

A multi-level control architecture for master-slave organized microgrids with power electronic interfaces



Niannian Cai*, Joydeep Mitra

Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:

Received 21 July 2013

Received in revised form

28 November 2013

Accepted 30 November 2013

Keywords:

Microgrid

Multi-agent system

Decentralized control

Power balance

Economic dispatch

Master-slave control

ABSTRACT

This paper presents a multi-level control architecture for autonomous operation of islanded microgrids with power electronic interfaces. An upper layer performs calculations for system dispatch while a lower layer computes appropriate control actions for the power electronic interfaces. The upper layer of communication-based intelligent control is realized through a decentralized multi-agent system (MAS). Two control strategies are implemented in this layer: MAS power balance control and economic dispatch. In MAS power balance control, a rational communication scheme is proposed to construct a minimal spanning tree structure among agents, so that the multi-agent system can efficiently discover system global information and dispatch generation and load to maintain real and reactive power balance; in economic dispatch, by comparing incremental cost with neighbors, generator agents can achieve minimum generation cost for system operation. To demonstrate the speed and effectiveness of the proposed algorithms, IEEE 14, 30, 57 and 118 bus systems are studied for the MAS power balance control and economic dispatch in MATLAB. A demonstration presents implementation of multi-agent system using Java agent development framework and includes a viable implementation of the lower layer, which is used to control power electronic interfaced distributed generators, and is realized through conventional local controllers for master-slave organized microgrids. A case study with time-varying loading conditions is demonstrated to validate the proposed multi-level control architecture.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Centralized control plays a vital role in traditional power systems. However, most microgrids are characterized by dynamically changing configurations and operating modes. Distribution automation systems may reconfigure the network; distributed generating (DG) units may exhaust their fuel and disconnect; renewable resources may cut in or out due to variations in wind or solar energy; storage units may switch between charging and discharging modes; different load resources may connect or disconnect. With the participation of plug-in hybrid/electric vehicles, the extent of this dynamic behavior will increase considerably. For these reasons, many researchers have recognized the benefits of autonomous control, particularly decentralized autonomous control [1–3], which affords maximum operational flexibility. This paper proposes a decentralized autonomous control framework for a microgrid that predominantly uses DG units that interface through power electronic devices, and offers the benefits of fast and robust power management, with the option of fast economic dispatch.

A common expectation of a microgrid is that it should be capable of islanded (off-grid) operation. However, in this mode, several factors affect the ability of microgrids to achieve and maintain stable operation. First, the unavailability of power support from the grid requires a global power balance controller which is able to coordinate power generation and consumption in microgrids and maintain system voltage and frequency. Further, if distributed generators (DGs) in a microgrid are required to be “plug and play”, no pre-determined economic dispatch would satisfy the requirement of the microgrid; thus, an on-line economic dispatch controller is anticipated to reduce generation cost. Moreover, most DGs in microgrids are non-utility-grade, and have to be converted to utility-grade ac through power electronic (PE) interfaces. Therefore, the underlying layer is implemented by local controllers for PE interfaces, which can comply with the top level operating policy or control system of the microgrid. To fulfill above requirements, this paper presents a comprehensive multi-level control architecture. It controls microgrids in three different layers: MAS power balance layer, economic dispatch layer and local control layer.

To achieve power balance, a well-known method proposed previously to maintain power balance is P/f and Q/V droop control [4–7]. This method obtains control signals locally and does not require communications with other components in the microgrid. It benefits from “plug and play”. However, it relies on the

* Corresponding author. Tel.: +1 5178027108.

E-mail addresses: cainiann@msu.edu (N. Cai), mitraj@msu.edu (J. Mitra).

output-voltage settings of DGs; therefore, its performance may be compromised if a tight voltage regulation is expected [8]. Ref. [9] uses a block concept to manage and control different entities and proposes a collection of decent communication rules for distribution grids, which can provide considerable speed advantage. Refs. [10,11] proposed an effective algorithm based on the Average Consensus Theorem to calculate system net power and maintain power balance of the microgrid. The algorithm in [10,11] is implemented in a completely decentralized manner, but may take long to converge for large systems. In this paper we propose a novel multi-agent system-based algorithm that accomplishes exact power balance in three sweeps, regardless of system size. In the proposed algorithm, the information flows in parallel and the results are obtained in a non-iterative way; therefore, this algorithm achieves superior performance in terms of speed without any convergence issues.

MAS-based methods have been widely applied to various power system problems, including microgrid control. Refs. [12,13] have provided an excellent overview of such applications. Ref. [14] decomposed the optimal power flow problem and solved for the smaller subsystems in parallel by using MAS. In [15,16], by making use of autonomous and decentralized characteristics, MAS was introduced to solve electricity market problem. Refs. [17–23] proposed various methods to utilize MAS technique for power balance control and operation. Ref. [24] used MAS technique for optimal reactive power dispatch. Other MAS applications include power system monitoring [25–27], protection [28], restoration [29–32] and stability enhancement [33–35].

Apart from MAS power balance control, this paper also proposes an innovative algorithm for MAS to conduct on-line economic dispatch to reduce generation cost in systems amenable to economic dispatch. This approach is also applicable to microgrids that incorporate energy or ancillary service markets, by replacing costs with bids and maximizing the social welfare function. Many researchers have argued that economic dispatch under decentralized architecture is more beneficial for microgrids in terms of flexibility, scalability and robustness [1,37,3]. Refs. [36,37] presented, respectively, a two-level incremental cost consensus algorithm, and an evolutionary programming method for economic dispatch based on MAS environment. These are both dedicated algorithms for solving economic dispatch problems, but are complex and slow to converge. Ref. [1] proposed a robust and effective method for economic dispatch based on consensus algorithm. The convergence is demonstrated under different communication topologies; however, it requires a leader agent. In this paper we propose an innovative algorithm that is robust and fast for on-line economic

dispatch based on a completely decentralized MAS architecture, which is more flexible and scalable to handle the dynamic feature of micrigrids. Besides, the proposed algorithm is fault-tolerant and allows the system to continue operation in the presence of single point failure. If any agent in the system fails, only the generation at the corresponding node cannot be optimized, but the cost for the remaining system can still be minimized.

In summary, we present here (a) a *comprehensive multi-level control strategy*, that controls the microgrid in a robust, continuous and economic manner, (b) a *novel algorithm* that accomplishes power balance in exactly three sweeps, and (c) a *distributed implementation* of economic dispatch suitable for the proposed control framework.

2. Multi-level control structure

This paper proposes a hierarchical control structure displayed in Fig. 1 for the control of microgrids. The upper decentralized multi-agent layer, acting as a quick-responding system operator, is responsible for power balance and economic dispatch control. The lower local control layer, in response to the power commands required by MAS, regulates real and reactive power output of local DGs. The whole system works as an interactive power electronic network [38].

For the MAS power balance control, a fast three-sweep algorithm is proposed to react to power unbalance in real time, so as to maintain stable operation of microgrids. The functions of the three sweeps are to determine information flow route, obtain net real and reactive power, and dispatch generation and load. All the information in these three sweeps is designed to flow in parallel, so that the required communication can be achieved rapidly in the proposed MAS architecture. MAS power balance control is critical for stable operation; therefore, it has the highest priority in MAS computation and control.

Economic dispatch is conducted on-line among agents that are responsible for generation nodes (also called generation agents). Since real power balance is maintained in MAS power balance control layer, economic dispatch will optimize total generation cost while maintaining total real power generation.

As shown in Fig. 1, each local controller will receive real and reactive power settings from the MAS layer and implement local control for PE interfaces. However, because only load information is transmitted and losses are not calculated, the MAS power balance control achieves an approximate allocation of generation. The losses, which equal the difference in actual generation and load, are compensated by the master PE interface that operates in the voltage

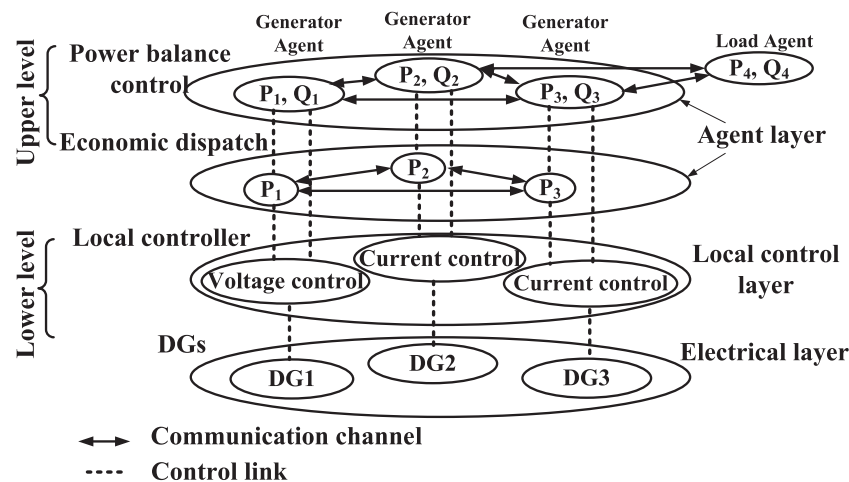


Fig. 1. Proposed hierarchical control architecture.

controlled mode and thereby serves as slack bus. This behavior is inherent in the operation of the voltage controlled inverter, which generates the necessary current to absorb the slack.

The multi-level inter-related control layers work as an integrated system. Each layer deals with different control objectives, and they work together to control the microgrid in a robust, continuous and economic manner. In the following sections, details of MAS power balance control layer and economic dispatch layer are provided.

3. MAS power balance control

In order to maintain power balance, dispatch generation or demand within microgrids, the entire process consists of three stages: determine information flow route, obtain net real and reactive power, and re-assign generation and load values. Since there is no central agent in the decentralized architecture of multi-agent system, no single agent unilaterally governs others to achieve a global goal. All agents are at the same hierarchical level and obey specific rules to communicate so that all those autonomous agents can coordinate to manage the microgrid. Consider a system connected as in Fig. 2. It is a simple but typical structure and embodies a ring. It is easy to find that a communication algorithm applied to this structure can be extended to a more complicated structure which may have multiple rings or radial lines. In this figure, each node represents an agent, and the position of the node shows the topological position of the corresponding agent. An arc connecting two nodes implies that the agents represented by these two nodes are neighbors and they can communicate with each other to exchange information and data. Two agents without a line connecting them are not considered neighbors and cannot communicate directly.

To explain the theory presented here, it is necessary to first provide some definitions. The agent that is processing system information at a given point of time is called the *current agent*. If agent A_1 transmits information to agent A_2 , then A_1 is called A_2 's *parent agent*, and A_2 is called A_1 's *child agent*. *View*, either local or global, is an agent's knowledge of system information. This information consists of non-dispatchable real and reactive power generation capacity P_{NG} , Q_{NG} , dispatchable real and reactive power generation capacity P_{DG} , Q_{DG} , vital real and reactive load demand P_{VL} , Q_{VL} , and non-vital real and reactive load demand P_{NL} , Q_{NL} . *View* is mathematically defined as a vector $\mathbf{u}$.

$$\mathbf{u} = [u_1 \ u_2 \ u_3 \ u_4 \ u_5 \ u_6 \ u_7 \ u_8]^T = [P_{NG} \ Q_{NG} \ P_{DG} \ Q_{DG} \ P_{VL} \ Q_{VL} \ P_{NL} \ Q_{NL}]^T \quad (1)$$

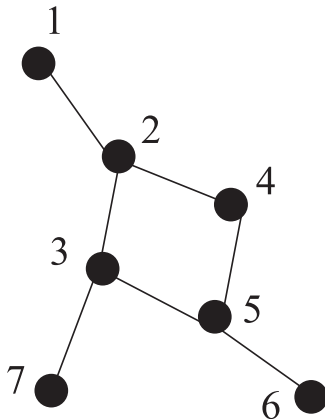


Fig. 2. Multi-agent system structure.

Table 1
Parent child relationship diagram.

Agent	1	2	3	4	5	6	7
Parent ID	–	1	2	2	4	5	3
Child ID	2	3, 4	7	5	6	–	–

3.1. Stage 1: discovery of minimal spanning tree

This process is intended to organize the decentralized multi-agent system and steer the flow of information. In the proposed method, there are three basic requirements that must be met in order to efficiently achieve the discovery of power information in the decentralized microgrid. First, a communication protocol should be defined to conduct information flow in a manner that all the nodes in the system are spanned. Second, this protocol should be designed to route the information flow such that every node receives and processes information only once. Finally, real time control of multi-agent system requires this protocol to be able to discover system information in parallel, so that it can quickly react to the disturbance in the system.

Toward meeting the above requirements, a minimal spanning tree is constructed first. This process is executed as follows:

- A token is generated by a starting agent.
- Every agent who receives the token, memorizes its *parent agent* ID, then it transmits the token to all its other neighbors, and stores its *child agent* IDs.
- Any agent, who receives multiple tokens simultaneously, keeps one and discards others. At the same time, removes child-parent relationships with those whose tokens are discarded.

To demonstrate this algorithm, consider the example shown in Fig. 2. Since all agents are identical, the starting agent can be selected randomly. In this case, let us simply choose agent 1 as starting agent. Initially, agent 1 generates a token and transmits it to agent 2. Then agent 2 receives the token and sends it to its neighbors, agents 3 and 4. After that, agent 3 sends the token to its neighbors, agents 5 and 7. In parallel, agent 4 will send the token to agent 5. At this point, agent 5 will have received two tokens, so it discards one. Let us assume it discards the token coming from agent 3, thereby removing a redundancy in information flow. Finally, the token will flow from agent 5 to agent 6. At this point, all the agents in the system have been discovered. During this process, all agents will store their *parent agent* and *child agent* IDs. Their relationships are shown in Table 1. Fig. 3(a) and (b) depicts the transmission path of the token and the minimal spanning tree established by Stage 1.

Note that although removal of any one of the arcs 2–3, 2–4 or 4–5 instead of arc 3–5 has the same effect of establishing a minimal spanning tree, the sequence of discovery described above ensures the shortest communication time.

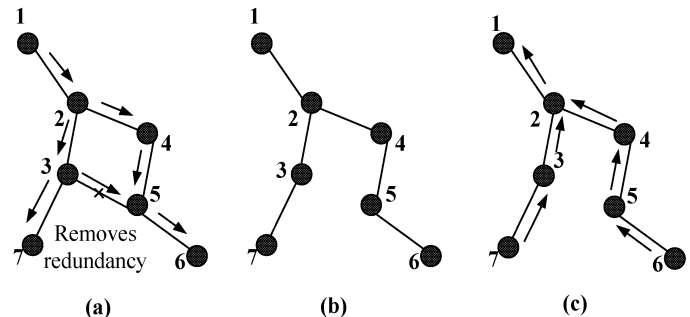


Fig. 3. (a) Token transmission route. (b) Minimal spanning tree constructed. (c) Information flow path for Stage 2.

3.2. Stage 2: information feedback process

Once the information flow path is established in Stage 1, the next step is designed to collect system generation and load data. This is achieved as follows:

- When no new neighbors are discovered, each agent now transmits to its *parent agent* its local view of the system.
- When an agent receives system view information from all its *child agents*, it processes this data as shown in (2) below and transmits its updated view information to its *parent agent*.

$$\mathbf{u} = \mathbf{u}^{local} + \sum_{i \in \phi} \mathbf{u}^i \quad (2)$$

where $\mathbf{u}$ is *current agent's* updated view; $\mathbf{u}^{local}$ is *current agent's* local view; ϕ is the set of child IDs for the *current agent*; $\mathbf{u}^i$ is *child agent i's* updated view.

Stage 2 ensures that at each node the local view information flows back from *child agent* to its *parent agent*. Finally, the starting agent can obtain system global view and calculates net power based on (3). Fig. 3(c) shows the information flow path for Stage 2.

$$\begin{bmatrix} P_{net} \\ Q_{net} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & 0 & -1 \end{bmatrix} \mathbf{u} \quad (3)$$

where P_{net} and Q_{net} are system net real and reactive power; $\mathbf{u}$ is updated view of starting agent.

3.3. Stage 3: generation and load dispatch

At the end of Stage 2, the starting agent obtains global view of system generation and load demand. Based on the information collected, decentralized agents can dispatch generation or curtail load from *parent agents* to *child agents*. This is achieved as follows:

- When receiving system net power values, each agent first adjusts its local available generation or load to minimize the unbalanced power. Mathematically, it is described by (4)–(6). Minimize:

$$|I_{net}^c| = |I_{net}^p - \tilde{I}_{DG} + \tilde{I}_{NL}|, \quad I \in \{P, Q\} \quad (4)$$

Subject to:

$$0 \leq \tilde{I}_{DG} \leq I_{DG} \quad (5)$$

$$0 \leq \tilde{I}_{NL} \leq I_{NL} \quad (6)$$

where I_{net}^c is system net P or Q after dispatch by *current agent*; I_{net}^p is system net P or Q obtained from *parent agent*; $\tilde{I}_{DG}$ and $\tilde{I}_{NL}$ are dispatched power values for *current agent*. Note that only dispatchable generators and non-vital load can be dispatched.

- If local availability is not sufficient, each agent, based on the updated system view obtained in Stage 2, will split the net power values according to its *child agents'* dispatchable capacity, and transmit new values to its *child agents*. Mathematically, this is stated as:

$$I_{net}^i = \begin{cases} \frac{I_{DG}^i}{\sum_{m \in \phi} I_{DG}^m} I_{net}^c & \text{if } I_{net}^c \geq 0 \\ \frac{I_{NL}^i}{\sum_{m \in \phi} I_{NL}^m} I_{net}^c & \text{if } I_{net}^c < 0 \end{cases}, \quad I \in \{P, Q\} \quad (7)$$

where I_{net}^i is net P or Q for *child agent i*; I_{DG}^i is u_3 or u_4 , updated dispatchable generation information that agent i obtained in

Stage 2. ϕ is the set of *child agents* for the *current agent*. I_{net}^c is the net P or Q obtained by *current agent*; I_{NL}^i is u_7 or u_8 , updated non-vital load information that *child agent i* obtained in Stage 2.

The information flow path in this step is in the opposite direction to that of Stage 2. To implement the proposed strategy, MAS system selects a bus with a certain amount of reserve generation as the slack bus. This bus regulates the voltage and sets the reference frequency of the microgrid, and picks up the slack in real and reactive power generation, including the system losses.

4. Economic dispatch

The algorithms in Section 3 contributes to maintain stable steady state operation, and are adequate for most microgrids. However, if a microgrid is amenable to optimal dispatch, i.e., if its generating capacity exceeds its load and has some DGs that are dispatchable, then the framework proposed in this paper can also perform economic dispatch. This can be achieved by applying the following algorithms, which modify the dispatched power values obtained in Section 3 to realize minimum generation cost.

Assume there are n generators in a microgrid. Each generator has a convex cost function $F_i(P_i)$ in terms of real power output P_i . If system losses are not considered, the economic dispatch problem can be stated as follows: given total system load, schedule real power output of each generator so that the total generation cost is minimized while satisfying upper and lower output constraints of generators, as well as real power balance requirement. Mathematically, this optimization problem can be stated as follows [39]:

Minimize:

$$F = \sum_{i=1}^n F_i(P_i) \quad (8)$$

Subject to:

$$P_D = \sum_{i=1}^n P_i \quad (9)$$

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (10)$$

where P_i is real power output of generating unit i ; P_D is total load demand in the microgrid; $P_i^{\min}$ and $P_i^{\max}$ are lower and upper limits respectively of generating unit i .

The Lagrange function can be constructed as (11) [39]:

$$L = \sum_{i=1}^n F_i + \lambda (P_D - \sum_{i=1}^n P_i) + \sum_{i=1}^n \lambda_i (P_i^{\min} + s_i^2 - P_i) + \sum_{i=1}^n \lambda_i (P_i^{\max} - t_i^2 - P_i) \quad (11)$$

where λ , λ_i , λ_i^i : greaterthan Lagrange multipliers s_i , t_i : real variables that convert inequality constraints to equality (i.e., slack and surplus variables).

Based on the necessary condition, it is easy to obtain that at the optimal point [39]:

$$\lambda^i = 0, \lambda_i = 0, \frac{\partial F_i}{\partial P_i} = \lambda, \quad \text{when } P_i^{\min} < P_i < P_i^{\max} \quad (12)$$

$$s_i = 0, \lambda_i = 0, \frac{\partial F_i}{\partial P_i} - \lambda - \lambda^i = 0, \quad \text{when } P_i = P_i^{\min} \quad (13)$$

$$t_i = 0, \lambda^i = 0, \frac{\partial F_i}{\partial P_i} - \lambda - \lambda_i = 0, \text{ when } P_i = P_i^{\max} \quad (14)$$

Eqs. (12)–(14) describe necessary conditions to obtain minimum generation cost. This just implies that at the optimal operating point, generators in the system are dispatched at equal incremental cost, except those units that have already reached upper or lower limits.

Now the power balance requirement in the microgrid can be satisfied by multi-agent control as described in Section 3. Each agent obtains an assignment that satisfies power balance described by Eq. (15), but it is not the optimal value that minimizes generation cost.

$$\sum_{i=1}^n P_i^{(0)} = P_D \quad (15)$$

where $P_i^{(0)}$ = initial generation assignment from MAS power balance control for generation agent i . Realize that this initial value is within generation limits of generation agent i .

Two neighbor agents m and n can meet optimal conditions (12)–(14) by implementing the behavior described by (16) and (17):

$$F'_m(P_m^{(k+1)}) = F'_n(P_n^{(k+1)}) \quad (16)$$

$$P_m^{(k+1)} + P_n^{(k+1)} = P_m^{(k)} + P_n^{(k)} \quad (17)$$

where $F'_m = dF_m/dP_m$. Agent behavior as in (16) and (17) provides a means of achieving minimum generation cost.

Proof of convergence: To establish the stability of this algorithm, a convergence analysis is performed, based on a Lyapunov function. Note that a convex and differentiable cost function $F_i: \chi \rightarrow \mathbb{R}$ satisfies the first-order convexity condition:

$$F_i(y) \geq F_i(x) + F'_i(x)(y - x), \quad \forall x, y \in \chi, \quad (18)$$

Define a Lyapunov function $V: \chi^N \subset \mathbb{R}^N \rightarrow \mathbb{R}$ of the form shown in (19):

$$V(\mathbf{P}^{(k)}) = \sum_{i \in \mathcal{V}} F_i(P_i^{(k)}) \quad (19)$$

where $\mathbf{P}^{(k)}$ is a vector consisting of nodal generation values in iteration k ; $P_i^{(k)}$ is the i th element of $\mathbf{P}^{(k)}$ representing generation value for agent i ; $\mathcal{V}$ is the set of agents in the system.

Note that F_i is the cost function for generator unit i , so it is non-negative if P_i is within generation limits. Therefore, we have:

$$V(\mathbf{P}^{(k)}) \geq 0 \quad (20)$$

Assume in iteration k , agent m and agent n accomplish behaviors described by (16) and (17). Then,

$$\begin{aligned} V(\mathbf{P}^{(k+1)}) - V(\mathbf{P}^{(k)}) &= F_m(P_m^{(k+1)}) + F_n(P_n^{(k+1)}) - F_m(P_m^{(k)}) \\ &\quad - F_n(P_n^{(k)}) \leq F_m(P_m^{(k+1)}) + F_n(P_n^{(k+1)}) - [F_m(P_m^{(k+1)}) \\ &\quad + F'_m(P_m^{(k+1)})(P_m^{(k)} - P_m^{(k+1)})] - [F_n(P_n^{(k+1)}) + F'_n(P_n^{(k+1)})(P_n^{(k)} - P_n^{(k+1)})] \\ &= -F'_m(P_m^{(k+1)})(P_m^{(k)} - P_m^{(k+1)}) - F'_n(P_n^{(k+1)})(P_n^{(k)} - P_n^{(k+1)}) = 0 \end{aligned} \quad (21)$$

Therefore, the convergence of the designed algorithm is guaranteed.

However, if agent m reaches its limit before it satisfies (16), then (16) does not need to be satisfied. Agent m chooses its limit as new assignment, and agent n calculates its assignment by (22). A similar logic applies to agent n .

$$P_n^{(k+1)} = P_m^{(k)} + P_n^{(k)} - P_m^{(k+1)} \quad (22)$$

where $P_m^{(k+1)} = P_m^{\text{limit}}$, the limit can be the upper bound or lower bound of agent m , depending on which one prevents agent m from having the same incremental cost as agent n . This behavior prevents generation output from violating generation limits, thus Eq. (20) is always valid.

The procedure for decentralized agents to achieve optimal economic dispatch is as follows:

- Agent m selects its neighbor agent n .
- Both agents implement behaviors (16) and (17), if one agent encounters a limit, then it uses this limit as its new assignment, meanwhile, the other agent implements behavior (22).
- If the assignments of all agents converge within the accepted tolerance, optimization process is terminated. Otherwise, steps (a) through (c) are repeated.

Generation agents can implement behaviors described above simultaneously; hence, the convergence time for this algorithm is more competitive than single-thread optimization. However, to implement this algorithm, generation agents may be required to discover extra neighbors, apart from topological neighbors. Because this algorithm is based on communications between generation agents, topologically, it is possible that one generation agent may not have any generation agent as a neighbor; then its cost cannot be optimized. Also note that not all microgrids permit economic dispatch; the algorithm presented in this section enhances the general framework for the control of microgrids, and remains available should a microgrid evolve to where it does permit economic dispatch. Where a microgrid incorporates energy or ancillary service markets, this overall approach can still be applied, by replacing costs with bids and maximizing the social welfare function.

5. Mathematical simulation of proposed multi-agent system

This section simulates MAS on MATLAB platform to show the performance of the proposed decentralized MAS and compares it with the work that has been reported in the literature.

5.1. MAS power balance control

MAS power balance control is critical for stable operation of microgrids; therefore, it has the highest priority in the proposed multi-level control architecture.

To demonstrate the speed and effectiveness of the proposed MAS power balance control strategy, IEEE 14-bus, 30-bus, 57-bus and 118-bus systems are tested. The system configurations and data can be obtained from [40].

If two nodes are adjacent to each other through direct electrical connection, their agents are neighbors and they can communicate directly with each other. The simulations are conducted by selecting agent 1 (corresponding to bus 1) as starting agent. Simulation results are shown in Table 2. A hop represents the path between two neighbor nodes. Note that this lists the maximum number of hops required for the MAS power balance control; because for the last sweep, once the net power reaches zero, power balance is achieved and net power information does not need to be transmitted further. However, in this simulation, power balance is obtained when the dispatch information has reached all the agents.

Table 2
Simulation results for power balance control.

Period	IEEE 14 bus	IEEE 30 bus	IEEE 57 bus	IEEE 118 bus
Hops	12	18	30	42

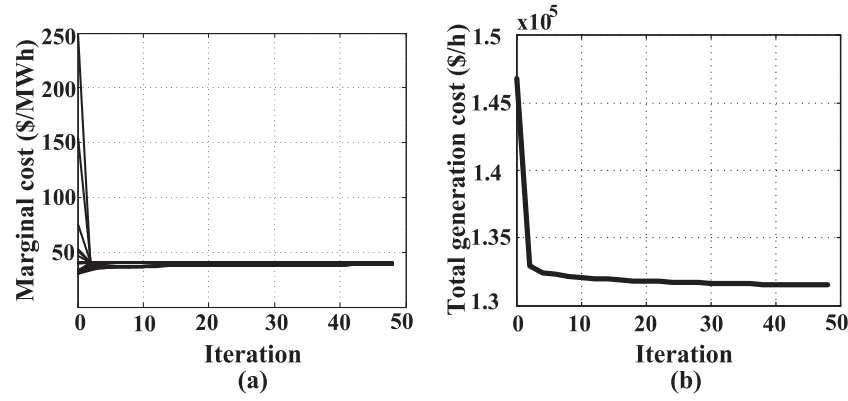


Fig. 4. Simulation results for economic dispatch.

The time to achieve power balance can be estimated from the number of hops shown in Table 2. The actual execution time of the algorithm depends on the specific implementation, in terms of hardware and software. Ref. [10] uses Eq. (23) to estimate the communication time.

$$T = \frac{n_i \times n_d \times n_b}{R} \quad (23)$$

where n_i is the number of hops for the algorithm; n_d is the number of data points required to be transmitted; n_b is the number of bits used to represent each data point; R is the communication speed in bits/s. For instance, for IEEE 118 bus system, the total number of hops is 42, with 14 hops per sweep. If the token in the first sweep is represented by 8 bits, and each data point in the second and third sweeps is represented by 16 bits, then for a network speed of 10 Mbit/s, the execution time can be estimated as $(14 \times 1 \times 8 + 14 \times 8 \times 16 + 14 \times 2 \times 16) / 10,000,000 = 0.000235$ s.

Ref. [10] used the IEEE 162-bus system [40]. We compare here the performance of the proposed algorithm with that in [10] for the same system. The agent at bus 1 is chosen as starting agent. It takes 9 hops for each sweep, requiring 27 hops in all to reach power balance. Assuming the same number of bits and communication rate as those in [10], the estimated time can be computed as $(9 \times 1 \times 8 + 9 \times 8 \times 16 + 9 \times 2 \times 16) / 10,000,000 = 0.0001512$ s. However, [10] reports that it took their algorithm 0.1283 s to converge. Note that the times calculated here and in [10] do not consider time delays or other factors. They vary for different implementations of hardware and software. However, this approximation clearly shows the speed advantage for the proposed algorithm, which results from the parallel nature of information flow in this algorithm.

Further discussion: The speed of the proposed algorithm depends highly on the coupling of the agents. For the same scale MAS, the tighter coupling (higher number of branches) the MAS has, the faster the solution can be reached.

5.2. Economic dispatch

Economic dispatch is an additional service provided by MAS where the capability exists. It can help to optimize total generation cost, but it cannot change total generation amounts determined by MAS power balance control.

The proposed economic dispatch algorithm is also applied to IEEE 118-bus test system. This system has 54 generators; therefore, 54 agents are involved in this control. Total load demand is 4.377×10^3 MW. Assume that each agent has six direct neighbors. (The number of neighbors in this case is not initially known, as explained in the last paragraph of section 4.) Generation cost and limit data are obtained from [41]. Initially, assume that the

54 generators share the load in proportion to their capacities. If the convergence tolerance is set to 1×10^{-4} pu, it takes 46 iterations to converge. Simulation results are shown in Fig. 4. If each agent stores cost coefficients and generation upper and lower limits of its neighboring generators, then the only data required to be communicated is the generation value. Therefore, an estimated time can be computed as $46 \times 1 \times 16 / 10,000,000 = 0.0000736$ s. This time does not include the computational time; however, considering the computation required in this algorithm is not complex and the computation time is usually much smaller than communication time, the total computation time will not be significant.

Further discussion. Fig. 4(b) shows that the cost decreases rapidly in the first few iterations, while in subsequent iterations the improvements diminish. In this example, the saving is about $\$1.5 \times 10^4$ /h in the first 12 iterations; however, in the last 34 iterations, the saving is only about $\$5 \times 10^2$ /h. Therefore, if the tolerance is set appropriately larger, the time required for the algorithm can be reduced significantly, while the saving is not compromised too much. In this case, if the tolerance is set to 1×10^{-3} , it takes 12 iterations to converge. So the time is reduced by 73.91% while the saving in cost is only reduced by about 3.34%.

6. Multi-agent implementation and demonstration platform

6.1. Multi-agent implementation

Multi-agent system is implemented in Java Agent Development (JADE) [42,43] framework, which complies with the Foundation of Intelligent Physical Agents (FIPA) [44]. FIPA Semantic Language (FIPA-SL) is adopted in this work for agent communication.

The action of an agent is realized by implementing behaviors. Each behavior is an object of a class that extends *Behaviour* class in the *jade.core.behaviours* package. There are two main behaviors for the proposed agents: Power Balance and Economic Dispatch.

1. **Power balance.** This behavior implements the algorithm described in Section 3. It contains three sub-behaviors:

- Token transfer.** This sub-behavior implements the algorithm in Section 3.1 and builds parent-child relationships among agents, based on a minimal spanning tree. In summary, this behavior waits for the first token, rejects all other tokens which come later, and removes neighborhood relationships with those agents whose tokens are rejected. Then this behavior sends the token to all its other *child agents*.
- Information feedback.** This sub-behavior implements the algorithm in Section 3.2 and helps collect view information from *child agents* to *parent agents*. When each agent receives

view from all its *child agents*, this sub-behavior processes the view and sends the updated view to the *parent agent*.

- (c) *Generation or load dispatch*. This sub-behavior implements the algorithm in Section 3.3. It receives net power values from *parent agent*, adjusts local dispatchable generation or non-vital load to minimize net power, and splits net power values for its *child agents* based on (7). When net real and reactive power values are both zeros after local adjustment, this sub-behavior informs neighbors of power balance completion. If an agent receives information of power balance completion from all its *child agents*, this sub-behavior informs *parent agent* of power balance completion. If an agent receives information of power balance completion from *parent agent*, this sub-behavior informs *child agents* of power balance completion.

2. *Economic dispatch*. This behavior, as the name suggests, realizes economic dispatch function by implementing the algorithm proposed in Section 4. There are two states of this behavior:

- (a) *Inquiry state*. The agent sends inquiring and local generation value to one of its neighbors to ask for economic dispatch implementation. If it receives *accept* and *local generation* values from its neighbor, it updates its generation value by computing (16) and (17) and goes to the *reply state*. If it receives *refuse* from the neighbor, it goes to the *reply state*. During *inquiry* state, whenever the agent receives other agents' inquiry for economic dispatch implementation, it refuses.
- (b) *Reply state*. This state waits for *inquiry* from other agents. If it receives *inquiry*, it replies with *accept* and *local generation* values; then it updates *local generation* by computing (16) and (17). There is also a time interval set for *reply state*. When the time expires, the behavior goes to *inquiry* state.

6.2. Simulation platform of lower level

Lower-level simulation platform consists of local control layer and electrical layer. In this simulation, electrical layer is constructed by master-slave organized DGs with PE interfaces. The configuration of a generic PE interface is shown in Fig. 5. L and C are, respectively, the inductance and capacitance of the output stage inductors and capacitors. In master-slave organization, PE-based DGs in an islanded microgrid can be classified into two types: a master inverter operates in voltage source mode, providing voltage support for the microgrid; all other (slave) inverters operate as current sources, supplying specific real and reactive power into the microgrid.

6.2.1. Master power electronic interfaces

The objective of the control strategy for voltage-source PE interface is to output required voltage at the sending point, which will be used as the reference voltage level for the microgrid. Normally, sources which are employed to offer voltage support in micro-

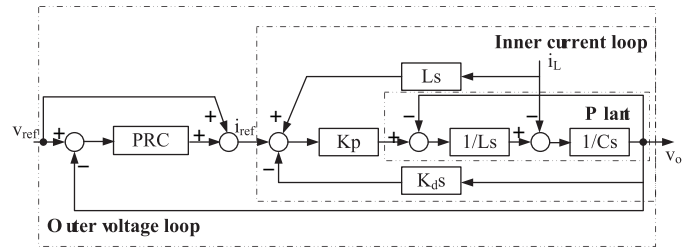


Fig. 6. Control block for voltage-source power electronic interface.

grids have large reserve capacity, so that it is able to provide or absorb extra power when any disturbance occurs in the microgrids, thereby maintaining the voltage.

A multi-loop control strategy with an inner current control loop and an outer voltage control loop is adopted for the voltage-source PE interface shown in Fig. 6. The transfer function of the inner current control loop is described by

$$V_o = \frac{K_p}{LCs^2 + K_p K_d s + 1} I_{ref} - \frac{(1 - K_p)Ls}{LCs^2 + K_p K_d s + 1} I_L \quad (24)$$

To improve the response to load variation, a feedforward loop is adopted for the load current. If $K_p = 1$, from the transfer function, the coefficient of I_L is zero; therefore, the effect of load current on the output voltage is compensated. The damping ratio of the inner loop is derived as

$$\xi = \frac{K_d K_p}{2\sqrt{LC}} \quad (25)$$

The differential feedback from the capacitor voltage introduces damping into the system, avoiding high resonant peak from a pure LC filter. The damping ratio can be adjusted by adjusting K_d . Theoretically, the optimal performance can be obtained when $\xi = 0.707$, from which K_d can be computed.

In order to smooth the transition from grid-connected mode to islanded mode when the grid fails, a reference voltage feed-forward loop is utilized to accelerate the response of the control system. Traditional stationary PI controllers suffer from steady state error when tracking sinusoidal waveforms. High gain can decrease the error, but it will also decrease magnitude and phase margin, potentially introducing instability. A popular method of minimizing the steady state error is to use a PI controller in DQ space. However, it requires computations for the frame transformation. Moreover, sources in microgrids can be single-phase or three-phase. The implementations of single-phase and three-phase DQ controls are different. This platform utilizes a low-pass to band-pass technique expressed in (26) that can transform a dc compensation network in rotating frame into an equivalent ac compensation network in stationary frame to achieve identical response feature in the frequency area of interest [45] (it should be noted that the terms in the parentheses on the right comprise the argument of the gain function):

$$G_{AC}(s) = G_{DC} \left(\frac{s^2 + \omega_0^2}{2s} \right) \quad (26)$$

Hence the equivalent PI controller in rotational frame can be expressed in the stationary frame as:

$$G_{PRC}(s) = K_{p1} + \frac{2K_i s}{s^2 + \omega_0^2} \quad (27)$$

This controller, which is also called proportional resonant (PR) controller, can vastly boost the gain at resonant frequency. It does not require frame transformation and can be uniformly implemented for both single-phase and three-phase sources. Theoretically, at resonant frequency, infinite gain can be obtained to

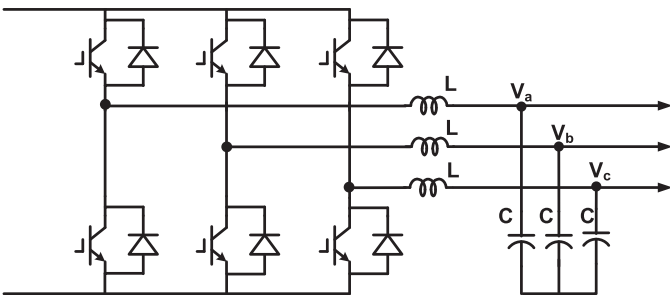


Fig. 5. Configuration of PE interface.

remove steady state error. Since PR controller can affect the phase within the band of resonant frequency, a careful selection of K_{p1} and K_i is required to decrease the bandwidth of PR controller while maintaining sufficient phase margin around resonant frequency.

6.2.2. Slave power electronic interfaces

Slave PE interfaces work in current control mode and supply specific real and reactive power assigned by multi-agent system into the microgrid. In this simulation platform, voltage level is set up by a master PE interface source, and slave PE interfaces do not directly participate in voltage control. Instead, they regulate their output current to provide specified real and reactive power output. Since voltage is maintained by the voltage source, in order to produce required real and reactive power output using multi-agent system, current-source PE interfaces can adjust their current to regulate power.

The controller shown in Fig. 7 is utilized for a slave inverter. A proportional resonant controller is also employed to decrease steady state error. However, the classic PR controller as expressed in Eq. (27) cannot be applied here, because current control inverter is a first order system. The plant originally has a -90° phase shift, while the classic PR controller will introduce another -90° phase shift at resonant frequency, which will cause the system to be unstable. To overcome this instability problem, a modified form of PR controller as expressed in (28) is utilized, where ω_c is the breakpoint frequency of dc transfer function [45]. The new controller can boost gain considerably at resonant frequency while its phase shift can be adjusted by changing K_i .

$$G_{MPRC}(s) = K_{p1} + \frac{2K_i\omega_c s}{s^2 + 2\omega_c s + \omega_0^2} \quad (28)$$

A feedforward loop of output voltage achieves acceleration of the controller's dynamic response to the commands from the multi-agent system, and diminishes the influence of the load's effect on the output current. If $K_p = 1$, the open loop transfer function for the reference current is described by (29). K_{p1} , K_i and ω_c can be designed to obtain sufficient phase margin, while the gain at fundamental frequency is still boosted sufficiently so that steady state error can be minimized.

$$G_o(s) = \frac{K_{p1}s^2 + 2(K_i + K_{p1})\omega_c s + K_{p1}\omega_0^2}{Ls^3 + 2L\omega_c s^2 + L\omega_0^2 s} \quad (29)$$

6.3. Multi-level control architecture

To demonstrate the effectiveness of the proposed multi-level control architecture, a prototype microgrid shown in Fig. 8 is simulated in MATLAB/SIMULINK. MAS is implemented using JADE described in section 6.1. This study system is connected to a utility through a breaker at point of common coupling (PCC). The breaker in this simulation is open; therefore the microgrid is operated in islanded mode. (Note that in grid-connected mode, the grid at the PCC would provide the voltage reference, and all inverters in the microgrid operate in current control mode.) Agents A1–A7 are

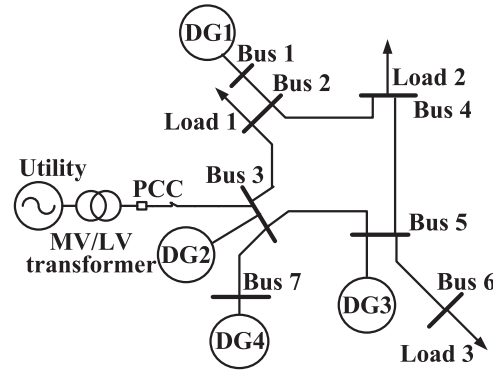


Fig. 8. Microgrid configuration under study.

Table 3
Generation data.

Node	Max gen		Min gen		Coefficient		
	kW	kvar	kW	kvar	a_0	b_0	c_0
DG1	20	10	10	0	8	0.012	0.008
DG2	10	6	0	0	12	0.016	0.01
DG3	15	0	15	0	4	0	0
DG4	10	4	0	0	6	0.014	0.013

responsible for bus 1–bus 7 respectively. The structure of the multi-agent system is the same as that of Fig. 2. Since buses 1, 3, 5 and 7 are equipped with DGs, their agents are called generation agents. To enable economic dispatch between these four agents, assume that they establish neighborhood relationships among themselves, and are able to communicate with each other. All the other neighborhood relationships are established based on electrical connections.

The generator data for this microgrid are shown in Table 3. The system consists of four DGs, among which DG3 is a solar panel. It is controlled at maximum power point, consequently, its real power output is not dispatchable, while DG1, DG2, DG4 are both fully dispatchable. Their power outputs can be adjusted through commands from the multi-agent system. The generator cost function of a micro-source is expressed as:

$$F = a_0 + b_0 P + c_0 P^2 \quad (30)$$

Detailed data for the PE interfaces and local controllers are displayed in Table 4. The studied load profile is shown in Table 5. The message traces of token transfer to build minimal spanning tree structure among MAS are shown in Fig. 9. The sequence of agents that receive the token verifies the description in Section 3.1.

Table 4
Data for PE interfaces and local controllers.

	DG1	DG2	DG3	DG4
L	1 mH	0.56 mH	0.4 mH	0.56 mH
C	6.34 μ F	11.6 μ F	13 μ F	11.6 μ F
K_p	1	1	1	1
K_d	0.000316	0.000114	0.000102	0.000114
K_{p1}	3	4	4	4
K_i	20	80	80	80

Table 5
Load profile.

Period	Load 1 (kvar)	Load 2 (kvar)	Load 3 (kvar)
0–0.05 s	10 + j3	20 + j10	10 + j3
0.05–0.1 s	5 + j3	18 + j5	10 + j5
0.1–0.15 s	25 + j8	20 + j6	13 + j4
0.15–0.2 s	20 + j5	15 + j6	13 + j4

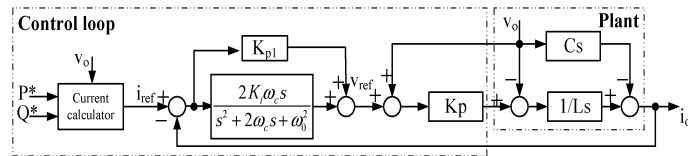


Fig. 7. Control block for current-source power electronic interface.

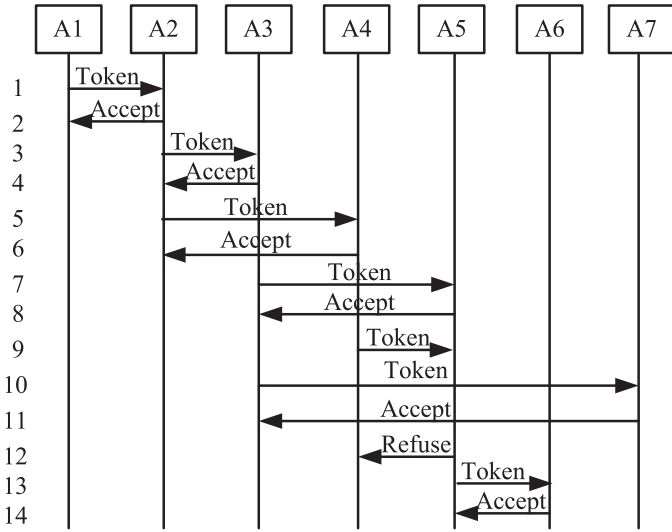


Fig. 9. Traces of messages for token transfer.

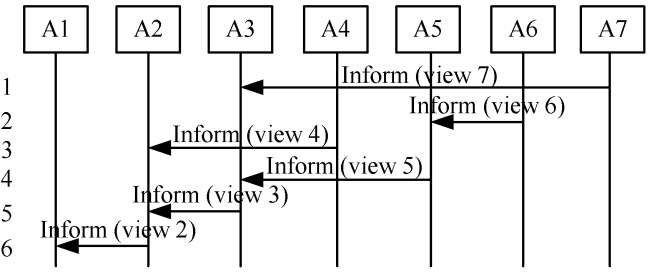


Fig. 10. Traces of messages for information feedback process.

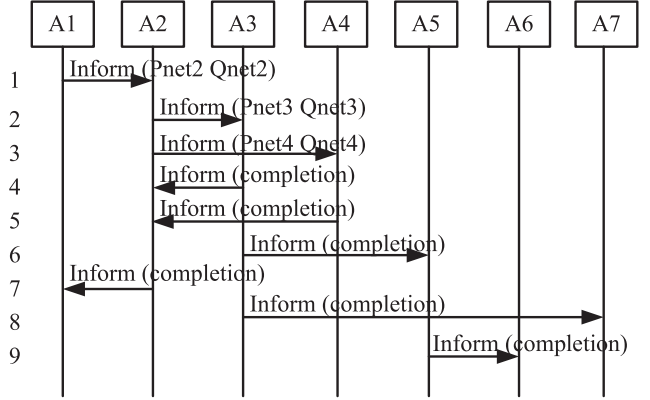


Fig. 11. Traces of messages for generation or load dispatch.

It should be pointed that after receiving the token from A3, agent A5 refuses the token which comes later from A4 and breaks the neighborhood relationship with A4. This rule helps remove the communication redundancy and construct minimal spanning tree structure. After building the minimal spanning tree, the view information is transmitted from *child agent* to *parent agent* shown in Fig. 9. Agents A7, A6 and A4 stay at leaves of the minimal spanning tree; therefore, they initialize the information feedback process described in Section 3.2 (Fig. 10).

The message traces of generation or load dispatch process during 0–0.05 s are shown in Fig. 11. Total power demand is $40 + j16$ kvar, less than total generation capacity, which is $55 + j20$ kvar. Therefore, the starting agent A1 will cut off its dispatchable generation $10 + j4$ and send net power value $5 + j0$ kW to A2. Since bus 2 does not have generation connected to it, A2 cannot reduce net power. It splits the net power values according to its *child agents'* updated view expressed by (7). A4 is the leaf agent and it does not have generation connected to it. From (7), the split net power values it receives are $P_{net4} = 0$, $Q_{net4} = 0$. The net power for A3 is $5 + j0$ kvar. A3's local dispatchable generation can reduce 5 kW to make net power zero, which initializes A3 to inform its neighbors of power balance completion. At the same time, A4 detects that the net power it receives is also zero. Then A4 will inform A2 of power balance completion. When A2 receives power balance completion from both A3 and A4, it will inform its *parent agent* A1 of power balance completion. After power balance, MAS can also conduct economic dispatch. The optimal generation values are obtained as follows: $P_{DG1} = 10.465$ kW, $P_{DG2} = 8.172$ kW,

$P_{DG3} = 15$ kW, $P_{DG4} = 6.3631$ kW. Convergence of the proposed economic dispatch is shown in Fig. 12(a). For this small system, only three iterations are needed for convergence. The global marginal cost is around \$0.18/kWh. DG3 is a solar panel. Its output power cannot be dispatched and so its marginal cost does not change. Fig. 12(b) shows the iterations to obtain the optimal generation values. None of dispatchable generators reaches its real power limit in this case. Total reactive power demand during this period is 16 kvar; hence, after MAS power balance control, reactive power generation values for the DGs are: $Q_{DG1} = 10$ kvar, $Q_{DG2} = 2$ kvar, $Q_{DG3} = 0$ kvar, $Q_{DG4} = 4$ kvar.

At time $t = 0.05$ s, load changes. The total real power demand is 33 kW, which is still less than the total generation capacity. The economic dispatch is displayed in Fig. 14. The global marginal cost converges to \$0.106/kWh. DG1 is constrained by its lower limit of real power output; therefore, its marginal cost converges to \$0.172/kWh, larger than global marginal cost. From Fig. 13(b), new optimal generation values for this load profile are: $P_{DG1} = 10$ kW, $P_{DG2} = 4.4783$ kW, $P_{DG3} = 15$ kW, $P_{DG4} = 3.5217$ kW. The total reactive power demand is 13 kvar. Reactive power generation

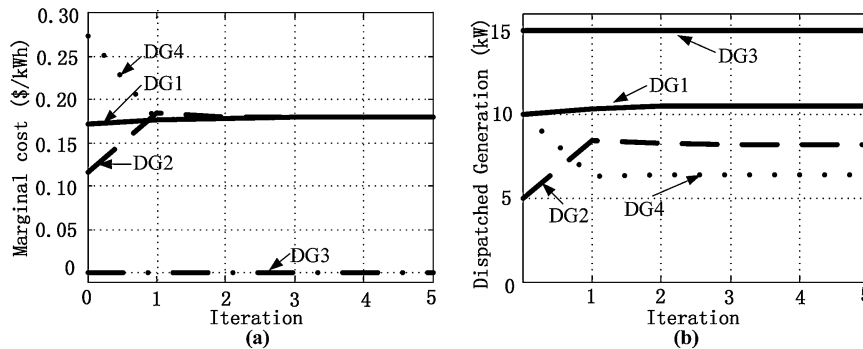


Fig. 12. (a) Convergence of marginal cost. (b) Dispatched values for the DGs.

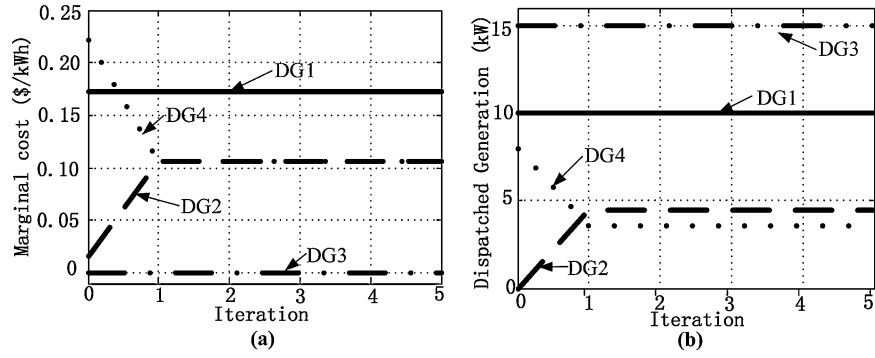


Fig. 13. (a) Convergence of marginal cost. (b) Dispatched values for the DGs.

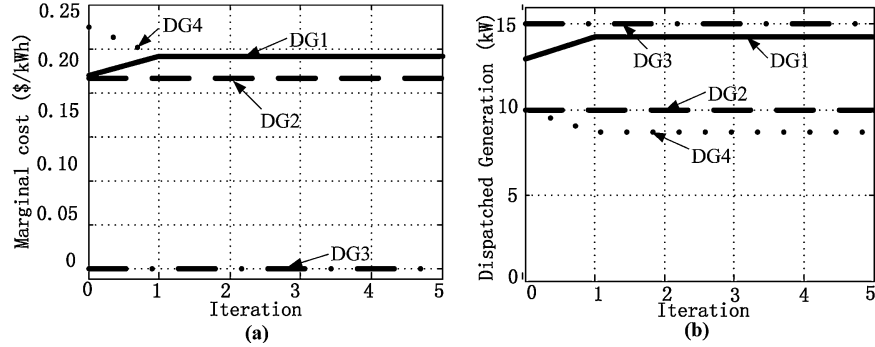


Fig. 14. (a) Convergence of marginal cost. (b) Dispatched values for the DGs.

dispatch values are: $Q_{DG1} = 10$ kvar, $Q_{DG2} = 0$ kvar, $Q_{DG3} = 0$ kvar, $Q_{DG4} = 3$ kvar.

At time $t = 1.0$ s, total real power demand increases to 58 kW, but total real power generation capacity is 55 kW. Therefore, MAS power balance control will shed 3 kW load. In this simulation, load 3 will be curtailed by 3 kW. Since load demand is larger than generation capacity, MAS will not conduct economic dispatch and all DGs are required to output maximum real power to reduce load shedding. Total reactive power demand is 18 kvar, still less than total reactive power capacity. The resulting reactive power dispatch values in this stage are: $Q_{DG1} = 8$ kvar, $Q_{DG2} = 6$ kvar, $Q_{DG3} = 0$ kvar, $Q_{DG4} = 4$ kvar.

At time $t = 1.5$ s, total real demand decreases to 48 kW shown in Table 5. Fig. 14 displays economic dispatch process. From the figure, it can be seen that real power output of DG2 reaches its upper limit, therefore its marginal cost is lower than the global value; while for DG1 and DG4, their output does not reach the limit, so their marginal cost converges. The optimal dispatch values are: $P_{DG1} = 14.29$ kW, $P_{DG2} = 10$ kW, $P_{DG3} = 15$ kW, $P_{DG4} = 8.71$ kW. Total reactive power demand decreases by 3 kvar; therefore Q_{DG1} drops to 5 kvar.

Fig. 15 displays simulation results for the DGs and loads. Fig. 16 shows output current transitions of DGs at the times of load changes.

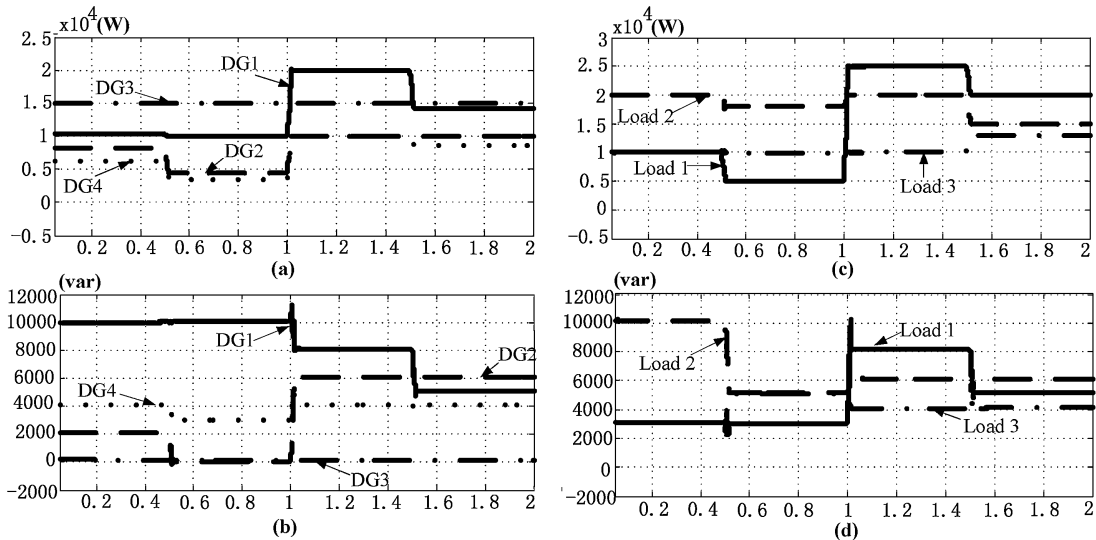


Fig. 15. (a) Real power output of DGs. (b) Reactive power output of DGs. (c) Real power consumed by load. (d) Reactive power consumed by load.

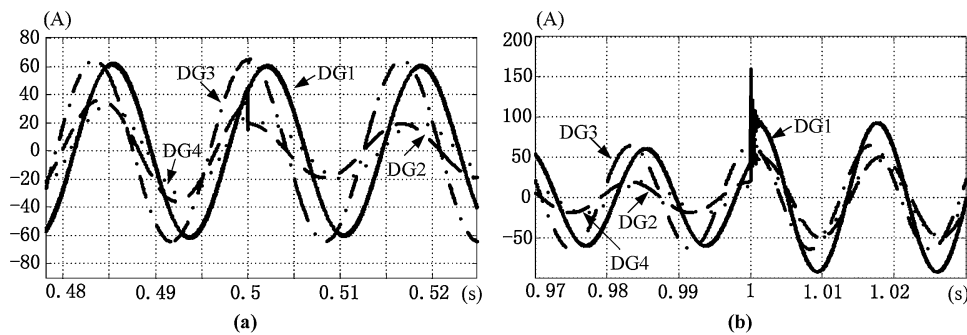


Fig. 16. (a) Output current transitions of DGs at 0.5 s. (b) Output current transitions of DGs at 1.0 s.

7. Conclusion

In this work, a comprehensive multi-level control architecture was described for master-slave organized microgrids with PE interfaced DGs. A new MAS power balance control strategy was presented that can accomplish exact power balance in three sweeps, regardless of system size. Unlike most of the power balance algorithm proposed in the literature, this algorithm achieves power balance in a non-iterative way; therefore, it does not encounter any convergence problem. Meanwhile, the information is designed to be transmitted in a parallel manner, which provides superior speed advantage to satisfy real-time control compared with the methods available in the literature. These advantages are clearly shown in Section 5.1, where it compares the speed of the proposed MAS algorithm with those exist in the literature. An economic dispatch algorithm suitable for the proposed decentralized MAS was also presented and demonstrated for microgrids that are amenable to optimal dispatch. By limited communications with neighbors, generator agents are able to optimize total generation cost in a non-increasing direction. This algorithm is robust and fault-tolerant that if one generator agent fails, only the generation at that single node is not optimized; while other agents can still collaborate to optimize total generation cost. In the presence of markets, the quadratic generation cost function can be replaced by bids, and the associated objective function, such as social welfare function, can also be optimized by the proposed MAS. The performance of the proposed strategy was demonstrated on several test systems. Its benefits, both in terms of speed as well as robustness in tracking time-varying loads, were demonstrated.

References

- [1] Z. Zhang, M.Y. Chow, Convergence analysis of the incremental cost consensus algorithm under different communication network topologies in a smart grid, *IEEE Trans. Power Syst.* 27 (4) (2012) 1761–1768.
- [2] T. Otani, H. Kobayashi, A SCADA system using mobile agents for a next-generation distribution system, *IEEE Trans. Power Del.* 28 (1) (2013) 47–57.
- [3] S. Yang, S. Tan, J. Xu, Consensus based approach for economic dispatch problem in a smart grid, *IEEE Trans. Power Syst.* 28 (4) (2013) 4416–4426.
- [4] M.C. Chandorkar, D.M. Divan, R. Adapa, Control of parallel connected inverters in standalone AC supply systems, *IEEE Trans. Ind. Appl.* 29 (1) (1993) 136–143.
- [5] R. Majumder, B. Chaudhuri, A. Ghosh, R. Ma, Improvement of stability and load sharing in an autonomous microgrid using supplementary droop control loop, *IEEE Trans. Power Syst.* 25 (2) (2010) 796–808.
- [6] S.J. Ahn, J.W. Park, I.Y. Chung, S.I. Moon, S.H. Kang, S.R. Nam, Power-sharing method of multiple distributed generators considering control modes and configurations of a microgrid, *IEEE Trans. Power Del.* 25 (3) (2010) 2007–2016.
- [7] F. Katiraei, M.R. Iravani, Power management strategies for a microgrid with multiple distributed generation units, *IEEE Trans. Power Syst.* 21 (4) (2006) 1821–1831.
- [8] S.K. Mazumder, M. Tahir, K. Acharya, Master-slave current-sharing control of a parallel dc-dc over a RF communication interface, *IEEE Trans. Ind. Electron.* 55 (1) (2008) 59–66.
- [9] D. Issicaba, N.J. Gil, J.A. Peças Lopes, Islanding operation of active distribution grids using an agent-based architecture, in: *Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*, IEEE PES, October 2010, 2010, pp. 1–8.
- [10] Y. Xu, W. Liu, Novel multiagent based load restoration algorithms for microgrids, *IEEE Trans. Smart Grid* 2 (1) (2011) 152–161.
- [11] Y. Xu, W. Liu, Stable multi-agent-based load shedding algorithm for power systems, *IEEE Trans. Power Syst.* 26 (4) (2011) 2006–2014.
- [12] S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A.L. Dimeas, N.D. Hatziairgiou, F. Ponci, T. Funabashi, Multi-agent systems for power engineering applications—Part I: Concepts, approaches, and technical challenges, *IEEE Trans. Power Syst.* 22 (4) (2007) 1743–1752.
- [13] S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A.L. Dimeas, N.D. Hatziairgiou, F. Ponci, T. Funabashi, Multi-agent systems for power engineering applications—Part II: Technologies, standards, and tools for building multi-agent systems, *IEEE Trans. Power Syst.* 22 (4) (2007) 1753–1759.
- [14] S. Talukdar, V.C. Ramesh, A multi-agent technique for contingency constrained optimal power flows, *IEEE Trans. Power Syst.* 9 (2) (1994) 855–861.
- [15] N.P. Yu, C.C. Liu, J. Price, Evaluation of market rules using multi-agent system method, *IEEE Trans. Power Syst.* 25 (1) (2010) 470–479.
- [16] P. Wei, Y. Yan, Y. Ni, J. Yen, F.E. Wu, A decentralized approach for optimal wholesale cross-border trade planning using multi-agent technology, *IEEE Trans. Power Syst.* 16 (4) (2001) 833–838.
- [17] J. Mitra, N. Cai, M.Y. Chow, S. Kamalasadan, W. Liu, W. Qiao, S.N. Singh, A.K. Srivastava, S.K. Srivastava, G.K. Venayagamoorthy, Z. Zhang, Intelligent methods for smart microgrids, in: *Proc. of IEEE Power & Energy Soc. General Meeting*, July, 2011, pp. 1–8.
- [18] J. Jung, C.C. Liu, S.L. Tanimoto, V. Vittal, Adaptation in load shedding under vulnerable operating conditions, *IEEE Trans. Power Syst.* 17 (4) (2002) 1199–1205.
- [19] A.L. Dimeas, N.D. Hatziairgiou, Operation of a multi-agent system for microgrid control, *IEEE Trans. Power Syst.* 20 (3) (2005) 1447–1455.
- [20] N. Cai, J. Mitra, A decentralized control architecture for a microgrid with power electronic interfaces, in: *Proc. of 42th Annual North American Power Symposium*, September, 2010, pp. 1–8.
- [21] T. Logenthiran, D. Srinivasan, A.M. Khambadkone, Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system, *Electric Power Syst. Res.* 81 (1) (2011) 138–148.
- [22] F. Ren, M. Zhang, D. Sutanto, A multi-agent solution to distribution system management by considering distributed generators, *IEEE Trans. Power Syst.* 28 (2) (2013) 1442–1451.
- [23] C.M. Colson, M.H. Nehrir, Comprehensive real-time microgrid power management and control with distributed agents, *IEEE Trans. Smart Grid* 4 (1) (2013) 617–627.
- [24] Y.J. Zhang, Z. Ren, Real-time optimal reactive power dispatch using multi-agent technique, *Electric Power Syst. Res.* 69 (2004) 259–265.
- [25] S.D.J. McArthur, S.M. Strachan, The design of a multi-agent transformer condition monitoring system, *IEEE Trans. Power Syst.* 19 (4) (2004) 1845–1852.
- [26] E.E. Mangina, S.D.J. McArthur, J.R. McDonald, A. Moyes, A multi-agent system for monitoring industrial gas turbine start-up sequences, *IEEE Trans. Power Syst.* 16 (3) (2001) 396–401.
- [27] S.D.J. McArthur, C.D. Booth, J.R. McDonald, I.T. McFadyen, An agent-based anomaly detection architecture for condition monitoring, *IEEE Trans. Power Syst.* 20 (4) (2005) 1675–1682.
- [28] H. Wan, K.K. Li, K.P. Wong, A multi-agent approach to protection relay coordination with distributed generators in industrial power distribution system, in: *Proc. of 40th IAS Annual Meeting Ind. Appl.*, October, 2005, pp. 830–836.
- [29] F. Daneshfar, H. Bevrani, Load-frequency control: a GA-based multi-agent reinforcement learning, *IET Gen. Transm. Distrib.* 4 (1) (2010) 13–26.
- [30] T. Nagata, H. Sasaki, A multi-agent approach to power system restoration, *IEEE Trans. Power Syst.* 17 (2) (2002) 457–462.
- [31] J.M. Solanki, S. Khushalani, N.N. Schulz, A multi-agent solution to distribution systems restoration, *IEEE Trans. Power Syst.* 22 (3) (2007) 1026–1034.
- [32] N. Cai, X. Xu, J. Mitra, A hierarchical multi-agent control scheme for a black start-capable microgrid, in: *Proc. of IEEE Power & Energy Soc. General Meeting*, July, 2011, pp. 1–6.
- [33] M.E. Baran, I.M. Markabi, A multi-agent-based dispatching scheme for distributed generators for voltage support on distribution feeders, *IEEE Trans. Power Syst.* 22 (1) (2007) 52–59.
- [34] M.E. Elkhatabi, R. El-Shatshat, M.M.A. Salama, Novel coordinated voltage control for smart distribution networks with DG, *IEEE Trans. Smart Grid* 2 (4) (2011) 598–605.

- [35] H. Ni, G.T. Heydt, L. Mili, Power system stability agents using robust wide area control, *IEEE Trans. Power Syst.* 17 (4) (2002) 1123–1131.
- [36] Z. Zhang, X. Ying, M.Y. Chow, Decentralizing the economic dispatch problem using a two-level incremental cost consensus algorithm in a smart grid environment, in: *Proc. of 43th Annual North American Power Symposium*, August, 2011, pp. 1–7.
- [37] A. Abbasy, S.H. Hosseini, A novel multi-agent evolutionary programming algorithm for economic dispatch problems with non-smooth cost functions, in: *Proc. of IEEE Power Eng. Soc. Gen. Meeting*, June, 2007, pp. 1–7.
- [38] S.K. Mazumder, K. Acharya, M. Tahir, Joint optimization of control performance and network resource utilization in homogeneous power networks, *IEEE Trans. Ind. Electron.* 56 (5) (2009) 1736–1745.
- [39] J.D. Glover, M.S. Sarma, T. Overbye, *Power System Analysis and Design*, 5th ed, Cengage Learning, Stamford, CT, 2011.
- [40] Power Systems Test Case Archive, 2013, available at: <http://www.ee.washington.edu/research/pstca>
- [41] R.D. Zimmerman, C. Murillo-Sánchez, *MATPOWER User's Manual*, 2013, available at: <http://www.pserc.cornell.edu/matpower/>
- [42] JADE, Java agent development framework, available at: <http://jade.tilab.com/>
- [43] F. Bellifemine, G. Caire, D. Greenwood, *Developing Multi-agent Systems with JADE*, Wiley, New York, 2007.
- [44] FIPA, Foundation of intelligent physical agents, available at: <http://www.fipa.org/>
- [45] D.N. Zmood, D.G. Holmes, Stationary frame current regulation of PWM inverters with zero steady-state error, *IEEE Trans. Power Electron.* 18 (3) (2003) 814–822.