Distributed Generation Control of Small-Footprint Power Systems

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

***Abstract*—This paper presents a strategy for controlling generators in a small-footprint power system without the need for a centralized decision maker. In particular, we discuss an iterative algorithm which is utilized to coordinate a set of generators such that they collectively operate according to some predetermined criterion while accounting for individual capacity constraints. We implement the algorithm using a hardware testbed comprised of low-complexity devices capable of performing simple computations and communicating wirelessly with other nearby devices. The hardware testbed is then used to control the generation output of the synchronous generators in a six-bus, small-footprint power system. Experimental results are presented for several load changes as well as for a case in which a spinning reserve is added to the system.**

I. Introduction

Motivated by the pursuit of increased efficiency and reliability, several applications which have traditionally had relatively modest electric power requirements are undergoing radical transformations. Combined with advancements in technology, this motivation has led to an increase in the prevalence of small-footprint power systems in applications such as more electric aircraft [1], ships with electric propulsion [2] and microgrids [3]. Compared to large power systems such as the utility grid, small-footprint power systems generally have less capacity and encompass a smaller area. Despite these differences, the control objectives for small-footprint power systems are similar; generators must be coordinated to maximize overall efficiency, generation must be constantly balanced with demand, and, in the case of ac systems, the frequency must be regulated.

Large-scale power systems are vast, often containing thousands of generators, and thus require complex control mechanisms to govern their operation. Typically, the generators in large power systems are controlled using a three-layered architecture. At the lowest layer, each generator has a governor, commonly referred to as a droop controller, which allows instantaneous load-generation balancing at the cost of system-wide frequency deviations. The middle layer, called automatic generation control (AGC), which operates on a slower time scale, serves to regulate the frequency to a specified nominal value, maintain the correct power interchange between control areas, and operate each generator in the most economical way. The third layer, called economic dispatch, optimizes the control of the generators according to their operating costs and is executed every few minutes [4].

Among the three layers of the aforementioned control architecture, only the lowest layer operates without an external command; thus, a decision maker is required to coordinate the generators to realize the middle and top layers. Traditionally, a centralized decision maker is used which requires complete knowledge of the devices in the system as well as a communication link from each device to the controller. An alternative approach is to coordinate the generators in a distributed fashion, relying only on communication links between local generator controllers and computations performed by each local controller. Compared to the centralized strategy, the distributed approach can more easily adapt to changes in the system, allowing it to be more resilient to failures and requiring less planning to begin operation. Using the latter control architecture to control generators in a small-footprint power system through a process we call *distributed generation control (DGC)* is the primary contribution of this paper. Specifically, we envision DGC replacing the functionality of AGC by accounting for some of the unique characteristics of small-footprint power systems.

The objective of DGC is to utilize a communication network linking a set of generators to distributively determine the output of each generator such that they collectively operate according to some predetermined criterion, e.g. at an electrical frequency of 60 Hz. It is assumed that the generation mismatch in the system is known by a leader which can only communicate with a limited number of other generators in the system and may not necessarily be aware of the total number of generation units available. The leader is responsible for initializing the execution of DGC; however, a leading generator is not required as the initialization procedure could be adjusted so that any generator could begin the execution of DGC. To address individual capacity constraints, DGC accounts for upper and lower generation limits by fairly splitting the overall demand for generation based upon the percentage loading of the system.

To demonstrate DGC, this paper presents results from an experimental setup consisting of small synchronous generators, each of which is outfitted with a wireless transceiver to create a communication network that can be thought of as a stationary, yet unplanned, ad-hoc network. An iterative algorithm is used to exchange information among the generators such that they collectively determine how to distribute the generation to meet the overall demand. At the end of the iterative process, the output of each generator is computed based upon the result of the algorithm and the capacity constraints of the respective

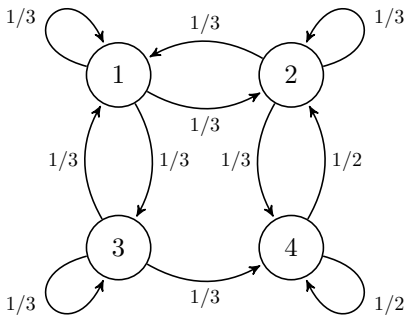

Fig. 1. Graph of 4-node network

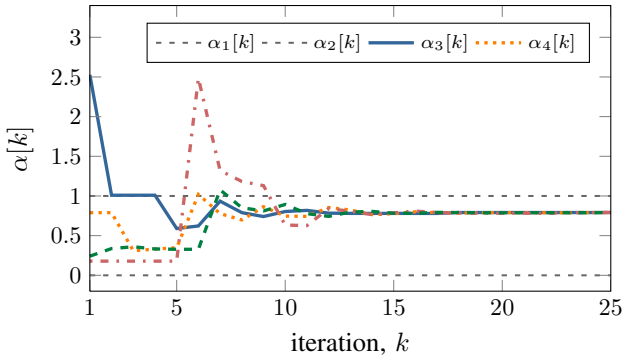

Fig. 2. Example evolution of the ratio-consensus algorithm

generator.

The remainder of this paper is organized as follows. Section II presents a graph-theoretic model used to describe the exchange of information between generators and formulates a distributed iterative algorithm that is the basis for DGC. Section III introduces a two-stage control architecture which combines the distributed algorithm and a proportional controller. Section IV provides a brief overview of a communication and computation testbed created to implement the distributed algorithm. Section V describes the small-footprint power system used for experimentation and presents results illustrating the ability of DGC to regulate the electrical frequency in the system. Section VI presents some concluding remarks.

II. Preliminaries

In this section, we develop a graph-theoretic model to represent the communication network linking generators. Using the network model, we introduce a distributed algorithm used to realize DGC and give a brief overview of its operation.

A. Communication Model

Let $\mathcal{G}$ be a directed graph describing the communication network linking a set of generators that are outfitted with wireless transceivers capable of exchanging packetized information. Define $\mathcal{V} := V(\mathcal{G})$ to be the set of vertices with each vertex corresponding to a generator and $\mathcal{E} := E(\mathcal{G})$ to be the set of directed edges with each edge corresponding to a communication link between a pair of generators. The exchange of information between two generators i and j need not be bidirectional; thus, the ordered pair $(i,j) \in \mathcal{E}$ only if generator i can receive information from generator j. For each $i \in \mathcal{V}$, we define the set of generators from which i can receive information to be the in-neighborhood of i, denoted, $\mathcal{N}_i^- := \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. Similarly, we define the out-neighborhood of i to be the set of generators that can receive information from i, i.e., $\mathcal{N}_i^+ := \{j \in \mathcal{V} : (j,i) \in \mathcal{E}\}$, and we denote the cardinality of the out-neighborhood by $\mathcal{D}_i^+ := |\mathcal{N}_i^+|$. All vertices are allowed self loops, i.e., $(i,i) \in \mathcal{E},\ \forall i \in \mathcal{V}$; thus, each generator is included in both its own in- and out-neighborhood. For the algorithm discussed next, it is assumed that the graph $\mathcal{G}$ is strongly connected; that is, for each ordered pair of vertices i, j there is a path from i to j [5].

B. The Ratio-Consensus Algorithm

Our basis for implementing DGC is a class of distributed algorithms originally proposed in [6] and [7]. In particular, we utilize an iterative algorithm in which each generator runs two linear iterations in parallel, initialized according to its minimum and maximum output limits, which we refer to as *the ratio-consensus algorithm*.

Consider a set of n generators described by the aforementioned communication model, i.e., $|\mathcal{V}| = n$, and assume that there exists one leader that knows the generation mismatch, ρ_e, which should be added to or removed from the system in order to operate according to some predetermined criterion. Let x_i be the output of generator i and define $\rho := \sum_{i=1}^{n} x_i$ to be the total output provided by the set of generators. Furthermore, define $\rho_d := \rho + \rho_e$ to be the total demand for generation and $l := \mathcal{D}_{leader}^+$ to be the out-degree of the leading DGR, with $l \geq 2$ since $\mathcal{G}$ is strongly connected.

Each generator i participating in the distributed algorithm will maintain two state variables, y_i and z_i, which will be updated at each iteration to be a linear combination of its previous state and the previous states of the other generators in its in-neighborhood. Let $k = 0, 1, \ldots$ index the iterations, then each generator i updates the value of its state variables at each iteration k as

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

$$z_i[k+1] = \sum_{j \in \mathcal{N}_i^-} \frac{1}{\mathcal{D}_j^+} z_j[k]. \tag{2}$$

Let the minimum and maximum output of generator i be denoted $\underline{x}_i$ and $\overline{x}_i$, respectively. Then, the initial value of (1) is $y_i[0] = \rho_e + x_i - \underline{x}_i$ if i is the leader and $y_i[0] = x_i - \underline{x}_i$ otherwise. Similarly, (2) is initialized to be $z_i[0] = \overline{x}_i - \underline{x}_i$, $\forall i \in \mathcal{V}$. After performing m iterations, where m is large enough to converge to a sufficiently accurate solution, each generator i computes its new output to be

$$x_i = \underline{x}_i + \alpha_i[m](\overline{x}_i - \underline{x}_i), \tag{3}$$

where $\alpha_i[m]$ is defined as

$$\alpha_i[m] := \frac{y_i[m]}{z_i[m]}, \tag{4}$$

and it can be shown that $\alpha_i[m] \cong \alpha_j[m]$, $\forall i, j \in \mathcal{V}$ [7]. Consequently, after the algorithm has converged, each generator i can independently determine if the collective capacity of the system is sufficient to meet the demand, i.e., if $\sum \underline{x}_i \leq \rho_d \leq \sum \overline{x}_i$, by evaluating the value of $\alpha_i[m]$. Given the initial conditions of (1) and (2), if $0 \leq \alpha_i[m] \leq 1$, the demand for generation is within the collective generator constraints. Thus, $\alpha_i[m]$ indicates the percent loading of the generators relative to their limits. For a more in-depth description of the algorithm and a proof of convergence, see [7].

III. Two-Stage Control Architecture

In this section, we use a simple generator model to evaluate the steady-state characteristics of a synchronous machine and to formulate a proportional controller which determines the generation mismatch given some error in the frequency. We then develop a two-stage control architecture which combines the proportional controller with the ratio-consensus algorithm to regulate the electrical frequency in a small-footprint power system.

A. Generator Model

In order to develop a suitable control strategy, we use a two-state model—the so-called *classical model* (see, e.g., [8])—to describe the dynamics of the synchronous generators and to evaluate their steady-state behavior following a transient such as an increase or decrease of the electrical load. For each synchronous machine, i, let δ_i denote the angle of the rotor (with respect to the synchronous reference rotating at ω_s [rad/s]) in electrical radians, ω_i denote the angular velocity of the rotor in electrical radians per second and P_i^m [pu] denote the power supplied by the prime mover. Furthermore, let V_i [pu] and θ_i [rad] denote the magnitude and angle of the machine terminal voltage, respectively, X_i [pu] denote the internal machine reactance, and E_i [pu] denote the magnitude of the internal machine voltage. Then the machine dynamics can be described by

$$\frac{d\delta_i}{dt} = \omega_i - \omega_s, \tag{5}$$

$$\frac{d\omega_i}{dt} = \frac{1}{M_i}P_i^m - \frac{D_i}{M_i}(\omega_i - \omega_s) - \frac{E_i V_i}{X_i M_i}\sin(\delta_i - \theta_i), \tag{6}$$

where D_i [rad/s] is the damping coefficient of the spinning mass, M_i [s^2/rad] is the scaled inertia constant of the machine, and ω_s is the synchronous speed of the machine in electrical radians per second.

Upon inspecting (6), we see that each of the three summands can be attributed to distinct sources. In particular, the first term is the power supplied by the prime mover while the second term is the power lost in the rotating mass due to friction and windage. The final term is the electrical power extracted from the terminals of the generator.

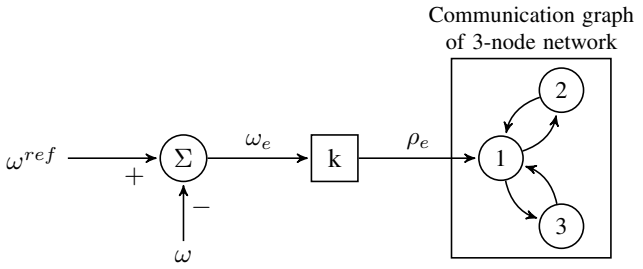

Fig. 3. Two-stage control architecture

Using (6), the steady-state behavior of the generator following a transient such as an increase or decrease in the electrical load can be evaluated. In steady-state, the derivative of (6) will be zero; thus, the speed is given as

$$\omega_i = \frac{1}{D_i}\left(P_i^m - \frac{E_i V_i}{X_i}\sin(\delta_i - \theta_i)\right) + \omega_s. \tag{7}$$

Immediately following a load change (or without the addition of a governor), the power supplied by the prime mover, P_m, remains constant. Thus by inspecting (7), we see that, in order for energy to be conserved after a load change, the speed of the rotating mass must change. Specifically, an increase in load will act to slow the system down while acts to increase the speed. The amount by which the speed changes is governed by the inverse of the damping coefficient, D_i, causing the synchronous machine to exhibit a natural drooping effect. It should be noted that in large-scale power systems, an external controller is added to each synchronous generator to provide the above-described droop characteristic. In out setup, however, the damping coefficient is sufficient; thus, an external droop controller is not used.

B. Two-Stage Control Architecture

The inherent droop characteristic of the synchronous machines provides a natural way for determining the amount of generation needed to be added to or removed from the system in order to operate at a specified electrical frequency. Thus the leading generator need only measure the frequency error in the system and multiply it by a gain to determine the generation mismatch

$$\rho_e = k(\omega^{ref} - \omega), \tag{8}$$

where k has units pu-s/rad. In combination with the ratio-consensus algorithm, the computation of the generation mismatch by the leading generator yields a two-stage closed loop controller as shown in Fig. 3. As the figure illustrates, the frequency error is used to determine the generation mismatch which is then dispersed among all generators by the communication network and allocated using the ratio-consensus algorithm.

As mentioned above, without a governor, adjusting the electrical load in the system will lead to a change in the mechanical speed of the generators and thus a change in the electrical frequency. Since the spinning mass of the generators and their associated prime movers have nonzero inertia, it will

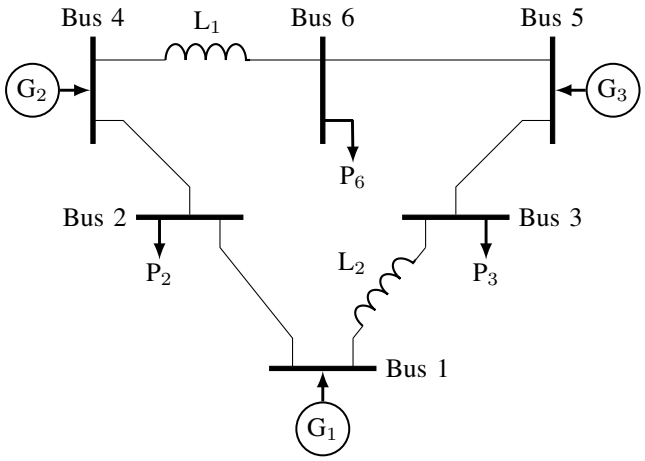

Fig. 4. One line diagram of small-footprint power system

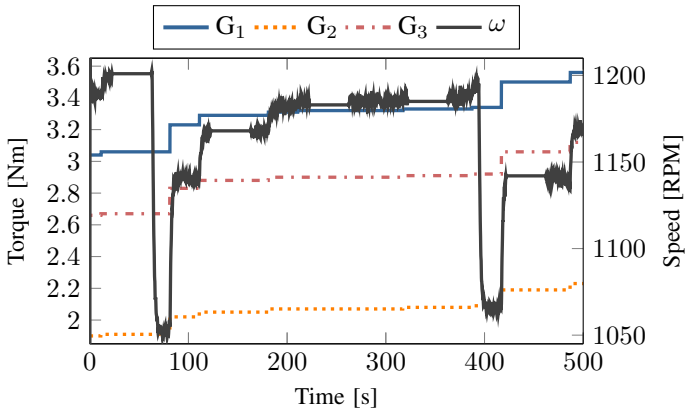

Fig. 5. Prime mover torque and average speed during load increase

take a small amount of time for the new steady-state speed to be reached (assuming a steady-state solution exists). Furthermore, each execution of the ratio-consensus algorithm requires time to reach convergence. Thus, the leading generator is restricted from initializing the two-stage control architecture when the change in rotor speed is nonzero. Once the leader has determined that the mechanical speed is constant, however, it computes the generation error using (8) and broadcasts a packet to schedule the ratio-consensus algorithm.

IV. Communication and Computation Testbed

To enable a set of generators to be coordinated using the ratio-consensus algorithm discussed in Section II, a hardware testbed comprised of nodes with embedded processors capable of wirelessly exchanging information with other nearby nodes is developed. In this section, we provide a brief overview of the hardware chosen and the software created to implement the testbed as well as an example of the evolution of the algorithm executed on the testbed. For a more thorough description of the hardware and software, see [9] and [10].

A. Node Hardware

Each node in the testbed consists of an Arduino Mega 2560 microcontroller board connected to a MaxStream XB24-DMCIT-250 revB XBee module via a SparkFun Electronics XBee shield. The Arduino Mega contains an AVR ATmega2560 8-bit microcontroller and was chosen for the wealth of available software libraries as well as for its flexibility and ease of use. The XBee is an embedded RF module operating at 2.4 GHz that utilizes a built-in chip antenna and enables the nodes to wirelessly exchange information. To be useful in various applications, the testbed hardware was chosen so that it can operate independently of the devices it is controlling. Furthermore, the Arduino Mega has several serial ports which can be used to communicate with other devices. Specifically, for the experimental results presented in Section V, commands are sent to the generators via a USB port.

B. Software Setup

To facilitate the exchange of values for the distributed algorithm, a three layer protocol stack is created. The lowest layer is based on the ZigBee protocol, the middle layer is a modified version of the xbee-arduino API, and the top layer is specially designed to transport the values needed to execute the distributed algorithm. Furthermore, all packets are broadcasted to take advantage of the wireless medium and, in order to minimize network traffic, no acknowledgements are sent upon successful receipt of packets. To create a partially connected network despite the close proximity of the nodes during testing, each node is programmed to only accept messages received from nodes in its in-neighborhood. In a more realistic setup, however, the testbed could be adapted to allow the availability of links between nodes to be based upon signal strength.

In the discussion of the ratio-consensus algorithm, we assumed that all participating generators update the value of their state variables in unison; i.e., generator i updates its states at iteration k at the same time generator j updates its states, $\forall i, j \in \mathcal{V}$. Without a common time reference and with no acknowledgements, however, it is possible for the generators to update their states at different times, which could cause the generators to converge to the wrong solution or possibly diverge. Thus, to ensure convergence to the correct solution, all nodes are synchronized to a common reference before initializing the distributed algorithm using a procedure based upon the hierarchy referencing time synchronization (HRTS) protocol proposed in [11]. Furthermore, to maintain synchronism throughout the iterative process, the number of iterations and the period of each iteration are known by all nodes *a priori*.

It should be noted that the actual algorithm implemented on the hardware testbed is slightly different from the ratio-consensus algorithm described in (1) and (2). In an uncontrolled environment, conditions such as temperature and humidity as well as obstructions between generators can negatively affect link availability. Thus, a modified algorithm based on the one proposed in [12] is used in the testbed which is resilient to temporary link failures.

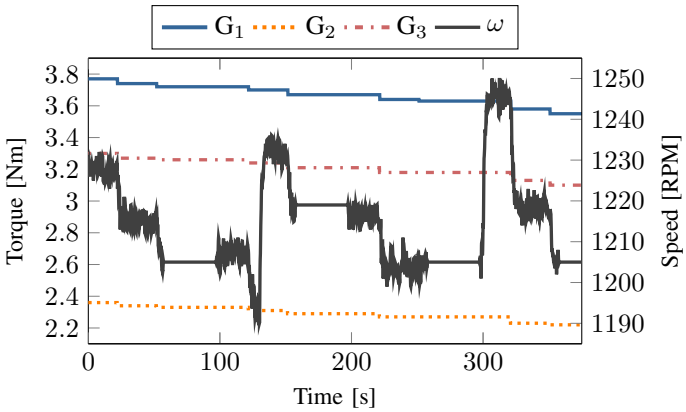

Fig. 6. Prime mover torque and average speed during load decrease

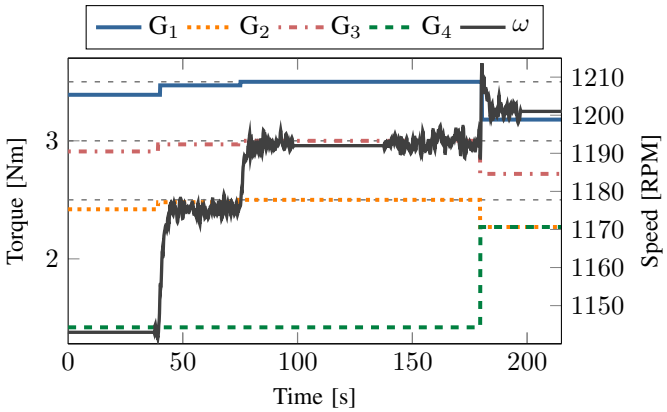

Fig. 7. Prime mover torque and average speed during reserve switch-in

C. Communication and Computation Experimental Results

Using the communication and computation testbed, the 4-node network depicted by the graph in Fig. 1 is created to demonstrate the evolution of the ratio-consensus algorithm. In this example, node 1 is the chosen to be the leader and the generation mismatch is $\rho_e = 0.25$. The initial output of each node is $x_1 = 0.25$, $x_2 = 0.25$, $x_3 = 0.1$ and $x_4 = 0.15$; thus the total demand for generation is $\rho_d = 1$. Furthermore, the upper and lower capacity constraints are

$$\underline{x} = \left[0.02, 0.1, 0.05, 0.08\right]^{\mathrm{T}},$$
$$\overline{x} = \left[0.21, 0.29, 0.33, 0.37\right]^{\mathrm{T}},$$

respectively, such that the collective capacity of the nodes is sufficient to meet the demand, i.e., $\sum \underline{x}_i = 0.25 \leq \rho_d \leq \sum \overline{x}_i = 1.2$.

The evolution of the value of $\alpha_i[k]$ as computed by the nodes according to (4) is shown in Fig. 2. From the figure, it can be seen that after approximately 20 iterations, the nodes converge to the final solution in which $\alpha_i[20] = 0.789$. Using this solution, the nodes can compute the new outputs according to (3) as $x = \left[0.17, 0.25, 0.27, 0.31\right]^{\mathrm{T}}$. Summing the individual results, we see that the total output equals the total demand for generation, i.e., $\rho = \rho_d = 1$. In this example, an iteration period of 50 milliseconds was chosen; thus, the nodes converge to the final solution in approximately 1 second.

V. Small-Footprint Power System Experimental Results

In this section, we describe the components and interconnections of a small-footprint power system that is used to demonstrate the efficacy of DGC. In particular, we use DGC to regulate the electrical frequency in the system to 60 Hz following a load change. We then present experimental results demonstrating the communication and computation testbed controlling the synchronous generators in the small-footprint power system. The results include cases in which the load is increased and decreased as well a case in which a fourth generator is added to the system to act as a spinning reserve. In all of the cases, the controller gain used by the leader is $k = 0.003$ Nm/RPM. The graph depicted in Fig. 3 represents the exchange of information between the generators in the first two cases while the graph in Fig. 1 represents the exchange of information for the third case.

TABLE I
Value of Added Inductances in Power System

Parameter	Inductance [mH]	Parameter	Inductance [mH]
$L_{1,A}$	2.041	$L_{2,A}$	4.175
$L_{1,B}$	1.905	$L_{2,B}$	4.162
$L_{1,C}$	1.961	$L_{2,C}$	4.059

A. Power System Setup

The small-footprint power system used for experimentation is a 240 V, 3-phase system consisting of six buses as shown in Fig. 4. The system is comprised of 3 Hampden Engineering synchronous machines, G_1, G_2, G_3, and 3 wye-connected resistive loads, labeled as P_2, P_3 and P_6. Each synchronous machine is connected at the shaft to a permanent magnet synchronous servomotor which serves as the prime mover. In order to regulate the frequency as the load in the system varies, the prime movers are operated in constant torque mode, with the torque command supplied by the nodes from the hardware testbed. The synchronous machines utilized in the power system have 3 pole-pairs, thus, to maintain an electrical frequency of 60 Hz, the mechanical speed of the generators is regulated to 1200 RPM.

In the results presented below, the generator at bus 1, G_1, is chosen to be the leader. Furthermore, the per-phase resistance of each load is adjusted by adding 500 Ω resistors in parallel. Each load can have up to 10 resistors in parallel per phase, yielding resistances in the range $500, 250, \ldots, 50\,\Omega$. Extra impedance is added via series inductors between bus 4 and bus 6 as well as between bus 1 and bus 3 as shown in Fig. 4. The per-phase inductances are given in Table I.

B. Load Increase

Figure 5 shows the torque supplied by the prime movers and the average mechanical speed of the generators as the electrical load is increased. In this case, the maximum constraints of the generators are chosen to be $\overline{x}_1 = 4.0$ Nm, $\overline{x}_2 = 2.5$

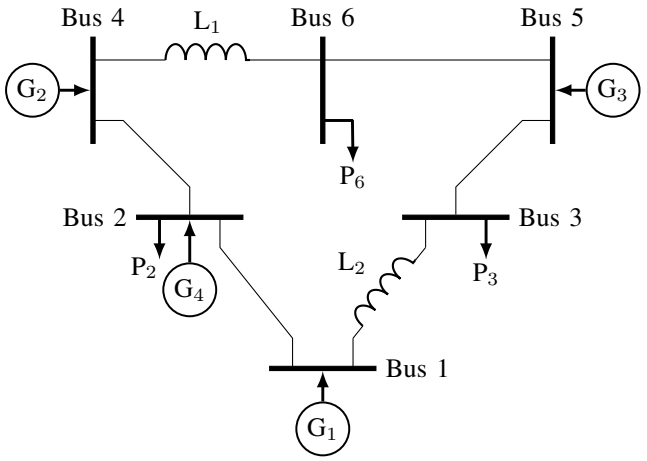

Fig. 8. One line diagram of modified small-footprint power system

Nm, and $\overline{x}_3 = 3.5$ Nm, while the minimum constraints are $\underline{x}_i = 0$ Nm, $i = 1, 2, 3$. Initially, the load in the system is $R_2 = 250\,\Omega$, $R_3 = 166.67\,\Omega$, and $R_6 = 166.67\,\Omega$. After 500 seconds, the resistances R_2 and R_3 have decreased to $166.67\,\Omega$ and $125\,\Omega$, respectively, while R_6 remained the same. As the figure illustrates, each generator increases its output such that they collectively drive the generation mismatch to zero and return the speed to approximately 1200 RPM.

C. Load Decrease

Using the same maximum and minimum constraints as in the previous case, the plot in Fig. 6 shows the torque supplied by the prime movers and the average mechanical speed of the generators as the electrical load is decreased. The initial resistances of the loads are $R_2 = 125\,\Omega$, $R_3 = 166.67\,\Omega$, and $R_6 = 83.33\,\Omega$. After approximately 350 seconds, the resistances R_3 and R_6 have increased to $166.67\,\Omega$ and $500\,\Omega$, respectively, while R_2 remained unchanged. As the figure shows, the output of each generator is decreased as the load is shed and the mechanical speed increases. As in the previous case, DGC serves to drive the generation mismatch toward zero and approximately return the speed to the desired operating point.

D. Load Increase with Spinning Reserve

In order to demonstrate the capability of each generator to independently determine when the demand for generation exceeds the collective constraints in the system, a fourth generator is added to the system at bus 2 as shown in Fig. 8. Initially, the additional generator limits its output to the minimum torque required to spin at synchronous speed, but participates in the distributed algorithm with a maximum torque of $\overline{x}_4^* = 1.42$ Nm. The true maximum output of generator 4 is $\overline{x}_4 = 2.5$ Nm, while the other generators have maximum constraints of $\overline{x}_1 = 3.5$ Nm, $\overline{x}_2 = 2.5$ Nm, and $\overline{x}_3 = 3.0$ Nm. As in the previous two cases, the minimum constraints of all generators is 0 Nm.

The plot in Fig. 7 shows the prime mover torque and the average mechanical speed as the electrical load is increased. After approximately 70 seconds, the non-reserve generators have reached the maximum capacities and thus cannot increase their output to regulate the frequency. At this point, however, the reserve generator observes that $\alpha_4[m] > 1$, thus it determines that the demand exceeds the collective capacity and adjusts its maximum output to its true value. At approximately 180 seconds, the full capacity of the reserve is available and the generators are able to increase the speed in the system and reduce the generation mismatch.

VI. Conclusions and Future Work

In this paper we have discussed the implementation of a distributed algorithm which is capable of controlling generators without relying on a centralized decision maker. A communication and computation testbed created to implement the algorithm was described and an example of the evolution of the algorithm was provided. Additionally, we introduced the notion of distributed generation control and provided experimental results demonstrating a two-stage control architecture which served to regulate the electrical frequency in a small-footprint power system. For future work, we plan to implement the third layer of generation control, i.e., economic dispatch, using the method proposed in [13].

References

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

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

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

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

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

[6] A. D. Domínguez-García and C. N. Hadjicostis, "Coordination and control of distributed energy resources for provision of ancillary services," in *Proc. IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, October 2010, pp. 537 –542.

[7] A. D. Dominguez-Garcia and C. N. Hadjicostis, "Distributed algorithms for control of demand response and distributed energy resources," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, dec. 2011, pp. 27 –32.

[8] P. Sauer and A. Pai, *Power System Dynamics and Stability*. Upper Saddle River, NJ: Prentice Hall, 1998, pp. 106–107.

[9] S. T. Cady, A. D. Domínguez-García, and C. N. Hadjicostis, "Robust implementation of distributed algorithms for control of distributed energy resources," in *North American Power Symposium (NAPS), 2011*, aug. 2011, pp. 1 –5.

[10] S. T. Cady, "Robust implementation of algorithms for distributed generation control of small-footprint power systems," Master's thesis, University of Illinois at Urbana-Champaign, 2012.

[11] H. Dai and R. Han, "Tsync: a lightweight bidirectional time synchronization service for wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, pp. 125–139, January 2004. [Online]. Available: http://doi.acm.org/10.1145/980159.980173

[12] A. D. Domínguez-García, C. N. Hadjicostis, and N. H. Vaidya, "Resilient networked control of distributed energy resources," in *IEEE Journal on Selected Areas in Communications: Smart Grid Communications Series, to appear*, 2012.

[13] A. D. Dominguez-Garcia, S. T. Cady, and C. N. Hadjicostis, "Decentralized optimal dispatch of distributed energy resources," under review for *Conference on Decision and Control (CDC), 2012 51th IEEE Conference on*, dec. 2012.