

Distributed Stopping Criterion for Consensus in the Presence of Delays

Mangal Prakash¹, Saurav Talukdar¹, Sandeep Attree¹, Vikas Yadav¹,
and Murti V. Salapaka¹, *Fellow, IEEE*

Abstract—Linear consensus protocol is an iterative distributed algorithm with asymptotic convergence guarantees. This paper develops and analyzes an algorithm for agents running linear consensus iterations to detect convergence to consensus within a specified error tolerance in a distributed manner. The distributed stopping criterion allows for time-varying bounded delays in information transmission and reception between agents. The algorithm relies on distributively determining the maximum and minimum values held by the agents. This paper further develops an algorithm for average consensus that utilizes a distributive stopping criterion, based on maximum and minimum consensus, where no centralized coordination is needed on how each agent weights its neighbor's values. Here, the doubly stochastic assumption on the weight matrix is relaxed and only column stochasticity is needed. The effectiveness of the algorithms is demonstrated by simulations and a comparison with prior work in the literature. Moreover, the demonstration of the proposed algorithms on an experimental test bed of Raspberry-Pi agents communicating wirelessly validates its applicability and utility.

Index Terms—Approximate consensus, average consensus with delays, consensus with delays, maximum consensus, minimum consensus.

I. INTRODUCTION

COORDINATION of multiagent systems to achieve system-level objectives requires sharing information among agents where the communication layer is determined by the network topology. A coordination strategy that has received considerable attention utilizes the consensus mechanism, where agents compute a common value of interest determined by initial values held by agents in a distributed manner [1], [2]. A special case of consensus is average consensus where agents converge

to the average of the initial conditions held by agents through local communication [2], [3]. Considerable research focuses on algorithms to distributively reach consensus/average consensus with contributions from communications, control, and computer science areas [4].

The convergence to the consensus/average consensus value, in the presence or absence of delays, when linear strategies are used, is achieved asymptotically [5]–[9]. The standard consensus protocol [2] converges asymptotically to a consensus value that may have a bias with respect to the average of the initial values held by the agents. Here, the bias can be removed if a doubly stochastic weight matrix (requires centralized weight matrix allocation) is employed and when no delays are present in the network. In [10] and [11], a ratio consensus algorithm is reported that only assumes a column-stochastic weight matrix (weights can be assigned in a distributed manner) and provides guarantees of the agents' iterations to asymptotically converge to the average of the initial values held by the agents even in the presence of time varying but bounded communication delays. In this approach, there is computational overhead over the standard consensus protocol [2] of propagating another added iteration.

In the approaches discussed previously, agents keep updating their states and communicating with their neighbors, in principle, forever, leading to wasted power and computational resources, which is untenable in resource-constrained applications, such as wireless-sensor networks. Moreover, in many real-time applications, including the power grid, the distributed stopping of consensus/average consensus iterations is essential as the consensus/average consensus value, when determined, is used in real time by local systems to perform important tasks [12], [13]. In such situations, it is necessary for each node to distributively detect if consensus/average consensus is achieved, possibly within a specified error margin and, thus, terminate computation as well as communication.

Distributed stopping of average consensus in the presence of time varying but bounded delays is presented in [11]. Charalambous *et al.* [11] rely on computing and storing Hankel matrices at every iteration, which can be computationally expensive if the size of the matrix is large; here, the size depends on the number of iterations needed for convergence. Manitaru and Hadjicostis [14] utilize an approach where nodes are allowed to broadcast only when necessary although without considering network delays. Here, nodes halt the consensus iterations based on the difference of their states with their in-neighbors where it is possible that not all nodes halt the computations simultaneously. The nodes wait for the neighbors to send new information and, thus,

Manuscript received November 25, 2018; revised December 2, 2018 and March 18, 2019; accepted May 2, 2019. Date of publication May 15, 2019; date of current version March 18, 2020. This work was supported by Advanced Research projects Agency-Energy (ARPA-E) Award DE-AR0000701. Recommended by Associate Editor L. Shi. (Mangal Prakash and Saurav Talukdar contributed equally to this work.) (Corresponding author: Murti V. Salapaka.)

M. Prakash, S. Attree, and M. V. Salapaka are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: praka027@umn.edu; attree002@umn.edu; murtis@umn.edu).

S. Talukdar is with the Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: taluk005@umn.edu).

V. Yadav is with Tesla, Inc., Santa Clara, CA 95051 USA (e-mail: vikasy@gmail.com).

Digital Object Identifier 10.1109/TCNS.2019.2917151

local communications are not terminated. As shown in the numerical illustrations in [14], this approach requires a large number of iterations to achieve termination of computations at all nodes. In many applications where coordinated action follows the distributed computation step, it is essential for every agent to realize that all agents have completed the consensus step.

In another approach, maximum and minimum consensus protocols that determine the maximum and minimum values held over the network of agents have proven effective for distributed decision making [15]. Numerous applications of maximum and minimum consensus strategies are reported in the literature, including [16], which uses maximum consensus for synchronization of wireless-sensor networks. Yadav and Salapaka [15] were the first to utilize the maximum and minimum consensus protocols to arrive at a distributed stopping criterion for termination of linear consensus iterations. It is also one of the earliest works on distributed stopping of linear consensus iterations. The maximum and minimum consensus protocols-based approach in [15] has a smaller computational footprint where all agents reach the stopping criterion simultaneously in comparison with [14]. Xie *et al.* [17] use minimum consensus for detecting consensus locally to characterize global consensus. Yadav and Salapaka [15] and Xie *et al.* [17] assume a row-stochastic weight matrix. Xie *et al.* [17] do not address convergence to the average of the initial values. With the more restrictive assumption of the weight matrix being doubly stochastic (that is, both, row and column stochastic), Yadav and Salapaka [15] guarantee that the consensus value reached is arbitrarily close to the average of the initial values held by the agents. We note that the implementation of a doubly stochastic weight matrix requires centralized coordination or weight balancing algorithms to disseminate the weight values to each agent prior to the initialization of the consensus iterations [18]. Moreover, the authors in [14], [15], and [17] do not address communication delays. We remark that addressing communication delays is essential given the prevalence of wireless communication and Internet of Things applications.

Our contributions are as follows.

- 1) Stopping criterion for consensus in the presence of delays: This paper presents an algorithm with a distributed stopping criterion for such linear consensus iterations in the presence of time varying but bounded communication delays. The stopping criterion can be set such that if the criterion is met by an agent, then it can access the consensus value within any preassigned error margin. The algorithm utilizes iterative computations of maximum and minimum values held by agents, which has a small computational footprint compared with existing approaches. Here, the maximum and minimum consensus-based distributed stopping approach in [15] is generalized to the case of time varying but bounded delays, for distributed termination of consensus iterations “close” to the consensus value.
- 2) Stopping criteria for average consensus (based on ratio consensus) in the presence of delays: Cady *et al.* [19] applied the maximum and minimum consensus presented in [15] to ratio consensus for resource allocation in ac microgrids. However, Cady *et al.* [19] do not provide provable guarantees and ignore communication delays. Here, we

establish the monotonicity of the maximum and minimum ratio of nodal states in the presence of time varying but bounded communication delays, which is then used to present an algorithm for distributed termination of ratio consensus iterations “close” to the average of the initial conditions. The algorithm can be set such that if the criterion is met by an agent, then it can access the average of the initial values held by the agents within any pre-assigned error margin. Prakash *et al.* [20] (conference paper) present a preliminary version of the work, where communication delays are not considered and proofs are not detailed.

- 3) The performance of the proposed algorithms is illustrated experimentally with a prototype network of Raspberry Pis, where agents experience time-varying delays on the communication channels. Compared with previous work, this is the only work that validates the distributed stopping criterion experimentally in the presence of stochastic communication delays and using a weight matrix realized distributively, thus providing evidence that our algorithm is indeed applicable practically.

The rest of this paper is organized as follows. In Section II, preliminaries and notations are presented. In Section III, the distributed stopping criterion for the standard consensus algorithm in the presence of delays based on maximum and minimum consensus is developed. Here, results are developed under the row-stochastic and doubly-stochastic assumptions on the weight matrix. In Section IV, a distributed algorithm where agents detect convergence to the average of the initial values held by the agents with any column-stochastic weight matrix is presented. The performance of the proposed distributed stopping criteria is illustrated through simulations and experiments in Section V. Finally, conclusions are presented in Section VI.

II. PRELIMINARIES AND NOTATION

In this section, definitions and notations needed for subsequent development (for details, see [21] and [22]) are provided. Consider the following definitions.

- 1) *Directed graph*: A directed graph G is a pair $\{V, E\}$, where V is a set of vertices or nodes and E is a set of edges, which are ordered subsets of two elements of V . If an edge from $j \in V$ to $i \in V$ exists, then it is denoted as $(i, j) \in E$.
- 2) *Directed path*: In a directed graph, a directed path from node j to node i exists if there is a sequence of distinct directed edges of G of the form $(i, k_1), (k_1, k_2), \dots, (k_m, j)$.
- 3) *Strongly connected graph*: A directed graph $G = \{V, E\}$ is strongly connected if for all pairs of vertices $i, j \in V$, there is a directed path from i to j and from j to i .
- 4) *In-neighbor of node*: Given a graph $G = \{V, E\}$, the set of in-neighbors of node $i \in V$ is denoted by $N_i^- := \{j \in V : (i, j) \in E\}$.
- 5) *Out-neighbor of node*: Given a graph $G = \{V, E\}$, the set of out-neighbors of node $i \in V$ is denoted by $N_i^+ := \{j \in V : (j, i) \in E\}$.

- 6) *Diameter of graph*: The longest shortest distance between any pair of nodes in a graph is the diameter of the graph and is denoted by D .
 - 7) *Stochastic matrices*: A real $n \times n$ matrix $A = [a_{ij}]$ is called a row-stochastic matrix if $1 \geq a_{ij} \geq 0$ for $1 \leq i, j \leq n$ and $\sum_{j=1}^n a_{ij} = 1$ for $1 \leq i \leq n$. A real $n \times n$ matrix $A = [a_{ij}]$ is called a column-stochastic matrix if A^T is a row-stochastic matrix. A real $n \times n$ matrix $A = [a_{ij}]$ is called a doubly stochastic matrix if it is both row stochastic and column stochastic.
 - 8) *Non-negative matrix*: A non-negative matrix is a matrix with all entries greater than or equal to zero.
 - 9) *Weight matrix P* : Consider a directed graph $G = \{V, E\}$ consisting of n agents. For every $(i, j) \in E$, a weight $p_{ij} > 0$ is associated, which represents the weight, and node i gives any information received from node j . If $(i, j) \notin E$, then $p_{ij} = 0$. $P = [p_{ij}]$ represents the weight matrix associated with graph G .
 - 10) *Delay $\tau_{ij}(k)$ at instant k* : The delay on the edge from node j to node i at any instant k is denoted by $\tau_{ij}(k)$.
- The following assumptions hold throughout this paper.
- A1. The delay in the network is uniformly bounded, that is, $\tau_{ij}(k) \leq \bar{\tau}$, $\bar{\tau} \in \mathbb{R}_{\geq 0}$, for all $i, j \in V$ and for all time instants k . In addition, $(i, i) \in E$ and $\tau_{ii}(k) = 0$ for all $i \in V$ and for all time instants k .
 - A2. Graph G is strongly connected.
- Agents (nodes) do not have the knowledge of the delay associated with the edges in their in-neighborhood.

III. DISTRIBUTED TERMINATION OF CONSENSUS

In this section, we consider a protocol where agent i updates its state $x_i(k+1)$ at $(k+1)$ th iteration according to

$$x_i(k+1) = p_{ii}x_i(k) + \sum_{j \in N_i^-} p_{ij}x_j(k - \tau_{ij}(k)) \quad (1)$$

where N_i^- is determined by the graph $G = \{V, E\}$. Let $x(k) := [x_1(k) \ x_2(k) \ \dots \ x_n(k)]^T$. Equation (1) implies that at any update iteration k , node i incorporates at most one packet of information from each of its in-neighbors. We assume that the following assumption holds throughout this section.

- A3. The weight matrix P associated with the graph G is row stochastic.

First, we state the following result from the prior work.

Theorem III.1: [6], [7] The following statements hold:

- 1) Update rule given by (1) under the assumptions A1 (delays with an uniform bound), A2 (graph is strongly connected), and A3 (weight matrix is row stochastic) achieves consensus, that is, for any set of initial conditions $x(0) \in \mathbb{R}^n$, there exists $\alpha \in \mathbb{R}$ such that $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for $i = 1, 2, \dots, n$.
- 2) Update rule given by (1) under the assumptions A2 (graph is strongly connected) and the weight matrix is doubly stochastic with $\tau_{ij}(k) = 0$ for all i, j and k (no delays) achieves average consensus, that is, for any set of initial conditions $x(0) \in \mathbb{R}^n$, there exists $\alpha \in \mathbb{R}$ such that $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for $i = 1, 2, \dots, n$, where $\alpha = \frac{1}{n} \sum_{i=1}^n x_i(0)$.

A. Maximum and Minimum Consensus Protocols

For node i and iteration k , define

$$M_i(k) := \max[x_j(k - r) | (j, r) \in (N_i^- \cup \{i\}) \times \{0, 1, \dots, \bar{\tau}\}] \quad (2)$$

$$m_i(k) := \min[x_j(k - r) | (j, r) \in (N_i^- \cup \{i\}) \times \{0, 1, \dots, \bar{\tau}\}]. \quad (3)$$

Here, $M_i(k)$ and $m_i(k)$ are the maximum and the minimum values held by any agent in the neighborhood of node i (including node i) at the k th and past $\bar{\tau}$ iterates from the k th iterate. Let

$$M(k) := \max_{j \in \{1, 2, \dots, n\}} M_j(k), \text{ and } m(k) := \min_{j \in \{1, 2, \dots, n\}} m_j(k). \quad (4)$$

Here, $M(k)$ and $m(k)$ are, respectively, the maximum and minimum taken over values held by any agent over the past $\bar{\tau}$ iterates, considering iterate k as present. Since $\tau_{ij}(k) \in \{0, 1, \dots, \bar{\tau}\}$, it follows that for $j \in N_i^- \cup \{i\}$, $x_j(k - \tau_{ij}(k)) \leq M_i(k) \leq M(k)$ and $x_j(k - \tau_{ij}(k)) \geq m_i(k) \geq m(k)$.

Lemma III.1: Consider the update rule (1). Then, for all $k' \geq k$ and $i = 1, \dots, n$

$$x_i(k') \leq M(k), \text{ and } x_i(k') \geq m(k). \quad (5)$$

Proof: See Appendix A for proof. ■

Lemma III.1 establishes that the value held by an agent in the future is always bounded above (below) by the maximum (minimum) over the current and delayed values over a horizon $\bar{\tau}$ of all the nodal states.

Lemma III.2: Consider a strongly connected graph $G = \{V, E\}$ running consensus protocol given by (1) with an initial condition $x(k)$. Let i be a node such that $x_i(k) < M(k)$ and let j be a node such that $x_j(k) > m(k)$, then for all $k' \geq k$, $x_i(k') < M(k)$, and $x_j(k') > m(k)$.

Proof: See Appendix A for proof. ■

Lemma III.2 establishes that if the value held by an agent i at the present instant of time is strictly less (greater) than the maximum (minimum) over the current and delayed values over a horizon $\bar{\tau}$ of all nodal states, then the value of agent i continues to be strictly less (greater) than this maximum (minimum) for all future instants.

Define $\max x(k) := \max_{i \in V} x_i(k)$ and $\min x(k) := \min_{i \in V} x_i(k)$, respectively. Note that for any $k \geq 0$, $\max x(k) \leq M(k)$ and $\min x(k) \geq m(k)$.

Lemma III.3: Consider a strongly connected graph $G = \{V, E\}$ with an update rule for the consensus protocol given by (1). If the initial condition $x(k)$ is such that $\min x(k) < \max x(k)$, then for all $k' \geq k + D(1 + \bar{\tau})$, $\max x(k') < M(k)$ and $\min x(k') > m(k)$.

Proof. See Appendix A for proof. ■

Lemma III.3 establishes that if the maximum value over all nodal states at the present instant is strictly greater than the minimum value over all nodal states at the present instant, then the maximum (minimum) over all nodal states after $D(1 + \bar{\tau})$ instants in the future will be strictly less (greater) than the present maximum (minimum) over the current and delayed values over a horizon $\bar{\tau}$ of all nodal states.

Define by $T^* := D(1 + \bar{\tau}) + \bar{\tau}$. Note that T^* is a constant for a fixed interconnection topology.

Theorem III.2: Consider a strongly connected graph $G = \{V, E\}$ with an update rule for the consensus protocol given by (1) and an initial condition $x(lT^*)$ such that $\min x(lT^*) < \max x(lT^*)$, where $l \geq 0$. Then, $M((l+1)T^*) < M(lT^*)$ and $m((l+1)T^*) > m(lT^*)$.

Proof: Using $k = lT^*$ in Lemma III.3, it follows that for $k' \geq lT^* + D(1 + \bar{\tau})$, $\max x(k') < M(lT^*)$. Note that

$$M((l+1)T^*) = \max_{v \in V} \max_{w \in N_v^- \cup \{v\}, r \in \{0, 1, \dots, \bar{\tau}\}} x_w((l+1)T^* - r)$$

where for $0 \leq r \leq \bar{\tau}$, $(l+1)T^* - r \geq (l+1)T^* - \bar{\tau} = lT^* + T^* - \bar{\tau} = lT^* + D(1 + \bar{\tau})$. It follows that $M((l+1)T^*) < M(lT^*)$. The rest of the proof is left to the reader. ■

Theorem III.2 implies that $M(lT^*)$ is a strictly decreasing sequence and $m(lT^*)$ is a strictly increasing sequence as a function of the index l .

Corollary III.1: Consider a strongly connected graph $G = \{V, E\}$ running consensus protocol given by (1) with an initial condition $x(lT^*)$ such that $\min x(lT^*) < \max x(lT^*)$, where $l \geq 0$. Then, the following conditions hold.

- $\max x((l+1)T^*) < M(lT^*)$ and $\min x((l+1)T^*) > m(lT^*)$.
- $\max x((l+1)T^*) - \min x((l+1)T^*) < M(lT^*) - m(lT^*)$.

Proof: The proof of (a) follows directly from Theorem III.2. The proof of (b) is a direct consequence of (a). ■

Corollary III.2: Consider the consensus protocol given by (1) with each node converging to α , that is, $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for all $i = 1, 2, \dots, n$. Then, the sequences $M(lT^*)$ and $m(lT^*)$ converge to α as $l \rightarrow \infty$ and, thus, the sequence $M(lT^*) - m(lT^*) \rightarrow 0$ as $l \rightarrow \infty$.

Proof: The proof follows from the fact that $M(lT^*)$ and $m(lT^*)$ are subsequences of convergent sequences $\{x_i(k)\}_{i=1,2,\dots,n}$.

Corollary III.1 and Corollary III.2 together imply that $\max x((l+1)T^*) - \min x((l+1)T^*) \rightarrow 0$ as $l \rightarrow \infty$. In what follows, an algorithm is devised which converges to $\max x((l+1)T^*)$ and $\min x((l+1)T^*)$ in a finite number of iterations. The distributed stopping criterion is based on the difference between $\max x((l+1)T^*)$ and $\min x((l+1)T^*)$ and evaluating whether the difference is less than the user-specified error tolerance. Toward this end, the maximum/minimum consensus protocols are developed next and the finite-time convergence of maximum/minimum consensus protocols in the presence of possible time-varying communication delays is proved.

Let $\tilde{T} := (\bar{\tau} + 1)$ and define the indicator function for the link from node $j \in V$ to node $m \in V$ at time k as

$$I_{k,mj}(\tau) = \begin{cases} 1, & \text{if } \tau_{mj}(k) = \tau \\ 0, & \text{if } \tau_{mj}(k) \neq \tau. \end{cases} \quad (6)$$

The maximum consensus protocol denoted by MXP computes the maximum of the given initial node conditions $z(0) = [z_1(0) \ z_2(0) \ \dots \ z_n(0)]^T$ in a distributed manner. It takes $z(0)$ as an input and generates a sequence of node values based on the

following update rule for node m for $k \geq 0$:

$$z_m(k\bar{\tau} + l) = z_m(k\bar{\tau} + l - 1), l \in \{k+1, \dots, k + \bar{\tau}\} \quad (7a)$$

$$z_m((k+1)\tilde{T}) = \max_{\substack{j \in N_m^- \cup \{m\} \\ r = \{0, 1, \dots, \bar{\tau}\} \\ q = \{0, 1, \dots, r\}}} \{z_j((k+1)\tilde{T} - (r+1))I_{(k+1)\tilde{T}-(r+1),mj}(q)\} \quad (7b)$$

where the indicator function is defined as in (6).

The minimum consensus protocol denoted by MNP computes the minimum of the given initial node conditions $y(0) = [y_1(0) \ y_2(0) \ \dots \ y_n(0)]^T$ in a distributed manner. It takes $y(0)$ as an input and generates a sequence of node values $y(k)$ based on the following update rule for $k \geq 0$:

$$y_m(k\bar{\tau} + l) = y_m(k\bar{\tau} + l - 1), l \in \{k+1, \dots, k + \bar{\tau}\} \quad (8a)$$

$$y_m((k+1)\tilde{T}) = \min_{\substack{j \in N_m^- \cup \{m\} \\ r = \{0, 1, \dots, \bar{\tau}\} \\ q = \{0, 1, \dots, r\}}} \{y_j((k+1)\tilde{T} - (r+1))I_{(k+1)\tilde{T}-(r+1),mj}(q)\} \quad (8b)$$

where the indicator function is defined as in (6).

Note that from (7a), (7b), (8a), and (8b), effectively every z_m update and every y_m update takes place once after every $\bar{\tau}$ iterations. In the time-varying delay framework, it is possible that any in-neighbor j of node m can send multiple packets of z_j to node m while it is waiting for $\bar{\tau}$ iterations, but each packet received from j during the waiting time contains the same information because node j also updates its maximum and minimum consensus states only once in $\bar{\tau}$ iterations in accordance with (7a), (7b), (8a), and (8b). The MNP is similar to the MXP since the minimum over a set of values is the negative of the maximum of the negative of the values. Next, results are established that will be useful to prove the convergence of MXP and MNP running through the update rules (7a), (7b), (8a), and (8b) in finite time. Let $z_{\max} := \max_{i \in V} z_i(0)$ and $y_{\min} := \min_{i \in V} y_i(0)$.

Lemma III.4: Consider a directed graph $G = \{V, E\}$ with possibly time-varying delays with bound $\bar{\tau}$, the MXP given by (7a) and (7b), and the MNP given by (8a) and (8b).

- For all $k' > 0$, $\max_{i \in V} z_i(k') = z_{\max}$ and $\min_{i \in V} y_i(k') = y_{\min}$.
- Let for some k , $z_j(k) = z_{\max}$ and $y_{j'}(k) = y_{\min}$, that is, nodes j and j' have the maximum and minimum values, respectively, in the network at the k th time instant. Then, for all instants $k' > k$, $z_j(k') = z_{\max}$ and $y_{j'}(k') = y_{\min}$, that is, node j continues to hold the maximum value and node j' continues to hold the minimum value for $k' > k$.

Proof: (a) The proof is left to the reader. (b) The proof follows from the fact that if the maximum value at the current iteration is held at node j , then node j continues to hold the maximum value in future iterations as well, as the update step (7b) of the MXP includes the past value of node j . The proof for the other part follows similarly.

Lemma III.5: Consider a directed graph $G = \{V, E\}$ with possibly time-varying delays with a uniform bound $\bar{\tau}$ and an

update rule for the MXP given by (7a) and (7b) and that for the MNP given by (8a) and (8b). Let $z_{\pi_1}(0) = \max_{j \in V} z_j(0)$ and $y_{\pi_1}(0) = \min_{j \in V} y_j(0)$. Then, for all $k \geq D(1 + \bar{\tau})$, and any $m \in V$, $z_m(k) = z_{\pi_1}(0)$, and $y_m(k) = y_{\pi_1}(0)$.

Proof: As $z_{\pi_1}(0) = \max_{j \in V} z_j(0)$, it follows from Lemma III.4 that:

$$z_{\pi_1}(k) = z_{\pi_1}(0) \text{ for all } k \geq 0.$$

Consider any node $i \in V$. Since the graph G is strongly connected, there exists a directed path $(\pi_2, \pi_1)(\pi_3, \pi_2) \dots (i, \pi_d)$ connecting i and π_1 . It follows from the update rule (7a) and (7b) that within $\bar{\tau}$ iterations, π_2 will have received the value $z_{\pi_1}(0)$ and, thus, $z_{\pi_2}(\bar{\tau} + 1) = z_{\pi_1}(0)$; and for any $k \geq \bar{\tau} + 1$, $z_{\pi_2}(k) = z_{\pi_1}(0)$. Using the above-mentioned steps for $\pi_3, \pi_4, \dots, \pi_d, i$, it follows that:

$$z_i(k) = z_{\pi_1}(0) \text{ for any } k \geq d(\bar{\tau} + 1).$$

Thus, if $k \geq D(\bar{\tau} + 1) \geq d(\bar{\tau} + 1)$; $z_i(k) = z_{\pi_1}(0)$. Since D is the diameter of the graph G , it follows that for $k \geq D(\bar{\tau} + 1)$:

$$z_m(k) = z_{\pi_1}(0) = \max_{j \in V} z_j(0) \text{ for all } m \in V.$$

The proof of the other equality is similar. ■

B. Distributed Algorithm for Stopping Consensus Protocol

In this section, an algorithm based on maximum–minimum consensus for stopping the consensus protocol distributively in finite time is synthesized using the results derived in the previous subsections. At time instants $T(j) := jT^*$, with $T^* := D(1 + \bar{\tau}) + \bar{\tau}$ and $j = 0, 1, 2, \dots$, MXP and MNP protocols are reset with initial conditions $x(jT^*)$. Thus, $z(T(j)) = x(T(j))$ and $y(T(j)) = x(T(j))$. Let

$$\bar{\alpha}(j) := \max z(T(j)), \underline{\alpha}(j) := \min y(T(j)), \text{ and} \\ \beta(j) := \bar{\alpha}(j) - \underline{\alpha}(j).$$

Lemma III.6: Consider the consensus protocol given by (1) with each node converging to α , that is, $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for all $i = 1, 2, \dots, n$. Then, the sequences $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ converge to α as $j \rightarrow \infty$. Furthermore, the sequence $\beta(j) \rightarrow 0$ as $j \rightarrow \infty$.

Proof: If $\{x(k)\}_{k=1}^{\infty}$ converges to α , that is, $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for all $i = 1, 2, \dots, n$, then it follows that $\lim_{k \rightarrow \infty} \max x_i(k) = \alpha$ and $\lim_{k \rightarrow \infty} \min x_i(k) = \alpha$ for all $i = 1, 2, \dots, n$.

Note that $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ are subsequences of convergent sequences $\max x_i(k)$ and $\min x_i(k)$, respectively. Thus, $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ converge to the same limit α . This implies that $\beta(j) \rightarrow 0$ as $j \rightarrow \infty$. ■

It should be noted that Lemma III.6 is a consequence of Corollary III.2.

Theorem III.3: The following results hold.

- 1) Under assumptions A1 (uniformly bounded delays), A2 (graph is strongly connected), and A3 (weight matrix is row-stochastic), given any $\rho > 0$, there exists $T_c \in \mathbb{N}$ such that $\beta(j) < \rho$, that is, Algorithm 1 terminates at line number 15 in a finite number of iterations.
- 2) Under assumptions of no delays in communication, A2 (graph is strongly connected) and with the additional

Algorithm 1: Distributed Stopping of Consensus in the Presence of Uniformly Bounded Delays.

```

1 Input:
2 |  $x(0), D, \bar{\tau}, \rho, P$ ; // Initial condition
3 Initialize:
4 |  $k := 0; l := 1; z_i := x_i(0); y_i := x_i(0); \theta := 1;$ 
   |  $\psi := 0;$ 
5 Repeat:
6 |  $x_i(k+1) := p_{ii}x_i(k) + \sum_{j \in N_i^-} p_{ij}x_j(k - \tau_{ij}(k))$ 
7 | if  $k+1 = \psi + l(1 + \bar{\tau})$  then
   |   /* maximum and minimum consensus updates
   |   given by (7b) and (8b) for each node
   |    $i \in V$  */
8 |    $z_i := \max_{j \in N_i^- \cup \{i\}} z_j;$ 
9 |    $y_i := \min_{j \in N_i^- \cup \{i\}} y_j;$ 
10 |   $l := l + 1$ 
11 end
12 emit:  $x_i(k+1), y_i$  and  $z_i$ 
13 if  $k+1 = \theta T^*$  then
14 |   if  $z_i - y_i < \rho$  then
15 |     break; // stop  $x_i, y_i$  and  $z_i$  updates
16 |   else
17 |      $z_i := x_i(\theta T^*); y_i := x_i(\theta T^*); \theta := \theta + 1;$ 
18 |      $l := 1;$  // Reset
19 |      $\psi := \psi + k$ 
20 |   end
21  $k := k + 1$ 

```

assumption that the weight matrix is doubly stochastic, given any $\rho > 0$, there exists $T_c \in \mathbb{N}$ such that $\beta(j) < \rho$, that is, Algorithm 1 terminates at line number 15 in a finite number of iterations, where each agent has access to the average of the initial values held by all agents within ρ .

Proof: 1) Under assumptions A1–A3, we have from Theorem III.2 that $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for all i . It follows from Lemma III.6 that $\beta(j) \rightarrow 0$ as $j \rightarrow \infty$. Thus, for any given $\rho > 0$, there exists an integer j_0 such that $\beta(j) < \rho$ for all $j \geq j_0$. This implies that Algorithm 1 converges in finite time $T_c = j_0(\rho)$. ■

2) The proof is provided in [15].

Thus, we first established the monotonicity of maximum and minimum consensus iterations and used it to develop Algorithm 1 for distributed termination of consensus iterations at the nodes, once the nodes are close to the consensus value. The analysis utilized the row-stochastic property of the weight matrix. Moreover, we have established that a consensus value reaches within a specified tolerance of the average of the initial values held by the agents in the delay-free case with a distributed stopping criterion for a doubly stochastic weight matrix. We remark that to the best of the authors' knowledge, even with a doubly stochastic weight matrix assumption in the presence of delays, the linear consensus algorithm does not necessarily converge to the average of the agents' initial values (please refer to Fig 2(a), which

is a simulation-based illustration of this point). To overcome the bias of the consensus value, we resort to ratio consensus iterations and distributed termination of ratio consensus iterations in the following section.

IV. DISTRIBUTED TERMINATION OF AVERAGE CONSENSUS

In Section III, under the row-stochastic weight matrix, finite-time termination of the consensus iteration described by (1) can be obtained where all agents arrive at a consensus value. However, the consensus value is not guaranteed to be the average of the initial values. Such a guarantee is possible only if column stochasticity holds for the weight matrix. Here, we only assume column stochasticity for the weight matrix and develop an algorithm to reach the average of the initial conditions in a distributed manner and in a finite number of iterations. As alluded to earlier, a column-stochastic weight matrix can be chosen in a distributed manner unlike a doubly stochastic weight matrix [10]. We remark that the techniques used in this section do not subsume the development of Section III where the row-stochastic assumption needs a different approach.

First, we review the framework of ratio consensus for arriving at the average of initial conditions using a column-stochastic weight matrix in the presence of time varying but bounded delays. In [10], it is shown that average consensus can be *asymptotically* reached by using the ratio of two consensus updates described as follows:

$$x_i(k+1) = \sum_{j \in N_i^- \cup \{i\}} p_{ij} \sum_{r=0}^{\bar{\tau}} x_j(k-r) I_{k-r,ij}(r) \quad (9)$$

$$u_i(k+1) = \sum_{j \in N_i^- \cup \{i\}} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k-r) I_{k-r,ij}(r). \quad (10)$$

The delay between the neighbors is time varying and the result from [10] is formally stated ahead.

Theorem IV.1: [10] Suppose the weight matrix P with $P(i, j) = p_{ij}$ associated with the directed graph G is primitive and column stochastic with $P(i, i) > 0$ for all $i \in V$. Let the initial conditions for the numerator states be $\{x_1(0), x_2(0), \dots, x_n(0)\}$ and for the denominator states be $\{u_i(0) = 1\}_{i \in V}$. Under assumptions A1 and A2, the ratio $\frac{x_i(k)}{u_i(k)}$ asymptotically converges to $\frac{\sum_{i=1}^n x_i(0)}{n}$ for all $i = 1, \dots, n$, that is, the ratio of the numerator and denominator states converge to the average of the initial conditions (referred to as ratio consensus).

The key point to note is that the weight matrix is required to be column stochastic. A column-stochastic weight matrix can be determined in a distributed manner; for example, $p_{ij} = 1/(\text{card}(N_j^+) + 1)$, where N_j^+ is the set of out-neighbors of node j and card denotes cardinality. Next, we present a distributed algorithm for termination of ratio consensus iterations near the average of initial conditions.

A. Distributed Termination Algorithm for Ratio Consensus

First, we define new maximum and minimum consensus iterations for ratio consensus. Then, we will prove monotonic

properties of the new maximum and minimum consensus iterations, similar to those of the previous section. Let us consider the maximum and minimum values of the ratio of nodal states [see (9) and (10)] over all nodes within a time horizon $\bar{\tau}$ from any time instant k given by

$$M(k) := \max_{j \in V} \frac{x_j(k-r)}{u_j(k-r)}, u_j(k-r) \neq 0, j \in V \quad (11)$$

$$r = \{0, 1, 2, \dots, \bar{\tau}\}$$

$$m(k) := \min_{j \in V} \frac{x_j(k-r)}{u_j(k-r)}, u_j(k-r) \neq 0, j \in V. \quad (12)$$

$$r = \{0, 1, 2, \dots, \bar{\tau}\}$$

Remark 1: Given that $u_j(0) = 1$ for all $j \in V$ and P is a non-negative matrix, $u_j(k-r) \neq 0$ for all $k \in \mathbb{N}$ and $r \in \{0, 1, 2, \dots, \bar{\tau}\}$.

Lemma IV.1: Consider the update rule (9) and (10) with the assumptions of Theorem IV.1. Then, for all $k' \geq k$

$$\frac{x_i(k')}{u_i(k')} \leq M(k), \text{ for all } i \in V, \text{ and} \quad (13a)$$

$$\frac{x_i(k')}{u_i(k')} \geq m(k), \text{ for all } i \in V. \quad (13b)$$

Proof: See Appendix B for proof. ■

Lemma IV.2: Consider the update rule (9) and (10) with the assumptions of Theorem IV.1. Let i be a node such that $\frac{x_i(k')}{u_i(k')} < M(k)$ and let j be a node such that $\frac{x_j(k')}{u_j(k')} > m(k)$ for some $k' \geq k$. Then, for all $k'' \geq k'$, $\frac{x_i(k'')}{u_i(k'')} < M(k)$, $\frac{x_j(k'')}{u_j(k'')} > m(k)$.

Proof: See Appendix B for proof. ■

Remark 2: Note that $\frac{x_i(k')}{u_i(k')} \leq M(k)$, for $k' \geq k$ follows from Lemma 4.1. Lemma 4.2 emphasizes the strict inequality $\frac{x_i(k'')}{u_i(k'')} < M(k)$ for $k'' > k'$ if at any time instant k' , the inequality $\frac{x_i(k')}{u_i(k')} < M(k)$ is satisfied. A similar remark holds for the min value as well.

Lemma IV.3: Consider the update rule (9) and (10) with the assumptions of Theorem IV.1. Let $R(k) := \{\frac{x_1(k)}{u_1(k)}, \dots, \frac{x_n(k)}{u_n(k)}\}$ such that $\min R(k) < \max R(k)$. Then, for all $k' \geq k + D(1 + \bar{\tau})$ and for all $v \in V$

$$\max R(k') < M(k), \text{ and} \quad (14a)$$

$$\min R(k') > m(k). \quad (14b)$$

Proof: See Appendix B for proof. ■

As in Section III, define by $T^* := D(1 + \bar{\tau}) + \bar{\tau}$. The following theorem shows that $M((l+1)T^*)$ is a strictly decreasing sequence as a function of index l . Similarly, $m((l+1)T^*)$ under is a strictly increasing sequence as a function of index l .

Theorem IV.2: Consider the update rule (9) and (10) with the assumptions of Theorem IV.1 and the initial ratio vector as $R(lT^*) := \{\frac{x_1(lT^*)}{u_1(lT^*)}, \dots, \frac{x_n(lT^*)}{u_n(lT^*)}\}$ such that $\min R(lT^*) < \max R(lT^*)$, where $l = 0, 1, 2, \dots$. Then, $M((l+1)T^*) < M(lT^*)$ and $m((l+1)T^*) > m(lT^*)$.

Proof: Using $k = lT^*$ in Lemma 4.3, it follows that for all $j \in V$ and for $r \in \{0, 1, 2, \dots, \bar{\tau}\}$, $\frac{x_j(lT^* + D(1+\bar{\tau}) + r)}{u_j(lT^* + D(1+\bar{\tau}) + r)} < M(lT^*)$. Thus, $\max_{j \in V, r \in \{0, \dots, \bar{\tau}\}} \frac{x_j((l+1)T^* - r)}{u_j((l+1)T^* - r)} < M(lT^*)$, implying, $M((l+1)T^*) < M(lT^*)$. This completes the proof of the first inequality. The other inequality can be proved similarly and is left to the reader. ■

Corollary IV.1: Consider the update rule (9) and (10) with the assumptions of Theorem IV.1. Then, $\lim_{l \rightarrow \infty} M(lT^*) = \frac{\sum_{j=1}^n x_j(0)}{n}$ and $\lim_{l \rightarrow \infty} m(lT^*) = \frac{\sum_{j=1}^n x_j(0)}{n}$. Moreover, $\lim_{l \rightarrow \infty} M(lT^*) - \lim_{l \rightarrow \infty} m(lT^*) = 0$.

Proof: The proof follows from Theorem IV.2 and reasoning, as mentioned in Corollary III.2, and is left to the reader. ■

Next, using maximum and minimum consensus definitions (2), (3) and the results established in Section III-A, we propose an algorithm for stopping the ratio consensus protocol in finite time based on a user-specified threshold ρ . The maximum and minimum consensus iterations are to be reset with the ratio of the nodal states every T^* iterations, that is, $z(T(j)) = \{\frac{x_1(T(j))}{u_1(T(j))}, \dots, \frac{x_n(T(j))}{u_n(T(j))}\}$, and $y(T(j)) = \{\frac{x_1(T(j))}{u_1(T(j))}, \dots, \frac{x_n(T(j))}{u_n(T(j))}\}$. Define $\bar{\alpha}(j)$, $\underline{\alpha}(j)$, and $\beta(j)$ as before. Then, the following result analogous to Lemma III.6 holds and the proof is omitted due to similarity with the proof of Lemma III.6 and space limitations.

Lemma IV.4: Consider the ratio consensus protocol given by (9) and (10) with each node converging to $\alpha := \frac{\sum_{j=1}^n x_i(0)}{n}$, that is, $\lim_{k \rightarrow \infty} x_i(k) = \alpha$ for all $i = 1, 2, \dots, n$. Then, the sequences $\bar{\alpha}(j)$ and $\underline{\alpha}(j)$ converge to α as $j \rightarrow \infty$. Furthermore, the sequence $\beta(j) \rightarrow 0$ as $j \rightarrow \infty$.

This leads to the algorithm for distributed finite-time termination of averaging using ratio consensus and the main result of this section. The proof is similar to the proof of Theorem III.3 and is skipped due to space limitations.

Theorem IV.3: Given any $\rho > 0$, there exists $T_c \in \mathbb{N}$ such that $\beta(j) < \rho$, that is, Algorithm 2 terminates at line number 15 in a finite number of iterations where each agent has access to the average of the initial values held by all agents within ρ .

Thus, we first established monotonicity of the maximum and minimum consensus for the ratio of the nodal states and used it to design an algorithm for distributed termination of ratio consensus near the average. We end this section with the following remark.

Remark 3: It is shown in [23] that no finite-time distributed averaging protocol is possible without knowledge of some global network parameter if each node only knows its history of estimates. The only global parameters needed for our algorithm are the knowledge of the graph diameter and an upper bound on the maximum delay in the network. However, it should be noted that each node does not need to know the actual diameter but some upper bound of the diameter. One of the many algorithms in the literature that can accomplish the task of finding diameters is shown in [24]. Each node can first run this protocol to determine the diameter in a distributed manner.

Remark 4: Note that the maximum and minimum consensus in Algorithms 1 and 2 could begin at a suitable multiple of $D(1 + \bar{\tau})$ instead of $k = 0$ if the network owner knows that convergence would not occur before a certain number of iterations. This step will reduce communication overhead in larger networks.

Algorithm 2: Finite-time Termination of average consensus in Presence of Uniformly Bounded Delays.

```

1 Input:
2    $x(0), u(0) = 1$  ; // Initial condition
3    $D, \bar{\tau}, \rho, P$ ;
4 Initialize:
5    $k := 0; z_i(0) := x_i(0)/u_i(0); y_i(0) := x_i(0)/u_i(0);$ 
6    $\theta := 1; l := 1; \psi = 0$ ;
7 Repeat:
8    $x_i(k+1) :=$ 
9      $p_{ii}x_i(k) + \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} x_j(k-r)I_{k-r,ij}(r);$ 
10   $u_i(k+1) :=$ 
11     $p_{ii}u_i(k) + \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k-r)I_{k-r,ij}(r);$ 
12  if  $k+1 = \psi + l(1+\bar{\tau})$  then
13    /* maximum and minimum consensus updates
14     given by (7b) and (8b) for each node
15      $i \in V$  */
16     $z_i(k+1) :=$ 
17       $\max_{j \in N_i^- \cup \{i\}} \{z_j(k-r)I_{k-r,ij}(r)\}_{r=0,1,\dots,\bar{\tau}};$ 
18     $y_i(k+1) :=$ 
19       $\min_{j \in N_i^- \cup \{i\}} \{y_j(k-r)I_{k-r,ij}(r)\}_{r=0,1,\dots,\bar{\tau}};$ 
20     $l := l + 1$ 
21  end
22  if  $k+1 = \theta((D+1)(1+\bar{\tau}))$  then
23     $\bar{\alpha}_i := z_i(k+1);$ 
24     $\underline{\alpha}_i := y_i(k+1);$ 
25     $\beta_i := \bar{\alpha}_i - \underline{\alpha}_i;$ 
26    if  $\beta_i < \rho$  then
27      break ; // stop  $x_i, y_i$  and  $z_i$  updates
28    else
29      set  $z_i(k+1) := x_i(k+1)/u_i(k+1);$ 
30       $y_i(k+1) := x_i(k+1)/u_i(k+1);$ 
31       $\theta = \theta + 1;$ 
32       $\psi = \psi + k;$ 
33       $l := 1$  ; // Reset
34    end
35  end
36  broadcast:  $x_i(k+1), u_i(k+1), z_i(k+1), y_i(k+1)$ 
37   $k := k + 1$ 

```

V. EXPERIMENTAL AND SIMULATION-BASED VALIDATION

A. Simulation-Based Validation of Algorithm 1

First, we consider random (time varying) but bounded communication delays on the network shown in Fig. 1(a). All of the delays are sampled from a uniform distribution on $\{0, 1, 2, 3\}$ (where $\bar{\tau} = 3$) for the simulation. The initial conditions are set as $x(0) = [1000 \ 0 \ 200 \ 100 \ 700]^T$ with $\rho = 0.01$. The communication weight matrix P is chosen to be

$$P = \begin{bmatrix} 1/2 & 0 & 0 & 1/2 & 0 \\ 1/4 & 3/4 & 0 & 0 & 0 \\ 1/4 & 1/8 & 1/2 & 0 & 1/8 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 1/8 & 1/2 & 0 & 3/8 \end{bmatrix}.$$

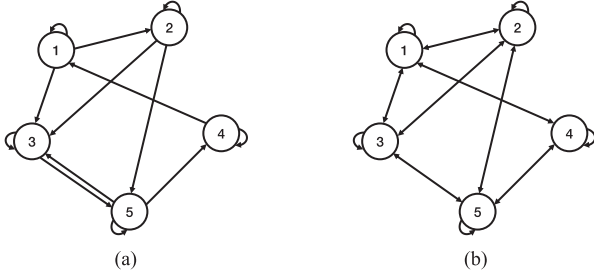


Fig. 1. (a) General representative network of five nodes for the simulation study. (b) Five-node network for experimental validation.

Under Algorithm 1, all nodes converge to a common value of 320.2623 [see Fig. 2(a)] in 96 iterations. We note that the consensus value in the presence of time-varying delays depends on the specific realization of the communication delays and is not necessarily the average of the initial conditions.

B. Simulation-Based Comparison of Algorithm 2 and [15]

Next, we compare Algorithm 2 with the algorithm in [15] for the network in Fig. 1(a) with the same communication weight matrix P as in the above-mentioned study and fixed (deterministic) communication delays, with the network delay matrix τ being

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 3 & 1 & 0 & 0 \end{bmatrix}.$$

The initial values of the nodes are set as $x(0) = [1000 \ 0 \ 200 \ 100 \ 700]^T$ and the error tolerance is set as $\rho = 0.1$. Fig. 2(b) illustrates the performance of the stopping algorithm in [15] with Algorithm 2. Here, it is seen that the algorithm in [15] fails to terminate the consensus iterations near the average of the initial conditions (stopping criterion is not satisfied) even in the presence of fixed delays (given by τ); unlike Algorithm 2.

C. Experimental Case Study

Here, we realize a five-node network shown in Fig. 1(b) where all of the edges are bidirectional. The initial values held by agents in this network are set as $x(0) = [1000 \ 0 \ 200 \ 100 \ 700]^T$ with the error tolerance of $\rho = 0.01$. The column-stochastic weight matrix is given by

$$P = \begin{bmatrix} 1/4 & 1/5 & 1/4 & 1/3 & 0 \\ 1/4 & 1/5 & 1/4 & 0 & 1/4 \\ 1/4 & 1/5 & 1/4 & 0 & 1/4 \\ 1/4 & 1/5 & 0 & 1/3 & 1/4 \\ 0 & 1/5 & 1/4 & 1/3 & 1/4 \end{bmatrix}.$$

Each agent in the network is realized using a Raspberry Pi 3 model b [25] with a configuration of 1.2-GHz CPU, 1-GB RAM, and 802.11n Wireless LAN support [26]. A NodeJS-based application [27], capable of bidirectional communication for running the consensus algorithm on each Raspberry Pi is developed. Data communications occur via an HTTP protocol [28], which is designed over TCP/IP [29] and ensures guaranteed, in-order delivery of data packets between the nodes using WiFi.

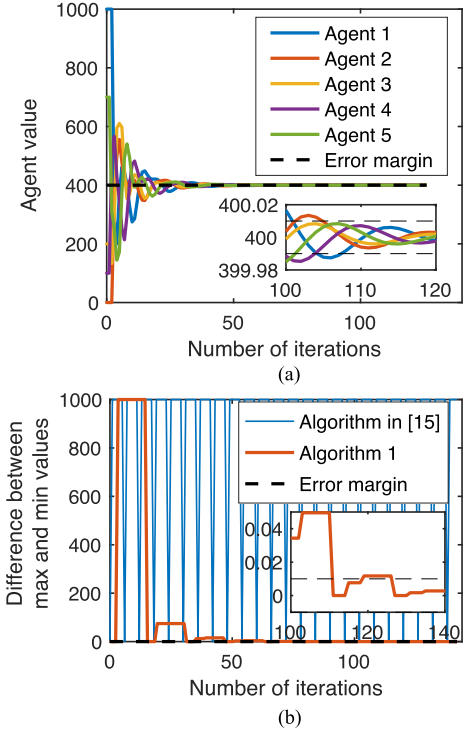


Fig. 2. Simulation results on the general representative network of five nodes, as shown in Fig. 1(a). (a) Application of Algorithm 1 for distributed stopping of consensus in the presence of uniformly bounded time-varying delays. (b) Comparison of Algorithm 2 with the algorithm presented in [15] in the presence of fixed delays in terms of the difference between maximum and minimum consensus values at a node.

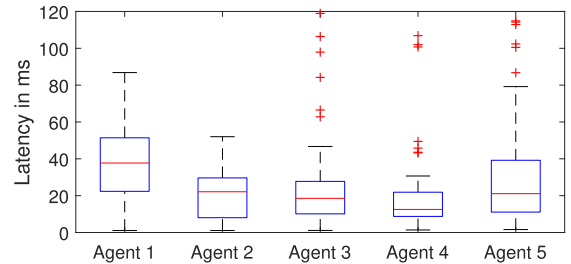


Fig. 3. Distribution of pairwise latency (in milliseconds) for each node in the network shown in Fig. 1(b).

The distribution of pairwise communication latency for each node in the experimental realization of the network given in Fig. 1(b) is summarized in Fig. 3. For instance, the box plot labeled as “Agent 1” depicts the latency experienced by the Raspberry Pi 1 for several communication requests sent to every other Raspberry Pi in the network, that is, Agents 2–4. It is seen that the network latency ranges from 10 to 120 ms. Each consensus update is run every 50 ms and $\bar{\tau}$ is chosen to be 3.

Algorithm 2 is implemented on each agent (emulated by Raspberry Pi) with the neighborhood topology given by Fig. 1(b).

The results are presented in Fig. 4. From Fig. 4, it is clear that the distributed stopping algorithm proposed in this paper terminates when the nodes get close (within tolerance ρ) to the average of the initial conditions in 34 iterations.

Remark 5: Similar to Algorithm 2, Charalambous *et al.* [11] study the distributed stopping of average consensus in the

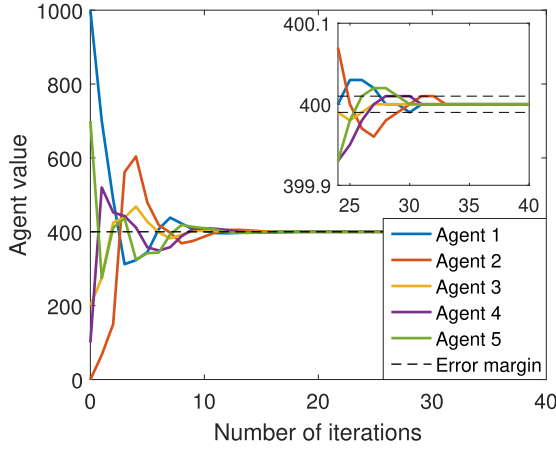


Fig. 4. Experimental results on the network of five nodes, as shown in Fig. 1(b) in the presence of uniformly bounded random delays with Algorithm 2.

presence of time varying but bounded delays using column-stochastic weight matrices to run ratio consensus. Including the two states of ratio consensus, Charalambous *et al.* [11] require the transmission of two messages at each iteration from a node to a neighbor, whereas *Algorithm 2* requires transmission of four messages. Even though the communication overhead is increased, the communication overhead is small compared with the storage and computational advantages of *Algorithm 2* over [11]. The method in [11] needs to compute and process Hankel matrices at each iteration for each node in the graph requiring the storage of Hankel matrices until a matrix rank condition is met. Here, the size of the Hankel matrices increases with iteration count. Thus, in large networks, computing, storing, and processing needs of [11] can be expensive as convergence may take several iterations.

Algorithm 2 computes max-min consensus states where the storage and computational requirements do not increase with the iteration count, thus avoiding scalability issues. This is particularly beneficial for applications where computational overhead is of critical importance and can be permitted at the expense of increased communication load.

VI. CONCLUSION

In this paper, novel maximum and minimum consensus-based distributed stopping criteria are proposed to terminate the consensus iterations by the agents when each of them has reached within a specified error tolerance of the consensus value. Here, when the agents stop the iteration, they are aware that all other agents' have terminated the algorithm as well; which is needed in applications where the distributed result is employed for a subsequent task. The methods proposed are robust to time-varying communication delays. Algorithms for finite-time termination are developed to accommodate the cases when the consensus iterations use a weight matrix that is row stochastic (the standard consensus iterations motivated by [2]) and weight matrix is column stochastic (the ratio consensus iterations for [11]). Here, in the latter case, convergence that is arbitrarily close to the average of the initial values held by agents is guaranteed, where the weight matrix is realizable in a distributed manner without the

need for any centralized coordination. The practicality of the proposed algorithms is tested via simulations and experiments on a network of agents experiencing time-varying communication delays. An interesting extension of the presented algorithms is to incorporate asynchronous computations, which would require the use of asynchronous maximum [30] and minimum consensus.

APPENDIX A

PROOFS OF LEMMAS IN SECTION III

Proof of Lemma III.1: The method of strong induction is employed. Note that $x_i(k) \leq M(k)$ for all $i = 1, \dots, n$, follows from the definition of $M(k)$. Consider the following induction hypothesis:

$$x_i(k+l) \leq M(k) \text{ for all } 0 \leq l \leq L \text{ and } i = 1, \dots, n. \quad (15)$$

Suppose $L \geq \bar{\tau}$. Then, for any $j \in N_i^-$, it follows that $0 \leq L - \bar{\tau} \leq L - \tau_{ij}(k) \leq L$. It follows from the induction hypothesis that $x_j(k+L-\tau_{ij}(k)) \leq M(k)$.

Suppose $0 \leq L \leq \bar{\tau}$. Then, for any $i \in \{1, \dots, n\}$ and $j \in N_i^-$, it follows that $-\bar{\tau} \leq L - \tau_{ij}(k) \leq L$. Consider the case that $-\bar{\tau} \leq L - \tau_{ij}(k) \leq 0$, which implies $k - \bar{\tau} \leq k + L - \tau_{ij}(k) \leq k$ and $x_j(k+L-\tau_{ij}(k)) \leq M(k)$. Consider the case $0 < L - \tau_{ij}(k) \leq L$. Then, from induction hypothesis, it follows that $x_j(k+l-\tau_{ij}(k)) \leq M(k)$. Thus, it is established that for any $j \in N_i^-$, $x_j(k+L-\tau_{ij}(k)) \leq M(k)$. It follows that:

$$\begin{aligned} x_i(k+L+1) &= p_{ii}x_i(k+L) + \sum_{j \in N_i^-} p_{ij}x_j(k+L-\tau_{ij}(k)) \\ &\leq p_{ii}M(k) + \sum_{j \in N_i^-} p_{ij}M(k) = M(k). \end{aligned} \quad (16)$$

Thus, by strong induction, it follows that $x_i(k') \leq M(k)$ for all $k' \geq k$ and for all i . The rest of the proof follows similarly.

Proof of Lemma III.2: The method of strong induction will be employed to prove the Lemma. Note that

$$\begin{aligned} x_i(k+1) &= p_{ii}x_i(k) + \sum_{j \in N_i^-} p_{ij}x_j(k-\tau_{ij}(k)) \\ &< p_{ii}M(k) + \sum_{j \in N_i^-} p_{ij}M(k) = M(k). \end{aligned} \quad (17)$$

The rest of the strong induction step follows the steps of Lemma 3.1 with strict inequalities replacing inequalities.

Proof of Lemma III.3: Since $\min x(k) < \max x(k)$, there exists a node i such that $x_i(k) < \max x(k)$. Since $\max x(k) \leq M(k)$, it follows that $x_i(k) < M(k)$.

Consider any node j . As G is strongly connected, there exists a directed path connecting nodes i and j . Assume the shortest directed path connecting i and j is given by $(m_1, i)(m_2, m_1) \dots (j, m_{d_j-1})$. Using (1), it follows that:

$$\begin{aligned} x_{m_1}(k+\bar{\tau}+1) &= p_{m_1 i}x_i(k+\bar{\tau}-\tau_{m_1 i}(k+\bar{\tau})) \\ &+ \sum_{j \in N_{m_1}^- \cup \{m_1\} \setminus \{i\}} p_{m_1 j}x_j(k+\bar{\tau}-\tau_{m_1 j}(k+\bar{\tau})). \end{aligned}$$

As $(k + \bar{\tau} - \tau_{m_1 j}(k + \bar{\tau})) \geq k$, it follows from the definition of $M(k)$ and (5) that:

$$x_{m_1}(k + \bar{\tau} + 1) \leq p_{m_1 i} x_i(k + \bar{\tau} - \tau_{m_1 i}(k + \bar{\tau})) + \sum_{j \in N_{m_1}^- \cup \{m_1\} \setminus \{i\}} p_{m_1 j} M(k).$$

As $(k + \bar{\tau} - \tau_{m_1 j}(k + \bar{\tau})) \geq k$, it follows from the fact $x_i(k) < M(k)$ and Lemma III.2 that:

$$x_{m_1}(k + \bar{\tau} + 1) < p_{m_1 i} M(k) + \sum_{j \in N_{m_1}^- \cup \{m_1\} \setminus \{i\}} p_{m_1 j} M(k).$$

Thus, $x_{m_1}(k + \bar{\tau} + 1) < M(k)$. Using Lemma III.2, it follows that for all $k'' \geq k + \bar{\tau} + 1$, $x_{m_1}(k'') < M(k)$. Similarly, $x_{m_2}(k + 2\bar{\tau} + 2) < M(k)$ and, hence, it can be concluded using Lemma III.2, that for all $k'' \geq k + 2\bar{\tau} + 2$, $x_{m_2}(k'') < M(k)$. Proceeding in a similar manner for d_j times, it follows that for all $k'' \geq k + d_j \bar{\tau} + d_j$, $x_j(k'') < M(k)$. Note that $k + D\bar{\tau} + D \geq k + d_j \bar{\tau} + d_j$, and, thus, for any $v \in V$, $v \neq i$, $k'' \geq k + D\bar{\tau} + D$ implies $x_v(k'') < M(k)$. It follows that for all $k' \geq k + D(\bar{\tau} + 1)$, $\max x(k') < M(k)$. The proof for the other inequality is similar.

APPENDIX B

PROOFS OF LEMMAS IN SECTION IV

Proof of Lemma IV.1: We prove the theorem using strong induction. By the definition of $M(k)$, for $k' = k$, the proof is trivial.

Suppose, it is asserted that for $k' = k + l'$, $l' \in \{1, 2, \dots, l\}$, $\frac{x_i(k+l')}{u_i(k+l')} \leq M(k)$ for all $i \in V$. Then, for $k' = k + l + 1$

$$\begin{aligned} & \frac{x_i(k+l+1)}{u_i(k+l+1)} \\ &= \frac{p_{ii} x_i(k+l) + \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} x_j(k+l-r) I_{k+l-r, ij}(r)}{p_{ii} u_i(k+l) + \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k+l-r) I_{k+l-r, ij}(r)} \\ &= \frac{p_{ii} \frac{x_i(k+l)}{u_i(k+l)}}{p_{ii} + \frac{1}{u_i(k+l)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k+l-r) I_{k+l-r, ij}(r)} \\ &+ \frac{\frac{1}{u_i(k+l)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} x_j(k+l-r) I_{k+l-r, ij}(r)}{p_{ii} + \frac{1}{u_i(k+l)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k+l-r) I_{k+l-r, ij}(r)}. \end{aligned}$$

Consider the case when $l > \bar{\tau}$. Then, $k+l \geq k+l-r > k + \bar{\tau} - \bar{\tau} = k$. Using the inductive assertion, it follows that $\frac{x_j(k+l-r)}{u_j(k+l-r)} \leq M(k)$. This implies that $x_j(k+l-r) \leq M(k) u_j(k+l-r)$. Thus, it follows that:

$$\begin{aligned} & \frac{x_i(k+l+1)}{u_i(k+l+1)} \leq \\ & \frac{p_{ii} M(k)}{p_{ii} + \frac{1}{u_i(k+l)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k+l-r) I_{k+l-r, ij}(r)} \\ &+ \frac{\frac{1}{u_i(k+l)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k+l-r) I_{k+l-r, ij}(r) M(k)}{p_{ii} + \frac{1}{u_i(k+l)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k+l-r) I_{k+l-r, ij}(r)}. \end{aligned}$$

It follows that $\frac{x_i(k+l+1)}{u_i(k+l+1)} \leq M(k)$ for any $i \in V$. The case when $1 \leq l \leq \bar{\tau}$ can be worked out following the steps in the proof of Lemma III.1.

This proves (13a). Similarly, (13b) can be proven. This completes the proof.

Proof of Lemma IV.2: The proof is based on induction. It follows from the hypothesis that for time instant $k'' = k'$, $\frac{x_i(k')}{u_i(k')} < M(k)$.

Suppose the assertion is that for some integer $q \geq 1$ and for $k'' = k' + q$, $\frac{x_i(k'+q)}{u_i(k'+q)} < M(k)$. This implies that for $k'' = k' + q$, $x_i(k' + q) < M(k) u_i(k' + q)$. Then, for $k'' = k' + q + 1$

$$\begin{aligned} & \frac{x_i(k' + q + 1)}{u_i(k' + q + 1)} < \\ & \frac{p_{ii} M(k)}{p_{ii} + \frac{1}{u_i(k'+q)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k' + q - r) I_{k'+q-r, ij}(r)} \\ &+ \frac{\frac{1}{u_i(k'+q)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} x_j(k' + q - r) I_{k'+q-r, ij}(r)}{p_{ii} + \frac{1}{u_i(k'+q)} \sum_{j \in N_i^-} p_{ij} \sum_{r=0}^{\bar{\tau}} u_j(k' + q - r) I_{k'+q-r, ij}(r)}. \end{aligned} \quad (18)$$

Using Lemma IV.1 and the steps therein, it follows that $\frac{x_j(k'+q-r)}{u_j(k'+q-r)} \leq M(k)$. Using this inequality along with the inductive assertion, the proof follows similar to the proof of Lemma III.2.

The proof of the other inequality is similar to the proof mentioned above and is left to the reader.

Proof of Lemma IV.3: Since $\min R(k) < \max R(k)$, there exists a node $i \in V$ such that $\frac{x_i(k)}{u_i(k)} < \max R(k) \leq M(k)$. Using Lemma IV.2, it follows that $\frac{x_i(k+\bar{\tau}-r)}{u_i(k+\bar{\tau}-r)} < M(k)$, $r \in \{0, 1, \dots, \bar{\tau}\}$. Thus

$$x_i(k + \bar{\tau} - r) < M(k) u_i(k + \bar{\tau} - r), r \in \{0, 1, \dots, \bar{\tau}\}. \quad (19)$$

Using Lemma IV.1, $\frac{x_l(k+\bar{\tau}-r)}{u_l(k+\bar{\tau}-r)} \leq M(k)$, $l \in V \setminus i$, $r \in \{0, 1, \dots, \bar{\tau}\}$. Thus

$$x_l(k + \bar{\tau} - r) \leq M(k) u_l(k + \bar{\tau} - r), l \in V \setminus i, r \in \{0, \dots, \bar{\tau}\}. \quad (20)$$

The rest of the proof is left to the reader. The other inequality can be proven similarly and is left to the reader.

REFERENCES

- [1] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
- [2] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Def. Tech. Inf. Center Doc., Fort Belvoir, VA, USA, 1984.
- [3] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. 44th IEEE Conf. Decis. Control/Eur. Control Conf.*, 2005, pp. 2996–3000.
- [4] Z. Li and Z. Duan, *Cooperative Control of Multi-Agent Systems: A Consensus Region Approach*. Boca Raton, FL, USA: CRC, 2014.

- [5] P. A. Bliman, A. Nedic, and A. Ozdaglar, "Rate of convergence for consensus with delays," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 4849–4854.
- [6] M. Cao, A. S. Morse, and B. D. Anderson, "Reaching a consensus in a dynamically changing environment: A graphical approach," *SIAM J. Control Optim.*, vol. 47, no. 2, pp. 575–600, 2008.
- [7] M. Cao, A. S. Morse, and B. Anderson, "Reaching an agreement using delayed information," in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 3375–3380.
- [8] K. I. Tsianos and M. G. Rabbat, "The impact of communication delays on distributed consensus algorithms," 2012, arXiv:1207.5839.
- [9] K. I. Tsianos and M. G. Rabbat, "Distributed consensus and optimization under communication delays," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 974–982.
- [10] C. N. Hadjicostis and T. Charalambous, "Average consensus in the presence of delays in directed graph topologies," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 763–768, Mar. 2014.
- [11] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N. Hadjicostis, and M. Johansson, "Distributed finite-time average consensus in digraphs in the presence of time delays," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 4, pp. 370–381, Dec. 2015.
- [12] A. D. Dominguez-Garcia and C. N. Hadjicostis, "Distributed algorithms for control of demand response and distributed energy resources," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 27–32.
- [13] S. Patel, S. Attree, S. Talukdar, M. Prakash, and M. Salapaka, "Distributed apportioning in a power network for providing demand response services," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, 2017, pp. 38–44.
- [14] N. E. Manitaras and C. N. Hadjicostis, "Distributed stopping for average consensus in digraphs," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 957–967, Sep. 2018.
- [15] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *Proc. 45th Annu. Allerton Conf. Commun., Control, Comput.*, 2007, pp. 715–720.
- [16] J. He, P. Cheng, L. Shi, J. Chen, and Y. Sun, "Time synchronization in WSNS: A maximum-value-based consensus approach," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 660–675, Mar. 2014.
- [17] P. Xie, K. You, and C. Wu, "How to stop consensus algorithms, locally?" in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 4544–4549.
- [18] A. D. Dominguez-Garcia and C. N. Hadjicostis, "Distributed matrix scaling and application to average consensus in directed graphs," *IEEE Trans. Autom. Control*, vol. 58, no. 3, pp. 667–681, Mar. 2013.
- [19] S. T. Cady, A. D. Dominguez-Garcia, and C. N. Hadjicostis, "Finite-time approximate consensus and its application to distributed frequency regulation in islanded ac microgrids," in *Proc. 48th Hawaii Int. Conf. Syst. Sci.*, 2015, pp. 2664–2670.
- [20] M. Prakash, S. Talukdar, S. Attree, S. Patel, and M. V. Salapaka, "Distributed stopping criterion for ratio consensus," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput.*, 2018, pp. 131–135.
- [21] R. Diestel, *Graph Theory*. Berlin, Germany: Springer-Verlag, 2006.
- [22] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [23] M. El Chamie, "Optimization, control, and game theoretical problems in consensus protocols," Ph.D. dissertation, Université Nice Sophia Antipolis, Nice, France, 2014.
- [24] S. Haldar, "An 'all pairs shortest paths' distributed algorithm using 2n2 messages," *J. Algorithms*, vol. 24, no. 1, pp. 20–36, 1997.
- [25] E. Upton and G. Halfacree, *Raspberry Pi User Guide*. Hoboken, NJ, USA: Wiley, 2014.
- [26] E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11 N and 802.11 AC*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [27] M. Cantelon, M. Harter, T. Holowaychuk, and N. Rajlich, *Node.js in Action*. Shelter Island, NY, USA: Manning, 2014.
- [28] R. Fielding, "Hypertext transfer protocol–http/1.1," Tech. Memo, Network Working Group, 1999.
- [29] K. R. Fall and W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, MA, USA: Addison-Wesley, 2011.
- [30] S. Giannini, A. Pettiti, D. Di Paola, and A. Rizzo, "Asynchronous max-consensus protocol with time delays: Convergence results and applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 2, pp. 256–264, Feb. 2016.



Mangal Prakash received the B.Tech degree in electrical engineering from the National Institute of Technology, Durgapur, India, in 2014, and the M.S. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 2016.

His research interests include control of network systems, probabilistic inference in graphical models, and their applications to biological systems.



Saurav Talukdar received the B.Tech and M.Tech degrees in mechanical engineering from the Indian Institute of Technology Bombay, Mumbai, India, both in 2013. He is currently working toward the Ph.D. degree in mechanical engineering at the University of Minnesota, Minneapolis, MN, USA.

His research interests include learning topology of dynamic systems, multiagent systems, and statistical physics.



Sandeep Attree received the B.Tech and M.Tech degrees in aerospace engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 2012.

He was a graduate student in electrical and computer engineering with the University of Minnesota, Minneapolis, MN, USA. His research interests include network optimization and distributed computing, with applications to smart and autonomous systems.



Vikas Yadav received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from Iowa State University, Ames, IA, USA, both in 2007.

He is currently a Hardware Engineer with the Autopilot division of Tesla, Inc., Palo Alto, CA, USA. He was with Garmin International, Inc., from 2007 to 2013; Qualcomm Technologies, Inc., from 2013 to 2015; LG Electronics from

2015 to 2016; and QuickLogic Corporation from 2016 to 2017. His research interests include sensor fusion, distributed control design, self-organization, and phase transition in large-scale systems.



Murti V. Salapaka (F'19) received the B.Tech. degree in mechanical engineering from the Indian Institute of Technology Madras, Chennai, India, in 1991, and the M.S. and Ph.D. degrees in mechanical engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1993 and 1997, respectively.

He was a faculty member with the Electrical and Computer Engineering Department, Iowa State University, Ames, IA, USA, from 1997 to 2007. He is currently the Director of Graduate Studies and the Vincentine Hermes Luh Chair Professor with the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN, USA. His research interests include control and network science, nanoscience, and single molecule physics.

Dr. Salapaka was a recipient of the 1997 National Science Foundation CAREER Award.