Finite-Time Approximate Consensus and its Application to Distributed Frequency Regulation in Islanded AC Microgrids

Stanton T. Cady and Alejandro D. Domínguez-García
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL, USA
{scady2,aledan}@ILLINOIS.EDU

Christoforos N. Hadjicostis
Department of Electrical and Computer Engineering
University of Cyprus
Nicosia, Cyprus
chadjic@UCY.AC.CY

***Abstract*—In this paper, we propose an iterative algorithm that allows a set of interconnected nodes to reach *approximate consensus* in a finite number of steps, i.e., allows the nodes to reach agreement to within a pre-specified bound in finite time. While this algorithm could be used in many applications, we discuss it within the context of a distributed control architecture for frequency regulation in islanded AC microgrids. Our approach to achieving finite-time approximate consensus builds upon *ratio consensus*, a distributed iterative algorithm in which each node maintains two state variables where the ratio of the states converges asymptotically to a constant that is equal for all the nodes. Simulation results illustrating the operation of our proposed algorithm for two networks are included.**

I. Introduction

Enabled by technological advancements and motivated by a desire to increase efficiency, reliability, and adaptability, microgrids have been proposed as a solution for supplying electric power in an increasing array of applications. The archetypical microgrid is comprised of small, land-based generation units and loads interconnected through an electrical network (e.g., [1], [2]); however, microgrids can also be found in more electric aircraft (e.g., [3]), ships with electric propulsion (e.g., [4]), and telecommunication installations (e.g., [5]).

Whereas large power systems such as the utility grid typically employ centralized control architectures (see, e.g., [6]), several recent works have proposed distributed alternatives for microgrids (see, e.g., [7], [8] and the references therein). Despite having similar control objectives, the desire for adaptability and reliability combined with the adhoc nature of microgrids make them particularly well-suited for control architectures that do not rely on a centralized processor [9], [10]. In this context, the use of consensus algorithms and their derivatives is one approach that has been proposed for distributively controlling the generation units in microgrids (e.g., [10]). More broadly, consensus-based control architectures have been proposed in applications for which adaptability and reliability are critical (see, e.g., [11], [12]).

Regardless of the application, a common approach to consensus relies on a linear iterative strategy in which a network of interconnected nodes repeatedly update locally-maintained states—each node updates its state to be a weighted sum of its own previous state and the previous states of the nodes from which it can receive information. Although the value to which the nodes converge in this scheme contains global information which can be used to make local control decisions, it requires that the nodes iterate indefinitely, i.e., convergence is obtained asymptotically. Moreover, while consensus-based approaches to distributed control have been suggested in the literature, very little work has been done to show how these proposed control architectures can be used in real applications which require finite-time responses. Of particular interest, then, and the primary contribution of this paper, is the development of a scheme for reaching consensus in finite time while providing a mechanism for each node to independently determine when all nodes in the network have converged to the correct solution to within a pre-specified level of accuracy.

The aforementioned development of finite-time consensus algorithms—often referred to as *approximate consensus*—has received very limited attention in the literature thus far. In particular, the authors in [13] propose a scheme which relies on periodically-reset min- and max-consensus computations to provide a signal that is used by the nodes to determine when to stop iterating. This signal ensures that the worst-case error of the primary value being computed remains within some bound. Compared with the algorithm proposed herein, the one in [13] requires that the graph modeling the communication network be balanced which is not a straightforward task in networks with directional links, i.e., communication networks that correspond to directed graphs. The work in [14] proposes a similar scheme that can be used to reach approximate consensus without requiring the parallel min- and max-consensus computations, resulting in a reduction in the amount of information exchanged. However, unlike the algorithm we propose and the one in [13], the scheme in [14] requires that the communication network be undirected and does not provide a mechanism for each node to independently determine when approximate consensus has been reached. In fact, in the approach in [14], nodes may stop transmitting at different times.

The main contribution of this work is the development of an algorithm that allows a collection of nodes to distributively compute a value of interest to within an error bound. Moreover, the algorithm we propose provides a signal which allows the nodes to determine (simultaneously) when to stop iterating while ensuring that the worst-case error lies within the bound. While our algorithm could be used to make adjustments to any type of actuator, we discuss it in the context of adjusting the set-points of generation units in a microgrid to drive the frequency to some nominal value.

Our approach builds upon the *ratio-consensus algorithm* (see, e.g., [15]), which enables operation in unbalanced directed communication networks and provides for fast convergence. To achieve these properties, our algorithm, like the one in [13], relies on finite-time min- and max-consensus iterations; however, compared with [13], we make use of the ratio-consensus algorithm which allows the nodes to reach consensus on values other than the average of the initial conditions.

The remainder of this paper is organized as follows. Section II provides some preliminaries including requisite notions from graph theory, an overview of the ratio-consensus algorithm, and a brief description of the min- and max-consensus algorithms. Section III introduces an event-triggered, distributed control architecture for frequency regulation in microgrids and provides motivation for a finite-time approximate consensus algorithm. In Section IV we formulate the approximate consensus problem, and propose an algorithm that solves it. In Section V we illustrate the operation of our proposed algorithm via simulations and compare it to other solutions. Finally, we provide some concluding remarks in Section VI.

II. Preliminaries

We begin this section by outlining a graph-theoretic model to represent a communication network interconnecting the local controllers of distributed generation resources (DGRs) in a microgrid. Next, we provide an overview of ratio consensus, a linear iterative algorithm on which we base our finite-time approximate consensus algorithm. The finite-time approximate consensus algorithm, and by extension ratio consensus, is the foundation of our distributed frequency regulation architecture. Finally, we introduce the min- and max-consensus algorithms which are used to provide a stopping signal in our finite-time scheme.

A. Graph-Theoretic Communication Network Model

To realize a distributed control architecture for frequency regulation, we assume that each DGR is outfitted with a local controller that is capable of simple computations. Furthermore, we assume that each local controller is equipped with a transceiver with which information can be sent to and received from the local controllers of other nearby DGRs.

In order to maintain a general communication modality and to allow for asymmetric communication links, we represent the exchange of information between DGRs by a directed graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Given a microgrid with n DGRs, $\mathcal{V} = \{1, \ldots, n\}$ is the set of vertices, where vertex (or node) $i \in \mathcal{V}$ corresponds to the local controller of DGR i, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges, where the ordered pair $(i, j) \in \mathcal{E}$ corresponds to a communication link through which node i can receive information from node j. For notational convenience, we include self-loops for each node; thus, $(i, i) \in \mathcal{E}$, $\forall i \in \mathcal{V}$.

We denote the set of nodes from which the local controller of DGR i can receive information by $\mathcal{N}_i^- = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Analogously, $\mathcal{N}_i^+ = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ is the set of nodes to which the local controller of DGR i can send information. We refer to $\mathcal{N}_i^-$ and $\mathcal{N}_i^+$ as the in- and out-neighborhood of node i and denote the cardinality of each set by $\mathcal{D}_i^- = |\mathcal{N}_i^-|$ and $\mathcal{D}_i^+ = |\mathcal{N}_i^+|$, respectively. Given that self-loops are included, each node is in both its own in- and out-neighborhood, i.e., $i \in \mathcal{N}_i^-$ and $i \in \mathcal{N}_i^+$, $\forall i \in \mathcal{V}$.

A directed path of length t from node j to i is a sequence of vertices, $j \equiv l_0, l_1, \ldots, l_t \equiv i$, such that $(l_{\tau+1}, l_\tau) \in \mathcal{E}$ for $\tau = 0, 1, \ldots, t-1$. The minimum distance from node j to node i, $j \neq i$, is defined to be the shortest length path from j to i and is denoted $d_{\min}(i, j)$; if no (directed) path exists, $d_{\min}(i, j) = \infty$. Throughout this paper, we assume that $\mathcal{G}$ is strongly connected, i.e., there exists a directed path of finite length from node j to i for all $i, j \in \mathcal{V}$. Furthermore, the diameter of $\mathcal{G}$ is defined as the longest shortest path between any two vertices, i.e., $D = \max_{i,j \in \mathcal{V}, i \neq j} d_{\min}(i, j)$ [16].

B. The Ratio-Consensus Algorithm

Ratio consensus is an iterative algorithm that relies on linear updates to a pair of local states maintained by each node. When properly initialized, the ratio-consensus algorithm allows the local controller of each DGR to asymptotically acquire aggregate information about the state of all the DGRs in system; this information is then used to make local control decisions, the collective effect of which can be used to, for example, drive the frequency in a microgrid to some nominal value.

Consider a microgrid comprised of n DGRs equipped with local controllers where the information exchange between them is described by the previously-mentioned directed graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. To participate in ratio consensus, the local controller of each DGR i maintains two states, y_i and z_i, which are iteratively updated to be a linear combination of the previous states of all the nodes in its in-neighborhood. More specifically, let $k \geq 0$ index the iterations; then, each node i updates its states according to

$$y_i[k+1] = \sum_{j \in \mathcal{N}_i^-} p_{ij} y_j[k], \tag{1}$$

$$z_i[k+1] = \sum_{j \in \mathcal{N}_i^-} p_{ij} z_j[k], \tag{2}$$

where the weights are chosen such that $p_{ij} = \frac{1}{\mathcal{D}_j^+}$. While this choice of weights is not unique, it is a simple one that can be determined distributively and ensures that the weight matrix, $P = [p_{ij}]$, is column stochastic which is necessary to guarantee the following result.

If we assume that $z_i[k] \neq 0$, $\forall k$, each node can compute the ratio

$$\gamma_i[k] = \frac{y_i[k]}{z_i[k]} \tag{3}$$

at each iteration, k. Given the choice of weights specified above and the strong connectivity of $\mathcal{G}$, it can be shown that the value in (3) converges to a constant that is equal for every node i. In particular, as proven in [15], we have that

$$\gamma := \lim_{k\to\infty} \gamma_i[k] = \frac{\sum_{j=1}^{n} y_j[0]}{\sum_{j=1}^{n} z_j[0]}, \ \forall i \in \mathcal{V}. \tag{4}$$

In Section III we will discuss how the ratio-consensus algorithm in (1)–(4) can be used to distributively regulate frequency in a microgrid.

C. Min and Max Consensus

As mentioned previously, min and max consensus will be used in our proposed scheme to provide a signal which indicates when all nodes have converged to within a specified error bound. Given a vector of initial values, $M[0] := \big[M_1[0], \ldots, M_n[0]\big]^{\mathrm{T}}$, the max-consensus algorithm allows each node to determine the maximum of the initial values, i.e, $\|M[0]\|_{\infty}$.

Each node i participating in the max-consensus algorithm maintains a value, $M_i[k]$, which is updated at each iteration, k, according to

$$M_i[k+1] = \max_{j\in\mathcal{N}_i^-} M_j[k]. \tag{5}$$

As proven in [13], after a finite number of iterations bounded by the diameter of the network, every node converges to the same value which is the maximum of the initial values, i.e., for some $k_0 \leq D$, we have $M_i[k] = \|M[0]\|_{\infty}$, $\forall i \in \mathcal{V}$, $k \geq k_0$. [Note that min consensus is identical to the discussion above on max consensus with the $\max$ operation in (5) replaced by the operation $\min$.]

III. Distributed Frequency Regulation

Compared with bulk power transmission systems, microgrids have fewer loads, each of which may be large relative to the total capacity of the DGRs; consequently, there may be periods during which the frequency does not vary significantly from its nominal value, while any single load change may result in a substantial frequency deviation. Furthermore, while high operational expenses in large power systems necessitate cost minimization, functioning at the desired operating point is paramount in microgrids. To account for these characteristics, we adopt a generation control scheme in which frequency regulation is the main objective [9], and the control actions are event-triggered, i.e., DGR set-points are adjusted only when the frequency deviates beyond some bound.

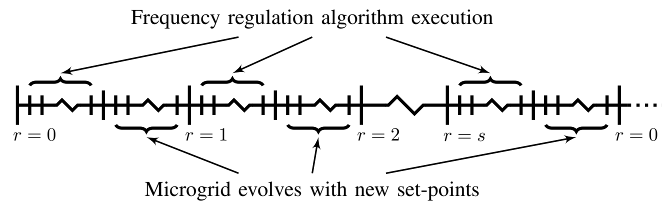

Fig. 1. Timeline of distributed generation control algorithms.

A. Fair-Splitting Allocation Scheme

To return the frequency in a microgrid to its nominal value following one or more load changes, our distributed regulation scheme relies on the execution of the ratio-consensus algorithm over several rounds (see Fig. 1). Let r index the rounds over which the scheme is executed, and denote the amount by which the frequency deviates from the nominal value (henceforth referred to as *frequency regulation error*) at round r by $\Delta\omega^r = \omega^r - \omega_0$, where ω^r is the microgrid frequency at round r and ω_0 is the nominal value of the frequency.[1] Additionally, let u_i^r be the set-point of DGR i at round r; then the set-point of DGR i at consecutive rounds, r and $r+1$, are related by

$$u_i^{r+1} = \Delta u_i^r + u_i^r, \tag{6}$$

where Δu_i^r is the amount by which DGR i should adjust its set-point at round r to drive the frequency regulation error, $\Delta\omega_r$, to zero.

Let $\underline{P}_i$ and $\overline{P}_i$ be the lower and upper limit on the set-point of DGR i, respectively; then, we define $\Delta\underline{P_i}^r := \underline{P}_i - u_i^r$ and $\Delta\overline{P_i}^r := \overline{P}_i - u_i^r$ to be the incremental lower and upper limit of DGR i at round r, respectively. If we assume that each DGR can get an estimate of $\Delta\omega^r$ by measuring the frequency of the bus to which it is connected, DGR i estimates the amount by which it should adjust its set-point as

$$\Delta\hat{u}_i^r = \kappa_i \Delta\omega^r, \tag{7}$$

where $\kappa_i < 0$ is some gain chosen by the local controller of DGR i. Note that choosing appropriate κ_i's is crucial for stable operation of the closed-loop system, but a detailed explanation is beyond the scope of this paper. Assuming a feasible solution exists, i.e.,

$$\sum_{i=1}^{n} \Delta\underline{P}_i^r \leq \sum_{i=1}^{n} \Delta\hat{u}_i^r \leq \sum_{i=1}^{n} \Delta\overline{P}_i^r, \tag{8}$$

each node i uses ratio consensus to determine Δu_i^r by initializing the states in (1) and (2) to $y_i^r[0] = \Delta\hat{u}_i^r - \Delta\underline{P}_i^r = \kappa_i\Delta\omega^r - \underline{P}_i + u_i^r$ and $z_i^r[0] = \Delta\overline{P_i}^r - \Delta\underline{P_i}^r = \overline{P}_i - \underline{P}_i$, respectively. Given these initial conditions, it follows from (4)

[1] We assume that control actions are taken after the system has reached steady state and all DGRs operate at the same common frequency.

that the local controller of DGR i can asymptotically obtain

$$\begin{aligned}\gamma^r &= \lim_{k\to\infty} \gamma_i^r[k] = \lim_{k\to\infty} \frac{y_i^r[k]}{z_i^r[k]}, \\ &= \frac{\sum_{j=1}^{n} \kappa_j \Delta\omega^r - \underline{P}_j + u_j^r}{\sum_{j=1}^{n} \overline{P}_j - \underline{P}_j},\end{aligned} \tag{9}$$

which corresponds to the ratio of the sum of the estimated set-point adjustments to the collective incremental power output limits of the DGRs.

After ratio consensus has converged, each node i obtains γ^r and can compute the amount by which it should adjust its set-point at round r as

$$\begin{aligned}\Delta u_i^r &= \Delta \underline{P}_i^r + \gamma^r \left(\Delta \overline{P}_i^r - \Delta \underline{P}_i^r\right), \\ &= \underline{P}_i - u_i^r + \gamma^r \left(\overline{P}_i - \underline{P}_i\right).\end{aligned} \tag{10}$$

Together with the choice of initial conditions specified above, the scheme in (7)–(10) allocates incremental demand to each DGR based upon incremental power output limits; as a result, we refer to this allocation scheme as *fair splitting*.

B. Frequency Regulation from Fair Splitting

Given that the allocation scheme described above adjusts generation set-points incrementally, when executed over several rounds, it drives the frequency error to zero, replicating the functionality of a discrete-time integral controller.

Let ϵ_f denote the maximum amount by which the frequency can deviate from the nominal value, ω_0; then, our event-triggered control architecture is initiated when one of the DGRs measures a frequency error that exceeds the bound, i.e., $|\Delta\omega^r| > \epsilon_f$. After using ratio consensus to obtain γ^r, adjusting generation set-points, and waiting for the system to reach steady state, the DGRs may initiate another execution of the fair-splitting scheme, continuing in this manner until the frequency returns to within ϵ_f of the nominal value. The timeline in Fig. 1 shows the execution of several rounds of the fair-splitting scheme, where the frequency error is driven sufficiently close to zero at round $r = s$. As the figure illustrates, each round consists of executing the ratio-consensus algorithm, adjusting the DGR set-points, and waiting for the microgrid to reach steady state.

The Need for Finite-Time Approximate Consensus: Although the distributed scheme described above can be shown to regulate the frequency in a microgrid (see, e.g., [10]), the result in (4) implies that the value with which DGR local controllers make decisions, γ, is found asymptotically. In practice, executing a finite number of iterations is adequate to compute γ to sufficient accuracy, however, the number of iterations to be executed must be specified beforehand to guarantee a sufficiently-accurate solution. In the next section, we propose an algorithm based upon ratio consensus that allows the nodes to simultaneously determine when to stop iterating while ensuring that γ is sufficiently close to the asymptotic solution without pre-specifying the number of iterations to be performed.

IV. Finite-Time Approximate Consensus

First, we provide a formal statement of the problem to be addressed. Then, we introduce an algorithm which uses min- and max-consensus computations to determine the error among the values computed by the nodes using ratio consensus; this approach allows the nodes to simultaneously determine when to stop iterating while ensuring that the worst-case error of the ratio is within a specified bound.

A. Approximate Consensus Problem

We consider fixed communication networks which can be described by the directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as introduced in Section II-A. Although it is not a requirement, a broadcast model is used for all information exchanges, i.e., each node sends the same data to all nodes in its out-neighborhood. Furthermore, we assume that all transmitted information is successfully received by the intended recipient, i.e., the communication network is completely reliable, however, our proposed strategy could be adapted to utilize the algorithm proposed in [17] to mitigate the effects of links that do not have perfect availability, i.e., they drop packets.

The following definition and problem statement serve to formalize the objective of our proposed algorithm.

Definition 1: (ϵ-Approximate Ratio Consensus) Consider a network of nodes described by a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. Each node i maintains two states, y_i and z_i, which are updated according to (1) and (2), respectively, for k_0 iterations. At the end of this iterative process, the nodes have reached ϵ-approximate ratio consensus if

$$\left|\frac{y_i[k_0]}{z_i[k_0]} - \gamma\right| < \epsilon, \ \forall i \in \mathcal{V},$$

where γ is as defined in (4).

Problem Statement: Let ϵ be an error bound specified *a priori* and known by all nodes; then, the goal is to develop a distributed scheme that allows each node i to use ratio consensus to compute γ_i in finite time such that $|\gamma_i - \gamma| < \epsilon$, $\forall i \in \mathcal{V}$. More specifically, the objective is to devise a distributed protocol that allows the nodes to reach ϵ-approximate ratio consensus.

B. Algorithm for Approximate Consensus

As mentioned previously, the proposed algorithm takes advantage of min- and max-consensus iterations to allow the nodes to determine the time step, k_0, when their ratios,

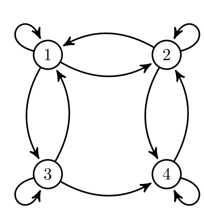

Fig. 2. Graph of 4-node network.

Algorithm 1: Approximate Ratio Consensus with Distributed Stop Detection

Input: $y_i[0], z_i[0], \epsilon$
Output: γ

begin
  Set

$$M_i[0] = \infty, m_i[0] = -\infty$$
$$ratio_i = y_i[0]/z_i[0], p_{ji} = \frac{1}{D_i^+}$$

  Let $k \geq 0$ index iterations
  while $u_i[k] = 0$ **do**
    if $k \mod D = 0$ **then**
      if $M_i[k] - m_i[k] < \epsilon$ **then**
        Set $u_i[k] = 1$
      Set $M_i[k] = m_i[k] = ratio_i = y_i[k]/z_i[k]$
    Broadcast to all $j \in \mathcal{N}_i^+$

$$p_{ji}y_i[k], p_{ji}z_i[k], M_i[k], m_i[k]$$

    Receive from all $l \in \mathcal{N}_i^-$

$$p_{il}y_l[k], p_{il}z_l[k], M_l[k[, m_l[k]$$

    Compute

$$y_i[k+1] = \textstyle\sum_{l \in \mathcal{N}_i^-} p_{il}y_l[k]$$
$$z_i[k+1] = \textstyle\sum_{l \in \mathcal{N}_i^-} p_{il}z_l[k]$$
$$M_i[k+1] = \max_{l \in \mathcal{N}_i^-} M_l[k]$$
$$m_i[k+1] = \min_{l \in \mathcal{N}_i^-} m_l[k]$$

  return $ratio_i =: \gamma$

$\{\gamma_i[k_0] \mid i \in \mathcal{V}\}$, are within ϵ of each other. Similar to ratio consensus, each node i participating in our proposed algorithm will maintain two states, $y_i[k]$ and $z_i[k]$, referred to as *primary states*, which will be used as before to compute the value γ for making local control decisions. Additionally, each node i will maintain two *auxiliary states*, $m_i[k]$ and $M_i[k]$, which are updated using min- and max-consensus, respectively. As in [13], the auxiliary states, $m_i[k]$ and $M_i[k]$, will be periodically reinitialized.

In order to allow the nodes to determine when all the ratios in the set $\{\gamma_i \mid i \in \mathcal{V}\}$ are close to the asymptotic value, we use min- and a max-consensus, reset every D steps. More specifically, when $k \mod D = 0$, the values of $m_i[k]$ and $M_i[k]$ are reinitialized to be $m_i[k] = M_i[k] = \gamma_i[k]$. Before reinitializing the values of $m_i[k]$ and $M_i[k]$, node i checks if the worst case error between the node ratios, given by $|M_i[k] - m_i[k]| = M_i[k] - m_i[k]$ is smaller than the desirable error bound, i.e., if $M_i[k] - m_i[k] < \epsilon$ for $k \mod D = 0$; if that is the case, then the nodes stop iterating. The pseudocode for the algorithm above is provided in Algorithm 1; we omit a convergence proof due to space limitations.

V. Simulation Results

In this section we provide simulation results which illustrate the operation of Algorithm 1 for two networks. The first network we consider matches the one in [10] which we use to illustrate how the approximate consensus algorithm could be used to regulate frequency in a microgrid. The second network contains 12 nodes which we use to compare the performance of our algorithm with the ones in [13] and [14].

A. 4-Node Network

In [10] we used a 4-bus laboratory microgrid to demonstrate a distributed frequency regulation scheme based upon ratio consensus. In the experimental setup in [10], the exchange of information between the DGR local controllers is modeled by the graph in Fig. 2 and the number of iterations to be executed by the nodes is specified beforehand to be 75. Furthermore, in the results in [10], it was shown that 3 rounds of the fair spitting scheme were necessary to drive the frequency sufficiently close to zero. Thus, following one load change, a total of 225 ratio consensus iterations were required.

Using Algorithm 1, we simulated the same network with similar initial conditions to determine the number of iterations necessary to converge to a solution in which $|\gamma_i - \gamma| < 0.0001$, i.e., $\epsilon = 0.0001$. The plots in Figs. 3, 4, and 5 show the evolution of the primary ratio, min consensus, and max consensus as computed by the nodes, respectively; as the figures illustrate, only 15 iterations are required before the nodes determine the maximum error is below the bound.

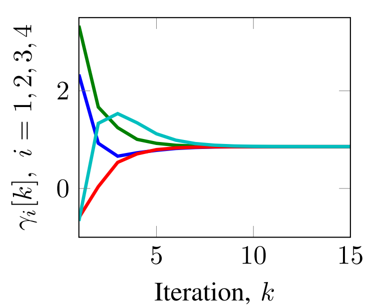

Fig. 3. Primary ratio evolution for 4-node network.

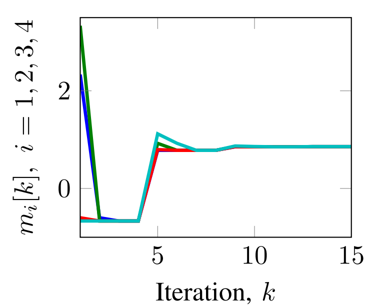

Fig. 4. Evolution of min consensus for 4-node network.

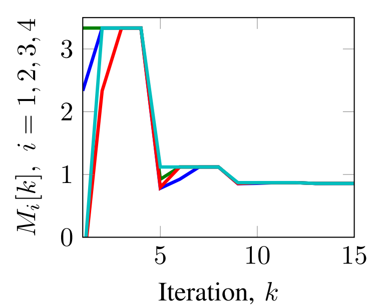

Fig. 5. Evolution of max consensus for 4-node network.

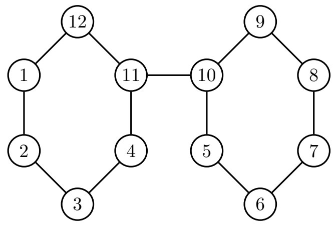

Fig. 6. Graph of 12-node network (self loops omitted).

Compared with the results in [10], the finite-time approximate consensus scheme requires only 48 iterations.

B. 12-Node Network

While ratio consensus, and by extension, the algorithm proposed in this paper, can be used to compute more general quantities over a network of interconnected nodes, if properly initialized, it can also be used to compute the average of some values possessed by the nodes. More specifically, suppose each node i possesses a value x_i, to find the average of these values, it suffices to initialize the states in (1) and (2) to $y_i[0] = x_i$ and $z_i[0] = 1$, for all $i \in \mathcal{V}$, respectively. As implied by (4), the ratio computed by the nodes will asymptotically approach $\gamma = \frac{\sum_{j=1}^{n} x_j}{\sum_{j=1}^{n} 1} = \frac{1}{n}\sum_{j=1}^{n} x_j$, i.e., the average of the values x_i. Using this approach, we compare our proposed algorithm with the ones in [13] and [14] by simulating it for the network illustrated by the graph in Fig. 6.

As in the previously-referenced work, we use our proposed algorithm to compute the average of the following values

$$x = \left[a, a, a, a, b, b, b, b, b, b, a, a\right]^{\mathrm{T}},$$

where $a = 0.001$ and $b = 100$; thus, the true average is, $\overline{x} = 50.0005$. Similarly, the accuracy to which the average is to be computed is $\epsilon = 0.0001$. Given these initial conditions and bound, the plots in Figs. 7, 8, and 9 show the evolution of the primary ratio, min consensus, and max consensus as computed by the nodes, respectively. As these figures illustrate, the nodes converge rapidly, requiring only 263 iterations to determine that all the ratios of the primary states are within ϵ of the true average.

To reach the same level of accuracy, the algorithms in [13] and [14] require 1059 and 1095 iterations, respectively; comparatively, our approach requires 24.8% and 24.0% fewer iterations, respectively. While this represents a significant reduction in the number of iterations required, nodes in our approach must maintain 4 state variables compared to 3 and 1 in [13] and [14], respectively. Despite this additional overhead, the algorithm we propose can be used for more general computations and works in networks for which the graph representing the information exchange is directed and unbalanced. Furthermore, the effects on the communication network would be minimal as all values exchanged between nodes in our approach can be included in a single packet of even the most simple protocols, e.g., IEEE 802.15.4.

VI. Concluding Remarks

In this paper we proposed a finite-time algorithm that can be used to distributively compute a value which a network of nodes can use to make local control decisions. Compared with previously-proposed algorithms, our approach works in directed communication networks and can be used for computations other than averaging.

While our algorithm is suitable for other applications for which distributed control is desirable (for example, voltage control in distribution systems [18]), we discussed its effectiveness for distributed frequency regulation in microgrids.

References

[1] R. Lasseter *et al.*, "Integration of distributed energy resources: The CERTS microgrid concept," Lawrence Berkeley National Laboratory, Tech. Rep. LBNL-50829, Apr. 2002.

[2] F. Katiraei and M. Iravani, "Power management strategies for a microgrid with multiple distributed generation units," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1821–1831, Nov. 2006.

[3] J. Rosero, J. Ortega, E. Aldabas, and L. Romeral, "Moving towards a more electric aircraft," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 3, pp. 3–9, Mar. 2007.

[4] K. Butler, N. Sarma, and V. Ragendra Prasad, "Network reconfiguration for service restoration in shipboard power distribution systems," *IEEE Transactions on Power Systems*, vol. 16, no. 4, pp. 653–661, Nov. 2001.

[5] T. Gruzs and J. Hall, "Ac, dc or hybrid power solutions for today's telecommunications facilities," in *International Telecommunications Energy Conference*, 2000, pp. 361–368.

[6] A. Wood and B. Wollenberg, *Power Generation, Operation, and Control*. New York, NY: Wiley, 1996.

[7] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, 2013.

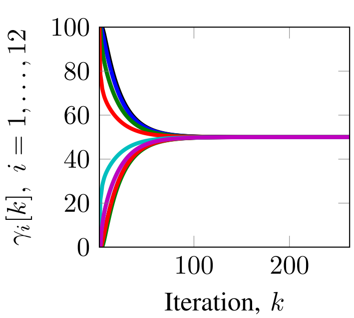

Fig. 7. Primary ratio evolution for 12-node network.

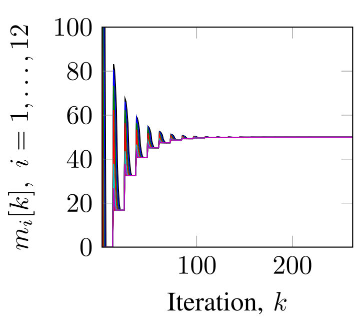

Fig. 8. Evolution of min consensus for 12-node network.

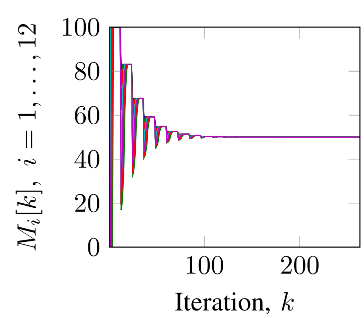

Fig. 9. Evolution of max consensus for 12-node network.

[8] A. D. Domínguez-García, S. T. Cady, and C. N. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources," in *Proc. IEEE Conference on Decision and Control*, 2012, pp. 3688–3693.

[9] J. Peças Lopes, C. Moreira, and A. Madureira, "Defining control strategies for microgrids islanded operation," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 916–924, May 2006.

[10] S. T. Cady and A. D. Dominguez-Garcia, "Distributed generation control of small-footprint power systems," in *Proc. of North American Power Symposium*, Sept. 2012, pp. 1–6.

[11] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, July 2006.

[12] W. Ren, R. Beard, and E. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Systems*, vol. 27, no. 2, pp. 71–82, April 2007.

[13] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *Communication, Control, and Computing (Allerton), 2007 45th Annual Allerton Conference on*, Oct. 2007, pp. 715–720.

[14] N. Manitara and C. N. Hadjicostis, "Distributed stopping in average consensus via event-triggered strategies," in *Allerton Conference on Communication, Control, and Computing*, Oct. 2013, pp. 1336–1343.

[15] A. D. Domínguez-García and C. N. Hadjicostis, "Distributed algorithms for control of demand response and distributed energy resources," in *Proc. of IEEE Conference on Decision and Control*, 2011, pp. 27–32.

[16] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, 2001.

[17] A. D. Domínguez-García, C. N. Hadjicostis, and N. Vaidya, "Resilient networked control of distributed energy resources," *IEEE Journal on Selected Areas in Comm.*, vol. 30, no. 6, pp. 1137–1148, Jul. 2012.

[18] B. Robbins, C. Hadjicostis, and A. Dominguez-Garcia, "A two-stage distributed architecture for voltage control in power distribution systems," *Power Systems, IEEE Transactions on*, vol. 28, no. 2, pp. 1470–1482, May 2013.