Independent University **Bangladesh (IUB)**

IUB Academic Repository

Electrical and Electronics Engineering Article

2016

A Self-Organizing Diffusion Mobile Adaptive Network for Pursuing a Target

Rastegarnia, Amir

Scientific & Academic Publishing

http://dir.iub.edu.bd:8180/handle/123456789/265

Downloaded from IUB Academic Repository

American Journal of Signal Processing 2016, 6(2): 25-31
DOI: 10.5923/j.ajsp.20160602.01

A Self-Organizing Diffusion Mobile Adaptive Network for Pursuing a Target

Amir Rastegarnia[1,*], Azam Khalili[1], Md Kafiul Islam[2]

[1]Department of Electrical Engineering, University of Malayer, Malayer, Iran
[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

Abstract In this paper we focus on designing self-organizing diffusion mobile adaptive networks where the individual agents are allowed to move in pursuit of an target. The well-known Adapt-then-Combine (ATC) algorithm is already available in the literature as a useful distributed diffusion-based adaptive learning network. However, in the ATC diffusion algorithm, fixed step sizes are used in the update equations for velocity vectors and location vectors. When the nodes are too far away from the target, such strategies may require large number of iterations to reach the target. To address this issue, we suggest two modifications on the ATC mobile adaptive network to improve its performance. The proposed modifications include (i) distance-based variable step size adjustment at diffusion algorithms to update velocity vectors and location vectors, (ii) to use a selective cooperation, by choosing the best nodes at every iteration, to reduce the number of communications. The performance of the proposed algorithm is evaluated by simulation tests where the obtained results show the superior performance of the proposed algorithm in comparison with the available ATC mobile adaptive network.

Keywords Adaptive networks, Mobile networks, LMS, Sensor networks

1. Introduction

Wireless sensor networks appear in many practical applications such as distributed sensing, intrusion detection and target localization [1-4]. In most of the aforementioned applications, nodes of a network collect data from environment and then process them collaboratively to estimate a desired parameter. Different strategies have been introduced in the literature to solve the distributed estimation problems including consensus strategies and adaptive networks [5, 6]. It has been shown in [7] that adaptive networks are more stable than consensus networks and they provide better steady-state error performance. So, in this paper we focus on adaptive network based solutions. We adopt the term adaptive networks from [8] to refer to a collection of nodes that interact with each other and function as a single adaptive entity that is able to track statistical variations of data in real-time.

Two major classes of adaptive networks are incremental strategy [9-14] and diffusion strategy [15-19]. In comparison, incremental algorithms require less communication among nodes of the networks while diffusion algorithms are scalable and more robust to link and node failure [20-22].

In general, diffusion based 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 neighbours 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 CTA algorithm [16]. The initial diffusion adaptive networks in [15-19] did not incorporate the node mobility. In [23-26], 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 modelling the various forms of sophisticated behaviour exhibited by biological networks [27, 28] and source localization [29, 30].

The current algorithms for mobile adaptive networks do not consider the distance to the target in their adaptation mechanisms. In other words, in the current algorithms every node in the network adjusts (updates) its velocity vector and location vector regardless of its distance to the target. When nodes are too far away from the target, such strategies may require large number of iterations to reach the target, which in turn, requires large amount of communications and computations. Thus, it is highly desirable that in a mobile adaptive network, the nodes incorporate the distance to the

* Corresponding author:
amir_rastegar2003@yahoo.com (Amir Rastegarnia)
Published online at http://journal.sapub.org/ajsp
Copyright © 2016 Scientific & Academic Publishing. All Rights Reserved

target information, e.g. using bigger step-sizes when they are too far from the target. To endow the ATC diffusion network with such ability, we firstly define a practical metric to describe the far field region as a region that is too far from the target. Note that the near filed is also defined as a region that includes the target. Then, according to the position of a node (inside or outside of the far field) different step size adjust mechanisms are applied. In general, as long as a node is inside the far field region, the step size parameter in update velocity vectors and location vectors are increased, whereas for a node outside of the far field the step size is iteratively reduced as it moves toward the target.

Moreover, in the combination step of diffusion LMS algorithm, each node needs to gather the intermediate estimates from all of its neighbours. Thus, this step may require large amount of communications for dense networks. In some applications, however, networks cannot afford large communication overhead. To further reduce the communication load, a selective cooperation is used where every nodes selects only a subset of its niobous to share the information. The performance of the proposed algorithm is evaluated by simulation tests where the obtained results show the superior performance of the proposed algorithm in comparison with the available ATC adaptive network.

Notation: We use boldface letters for matrices and vectors and small letters for scalars. The notation $\| \mathbf{x} \|^2 = \mathbf{x}^* \mathbf{x}$ stands for the Euclidean norm of $\mathbf{x}$. E denotes the expectation operator. The set of neighbours of node k, including itself, is denoted by $\mathcal{N}_k$.

2. Mobile Adaptive Networks

Let us consider a network with N mobile nodes that are randomly distributed over a space. Let $\mathbf{x}_{k,i}$ be the location of node k at time i, relative to some global coordinate system and $\mathbf{w}^o \in \mathbb{R}^3$ denote the location of the target. Each node k finds its neighbours within a range R radius in each time i, i.e.

$$j \in \mathcal{N}_k \quad \text{if} \quad \| \mathbf{x}_{j,i} - \mathbf{x}_{k,i} \| \le R \tag{1}$$

The objectives of each node in the network are (i) to estimate the position of the desired target and move towards it in coherence and synchrony with the other nodes, and (ii), avoid possible collisions by keeping a certain distance, say r from neighbours during the motion to the target location. The distance between the target $\mathbf{w}^o$ and a node k at any time i is given by (See Fig. 1)

$$d_{k,i}^o = \mathbf{u}_{k,i} (\mathbf{w}^o - \mathbf{x}_{k,i}) \tag{2}$$

where $\mathbf{u}_{k,i}$ denotes the direction of the target including the azimuth angle $\theta_{k,i}$, and the elevation angle $\varphi_{k,i}$ which is given as

$$\mathbf{u}_{k,i} = [\cos(\theta_{k,i})\cos(\varphi_{k,i}) \;\; \sin(\theta_{k,i})\cos(\varphi_{k,i}) \;\; \sin(\varphi_{k,i})] \tag{3}$$

We assume that each node observes a noisy measurement of the distance to the target, via a linear model as follows

$$d_{k,i}^n = \mathbf{u}_{k,i} (\mathbf{w}^o - \mathbf{x}_{k,i}) + n_{k,i} \tag{4}$$

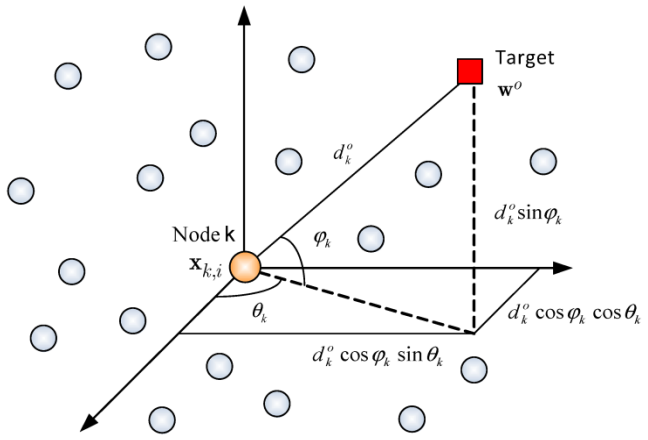


Figure 1. Distance and direction of the target $\mathbf{w}^o$ from node k at location $\mathbf{x}_{k,i}$

where $n_{k,i}$ denotes the noise term which is assumed to be zero-mean Gaussian noise. Intuitively, the noise variance, $\sigma_{n,k}^2(i)$ can follow a relation such as

$$\sigma_{n,k}^2(i) \approx \kappa \| \mathbf{x}_{k,i} - \mathbf{w}^o \|^2 \tag{5}$$

We select $\kappa = 0.01$ as [23]. Note that (5) is reasonable since we usually assume the signal power to decrease in proportional to the square of the propagation distance. We can rewrite the above equation as [23, 25]

$$d_{k,i} = d_{k,i}^n + \mathbf{u}_{k,i} \mathbf{x}_{k,i} = \mathbf{u}_{k,i} \mathbf{w}^o + n_{k,i} \tag{6}$$

At every time instant i, every node k has access to local data $\{d_{k,i}, \mathbf{u}_{k,i}\}$ and the local data from its neighbours. Using these data every node estimates the position of the target at $\mathbf{w}^o$ which can be achieved by solving the following optimization problem

$$\arg\min_{\mathbf{w}} \Big(\sum_{k=1}^{N} E\Big[\big| d_{k,i} - \mathbf{u}_{k,i} \mathbf{w} \big|^2 \Big] \Big) \tag{7}$$

The ATC diffusion algorithm has been developed for solving (7) in a distributed manner. The ATC algorithm consists of two steps: adaptation and combination. In adaptation step, node k uses its own data to update the weight estimate $\mathbf{w}_{k,i-1}$ to intermediate value $\mathbf{m}_{k,i}$. In the combination step each node gathers the intermediate estimates $\{\mathbf{m}_{\ell,i}\}, \ell \in \mathcal{N}_k$ combines them to obtain the updated weight estimate $\mathbf{w}_{k,i}$. The algorithm is described as follows:

$$\mathbf{m}_{k,i} = \mathbf{w}_{k,i-1} + \mu_k \sum_{\ell \in N_k} c^w_{\ell,k} \mathbf{u}^T_{\ell,i} \left(d_{\ell,i} - \mathbf{u}_{\ell,i} \mathbf{w}_{k,i-1} \right) \quad (8)$$

$$\mathbf{w}_{k,i} = \sum_{\ell \in N_k} a^w_{\ell,k} \mathbf{m}_{\ell,i} \quad (9)$$

where μ_k is the learning step size. The two sets of non-negative real coefficients $\{c^w_{\ell,k}\}$ and $\{a^w_{\ell,k}\}$ satisfy

$$\sum_{\ell=1}^{N} c^w_{\ell,k} = \sum_{\ell=1}^{N} a^w_{\ell,k} = 1, \quad c^w_{\ell,k} = a^w_{\ell,k} = 0, \ \ell \notin \mathcal{N}_k \quad (10)$$

In a network of mobile nodes, every node k can update its location as

$$\mathbf{x}_{k,i} = \mathbf{x}_{k,i-1} + \mathbf{v}_{k,i} \Delta t \quad (11)$$

where Δt is the time step and $\mathbf{v}_{k,i+1}$ denoted the velocity of the node k. Every node adjusts its velocity vector according to the following expression [23]

$$\begin{aligned} \mathbf{v}_{k,i} &= \xi_1 \frac{\mathbf{w}_{k,i-1} - \mathbf{x}_{k,i-1}}{\left\| \mathbf{w}_{k,i-1} - \mathbf{x}_{k,i-1} \right\|} + \xi_2 \mathbf{v}^g_{k,i-1} \\ &+ \xi_3 \sum\nolimits_{l \in \mathcal{N}_k \setminus \{k\}} (\| \mathbf{x}_{l,i} - \mathbf{x}_{k,i} \| - r) \frac{\mathbf{x}_{l,i} - \mathbf{x}_{k,i}}{\| \mathbf{x}_{l,i} - \mathbf{x}_{k,i} \|} \end{aligned} \quad (12)$$

where ξ_1, ξ_2 and ξ_3 are non-negative weighting factors, and $\mathbf{v}^g_{k,i}$ is the local estimate for the global velocity of the center of gravity of the network which is designed to allow for coherent motion. To use (12) each node needs to estimate $\mathbf{v}^g_{k,i}$ in a distributed way. Since the velocities of nodes are changing in time, we need to keep track of $\mathbf{v}^g_{k,i}$ over time. So we introduce the global cost function as follows

$$\arg\min_{\mathbf{v}^g} \left(\sum_{k=1}^{N} E\left[\left| \mathbf{v}_{k,i} - \mathbf{v}^g \right|^2 \right] \right) \quad (13)$$

Similar to (8) we can arrive at the following diffusion algorithm for estimating $\mathbf{v}^g_{k,i}$

$$\mathbf{s}_{k,i} = \mathbf{v}^g_{k,i-1} + \delta_k \sum_{\ell \in N_k} c^v_{\ell,k} \left(\mathbf{v}_{\ell,i} - \mathbf{v}^g_{k,i-1} \right) \quad (14)$$

$$\mathbf{v}^g_{k,i} = \sum_{\ell \in N_k} a^v_{\ell,k} \mathbf{s}_{\ell,i} \quad (15)$$

where δ_k is a positive step size and $\{c^v_{\ell,k}\}$ and $\{a^v_{\ell,k}\}$ are two sets of non-negative real coefficients satisfying the same properties as (10).

3. Proposed Algorithm

3.1. Motivation for Current Work

In the existing algorithms for adaptive mobile networks, there is no specific strategy for going faster toward the target. In other words, in the existing algorithms although the algorithm is designed in such a way that the set of nodes can move towards the goal harmoniously, but they do not consider the distance to the target in their velocity and location update equations. To address this issue, instead of using fixed learning parameter μ_k in (8), we use variable step-size parameter which has the following conditions

- if node k at iteration i is too far from the target, $\mu_{k,i}$ should be increased. Note that to prevent algorithm divergence and movement control of the set of nodes, we have to select the upper limit for step-size parameters.
- if distance of node k at iteration i becomes less than a predefined value, $\mu_{k,i}$ should be iteratively decreased as nodes approach the target. In this case we consider a lower bound for $\mu_{k,i}$ to avoid slow convergence rate.

3.2. Algorithm Development

To begin with, we define the far field region as follows.

Definition 1: *By far field, we mean a region* $\mathcal{D} \in \mathbb{R}^3$ *that the nodes inside it are far from the target, or*

$$\| \mathbf{w}^o - \mathbf{x}_{k,i} \|^2 \gg s, \quad (16)$$

To use (16), every node needs to have the target position $\mathbf{w}^o$ which is not available. To have a practical metric, we use $\mathbf{w}_{k,i}$ as an estimate of $\mathbf{w}^o$ at iteration i and rewrite (16) as

$$\| \mathbf{w}_{k,i} - \mathbf{x}_{k,i} \|^2 \gg s, \quad (17)$$

For a node inside $\mathcal{D}$ (i.e. $\mathbf{x}_{k,i} \in \mathcal{D}$) a bigger step-size in the update equation is required to move in the direction of $\mathbf{w}^o$. So we replace $\mu_{k,i}$ in (8) as follows

$$\mu_{k,i} = \alpha \mu_{k,i-1} \quad (18)$$

with $\alpha > 1$. Obviously, in this case we have $\mu_{k,i} > \mu_{k,i-1}$. It should be noted that according to the recursive equation in (8), increasing the step size may lead to algorithm divergence. So, in order to avoid algorithm divergence we consider an upper bound for step sizes as

$$\mu_{k,i} = \begin{cases} \alpha \mu_{k,i-1} & \alpha \mu_{k,i-1} < \mu_{\max} \\ \mu_{\max} & \alpha \mu_{k,i-1} \geq \mu_{\max} \end{cases} \quad (\text{if } \mathbf{x}_{k,i} \in \mathcal{D}) \quad (19)$$

When $\mathbf{x}_{k,i} \notin \mathcal{D}$ we need to reduce the step size as nodes moves toward the target at $\mathbf{w}^o$. In this case the step size adaptation function can be given by

$$\mu_{k,i} = \beta \mu_{k,i-1} + \gamma e^2_k(i) \quad (20)$$

where $e_k(i) = d_k(i) - \mathbf{u}_{k,i}(\mathbf{w}_{k,i-1} - \mathbf{x}_{k,i-1})$. In this case $\mu_{k,i}$ becomes smaller as node approaches the target. To avoid slow convergence rate, we can consider a lower bound for step size as follows

$$\mu_{k,i} = \begin{cases} \alpha\mu_{k,i-1} & \beta\mu_{k,i-1} + \gamma e_k^2(i) > \mu_{\min} \\ \mu_{\min} & \beta\mu_{k,i-1} + \gamma e_k^2(i) \le \mu_{\min} \end{cases} \quad (\text{if } \mathbf{x}_{k,i} \notin \mathcal{D}) \quad (21)$$

As mentioned in the introduction, 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 this application, we must introduce a new metric that considers only a subset of neighbours to consult at node k. To this end, we select neighbours of node k that have small estimated variance product measure and ignore the other neighbours. One way is to change (1) as

$$j \in \mathcal{N}_k \ \text{ if } \ \| \mathbf{x}_{j,i} - \mathbf{x}_{k,i} \| \le R \ \text{ and } \ \sigma_{j,i}^2 \le \sigma_{k,i}^2 \quad (22)$$

Remark 1: To use (22) we need $\sigma_{k,i}^2$ which are unknown in general and for practical usage they must be estimated. At every node k, $\sigma_{k,i}^2$ can be estimated by time-averaging as

$$\hat{\sigma}_{k,i}^2 = \eta\hat{\sigma}_{k,i-1}^2 + (1-\eta)(d_k(i) - \mathbf{u}_{k,i}\mathbf{w}_{k,i-1}) \quad (23)$$

with a forgetting factor $0 < \eta < 1$.

Finally, using the introduced modifications we arrive at the proposed algorithm as given in the Table. 1.

Table 1. Proposed Diffusion Mobile Adaptive Network with Selective Cooperation

Every node k in the network performs the following steps for $i > 0$.

The node has access to the local data $\{d_k(i), \mathbf{u}_{k,i}, \mathbf{v}_{k,i}, \sigma_{n,k}^2\}$

If $\| \mathbf{w}_{k,i} - \mathbf{x}_{k,i} \|^2 \gg s$ then

$$\mu_{k,i} = \begin{cases} \alpha\mu_{k,i-1} & \alpha\mu_{k,i-1} < \mu_{\max} \\ \mu_{\max} & \alpha\mu_{k,i-1} \ge \mu_{\max} \end{cases} \quad (\text{if } \mathbf{x}_{k,i} \in \mathcal{D})$$

$$\mu_{k,i} = \begin{cases} \alpha\mu_{k,i-1} & \beta\mu_{k,i-1} + \gamma e_k^2(i) > \mu_{\min} \\ \mu_{\min} & \beta\mu_{k,i-1} + \gamma e_k^2(i) \le \mu_{\min} \end{cases} \quad (\text{if } \mathbf{x}_{k,i} \notin \mathcal{D})$$

Use (22) to find set of neighbour nodes of every node k (i.e. $\mathcal{N}_{k,i}$)
Compute the following local adaptation and criterion

$$\mathbf{m}_{k,i} = \mathbf{w}_{k,i-1} + \mu_k \sum_{\ell \in N_k} c_{\ell,k}^w \mathbf{u}_{\ell,i}^T \left(d_{\ell,i} - \mathbf{u}_{\ell,i}\mathbf{w}_{k,i-1} \right)$$

$$\mathbf{s}_{k,i} = \mathbf{v}_{k,i-1}^g + \delta_k \sum_{\ell \in N_k} c_{\ell,k}^v \left(\mathbf{v}_{\ell,i} - \mathbf{v}_{k,i-1}^g \right)$$

Perform two local combination steps using data from selected neighbours

$$\mathbf{w}_{k,i} = \sum_{\ell \in N_k} a_{\ell,k}^w \mathbf{m}_{\ell,i} \qquad \mathbf{v}_{k,i}^g = \sum_{\ell \in N_k} a_{\ell,k}^v \mathbf{s}_{\ell,i}$$

Update the node velocity and its location

$$\mathbf{v}_{k,i} = \xi_1 \frac{\mathbf{w}_{k,i-1} - \mathbf{x}_{k,i-1}}{\left\| \mathbf{w}_{k,i-1} - \mathbf{x}_{k,i-1} \right\|} + \xi_2 \mathbf{v}_{k,i-1}^g + \xi_3 \sum\nolimits_{l \in \mathcal{N}_k \setminus \{k\}} (\| \mathbf{x}_{l,i} - \mathbf{x}_{k,i} \| - r) \frac{\mathbf{x}_{l,i} - \mathbf{x}_{k,i}}{\| \mathbf{x}_{l,i} - \mathbf{x}_{k,i} \|}$$

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

As i evolves we have $\mathbf{w}_{k,i} \approx \mathbf{w}^o$, $\mathbf{v}_{k,i} \approx 0$

4. Simulation Results

In this section we present simulation results to evaluate the performance of our proposed algorithm. We use a network with $N = 50$ nodes that are initially uniformly distributed inside a cube with length 10. Their velocities are set at random directions and unit magnitude. The simulation parameters are set as follows. The factors of velocity control are $\xi_1 = 0.8$, $\xi_2 = 0.5$, $\xi_2 = 0.8$. We further set $\beta = 0.85, \gamma = 0.001$. The combination coefficients are set as $a_{l,k}^w = a_{l,k}^v = c_{l,k}^w = c_{l,k}^v = 1/|\mathcal{N}_k(i)|$ if $l \in \mathcal{N}_k(i)$. We set the time duration to $\Delta t = 0.5$ and $\delta = 0.5$. Moreover, optimal distance between two neighbours is set to $r = 2$. A node chooses nearest neighbours from neighbours within the radius $R = 6$. The observation noise $n_k(i)$ is assumed to be zero-mean Gaussian noise which is given by (5).

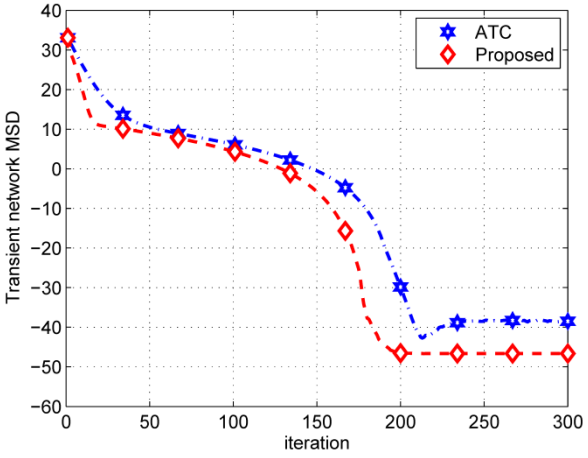


Figure 2. Transient network MSD for estimating the target location at $\mathbf{w}^o = [120\ 120]^T$

Fig. 2 shows the network transient mean-square deviation (MSD) (for two different algorithms) which is defined as

$$\text{MSD}_i = \frac{1}{N}\sum_{k=1}^{N} E[\|\mathbf{w}^o - \mathbf{w}_{k,i}\|^2] \qquad (24)$$

In the following, we compare the performance of the proposed algorithm with the ATC algorithm given in [25]. Note that for the ATC algorithm each node cooperates with fixed number of its nearest neighbours at every iteration ($\mathcal{N}_{k,i} = 4$). Moreover, each node uses fixed step-size $\mu_k = 0.5$ at every iteration. Fig. 2 shows the transient network MSD for estimating the target location at $\mathbf{w}^o = [120\ 120]^T$. These curves are averaged over 50 experiments with the same initial state of $\mathbf{w}_{k,-1}$ for all k . We observe that for both algorithms the transient MSD decreases dramatically in the first phase, then the network moves towards the target and finally, at steady state, the network arrives at the target. However, the proposed algorithm provides better performance compared to the ATC diffusion algorithm. Fig. 3 illustrates the maneuver of a mobile network (with ATC and proposed algorithms) over time.

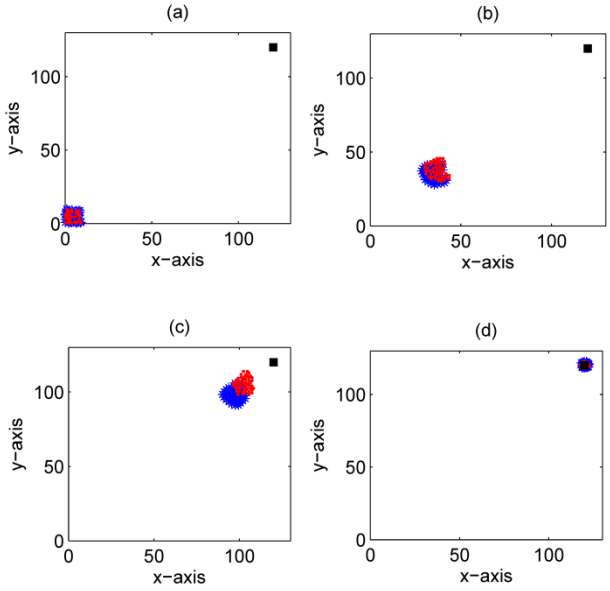


Figure 3. Maneuvers of mobile networks (with the ATC and the proposed algorithms) over time. (a) $i = 1$, (b) $i = 50$, (c) $i = 150$, and (d) $i = 300$. Note that * and • indicates the locations and moving directions of the nodes with ATC, proposed algorithm respectively and ■ denotes the location of the target

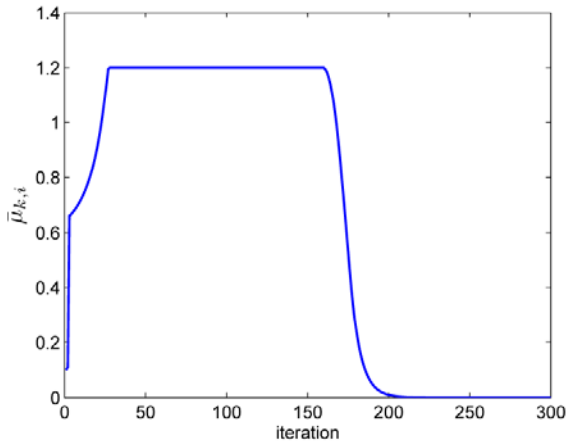


Figure 4. The network average step size $\bar{\mu}_{k,i}$ over iteration

We observe that both algorithms exhibits harmonious movement, but the proposed algorithm moves faster to the target. Fig. 4 shows how the network average step size, i.e.

$$\bar{\mu}_{k,i} = \frac{1}{N}\sum_{k=1}^{N} \mu_{k,i}$$

evolves over time. We can see that at the first iteration since the nodes are too far from the target, at every node the step size parameter has been increased iteratively so that after

some iterations we have $\mu_{k,i} = \mu_{\max}$. As the nodes move toward the target, at every node the step size parameter has been decreased iteratively.

Fig. 5 shows the average number of neighbours for every node. Note that for the ATC algorithm we have $\mathcal{N}_{k,i} = 4$ (according to our simulation setup) while the proposed algorithm uses small number of neighbours which means it requires less communication load.

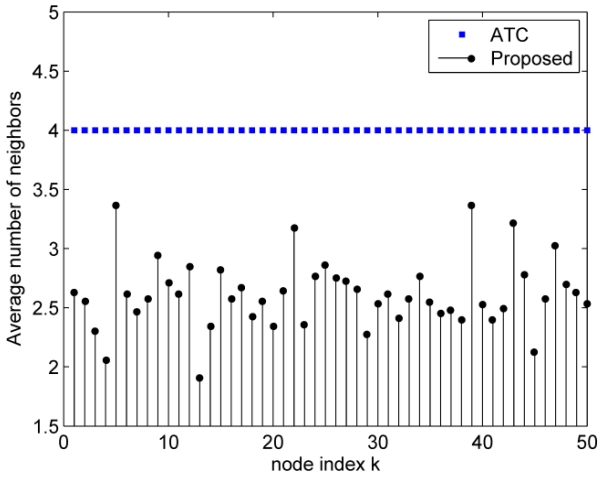


Figure 5. The average number of neighbours for every node

5. Conclusions

In this paper we proposed a modified ATC diffusion algorithm for mobile adaptive networks where the individual agents are allowed to move in pursuit of a target. The motivation was that in the ATC algorithm fixed step sizes are used in the update equations for velocity vectors and location vectors. When the nodes are too far away from the target, such strategies may require large number of iterations to reach the target. To address this issue, in the proposed algorithm we used distance-based variable step sizes for adjustment at diffusion algorithms to update velocity vectors and location vectors. We also used a selective cooperation where only a subset of nodes at each iteration is used to share information. The performance of the proposed algorithm was evaluated by simulation tests where the obtained results showed the superior performance of the proposed algorithm in comparison with the available ATC mobile adaptive network. In our future work we will develop diffusion mobile adaptive network for pursuing multiple targets.

REFERENCES

[1] D. Estrin, G. Pottie, and M. Srivastava, Instrumenting the world with wireless sensor networks, in Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Salt Lake City, UT, May 2001, pp. 2033-2036.

[2] H Leung, S Chandana, and S Wei, Distributed sensing based on intelligent sensor networks, IEEE Circuits and Systems, Mag., vol. 8, no. 2, pp. 38-52, May 2008.

[3] 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.

[4] 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.

[5] A. H. Sayed, Adaptation, learning, and optimization over networks, Foundations and Trends in Machine Learning, vol. 7, issue 4-5, pp. 311-801, NOW Publishers, Boston-Delft, 2014.

[6] A. H. Sayed, Adaptive networks, Proceedings of the IEEE, vol. 102, no. 4, pp. 460-497, April 2014.

[7] S.-Y. Tu and A. H. Sayed, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks, IEEE Trans. Signal Processing, vol. 60, no. 12, pp. 6217-6234, Dec. 2012.

[8] A. H. Sayed and C. Lopes, Distributed recursive least-squares strategies over adaptive networks, in: Proc. 40th Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, Oct. 2006, pp. 233-237.

[9] C. G. Lopes and A. H. Sayed, Incremental adaptive strategies over distributed networks, IEEE Trans. on Signal Process., vol. 55, no. 8, August 2007, pp. 4064-4077.

[10] 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.

[11] T. Panigrahi and G. Panda, Robust distributed learning in wireless sensor network using efficient step size, in 2nd International Conference on Communication Computing and Security at NIT Rourkela, India, 06-08 October 2012, Procedia Technology 6 (2012), pp. 864-871.

[12] G. Azarnia, M. A. Tinati, Steady-state analysis of the deficient length incremental LMS adaptive networks with noisy links, AEU - International Journal of Electronics and Communications, Volume 69, Issue 1, January 2015, Pages 153-162.

[13] A. Khalili, A Rastegarnia, WM Bazzi, Incremental augmented affine projection algorithm for collaborative processing of complex signals, 2015 International Conference on Information and Communication Technology Research (ICTRC), 60-63, 2015.

[14] 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.

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

[16] F. Cattivelli and A. H. Sayed, Diffusion LMS strategies for distributed estimation, IEEE Transactions on Signal

Processing, vol. 58, no. 3, pp. 1035-1048, March 2010.

[17] 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, Aug. 2012.

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

[19] X. Zhao and A. H. Sayed, Distributed clustering and learning over networks, IEEE Trans. Signal Process., vol. 63, no. 13, pp. 3285-3300, July 2015.

[20] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, Steady-state analysis of diffusion LMS adaptive networks with noisy links, IEEE Trans. Signal Process., vol. 60, no. 2, pp. 974-979, Feb. 2012.

[21] A. Rastegarnia, W. M. Bazzi, A. Khalili and J. A. Chambers, Diffusion adaptive networks with imperfect communications: link failure and channel noise, IET Signal Processing, vol. 8, no. 1, pp.59-66, 2014.

[22] X. Zhao, S.-Y. Tu, and A. H. Sayed, Diffusion adaptation over networks under imperfect information exchange and non-stationary data, IEEE Trans. on Signal Process., vol. 60, no. 7, pp. 3460-3475, July 2012.

[23] S.-Y. Tu and A. Sayed, Mobile adaptive networks with selforganization abilities, in Wireless Communication Systems (ISWCS), 2010 7th International Symposium on, Sept 2010, pp. 379-383.

[24] S.-Y. Tu and A. Sayed, Tracking behavior of mobile adaptive networks, in Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on, Nov 2010, pp. 698-702.

[25] S.-Y. Tu and A. Sayed, Mobile adaptive networks, Selected Topics on Signal Processing, vol. 5, no. 4, pp. 649-664, August 2011.

[26] May Zar Lin, Manohar N. Murthi and Kamal Premaratne, Mobile adaptive networks for pursuing multiple targets, in Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)}, Brisbane, Australia, 2015, pp. 3217-3221.

[27] 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.

[28] 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.

[29] 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.

[30] 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), 35th Annual International Conference of the IEEE, July 2013, pp. 3515-3518.

[31] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.