IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 31, 2020, accepted January 19, 2021, date of publication March 1, 2021, date of current version April 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063023

A Convergence-Accelerated Distributed Time Synchronization Algorithm for Energy-Harvesting Wireless Sensor Networks

QI YANG, (Member, IEEE), RONGPING ZHENG, JUNYU GUO, AND TAO CHEN
School of Informatics, Xiamen University, Xiamen 361005, China

Corresponding author: Qi Yang (yangqi@xmu.edu.cn)

This work was supported in part by the Science and Technology Key Project of Fujian Province, China, under Grant 2018HZ0002-1, in part by the China Scholarship Council (CSC), and in part by the Natural Science Foundation of China under Grant 61201196.

ABSTRACT Time synchronization is an essential problem for energy-harvesting wireless sensor networks (EH-WSNs), which is closely related to efficient resource schedules, energy harvesting, data fusion, location, etc. With the advantage of being more robust than master controlling synchronization, distributed time synchronization algorithms are usually used to EH-WSNs for cooperating sleeping nodes. This paper proposes a novel accelerated time co-synchronization algorithm based on the storage-and-prediction method to improve the convergence rate. In this algorithm, each node in the network first predicts the estimated current time state value according to previous time state values stored in the local node, and then adjusts the time state value according to the estimated time state value deviations between all its adjacent nodes. Theoretical analysis in a more general case shows that the proposed algorithm can improve the convergence rate of distributed time synchronization when selecting the appropriate parameter, and the closed-form solution of the optimal parameter is also given. Finally, the simulation of comparing the classical algorithm with the proposed algorithm based on different scenarios is completed.

INDEX TERMS Energy-harvesting, wireless sensor network, time synchronization, distributed consensus, convergence rate.

I. INTRODUCTION

Wireless sensor networks (WSNs) have broad application prospects in various fields such as military, environmental, industrial, medical and many other fields. The energy limitation of nodes in WSNs is one of the most important constraints as they are generally powered by batteries. With the development of energy harvesting technology, researchers have put forward the energy harvesting wireless sensor networks (EH-WSNs) to cope with the problem of the battery power limitation due to the difficulty of replacing batteries. EH-WSNs collect energy from the environment by energy harvesting technology to extend the lifetime of nodes.

Many researchers have studied the application of energy harvesting technology in WSNs. Paper [1] proposed an indoor energy harvesting system mixes light and heat to enhance the performance of indoor wireless sensor networks. A prototype hardware platform to collect solar and wind energy, which can realize long-term monitoring in wireless sensor networks, is designed Paper [2]. Paper [3] designed a multi-source energy collector to collect environmental energy including wind, solar, and thermal energy. Paper [4] utilized LoRa technology to propose energy harvesting technology based on photovoltaics and thermoelectric generators to collect energy for floating devices and use multi-hop technology to expand the communication range. Another way to obtain energy is harvesting radio frequency (RF) energy from wireless signals which is lower in cost and smaller in equipment size [5]. Paper [6] proposed a RF-based energy harvesting technology to power the sensor nodes, and it designed a novel compact folded dipole rectenna to improve the energy harvesting capability. Paper [7] analyzed the wireless radio frequency energy harvesting network based on the Boolean-Poisson model when the sensor nodes collect energy simultaneously.

However, in EH-WSNs, energy harvesting provides limited and unstable energy. Sensor nodes need to accumulate energy and fall into intermittent sleeping [8]–[13]. When

The associate editor coordinating the review of this manuscript and approving it for publication was Deyu Zhang.

sensor nodes are awake for communication, the transmitting nodes and the receiving nodes should be active at the same time. Therefore, the transmitting nodes and receiving nodes need to determine the execution time of communication according to the network common time, hence time synchronization is an important premise of EH-WSNs [14].

Distributed average consensus algorithms are usually used for time synchronization in EH-WSN because it has no reference nodes and does not depend on network topology. Distributed time synchronization algorithms, such as distributed consensus and gossip algorithms, calculate the weighted sum of current time values of each node and all its neighboring nodes to serve as the next local time values [15]–[22]. When the network is fully connected, the weight matrix is a doubly stochastic, symmetrical and non-negative matrix, it has been proved that the algorithm will converge and achieve consensus [23]. However, the convergence rate and synchronization accuracy of the algorithm is affected by the network topology. Because the distributed average consensus algorithm converges gradually via continuous iterative operation, its convergence rate is very slow and is determined by the initial time state of nodes, the number of network connections, and the topology structure. In practice, convergence rate is a key factor that limits the application of distributed average consensus algorithms. Therefore, how to improve the convergence rate of distributed average consensus algorithm has been the focus of researchers.

Much work has been conducted in the literature to improve the convergence rate of distributed average consensus algorithms [24]–[29]. Early research on distributed average consensus algorithms determined the weight matrix mainly based on the node degree. However, such algorithms have a slow convergence rate in random topology [15], [30]. Later improvements on the algorithm can be roughly divided into four categories of methods. The first method is to optimize the weight matrix. Because the convergence rate of the distributed average consensus algorithm is related to the sub-large eigenvalue of the weight matrix, faster convergence rate and a better convergence effect can be achieved through the optimization of the weight matrix [16], [31]–[36]. Therefore, minimizing the spectral radius of the weight matrix can serve to improve the convergence rate. However, the spectral radius is constrained by the network topology and may be unpractical.

Another way to improve the convergence rate of distributed average algorithms is by using the information of second-order neighbors [37], [38]. Although this method can achieve a faster convergence rate, the information of second-order neighbors cannot be directly acquired by the local node, it should be forwarded through a relay node in two hops, which increases the communication overhead and node power consumption. The third method is a nonlinear iterative method, in which the stability of the algorithm is a key issue [39]. The last category of methods uses the historical information stored on nodes to improve the convergence rate. In such methods, the former and current state values of nodes are used for the calculation of state updates [26], [40]–[42]. Moreover, some researchers showed that better convergence performance can be achieved by using a predictor to estimate the future value based on the past state value of the node, and then updating the state value by the weighted sum of the estimated value [43]–[49]. In distributed time synchronization algorithms, it has been shown that jointly exploiting the stored information and predicted information can greatly improve the convergence rate. However, the selection of predictor parameters will greatly affect the accuracy and convergence rate of time synchronization. Moreover, the calculation of optimal predictor parameters is very complex, making parameter optimization a major challenge in the last type of method.

In the paper [50], the convergence of the distributed average linear iterative algorithm has been theoretically analyzed based on second-order storage information under the condition that the network topology is fully-connected and the weight matrix is doubly stochastic, symmetric and non-negative. However, the effect of algorithm parameters on the convergence rate of the algorithm is not fully discussed in this paper. When comparing the convergence rate with the traditional distributed average algorithm, paper [50] proved that the optimal parameters exist to make the new storage-based scheme converge faster than traditional schemes, but the convergence rate of the algorithm with non-optimal parameters was not discussed. Based on [50], paper [51] discussed a general case. It theoretically analyzed the convergence performance of a new algorithm and the influence of different parameters on the convergence rate of the algorithm when the weight matrix is not necessarily non-negative. It also gave the solution of the optimal parameter to archive the convergence of the algorithm. However, paper [51] considered the case that the weight matrix is symmetric, and the asymmetric matrix is not in the scope of discussion. Besides, the improvement of convergence rate has not been verified by simulations.

In this paper, we propose a novel accelerated distributed time synchronization algorithm, which uses storage and predictors to improve the convergence rate. The algorithm first predicts the current time base reference value based on the node's previous time state value, and then adjusts the current time reference value based on the deviation from the neighbor's time reference value. The selection of predictor parameters in the algorithm is entirely based on local information. In this paper, the convergence conditions of the second-order model of the algorithm are proved theoretically under the premise that the network is fully-connected and the weight matrix is a real stochastic matrix. The influence of algorithm parameters on the convergence rate is analyzed, and the convergence rate of the algorithm is compared with that of the classical algorithm. The optimal convergence solution of the predictor parameter is derived. Finally, we show simulation results for different setups and scenarios.

The remainder of the paper is organized as follows. Section II reviews related works. Section III introduces energy-harvesting wireless sensor networks and the classical distributed time synchronization algorithm for EH-WSNs.

Section IV describes the theoretical model of the proposed novel time synchronization algorithm for EH-WSNs, analyses the convergence and compares the convergence rate of two algorithms. Section V simulates our proposed time synchronization algorithm for EH-WSNs. Finally, conclusions are presented in Section VI.

II. RELATED WORK

Distributed synchronization algorithm can be applied to the time synchronization problem of distributed network. The key problem in distributed synchronization is how to improve the convergence rate. In the past few years, methods for accelerating distributed synchronization algorithms have been widely discussed. Paper [16] proposed a neighbor algorithm for the clustering network, in which each node selects its own weight based on the information of its neighbor nodes. The convergence of the algorithm is mainly determined by the weight of links connecting different clusters. The neighbor algorithm can be used to identify such links and give them higher weights to improve convergence rate. Paper [31] and [32] transformed the problem of improving the convergence rate into the problem of minimizing the sub-large eigenvalues of the weight matrix, which is a semi-definite programming problem. By using the sub-gradient method, the optimization problem can be solved, and the convergence rate is hereby improved. Paper [33] improved the convergence rate by properly selecting an asymmetric weight matrix, and it achieved a faster convergence rate than the algorithm using symmetric optimal weight and the Metropolis-Hastings weight. A distributed weight graph updating algorithm is proposed in [34], in which the weight at the next moment is selected adaptively according to the current value of the weight. The adaptive weight design can improve the convergence rate of the algorithm and the robustness of the network. Paper [35] and [36] not only optimized the weight matrix, but also used the storage information of node state for iterative updating. The node state information is updated by the last moment iteration, which further improves the convergence rate. In this kind of algorithms, the spectral radius of the weight matrix determines the convergence rate of the algorithm. Therefore, these algorithms optimize the spectral radius to be as small as possible to improve the convergence rate. However, the spectral radius is constrained by the network topology. Meanwhile, when the optimization problem is transformed into a semi-definite programming problem, which demands much work in initialization.

Considering the importance of second-order neighbors, many researchers proposed to improve the network convergence performance based on second-order neighbors' information. Paper [37] and [38] used the information of both neighbors and second-order neighbors in each iteration. Compared with the classical algorithm that only uses the information of first-order neighbors, this algorithm can achieve a faster convergence rate. However, the information of second-order neighbors is not obtained via a direct connection, it is forwarded to the local node through two hops, which increases network traffic and control overhead. In addition, nonlinear iterative methods have been proposed to improve the convergence rate. Paper [39] proposed a nonlinear iterative algorithm, in which the selection of parameters is closely related to the sub-large eigenvalues and the maximum eigenvalues of the Laplacian matrix. When the weight matrix satisfies certain conditions, the nonlinear iterative algorithm can converge to the initial mean value, and the convergence rate of the algorithm is better than the classical linear iterative algorithm. However, because the algorithm adopts nonlinear iteration, the stability of the system becomes an issue of concern.

In recent years, the main research focus of distributed average algorithms is to use the storage information of nodes to improve the convergence rate. Paper [40] proposed an acceleration framework of the storage-based gossip algorithm, in which each node uses a multi-order shift register to store the past values. Taking the second-order as an example, the paper proved the convergence of the algorithm, and compared the performance of the algorithm with different order shift registers by simulation. However, the paper did not prove that the algorithm can improve the convergence rate theoretically. Paper [41] made further theoretical analysis of [40]. It analyzed the convergence of the algorithm in the case of two registers under the assumption of symmetry weighed matrix, then theoretically proved that the accelerated gossip algorithm has a faster convergence rate than the standard gossip algorithm. The necessary conditions for the convergence of symmetric gossip algorithm in the case of multiple registers are also given, but the convergence rate is not discussed. Paper [42] presented a system consisting of filters and time-delays, and then converts the problem of maximization of convergence rate to the minimization of the maximum spectral radius of the system transfer function. The paper only demonstrates that the algorithm can improve the convergence rate by simulation, it does not prove it by the theoretical method. Paper [26] proposed a distributed average consensus algorithm for fixed undirected graphs. The algorithm uses the classical algorithm and adds a momentum term to the differential iterative equation to speed up the linear iteration. The convergence time is linearly related to the number of nodes. Compared with the classical algorithm, the convergence rate of the algorithm in the two-dimensional grid graph and geometric random graph can be greatly improved. However, the calculation process of algorithm parameters is complex, and the selection method of algorithm parameters is not given in the paper.

Another similar method is using a predictor. Paper [43] proposed a faster distributed average algorithm, which combines linear predictor and standard consensus algorithm. The convergence of second-order prediction is analyzed, and the optimal parameters are derived. However, only the second largest and smallest eigenvalues of the weight matrix are considered in the optimization of parameters, and the suboptimal solution is given. Paper [47] proposed a distributed average algorithm of polynomial filter, which filters the current value

based on the past state value, so it can be regarded as a linear prediction process. The algorithm can achieve a faster convergence rate. However, the input of the filter in the paper is not the stored actual state value, but the 1 classical iteration value. Besides, the optimal filter coefficient is determined by the semi-definite programming method which is not very suitable in practice. Paper [48] obtained mixed parameter feasible domain and optimal mixing parameter for the convergence of the distributed average consensus algorithm using a linear predictor in undirected networks, but it does not theoretically analyze the convergence rate of the algorithm. Paper [49] theoretically deduced the advantages of predictive synchronization algorithm. The closed expression of the optimized mixed parameter is obtained, and the optimal value of the mixed parameter and the upper bound of the spectral radius of the weight matrix of the fast convergence algorithm is derived. However, the calculation process of algorithm parameters in it is complex, which is not very suitable in practice.

Based on the above related work, this paper proposes a distributed time synchronization algorithm for EH-WSN, which uses storage and predictor. The improvement of convergence rate is proved by theory without the assumption of the doubly stochastic and symmetric weight matrix. Simulations in different scenes are completed.

III. ENERGY-HARVESTING WIRELESS SENSOR NETWORKS

Energy-harvesting wireless sensor networks have been proposed as a main solution to tackle the challenge of energy limitation in wireless sensor networks. A general energy-harvesting wireless sensor network is shown in Fig.1. Widely distributed sensor nodes are used to monitor a certain area, the data collected by a sensor node is sent to the sink node in a multi-hop manner. Each sensor node could harvest energy to extend the lifetime when the power fails. Due to severe energy constraints, the intermittent sleeping method is widely adopted in EH-WSHs. To ensure reliable communication in such networks, both the transmitting and receiving nodes should be active according to the consented network time. Therefore, as a prerequisite for communication, cooperative sensing and energy management, time synchronization is of great importance in EH-WSNs.

Distributed time synchronization algorithms are usually used in EH-WSNs for the coordination of sleeping nodes. Compared with centralized algorithms that use the master-slave type of control mechanisms, distributed synchronization algorithms can achieve higher robustness in large-scale networks. By allowing nodes in the network to exchange information iteratively to update their local time, all nodes in the network will finally achieve time synchronization. Since there is no central control node in the distributed time synchronization algorithm, each node communicates only with its neighbors to achieve a consensus time.

Consider an energy-harvesting wireless sensor network composed of N energy harvesting nodes, which can be represented by a connected undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots, N\}$ is the set of nodes in the network and E is the set of links between nodes. For the classical distributed time synchronization algorithm, the idea is to adjust the state value of the local node based on the state value of its neighbor nodes, and finally achieve synchronization through mutual adjustment. The iterative equation of classical algorithm is shown as below:

$$x_i(n+1) = \sum_{j=1}^{N} w_{ij} x_j(n), i \in \{1, 2, \ldots N\} \tag{1}$$

where $N > 1$ is an integer, and $x_i(n)$ is the time state value of node i at the nth adjustment. w_{ij} is a constant value satisfying Equation (2)-(4):

$$w_{ij} = \begin{cases} w_{ij}, j \in N_i, i \in \{1, 2 \ldots, N\} \\ 1 - \sum\limits_{k \in N_i} w_{ik}, j = i, i \in \{1, 2 \ldots, N\} \end{cases} \tag{2}$$

$$w_{ij} = 0, j \notin N_i, i \in \{1, 2, \ldots, N\} \tag{3}$$

$$\sum_{j=1}^{N} w_{ij} = 1, i \in \{1, 2 \ldots, N\} \tag{4}$$

where N_i is the set of neighbors of node i. Fig. 2 shows an example of the EH-WSN topology.

Equation (1) can be expressed as matrix form:

$$X(n) = WX(n-1) = W^n X(0) \tag{5}$$

where $X(n) = \left[x_1(n)\ x_2(n)\ \cdots\ x_N(n)\right]^T$, which is the time state value vector of network nodes after n iterations, and $W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}$, which is an $N \times N$ stochastic matrix. Each element in W is a real number.

Suppose that $\mu_i, i \in \{1, 2, \ldots, N\}$ denote the eigenvalue of the weight matrix W. Obviously, the unique maximum eigenvalue of matrix W equals to 1, i.e. $|\mu_1| = 1$. When all the remaining eigenvalues fall in the unit circle of the complex plane, $X(n)$ can converge to a finite value, i.e. $|\mu_N| \leq |\mu_{N-1}| \leq \ldots < |\mu_1| = 1$. The convergence rate of the algorithm depends on the second largest eigenvalue of W. And the smaller the secondary eigenvalue, the faster the convergence rate of the algorithm.

IV. THE CONVERGENCE-ACCELERATED DISTRIBUTED TIME SYNCHRONIZATION ALGORITHM

In order to improve the convergence rate of the classical distributed time synchronization algorithm, an accelerated time synchronization algorithm based on storage and prediction scheme is proposed in this paper. The new algorithm not only considers the current state value of the node, but also considers the previous state value.

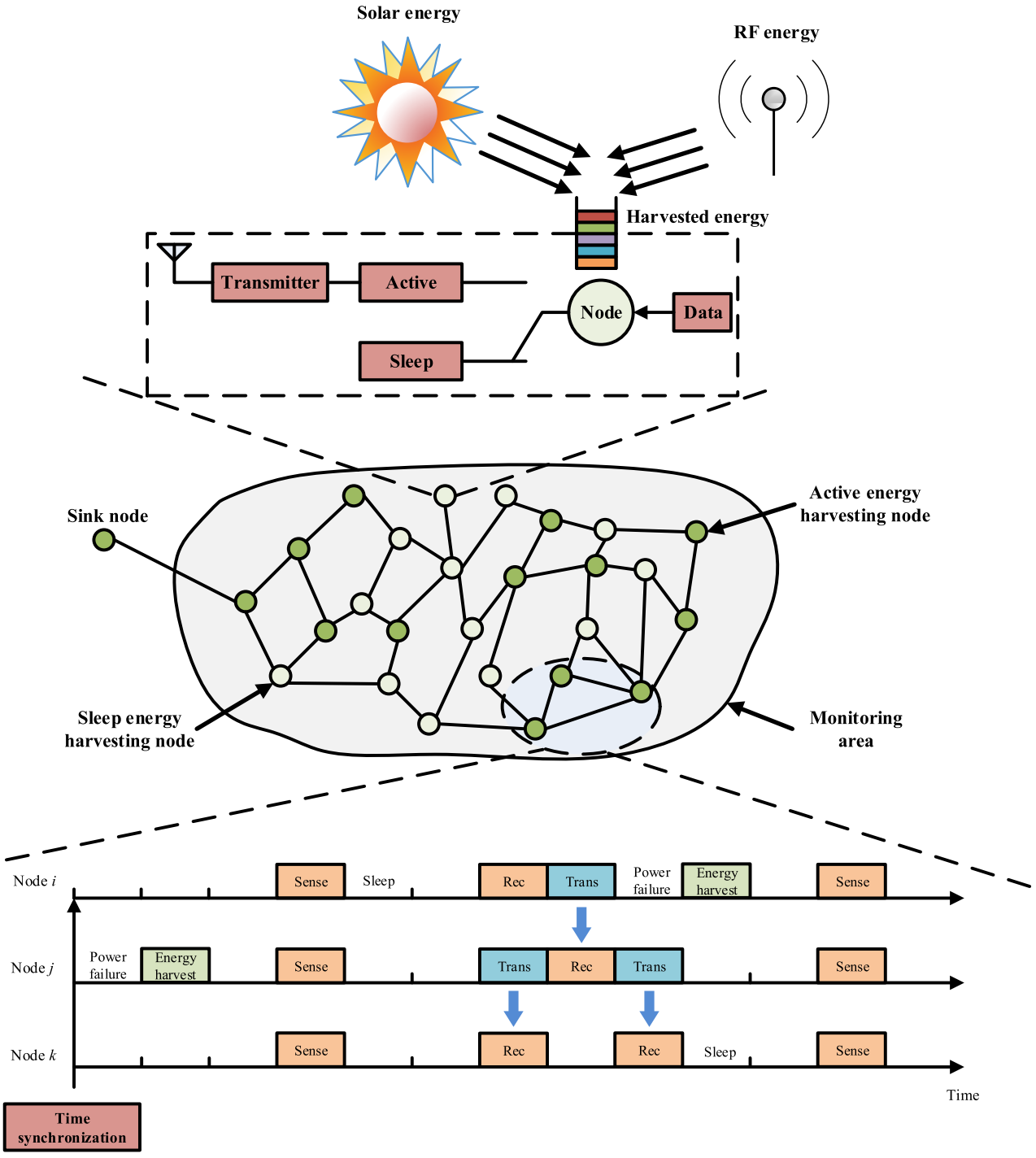


FIGURE 1. The energy-harvesting wireless sensor network.

A. THE GENERAL THEORETICAL MODEL OF THE NOVEL ALGORITHM

It is assumed that each node predicts the current state value based on its previous $M + 1$ stored state information values:

$$\hat{x}_i(n) = x_i(n-1) + \sum_{m=1}^{M} a_m \left(x_i(n-m) - x_i(n-m-1)\right) \quad (6)$$

where $\hat{x}_i(n)$ is the predicted value of node i at the nth adjustment, and a_m is a real constant, indicating the prediction parameter.

Then, the nth state value can be obtained by using the weighted coupling of the time deviation with neighbor nodes to modify the predicted value:

$$x_i(n) = \hat{x}_i(n) - \sum_{j \in N_i} w_{ij} \left(\hat{x}_i(n) - \hat{x}_j(n)\right) \quad (7)$$

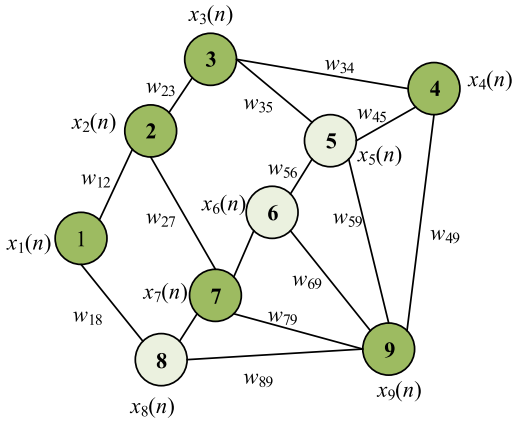


FIGURE 2. An example of the EH-WSN topology.

According to Equation (2)-(4), Equation (7) can be presented as:

$$x_i(n) = \sum_{j\in N_i} w_{ij}\left\{x_j(n-1) + \sum_{m=1}^{M} a_m\left(x_j(n-m) - x_j(n-m-1)\right)\right\} \quad (8)$$

Thus, the matrix form of (7) is as follows:

$$X(n) = (1+a_1)WX(n-1) + \ldots + (a_M - a_{M-1})WX(n-M) - a_M WX(n-M-1) \quad (9)$$

To be further define

$$U(n) = \left[X^T(n), X^T(n-1), \ldots, X^T(n-M)\right]^T \quad (10)$$

$$B = \begin{bmatrix} (1+a_1)W & \cdots & (a_M - a_{M-1})W & -a_M W \\ I & & & \\ & I & & \\ & & I & \end{bmatrix} \quad (11)$$

Equation (9) can be expressed as follows:

$$U(n) = BU(n-1) \quad (12)$$

So far, the general theoretical model of the new algorithm has been established. It can be seen that the higher the prediction order, the more complex the model is. However, no matter what the prediction order is, after appropriate equivalent transformation, the model can be transformed into the matrix form of a similar general model. At the same time, we can see that the classical time synchronization model is a special case of this model.

B. THE SECOND-ORDER MODEL OF THE NOVEL ALGORITHM

Then, we will focus on the case of second-order model, that is, $M+1=2$. And the Equation (6) can be given as:

$$\hat{x}_i(n) = x_i(n-1) + a\left[x_i(n-1) - x_i(n-2)\right] \quad (13)$$

And we can obtain:

$$x_i(n) = \sum_{j\in N_i} w_{ij}\left\{(1+a)x_j(n-1) - ax_j(n-2)\right\} \quad (14)$$

Thus, Equation (14) could be rewritten as the matrix form:

$$X(n) = (1+a)WX(n-1) - aWX(n-2) \quad (15)$$

Finally, the second-order model of the new algorithm is:

$$U(n) = BU(n-1) \quad (16)$$

The vector $U(n)$ and matrix B was defined as follow:

$$U(n) = \begin{bmatrix} X^T(n) \\ X^T(n-1) \end{bmatrix} \quad (17)$$

$$B = \begin{bmatrix} (1+a)W & -aW \\ I & 0 \end{bmatrix} \quad (18)$$

Weather the convergence rate of second order model algorithm is faster than that of classical distributed time synchronization algorithm depends of the second large eigenvalues of matrix W and matrix B.

C. THE CONVERGENCE ANALYSIS OF THE SECOND-ORDER MODEL

In this part, we will analyze the convergence of the second-order model of the new algorithm. From the previous analysis, we can see that the convergence of the second-order model (16) depends on the eigenvalue characteristics of its weight matrix B. If the secondary spectral radius of matrix B is not more than 1, then the algorithm will eventually converge, and the convergence rate depends on the second largest eigenvalue of B. It is noted that the eigenvalues of the weight matrix B are closely related to the eigenvalues of the weight matrix W. The comparison of the convergence rate of the two synchronization algorithms depends on the analysis and comparison of the eigenvalues of the weight matrix B and W. So, the comparison of the convergence rate can be attributed to the comparison of the second largest eigenvalue of the weight matrix B and W.

Suppose that λ is a eigenvalue of the weight matrix B, and the corresponding feature vector is $\begin{bmatrix} z_1 & z_2 \end{bmatrix}^T$, and μ is a eigenvalue of the weight matrix W, and the corresponding feature vector is z_2. It is easy to prove that λ and μ have the following relationship:

$$\mu = \frac{\lambda^2}{(1+a)\lambda - a} \quad (19)$$

From Equation (19) we can obtain:

$$\lambda^2 - \mu(1+a)\lambda + \mu a = 0 \quad (20)$$

Next, we will discuss the convergence of the proposed storage algorithm when μ_i is real or complex, respectively. Firstly, we consider the case that μ_i is a real number.

1) THE ANALYSIS OF CONVERGENCE CONDITIONS

It can be seen from Equation (20) that there may be two eigenvalues $\lambda_{i1}, \lambda_{i2}$ of the weight matrix B corresponding to μ_i. Provided that the classical model is convergent, then the largest eigenvalue of the matrix W must equal one, and the others are all smaller than one. In order to ensure that the second-order model is convergent, the eigenvalue of its weight matrix B must satisfying $|\lambda_{2N}| \leq |\lambda_{2N-1}| \leq \cdots < |\lambda_1| = 1$.

Since the classical model must have eigenvalues $\mu_1 = 1$, by substituting it into the Equation (20) we can obtain:

$$\lambda_1^2 - (1+a)\lambda_1 + a = 0 \tag{21}$$

And the solution is obtained:

$$\lambda_{11} = 1, \lambda_{12} = a \tag{22}$$

According to $|\lambda_{2N}| \leq |\lambda_{2N-1}| \leq \cdots < |\lambda_1| = 1$, we can obtain $|a| < 1$, i.e. $-1 < a < 1$. It should be note that a is a real number. Then, we will prove the following result. For the stochastic matrix $W \in R^{N\times N}$, when $W^T = W$ and $-1 < a < 1$, the second-order model of the new algorithm is convergent.

Let $\rho_2(B)$ be the second largest spectral radius of matrix B. If $\rho_2(B) < 1$, the second-order model will eventually be convergent. And the smaller $\rho_2(B)$, the faster the convergence rate of the algorithm. Apparently, the second-order model has a special eigenvalue that equals one, the further work is to prove that the other $2N-1$ eigenvalues are smaller than one.

From Equation (20) we can obtain:

$$\begin{cases} \lambda_{i1} = \dfrac{\mu_i(1+a) + \sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}}{2} \\ \lambda_{i2} = \dfrac{\mu_i(1+a) - \sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}}{2} \end{cases} \tag{23}$$

Here we want to compare the relationship between the modulus of eigenvalues of the weight matrix B and. Since the weight matrix W is real symmetric matrices, its eigenvalues μ_i are all real number. Therefore, we define $\rho_i = \max\{|\lambda_{i1}|, |\lambda_{i2}|\}$. If $\Delta = \mu_i^2(1+a)^2 - 4\mu_i a \geq 0$, then λ_{i1} and λ_{i2} are real. If $\mu_i > 0$, then $|\lambda_{i1}| > |\lambda_{i2}|$, and $\rho_i = |\lambda_{i1}|$; if $\mu_i \leq 0$, then $|\lambda_{i1}| \leq |\lambda_{i2}|$, and $\rho_i = |\lambda_{i2}|$. On the other hand, if $\Delta = \mu_i^2(1+a)^2 - 4\mu_i a < 0$, then λ_{i1} and λ_{i2} are complex, and $\rho_i = |\lambda_{i1}| = |\lambda_{i2}|$.

Firstly, we discuss the case of μ_i being non-negative, i.e. $0 \leq \mu_i \leq 1$. Note that if μ_i=0, then ρ_i=0. If μ_i=1, then $\rho_i = \rho_1$=1. Next, we analyze the relationship between the spectral radius of the weight matrix W and B, i.e. the relationship between ρ_i and $|\mu_i|$.

1) a=0

Obviously, we can obtain that $\rho_i = |\mu_i|$. Therefore, ρ_i is an increasing function of $|\mu_i|$.

2) $-1 < a < 0$

Obviously, we can see that $\Delta = \mu_i^2(1+a)^2 - 4\mu_i a \geq 0$, then

$$\rho_i = \frac{\mu_i(1+a) + \sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}}{2} \tag{24}$$

And the partial derivative of $|\mu_i|$ is

$$\frac{\partial \rho_i}{\partial |\mu_i|} = \frac{1}{2}\left[(1+a) + \frac{\mu_i(1+a)^2 - 2a}{\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}}\right] \tag{25}$$

Therefore, ρ_i is an increasing function of $|\mu_i|$.

3) $0 < a < 1$

i) If $\Delta = \mu_i^2(1+a)^2 - 4\mu_i a \geq 0$, i.e. $\frac{4a}{(1+a)^2} \leq \mu_i < 1$, we can obtain

$$\rho_i = \frac{\mu_i(1+a) + \sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}}{2} \tag{26}$$

And the partial derivative of $|\mu_i|$ is

$$\frac{\partial \rho_i}{\partial |\mu_i|} = \frac{1}{2}\left[\frac{(1+a)\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a} + \mu_i(1+a)^2 - 2a}{\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}}\right] \tag{27}$$

It is formidable to see the monotony here, thus let

$$f(\mu_i) = (1+a)\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a} + \mu_i(1+a)^2 - 2a \tag{28}$$

where $\mu_i \in [\frac{4a}{(1+a)^2}, 1)$.

Then,

$$\begin{aligned} \frac{\partial f(\mu_i)}{\partial \mu_i} &= \frac{(1+a)^2\left[\mu_i(1+a) + \sqrt{\mu_i(1+a)^2 - 4\mu_i a}\right] - 2a}{\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}} \\ &> \frac{(1+a)^2\mu_i - 2a}{\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}} \\ &\geq \frac{(1+a)^2\frac{4a}{(1+a)^2} - 2a}{\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}} = \frac{2a}{\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}} \end{aligned} \tag{29}$$

Therefore, $f(\mu_i)$ is the increasing function of μ_i when $\mu_i \in [\frac{4a}{(1+a)^2}, 1)$. And

$$f(\mu_i) \geq f\left(\frac{4a}{(1+a)^2}\right) = 2a > 0 \tag{30}$$

So $\frac{\partial \rho_i}{\partial |\mu_i|} > 0$, and ρ_i is an increasing function of $|\mu_i|$.

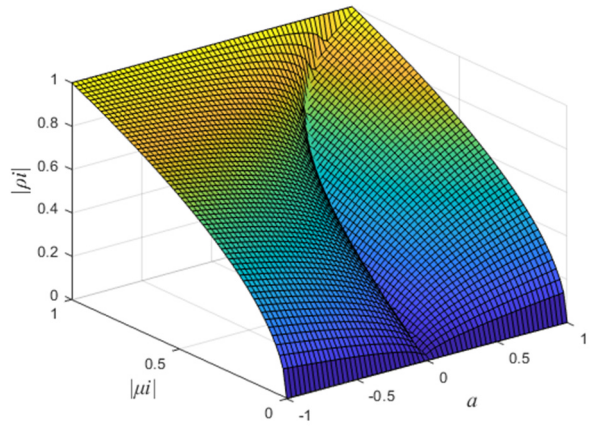


FIGURE 3. The relationship between $|\rho_i|$, $|\mu_i|$ and a.

ii) If $\Delta = \mu_i^2(1+a)^2 - 4\mu_i a < 0$, i.e. $0 < \mu_i < \frac{4a}{(1+a)^2}$, we can obtain

$$\begin{aligned}\rho_i &= \frac{1}{2}\sqrt{[\mu_i(1+a)]^2 + [\sqrt{\mu_i^2(1+a)^2 - 4\mu_i a}]^2} \\ &= \sqrt{\mu_i a}\end{aligned} \tag{31}$$

Obviously, ρ_i is an increasing function of $|\mu_i|$.

The analysis of the situation when $-1 < \mu_i < 0$ is similar to the above, which is omitted here.

In summary, when $a \in (-1, 1)$, ρ_i is the increasing function of $|\mu_i|$. Thus, for $\forall i \in (2, .., 2N-1)$,we can obtain $\rho_i < \rho_1 = 1$. That is to say, the largest eigenvalue of the matrix B equals one, and the others are smaller than one. So far, the convergence of the new algorithm has been proved.

Fig. 3 shows the relationship between the eigenvalues of two matrices and the parameter a.

2) THE CONVERGENCE RATE

Next, we will compare the convergence rate of the second-order model with that of the classical model. Because the convergence rate of the algorithm depends on the sub-spectral radius of the weight matrix. Therefore, we compare the sub-spectral radius to compare the convergence rate of two models. Noticing that ρ_i is the monotonic increasing function of $|\mu_i|$, we can know that when $|\mu_i| = |\mu_2|$, $\rho_i = \rho_2$.

Similarly, we need to discuss the convergence rate in different cases. We mainly discuss the case of μ_2 being non-negative, i.e. $0 \le \mu_2 < 1$, and the analysis of $-1 < \mu_2 < 0$ is similar to that, which is omitted here.

1) $-1 < a < 0$

From Equation (24) we can obtain:

$$\rho_2 = \frac{\mu_2(1+a) + \sqrt{\mu_2^2(1+a)^2 - 4\mu_2 a}}{2} \tag{32}$$

And then,

$$\begin{aligned}\frac{\rho_2}{|\mu_2|} &= \frac{1}{2}\left[(1+a) + \sqrt{(1+a)^2 - \frac{4a}{\mu_2}}\right] \\ &> \frac{1}{2}\left[(1+a) + \sqrt{(1+a)^2 - 4a}\right] \\ &= \frac{1}{2}[(1+a) + (1-a)] \\ &= 1\end{aligned} \tag{33}$$

Thus, $\rho_2 > |\mu_2|$. It indicates that the convergence rate of the classical model is faster than that of the second-order model.

2) $0 < a < 1$

i) If $0 < \mu_2 < \frac{4a}{(1+a)^2}$, from Equation (31) we can obtain $\rho_2 = \sqrt{\mu_2 a}$. And then,

$$\frac{\rho_2}{|\mu_2|} = \sqrt{\frac{a}{\mu_2}} \tag{34}$$

If $a > \mu_2$, then $\rho_2 > |\mu_2|$, it indicates that the convergence rate of the second-order model is slower than that of the classical model.

If $a = \mu_2$, then $\rho_2 = |\mu_2|$, it indicates that the convergence rate of the second-order model is as fast as that of the classical model.

If $a < \mu_2$, then $\rho_2 < |\mu_2|$, it indicates that the convergence rate of the second-order model is faster than that of the classical model.

ii) If $\frac{4a}{(1+a)^2} \le \mu_2 < 1$, from Equation (26) we can obtain

$$\rho_2 = \frac{\mu_2(1+a) + \sqrt{\mu_2^2(1+a)^2 - 4\mu_2 a}}{2} \tag{35}$$

And then,

$$\begin{aligned}\frac{\rho_2}{|\mu_2|} &= \frac{1}{2}\left[(1+a) + \sqrt{(1+a)^2 - \frac{4a}{\mu_2}}\right] \\ &< \frac{1}{2}\left[(1+a) + \sqrt{(1+a)^2 - 4a}\right] \\ &= \frac{1}{2}[(1+a) + (1-a)] \\ &= 1\end{aligned} \tag{36}$$

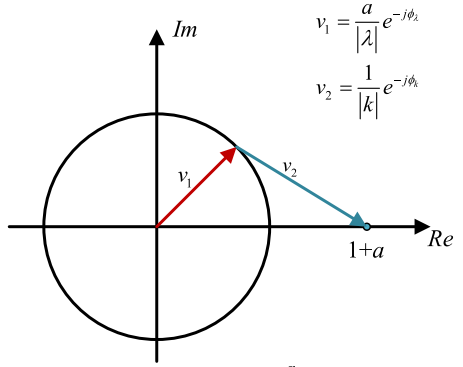
Thus, $\rho_2 < |\mu_2|$, it indicates that the convergence rate of the second-order model is faster than that of the classical model.

3) a=0

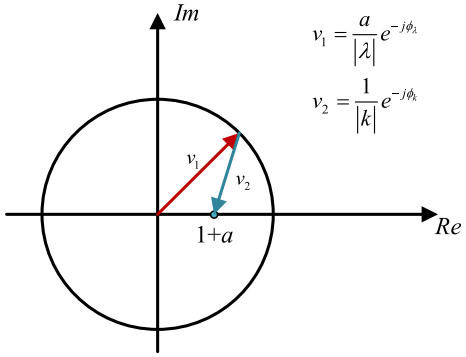
Obviously, we can obtain $\rho_2 = |\mu_2|$, it indicates that the convergence rate of the second-order model is as fast as that of the classical model.

Because the convergence of the second-order model depends on the eigenvalue characteristics of its weight matrix B and the smaller the secondary eigenvalue, the faster the convergence rate of the algorithm. Therefore, when ρ_2 reaches the minimum, the second-order model has the fastest convergence rate. From Equation (23) we can see that when $\Delta = \mu_2^2(1+a)^2 - 4\mu_2 a = 0$, ρ_2 gets the minimum, i.e. $\rho_2^* = 1 - \sqrt{1-\mu_2}$, and

$$a^* = \frac{2 - \mu_2 - 2\sqrt{1-\mu_2}}{\mu_2} \tag{37}$$

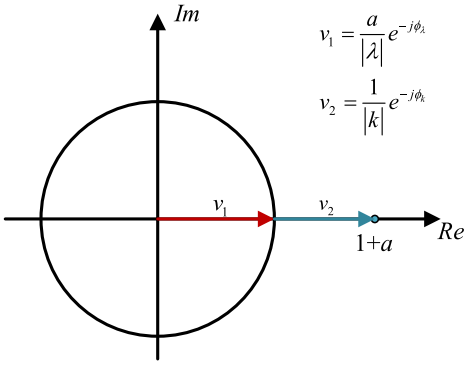


(a) $1+a > \frac{a}{|\lambda|}$

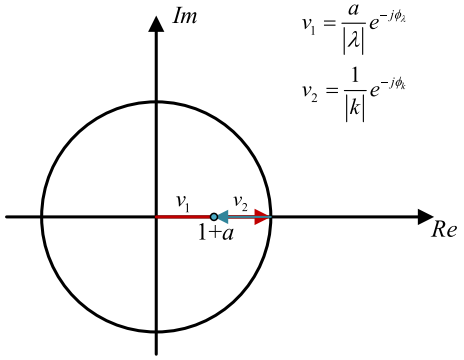


(b) $1+a < \frac{a}{|\lambda|}$

FIGURE 4. The relationship between v_1 and v_2 when (a) $1+a > \frac{a}{|\lambda|}$ and (b) $1+a < \frac{a}{|\lambda|}$.



(a) $1+a > \frac{a}{|\lambda|}$



(b) $1+a < \frac{a}{|\lambda|}$

FIGURE 5. The values of v_1 and v_2 when the convergence rate of the proposed algorithm is fastest for (a) if $a > 0$ or $a < 0$ when $1+a > \frac{a}{|\lambda|}$ and (b) if $a > 0$ or $a < 0$ when $1+a < \frac{a}{|\lambda|}$.

Then, we further consider the case when μ_i is complex number. We express a complex number in the form of modulus and argument: $\mu = |\mu| e^{j\phi_\mu}$, $\lambda = |\lambda| e^{j\phi_\lambda}$, Equation (19) could be expressed as following:

$$|\mu| e^{j\phi_\mu} = \frac{|\lambda|^2 e^{j2\phi_\lambda}}{(1+a)\cdot|\lambda|\cdot e^{j\phi_\lambda} - a} \tag{38}$$

Suppose that $k = \mu/\lambda$, that is: $|\mu| = |\lambda|\cdot|k|$ and $e^{j\phi_\mu} = e^{j(\phi_\lambda+\phi_k)}$. Then Equation (40) can be rewritten as

$$\frac{1}{|k|}e^{-j\phi_k} + \frac{a}{|\lambda|}e^{-j\phi_\lambda} = 1+a \tag{39}$$

Let $v_1 = \frac{a}{|\lambda|}e^{-j\phi_\lambda}$ and $v_2 = \frac{1}{|k|}e^{-j\phi_k}$, then $v_1 + v_2 = 1+a$. Fig. 4 shows the vector relationship between v_1 and v_2. $|k|$ is an indicator of the convergence rate of the second-order model algorithm. We hope $|k|$ to be larger to get faster convergence rate, which is equivalent to get a smaller $|v_2|$. According to Equation (39) and as shown in Fig. 5, to get the fastest convergence rate, we should obtain:

4) *IF* $a > 0$ *and* $1+a > \frac{a}{|\lambda|}$

$$\phi_\lambda = \phi_k = 0 \tag{40}$$

And then

$$|\lambda|^2 - |\mu|(1+a)|\lambda| + |\mu| a = 0 \tag{41}$$

5) *IF* $a > 0$ *AND* $1+a < \frac{a}{|\lambda|}$

$$\begin{cases} \phi_\lambda = 0 \\ \phi_k = \pi \end{cases} \tag{42}$$

And then

$$-|\lambda|^2 - |\mu|(1+a)|\lambda| + |\mu| a = 0 \tag{43}$$

6) *IF* $a < 0$ *AND* $1+a > \frac{a}{|\lambda|}$

$$\begin{cases} \phi_\lambda = \pi \\ \phi_k = 0 \end{cases} \tag{44}$$

And then

$$|\lambda|^2 - |\mu|(1+a)|\lambda| - |\mu| a = 0 \tag{45}$$

7) *IF* $a < 0$ *AND* $1+a < \frac{a}{|\lambda|}$

$$\phi_\lambda = \phi_k = \pi \tag{46}$$

And then

$$-|\lambda|^2 - |\mu|(1+a)|\lambda| - |\mu| a = 0 \tag{47}$$

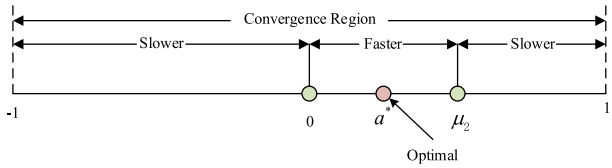


FIGURE 6. The convergence performance of the second-order algorithm with different parameter a.

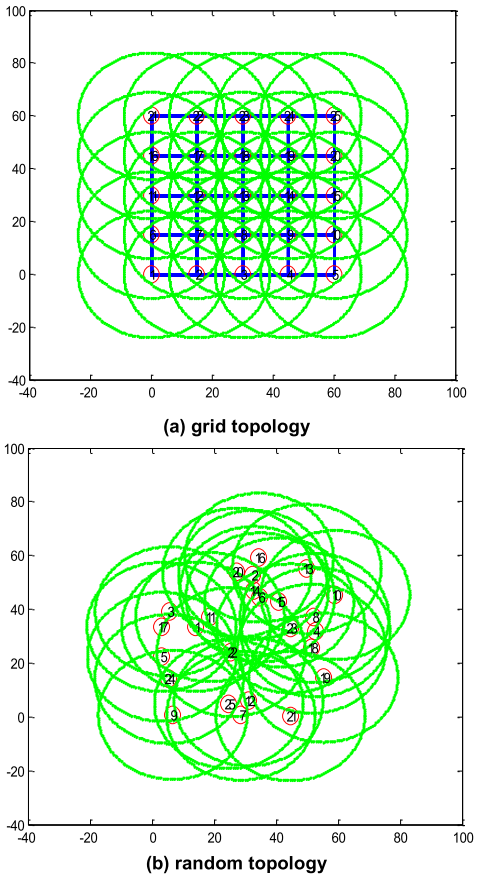


FIGURE 7. The topological structure diagrams of (a) grid topology and (b) random topology.

The convergence analysis of these four cases is similar to some cases when μ_i is a real number.

Based on the above analysis, we can draw the following conclusions: It is assumed that the classical model is convergent, and μ_2 is the second largest eigenvalue of W, then the second-order model has the following characteristics:

When $-1 < a < 1$, the second-order model is convergent.

When a=0 or $a = \mu_2$, the second-order model is convergent as fast as the classical model.

When $-1 < a < 0$ or $\mu_2 < a < 1$, the convergence rate of the second-order model is slower than that of the classical model.

When $0 < a < \mu_2$, the convergence rate of the second-order model is faster than that of the classical model.

When $a = a^* = \frac{2-\mu_2-2\sqrt{1-\mu_2}}{\mu_2}$, the second-order model has the fastest convergence rate.

All the conclusion mentioned above can be presented as Fig. 6.

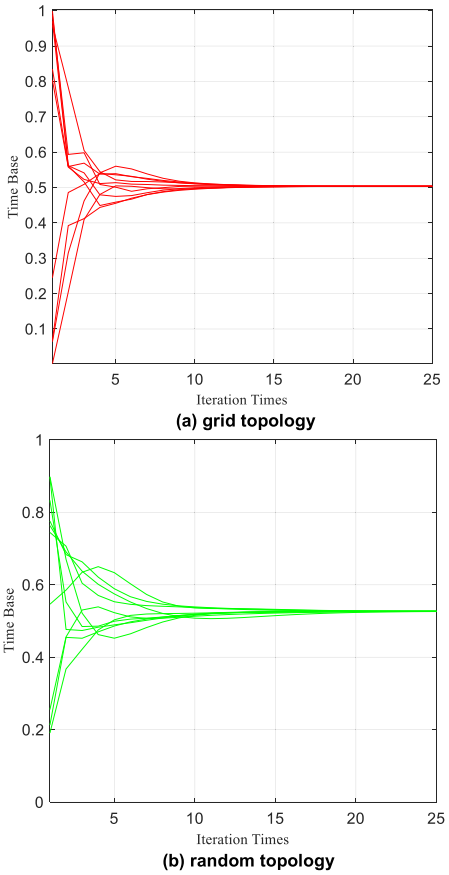


FIGURE 8. Time base adjustment of each node using the Metropolis weight matrix with $N = 25$, $R = 100$, $r = 0.4R$ and $a = a^*$ in (a) grid topology and (b) random topology.

V. SIMULATION AND ANALYSIS

A. SIMULATION SCENE

The simulation scene settings are as follows: N nodes distribute in the two-dimensional space of $R \times R$, and the communication radius of each node is r. Provided that the coordinates of node i and node j is (x_i, y_i) and (x_j, y_j) respectively, and if the Euclidean distance d_{ij} between node i and j is less than r, node i and node j can communicate directly.

There are two kinds of topology structure as shown in Fig. 7. In grid topology structure, N nodes distribute around in a rectangular area of $R \times R$, and these nodes have been randomly uniformly deployed. The distance between two nodes in grid network is $R/(\sqrt{N}-1)$. In random topology structure, N nodes randomly distribute in the square area of $R \times R$ and the connectivity probability of random networks is 0.96.

B. DIFFERENT WEIGHT MATRIX ALGORITHM SETTINGS

There are three weight matrix algorithms:

1) UNIFORM ALGORITHM

$$w_{ij} = \begin{cases} \alpha, j \in N_i \\ 1 - \alpha \left| N_i \right|, i = j \\ 0, i \neq j, j \notin N_i \end{cases} \tag{48}$$

where $\alpha \in [0, 1]$ is the optional parameter.

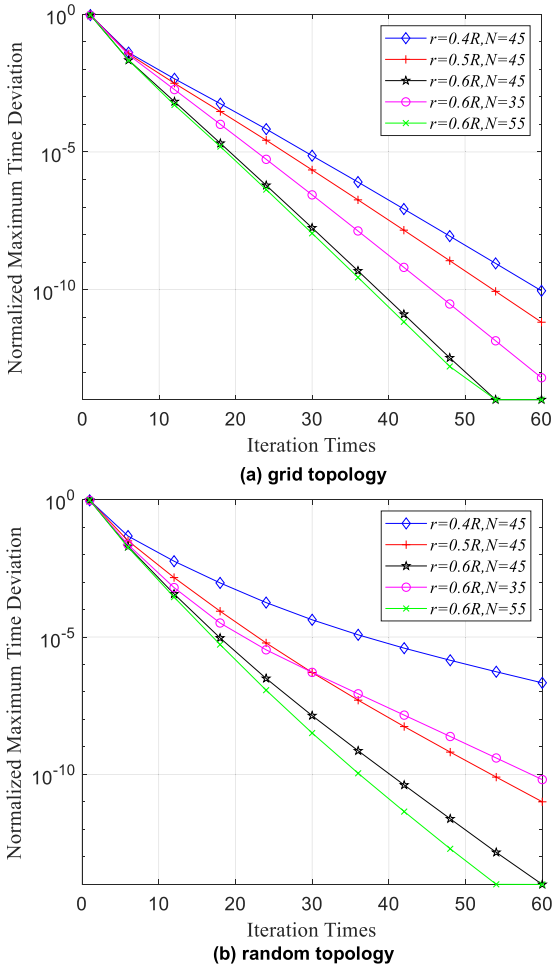


FIGURE 9. The relation between the normalized maximum time deviation and iteration times for variable communication radius r and node number N using the Metropolis weight matrix with $R = 100$ and $a = a^*$ in (a) grid topology and (b) random topology.

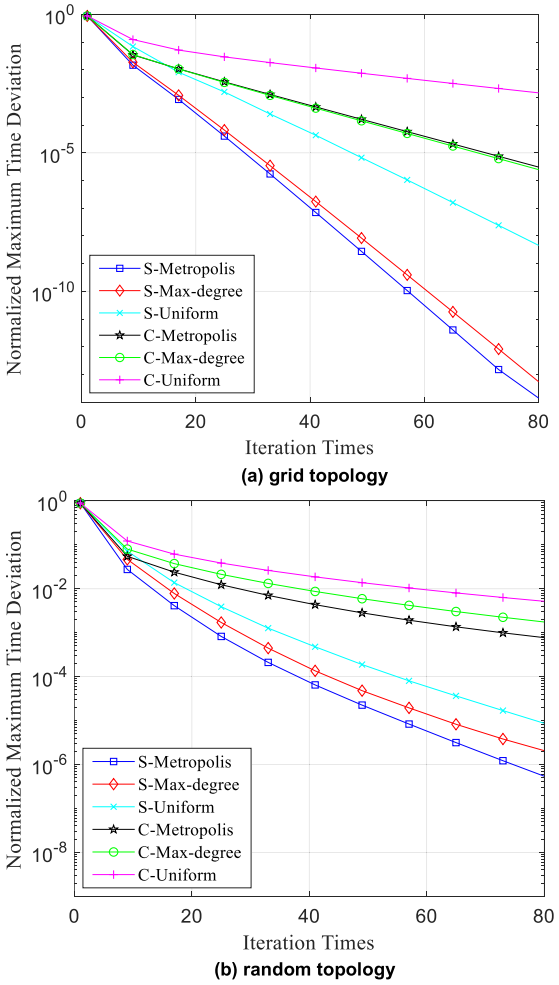


FIGURE 10. The relation between the normalized maximum time deviation and iteration times under different weight matrix algorithms with $N = 25$, $R = 100$, $r = 0.4R$ and $a = a^*$ in (a) grid topology and (b) random topology.

2) METROPOLIS ALGORITHM

$$w_{ij} = \begin{cases} \dfrac{1}{\max\{d_i, d_j\} + 1}, j \in N_i \\ 1 - \sum\limits_{j \in N_i} \dfrac{1}{\max\{d_i, d_j\} + 1}, i = j \\ 0, i \neq j, j \notin N_i \end{cases} \tag{49}$$

3) MAX-DEGREE ALGORITHM

$$w_{ij} = \begin{cases} \dfrac{1}{d+1}, j \in N_i \\ 1 - \dfrac{d_i}{d+1}, i = j \\ 0, i \neq j, j \notin N_i \end{cases} \tag{50}$$

where d is the largest node degree in the network.

C. ANALYSIS

1) THE CONVERGENCE PERFORMANCE

At the first, we will discuss the convergence of the proposed algorithm. Fig. 8 shows the time base adjustment of each node in grid topology and random topology using the Metropolis weight matrix with $N = 25$, $R = 100$, $r = 0.4\,R$ and $a = a^*$. It can be seen from the figure that the time base of each node can eventually converge to the same value after a certain number of iterations no matter in grid topology or random topology, which verifies the convergence of storage algorithm. And it can be roughly seen that the time base of each node in grid topology converges faster than that in random topology.

2) THE CONVERGENCE RATE

Next, we will discuss the convergence rate of the proposed algorithm. The simulation work has been run 1000 times.

Fig. 9 shows the relationship between the normalized maximum time deviation and the iteration times under different communication radius r and node number N using the Metropolis weight matrix in grid network topology and random network topology. It can be seen from Fig. 9 that with the increase of communication radius, the convergence rate of the algorithm increases. This is because when the

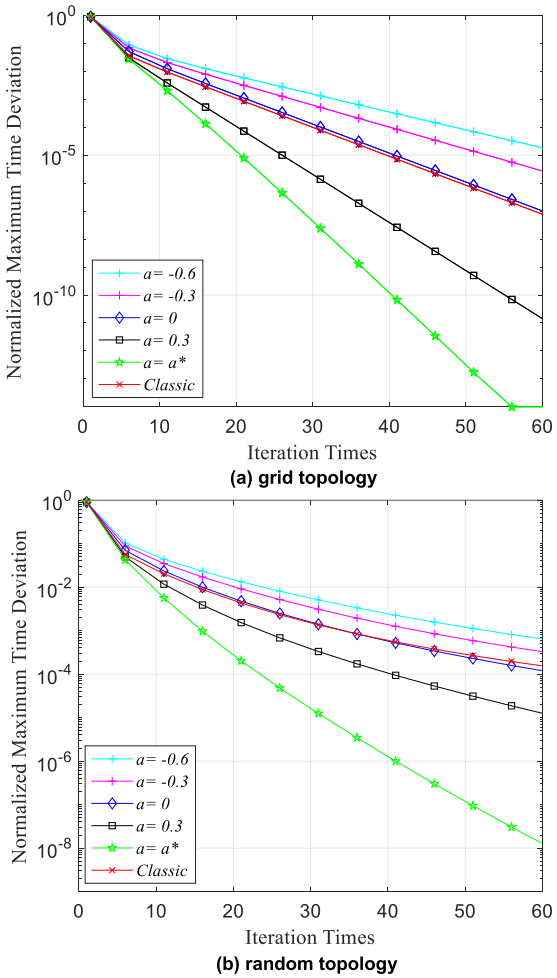


FIGURE 11. The relation between the normalized maximum time deviation and iteration times under different parameter a using the Metropolis weight matrix with $N = 25$, $R = 100$ and $r = 0.4R$ in (a) grid topology and (b) random topology.

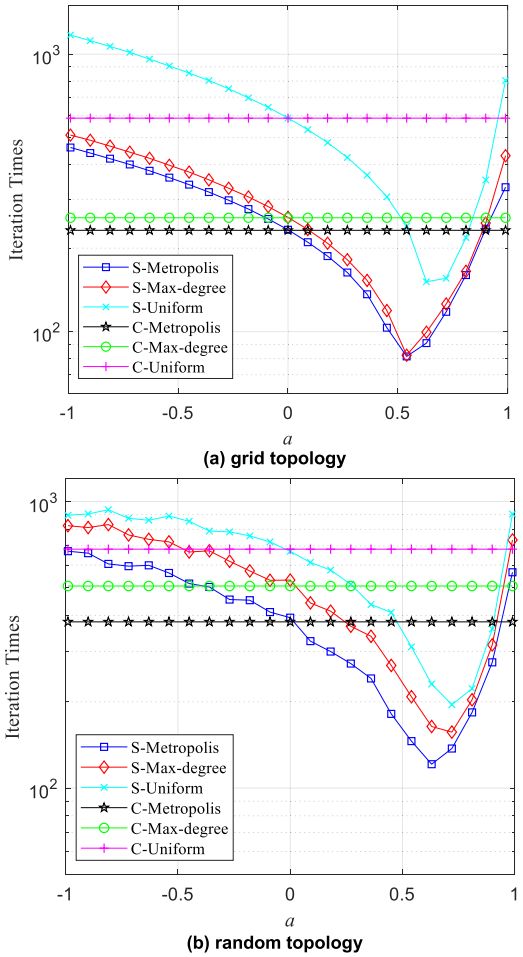


FIGURE 12. The relation between iteration times when δ=10^{-14} and the parameter a under different weight matrix algorithms with N=25, R=100 and r=0.4R in (a) grid topology and (b) random topology.

communication radius of the node increases, the number of neighbor nodes that the node can communicate increases. So the time state information of neighbor nodes that the node can obtain increases, and then it is faster for nodes to reach a consistent state. Besides, with the increase of node number N, more information is used to update the time value of nodes, and the convergence rate of the algorithm increases.

Fig. 10 shows the influence of different weight matrix algorithm selection on the convergence rate of the algorithm in grid network topology and random network topology. Where 'S' represents the proposed algorithm and 'C' represents the classical algorithm. From Fig. 10, it can be seen that no matter which weight matrix algorithm is chosen, the convergence rate of the storage system is significantly better than that of the classic system. And it can be seen that the Metropolis algorithm is slightly better than the Max-degree algorithm, while the Max-degree algorithm is slightly better than the Uniform algorithm. At the same time, the influence of three weight algorithms in grid topology and random topology on convergence is very different. The convergence rate in grid topology is faster than that in random topology.

Fig. 11 and Fig. 12 show the influence of different parameter a on the convergence rate of the algorithm in grid network topology and random network topology. Fig. 11 shows the relationship between iteration times and parameter a when the normalized maximum time deviation reaches the accuracy $\delta = 10^{-14}$, and compares the convergence rates of different synchronization algorithms with different weight matrix algorithms. Fig. 12 shows the relationship between the normalized maximum time deviation and iteration times under different parameter a using the Metropolis weight matrix . It can be seen from Fig. 11 and Fig. 12 that no matter what weight matrix is adopted, when $a = 0$ or $a = \mu_2$, the convergence rate of the storage algorithm is the same as that of the classical algorithm. When $-1 < a < 0$ or $\mu_2 < a < 1$, the convergence rate of the storage algorithm is slower than that of the classical algorithm. When $0 < a < \mu_2$, the convergence rate of the storage algorithm is faster than that of the classical algorithm. When $a = a^* = \frac{2-\mu_2-2\sqrt{1-\mu_2}}{\mu_2}$, the speed of convergence of storage system gets the optimal value. The simulation results are consistent with the theoretical derivation, which proves the correctness of the theoretical derivation.

VI. CONCLUSION

An accelerated time co-synchronization algorithm for energy-harvesting wireless sensor networks is proposed in this paper. This algorithm based on the weight sum of storage previous time reference and the current time reference to predict the next state estimate local time reference, after that, the next state local time reference is calculated by its estimate local time reference and all its neighbors estimate time reference. A general theoretical model for the novel algorithm is built to analysis its convergence characteristics. We establish a second-order model to analysis the convergence conditions, which is strictly proof based on matrix theory. Secondly, the convergence rate of the classical synchronization model and the proposed second-order synchronization model is compared theoretically by the second largest eigenvalue of weight matrix under more general assumptions. Then the influence of the algorithm parameter on the convergence rate is analyzed, based on which the optimal parameter is derived to reach the fastest convergence rate. Simulations on different scenarios shows that the theoretical analysis results are consistent with the simulation results, and the proposed algorithm gets a faster convergence rate than the classical distributed consensus algorithm.

REFERENCES

[1] Y. K. Tan and S. K. Panda, "Energy harvesting from hybrid indoor ambient light and thermal energy sources for enhanced performance of wireless sensor nodes," *IEEE Trans. Ind. Electron.*, vol. 58, no. 9, pp. 4424–4435, Sep. 2011.

[2] M. Magno, D. Boyle, D. Brunelli, B. O'Flynn, E. Popovici, and L. Benini, "Extended wireless monitoring through intelligent hybrid energy supply," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 1871–1881, Apr. 2014.

[3] F. Deng, X. Yue, X. Fan, S. Guan, Y. Xu, and J. Chen, "Multisource energy harvesting system for a wireless sensor network node in the field environment," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 918–927, Feb. 2019.

[4] W.-K. Lee, M. J. W. Schubert, B.-Y. Ooi, and S. J.-Q. Ho, "Multi-source energy harvesting and storage for floating wireless sensor network nodes with long range communication capability," *IEEE Trans. Ind. Appl.*, vol. 54, no. 3, pp. 2606–2615, May 2018.

[5] J. M. Gilbert and F. Balouchi, "Comparison of energy harvesting systems for wireless sensor networks," *Int. J. Autom. Comput.*, vol. 5, no. 4, pp. 334–347, Oct. 2008.

[6] H. Mahdavi-Doost and R. D. Yates, "Fading channels in energy-harvesting receivers," in *Proc. 48th Annu. Conf. Inf. Sci. Syst. (CISS)*, Princeton, NJ, USA, Mar. 2014, pp. 1–6.

[7] I. Flint, H.-B. Kong, N. Privault, P. Wang, and D. Niyato, "Wireless energy harvesting sensor networks: Boolean–Poisson modeling and analysis," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7108–7122, Nov. 2017.

[8] H. Azarhava and J. M. Niya, "Energy efficient resource allocation in wireless energy harvesting sensor networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 7, pp. 1000–1003, Jul. 2020.

[9] D. K. Sah and T. Amgoth, "Renewable energy harvesting schemes in wireless sensor networks: A survey," *Inf. Fusion*, vol. 63, pp. 223–247, Nov. 2020.

[10] Y. Xiong, G. Chen, M. Lu, X. Wan, M. Wu, and J. She, "A two-phase lifetime-enhancing method for hybrid energy-harvesting wireless sensor network," *IEEE Sensors J.*, vol. 20, no. 4, pp. 1934–1946, Feb. 2020.

[11] N. Aslam, K. Xia, and M. U. Hadi, "Optimal wireless charging inclusive of intellectual routing based on SARSA learning in renewable wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8340–8351, Sep. 2019.

[12] L. Pei, Z. Yang, C. Pan, W. Huang, M. Chen, M. Elkashlan, and A. Nallanathan, "Energy-efficient D2D communications underlaying NOMA-based networks with energy harvesting," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 914–917, May 2018.

[13] Q. Wu, W. Chen, D. W. K. Ng, and R. Schober, "Spectral and energy-efficient wireless powered IoT networks: NOMA or TDMA?" *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6663–6667, Jul. 2018.

[14] H. Kawagoe and M. Sugano, "Implementation of time synchronization for energy harvesting wireless sensor network," in *Proc. VI Int. Conf. Netw., Commun. Comput. (ICNCC)*, Kunming, China, 2017, pp. 175–178.

[15] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Boise, ID, USA, Apr. 2005, pp. 63–70.

[16] K. Avrachenkov, M. E. Chamie, and G. Neglia, "A local average consensus algorithm for wireless sensor networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops (DCOSS)*, Barcelona, Spain, Jun. 2011, pp. 1–6.

[17] G. Xiong and S. Kishore, "Linear high-order distributed average consensus algorithm in wireless sensor networks," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, pp. 1–6, Dec. 2010.

[18] J. Wu, L. Zhang, Y. Bai, and Y. Sun, "Cluster-based consensus time synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1404–1413, Mar. 2015.

[19] S. Khobahi and M. Soltanalian, "Optimized transmission for consensus in wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 3419–3423.

[20] F. Shi, X. Tuo, L. Ran, Z. Ren, and S. X. Yang, "Fast convergence time synchronization in wireless sensor networks based on average consensus," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1120–1129, Feb. 2020.

[21] L. Altoaimy, H. Kurdi, A. Alromih, A. Alomari, E. Alrogi, and S. H. Ahmed, "Enhanced distance-based gossip protocols for wireless sensor networks," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2019, pp. 1–4.

[22] C. Guyeux, M. Haddad, M. Hakem, and M. Lagacherie, "Efficient distributed average consensus in wireless sensor networks," *Comput. Commun.*, vol. 150, pp. 115–121, Jan. 2020.

[23] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annu. Rev. Control*, vol. 47, pp. 278–305, May 2019.

[24] A. Khosravi and Y. S. Kavian, "Challenging issues of average consensus algorithms in wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 6, no. 3, pp. 60–66, Jun. 2016.

[25] F. Mirali and H. Werner, "Distributed weighting strategies for improved convergence speed of first-order consensus," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 942–947.

[26] A. Olshevsky, "Linear time average consensus and distributed optimization on fixed graphs," *SIAM J. Control Optim.*, vol. 55, no. 6, pp. 3990–4014, Jan. 2017.

[27] J. Gutiérrez-Gutiérrez, M. Zárraga-Rodríguez, and X. Insausti, "Analysis of known linear distributed average consensus algorithms on cycles and paths," *Sensors*, vol. 18, no. 4, p. 968, Mar. 2018.

[28] H. Ma, X. Jia, and N. Cai, "Adaptive guaranteed-performance consensus control for multi-agent systems with an adjustable convergence speed," *Discrete Dyn. Nature Soc.*, vol. 2019, Jun. 2019, Art. no. 5190301.

[29] S. Kouachi, S. Dhuli, and Y. N. Singh, "Convergence rate analysis of periodic gossip algorithms for one-dimensional lattice WSNs," *IEEE Sensors J.*, vol. 20, no. 21, pp. 13150–13160, Nov. 2020.

[30] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[31] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[32] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[33] H. Hao and P. Barooah, "Improving convergence rate of distributed consensus through asymmetric weights," in *Proc. Amer. Control Conf. (ACC)*, Montreal, QC, Canada, Jun. 2012, pp. 787–792.

[34] E. Omidi and F. Abdollahi, "A distributed adaptive algorithm for speeding up consensus in networked multi-agent systems," in *Proc. 3rd Int. Conf. Control, Instrum., Autom.*, Tehran, Iran, Dec. 2013, pp. 323–327.
[35] E. Montijano, J. I. Montijano, and C. Sagues, "Chebyshev polynomials in distributed consensus applications," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 693–706, Feb. 2013.
[36] S. Apers and A. Sarlette, "Accelerating consensus by spectral clustering and polynomial filters," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 544–554, Sep. 2017.
[37] D. Yuan, S. Xu, H. Zhao, and Y. Chu, "Accelerating distributed average consensus by exploring the information of second-order neighbors," *Phys. Lett. A*, vol. 374, no. 24, pp. 2438–2445, May 2010.
[38] Z. Jin and R. M. Murray, "Multi-hop relay protocols for fast consensus seeking," in *Proc. 45th IEEE Conf. Decis. Control*, San Diego, CA, USA, Dec. 2006, pp. 1001–1006.
[39] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed average consensus: Beyond the realm of linearity," in *Proc. Conf. Rec. 43rd Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2009, pp. 1337–1342.
[40] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *Proc. 44th Annu. Allerton Conf. Commun., Control, Comput.*, 2006, pp. 952–959.
[41] J. Liu, B. D. O. Anderson, M. Cao, and A. S. Morse, "Analysis of accelerated gossip algorithms," *Automatica*, vol. 49, no. 4, pp. 873–883, Apr. 2013.
[42] D. S. Scherber and H. C. Papadopoulos, "Distributed computation of averages over ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 776–787, Apr. 2005.
[43] B. N. Oreshkin, T. C. Aysal, and M. J. Coates, "Distributed average consensus with increased convergence rate," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Las Vegas, NV, USA, Mar. 2008, pp. 2285–2288.
[44] D. Irofti, "An anticipatory protocol to reach fast consensus in multi-agent systems," *Automatica*, vol. 113, Mar. 2020, Art. no. 108776.
[45] F. M. Atay and D. Irofti, "A delayed consensus algorithm in networks of anticipatory agents," in *Proc. Eur. Control Conf. (ECC)*, Aalborg, Denmark, Jun. 2016, pp. 1880–1885.
[46] D. Irofti and F. M. Atay, "On the delay margin for consensus in directed networks of anticipatory agents," *IFAC-PapersOnLine*, vol. 49, no. 10, pp. 206–211, Jul. 2016.
[47] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 342–354, Jan. 2009.
[48] B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *IEEE Trans. Signal Process.*, vol. 58, no. 5, pp. 2850–2865, May 2010.
[49] H. Wang, X. Liao, and T. Huang, "Accelerated consensus to accurate average in multi-agent networks via state prediction," *Nonlinear Dyn.*, vol. 73, nos. 1–2, pp. 551–563, Feb. 2013.
[50] S. Muthukrishnan, B. Ghosh, and M. H. Schultz, "First-and second-order diffusive methods for rapid, coarse, distributed load balancing," *Theory Comput. Syst.*, vol. 31, no. 4, pp. 331–354, Jul/Aug. 1998.
[51] J. Liu and A. S. Morse, "Accelerated linear iterations for distributed averaging," *Annu. Rev. Control*, vol. 35, no. 2, pp. 160–165, Dec. 2011.

QI YANG (Member, IEEE) received the B.E. and M.S. degrees from the Department of Communication Engineering, Fuzhou University, Fuzhou, China, in 2000 and 2003, respectively, and the Ph.D. degree from the Department of Communication Engineering, Xiamen University, Xiamen, China, in 2008. From 2003 to 2005, he was a Teaching Assistant with the School of Information Science and Technology, Fuzhou University. From 2008 to 2015, he was an Assistant Professor with the School of Information Science and Technology, Xiamen University. From 2016 to 2019, he visited the Department of Electrical and Computer Engineering, University of Waterloo, Canada, and the Department of Engineering Science, National Cheng Kung University, Taiwan. He is currently an Associate Professor with the School of Informatics, Xiamen University. His research interest includes wideband wireless communication and networks.

RONGPING ZHENG received the B.E. degree in communication engineering from the University of Science and Technology Beijing, Beijing, China, in 2018. She is currently pursuing the M.S. degree with the School of Information Science and Engineering, Xiamen University, Fujian, China. Her current research interests include related issues of wireless communication and vehicular ad-hoc networks.

JUNYU GUO received the B.E. degree from Nanchang University, China, in 2018. He is currently pursuing the M.S. degree with the School of Information Science and Engineering, Xiamen University, Fujian, China. His current research interests include related issues of LAA-WiFi coexistence and vehicular ad-hoc networks.

TAO CHEN received the B.S. degree in electronic information engineering from Northeast Forestry University, in July 2010, and the M.S. degree in communications engineering from Xiamen University, in July 2013. He is currently a Supervisor of access network with Jiangxi Telecom Company. His research interests includes algorithm and optimization in access networks.