Hindawi
Mathematical Problems in Engineering
Volume 2018, Article ID 4635701, 13 pages
https://doi.org/10.1155/2018/4635701

Research Article

Asynchronous Communication under Reliable and Unreliable Network Topologies in Distributed Multiagent Systems: A Robust Technique for Computing Average Consensus

Ali Mustafa ,[1,2] Muhammad Najam ul Islam,[1] Salman Ahmed,[3] and Muhammad Ahsan Tufail[2]

[1]*Department of Electrical Engineering, Bahria University, Islamabad, Pakistan*
[2]*Department of Electrical Engineering, COMSATS Institute of IT, Attock, Pakistan*
[3]*Department of Computer Systems Engineering, University of Engineering and Technology, Peshawar, Pakistan*

Correspondence should be addressed to Ali Mustafa; ali.mustafa@ciit-attock.edu.pk

Received 8 September 2017; Accepted 31 January 2018; Published 11 March 2018

Academic Editor: Ton D. Do

Nearly all applications in multiagent systems demand precision, robustness, consistency, and rapid convergence in designing of distributed consensus algorithms. Keeping this thing in our sight, this research suggests a robust consensus protocol for distributed multiagent networks, continuing asynchronous communications, where agent's states values are updated at diverse interval of time. This paper presents an asynchronous communication for both reliable and unreliable network topologies. The primary goal is to delineate local control inputs to attain time synchronization by processing the update information received by the agents associated in a communication topology. Additionally in order to accomplish the robust convergence, modelling of convergence analysis is conceded by commissioning the basic principles of graph and matrix theory alongside the suitable lemmas. Moreover, statistical examples presenting four diverse scenarios are provided in the end; produced results are the recognisable indicator to authenticate the robust effectiveness of the proposed algorithm. Likewise, a simulation comparison of the projected algorithm with the other existing approaches is conducted, considering different performance parameters are being carried out to support our claim.

1. Introduction

Multiagent systems can be defined as a system that is distributed in nature; multiple agents in a network communicate with each other with their limited resources to achieve their local and global goal in a cooperative manner. It is important to mention here that agents are intelligent nodes and have some degree of independence to take their decisions to achieve their desired goals. Such coordination in multiagent system is referred to as a consensus, and it is considered as one the fundamental research activities in distributed control strategy. Study of multiagent systems attracted scientists and researchers from diverse fields of control theory, applied mathematical modelling, computer sciences, engineering, and so on to study the significance of distributed control of agents in a network, as distributed multiagents systems gain a lot of significance in various real time applications such as flocking [1–4], wireless sensor networks (WSN) [5, 6], coordination control of robots [7, 8], underwater vehicles, unmanned aerial vehicles [9, 10], intelligent transportation systems [11], robotics [12], formation control [13, 14], balancing of load in parallel computers [15], and intelligent decentralized mechanism for smart grids [16, 17].

Consensus problem was first investigated in late 1960s in the modelling of statistical theory and management science [18]. Moreover in the fields of control system theory, consensus problem was initially discussed by [19]. Computer sciences problem focusing on animal aggregation through consensus control is studied by the author in [20]. In latest research horizons of consensus control is pioneered by Olfati-Saber and Murray; they utilize the concepts of graph and matrix theories along with the different network topologies with directed and undirected graph connectivity [21, 22]. Later on authors in [23] implement the consensus control for dynamically changing topologies. Moreover, consensus in switching network topologies is addressed by [24, 25], filter designing for consensus is addressed by [26], optimized solution for large scale industry using distributed consensus algorithm is proposed by the authors in [27], and Kalman filtering design for distributed consensus approach is initially addressed in [28]. Impulsive control as an important mechanism is highlighted in [29]. Time synchronized algorithms for various application are discussed by authors in [30–32]. Likewise, concept of asynchronous consensus under different network environments is presented by authors in [33, 34].

After concisely discussing various theories and the existing approaches, we are suggesting a new control mechanism for attaining robust average consensus for asynchronous communication in distributed multiagent network centred on discrete time information sharing among the neighbours within the communication network. Practically such networks may suffer communication delays, information loss, broken communication links, unreliable switching network topology, and limitation in channel capacity. So in our proposed solution, we are considering resource constrained intelligent agents with limited communication abilities and also simulated the proposed network control mechanism under both reliable and unreliable random network topologies. Most importantly as mentioned above it is asynchronous in nature, which means that each agent communicating in a network is adjusting its dynamics independently with respect to other neighbouring agents. In proposed scheme starting time of each agent is common but the update time of each agent in a network is assigned randomly; in other words we can say update time of each agent is asynchronous and depending on upper and lower bound of the time. For designing of the proposed control algorithm we used the key concepts from the theory of graph and nonnegative matrix along with state space model to prove the convergence conditions. Proposed algorithm is compared with other existing protocols with a synchronous communication in four different scenarios under reliable and unreliable network topologies and results are compared it a tabular form consisting of various performance parameters. Structure of the paper is organised as follows. Key concepts from the matrix theory and algebraic graph theory are presented in Section 2, convergence analysis is proved in Section 3, and Section 4 is enriched with the proposed control algorithm. Numerical simulation and comparison analysis of performance parameters are presented in Section 5. Section 6 is devoted for concluding remarks.

2. Preliminaries from Algebraic Graph and Matrix Theory

Algebraic graph theory and nonnegative matrix theory are considered as a preliminary tool for designing and convergence analysis in distributed algorithms in multiagent systems. In this segment significant fundamental concepts are concisely deliberated to enhance the understanding of the projected investigation.

In algebraic graph theory, graph is fundamentally a group of linked agents communicating with each other through communication links. Mathematically it can be represented as $G = (V, E)$. Agents set in a graph are called vertex set and they can be written as $V = v_1, v_2, \ldots, v_n$, where $1, 2, n$ are the number of agents in a graph. Moreover, connection links between vertices are referred to as an edge set $\mathscr{E} = (i, j)$. Graph can be classified into two types, directed graph and undirected graph. In directed communication between the agents is one way (i, j) demonstrating data flowing from jth vertex to ith and vice versa. Similarly in undirected graph the communication between vertices is bidirectional. Furthermore, edge set consists of order pairs of vertices representing the information flow. Directly connected vertices which are directly communicating with each other through edges are known as neighbour and algebraically notated as $\mathscr{N}_i$ for agent i. In graph theory connectivity of the agents in a network is represented by $n * n$ matrix, known as adjacency matrix $A = [a_{ij}]$. Degree matrix $D = [d_{ij}]$ delivers the information about the total neighbours connected with any agent in a communication network. Additionally if one vertex in a network has the direct path to all the vertices in a network, then the connectivity graph is supposed to procedure a spanning tree. Spanning tree is considered as one vital convergence condition for consensus in reliable communication but not considerably enough for the systems with time varying delays [35]. If edges in a graph have weights then the matrix is called weighted matrix and summation of the weights coming towards the vertex is called in-degree and alike sum of outward weight is said to be out-degree [36, 37]. Few other matrices which are considered to be of prime importance in convergence analysis for the distributed consensus algorithms are named Laplacian matrix $L = D - A$, rank 1 matrix, stochastic matrix P, and SIA matrix [38].

Furthermore, when designing the weighting matrix β to validate the convergence condition towards a targeted value, then there are multiple approaches available to design such matrices. Local degree weight matrix is one of the popular methodologies, which states that the highest out-degree of any two incident agents is assigned as weight on each edge [39]. It is significant to mention here that it is compulsory that each agent in a network must possesses the knowledge of out-degrees of all of its neighbors. Mathematically we can express

$$\beta_{ij}\left(t_k^i\right) = \begin{cases} \dfrac{1}{\left(\left(\max\left(d_i\left(t_k^i\right), d_j\left(t_k^i\right)\right)\right)\right)} & i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Similarly one of the popular techniques is for designing weigh matrix as metropolis hasting weights. It can be defined as the fact that all agents must know the out-degree of all of its neighbors but it is essential to mention here that the graph topology is changing at all instants of time. It can be algebraically expressed as [40, 41]

$$\beta_{ij}\left(t_k^i\right) = \begin{cases} \dfrac{1}{\left(\left(\max\left(d_i\left(t_k^i\right), d_j\left(t_k^i\right)\right)\right)\right) + 1} & i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Moreover, while in designing process of distributed consensus algorithm, some useful lemmas and facts from the theory of controls play a very significant role to produce desire results to achieve convergence.

Lemma 1. *Assume a collection of stochastic matrices $Z_1, Z_2, \ldots, Z_n$. The final graph which is produced after performing the union operation of all the graphs in matrices set $\mathscr{G}(Z_1), \mathscr{G}(Z_2), \ldots, \mathscr{G}(Z_n)$ must possess the spanning tree; afterwards the produced matrix outcome $Z_n Z_{n-1} \cdots Z_1$ is called a SIA matrix [23].*

Lemma 2. *Stochastic matrix Z is supposed to be a SIA, if and only if a spanning tree in $\mathscr{G}(Z)$ is present [42].*

3. Convergence Condition

This section will deal with the convergence analysis of the proposed algorithm based on the given below distributed iteration:

$$z_i\left(t_k^i + 1\right) = \beta_{ii}\left(t_k^i\right) z_i\left(t_k^i\right) + \sum_{j \epsilon N_i} \beta_{ij}\left(t_k^i\right) z_j\left(t_k^i\right); \tag{3}$$

where i ranges between $i = 1, 2, 3, \ldots, n$, and similarly $t_k^i = 0, 1, 2, \ldots$.

β_{ij} is the weight on z_j at agent i, where $\beta_{ii} = (1 - \sum_{j \epsilon N_i} \beta_{ij})$. By assigning $\beta_{ij} = 0$ for $j \epsilon N_i$, we can write (3) in a vector form as

$$Z\left(t_k^i + 1\right) = \beta\left(t_k^i\right) Z\left(t_k^i\right) \tag{4}$$

by utilizing the concept of a t-step transition matrix, we can state that

$$\beta\left(t_k^i\right) = \beta(t-1) \cdots \beta(1)\,\beta(0), \tag{5}$$

where in (4)

$$Z\left(t_k^i + 1\right) = \begin{bmatrix} Z_1\left(t_k^1 + 1\right) \\ Z_2\left(t_k^2 + 1\right) \\ \cdots \\ \cdots \\ Z_n\left(t_k^n + 1\right) \end{bmatrix}$$

$$Z\left(t_k^i\right) = \begin{bmatrix} Z_1\left(t_k^1\right) \\ Z_2\left(t_k^2\right) \\ \cdots \\ \cdots \\ Z_n\left(t_k^n\right) \end{bmatrix} \tag{6}$$

$\beta \epsilon \psi$ represents the sparsity design of the matrix β with restraint, where

$$\psi = \left\{\beta \epsilon R^{n \times n} \mid \beta_{ij} = 0 \text{ if } \{i, j\} \notin \mathscr{E},\ i \neq j\right\} \tag{7}$$

After defining the sparsity pattern of the matrix, we can now transform (4) as

$$Z\left(t_k^i\right) = \left(\prod_{i=0}^{t_k^i - 1} \beta\left(t_k^i\right)\right) Z(0). \tag{8}$$

Here it is important to highlight that (8) is valid for all t_k^i:

$$Z\left(t_k^i\right) = \beta\left(t_k^i\right) Z(0). \tag{9}$$

Now the primary goal is to achieve a consensus and to select a weighting matrix in such a way that the convergence condition is satisfied and $Z(t_k^i)$ converge to the average vector, with any initial value $Z(0)$:

$$\lim_{t_k^i \to \infty} Z\left(t_k^i\right) = \left(\frac{1}{n}\right) 11^T Z(0), \tag{10}$$

where n represents the number of participating agents in communication network and a 1 is a vector of ones:

$$\lim_{t_k^i \to \infty} Z\left(t_k^i\right) = \lim_{t_k^i \to \infty} \beta\left(t_k^i\right) Z(0). \tag{11}$$

If we now compare (10) with (11), we will attain

$$\lim_{t_k^i \to \infty} \beta\left(t_k^i\right) Z(0) = \left(\frac{1}{n}\right) 11^T Z(0). \tag{12}$$

Here we compare the terms in (12), and we will come up with

$$\lim_{t_k^i \to \infty} \beta\left(t_k^i\right) = \left(\frac{1}{n}\right) 11^T. \tag{13}$$

From (13) we can compute the equations for different performance parameters for convergence analysis; the asymptotic convergence factor [26] is calculated as

$$r_{\text{asym}}\left(\beta\left(t_k^i\right)\right) = \sup_{z(0) \neq \dot{z}} \left(\frac{\left\|z\left(t_k^i\right) - \dot{z}\right\|_2}{\left\|z(0) - \dot{z}\right\|_2}\right)^{1/t_k^i}, \tag{14}$$

where $\dot{z} = \lim_{t \to \infty} z(t_k^i)$.

Generally convergence is mentioned as a degree of error reduction with refinement towards zero for finite elements with grid spacing. Similarly asymptotic convergence is defined as a convergence behaviour of the system when grid arrangement is considered very trivial in a matrix and error between two states ranges towards zero.

From (14), the convergence time can be computed as

$$\tau_{\text{asym}} = \frac{1}{\log\left(1/r_{\text{asym}}\left(\beta\left(t_k^i\right)\right)\right)}. \tag{15}$$

Convergence time of any system is defined as a computation time, exactly how speedily different agents in a network touch a state of consistent convergence. It is considered to be one of the foremost goals and vital performance indicators in designing of distributed algorithms.

One of the other important performance indicators used for network convergence is matrix spectral radius and it is denoted by $\rho(\cdot)$. It is a biggest absolute value of its eigenvalues in the matrix spectrum. Mathematically it can be expressed as

$$r_{\text{asym}}(\beta) = \rho\left(\beta - \frac{11^T}{n}\right). \tag{16}$$

4. Proposed Algorithm

In the field of distributed multiagent systems, its primary impression that consensus algorithms enforce the network agents communicating in a network towards a common value by information exchange by their neighbourhood if the communication is reliable; network topology is considered fixed. But achieving consensus for the cases in which network connectivity is unreliable, communication is asynchronous and update time for the network agents varying time to time is really challenging.

In this section we are proposing a consensus algorithm for distributed network consisting of n autonomous agents (i.e., $1, 2, \ldots, n$), whose directed graph topology is considered as a fixed (reliable) and for switching topology (unreliable). Agents are sharing a common state space $\mathbb{R}$ with asynchronous communication in terms of different update time for each agent. Initial start time for all agent is considered the same but later on each agent adjusts its current state value, depending on the information received from other set of defined neighbors $N(t, i)$ at a particular instance of time. Let $z_i(t^i_k)\epsilon\mathbb{R}$ represents the initial state of an agent i and let $Z = [z_1, z_2, \ldots, z_n]^T$. We can represent designed discrete time multiagent system as $z(t^i_k + 1) = f(t^i_k, u(t^i_k))$, where $u(t^i_k)$ is a state feedback controller which needs to be designed as well with the help of weighting matrix. If we suppose $z(t^i_k)$ as an initial value and it converges to a stable value $\dot{z}$, in such away that $\dot{z}_i = \dot{z}_j$ for all i, j as $t \to \infty$, then it is said to achieve consensus. If agent i receives the state information of its neighbors at t^i_k, then agent i is assumed to take the following dynamics in time interval (t^i_k, t^i_{k+1}):

$$\dot{z}_i(t) = z_i(t) + \sum_{j\epsilon N_{(t^i_k,i)}} \beta_{ij}\left(t^i_k\right)\left(z_j\left(t^i_k\right) - z_i(t)\right) \qquad \text{if } N_{(t^i_k,i)} \neq \phi, \tag{17}$$

where $t\varepsilon[t^i_k, t^i_{k+1})$.

We assume that agent i receives its neighbors states at update times $t^i_0, t^i_1, \ldots, t^i_k, \ldots$ which can be denoted as $\{t^i_k\}$. In proposed model we are considering the following condition, which presumed to be essentially contented by $\{t^i_k\}$.

Condition. For any $k\epsilon\mathbb{Z}_+, 0 < \check{T}_u \leq t^i_{k+1} \leq \widehat{T_u}$, where $\check{T}_u, \widehat{T_u}$.

The core intention behind the update times is to acquire an information of neighbours state value by an agent i to apprise its dynamic consequently. Two time bounds are presented in the proposed algorithm, upper bound $\widehat{T_u}$ and lower bound $\check{T}_u$. Both have their own importance and significance in convergence analysis under asynchronous time update communication. Lower bound $\check{T}_u$ of time interval among any two consecutive update time offers assurance to the authentication of the consensus algorithm, while on the other hand upper bound $\widehat{T_u}$; of $t^i_{k+1} - t^i_k$ supports the dissimilar states of an agents to assemble themselves on a common value.

While designing a proposed algorithm, primary goals are to achieve quick convergence, reduce the iterations for bounded convergence, and also meet the best of performance indicators as mentioned in the previous section.

Proposed algorithm can be mathematically expressed as

$$z_i\left(t^i_{k+1}\right) = z_i\left(t^i_k\right) + \sum_{j\epsilon N_{(t^i_k,i)}} \beta_{ij}\left(t^i_k\right)\left(z_j\left(t^i_k\right) - z_i\left(t^i_k\right)\right) \tag{18}$$

we can write

$$\beta_{ij}\left(t^i_k\right) = \begin{cases} \dfrac{\left(d_i\left(t^i_k\right) + d_j\left(t^i_k\right)\right)}{\left(2d_i\left(t^i_k\right) * d_j\left(t^i_k\right)\right)} & i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

In the above equation d_i is the degree of agent i or we can say the total neighbors of an agent i.

Proof of an Algorithm. Suppose

$$Z\left(t^i_k\right) = \begin{bmatrix} Z_1\left(t^1_k\right) \\ Z_2\left(t^2_k\right) \\ \cdots \\ \cdots \\ Z_n\left(t^n_k\right) \end{bmatrix}, \qquad U\left(t^i_k\right) = \begin{bmatrix} u_1\left(t^1_k\right) \\ u_2\left(t^2_k\right) \\ \cdots \\ \cdots \\ u_n\left(t^n_k\right) \end{bmatrix}. \tag{20}$$

Similarly

$$Z\left(t^i_k + 1\right) = \begin{bmatrix} z_1\left(t^1_k + 1\right) \\ z_2\left(t^2_k + 1\right) \\ \cdots \\ \cdots \\ z_n\left(t^n_k + 1\right) \end{bmatrix}. \tag{21}$$

At this point it is significant to articulate a global state equation for a system, which is intended for average convergence in the given form as

$$Z\left(t^i_k + 1\right) = Z\left(t^i_k\right) + U\left(t^i_k\right). \tag{22}$$

Global input vector can be formulated as

$$U\left(t_k^i\right) = \beta_{ij}\left(t_k^i\right)[A - D]\, Z\left(t_k^i\right). \tag{23}$$

For unreliable network topology the weighting matrix β is not fixed, so for the particular case we used the following symbolization $\beta_{ij}(t_k^i)$ in (23). After deputizing (23) in (22), we will get

$$Z\left(t_k^i + 1\right) = Z\left(t_k^i\right) + \beta_{ij}\left(t_k^i\right)[A - D]\, Z\left(t_k^i\right). \tag{24}$$

We can additionally write the following after applying the concept of Laplacian matrix:

$$Z\left(t_k^i + 1\right) = Z\left(t_k^i\right) - L\beta_{ij}\left(t_k^i\right) Z\left(t_k^i\right), \tag{25}$$

$$Z\left(t_k^i + 1\right) = \left[I - L\beta_{ij}\left(t_k^i\right)\right] Z\left(t_k^i\right). \tag{26}$$

Distributed linear iteration in (26) designates for $t_k^i = 0, 1, 2, \ldots$:

$$Z\left(t_k^i\right) = \left[I - L\beta_{ij}\left(t_k^i\right)\right] Z(0), \tag{27}$$

$$\lim_{t_k^i \to \infty} Z\left(t_k^i\right) = \lim_{t_k^i \to \infty} \left(I - \beta_{ij}\left(t_k^i\right) L\right) Z(0). \tag{28}$$

Now equating (11) and (28),

$$\lim_{t_k^i \to \infty} \beta\left(t_k^i\right) = \lim_{t_k^i \to \infty} \left(I - \beta_{ij}\left(t_k^i\right) L\right). \tag{29}$$

Now referring to (13)

$$\lim_{t_k^i \to \infty} \beta\left(t_k^i\right) = \left(\frac{1}{n}\right) 11^T. \tag{30}$$

Therefore,

$$\lim_{t_k^i \to \infty} \left(I - \beta_{ij}\left(t_k^i\right) L\right) = \left(\frac{1}{n}\right) 11^T. \tag{31}$$

If we consider a system, with infinite number of agents, that is, $n \to \infty$, then we can write

$$\lim_{t_k^i \to \infty} \left(I - \beta_{ij}\left(t_k^i\right) L\right) = 0. \tag{32}$$

The concluding outcome of (32) will develop as

$$\beta_{ij}\left(t_k^i\right) = \begin{cases} \dfrac{\left(d_i\left(t_k^i\right) + d_j\left(t_k^i\right)\right)}{\left(2d_i\left(t_k^i\right) * d_j\left(t_k^i\right)\right)} & i \neq j \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

□

5. Numerical Examples and Simulation Results

In this section stimulation results are presented for various numerical examples. All examples are simulated in Matlab environment for generating results. This section also validate the efficiency of projected algorithm in four different scenarios for asynchronous time update communication under reliable and unreliable communication. To compare the benchmarks of proposed algorithm, a detailed simulation comparison with other existing approaches is performed. For that particular purpose four different performance parameters, that is, total number of iterations, CPU time, asymptotic convergence factor, and convergence time, have been considered to compute the results for each scenario. Total number of iterations is basically the iteration count in which the consensus is achieved using Matlab software and similarly CPU time is the processing time of the machine on which all the simulation are carried out with same computational resources. Error graphs of proposed method and other existing approaches are also produced in this section for better understanding. Initial values are assigned to agents as $x_n(0) = 1$ and $x_i(0) = 0$, $\forall i = 1, 2, 3, \ldots, n - 1$ in all four numerical examples. For any $i\varepsilon I_n, t_{k+1}^i - t_k^i$ is evenly distributed between 0.2 and 0.6, lower time bound is $\check{T}_u = 0.2$, and upper time bound is $\widehat{T_u} = 0.6$. Additionally error tolerance considered is $e = 10^{-15}$, in which the total number of iterations is represented by NI, where

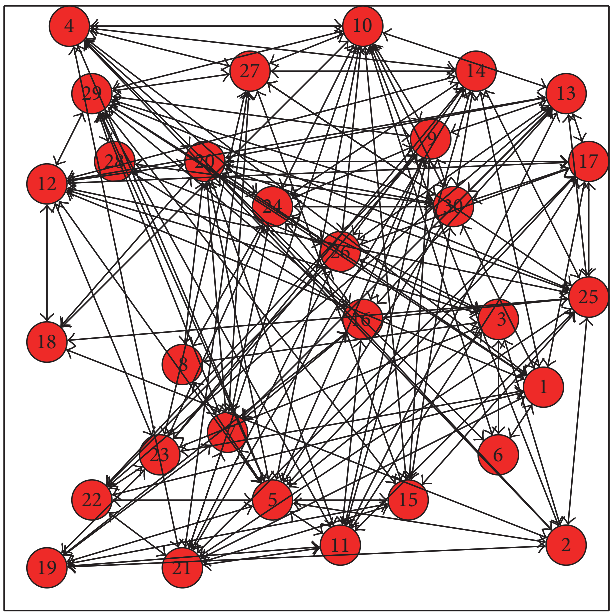

Figure 1: Network topology considered in Example 1.

$$e_i(k) = \sum_{j \epsilon N_i} \left| x_i(k) - x_j(k) \right|, \quad i = 1, 2, \ldots, n. \tag{34}$$

Example 1 (fixed network topology with asynchronous time update). In this example, a case of reliable communication among multiagents with asynchronous time update is considered (Figure 1). Communication topology remains fixed throughout the simulation but the update times of agents are diverse and independent of each others. Each agent can get all of its neighbour states only at its update times. For any $i\varepsilon I_n, t_{k+1}^i - t_k^i$ is evenly distributed between 0.2 and 0.6. A network of 30 agents is considered which are communicating with each other under reliable conditions. All agents have to develop an average consensus on the value of 0.0333. To

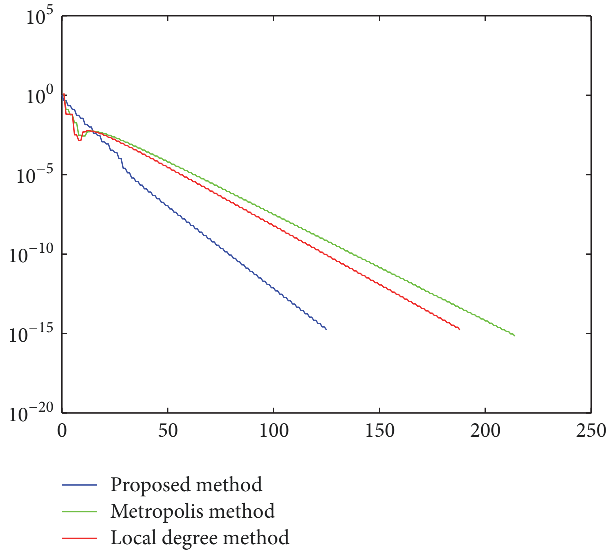

FIGURE 2: Plot of error graph generated for Example 1.

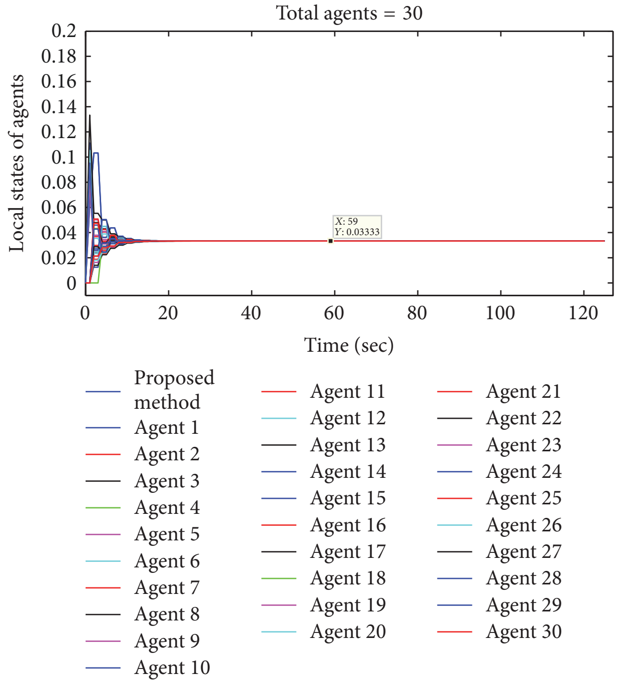

FIGURE 3: Agents approaching to a consensus value using proposed algorithm in Example 1.

evaluate the performance of proposed algorithm, the comparison results of Example 1, simulated by proposed and existing methods are shown in Table 1. An error graph is generated in Figure 2, which evaluates how fast a method converges to its consensus value. It also helps us to graphically evaluate the performance of different methods. The consensus graphs of proposed method, Metropolis method, and local degree method are shown in Figures 3, 4, and 5 respectively.

Example 2 (unreliable switching network topology with asynchronous time update). The second case is on unreliable switching topologies with asynchronous update time. In this example two major challenges in cooperative control are highlighted, that is, unreliable communication among agents and asynchronous behavior in update times. When the communication topology is changing dynamically and

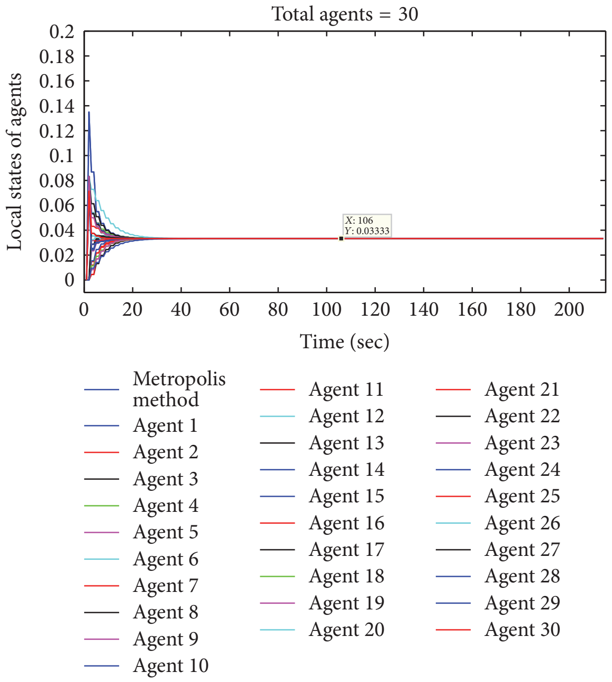

FIGURE 4: Agents approaching to a consensus value using Metropolis method in Example 1.

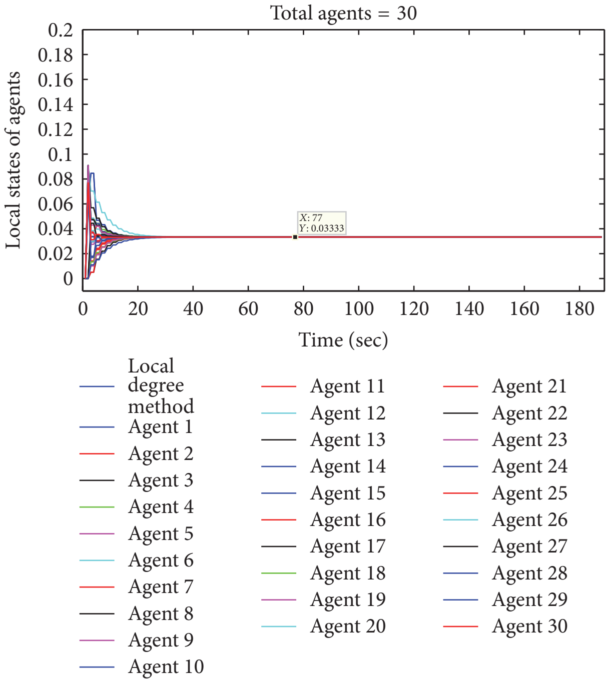

FIGURE 5: Agents approaching to a consensus value using local degree method in Example 1.

Table 1: Comparison results of Example 1 for different performance parameters.

Methods	NI	CPU_{Min}	$\rho(W - 11^T/n)$	τ_{asym}
Proposed method	127	0.342	0.7518	3.505
Metropolis method	215	0.3972	0.8461	6.003
Local degree method	189	0.3617	0.8284	5.3131

Table 2: Comparison results of Example 2 for different parameters.

Methods	NI	CPU_{Min}	$\rho(W - 11^T/n)$	τ_{asym}
Proposed method	163	0.2092	0.7998	4.778
Metropolis method	245	0.2868	0.8635	6.8173
Local degree method	215	0.2540	0.8456	5.9660

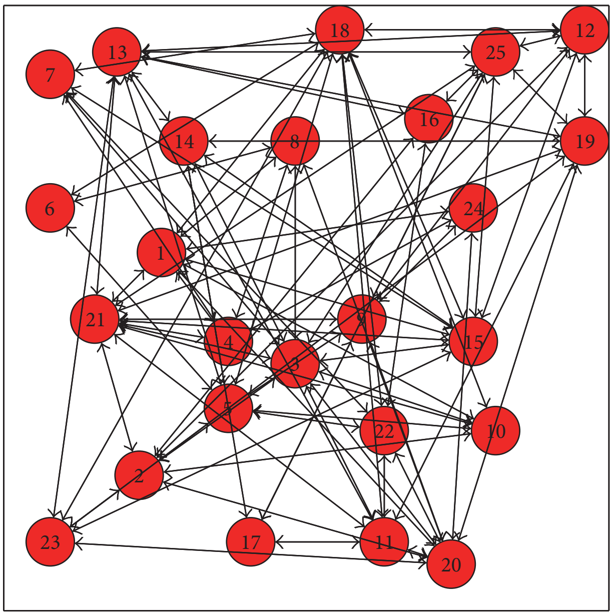

Figure 6: Network topology considered in Example 2.

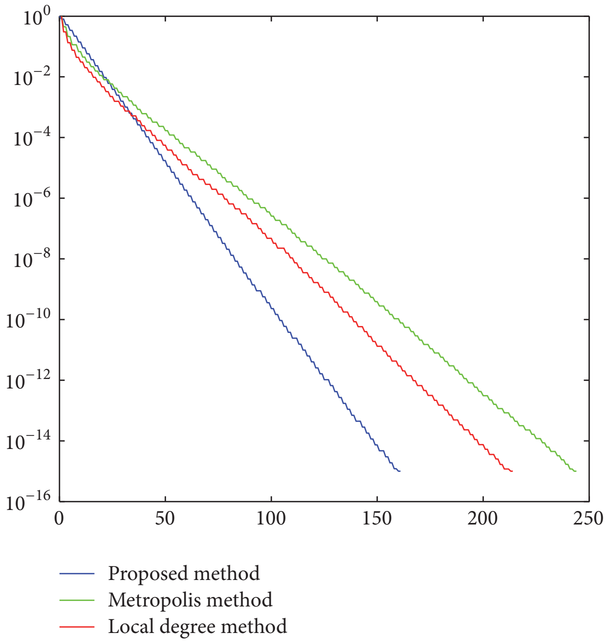

Figure 7: Pot of an error graph for Example 2.

additionally the update times are also varying and independent of each other, it becomes more challenging for agents to make consensus on a single value. These kinds of problems are really helpful to compare the performance of consensus methods. The numerical results of case 2 are shown in Table 2. The graphical comparison of consensus methods and the network topology are shown in Figures 7 and 6, respectively. The consensus graphs of proposed method, Metropolis method, and local degree method are shown in Figures 8, 9, and 10, respectively, which are the evidence of perfectness of proposed method.

Example 3 (addition of agents in reliable network topology at any time instant). This example is the continuity case of Example 1 and is called forced consensus problem. As in Example 1, some agents are selected randomly with fixed topology with asynchronous time updates, and finally they develop a consensus on a single value. Similarly in this example, some agents are randomly added in the network at any time instant and now they have to develop consensus on some new unique value. This means that at any time instant, the agents are forced from one value to another consensus value. In this example, 10 agents are added at some time interval; that is, $t = 50$ ms; the total number of agents becomes 40 and they have to develop consensus on value 0.025 with utilizing the asynchronous time updates; that is, it is fixed between 0.2 and 0.6. The numerical results of Example 3 are produced in Table 3. The network topology considered for this example is shown in Figure 11. In Figure 12, error is plotted for the proposed method, metropolis, and local degree method, respectively. Consensus graphs for this case are presented in Figures 13, 14, and 15.

Example 4 (addition of agents in unreliable network topology with asynchronous time update at any instant of time). This example is the second problem of forced consensus and is the continuity of case in Example 2. As in Example 2, some agents are selected randomly with dynamically changing interaction

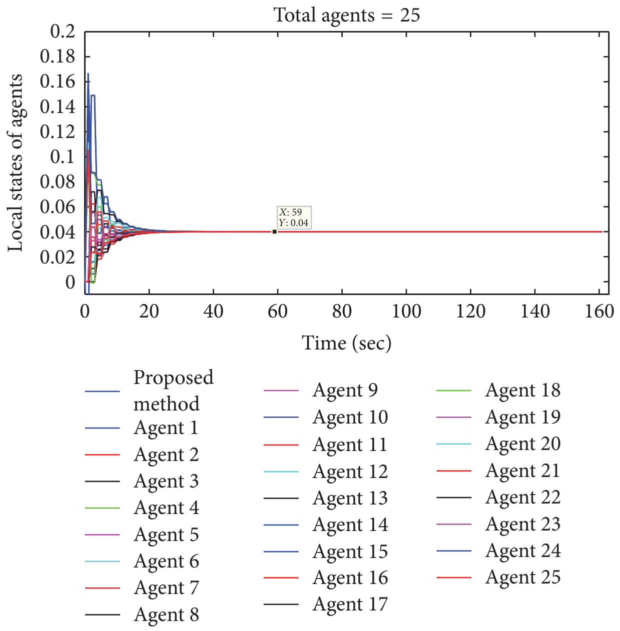

Figure 8: Agents approaching to a consensus value using proposed algorithm in Example 2.

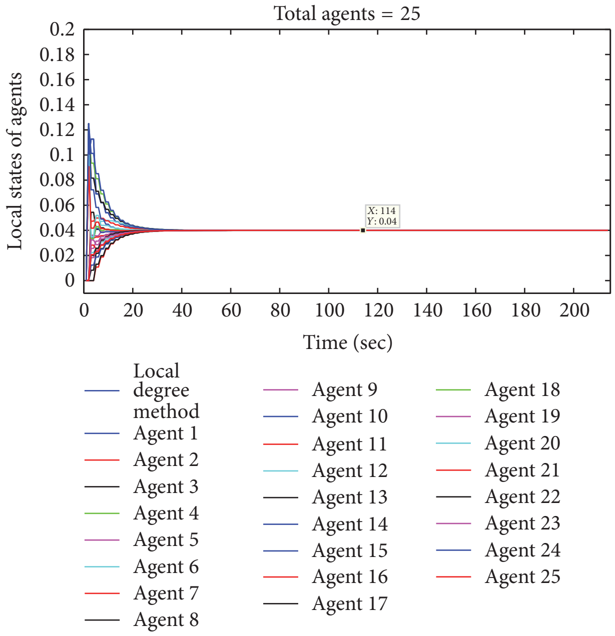

Figure 10: Agents approaching to a consensus value using local degree method in Example 2.

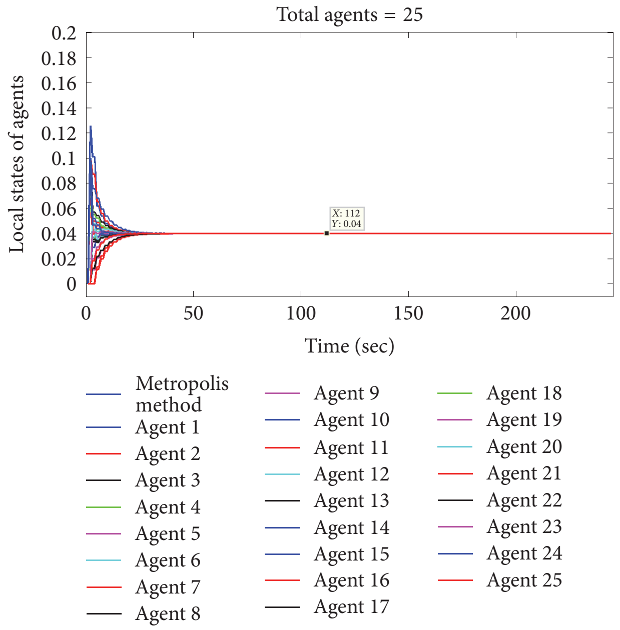

Figure 9: Agents approaching to a consensus value using Metropolis method in Example 2.

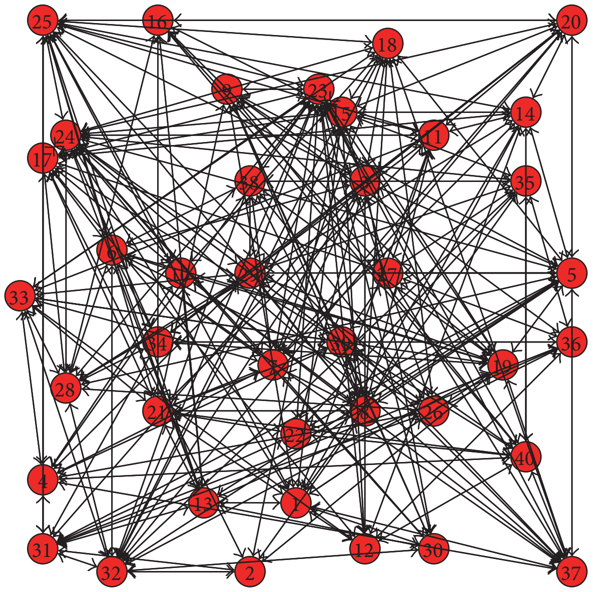

Figure 11: Communication topology considered in Example 3.

topology and asynchronous time updates; they developed consensus on a single value. In this example some agents are randomly added in the system at any time instant. Addition of agents disturbs the performance of consensus methods as the communication strategy is unreliable and changing randomly and also the time updates are asynchronous, so it becomes difficult for the agents to develop agreement on a single value but with proposed method it reaches to the consensus value in a less convergence time as compared to other techniques. The numerical results of Example 4 are shown in Table 4. The network topology considered for this example is given in Figure 16. In Figure 17 error is plotted for the proposed method, metropolis, and local degree method, respectively. Consensus graphs for this case are shown in Figures 18, 19, and 20. From the numerical and graphical analysis of different consensus techniques, it is observed that the proposed method converges faster to consensus value than other well-known consensus methods.

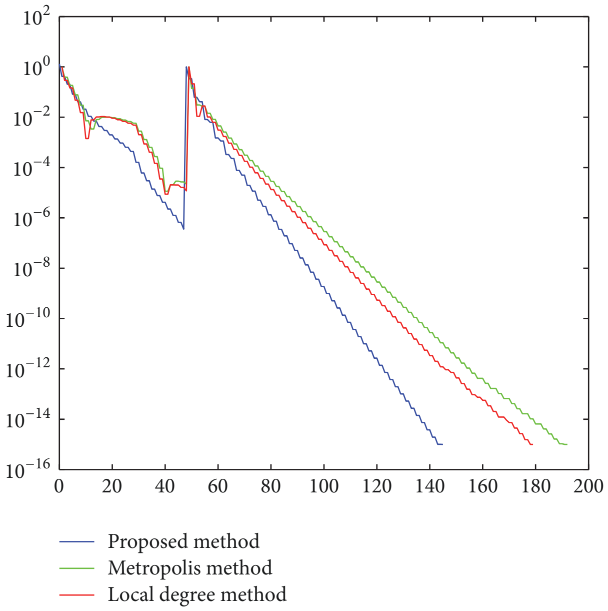

Figure 12: Plot of error graph generated for Example 3.

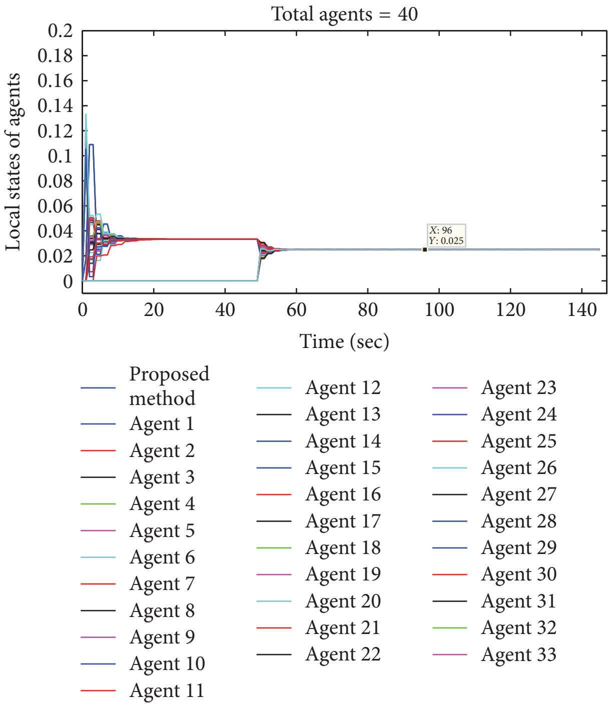

Figure 13: Agents approaching to a consensus value using proposed algorithm in Example 3.

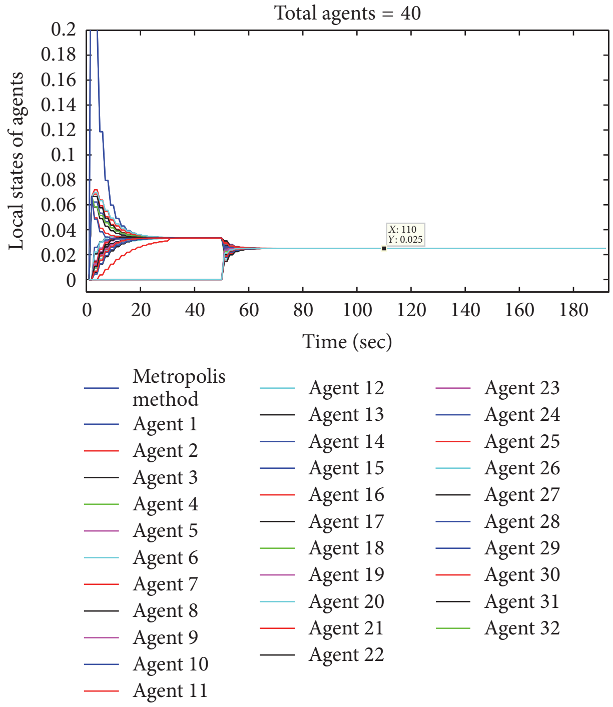

Figure 14: Agents approaching to a consensus value using Metropolis method in Example 3.

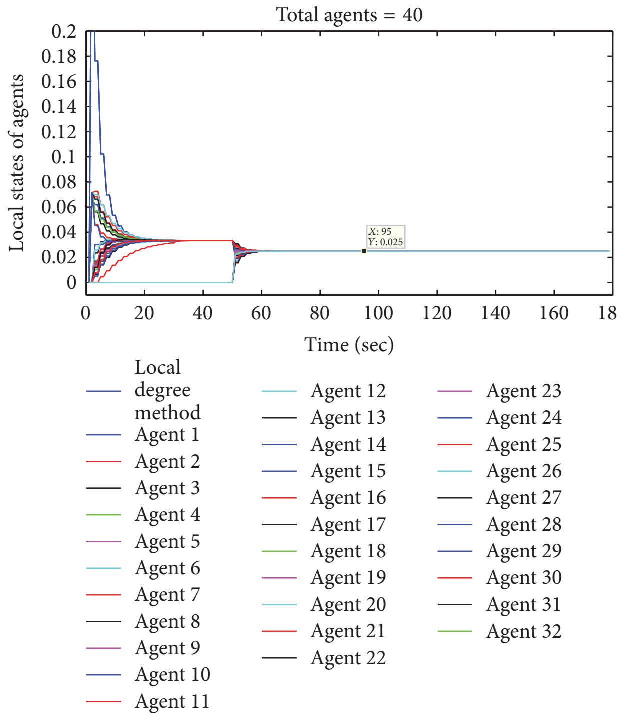

Figure 15: Agents approaching to a consensus value using local degree method in Example 3.

6. Conclusion

This paper proposed a new convergence algorithm for distributed multiagent systems. A unique approach of asynchronous time updates under reliable and unreliable network topology is implemented to produce better results in terms of different performance indicators, such as number of iterations and computational cost, and, capable of operability in large dynamic networks with unreliable communication, CPU processing time, asymptotic convergence factor, and

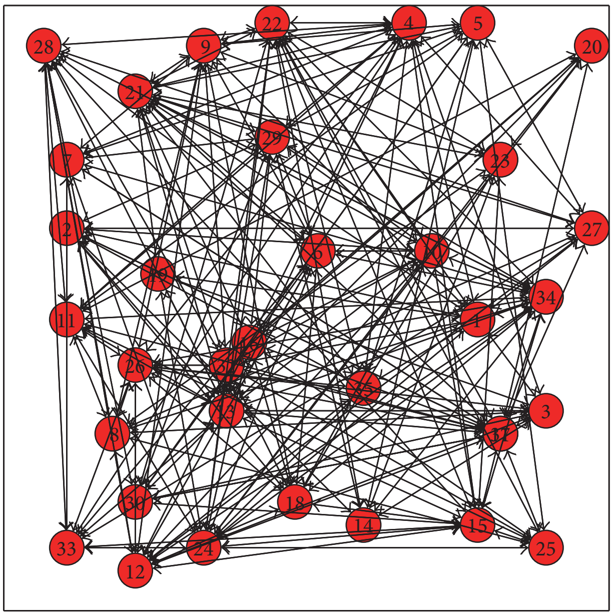

Figure 16: Network topology considered in Example 4.

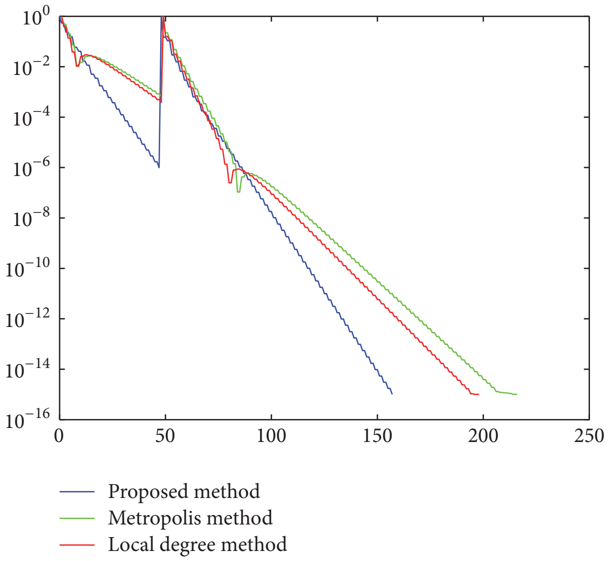

Figure 17: Plot of an error graph for Example 4.

Table 3: Comparison results of Example 3 for different parameters.

Methods	NI	CPU_{Min}	$\rho(W - 11^T/n)$	τ_{asym}
Proposed method	147	0.3515	0.7743	3.909
Metropolis method	193	0.4013	0.8216	5.089
Local degree method	180	0.3961	0.8070	4.663

Table 4: Comparison results of Example 4 for different parameters.

Methods	NI	CPU_{Min}	$\rho(W - 11^T/n)$	τ_{asym}
Proposed method	159	0.4571	0.7950	4.359
Metropolis method	217	0.4767	0.8494	6.1297
Local degree method	199	0.4654	0.8370	5.6231

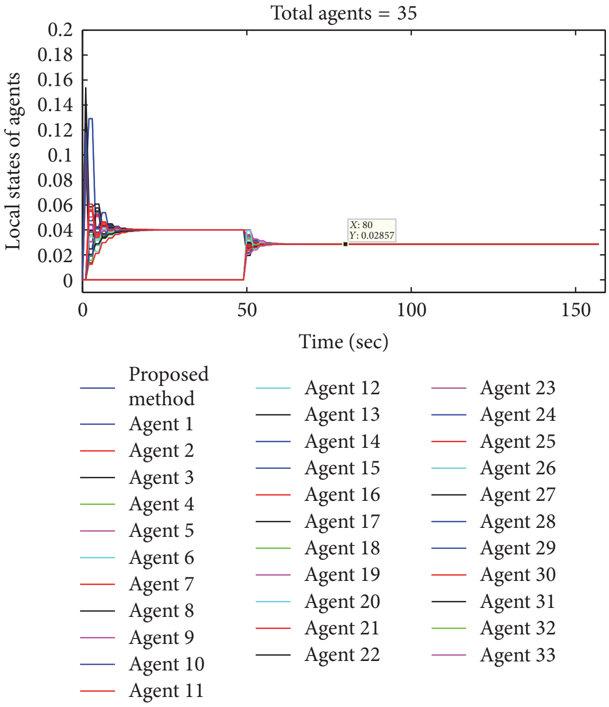

Figure 18: Agents approaching to a consensus value using proposed algorithm in Example 4.

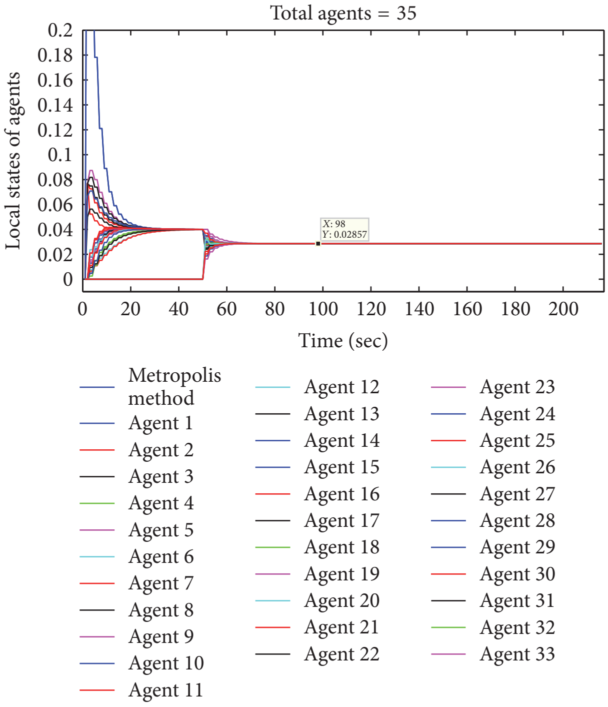

Figure 19: Agents approaching to a consensus value using Metropolis method in Example 4.

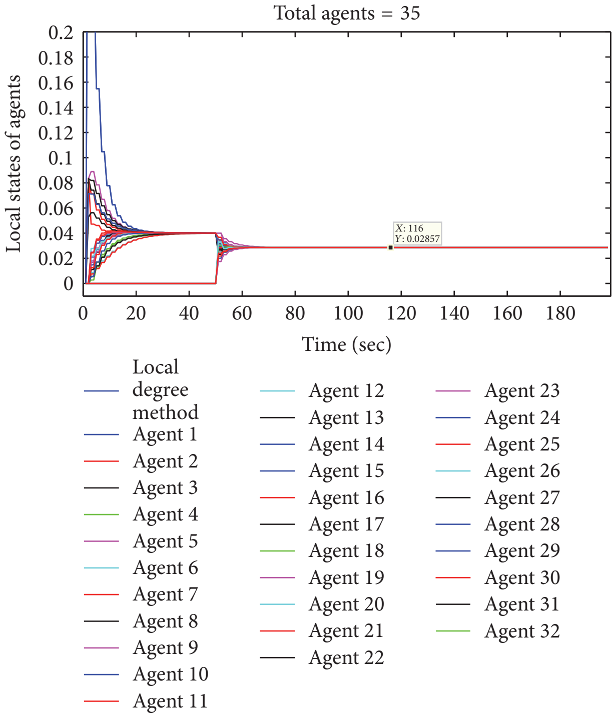

Figure 20: Agents approaching to a consensus value using local degree method in Example 4.

convergence time have been considered. A detailed comparison is produced with each example with different operational scenario, as an effective evidence to prove the efficiency of the proposed algorithm. In future we will extend our research for systems experiencing time varying delays under asynchronous communication to study their performance.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] Q. Zhang, P. Li, Z. Yang, and Z. Chen, “Distance constrained based adaptive flocking control for multiagent networks with time delay,” *Article ID*, Article ID 901282, 2014.

[2] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: algorithms and theory,” *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[3] J. Zhu, J. Lu, and X. Yu, “Flocking of multi-agent non-holonomic systems with proximity graphs,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 199–210, 2013.

[4] M. M. Zavlanos, A. Jadbabaie, and G. J. Pappas, “Flocking while preserving network connectivity,” in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 2919–2924, December 2007.

[5] M. Khan, G. Pandurangan, and V. S. A. Kumar, “Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 1, pp. 124–139, 2009.

[6] B. K. N. Trigoni and B. Krishnamachari, “Sensor network algorithms and applications,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 370, no. 1958, pp. 5–10, 2012.

[7] J. Dai, A. Benini, H. Lin, P. J. Antsaklis, M. J. Rutherford, and K. P. Valavanis, "Learning-based formal synthesis of cooperative multi-agent systems with an application to robotic coordination," in *Proceedings of the 24th Mediterranean Conference on Control and Automation, MED 2016*, pp. 1008–1013, Greece, June 2016.

[8] M. F. Ge, Z. H. Guan, B. Hu, D. X. He, and R. Q. Liao, "Distributed controller-estimator for target tracking of networked robotic systems under sampled interaction," *Automatica*, vol. 69, pp. 410–417, 2016.

[9] J. Barreiro-Gomez, I. Mas, C. Ocampo-Martinez, R. Sanchez-Pena, and N. Quijano, "Distributed formation control of multiple unmanned aerial vehicles over time-varying graphs using population games," in *Proceedings of the 55th IEEE Conference on Decision and Control, CDC 2016*, pp. 5245–5250, USA, December 2016.

[10] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 777–790, 2001.

[11] R. Toyot and T. Namerikawa, "Formation control of multi-agent system considering obstacle avoidance," in *Proceedings of the 2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 446–451, Kanazawa, September 2017.

[12] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *International Journal of Robotics Research*, vol. 33, no. 3, pp. 401–425, 2014.

[13] T. Han, Z. H. Guan, M. Chi, B. Hu, T. Li, and X. H. Zhang, "Multi-formation control of nonlinear leader-following multi-agent systems," *ISA Transactions®*, vol. 69, pp. 140–147, 2017.

[14] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[15] A. Kashyap, T. Başar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, 2007.

[16] X. Hu, H. Zhou, Z. Liu, X. Yu, and C. Li, "Hierarchical Distributed Scheme for Demand Estimation and Power Reallocation in a Future Power Grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2279–2290, 2017.

[17] C. Li, X. Yu, W. Yu, T. Huang, and Z.-W. Liu, "Distributed event-triggered scheme for economic dispatch in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1775–1785, 2016.

[18] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 14, pp. 733–737, 1963.

[19] V. Borkar and P. P. Varaiya, "Asymptotic agreement in distributed estimation," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 27, no. 3, pp. 650–655, 1982.

[20] C. W. Reynolds, "Flocks, herds, and schools: a distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[22] 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.

[23] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[24] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[25] L. Moreau, "Stability of continuous-time distributed consensus algorithms," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, pp. 3998–4003, December 2004.

[26] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 342–354, 2009.

[27] D. Yuan, D. W. C. Ho, and S. Xu, "Regularized Primal-Dual Subgradient Method for Distributed Constrained Optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2109–2118, 2016.

[28] W. Li, G. Wei, F. Han, and Y. Liu, "Weighted average consensus-based unscented Kalman filtering," *IEEE Transactions on Cybernetics*, 2015.

[29] Z. H. Guan, D. J. Hill, and X. Shen, "On hybrid impulsive and switching systems and application to nonlinear control," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 50, no. 7, pp. 1058–1062, 2005.

[30] Y. Wu, H. Su, P. Shi, Z. Shu, and Z. Wu, "Consensus of multi-agent systems using aperiodic sampled-data control," *IEEE Transactions on Cybernetics*, vol. 46, no. 9, pp. 2132–2143, 2016.

[31] G. S. Han, Z. H. Guan, J. Li, R. Q. Liao, and X. M. Cheng, "Multi-consensus of multi-agent networks via a rectangular impulsive approach," *Systems Control Letters*, vol. 76, pp. 28–34, 2015.

[32] W. Yu, L. Zhou, X. Yu, J. Lu, and R. Lu, "Consensus in multi-agent systems with second-order dynamics and sampled data," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2137–2146, 2013.

[33] M. T. Hale, A. Nedi, and M. Egerstedt, "Asynchronous multiagent primal-dual optimization," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 62, no. 9, pp. 4421–4435, 2017.

[34] Z. Liu, "Asynchronous impulsive consensus of multi-agent systems," in *Proceedings of the 2017 Eighth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 278–281, Hangzhou, November 2017.

[35] X. Wang, Z. Zeng, and Y. Cong, "Multi-agent distributed coordination control: developments and directions via graph viewpoint," *Neurocomputing*, vol. 199, pp. 204–218, 2016.

[36] V. K. Yanumula, I. Kar, and S. Majhi, "Consensus of second-order multi-agents with actuator saturation and asynchronous timedelays," *IET Control Theory Applications*, pp. 3201–3210, 2017.

[37] S. Bolognani, R. Carli, E. Lovisari, and S. Zampieri, "A randomized linear algorithm for clock synchronization in multi-agent systems," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 61, no. 7, pp. 1711–1726, 2016.

[38] Y. Feng, S. Xu, and B. Zhang, "Group consensus control for double-integrator dynamic multiagent systems with fixed

communication topology," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 532–547, 2014.

[39] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proceedings of the 42nd IEEE International Conference on Decision and Control*, pp. 4997–5002, Maui, HI, USA.

[40] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[41] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[42] F. Xiao and L. Wang, "State consensus for multi-agent systems with switching topologies and time-varying delays," *International Journal of Control*, vol. 79, no. 10, pp. 1277–1284, 2006.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
The Scientific World Journal
Journal of Probability and Statistics
International Journal of Mathematics and Mathematical Sciences
Journal of Optimization
International Journal of Engineering Mathematics
International Journal of Analysis
Journal of Complex Analysis
Advances in Numerical Analysis
Mathematical Problems in Engineering
International Journal of Differential Equations
Discrete Dynamics in Nature and Society
International Journal of Stochastic Analysis
Journal of Mathematics
Journal of Function Spaces
Abstract and Applied Analysis
Advances in Mathematical Physics