

Finite Time Protocols for Multi-Agent Control of Distributed Generation and Responsive Loads

Polymeneas, E.; Benosman, M.

TR2014-040 June 2014

Abstract

Distributed Resources in the Smart Grid, such as Distributed Generation (DG) and Responsive Loads (RL) are capable of providing a wide range of ancillary services, if properly coordinated. However, because of their large numbers and distributed nature, using a fully centralized communication structure to achieve coordination can be prohibitive, due to scale and cost. Decentralized approaches based on multi-agent systems theory have been proposed and are focused on linear consensus algorithms that asymptotically converge to a cooperative solution. In this paper, an alternative approach, which converges in a finite number of steps is formulated. The protocol is based on linear iterative updates and known observability results from graph structured linear systems. Here, the protocol is also modified to address a generalized cooperation problem in the context of DG/RL coordination. Furthermore, the protocol is modified to reduce the number of operations per step, ensuring that it is applicable to a large scale grid, which is usually the case in practice. The applicability of the approach is verified through simulations, for the case of grid voltage support through Distributed Generation.

American Control Conference (ACC)

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Finite Time Protocols for Multi-Agent Control of Distributed Generation and Responsive Loads

Evangelos Polymeneas and Mouhacine Benosman

Abstract—Distributed Resources in the Smart Grid, such as Distributed Generation (DG) and Responsive Loads (RL) are capable of providing a wide range of ancillary services, if properly coordinated. However, because of their large numbers and distributed nature, using a fully centralized communication structure to achieve coordination can be prohibitive, due to scale and cost. Decentralized approaches based on multi-agent systems theory have been proposed and are focused on linear consensus algorithms that asymptotically converge to a cooperative solution. In this paper, an alternative approach, which converges in a finite number of steps is formulated. The protocol is based on linear iterative updates and known observability results from graph structured linear systems. Here, the protocol is also modified to address a generalized cooperation problem in the context of DG/RL coordination. Furthermore, the protocol is modified to reduce the number of operations per step, ensuring that it is applicable to a large scale grid, which is usually the case in practice. The applicability of the approach is verified through simulations, for the case of grid voltage support through Distributed Generation.

I. INTRODUCTION

Moving towards the Smart Grid, it has been recognized [1] that distributed resources have the potential to provide a wide range of valuable grid ancillary services, such as voltage support in the distribution system and/or active power reserves [2]. The future grid is expected to be characterized by increased participation of distributed agents in the control procedure. In this context, it becomes increasingly important to develop viable procedures to monitor and control the numerous devices that are projected to be involved in the control flow. Multi-Agent Systems (MAS), consensus-based control theory has been extensively studied for the control of distributed systems, e.g. [3], and seems suitable for application in the Smart Grid. Such approaches have already been investigated: For instance, [4] discusses a class of consensus based linear iterative algorithms to be used for Distributed Energy Resources (DER) coordination and [5] presents an extension of that work for asynchronous communication between the DER's. [6] presents a hierarchical approach for controlling DER's using MAS theory, en route to achieving the Virtual Power Plant (VPP) concept. According to this paradigm, clusters of DER's will be able to collectively provide the same control functions or services actual power plants provide today. The feasibility of using distributed inverters for consensus-based reactive power support was first proposed in [7], while a voltage control scheme is analyzed in [8]. The above methods rely on consensus algorithms based on linear updates with neighbors. Such algorithms have been proven to *asymptotically* achieve consensus between all nodes in a system with graph-structured communications, with assumptions on the connectivity of the graph. This paper focuses on enhancing the convergence speed of the consensus using a different type of consensus protocols, based on the work in [9], that can solve the DER

Mouhacine Benosman (m_benosman@ieee.org) is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA.

Evangelos Polymeneas (epolym@gatech.edu) is with School of Electrical and Computer Engineering Georgia Institute of Technology, Atlanta, GA 30332 .

coordination problems in *finite time*. After the DER coordination problem is defined, this paper focuses on modifying the protocol in [9] to address this problem. After these modifications, the resulting protocol requires a number of iterations per step at each agent that increase linearly with respect to the number of the agent's neighbors in the communications graph, which is a desirable property.

As an example application of the finite-time algorithm, reactive power control from a set of distributed inverters is considered in this work. The premise is as follows: the power electronics inverters that interface DER's to the grid have the capability to provide reactive power support. Given that noticeable problems arise in the voltage profile of distribution grids due to the interconnection of DER's [10], controlling the reactive power supplied by the utility-connected inverters to maintain an improved voltage profile has been suggested. Centralized voltage control has been proposed [11], but it requires considerable communication overhead and all the calculations are performed centrally. Fully decentralized voltage control using a voltage-reactive power droop function have also been studied [12]. These approaches are sensitive to the topology of the distribution grid and its operating point. Consensus based voltage-control was first proposed in [13], where a two-level voltage control scheme is presented: all inverters participate in local control and request additional support when they reach their reactive power limits using a multi-agent consensus protocol. The outlined protocol converges asymptotically, and the convergence rate might increase considerably for large networks. In this work, the derived finite-time DER coordination protocol is applied to the problem of reactive power allocation among inverters. It is demonstrated in section V that the proposed protocol offers improved convergence rate , thus allowing faster voltage control in the distribution grid.

The rest of the paper is organized as follows. Section II formally introduces the generalized problem of decentralized DER coordination and highlights the main result of this paper. Section III discusses key results regarding finite-time coordination between agents based on linear weighted updates and observability of graph-structured systems, and introduces an improved protocol for finite-time calculation of the sum of the initial values of the nodes in a connected graph. Section IV utilizes these concepts in order to provide a complete decentralized protocol for DER coordination. Section V presents numerical results from the application of the protocol to an actual DER coordination problem, and compare it with an asymptotically converging consensus protocol. Section VI offers conclusions and remarks regarding the direction for future research.

II. PROBLEM DEFINITION - HIGHLIGHT OF MAIN RESULT

Assume a group of N distributed energy resources (DER's) and/or responsive loads (RL's). Let each DER/RL be represented by a node and let each node be uniquely represented by an integer. Without loss of generality, let the index set be $\mathcal{V} = \{1, 2, \dots, N\}$. It is assumed that each of the distributed resources is able to receive messages with a set of neighbors $\mathcal{N}_j^- \subseteq \mathcal{V}$. Define the in-degree

D_j^- of node j as the number of nodes (including node itself) from which node j receives messages: $D_j^- = |\mathcal{N}_j^-|$. Thus, the availability of communications channels can be represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} = \{(j, i) \in \mathcal{V} \times \mathcal{V} : i \in \mathcal{N}_j^-\}$. The main standing assumptions for the sequel are that: i) Every node can receive messages from itself, i.e. $(i, i) \in \mathcal{E}, \forall i \in \mathcal{V}$ ii) The graph \mathcal{G} is connected, i.e. there exists a (directed) path between every pair of nodes in \mathcal{V} and iii) The graph is time-invariant, i.e. both the vertex set \mathcal{V} and the edge set \mathcal{E} remain constant with respect to time.

Let the state of node j be denoted by x_j , which will represent a quantity of interest, such as the node's injected active or reactive power. Each node is able to control its state instantaneously (i.e. the node's dynamics are disregarded in the time scale of interest), within a minimum and maximum boundary, \underline{x}_j and \bar{x}_j respectively. Let us assume that a coordinating controller is responsible for setting the reference signal for the DER's/RL's to collectively follow and periodically determines a reference value X_r , but due to constraints on available communications it only has the capability to convey that command to a nonempty subset of the nodes $\mathcal{I}_0 \subseteq \mathcal{V}$. In this work, we are interested in deriving a distributed iterative protocol between neighboring nodes that will converge to a feasible solution in a *finite number of iterations* (message exchanges). If $x \in \mathbb{R}^N$ is the collective state vector of all nodes, the set \mathcal{F} of feasible solutions is a polyhedron defined by each node's individual capacity constraints and the requirement that the individual states must sum up to the reference:

$$\mathcal{F} = \left\{ x \in \mathbb{R}^N : \sum_{j=1}^N x_j = X_r, \underline{x}_j \leq x_j \leq \bar{x}_j \right\} \quad (1)$$

The objective of this paper is to formulate a discrete-time protocol that will consist of linear operations at each node and will allow each node to calculate its state element $x_j \in R$ such that the overall state vector $x \in R^N$ is feasible, assuming that \mathcal{F} is nonempty. Let each node store m information states $z_{j,l}[k]$, $l \in \{1, 2, \dots, m\}$ at step k and let $z_l[k] \in \mathbb{R}^N$ denote node the state vector for the entire network with respect to the l -th state. Each node initializes its information states based on local information. Nodes in \mathcal{I}_0 initially have information regarding the reference X_r , while all other nodes do not. All nodes have information regarding their maximum and minimum boundaries, \bar{x}_j and \underline{x}_j respectively. Let $y_{j,l}[k] \in \mathbb{R}^{D_j^-}$ denote the observation vector of node j with respect to state l at step k and $\hat{z}_{j,l}[k] \in \mathbb{R}$ be an auxiliary estimate of node j at step k . In this work distributed protocol will be derived that allows the nodes to converge to a feasible solution via the following steps:

- 1) Each node initializes its information vector based on local information, as well as its estimate:

$$z_{j,l}[0] = \begin{cases} f_{j,l}(\bar{x}_j, \underline{x}_j) & , \text{if } j \notin \mathcal{I}_0 \\ f_{j,l}(\bar{x}_j, \underline{x}_j, X_r) & , \text{if } j \in \mathcal{I}_0 \end{cases} \quad (2a)$$

$$\hat{z}_{j,l}[0] = 0 \quad (2b)$$

Where the $f_{j,l}$ are certain initialization functions, which will be appropriately chosen in the sequel.

- 2) In each step k , each node j performs three linear operations:

$$z_{j,l}[k+1] = p_{jj} z_{j,l}[k] + \sum p_{ji} z_{i,l}[k] \quad (3a)$$

$$y_{j,l}[k] = C_j z[k] \quad (3b)$$

$$\hat{z}_{j,l}[k+1] = \hat{z}_j[k] + a_j^T[k] y_{j,l}[k] \quad (3c)$$

These operations are parametrized by $P \in M_N^1$, $C_j \in M_{D_j^-, N}$ and $a_j[k] \in R^{D_j^-}$ which will be appropriately defined in the sequel. Note that (3a) is a linear weighted update of node j 's information states based on its neighbors' states, (3b) is a linear observation of the neighbors' state values and (3c) is an update of node j 's auxiliary states, according to the observations at step k . Each node's auxiliary states are eventually used to calculate a feasible state vector, when enough information has been accumulated.

- 3) After s_{max} steps, where s_{max} is known (computed) by all nodes, all nodes calculate their new state element:

$$\hat{x}_j = g_j(\hat{z}_{j,1}[s_{max}], \hat{z}_{j,2}[s_{max}], \dots, \hat{z}_{j,m}[s_{max}]) \quad (4)$$

After this calculation is performed, it is guaranteed that:

$$\hat{x} = (\hat{x}_1[s_{max}] \hat{x}_2[s_{max}] \dots \hat{x}_N[s_{max}])^T \in \mathcal{F}. \quad (5)$$

The functions $g_j : \mathbb{R}^m \rightarrow \mathbb{R}$, as well as the required number of steps s_{max} will be defined in the sequel. Note that this procedure converges in a finite-time, namely s_{max} steps, to a feasible state vector in \mathcal{F} .

Designing the protocol requires defining the number m of needed information states per node, the initialization functions $f_{j,l}$ for these states, the matrices P , C_j , the vectors $a_j^T[k]$ and the functions g_j , as well as a procedure to calculate s_{max} in all nodes. It will be discussed in later sections that an initialization phase for the protocol will be required to calculate these quantities and that for a time-invariant graph \mathcal{G} this initialization phase will need to be executed only once.

III. FINITE - TIME ALGORITHMS FOR COOPERATION OF DER'S AND RL'S

In this section we derive the method outlined in section II to solve the coordination problem in a finite time. The proposed method is based on a protocol derived in [9], regarding the calculation of the initial state of a network of nodes using weighted linear updates of the form (3a). The main results from [9] regarding this protocol are summarized in section III-A.

A. Finite Time Algorithm for Initial State Calculation

Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of N nodes, that is connected, time-invariant and all nodes have self-loops, i.e. $\forall i \in \mathcal{V} : (i, i) \in \mathcal{E}$. Suppose that each node's information state is $z_j \in \mathbb{R}$, while $z \in \mathbb{R}^N$ is the state vector of the entire network. Suppose that for each node z_j is initialized at $z_j[0]$ and subsequently, at each step k all nodes perform weighted linear updates with their neighbors

$$z_j[k+1] = p_{jj} z_j[k] + \sum_{j \in \mathcal{N}_j^-} p_{ji} z_i[k] \quad (6)$$

where p_{ij} denotes the element in the i -th row and j -th column of the square matrix of update weights $P \in M_N$. Furthermore, at each step k , each node observes its neighbors' state and that of itself, forming an observation vector $y \in \mathbb{R}^{D_j^-}$:

$$y_j[k] = C_j z[k] \quad (7)$$

where $C_j \in M_{D_j^-, N}$ is an observation matrix.

Both P and $C_j, \forall j \in \mathcal{V}$ are randomly chosen structured matrices. Element p_{ij} of P is zero iff $(i, j) \notin \mathcal{E}$ and nonzero

¹In this paper, the set of $N \times N$ square matrices is denoted by M_N and the set of $M \times N$ matrices is denoted by $M_{M,N}$

otherwise. C_j has a single nonzero element in each of its D_j^- rows, at a column corresponding to one of the in-neighbors of node j . The nonzero elements of P and C_j are chosen randomly according to a continuous distribution, as suggested in [9]. Let $Y_{j,k} = (y_j[0] y_j[1] \dots y_j[k])^T$ denote the consolidated vector of observations up to step k for node j . If the k -step observability matrix is defined as:

$$O_{j,k} = [C_j, C_j P, \dots, C_j P^k]^T \quad (8)$$

then the following linear expression holds:

$$Y_{j,k} = O_{j,k} z[0] \quad (9)$$

The expression (9) implies that, given the consolidated observations of node j up to step k , the initial state vector z_0 lies in an affine subspace \mathcal{A} that can be written as:

$$\mathcal{A} = \bar{z}_0 + \text{Null}(O_{j,k}) \quad (10)$$

where $\bar{z}_0 \in \mathbb{R}^N$ is one solution of (9). In the case that $O_{j,k}$ is full column rank, that subspace reduces to a single point in \mathbb{R}^N and node j can uniquely determine the initial state of the entire network. The following result from [9] addresses the issue of the rank of the observability matrix:

Theorem 1: ([9]) Let \mathcal{G} be a time-invariant connected graph with self-loops in all nodes, and suppose the nodes perform linear iterations as in (6-7). Then, for almost any choice of update matrix $P \in M_N$ and observation matrices $C_j \in M_{D_j^-, N}$ (both matrices are structured - subject to the restrictions imposed by the graph \mathcal{G}), the observability matrix of each node eventually becomes full column-rank, given enough iterations are performed. Furthermore, an upper bound for the number of steps required exists, i.e:

$$\forall j \in \mathcal{V}, \exists s_j \leq N - D_j^- ; \text{rank}(O_{j,s_j}) = N \quad (11)$$

Theorem 1 implies that all nodes will be able to calculate a unique solution to (9) after $s_{max} = \max_{j \in \mathcal{V}} (N - D_j^-)$ steps of linear iterations of the form (6-7), thus determining the initial state vector of the entire network.

With Theorem 1 at hand, one can determine a distributed protocol for initial state calculation at each node. The protocol, given in [9] consists of a distributed initialization phase, performed only once and an initial state calculation phase, performed continuously. In the initialization phase the nodes randomly choose their weight matrices (e.g. according to a uniform distribution) and engage in a distributed protocol allowing each node to obtain and store s_j , O_{j,s_j} , s_{max} . The matrix O_{j,s_j} is obtained columnwise as follows: The N nodes perform N separate linear update iterations, with each node executing (6-7) for $N-2$ steps each. In each of the N iterations, the nodes are initialized differently. In the i -th iteration the initial state vector is chosen as:

$$z_j^{(i)}[0] = \begin{cases} 1 & , \text{ if } i = j \\ 0 & , \text{ otherwise} \end{cases} \quad (12)$$

Hence, the consolidated observation vector for node j is, after $N-2$ steps:

$$Y_{j,N-2}^{(i)} = [y_j^{(i)}[0], \dots, y_j^{(i)}[N-2]]^T = O_{j,N-2} e_i \quad (13)$$

As suggested by (13), in the i -th iteration, the observations of node j are exactly the i -th column of $O_{j,N-2}$. The choice for $N-2$ steps is made because Theorem 1 states that at most $N - D_j^-$ steps are needed for full observability, and thus a valid upper bound for the number of steps is $N-2$, because every node's in-degree is greater

than 1 in a connected graph. After the N protocols are completed, each node will have obtained the full observability matrix as:

$$O_{j,N-2} = [Y_{j,N-2}^{(1)} \quad Y_{j,N-2}^{(2)} \quad \dots \quad Y_{j,N-2}^{(N)}] \quad (14)$$

After obtaining $O_{j,N-2}$, each node can obtain s_j by checking the rank of its principal submatrices $O_{j,k}$, $k \leq N-2$, as:

$$s_j = \max\{k \in \mathbb{Z} : \text{rank}(O_{j,k}) = N\} \quad (15)$$

Once all nodes have calculated s_j they can converge to the number of steps needed to guarantee full observability for the entire network by performing the following iteration for $N-1$ steps, $\forall j \in \mathcal{V}$:

$$\begin{aligned} \hat{s}_{\max,j}[0] &= s_j \\ \hat{s}_{\max,j}[k+1] &= \max_{i \in \mathcal{N}_j^-} (\hat{s}_{\max,i}[k]) \end{aligned} \quad (16)$$

This iteration guarantees that all nodes will converge to s_{max} .

Once the initialization protocol outlined by (12)-(16) has been completed, all nodes have obtained s_j , O_{j,s_j} , s_{max} and are ready to engage in the initial state calculation phase.

In the initial state calculation phase, each node performs linear iterations with neighbors and observes their states as in (6)-(7). After s_j steps, node j calculates the initial value of all nodes by obtaining the unique solution of (9). The linear updates terminate after s_{max} steps, at which point all nodes have uniquely determined the initial value $z[0]$. In the interest of space, the detailed protocol for each phase is not reproduced here, but the interested reader is referred to [9]. An revised version of both the initialization phase and the second stage, designed to address the specific problem of section II is developed in later sections.

B. Result 1: Improved Protocol for Initial State Calculation

The protocol mentioned above can be used to solve the DER/RL coordination problem of section II. However, certain modifications are made to the protocols in [9] to distribute the calculations evenly across steps. To motivate these modifications, consider the following: as outlined in section III-A, after s_j steps, node j must solve the linear system:

$$O_{j,s_j} z[0] = Y_{j,s_j} \quad (17)$$

The linear system in (17) is guaranteed to have a unique solution, as per Theorem 1, since O_{j,s_j} is guaranteed to have rank N . However, solution of the linear system (17) is known to cost $\mathcal{O}(N^2 m_j)$ operations with simple gaussian elimination. Since this matrix is pre-calculated and stored locally at each node and remains constant for all subsequent operations of the network it is possible to factorize O_{j,s_j} in the initialize stage for each node, thus reducing the number of operations performed in node j at step s_j to $\mathcal{O}(N^2)$ by back substitution. Even after this modification, however, the number of required computations in the final step depends on the size of the network N , and thus it will scale badly if the protocol is applied to networks with a large number of nodes. This section, as well as the next, is dedicated to deriving a protocol that will solve the DER coordination problem and will require less computations per step.

Let $s_j D_j^- = m_j$ be the number of rows of O_{j,s_j} . From Theorem 1 O_{j,s_j} is rank N , $m_j \geq N$. Thus, the Singular Value Decomposition (SVD) of O_{j,s_j} can be written as [14]:

$$O_{j,s_j} = V \Sigma W^T \quad (18)$$

where $V \in M_{m_j, m_j}$ and $W \in M_{N, N}$ are orthogonal and $\Sigma = \begin{bmatrix} \Sigma_N \\ 0 \end{bmatrix}$, where $\Sigma_N = \text{diag} \{ (\sigma_1 \sigma_2 \dots \sigma_N)^T \}$.

Let $V = [V_L \ V_R]$ where $V_L \in M_{m_j, N}$ and $V_R \in M_{m_j, m_j - N}$. Thus, the solution of (17) can be written as:

$$z[0] = W \Sigma_N^{-1} V_L^T Y_{j, s_j} \quad (19)$$

If we define the pseudo-inverse of O_{j, s_j} as $O_{j, s_j}^+ = W \Sigma_N^{-1} V_L^T$, then (19) is written as:

$$z[0] = O_{j, s_j}^+ Y_{j, s_j} \quad (20)$$

We can write O_{j, s_j}^+ as:

$$O_{j, s_j}^+ = \begin{bmatrix} O_{j(0)}^+ & O_{j(1)}^+ & \cdots & O_{j(s_j)}^+ \end{bmatrix} \quad (21)$$

where $O_{j(k)}^+ \in M_{N, D_j^-}$, $k \in \{0, 1, \dots, s_j\}$. Using this partition of O_{j, s_j}^+ and (20)

$$z[0] = \sum_{k=0}^{s_j} O_{j(k)}^+ y_j[k] \quad (22)$$

As suggested by (22), once node j obtains the observation vector $y_j[k]$ in step k it can calculate the k -th term of the sum in the right-hand side of (22). Thus, the initial state vector can be obtained by storing an intermediate estimate $\hat{z}_0[k]$ in each step, and updating it as:

$$\hat{z}_{0,j}[k] = \hat{z}_0[k-1] + O_{j(k)}^+ y_j[k] \quad (23a)$$

$$\hat{z}_{0,j}[-1] = 0 \quad (23b)$$

Note that (22) guarantees that $\hat{z}_{0,j}[s_j] = z[0]$. This means that, since the pseudo-inverse in (21) has been pre-calculated (18) and partitioned (21) in the initialization phase, node j will need to perform $\mathcal{O}(N \cdot D_j^-)$ operations at each step of the execution phase, instead of $\mathcal{O}(N^2)$ in a single (final) step, to obtain the initial state vector for all nodes. Hence, the calculation of the initial state is now fully distributed to s_j steps.

C. Result 2: Finite Time Consensus on Sum of Initial Values

In certain cases, calculation of the entire initial state vector of the network in each node is more than what is needed. For most applications, a consensus on the average of the initial values of the nodes is required and has been extensively studied in the literature [3]. The observability-based protocol of section III-B suggests a modification of the algorithm that would allow each node to calculate the sum of the initial node values.

Let $\mathbf{1} = [1 \ 1 \ \cdots \ 1]^T$ be an N -dimensional vector. Suppose that each node is required to calculate:

$$\mathbf{1}^T z[0] = \sum_{j=1}^N z_j[0] \quad (24)$$

Of course, each node could perform this calculation by using the protocol of section III-B to calculate the entire state $z[0]$ and subsequently use (24) to obtain the required sum. However, that would require unnecessary calculations per node in each step, namely $\mathcal{O}(N \cdot D_j^-)$, which can be reduced. Indeed, by multiplying each side of (22) by $\mathbf{1}^T$, we obtain:

$$\mathbf{1}^T z_0 = \sum_{k=0}^{s_j} \mathbf{1}^T O_{j(k)}^+ y_j[k] \quad (25)$$

By defining the D_j^- -dimensional vector:

$$a_j[k] = \left(\mathbf{1}^T O_{j(k)}^+ \right)^T \quad (26)$$

then (25) can be rewritten as:

$$\sum_{j=1}^N z[0] = \sum_{k=0}^{s_j} a_j^T[k] y_j[k] \quad (27)$$

Thus, each node can obtain the exact sum of the initial node states in at most s_j steps, by performing the following operations in each step $0 \leq k \leq s_j$:

$$z_j[k+1] = p_{jj} z_j[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji} z_i[k] \quad (28a)$$

$$y_j[k] = C_j z[k] \quad (28b)$$

$$\hat{z}_j[k] = \hat{z}_j[k-1] + a_j^T[k] y_j[k] \quad (28c)$$

All three operations (28a)-(28c) cost $\mathcal{O}(D_j^-)$ computations. Hence, if all that is needed is the sum of the node initial values, the finite time algorithm requires, per step and per node, a number of computations that is linear with respect to the number of neighbors of the node, and does not depend on the size of the network, which is the desired feature we set out to achieve. The added advantage of this approach is that an upper bound (namely $N - 2$) on the number of steps needed for exact convergence is known, while the convergence time of asymptotic average consensus algorithms depends on the needed accuracy and the second largest eigenvalue of the graph Laplacian [3]. The disadvantage of this protocol, however, is the need for an initialization phase and the need to store the vectors $a_j[k]$, $k \leq s_j$ locally at each node j .

IV. DISTRIBUTED FINITE-TIME DER/RL COORDINATION

The protocol of section III-C outlines a methodology to solve the DER/RL problem of section II. The main result of this paper regarding the solution of the DER coordination problem in finite time is summarized in the following Theorem.

Theorem 2: Suppose the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is time-invariant and connected, and that the reference value X_r is known to a subset of the nodes $\mathcal{I}_0 \subseteq \mathcal{V}$. Also, each node's maximum/minimum capacities $\bar{x}_j \geq 0$ and $\underline{x}_j \leq 0$ are available to that node. Assume that the feasible set \mathcal{F} defined in (1) is non-empty. If all nodes initialize two information states and estimates as:

$$z_{j,1}[0] = \bar{x}_j - \underline{x}_j \quad (29a)$$

$$z_{j,2}[0] = \begin{cases} \frac{X_r}{|\mathcal{I}_0|} - \underline{x}_j & , \text{ if } j \in \mathcal{I}_0 \\ -\underline{x}_j & , \text{ else} \end{cases} \quad (29b)$$

$$\hat{z}_{j,l}[-1] = 0, \forall l \in \{1, 2\} \quad (29c)$$

and subsequently perform the following operations in each step, $\forall l \in \{1, 2\}$:

$$z_{j,l}[k+1] = p_{jj} z_{j,l}[k] + \sum_{i \in \mathcal{N}_j^-} p_{ji} z_{i,l}[k] \quad (30)$$

$$y_{j,l}[k] = C_j z_l[k] \quad (31)$$

$$\hat{z}_{j,l}[k] = \begin{cases} \hat{z}_{j,l}[k-1] + a_j^T[k] y_{j,l}[k] & \text{if } k \leq s_j \\ \hat{z}_{j,l}[k-1] & \text{otherwise} \end{cases} \quad (32)$$

if we define:

$$\hat{x}_j[k] = \underline{x}_j + \hat{z}_{j,2}[k] \frac{\bar{x}_j - \underline{x}_j}{\hat{z}_{j,1}[k]} \quad (33a)$$

$$\hat{x}[k] = [\hat{x}_1[k] \ \cdots \ \hat{x}_N[k]]^T \quad (33b)$$

Then, for almost any random choice of P and C_j and with a_j^T chosen as in (26) the following holds:

$$\forall j \in \mathcal{V}, \exists s_j \leq N - D_j^- \text{ such that } \hat{z}_{j,l}[s_j] = \sum_{i=1}^N z_{j,l}[0] \quad (34)$$

Furthermore, if we define:

$$D_{max} = \max_{j \in \mathcal{V}} \{D_j^-\} \quad (35)$$

then:

$$\exists s_{max} = \max_{j \in \mathcal{V}} \{s_j\} \leq N - D_{max} : \hat{x}[s_{max}] \in \mathcal{F} \quad (36)$$

where \mathcal{F} is defined in (1).

Proof: From theorem 1, since \mathcal{G} is connected and time invariant:

$$\forall j \in \mathcal{V}, \exists s_j \leq D_j^- : \text{rank}(O_{j,s_j}) = N$$

Hence $\forall l \in \{1, 2\}$, the linear system:

$$O_{j,s_j} z_l[0] = [y_{j,l}[0], \dots, y_{j,l}[s_j]]^T$$

has a unique solution, which can be obtained by:

$$z_l[0] = \sum_{k=0}^{s_j} O_{j(k)}^+ y_{j,l}[k] \quad (37)$$

where $O_{j(k)}^+$ are defined in (21). By multiplying both sides of (37) with $\mathbf{1}^T$, we obtain:

$$\sum_{j=1}^N z_{j,l}[0] = \sum_{k=0}^{s_j} \mathbf{1}^T O_{j(k)}^+ y_{j,l}[k] = \sum_{k=0}^{s_j} a_j^T[k] y[k] = \hat{z}_{j,l}[s_j] \quad (38)$$

This concludes the proof of the first claim.

Combining (38) and the state initialization (29a)-(29b) we obtain $\forall j \in \mathcal{V}$:

$$\hat{z}_{j,1}[s_j] = \sum_{i=1}^N z_{i,1}[0] = \sum_{i=1}^N \bar{x}_i - \underline{x}_i \quad (39)$$

$$\hat{z}_{j,2}[s_j] = \sum_{i=1}^N z_{i,2}[0] = \sum_{i \in \mathcal{I}_0} \frac{X_r}{|\mathcal{I}_0|} + \sum_{i=1}^N \underline{x}_i = X_r + \sum_{i=1}^N \underline{x}_i \quad (40)$$

Hence using (33a), each node $j \in \mathcal{V}$ converges to the following solution in s_j steps:

$$\hat{x}_j[s_j] = \underline{x}_j + \left(X_r - \sum_{i=1}^N \underline{x}_i \right) \frac{\bar{x}_j - \underline{x}_j}{\sum_{i=1}^N \bar{x}_i - \underline{x}_i} \quad (41)$$

By the definition of s_{max} and (32), we have $\forall j \in \mathcal{V}$:

$$\hat{x}_j[s_{max}] = \hat{x}_j[s_j] \quad (42)$$

All that is left is to prove is that $\hat{x}[s_{max}] \in \mathcal{F}$. Suppose that $\hat{x}[s_{max}] \notin \mathcal{F}$. It is obvious that $\sum_{j=1}^N \hat{x}_j[s_{max}] = X_r$. Hence, at least one of the inequality constraints defining \mathcal{F} in (1) is violated:

$$\exists j_0 \in \mathcal{V} : \hat{x}_{j_0}[s_{max}] > \bar{x}_{j_0} \text{ or } \hat{x}_{j_0}[s_{max}] < \underline{x}_{j_0}$$

Which in turn implies, by (41),(42) for $j = j_0$:

$$X_r > \sum_{i=1}^N \bar{x}_i \text{ or } X_r < \sum_{i=1}^N \underline{x}_i \quad (43)$$

It is easy to verify that, if either of the conditions in (43) holds, \mathcal{F} is empty. However, \mathcal{F} is nonempty by assumption. Thus $\hat{x}[s_{max}] \in \mathcal{F}$ by contradiction. ■

Theorem 2 proves that the outlined method can solve the DER coordination problem in a finite number of steps (s_{max}) given that a feasible solution exists. As such, it outlines a procedure to solve the DER coordination problem in a distributed fashion, given that each node j has locally stored its update weights, the matrix C_j , as well as the vectors $a_j[k]$ and the values s_j and s_{max} . For this purpose, the proposed method includes an initialization phase, executed only once, during which all these quantities are locally computed at each node. In the protocol for the initialization phase, each node $j \in \mathcal{V}$ chooses its weights for the linear updates and its observation matrix C_j , calculates in a distributed manner its observability matrix O_{j,s_j} as well as the number of steps s_j needed to guarantee a full rank observability matrix. Subsequently all nodes agree on what is the maximum of s_j , denoted by s_{max} . After s_{max} steps, each node proceeds to calculate the partitioned pseudo-inverse as in section III-B, and the vectors $a_j[k]$ defined in section III-C. In the distributed coordination phase, each node performs the operations outlined in Theorem 2. After having performed s_{max} steps, each node re-initializes its states as in (29a)-(29b), and repeats the protocol. This allows tracking of a reference X_r that varies with time and/or changing available capacities of the nodes. As per Theorem 2, this protocol is guaranteed to provide a feasible solution in at most $N - D_{max}$ steps where D_{max} is defined in (35).

V. SIMULATION RESULTS

In this section an example application is presented: coordination of reactive power amongst PLL-synchronized inverters. Suppose a set \mathcal{V} of utility interfaced PV systems, fed by DC/AC inverters are connected to a certain bus in the system. The inverters are synchronized to grid voltage via Phase Lock Loops (PLL's). Each inverter's active power P_j is controlled by its Maximum Power Point Tracker (MPPT) and is considered an exogenous variable. Thus, the inverters can be considered as having controllable reactive power contribution Q_j , within certain boundaries. There are two factors that constrain an inverter's capacity for reactive power support: i. the inverter's rated power constraint $\bar{Q}_{j,1} = [S_{(r),j}^2 - P_j^2]^{\frac{1}{2}}$ and ii. utility-imposed constraints on the inverter's power factor $\bar{Q}_{j,2} = P_j/\bar{\lambda} \cdot [1 - (\bar{\lambda})^2]^{\frac{1}{2}}$. Hence, in the context of our DER/RL coordination protocol, the actual reactive power bound is the tightest of the two bounds. Suppose a coordinating controller generates a request for reactive power $Q^*(t)$ support of the grid from the group of inverters, but can only communicate with a subset $\mathcal{I}_0 \subseteq \mathcal{V}$. In the context of the coordination problem of section II, this constitutes to specifying a reactive power reference $Q_j^{(ref)}$ for each inverter $j \in \mathcal{V}$ that is within its capacity limits. A case of a network of 27 inverters was examined to demonstrate the approach. A coordinating controller monitors the difference between the bus voltage and the reference and generates a reference reactive power Q^* for the network of inverters to track. Only the first inverter receives Q^* from the coordinator ($\mathcal{I}_0 = \{1\}$). All inverters have been initialized using the initialization protocol introduced in Section III. Fig. 1 shows, for various nodes, the step-by-step evolution of the proposed distributed coordination algorithm for a reactive power request of $Q^* = 10 pu$ by the coordinating controller, i.e. $\hat{x}_j[k]$ is plotted. For comparison purposes, the result is plotted against the asymptotically converging consensus based algorithm of [4], that converges to the same value per node. Note that all nodes converge to an allocation of Q^* that is proportional to their maximum capacity. Also, it is worth noting that the proposed algorithm converges in a finite number of steps, that is indeed bounded above by $N - 2 = 25$ steps, as expected. It can be observed

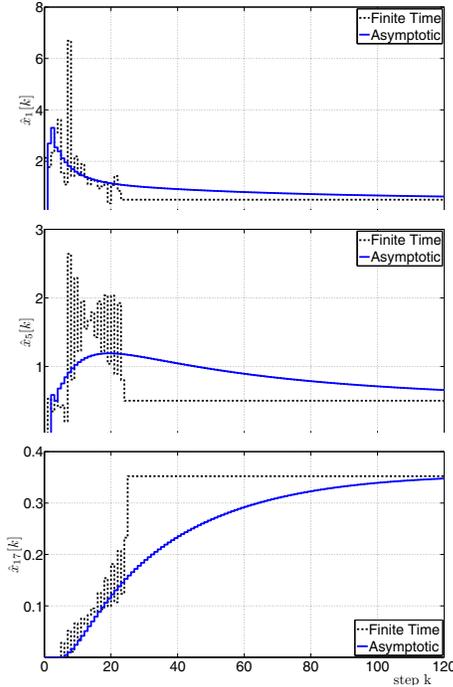


Fig. 1. Step-by-step execution of the inverter reactive power coordination algorithm: (a) Node 5 (b) Node 17 (c) Node 1

that the proposed protocol does not converge to the steady state solution as smoothly as the asymptotically converging consensus protocol. This however should not be an issue in practice, because the inverters will update their reactive power contribution only after the protocol converges - and each of them knows an upper bound s_{max} of the number of steps needed for that to happen. In other words, the transient behavior of the protocol in Fig. 1 is disregarded by the inverters, and each inverter only uses the final value after $s_{max} = 25$ steps. The capability of the proposed protocol to allocate the reactive power command Q^* among the inverters is shown in Fig. 2. This figure illustrates the step-by-step evolution of the sum of the intermediate estimates of the feasible solution $\hat{x}_j[k]$, saturated by the upper and lower capacity limits of each inverter: $\hat{Q}[k] = \sum_{j=1}^N \left\{ \max \left[Q_j, \min \left(\bar{Q}_j, \hat{x}_j[k] \right) \right] \right\}$. The quantity $\hat{Q}[k]$ is compared to the same quantity when the asymptotic consensus algorithm in [4] is used. This quantity represents the allocation of the reactive power command among inverters as the protocol is executed, while considering their capacity constraints. As such it constitutes a good indicator of protocol performance. $\hat{Q}[k]$ converges to the requested reference $Q^* = 10$ for both protocols. As expected, it converges to the reference in a finite 25 steps for the proposed finite-time protocol.

VI. CONCLUSIONS AND FUTURE WORK

In this work, a finite-time decentralized protocol, suitable for the coordination of controllable resources in the smart grid, was introduced. The protocol is based on linear updates with neighbors, and the concept of observability in linear structured systems and connected directed graphs. It is guaranteed to converge in finite time, namely in no more than $N - D_{max}$ steps. The protocol was eventually formulated in such a way that the operations per step and per node are linear with respect to the number of the node's in-neighbors. Following this protocol, each node converges to an

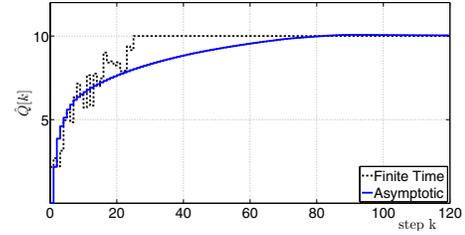


Fig. 2. Step-by-step evolution of the quantity $\hat{Q}[k]$: finite time versus asymptotic protocol

allocation of a reference value that is proportional to its capacity. An example network of inverters, participating in voltage support of a single bus was studied. The performance of the algorithm, in terms of steps needed for convergence, was compared to the asymptotic consensus case and was shown to be faster.

REFERENCES

- [1] Joos, G.; Ooi, B. -T; Mcgillis, D.; Galiana, F.D.; Marceau, R., The potential of distributed generation to provide ancillary services, *Power Engineering Society Summer Meeting, 2000. IEEE*, vol.3, pp.1762,1767 vol. 3, 2000
- [2] Lasseter, R.H., MicroGrids, *Power Engineering Society Winter Meeting, 2002. IEEE*, vol.1, pp.305,308 vol.1, 2002
- [3] Olfati-Saber, R.; Fax, J.A.; Murray, R.M., Consensus and Cooperation in Networked Multi-Agent Systems, *Proceedings of the IEEE*, vol.95, no.1, pp.215,233, Jan. 2007
- [4] Dominguez-Garcia, A.D.; Hadjicostis, C.N., Coordination and Control of Distributed Energy Resources for Provision of Ancillary Services, *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp.537,542, 4-6 Oct. 2010
- [5] Hadjicostis, C.N.; Charalambous, T., Asynchronous coordination of distributed energy resources for the provisioning of ancillary services, *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pp.1500,1507, 28-30 Sept. 2011
- [6] Dimeas, A.L.; Hatzigiargyriou, N.D., Agent based control of Virtual Power Plants, *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, pp.1,6, 5-8 Nov. 2007
- [7] Robbins, B.A.; Dominguez-Garcia, A.D.; Hadjicostis, C.N., Control of distributed energy resources for reactive power support, *North American Power Symposium (NAPS), 2011*, pp.1,5, 4-6 Aug. 2011
- [8] Baran, M.E.; El-Markabi, I.M., A Multiagent-Based Dispatching Scheme for Distributed Generators for Voltage Support on Distribution Feeders, *Power Systems, IEEE Transactions on*, vol.22, no.1, pp.52,59, Feb. 2007
- [9] Sundaram, S.; Hadjicostis, C.N., Distributed function calculation and consensus using linear iterative strategies, *Selected Areas in Communications, IEEE Journal on*, vol.26, no.4, pp.650,660, May 2008
- [10] Masters, C. L., Voltage rise: the big issue when connecting embedded generation to long 11 kV overhead lines, *Power Engineering Journal*, vol.16, no.1, pp.5,12, Feb. 2002
- [11] Senjyu, T.; Miyazato, Y.; Yona, A.; Urasaki, N.; Funabashi, T., Optimal Distribution Voltage Control and Coordination With Distributed Generation, *Power Delivery, IEEE Transactions on*, vol.23, no.2, pp.1236,1242, April 2008
- [12] Carvalho, P. M S; Correia, Pedro F; Ferreira, L.A.F., Distributed Reactive Power Generation Control for Voltage Rise Mitigation in Distribution Networks, *Power Systems, IEEE Transactions on*, vol.23, no.2, pp.766,772, May 2008
- [13] Robbins, B.A.; Hadjicostis, C.N.; Dominguez-Garcia, A.D., 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
- [14] R. Horn and C. Johnson, *Matrix Analysis*. New York, NY: Cambridge University Press, 1985
- [15] Wang, H.F., Multi-agent co-ordination for the secondary voltage control in power-system contingencies, *Generation, Transmission and Distribution, IEE Proceedings-*, vol.148, no.1, pp.61,66, Jan 2001