

Research Article

Linear High-Order Distributed Average Consensus Algorithm in Wireless Sensor Networks

Gang Xiong and Shaline Kishore

Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA

Correspondence should be addressed to Shaline Kishore, skishore@lehigh.edu

Received 23 November 2009; Revised 17 March 2010; Accepted 27 May 2010

Academic Editor: Husheng Li

Copyright © 2010 G. Xiong and S. Kishore. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a linear high-order *distributed average consensus* (DAC) algorithm for wireless sensor networks. The average consensus property and the convergence rate of the high-order DAC algorithm are analyzed. In particular, the convergence rate is determined by the spectral radius of a network topology-dependent matrix. Numerical results indicate that this simple linear high-order DAC algorithm can accelerate the convergence without additional communication overhead and reconfiguration of network topology.

1. Introduction

The distributed average consensus (DAC) algorithm aims to provide distributed nodes in a network agreement on a common measurement, known at any one node as the local state information. As such, it has many relevant applications in wireless sensor networks [1, 2], for example, moving-object acquisition and tracking, habitat monitoring, reconnaissance, and surveillance. In the DAC approach, average consensus can be sufficiently reached within a connected network by averaging pair-wise local state information at network nodes. In [1], Olfati-Saber et al. established a theoretical framework for the analysis of consensus-based algorithms.

In this paper, we study a simple approach to improve the convergence rate of DAC algorithms in wireless sensor networks. The author of [3] demonstrates that the convergence rate of DAC can be increased by using the “small-world” phenomenon. This technique, however, needs to redesign the network topology based on “random rewiring”. In [4], an extrapolation-based DAC approach is proposed; it utilizes a scalar epsilon algorithm to accelerate the convergence rate without extra communication cost. However, numerical results show that mean square error does not decrease monotonically with respect to iteration time, which

may not be desirable in practical applications. In [5], the authors extend the concept of average consensus to a higher dimension one via the spatial point of view, where nodes are spatially grouped into two disjoint sets: leaders and sensors. Specifically, it is demonstrated that under appropriate conditions, the sensors’ states converge to a linear combination of the leaders’ states. Furthermore, multi-objective optimization (MOP) and Pareto optimality are utilized to solve the learning problem, where the goal is to minimize the error between the convergence state and the desired estimate subject to a targeted convergence rate. In [6], the authors introduce the concept of nonlinear DAC algorithm, where standard linear addition is replaced by a sine operation during local state update. The convergence rate of this nonlinear DAC algorithm is shown to be faster under appropriate weight designs.

In this paper, we apply the principles of high-order consensus to the distributed computation problem in wireless sensor networks. This simple linear high-order DAC requires no additional communication overhead and no reconfiguration of the network topology. Instead, it utilizes gathered data from earlier iterations to accelerate consensus. We study here the convergence property and convergence rate of the high-order DAC algorithm and show that its convergence rate is determined by the spectral radius of

a network topology-dependent matrix. Moreover, numerical results indicate that the convergence rate can be greatly improved by storing and using past data.

This paper is outlined as follows. Section 2 provides background and system model for the high-order DAC algorithm. Section 3 discusses convergence analysis for this scheme. Simulation results are presented in Section 4, and conclusions are provided in Section 5.

2. Background and System Model

2.1. Linear High-Order DAC Algorithm. We assume a synchronized, time-invariant connected network. In each iteration of the M -th order DAC algorithm, each node transmits a data packet to its neighbor which contains the local state information. Each node then processes and decodes the received message from its neighbors. After retrieving the state information, each node updates its local state using the weighted average of the current state between itself and its neighboring nodes as well as stored state information from the $M - 1$ previous iterations of the algorithm.

The update rule of the M -th order DAC algorithm at each node i is given as

$$x_i(k) = x_i(k-1) + \varepsilon \sum_{m=0}^{M-1} c_m(-\gamma)^m \Delta x_i(k, m), \quad (1)$$

$$\Delta x_i(k, m) = \sum_{j \in \mathcal{N}_i} (x_j(k-m-1) - x_i(k-m-1)),$$

where $x_i(k)$ is the local state at node i during iteration k ; \mathcal{N}_i is the set of neighboring nodes that can communicate reliably with node i ; ε is a constant step size; c_m are predefined constants with $c_0 = 1$ and $c_m \neq 0 (m > 0)$; γ is a forgetting factor, such that $|\gamma| < 1$. We assume initial conditions of the M -th order DAC algorithm are $x_i(-M+1) = \dots = x_i(-1) = x_i(0) = \theta_i$, where θ_i is initial local state information for node i . It is worth mentioning that when $\gamma = 0$, the high-order DAC algorithm reduces to the (conventional) first-order DAC algorithm.

This linear high-order DAC algorithm can be regarded as a generalized version of DAC algorithm; it requires no additional communication cost and no reconfiguration of network topology. Compared to the conventional first-order DAC algorithm, with negligible increase in memory size and computation load in each sensor node, the convergence rate can be greatly improved with appropriate algorithm design. In [7], the authors propose an average consensus algorithm with improved convergence rate by considering a convex combination of conventional operation and linear predication. In particular, a special case of one step predication is presented for detailed analysis. We note that the major difference between the DAC algorithm in [7] and our proposed scheme is that we utilize stored state difference for high-order updating and show that optimal convergence rate can be significantly improved by this simple extension. Furthermore, we present explicitly the optimal convergence rate of second-order DAC algorithm in Section 3.2.

2.2. Network Model and Some Preliminaries. In the following, we model the wireless sensor network as an undirected graph (The convergence properties presented here can be easily extended for a directed graph. We omit this extension here.) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a set of N nodes $\mathcal{V} = \{1, 2, \dots, N\}$ and a set of edges \mathcal{E} . Each edge is denoted as $e = (i, j) \in \mathcal{E}$ where $i \in \mathcal{V}$ and $j \in \mathcal{V}$ are two nodes connected by edge e . We assume that the presence of an edge (i, j) indicates that nodes i and j can communicate with each other reliably. We assume here a connected graph, that is, there exists a path connecting any pair of distinct nodes.

Given this network model, we denote $A = [a_{ij}]$ as the adjacency matrix of \mathcal{G} such that $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Next, let L be the graph Laplacian matrix of \mathcal{G} which is defined as $L = D - A$, where $D = \text{diag}\{d_1, d_2, \dots, d_N\}$ is the degree matrix of \mathcal{G} , and $d_i = |\mathcal{N}_i|$. Given this matrix L , we have $L\mathbf{1} = \mathbf{0}$ and $\mathbf{1}^T L = \mathbf{0}^T$, where $\mathbf{1} = [1, 1, \dots, 1]^T$ and $\mathbf{0} = [0, 0, \dots, 0]^T$. Additionally, L is a symmetric positive semidefinite matrix. And for a connected graph, the rank of L is $N - 1$ and its eigenvalues can be arranged in increasing order as $0 = \lambda_1(L) < \lambda_2(L) \leq \dots \leq \lambda_N(L)$ [8].

Let us define $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_N(k)]^T$. The M -th order DAC algorithm in (1) thus evolves as

$$\mathbf{x}(k) = (I_N - \varepsilon L)\mathbf{x}(k-1) - \varepsilon \sum_{m=1}^{M-1} c_m(-\gamma)^m L\mathbf{x}(k-m-1), \quad (2)$$

with the initial conditions $\mathbf{x}(-M+1) = \dots = \mathbf{x}(-1) = \mathbf{x}(0) = \boldsymbol{\theta}$, where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N]^T$ and I_N denotes the $N \times N$ identity matrix.

3. Convergence Analysis of High-Order DAC Algorithm

3.1. Average Consensus Property of High-Order DAC Algorithm. Before we investigate the convergence property of the high-order DAC algorithm, we define two $MN \times MN$ matrices

$$H = \begin{bmatrix} I_N - \varepsilon L & c_1 \gamma \varepsilon L & \cdots & -c_{M-1}(-\gamma)^{M-1} \varepsilon L \\ I_N & \mathbf{0}_{N \times N} & \cdots & \mathbf{0}_{N \times N} \\ \vdots & \ddots & & \vdots \\ \mathbf{0}_{N \times N} & \cdots & I_N & \mathbf{0}_{N \times N} \end{bmatrix}, \quad (3)$$

$$J = \begin{bmatrix} K & \mathbf{0}_{N \times N} & \cdots & \mathbf{0}_{N \times N} \\ K & \mathbf{0}_{N \times N} & \cdots & \mathbf{0}_{N \times N} \\ \vdots & \ddots & & \vdots \\ K & \mathbf{0}_{N \times N} & \cdots & \mathbf{0}_{N \times N} \end{bmatrix},$$

where $K = (1/N)\mathbf{1}\mathbf{1}^T$, and $\mathbf{0}_{N \times N}$ denotes the $N \times N$ all-zero matrix. Then we have the following lemma:

Lemma 1. *The eigenvalues of $H - J$ agree with those of H except that $\lambda_1(H) = 1$ is replaced by $\lambda_1(H - J) = 0$.*

Proof. Let us define two $MN \times 1$ vectors $\mathbf{h}_l = (1/N)[\mathbf{1}^T \mathbf{0}^T \cdots \mathbf{0}^T]^T$ and $\mathbf{h}_r = [\mathbf{1}^T \cdots \mathbf{1}^T \mathbf{1}^T]^T$. It is easy to check that \mathbf{h}_l and \mathbf{h}_r are left and right eigenvectors of H corresponding to $\lambda_1(H) = 1$, respectively, that is, $\mathbf{h}_l^T H = \mathbf{h}_l^T$ and $H \mathbf{h}_r = \mathbf{h}_r$. Additionally, $J = \mathbf{h}_r \mathbf{h}_l^T$, $\mathbf{h}_l^T \mathbf{h}_r = \mathbf{h}_l^T \mathbf{h}_l = 1$. In order to obtain the eigenvalues of $H - J$, we have [9]

$$\begin{aligned} \det(H - J - \lambda I_{MN}) &= \det(H - \lambda I_{MN}) \left[1 - \mathbf{h}_l^T (H - \lambda I_{MN})^{-1} \mathbf{h}_r \right] \\ &= \left[\pm \prod_{i=1}^{MN} (\lambda_i(H) - \lambda) \right] \left(1 - \frac{\mathbf{h}_l^T \mathbf{h}_r}{1 - \lambda} \right) \\ &= \left[\pm \prod_{i=2}^{MN} (\lambda_i(H) - \lambda) \right] (-\lambda). \end{aligned} \quad (4)$$

The above equation is valid because

$$\mathbf{h}_r = (H - \lambda I_{MN})^{-1} (H - \lambda I_{MN}) \mathbf{h}_r = (H - \lambda I_{MN})^{-1} (1 - \lambda) \mathbf{h}_r. \quad (5)$$

Thus, the eigenvalues of $H - J$ are $\lambda_1(H - J) = 0$ and $\lambda_i(H - J) = \lambda_i(H)$, $i = 2, \dots, MN$. This completes the proof. \square

The average consensus property of the M -th order DAC algorithm in wireless sensor networks is stated in the following theorem.

Theorem 1. *Consider the M -th order DAC algorithm in (2) in a time-invariant, connected, undirected wireless sensor network, with initial conditions $\mathbf{x}(-M + 1) = \cdots = \mathbf{x}(-1) = \mathbf{x}(0) = \boldsymbol{\theta}$. When $\rho(H - J) < 1$, an average consensus is achieved asymptotically, or equivalently,*

$$\lim_{k \rightarrow \infty} x_i(k) = \frac{1}{N} \mathbf{1}^T \boldsymbol{\theta} = \frac{1}{N} \sum_{i=1}^N \theta_i, \quad \forall i \in \mathcal{V}, \quad (6)$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix.

Proof. Let us define $\boldsymbol{\psi}(k) = [\mathbf{x}(k)^T \mathbf{x}(k-1)^T \cdots \mathbf{x}(k-M+1)^T]^T$. Then, the M -th order DAC algorithm in (2) can be rewritten as $\boldsymbol{\psi}(k) = H \boldsymbol{\psi}(k-1)$, which implies that $\boldsymbol{\psi}(k) = H^k \boldsymbol{\psi}(0)$. To calculate the eigenvalues of H , we have [9]

$$\begin{aligned} \det(H - \lambda I_{MN}) &= \prod_{i=1}^N \left(\lambda^M - (1 - \varepsilon \lambda_i(L)) \lambda^{M-1} \right. \\ &\quad \left. + \varepsilon \sum_{m=1}^{M-1} c_m (-\gamma)^m \lambda_i(L) \lambda^{M-1-m} \right) = 0. \end{aligned} \quad (7)$$

Thus, the eigenvalues of H should satisfy the following equation:

$$\begin{aligned} f(\lambda) &= \lambda^M - (1 - \varepsilon \lambda_i(L)) \lambda^{M-1} \\ &\quad + \varepsilon \sum_{m=1}^{M-1} c_m (-\gamma)^m \lambda_i(L) \lambda^{M-1-m} = 0. \end{aligned} \quad (8)$$

Note that there are M roots corresponding to one $\lambda_i(L)$. For a time invariant and connected network, L has only one eigenvalue, $\lambda_1(L) = 0$. From (8), when $\lambda_1(L) = 0$, the eigenvalues of H satisfy $f(\lambda) = \lambda^M - \lambda^{M-1} = 0$. Then, for this $\lambda_1(L) = 0$, H has only two distinct eigenvalues, $\lambda_1(H) = 1$ (with algebraic multiplicity 1) and $\lambda_2(H) = 0$ (with algebraic multiplicity $M - 1$). Additionally, it is easy to show that the algebraic multiplicity of eigenvalue $\lambda(H) = 1$ is equal to 1. Based on Lemma 1, we know that the eigenvalues of $H - J$ agree with those of H except that $\lambda_1(H) = 1$ is replaced by $\lambda_1(H - J) = 0$. Since $\rho(H - J) < 1$, we see that the eigenvalues of H stay inside the unit circle except for $\lambda_1(H) = 1$. Thus, we have

$$\begin{aligned} \lim_{k \rightarrow \infty} H^k &= V \lim_{k \rightarrow \infty} \begin{bmatrix} 1 & \mathbf{0}_{1 \times (MN-1)} \\ \mathbf{0}_{(MN-1) \times 1} & \Lambda^k \end{bmatrix} V^{-1} \\ &= V \begin{bmatrix} 1 & \mathbf{0}_{1 \times (MN-1)} \\ \mathbf{0}_{(MN-1) \times 1} & \mathbf{0}_{(MN-1) \times (MN-1)} \end{bmatrix} V^{-1} \\ &= \mathbf{h}_r \mathbf{h}_l^T, \end{aligned} \quad (9)$$

where Λ is the Jordan form matrix corresponding to eigenvalues $\lambda_i(H) \neq 1$ [9]. Thus, we have $\lim_{k \rightarrow \infty} H^k = J$. Then, $\lim_{k \rightarrow \infty} \boldsymbol{\psi}(k) = J \boldsymbol{\psi}(0)$, which indicates

$$\lim_{k \rightarrow \infty} x_i(k) = \frac{1}{N} \mathbf{1}^T \boldsymbol{\theta}. \quad (10)$$

This completes the proof. \square

According to Theorem 1, we see that when this linear high-order DAC algorithm is employed in an undirected wireless sensor network, average consensus can be achieved asymptotically. We also note that our proposed high-order DAC algorithm relies heavily on local state information exchange between two or more nodes in the networks. Noisy links [10] and packet drop failures [11] will certainly affect the performance of our proposed high-order DAC algorithm. We will investigate these important issues in the future.

3.2. Convergence Rate for High-Order DAC Algorithm. One of the most important measures of any distributed, iterative algorithm is its convergence rate. As we show next, the convergence rate of the high-order DAC algorithm is determined by the spectral radius of $H - J$, which is similar to the first-order DAC algorithm [1].

Let us define the average consensus value in each iteration as $m(k) = (1/N) \mathbf{1}^T \mathbf{x}(k)$. In the high-order DAC algorithm, this value remains invariant during each iteration since

$$\begin{aligned} m(k) &= \frac{1}{N} \mathbf{1}^T \left[(I_N - \varepsilon L) \mathbf{x}(k-1) \right. \\ &\quad \left. - \varepsilon \sum_{m=1}^{M-1} c_m (-\gamma)^m L \mathbf{x}(k-m-1) \right] \\ &= m(k-1) = \cdots = m(0). \end{aligned} \quad (11)$$

We now define the disagreement vector as $\boldsymbol{\delta}(k) = \mathbf{x}(k) - m(k) \mathbf{1}$, which indicates the difference between the updated

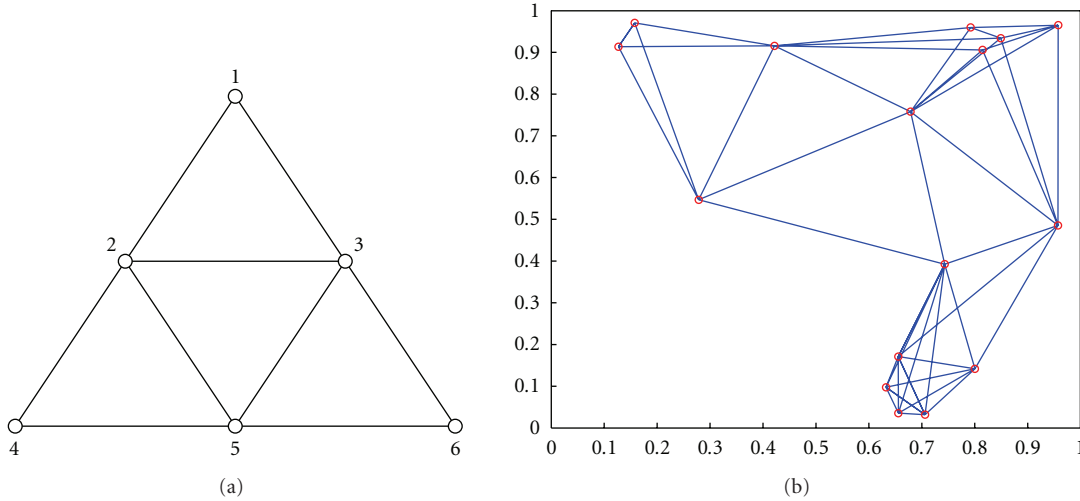


FIGURE 1: Network topologies used in numerical results: (a) fixed network with 6 nodes (Case 1) and (b) random network with 16 nodes (Case 2).

local state and the average state of the network nodes. Then, the evolution of the disagreement vector is obtained as.

$$\delta(k) = (I_N - \varepsilon L)\delta(k-1) - \varepsilon \sum_{m=1}^{M-1} c_m(-\gamma)^m L\delta(k-m-1). \quad (12)$$

Given this dynamic of the disagreement vector, we note.

Lemma 2. For the M -th order DAC algorithm in (2) in a time invariant, connected, undirected wireless sensor network, with initial conditions $\mathbf{x}(-M+1) = \dots = \mathbf{x}(-1) = \mathbf{x}(0) = \boldsymbol{\theta}$ and $\alpha = \rho(H - J) < 1$, an average consensus is exponentially reached in the following form:

$$\frac{\sum_{m=0}^{M-1} \|\delta(k-m)\|^2}{\|\delta(0)\|^2} \leq M\alpha^{2k}, \quad (13)$$

where $\|\cdot\|$ denotes the ℓ_2 norm of a vector.

Proof. Let us define the error vector as $\mathbf{e}(k) = [\delta^\top(k) \delta^\top(k-1) \dots \delta^\top(k-M+1)]^\top$ which can be obtained from $\mathbf{e}(k) = \boldsymbol{\psi}(k) - J_1\boldsymbol{\psi}(k)$, where $J_1 = I_M \otimes K$, and \otimes denotes the Kronecker product.

Based on this definition, we see that the error vector results in the following evolution:

$$\begin{aligned} \mathbf{e}(k) &= (H - J_1H)\boldsymbol{\psi}(k-1) \\ &= (H - J)[\boldsymbol{\psi}(k-1) - J_1\boldsymbol{\psi}(k-1)] \\ &= (H - J)\mathbf{e}(k-1). \end{aligned} \quad (14)$$

The above equation is valid because $(H - J)J_1 = \mathbf{0}_{MN \times MN}$, and $J_1H = J$. Then, we have

$$\begin{aligned} \|\mathbf{e}(k)\|^2 &= \|(H - J)\mathbf{e}(k-1)\|^2 \\ &\leq \alpha^2 \|\mathbf{e}(k-1)\|^2 \leq \dots \leq \alpha^{2k} \|\mathbf{e}(0)\|^2, \end{aligned} \quad (15)$$

which is equivalent to (13). This completes the proof. \square

Let us define the convergence region \mathcal{R} to satisfy $\rho(H - J) < 1$, that is,

$$\mathcal{R} = \{(\varepsilon, \gamma) \mid \rho(H - J) < 1\}. \quad (16)$$

Based on Lemma 2, we see that the convergence rate for the M -th order DAC algorithm in wireless sensor networks is determined by the spectral radius of $H - J$, which depends solely on the network topology. Furthermore, we note that there may exist possible choices of ε and γ to achieve the optimal convergence rate of the high-order DAC algorithm. To see this, we formulated the following spectral radius minimization problem to find the optimal ε and γ for the high-order DAC algorithm, that is,

$$\begin{aligned} \min_{\varepsilon, \gamma} \quad & \rho(H - J) \\ \text{s.t.} \quad & (\varepsilon, \gamma) \in \mathcal{R}. \end{aligned} \quad (17)$$

From (17), we see that the optimal convergence rate of our proposed high-order DAC algorithm depends solely on the eigenvalues of Laplacian matrix. Let us define the minimal spectral radius of $H - J$ as $\alpha_{\text{opt}} = \min\{\rho(H - J)\}$, and the optimal convergence rate as $\nu_{\text{opt}} = -\log(\alpha_{\text{opt}})$. When $M = 2$, the optimal convergence rate of second-order DAC algorithm can be obtained as [12]

$$\nu_{\text{opt,SO}} = \log \frac{\lambda_N(L) + 3\lambda_2(L)}{\lambda_N(L) - \lambda_2(L)}. \quad (18)$$

Recall that in the first-order DAC algorithm, we have [2]

$$\nu_{\text{opt,FO}} = \log \frac{\lambda_N(L) + \lambda_2(L)}{\lambda_N(L) - \lambda_2(L)}. \quad (19)$$

Clearly, we see that $\nu_{\text{opt,SO}} \geq \nu_{\text{opt,FO}}$. In the case when $M \geq 3$, we note that, in general, the closed-form solution for this optimization problem is hard to find due to the fact that high-order polynomial equations are involved in calculating

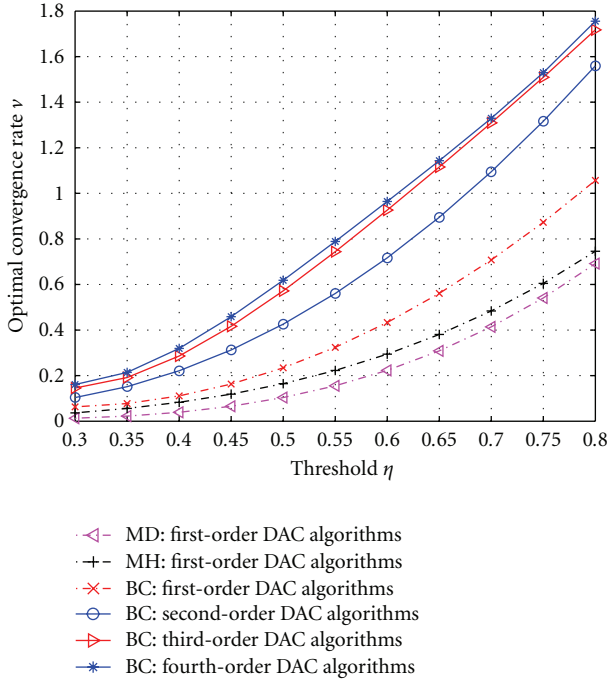


FIGURE 2: Convergence rate comparison of DAC algorithms with various weights in random networks versus distance threshold when $N = 16$.

the eigenvalues of $H - J$. For example, when $M = 3$ and $c_1 = 1, c_2 = 1$, we need to find the roots of the following cubic equation to obtain the eigenvalues of $H - J$:

$$f(\lambda) = \lambda^3 - (1 - \varepsilon\lambda_i(L))\lambda^2 - \gamma\varepsilon\lambda_i(L)\lambda + \gamma^2\varepsilon\lambda_i(L) = 0. \quad (20)$$

In practical applications, since the optimal ε and γ depend only on the network topology, a numerical solution can be obtained offline based on node deployment, and all design parameters can be flooded to the sensor nodes before they run the distributed algorithm. As we will show in the simulations, the optimal convergence rate can be greatly improved by this linear high-order DAC algorithm.

4. Simulation Results

In the following, we simulate networks in which the initial local state information of node i is equally spaced (trends similar to the ones noted below were observed when initial local state information between nodes were arbitrary (e.g., when they were uniformly distributed over $[-\beta, \beta]$). We use this fixed local state assumption here for comparison purposes) in $[-\beta, \beta]$, where $\beta = 500$. For the sake of simplicity, we only consider $M = 3$ and $M = 4$ for the higher-order DAC approach. In the simulations, we denote our proposed DAC algorithm as best constant (BC) high-order DAC algorithm and choose two types of ad hoc weights as comparison: maximum degree (MD) and metropolis hastig (MH) weights [13]. Furthermore, we assume $c_1 = 1, c_2 = 1, c_3 = 1/6$ and study the following two network topologies:

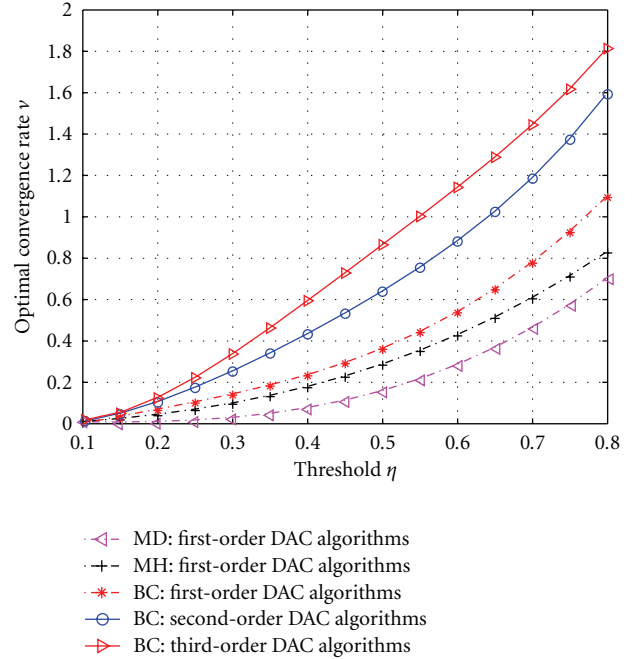


FIGURE 3: Convergence rate comparison of DAC algorithms with various weights in random networks versus distance threshold when $N = 256$.

Case 1. Fixed network with 6 nodes as shown in Figure 1(a).

Case 2. Random network with 16 nodes. The 16 nodes were randomly generated with uniform distribution over a unit square; two nodes were assumed connected if the distance between them was less than η , a predefined threshold. One realization of such a network is shown in Figure 1(b).

Figure 2 shows the optimal convergence rates for the DAC algorithms with various weights in random networks with 16 nodes as a function of η . The results are based on 1000 realizations of the random network where we excluded disconnected networks. From the plots, we note that the first-order BC DAC algorithm outperforms the first-order MH and MD DAC algorithms. Furthermore, we see that the optimal convergence rate increases as M increases. However, we also observe that the fourth-order DAC algorithm has negligible improvement compared to the third-order algorithm. Based on this, we restrict our examination of higher-order DAC algorithm to $M = 3$ in the subsequent results.

In addition to the results shown here, we ran this simulation setup for various realizations of random networks, assuming a large number of nodes. Figure 3 shows the convergence rate comparison for DAC algorithms with various weights when $N = 256$. As expected, we see that the results show a similar trend, that is, the optimal convergence rate of DAC algorithm increases as M increases.

In Figure 4, we compare the convergence rates of the third-order DAC algorithm with the first- and second-order DAC algorithms for both the random and fixed network

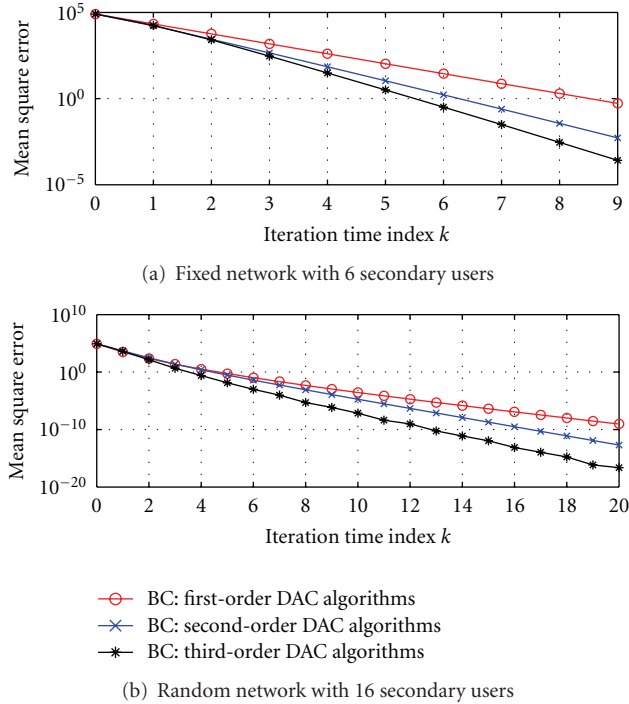


FIGURE 4: Convergence rate comparison of first-, second-, and third-order DAC algorithms: (a) fixed network with 6 nodes (Case 1) and (b) random network with 16 nodes (Case 2).

topologies. Specifically, we plot the mean square error (defined as $(1/N)\|\delta(k)\|^2$). In simulating random networks, we average out results over 1000 network realizations and assume $\eta = 0.9$, that is, network nodes are well connected with one another. As expected, we see that the third-order DAC algorithm converges faster than the first- and second-order DAC algorithms for both network scenarios.

5. Conclusions

In this paper, we present a linear high-order DAC algorithm to address the distributed computation problem in wireless sensor networks. Interestingly, the high-order DAC algorithm can be regarded as a spatial-temporal processing technique, where nodes in the network represent the spatial advantage, the high-order processing represents the temporal advantage, and the optimal convergence rate can be viewed as the diversity gain. In the future, we intend to investigate the effects of fading, link failure, and other practical conditions when utilizing the DAC algorithm in wireless sensor networks.

References

- [1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [2] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 5, pp. 4997–5002, December 2003.

- [3] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *Proceedings of the American Control Conference (ACC '05)*, vol. 4, pp. 2371–2378, June 2005.
- [4] E. Kokiopoulou and P. Frossard, "Accelerating distributed consensus using extrapolation," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 665–668, 2007.
- [5] U. A. Khan, S. Kar, and J. M. F. Moura, "Higher dimensional consensus: learning in large-scale networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2836–2849, 2010.
- [6] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed average consensus: beyond the realm of linearity," in *Proceedings of the 43rd IEEE Asilomar Conference on Signals, Systems and Computers*, November 2009.
- [7] B. N. Oreshkin, T. C. Aysal, and M. J. Coates, "Distributed average consensus with increased convergence rate," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 2285–2288, April 2008.
- [8] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [9] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2001.
- [10] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.
- [11] Y. Hatano and M. Mesbahi, "Agreement over random networks," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1867–1872, 2005.
- [12] G. Xiong and S. Kishore, "Discrete-time second-order distributed consensus time synchronization algorithm for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, Article ID 623537, 12 pages, 2009.
- [13] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 63–70, April 2005.