{T}})^{-1} \times [\mathbf{d}_{k,i} - \mathbf{U}_{k,i}(\mathbf{w}_{k,i-1} - \mathbf{x}_{k,i})] \tag{26}$$

where

$$\mathbf{U}_{k,i} \triangleq \begin{bmatrix} \mathbf{u}_{k,i} \\ \mathbf{u}_{k,i-1} \\ \vdots \\ \mathbf{u}_{k,i-K+1} \end{bmatrix} \quad \mathbf{d}_{k,i} = \begin{bmatrix} d_k(i) \\ d_k(i-1) \\ \vdots \\ d_k(i-K+1) \end{bmatrix} \tag{27}$$

According to the above remarks, the intermediate estimation of the position of the target step in the proposed algorithm represented in the Fig. 4, can be changed as will be shown in Fig. 5. We can see that for specified time ranged ($i < L$) LMS algorithm is used and for the rest ($i > L$) APA is applied for estimation.

4. SIMULATION RESULTS

In this section we present simulation results to evaluate the performance of our proposed algorithm. We use a network with $N = 100$ nodes that are uniformly distributed inside a cube with length 12. The simulation parameters are set as follows. The step sizes of updates are set to $\mu_k = \lambda_k = 0.5$ for all nodes at each iteration. The combination coefficients are set as $a_{l,k}^w = a_{l,k}^v = \frac{1}{\mathcal{N}_k(i)}$ if $l \in \mathcal{N}_k(i)$. The factors of velocity control are $\alpha = \beta = 0.5$ and $\gamma = 1$. We set the time duration to $\Delta t = 0.5$. Finally, optimal distance between two neighbours is set to $r = 3$. The observation noise $n_k(i)$ is assumed to be zero-mean Gaussian noise. The noise variance varies with the distance between the target and the node since the measurements are noisier at farther distance. Thus we use the following relation

$$\sigma_{n,k}^2(i) = 0.1\|\mathbf{w}^o - \mathbf{x}_{k,i}\|^2$$

Fig. 6 shows the network transient MSD for two different cooperation rules. In the first case, we assume that each node cooperates with specified number of its nearest neighbours at every iteration. To do so, we do the simulation, first for 10 nearest neighbours and then for 20 nearest neighbours. In the second case we only use subset of 10 and 20 nearest neighbours, so the number of neighbours will vary during each iteration. We can see from Fig. 6 that when we use the proposed selective cooperation algorithm the steady-state error gets better, but at the cost of penalty on convergence rate. For instance, Fig. 7 shows the transient MSD performance of 48th node with the number of neighbours that cooperate with the 48th node, for both full and selective cooperation cases. As we see, the proposed algorithm reduces the amount of communication overhead (number of nodes reduces) while gives a better steady- state performance. Table 1 shows the number of the communications exchanged among the nodes of the network for the usual cooperation rule and selective cooperation. Results show that in selective cooperation we have about 30 percent reduction in the amount of communications between nodes.

	Amount of communications	
	10 nearest neighbors	20 nearest neighbor
Full cooperation	173897	346522
Selective cooperation	105790	214466

Table 1: *Amount of communications between nodes for two different cooperation rules.*

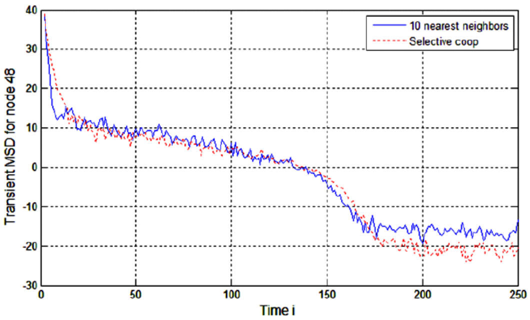

Fig.7: *Transient MSD of the target position for 48th node (top), and Number of neighbours that consult with node 48 at each iteration for both full and selective cooperation cases (bottom).*

As we mentioned earlier, the selective cooperation deteriorates the convergence rate of the network MSD. To solve this problem we can use AP algorithm instead of LMS algorithm in estimation process for part of time interval. Fig. 8 shows the simulation results when we apply the AP algorithm for part of time and compare it with the usual way when we just use LMS algorithm for target position estimation. As we can see from Fig. 8, for the same steady-state error, combing LMS and AP algorithms leads to better convergence rate in compared to the case when we just use LMS algorithm. In this simulation we used LMS for iteration $i < 170$ and then we applied AP algorithm. In Fig. 9 presents the results when we use APA to compensate for the low convergence speed caused by using selective cooperation.

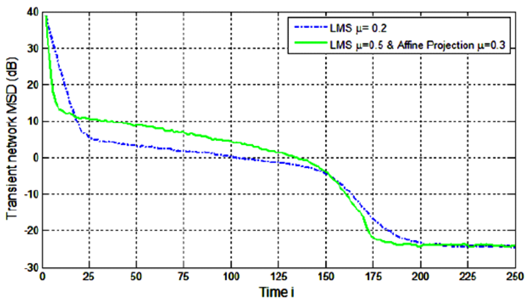

Fig.8: *Transient network MSD when AP has been applied.*

Finally, in Fig. 10 we illustrate the maneuver of mobile network in space over time for two different methods. Column (a) shows the motion of the nodes toward the target while using usual algorithm, and (b) represents the moving of the nodes while using

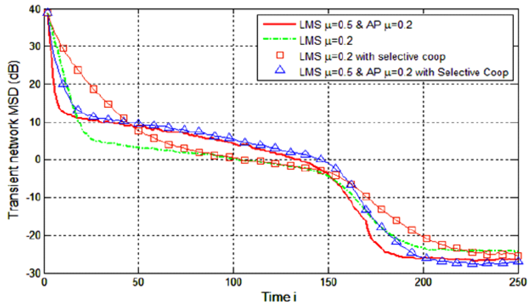

Fig.9: *Transient network MSD of proposed algorithm compared to the usual method.*

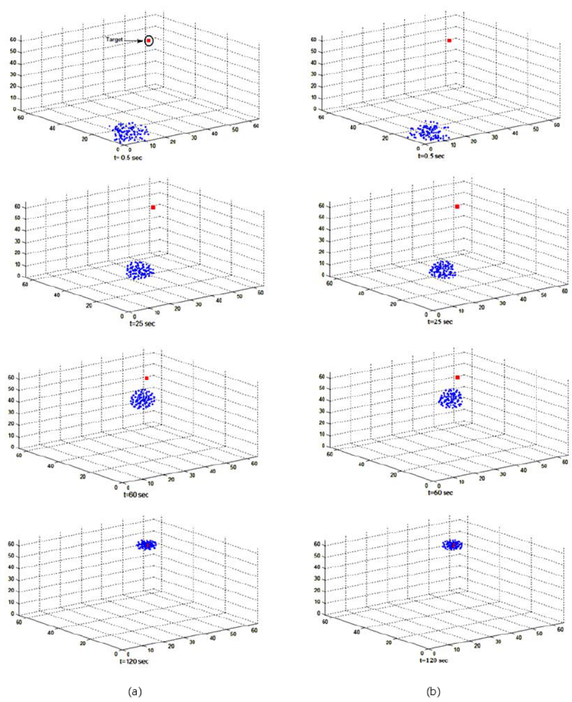

Fig.10: *Mode of movement of nodes towards the target in $\mathbb{R}^3$ over four different times for two different algorithm.*

proposed algorithms.

5. CONCLUSION

Recently, adaptive networks are being used in a wide range of contexts due to their real-time adaptation and learning abilities. Incorporating the node mobility enables the resulting diffusion networks to use them in new applications such as modeling the various forms of sophisticated behavior exhibited by biological networks and source localization. In this paper we proposed an algorithm for estimation of position of a target in a mobile network. In this algorithm, every node selects a subset of its neighbor that have to satisfy the given threshold. The proposed algorithm is able to provide steady-state performance near to ATC algorithm while it has reduced communication overhead. We presented some simulation results to clarify the discussions. The proposed algorithm can be used in practical applications such as odor source localization using multiple cooperative robots where we need to solve distributed source localization problem with mobile nodes. The current work assumes ideal connecting links among the nodes. In our future work we will develop mobile adaptive networks with selective cooperation in a practical scenario when the links are subject to channel noise.

References

[1] S. Sardellitti, and P. Di Lorenzo, Distributed detection and estimation in wireless sensor networks, in E-Reference Signal Processing, R. Chellapa and S. Theodoridis, Eds. Elsevier, 2013, pp. 329408.

[2] R. P. Yadav, P. V. Varde, P. S. V. Nataraj et al. Model-based tracking for agent-based control systems in the case of sensor failures, *International Journal of Automation and Computing*, 2012, 9(6): 561-569.

[3] Chao-Lei Wang, Tian-Miao Wang, Jian-Hong Liang et al. Bearing-only visual SLAM for small unmanned aerial vehicles in GPS-denied environments, *International Journal of Automation and Computing*, 2013, 10(5): 387-396.

[4] S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta, "Distributed data mining in peer-to-peer networks," *IEEE Internet Computing.*, vol. 10, no. 4, pp. 1826, 2006.

[5] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," *in Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM*, 2012, pp. 1316.

[6] J. McLurkin and D. Yamins, "Dynamic Task Assignment in Robot Swarms," *in Robotics: Science and Systems*, vol. 8, 2005.

[7] G. Kowadlo and R. Russell, "Robot odor localization: a taxonomy and survey," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 869894, Aug. 2008.

[8] A. Dhariwal, G. Sukhatme, and A. Requicha, "Bacterium-inspired robots for environmental monitoring," *in IEEE International Conference on Robotics and Automation*, 2004, pp. 14361443.

[9] S. Squartini, D. Liu, F. Piazza, D. Zhao, and H. He, "Computational Energy Management in Smart Grids," *Neurocomputing*, vol. 170, pp. 267 269, 2015.

[10] Y. Xia, S. C. Douglas, and D. P. Mandic, "Adaptive frequency estimation in smart grid applications: Exploiting noncircularity and widely linear adaptive estimators," *Signal Processing Magazine, IEEE*, vol. 29, no. 5, pp. 4454, 2012.

[11] 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.

[12] S. Kar and J. 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, Aug 2011.

[13] 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.

[14] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based lms for node-specific parameter estimation over adaptive networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 5425–5429.

[15] 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.

[16] S. Ram, A. Nedic, and V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*, Nov 2007, pp. 582–586.

[17] L. Li and J. A. Chambers, "A new incremental affine projection-based adaptive algorithm for distributed networks," *Signal Processing*, vol. 88, no. 10, pp. 2599 – 2603, 2008.

[18] 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.

[19] A. Khalili, M. Tinati, and A. Rastegarnia, "An incremental block lms algorithm for distributed adaptive estimation," in *Communication Systems (ICCS), 2010 IEEE International Conference on*, Nov 2010, pp. 493–496.

[20] M. Saeed and A. U. H. Sheikh, "A new lms strategy for sparse estimation in adaptive networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, Sept 2012, pp. 1722–1733.

[21] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based lms for node-specific parameter estimation over adaptive networks," in *Acoustics, Speech*

and Signal Processing (ICASSP), 2013 IEEE International Conference on, May 2013, pp. 5425–5429.

[22] W. Bazzi, A. Rastegarnia, and A. Khalili, "A quality-aware incremental lms algorithm for distributed adaptive estimation," *International Journal of Automation and Computing*, vol. 11, no. 6, pp. 676–682, 2014.

[23] Y. Liu and W. K. Tang, "Enhanced incremental LMS with norm constraints for distributed in-network estimation," *Signal Processing*, vol. 94, no. 0, pp. 373 – 385, 2014.

[24] A. Khalili, A. Rastegarnia, W. Bazzi, and Z. Yang, "Derivation and analysis of incremental augmented complex least mean square algorithm," *Signal Processing, IET*, vol. 9, no. 4, pp. 312–319, 2015.

[25] A. Rastegarnia, A. Khalili, W. Bazzi, and S. Sanei, "An incremental LMS network with reduced communication delay," *Signal, Image and Video Processing*, pp. 1–7, 2015.

[26] W. M. Bazzi, A. Rastegarnia, and A. Khalili, A Quality-aware Incremental LMS Algorithm for Distributed Adaptive Estimation, International Journal of Automation and Computing, vol. 11, no. 6, pp. 676-682, 2015.

[27] 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.

[28] O. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed energy-aware diffusion least mean squares: Game-theoretic learning," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 5, pp. 821–836, Oct 2013.

[29] S. Kumar, A. Sahoo, and D. Acharya, "Block diffusion adaptation over distributed adaptive networks under imperfect data transmission," in *India Conference (INDICON), 2014 Annual IEEE*, Dec 2014, pp. 1–5.

[30] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *Signal Processing, IEEE Transactions on*, vol. 62, no. 2, pp. 472–484, Jan 2014.

[31] P. Di Lorenzo and S. Barbarossa, "Distributed least mean squares strategies for sparsity-aware estimation over gaussian markov random fields," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 5472–5476.

[32] F. Wen, "Diffusion lmp algorithm with adaptive variable power," *Electronics Letters*, vol. 50, no. 5, pp. 374–376, Feb 2014.

[33] Y. Zhu, A. Jiang, H. K. Kwan, and K. He, "Distributed sensor network localization using combination and diffusion scheme," in *Digital Signal Processing (DSP), 2015 IEEE International Conference on*, July 2015, pp. 1156–1160.

[34] R. Arablouei, S. Werner, K. Doanay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion {LMS} algorithm," *Signal Processing*, vol. 117, pp. 355 – 361, 2015.

[35] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, J. Arenas-Garca, and M. T. Silva, "Distributed estimation in diffusion networks using affine least-squares combiners," *Digital Signal Processing*, vol. 36, pp. 1 – 14, 2015.

[36] X. Zhao, S-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data" *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 34603475, July 2012.

[37] 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.

[38] X. Zhao and A. Sayed, "Performance limits for distributed estimation over lms adaptive networks," *Signal Processing, IEEE Transactions on*, vol. 60, no. 10, pp. 5107–5124, Oct 2012.

[39] F. Cattivelli and A. Sayed, "Diffusion lms strategies for distributed estimation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[40] S. Y. Tu and A. H. Sayed, "Mobile adaptive networks," *Selected Topics on Signal Processing*, vol. 5, no. 4, pp. 649–664, August 2011.

[41] M. Lin, M. Murthi, and K. Premaratne, "Mobile adaptive networks for pursuing multiple targets," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 3217–3221.

[42] F. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. Signal Process*, vol. 59, no. 5, pp. 2038–2051, May 2011.

[43] J. Li and A. H. Sayed, "Modeling bee swarming behavior through diffusion adaptation with asymmetric information sharing," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 18, January 2012.

[44] W. Ma, Z. Sun, J. Li, M. Song, and X. Lang, "An improved artificial bee colony algorithm based on the strategy of global reconnaissance," *Soft Computing*, pp. 1–33, 2015.

[45] J.-G. Li, Q.-H. Meng, Y. Wang, and M. Zeng, "Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm," *Autonomous Robots*, vol. 30, no. 3, pp. 281–292, 2011.

[46] A. Khalili, A. Rastegarnia, M. Islam, and Z. Yang, "A bio-inspired cooperative algorithm for distributed source localization with mobile

nodes," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, July 2013, pp. 3515–3518.

[47] A. Pak, S. Aghazadeh, A. Rastegarnia, and A. Khalili, "Target position estimation with mobile adaptive network with selective cooperation," in *Computer Architecture and Digital Systems (CADS), 2013 17th CSI International Symposium on*, Oct 2013, pp. 107–110.

Amin Lotfzad Pak received the M.E. degree from University of Tabriz in 2013. His research interests include wireless communication and networks, sensor network and adaptive filters.

Amir Rastegarnia completed his PhD degree in the electrical engineering at the University of Tabriz, Tabriz, Iran, in 2011. In 2011, he joined the Department of Electrical Engineering, Malayer University, as Assistant Professor. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive estimation, as well as signal processing for communications. He is a Member of IEEE.

Azam Khalili received the PhD degree in electrical engineering from the University of Tabriz, Tabriz, Iran, in 2011. In 2011, she joined the Department of Electrical Engineering, Malayer University, as Assistant Professor. Her current research interests are theory and methods for adaptive filtering, distributed adaptive estimation, as well as signal processing for communications. She is a Member of IEEE.

Md Kafiul Islam Md. Kafiul Islam was born in 1988 in Bangladesh where he is currently serving as Assistant Professor at dept. of EEE, independent University, Bangladesh. Before that he has received his PhD degree from Dept. of ECE, National University of Singapore in 2015. His B.Sc in EEE was awarded in 2008 from Islamic University of Technology, Dhaka, Bangladesh. His research interests include biomedical signal processing, neural engineering, Brain-Computer Interface