Distributed Control of Hybrid AC/DC Microgrids: from Synchronous to Asynchronous Algorithms

Zhaojian Wang, Shengwei Mei, Fellow, IEEE, Feng Liu, Peng Yi, Ming Cao

Abstract—Hybrid AC/DC microgrids (MGs) have become a promising manner for better interconnection of various kinds of distributed generators that are inherently AC or DC electric sources. This paper addresses the distributed asynchronous power control problem of hybrid microgrids, considering imperfect communication due to non-identical sampling rates and communication delays, etc. To this end, we first formulate the optimal power control problem of MGs and devise a synchronous algorithm. Then, we analyze the impact of asynchrony on optimal power control and propose an asynchronous iteration algorithm based on the synchronous version. By introducing a random clock at each iteration, different types of asynchrony are fitted into a unified framework, where the asynchronous algorithm is converted into a fixed-point problem, leading to a convergence proof. We further provide an upper bound estimation of the time delay in communication. Finally, a benchmark MG is utilized to verify the effectiveness and advantages of the proposed algorithm.

Index Terms—Asynchronous control, distributed power control; hybrid AC/DC microgrids, time delay.

I. INTRODUCTION

Microgrids (MGs) are clusters of distributed generators (DGs), energy storage systems and loads, which are generally categorized into three types: AC, DC and hybrid AC/DC MGs [1], [2]. A hybrid AC/DC MG has the great advantage of reducing processes of multiple inverse conversions in the involved individual AC or DC grid [3], [4]. In this paper, we address the distributed power control problem of hybrid AC/DC MGs considering asynchrony.

Traditionally, a hierarchical control structure is utilized in MGs for power control, which is composed of primary control, secondary control and tertiary control [5], [6], usually in a centralized way. Such a centralized control architecture, however, may face great challenges raised by ever-increasing uncertain and volatile renewable generations that require fast response of controllers [7]. On the other hand, as MGs usually belong to different owners, privacy concerns may prevent the control center acquiring information from individual MGs. In this context, breaking the hierarchy of MG control architecture becomes an emerging research topic, supported by the new idea that real-time coordination could be embedded in the local steady-state optimization of individual MGs by exchanging information only between neighboring MGs. This essentially advocates a distributed control paradigm [8], especially the idea of reverse-engineering [9], [10].

Different distributed strategies have been developed in literatures for optimal real-time coordination, which can roughly be divided into two categories in terms of methodology: consensus based methods [11]–[17], and (sub)gradient based decomposition methods [18]–[21]. In the consensus based control, the agents, such as generators, loads, MGs, or other forms of local systems, manage to estimate the global variable using a consensus algorithm [22]. Specifically, in power systems, the global variable could be generation ratio and marginal cost. The former implies that all generators have the same generation ratio with respect to its maximal capability [11]–[14]. The latter implies that all generators share the same marginal cost, and hence the generation configuration is economically optimal [15], [16], [23]. Even though the consensus methods are easy to be implemented, they are difficult to address complicated (global) constraints. In this situation, (sub)gradient based decomposition methods could be applied. The main idea is that an optimization problem is formulated first, then the corresponding (sub)gradient algorithm solving the problem can render a dynamical controller. Combining with the power system dynamics, the closed-loop system converges to an equilibrium that corresponds to the optimal solution of the original optimization problem. The most commonly used method is the primal-dual gradient algorithm [18]–[21], where power system dynamics serve as the primal update while controller serves as the dual update.

Although both the aforementioned methods have achieved great success, most of them have considered only synchronous distributed control (except for [17]). Therefore, all MGs must be assumed to carry out computation simultaneously, implying that a global clock is necessary to ensure the instants for control actions getting strictly synchronized [24]. This is computationally inefficient and impractical since in each iteration all MGs have to wait for the slowest one to finish before executing their local actions in the next iteration. Meanwhile, all MGs are required to utilize the information at synchronized computation rounds. Hence, the synchrony requirement limits the application of distributed control in power systems. In fact, asynchrony is a fundamental feature of power systems. For example, time delays in communication lines always exist, which sometimes could be very long. In addition, the sampling rates of different controllers vary greatly. Thus, it is significant to investigate the controller that adapts to asynchronous implementation.

In this paper, asynchrony is considered in the active power control of hybrid AC/DC MGs, which may result from different sampling rates of controllers, delays in communications, and so on. By introducing a random clock, different kinds of asynchrony are fitted into one unified form, i.e., the time interval between two consecutive iterations (Without confusion, it is also called time delay in the paper). Then, an asynchronous algorithm is devised. To prove its convergence,
the algorithm is transformed into a fixed point problem with a nonexpansive operator. Finally, we give an upper bound of time delay that guarantees the convergence. This paper is partly motivated by [17], which investigates the distributed power sharing problem in hybrid AC/DC MGs. The main difference between this paper and [17] lies in two aspects: 1) we develop a distributed asynchronous method to cope with different types of asynchrony; and 2) we obtain economically optimal power generations of each MGs different from proportional sharing with respect to droop coefficients in [17]. Main contributions of this paper are as follows.

- A distributed asynchronous algorithm is devised to solve the optimal power control problem of hybrid AC/DC MGs, which only requires communications between neighbors. Moreover, different types of asynchrony, such as random time delays and non-identical sampling rates, are unified by introducing a random clock at each iteration;
- The distributed algorithm is converted into a fixed-point iteration of a nonexpansive operator, leading to a convergence proof of the asynchronous algorithm;
- The upper bound of communication delay that guarantees the convergence is estimated, which is approximately proportional to the square root of the number of MGs for a fixed step-size. Similarly, with a fixed time interval, the upper bound estimation of step size is provided.

The rest of this paper is organized as follows. In Section II, some notations and preliminaries are introduced. Section III formulates the power dispatch problem in hybrid MGs and proposes the synchronous algorithm. In Section IV, different types of asynchrony are introduced and an asynchronous algorithm is proposed. The optimality of its equilibrium point and convergence of the asynchronous algorithm are proved in Section V. The implementation method in hybrid MGs is introduced in Section VI. We confirm the performance of the controller via simulations on a benchmark low voltage MG system in Section VII. Section VIII concludes the paper.

II. NOTATIONS AND PRELIMINARIES

Notations: A hybrid MG system is composed of a cluster of AC and DC MGs connected by lines. Each MG is treated as a bus with both generation and load. Denote AC MGs by \( N_{ac} = \{1, 2, \ldots, n_{ac}\} \), and DC MGs by \( N_{dc} = \{n_{ac} + 1, n_{ac} + 2, \ldots, n_{ac} + n_{dc}\} \). Then the set of MG buses is \( N = N_{ac} \cup N_{dc} \). Let \( E \subseteq N \times N \) be the set of lines, where \((i, k) \in E\) if MGs \( i \) and \( k \) are connected directly. Then the overall system is modeled as a connected graph \( G := (N, E) \). Besides the physical connection among MGs, we also define a communication graph for MGs. Denote by \( N_i \) the set of informational neighbors of MG \( i \) over the communication graph, implying MGs \( i, j \) can communicate if and only if \( j \in N_i \). The cardinality of \( N_i \) is denoted by \( |N_i| \). The communication graph is also assumed to be undirected and connected, which could be different from the physical graph. Denote by \( L \) the Laplacian matrix of communication graph.

Preliminaries: In this paper, \( \mathbb{R}^n \) (\( \mathbb{R}^n_+ \)) is the \( n \)-dimensional (nonnegative) Euclidean space. For a column vector \( x \in \mathbb{R}^n \) (matrix \( A \in \mathbb{R}^{m \times n} \)), \( x^T (A^T) \) denotes its transpose. For vectors \( x, y \in \mathbb{R}^n \), \( x^T y = \langle x, y \rangle \) denotes the inner product of \( x, y \). \( \|x\| = \sqrt{x^T x} \) denotes the Euclidean norm of \( x \). For a positive definite matrix \( G \), \( \|x\|_G = \sqrt{Gx^T x} \). Similarly, the \( G \)-matrix induced norm \( \|x\| \leq \sqrt{Gx^T x} \). Use \( I \) to denote the identity matrix with proper dimensions. For a matrix \( A = [a_{ij}] \), \( a_{ij} \) stands for the entry in the \( i \)-th row and \( j \)-th column of \( A \). Use \( \prod_{i=1}^n \Omega_i \) to denote the Cartesian product of the sets \( \Omega_i \), \( i = 1, \ldots, n \).

Given a collection of \( y_i \) for \( i \) in a certain set \( Y \), \( y \) denotes the column vector \( y := (y_i, i \in Y) \) with a proper dimension with \( y_i \) as its components.

Define the projection of \( x \) onto a set \( \Omega \) as

\[
P_{\Omega}(x) = \arg \min_{y \in \Omega} \|x - y\| \tag{1}
\]

Use \( Id \) to denote the identity operator, i.e., \( Id(x) = x \), \( \forall x \). Define \( N_{\Omega}(x) = \{v | \langle x, v \rangle \leq 0, \forall y \in \Omega \} \). We have \( P_{\Omega}(x) = (Id + N_{\Omega})^{-1}(x) \) [25], [26 Chapter 23.1].

For a single-valued operator \( T : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \), a point \( x \in \Omega \) is a fixed point of \( T \) if \( T(x) = x \). The set of fixed points of \( T \) is denoted by \( Fix(T) \). \( T \) is nonexpansive if \( \|T(x) - T(y)\| \leq \|x - y\|, \forall x, y \in \Omega \). \( T \) is called \( \alpha \)-averaged if there exists a nonexpansive operator \( R \) such that \( T = (1 - \alpha) Id + \alpha R \). We use \( A(\alpha) \) to denote the class of \( \alpha \)-averaged operators. For \( \beta \in \mathbb{R}^+_1 \), \( T \) is called \( \beta \)-cocoercive if \( \beta T \in A(\frac{\beta}{2}) \).

III. SYNCHRONOUS DISTRIBUTED ALGORITHM

In this section, we introduce the economic dispatch problem in MGs and propose a synchronous algorithm.

A. Economic dispatch model

The power dispatch is to achieve the power balance in MGs while minimizing the generation cost, which can be formulated as the following optimization problem

\[
\min_{P^g} \sum_{i \in N} f_i(P^g_i) \tag{2a}
\]

s.t. \( \sum_{i \in N} P^g_i = \sum_{i \in N} P^d_i \) \( \leq \) \( P^g \leq P^d \) \( \tag{2b} \)

where \( f_i(P^g_i) = \frac{1}{2} a_i (P^g_i)^2 + b_i P^g_i \), with \( a_i > 0, b_i > 0 \), \( P^g \), \( P^d \) are the power generation and load demand of MG \( i \) respectively. \( P^g, P^d \) are the lower and upper bounds of \( P^g \) respectively. The objective function (2a) is to minimize the total generation cost of the MGs. Constraint (2b) is the power balance over MGs. And (2c) is the generation limit of each MG.

For the optimization problem (2), we make the following assumption.

Assumption 1. The Slater’s condition [27] Chapter 5.2.3 of (2) holds, i.e., problem (2) is feasible due to affine constraints.

In practice, the load demand of each MG, \( P^d \), cannot be measured accurately. To circumvent such an obstacle in design, we alternatively introduce a set of new variables, \( P^d \), and reformulate (2) as the following equivalent problem:

\[
\min_{P^d} \sum_{i \in N} f_i(P^d_i) \tag{3a}
\]
where $\hat{P}_d^i$ is the virtual load demand supplied by MG $i$ in steady state. It is a constant that equals the actual generation of MG $i$ in steady state, satisfying $\sum_{i\in N} P_t^i = \sum_{i\in N} \hat{P}_d^i$. Similar treatment is also adopted in [28].

Note that the power balance constraint (2b) only requires that all the MGs supply the total load even if one does not know which loads are supplied exactly by which MGs. Hence, we treat virtual load demands $\hat{P}_d^i$ as the effective demands supplied by MG $i$. Then we easily have the following Lemma.

**Lemma 1.** The problems (2) and (3) have the same optimal solutions.

### B. Synchronous Algorithm

The Lagrangian function of (3) is

$$L = \sum_{i\in N} f_i(P_t^i) + \mu \left( \sum_{i\in N} P_t^i - \sum_{i\in N} \hat{P}_d^i \right) + \sum_{i\in N} \gamma_i \left( P^g_i - P_t^i \right) + \sum_{i\in N} \gamma_i^d \left( P^d_i - \hat{P}_d^i \right)$$

where $\mu, \gamma_i, \gamma_i^d$ are Lagrangian multipliers. Here $\mu$ is a global variable, but will be estimated by individual MGs locally.

Define the sets

$$\Omega_i := \left\{ P_t^i \mid P_t^i \leq P^g_i \leq P^d_i \right\}, \quad \Omega = \prod_{i=1}^N \Omega_i$$

Then, we give the synchronous distributed algorithm for power dispatch (SDPD). In this case, the update of agent $i$ at iteration $k$ is given as

$$\mu_{i,k} = \mu_{i,k-1} + \sigma_\mu \left( -\sum_{j \in N_i} (\mu_{i,k-1} - \mu_{j,k}) + P^g_{i,k-1} - \hat{P}_d^i \right)$$

$$\hat{P}_d^i = P_{\Omega_i} \left( P^g_i - \sigma_g \left( f_i(P_t^i) + 2\hat{\mu}_{i,k-1} - \mu_{i,k} \right) \right)$$

$$\mu_{i,k+1} = \mu_{i,k} + \eta \left( \hat{\mu}_{i,k} - \mu_{i,k} \right)$$

$$P^g_{i,k+1} = P^g_{i,k} + \eta \left( \hat{P}_d^i - P^g_i \right)$$

where $\sigma_\mu, \sigma_g, \eta$ are positive constants, and $\sigma_\mu, \sigma_g$ are supposed to be chosen such that $\Phi$ in (10) (given in Section V.B) is positive definite.

Here no time delay is considered, and in each iteration $k$, each MG has to wait for the slowest one to finish its $k$th iteration before proceeding to iteration $k+1$. Based on (5), we have the following synchronous algorithm.

**Algorithm 1 SDPD**

**Input:** For MG $i$, the input is $\mu_{i,0} \in \mathbb{R}^\alpha, P_{i,0}^g \in \Omega_i$.

**Iteration at $k$:** MG $i$ updates its local variables as follows:

**Step 1:** Reading phase

Get $\mu_{j,k}$ from its neighbours’ output cache.

**Step 2:** Computing phase

Calculate $\hat{\mu}_{i,k}$ and $P^g_{i,k}$ according to (5a) and (5b) respectively.

Update $\mu_{i,k+1}$ and $P^g_{i,k+1}$ according to (5c) and (5d) respectively.

**Step 3:** Writing phase

Write $\mu_{i,k+1}$ to its output cache and $P^g_{i,k+1}$, $\mu_{i,k+1}$ to its local storage. Increase $k$ to $k+1$.

In Algorithm 1, the virtual power $\hat{P}_d$ is still unknown. We will provide a practical method to estimate $P^g_i - \hat{P}_d^i$ instead of directly measuring $\hat{P}_d^i$ in the implementation, as explained in Section VI. Later in Section IV, we will show that the SDPD is simply a special case of the asynchronous algorithm. Therefore, its properties, such as the optimality of the equilibrium point and the convergence, are immediate consequence of the results of asynchronous algorithm, which are skipped here due to the space limitation.

### IV. DISTRIBUTED ASYNCHRONOUS ALGORITHM

In this section, we first introduce several typical types of asynchrony existing in MGs. Then, we devise an asynchronous algorithm by modifying Algorithm 1.

#### A. Asynchrony in Microgrids

In Algorithm 1, each MG gathers information, computes locally and conveys new information to its neighbors over the communication graph. In this process, asynchrony may arise in each step. When gathering information, individual MGs may have different sampling rates, which results in non-identical computation rates accordingly. In addition, other imperfect communication situations such as time delay caused by congestion or even failure are very common in power systems, which essentially result in asynchrony.

In synchronous computation, an MG has to wait for the slowest neighbor to complete the computation by inserting certain idle time. Communication delay, congestion or even package loss can further lengthen the waiting time. This process is illustrated in Fig.1(a). Thus, the slowest MG and communication channel may cripple the system in the synchronous execution. In contrast, the MGs with asynchrony computation do not need to wait and computes continuously with little idling, as shown in Fig.1(b). Even if some of its neighbors fail to update in time, the MG can use the previously stored information. That means, the MG could execute an iteration without the latest information from its neighbors.

#### B. Asynchronous Algorithm

In this subsection, we propose an asynchronous distributed algorithm for power dispatch (ASDPD) based on Algorithm 1. Different from the iteration number $k$ in (5), here each MG has its own iteration number $k_i$, implying that a local clock is used instead of the global clock. At each iteration $k_i$, MG $i$ computes in the following way.

$$\hat{\mu}_{i,k_i} = \mu_{i,k_i-\tau_{k_i}} + \sigma_\mu \left( -\sum_{j \in N_i} (\mu_{i,k_i-\tau_{k_i}} - \mu_{j,k_i-\tau_{j_i}}) + P^g_{i,k_i-\tau_{k_i}} - \hat{P}_d^i \right)$$

Fig. 1. Synchronous versus asynchronous computation
\( \hat{P}_{i,k+1}^g = \mathcal{P}_\Omega \left( P_{i,k+1}^g - \sigma_g \left( f_i^g \left( P_{i,k+1}^g \right) \right) + 2 \mu_{i,k} \right) \)  
(6b)

\[ \mu_{i,k+1} = \mu_{i,k} - \tau_{i,k} + \eta_k \left( \mu_{i,k} - \mu_{i,k-1} \right) \]  
(6c)

\[ P_{i,k+1}^g = P_{i,k+1}^g - \tau_{i,k} + \eta_k \left( \hat{P}_{i,k}^g - P_{i,k+1}^g \right) \]  
(6d)

where \( \tau_{i,k}^k \) is the random time delay. Denote \( w = (\mu, P^g) \). 

\( w_{i,k+1}^g = \) the state of MG \( i \) at iteration \( k \), 

\( w_{j,k+1}^g = \) the latest information obtained from MG \( j \). Considering each MG has its own local clock, we have the following asynchronous algorithm.

Algorithm 2 ASDPD

Input: For MG \( i \), the input is \( \mu_{i,0} \in \mathbb{R}^n \), \( P_{i,0}^g \in \Omega \).

Iteration at \( k \): Suppose MG \( i \)'s clock ticks at time \( k \), then MG \( i \) is activated and updates its local variables as follows:

Step 1: Reading phase
Get \( \mu_{j,k+1}^g \) from its neighbours’ output cache.

Step 2: Computing phase
Calculate \( \hat{\mu}_{i,k} \) and \( \hat{P}_{i,k}^g \) according to (6a) and (6b) respectively.
Update \( \mu_{i,k+1} \) and \( P_{i,k+1}^g \) according to (6c) and (6d) respectively.

Step 3: Writing phase
Write \( \mu_{i,k+1} \) to its output cache and \( \mu_{i,k+1}, P_{i,k+1}^g \) to its local storage. Increase \( k \) to \( k+1 \).

Remark 1. In Algorithm 2 if MG \( i \) is activated, it will read the latest information from its neighbors. Even if some neighbors are not accessible in time due to communication issue, it can still execute the iteration by using the previous information stored in its input cache. Despite asynchrony caused by different reasons, their effect can be characterized by the time interval between two successive iterations. Thus, our algorithm can admit different types of asynchrony. It is obvious that the synchronous algorithm SDPD is a special case of ASDPD by simply letting \( \tau_{i,k}^k = 0 \).

V. OPTIMAlITY AND CONVERGENCE ANALYSIS

In this section, we analyse the optimality of the equilibrium point of dynamic system (6), as well as the convergence of Algorithm 2. To this end, we need to introduce a sequence of global iteration numbers that serve as a reference global clock to unify the local iterations of individual MGs in a coherence manner [29]. Note that the global clock is only used for convergence analysis, but not required in ASDPD.

Specifically, we queue \( k_i \) of all MGs in the order of time, and use a new number \( k \) to denote the \( k \)th iteration in the queue. This treatment is shown in Fig. 2 by taking two MGs as an example. Suppose that, at the iteration \( k \), the probability that MG \( i \) is activated to update its local variables follows a uniform distribution. Hence, each MG is activated with the same probability, which simplifies the convergence proof.

To prove the convergence, we first convert the synchronous algorithm to a fixed-point iteration with an averaged operator. Then a nonexpansive operator is constructed, leading to the convergence results of the asynchronous algorithm. Finally, we provide an upper bound of the time delay.

A. Optimality of the equilibrium point

Considering dynamic system (6), we give the following definition of its equilibrium point.

Definition 1. A point \( w^* = (w^*_i, i \in \mathcal{N}) = (\mu^*_i, P^g_i) \) is an equilibrium point of system (6) if \( \lim_{k \to +\infty} w_{k} = w^* \) holds for all \( i \).

Then, we have the following result.

Theorem 2. Suppose Assumption 1 holds. If \( P^g_i = \hat{P}_i^d \) in steady state, then the component \( P^g_i, \mu^*_i \) of the equilibrium point \( w^* \) is the primal-dual optimal solution to (2).

The proof of Theorem 2 can be found in Appendix A. In Section VI, we further provide a practical method to estimate \( \hat{P}_d \) such that the condition \( P^g_i = \hat{P}_i^d \) holds in steady state.

B. Algorithm Reformulation

Equations (5a) – (5d) can be rewritten in a compact form

\[ \hat{\mu}_k = \mu_k + \sigma \mu \left( -L \cdot \mu_k + P_k^d - \hat{P}_d \right) \]  
(7a)

\[ \hat{P}_k^g = \mathcal{P}_\Omega \left( P_k^g - \sigma_g \left( \nabla f(P_k^g) + 2\hat{\mu}_k - \mu_k \right) \right) \]  
(7b)

\[ \mu_{k+1} = \mu_k + \eta_k (\hat{\mu}_k - \mu_k) \]  
(7c)

\[ P_{k+1}^g = P_k^g + \eta_k \left( \hat{P}_k^g - P_k^g \right) \]  
(7d)

where, \( \nabla f(P_k^g) \) is the gradient of \( f(P_k^g) \), and \( L \) is the Laplacian of the communication graph.

Next we show that (7a) – (7d) can be converted into a fixed-point iteration problem with an averaged operator [25], [30]. Equation (7a) is equivalent to

\[ -L \cdot \mu_k - P_d = -P_k^g + \sigma^{-1}_\mu (\hat{\mu}_k - \mu_k) \]

\[ = -P_k^g + \sigma^{-1}_\mu (\hat{\mu}_k - \mu_k) + \hat{P}_d - P_k^g \]  
(8)

From the fact that \( \mathcal{P}_\Omega(x) = (1d + N_\Omega)^{-1}x \), (7b) can be rewritten as \( \hat{P}_k^g = (1d + N_\Omega)^{-1} \left( P_k^g - \sigma_g (\nabla f(P_k^g) + 2\hat{\mu}_k - \mu_k) \right) \), or equivalently,

\[ -\nabla f(P_k^g) = 2\hat{\mu}_k - \mu_k + N_\Omega(\hat{P}_k^g) + \sigma_g^{-1}(\hat{P}_k^g - P_k^g) \]  
(9)

Then, (7a) – (7d) are rewritten as

\[ \left[ \begin{array}{c} \nabla \hat{\mu}_k + P_d \\ \nabla f(P_k^g) \end{array} \right] = \left[ \begin{array}{c} \hat{P}_k^g \\ \hat{\mu}_k - \sigma^{-1}_\mu \hat{\mu}_k - \mu_k \end{array} \right] + \hat{\mu}_k - \mu_k \]  
(10)

where \( \Phi := \left[ \begin{array}{c} \sigma^{-1}_\mu I \\ I \\ \sigma_g^{-1} I \end{array} \right] \).

Define the following two operators

\[ B : \left[ \begin{array}{c} \mu \\ P^g \end{array} \right] \mapsto \left[ \begin{array}{c} \nabla \hat{\mu}_k + P_d \\ \nabla f(P_k^g) \end{array} \right] \]  
(11)
Lemma 3. Take have the following result about the operator $T$ simultaneously, which can be modeled as two or more iterations in analysis. In implementation, we allow that two or more MGs are activated in parallel. And then the results in [31] can be applied.

In this subsection, we investigate the convergence of ASDPD, as we explain in the next subsection.

C. Convergence analysis of asynchronous algorithm

In this subsection, we investigate the convergence of ASDPD. The basic idea is to treat ASDPD as a randomized block-coordinate fixed-point iteration problem with delayed information. And then the results in [31] can be applied.

Define vectors $\phi_i \in \mathbb{R}^{2n}, i \in \mathcal{N}$. The $j$th entry of $\phi_i$ is denoted by $[\phi_i]_j$. Define $[\phi_i]_j = 1$ if the $j$th coordinate of $w$ is also a coordinate of $w^i$, and $[\phi_i]_j = 0$, otherwise. Denote by $\varphi$ a random variable (vector) taking values in $\phi_i, i \in \mathcal{N}$. Then \textbf{Prob}($\varphi = \phi_i) = 1/n$ also follows a uniform distribution. Let $\varphi_k$ be the value of $\varphi$ at the $k$th iteration. Then, a randomized block-coordinate fixed-point iteration for [14] is given by

$$w_{k+1} = w_k + \eta_k \varphi_k \circ (T(w_k) - w_k)$$

where $\varphi$ is the Hadamard product of two matrices. Here, we assume only one MG is activated at each iteration without loss of generality.\footnote{Note that this model helps formulate the algorithm and analyze its convergence. In implementation, we allow that two or more MGs are activated simultaneously, which can be modeled as two or more iterations in analysis.}

The proof of Lemma 3 can be found in Appendix B. So far, we convert the synchronous algorithm into a fixed-point iteration problem with an averaged operator (see (16)). Moreover, we also construct a nonexpansive operator $\mathcal{R}$ that enables us to prove the convergence of the asynchronous algorithm ASDPD, as we explain in the next subsection.

VI. IMPLEMENTATION

In this section, we introduce how to implement the ASDPD in both AC and DC MGs. First, we introduce practical methods to estimate $P_{i,k}^d - P_{i}^d$ (instead of $P_i^d$) in AC and DC MGs. Then, the optimality of results using such estimation is proved.

A. Control of AC MGs

In AC MGs, we use the frequency deviation $\omega_i - \omega_{ref}$ to estimate the power imbalance in each MG, i.e., $P_{i,k}^g - P_{i,k}^d$ in (6a). The power balance of AC bus $i$ is

$$M_i \dot{\omega}_i = P_{i,k}^g - P_{i,k}^d - D_i \dot{\omega}_i - \sum_{j \in \mathcal{N}_i} P_{ij}, \ i \in \mathcal{N}_{ac}$$

where, $M_i > 0$, $D_i > 0$ are constants, and $P_{ij}$ is the line power from bus $i$ to bus $j$. This model is suitable for both synchronous generators and converters [32]. Invoking the definition of $\dot{P}_{i,k}^d$, (20) can be rewritten as

$$P_{i,k}^g - P_{i,k}^d = M_i \dot{\omega}_i + D_i \dot{\omega}_i, \ i \in \mathcal{N}_{ac}$$

It can be used to estimate the power imbalance in the AC MG, i.e., $P_{i,k}^g - P_{i,k}^d$ in (6a). This method is also used in [20], [28]. By this control structure, the asynchronous algorithm [2] is integrated to the real-time control in AC MGs.
Asynchronous Information

The error between $P$ are sent to the asynchronous power control level. Algorithm
reactive power control, where the active power and frequency
are measured as the input of the primary power control level. The
primary power control level includes three loops, i.e.,
the current loop, the voltage loop and the power loop. In the
power loop, droop control is utilized for both active power and
reactive power control, where the active power and frequency
are sent to the asynchronous power control level. Algorithm
2 is integrated in the asynchronous control level, where the
asynchronous information from neighboring MGs is utilized.
The output $P_{i,k+1}$ is the reference of the active power control. The
error between $P_{i,k+1}$ and the measured active power is
fed to the primary power control via an integral operator. Other
outputs $\mu_{i,k+1}$ is written to its output cache, which are sent
to its neighbors via the communication network.

B. Control of DC MGs

In DC MGs, DC capacitors are utilized to maintain the voltage
stability of DC buses [34]. Then, the model of DC MGs can be
simplified (see Fig4). The power balance on DC bus $i$ is
\[ V^{dc}_i C_i V^{dc}_i = P^g_i - P^d_i - \sum_{j \in N_i} P_{ij}, \quad i \in N_{dc} \]
(22)
where, $C_i$ is the capacitor connected to the DC bus; $V^{dc}_i$
the voltage of the DC bus. Invoking the definition of $P^d_i,$ equation
22 can be rewritten as
\[ P^g_i - P^d_i = V^{dc}_i C_i V^{dc}_i, \quad i \in N_{dc} \]
(23)
It can be used to estimate the power imbalance in the DC
MG, i.e., $P_{i,k}^g - P_{i,k}^d$ in (6a). Thus, we only need to measure
the voltage and the line power injected to the DC bus, which
is much easier to implement. The control diagram is similar to
that in Fig3, which is omitted here. Then, the asynchronous
algorithm 2 is integrated to the real-time control in DC MGs.

By the implementation method, we claim $P_{i,k}^g = P_{i,k}^d$ holds
in steady state and the equilibrium point is also optimal with
respect to the optimization problem (2). We put the detailed
analysis in Appendix D due to space limitation.

VII. CASE STUDIES

A. System Configuration

To verify the performance of the proposed method, a 44-
bus system shown in Fig5 is used for the test, which is a
modified benchmark of low-voltage MG systems [23, 35].
The system includes three feeders with six dispatchable MGs,
where MG2 and MG5 are DC MGs while the others are AC
MGs. The Breaker 1 is open, which implies that the system
operates in an isolated mode. All simulations are implemented
in the professional power system simulation software PSCAD.

The simulation scenario is: 1) at $t = 2s$, there is a
60kW load increase in the system; 2) at $t = 8s$, there is a
30kW load drop. Then, each MG increases its generation to
balance the power and restore system frequency. Their initial
generations are (58.93, 46.94, 66.43, 59.95, 52.06, 55.09) kW.
The communication graph is undirected, which is shown in
Fig6. Other parameters are given in Table I.

![Fig. 3. Control diagram of the proposed method](image)

![Fig. 4. Simplified model of a DC MG](image)

![Fig. 5. A schematic diagram of a typical 43-bus MG system](image)

![Fig. 6. Communication graph of the system](image)

**TABLE I**

| System parameters |
|-------------------|
| $a_i$ | $b_i$ | $P_i^g$ (kW) | $P_i^d$ (kW) |
|-------|-------|---------------|---------------|
| 0.8   | 0.01  | 85            | 0             |
| 1     | 0.01  | 80            | 0             |
| 0.65  | 0.014 | 90            | 0             |
| 0.75  | 0.012 | 85            | 0             |
| 0.9   | 0.01  | 80            | 0             |
| 0.85  | 0.01  | 80            | 0             |
B. Non-identical sampling rates

Individual MGs may have different sampling rates (or control period) in practice, which could cause asynchrony and compromise the control performance. In this part, we consider the impact of non-identical sampling rates. The sampling rates of MG1-MG6 are set as 10,000Hz, 12,000Hz, 14,000Hz, 16,000Hz, 18,000Hz, 20,000Hz, respectively. The dynamics of the frequencies and voltages of MGs are shown in Fig 8.

As the load change is located in MG2, the frequency nadir of MG2 is the lowest (about 0.26Hz). The system frequency recovers in 4 seconds after the load change. When the load decreases, the frequency experiences an overshoot of 0.1Hz, and recovers in 2 seconds. Voltages on the DC buses of MG2 and MG5 have a small drop when load increases. On the contrary, they voltages slightly increase after the load drop. The result demonstrates that the system is fairly stable to load variation even with non-identical sampling rates.

Dynamics of generations and $-\mu_{i}$ are given in Fig 8. At the end of stage one (from 2s to 8s), generations of MGs are (79.32, 63.60, 90, 81.82, 70.47, 75.08)kW respectively. At the end of stage two (from 8s to 14s), their values are (69.50, 55.46, 79.20, 70.97, 61.86, 65.04)kW respectively. Generations are identical with that obtained by solving the centralized optimization problem (implemented by CVX). This result verifies the optimality of the proposed method. $-\mu_{i}$ stands for the marginal cost of MG $i$, whose dynamic is given on the right part of Fig 8. The marginal cost of different MGs converges to the same value when the system is stabilized, which indicates that the system operates in an optimal state.

C. Random time delays

In practice, time delay always exists in the communication, which is usually varying up to channel situations. In this part, we examine the impact of time-varying time delays. Initially, all the time delays in communication are set as 20ms. And then we intentionally increase the time delays on the channels of MG1-MG2 and MG5-MG6. Additionally, we have the time delays on these two channels varying in ranges [100ms, 200ms], [200ms, 500ms], [500ms, 800ms] and [800ms, 1000ms], respectively, while the delays on other channels remain 20ms. Frequency and generation dynamics of MG1 under different scenarios are shown in Fig 9. It is observed that, when time delays increase, the convergence becomes slower with larger overshoots in frequency. However, the steady-state generations are still exactly identical to the optimal solution, which verifies the effectiveness of our controller under varying time delays.

D. Comparison with synchronous algorithm

In this part, we compare the performances of the asynchronous and synchronous algorithms under imperfect communication. In the asynchronous case, the sampling rates of MGs are set to the same as that in Section VII B and the time delay varies between [500, 800]ms. The dynamics of MG1 with two algorithms are shown in Fig 10. With the synchronous algorithm SDPD, the system remains stable after load perturbations. However, the frequency nadir and overshoot deteriorate, and the convergence becomes slower. The generation takes more time to reach the optimal solution, with an obvious fluctuation. This result confirms the advantage of the asynchronous algorithm ASDPD.

E. Plug-n-play test

In this part, we examine the performance of ASDPD under the plug-n-play operation mode. The simulation scenario is that DG4 is switched off at $t=2s$ and switched on at $t=8s$. Dynamics of frequencies and generations are illustrated in Fig 11. When DG4 is switched off, there is a small frequency oscillation. Since MG3 is close to DG4, its frequency nadir is the lowest. When DG4 is connected, the frequency oscillation is more fierce. However, the system is stabilized rapidly in 2s. Generation of DG4 drops to zero when it is switched off. Then other DGs increase their generations to re-balance the power. After DG4 is re-connected, all the generations recover to the initial values. This demonstrate that our controller can adapt to the plug-n-play mode.
DG2 is switched off and on. in power systems and energy control in multi-energy systems. The proposed methodology can also paper gives a framework to design distributed controller under an upper bound of the time delay. Furthermore, we have presented the implementation of the asynchronous distributed power control in hybrid AC and DC MGs. Numerical experiments on PSCAD confirm the superior performance of the proposed methodology.

Communication asynchrony widely exists in MGs. This paper gives a framework to design distributed controller under imperfect communication. The proposed methodology can also be extended to other related problems, such as voltage control in power systems and energy control in multi-energy systems.

REFERENCES

[1] F. Nejatabkhah and Y. W. Li, “Overview of power management strategies of hybrid ac/dc microgrid,” IEEE Trans. Power Electron., vol. 30, no. 12, pp. 7072–7089, 2015.

[2] Q. Xu, J. Xiao, P. Wang, and C. Wen, “A decentralized control strategy for economic operation of autonomous ac, dc, and hybrid ac/dc microgrids,” IEEE Trans. Energy Convers., vol. 32, no. 4, pp. 1345–1355, 2017.

[3] X. Liu, P. Wang, P. C. Loh et al., “A hybrid ac/dc microgrid and its coordination control.” IEEE Trans. Smart Grid, vol. 2, no. 2, pp. 278–286, 2011.

[4] Y. Xia, W. Wei, M. Yu, X. Wang, and Y. Peng, “Power management for a hybrid ac/dc microgrid with multiple subgrids,” IEEE Trans. Power Electron., vol. 33, no. 4, pp. 3520–3533, 2018.

[5] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. De Vicuña, and M. Castilla, “Hierarchical control of droop-controlled ac and dc microgrids? a general approach toward standardization,” IEEE Trans. Ind. Electron., vol. 58, no. 1, pp. 158–172, 2011.

[6] A. Bidram and A. Davoudi, “Hierarchical structure of microgrids control system,” IEEE Trans. Smart Grid, vol. 3, no. 4, pp. 1963–1976, 2012.

[7] Z. Wang, F. Liu, S. H. Low, P. Yang, and S. Mei, “Distributed load-side control: Coping with variation of renewable generations,” arXiv preprint arXiv:1804.04941, 2018.

[8] F. Dörfler, J. W. Simpson-Porco, and F. Bullo, “Breaking the hierarchy: Distributed control and economic optimality in microgrids,” IEEE Trans. Control Netw. Syst., vol. 3, no. 3, pp. 241–253, 2016.

[9] M. Yazdanian and A. Mehrizi-Sani, “Distributed control techniques in microgrids,” IEEE Trans. Smart Grid, vol. 5, no. 6, pp. 2901–2909, Nov 2014.

[10] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, “A survey of distributed optimization and control algorithms for electric power systems,” IEEE Trans. Smart Grid, vol. 8, no. 6, pp. 2941–2962, 2017.

[11] H. Xin, Y. Liu, Z. Qu, and D. Gan, “Distributed control and generation estimation method for integrating high-density photovoltaic systems,” IEEE Trans. Energy Convers., vol. 29, no. 4, pp. 988–996, Dec 2014.

[12] F. Guo, C. Wen, J. Mao, and Y. D. Song, “Distributed secondary voltage and frequency restoration control of droop-controlled inverter-based microgrids,” IEEE Trans. Ind. Electron., vol. 62, no. 7, pp. 4355–4364, July 2015.

[13] J. W. Simpson-Porco, Q. Shafiee, F. Drffer, J. C. Vasquez, J. M. Guerrero, and F. Bullo, “Secondary frequency and voltage control of islanded microgrids via distributed averaging,” IEEE Trans. Ind. Electron., vol. 62, no. 11, pp. 7025–7038, Nov 2015.

[14] N. M. Dekhordi, H. R. Baghaee, N. Sadati, and J. M. Guerrero, “Distributed noise-resilient secondary voltage and frequency control for islanded microgrids,” IEEE Trans. Smart Grid, pp. 1–1, 2018.

[15] T. Zhao and Z. Ding, “Distributed agent consensus-based optimal resource management for microgrids,” IEEE Trans. Sustainable Energy, vol. 9, no. 1, pp. 443–452, 2018.

[16] G. Chen, F. L. Lewis, E. N. Feng, and Y. Song, “Distributed optimal active power control of multiple generation systems,” IEEE Trans. Ind. Electron., vol. 62, no. 11, pp. 7079–7090, Nov 2015.

[17] J. Zhou, H. Zhang, Q. Sun, D. Ma, and B. Huang, “Event-based distributed active power sharing control for interconnected ac and dc microgrids,” IEEE Trans. Smart Grid, 2017.

[18] C. Zhao, U. Topcu, N. Li, and S. H. Low, “Design and stability of load-side primary frequency control in power systems,” IEEE Trans. Autom. Control, vol. 59, no. 5, pp. 1177–1189, Jan. 2014.

[19] X. Zhang and A. Papachristodoulou, “A real-time control framework for smart power networks: Design methodology and stability,” Automatica, vol. 58, pp. 43–50, 2015.

[20] Z. Wang, W. Wu, and B. Zhang, “A fully distributed power dispatch method for fast frequency recovery and minimal generation cost in autonomous microgrids,” IEEE Trans. Smart Grid, vol. 7, no. 1, pp. 19–31, 2016.

[21] Z. Wang, F. Liu, Y. Chen, S. H. Low, and S. Mei, “Unified distributed control of stand-alone dc microgrids,” IEEE Trans. Smart Grid, 2017.

[22] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” Proc. IEEE, vol. 95, no. 1, pp. 215–233, Jan 2007.

[23] X. Wu and C. Shen, “Distributed optimal control for stability enhancement of microgrids with multiple distributed generators,” IEEE Trans. Power Syst., vol. 32, no. 5, pp. 4045–4059, 2017.

[24] D. Alkano, J. M. Scherpen, and M. Cao, “Distributed asynchronous supply coordination for energy producers embedded in the energy grids,” in Decision and Control (CDC), 2015 IEEE 54th Annual Conference on. IEEE, 2015, pp. 5239–5244.

[25] P. Yi and L. Pavel, “A distributed primal-dual algorithm for computation of generalized nash equilibria with shared affine coupling constraints via operator splitting methods,” arXiv preprint arXiv:1703.05388, 2017.

[26] H. Bauschke and P. L. Combettes, “Convex analysis and monotone operator theory in hilbert spaces,” 2017.

[27] S. Boyd and L. Vandenberghe, Convex optimization. Cambridge university press, 2004.

[28] Z. Wang, F. Liu, J. Z. F. Pang, S. Low, and S. Mei, “Distributed optimal frequency control considering a nonlinear network-preserving model,” IEEE Trans. Power Syst., in press, 2018.

[29] M. Cao, A. S. Morse, B. D. Anderson et al., “Agreeing asynchronously,” IEEE Transactions on Automatic Control, vol. 53, no. 8, p. 1826, 2008.

[30] P. Yi and L. Pavel, “Asynchronous distributed algorithm for seeking generalized nash equilibria,” 2018 European Control Conference, 2018.

[31] Z. Peng, Y. Xu, M. Yan, and W. Yin, “Arock: an algorithmic framework for asynchronous parallel coordinate updates,” SIAM J. Sci. Comput., vol. 38, no. 5, pp. A2851–A2879, 2016.

[32] C. De Persis and N. Monshizadeh, “Infinite-dimensional generalized nash equilibria with shared affine coupling constraints via operator splitting methods,” Proc. IEEE 54th Conference (ECC), 2018 European Control Conference, 2018.

[33] Z. Wang, F. Liu, S. H. Low, C. Zhao, and S. Mei, “Distributed frequency control with operational constraints, part ii: Network power balance,” IEEE Trans. Smart Grid, in press, 2017.

[34] Z. Wang, W. Wu, and B. Zhang, “A distributed control method with minimum generation cost for dc microgrids,” IEEE Trans. Energy Convers., vol. 31, no. 4, pp. 1462–1470, 2016.

[35] S. Papathanassiou, N. Hatziargyriou, K. Strunz et al., “A benchmark low voltage microgrid network,” in Proceedings of the CIGRE symposium: power systems with dispersed generation, 2005, pp. 1–8.

[36] P. L. Combettes and I. Yamada, “Compositions and convex combinations of averaged nonexpansive operators,” J. Math. Anal. Appl., vol. 425, no. 1, pp. 55 – 70, 2015.
APPENDIX A

PROOFS OF THEOREM 2

Proof. By (6a) – (6d) and Definition 1 we have
\[ 0 = \sum_{j \in \mathcal{N}_i} (\mu_i^* - \mu_j^*) + P_j^{\theta_i} - \hat{P}_i \]
(A.1a)
\[ P_j^{\theta_i} = \mathcal{P}_{\Omega_i}, \quad P_j^{\theta_i} - \sigma \left( f'_i(P_j^{\theta_i}) + \mu_i^* \right) \]
(A.1b)
As \( P_j^{\theta_i} = \hat{P}_i \), we have
\[ 0 = \sum_{i \in \mathcal{N}} P_j^{\theta_i} - \sum_{i \in \mathcal{N}} \hat{P}_i \]
(A.2a)
where \( \mu^* \) is a constant.

In addition, (A.1b) is equivalent to
\[ f'_i(P_j^{\theta_i}) + \mu_i^* \begin{cases} \geq 0, & P_j^{\theta_i} = P_j^g \\ = 0, & P_j^{\theta_i} < P_j^g < P_j^g \\ \leq 0, & P_j^{\theta_i} = P_j^g \end{cases} \] (A.2c)
The KKT condition of problem (2) is
\[ 0 = \frac{\partial \mathcal{L}}{\partial P_j^{\theta_i}} = f'_i(P_j^{\theta_i}) + \mu_i^* - \gamma_i^* + \gamma_i^+ \]
(A.3a)
\[ 0 = \frac{\partial \mathcal{L}}{\partial \mu_i} = \sum_{i \in \mathcal{N}} P_j^{\theta_i} - \sum_{i \in \mathcal{N}} \hat{P}_i \]
(A.3b)
\[ 0 = \gamma_i^- \frac{\partial \mathcal{L}}{\partial \gamma_i} = \gamma_i^-(P_j^{\theta_i} - P_j^g) \]
(A.3c)
\[ 0 = \gamma_i^+ \frac{\partial \mathcal{L}}{\partial \gamma_i^+} = \gamma_i^+(P_j^{\theta_i} - P_j^g) \]
(A.3d)
\[ 0 \leq \gamma_i^-, \quad 0 \leq \gamma_i^+ \] (A.3e)

It is straightforward to see that (A.2) is equivalent to (A.3).

In addition, (2) is a convex optimization problem and Slater’s condition holds, which completes the proof.

APPENDIX B

PROOFS OF LEMMA 3

Proof. For the assertion 1), we know \((\text{Id} + \Phi^{-1}U) - 1 \in \mathcal{A} \left( \frac{1}{2} \right)\)
and \(\text{Id} - \Phi^{-1}B \in \mathcal{A} \left( \frac{1}{2 \gamma_i} \right)\) from [25, Lemma 5.6]. Then, following from [36, Proposition 2.4], we know \(T \in \mathcal{A} \left( \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} \right)\).

From assertion 1) and definition of averaged operators, there exists a nonexpansive operator \(R\) such that
\[ T = \left( 1 - \frac{2 \kappa \zeta}{4 \kappa \zeta - 1} \right) \text{Id} + \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} R \] (B.1)

Then, we have the assertion 2).

Since \(T\) is \(\frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1}\)-averaged, \(T\) is also a nonexpansive operator [26, Remark 4.24]. For any nonexpansive operator \(T\), \(\text{Fix}(T) \neq \emptyset\) [26, Theorem 4.19]. Suppose \(x\) is a fixed point of \(T\), and we have \(T(x) = x = \left( 1 - \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} \right) \text{Id}(x) + \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} R(x)\). Thus, \(\text{Id}(x) = \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} R(x)\), which is equivalent to \(x = R(x)\).

Similarly, suppose \(x\) is a fixed point of \(R\), and we have \(T(x) = \left( 1 - \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} \right) \text{Id}(x) + \frac{2 \gamma_i \zeta}{4 \kappa \zeta - 1} R(x) = x\). Thus, assertion 3) holds, which completes the proof.

APPENDIX C

PROOFS OF THEOREM 4

Proof. Combining (B.1) and (18), we have
\[ w_{k+1} = w_k + \eta_k \varphi_k \circ \left( 1 - \frac{2 \kappa \zeta}{4 \kappa \zeta - 1} \right) \hat{w}_k \]
where \(\varphi_k\) is a nonexpansive operator [26, Remark 4.24]. For any nonexpansive operator \(\mu\), where \(\mu^*\) is a constant.

In addition, (2) is a convex optimization problem and Slater’s condition holds, which completes the proof.

Proof. By (6a) – (6d) and Definition 1 we have
\[ 0 = \sum_{j \in \mathcal{N}_i} (\mu_i^* - \mu_j^*) + P_j^{\theta_i} - \hat{P}_i \]
(A.1a)
\[ P_j^{\theta_i} = \mathcal{P}_{\Omega_i}, \quad P_j^{\theta_i} - \sigma \left( f'_i(P_j^{\theta_i}) + \mu_i^* \right) \]
(A.1b)
As \( P_j^{\theta_i} = \hat{P}_i \), we have
\[ 0 = \sum_{i \in \mathcal{N}} P_j^{\theta_i} - \sum_{i \in \mathcal{N}} \hat{P}_i \]
(A.2a)
where \( \mu^* \) is a constant.

In addition, (A.1b) is equivalent to
\[ f'_i(P_j^{\theta_i}) + \mu_i^* \begin{cases} \geq 0, & P_j^{\theta_i} = P_j^g \\ = 0, & P_j^{\theta_i} < P_j^g < P_j^g \\ \leq 0, & P_j^{\theta_i} = P_j^g \end{cases} \] (A.2c)
The KKT condition of problem (2) is
\[ 0 = \frac{\partial \mathcal{L}}{\partial P_j^{\theta_i}} = f'_i(P_j^{\theta_i}) + \mu_i^* - \gamma_i^* + \gamma_i^+ \]
(A.3a)
\[ 0 = \frac{\partial \mathcal{L}}{\partial \mu_i} = \sum_{i \in \mathcal{N}} P_j^{\theta_i} - \sum_{i \in \mathcal{N}} \hat{P}_i \]
(A.3b)
\[ 0 = \gamma_i^- \frac{\partial \mathcal{L}}{\partial \gamma_i} = \gamma_i^-(P_j^{\theta_i} - P_j^g) \]
(A.3c)
\[ 0 = \gamma_i^+ \frac{\partial \mathcal{L}}{\partial \gamma_i^+} = \gamma_i^+(P_j^{\theta_i} - P_j^g) \]
(A.3d)
\[ 0 \leq \gamma_i^- , \quad 0 \leq \gamma_i^+ \] (A.3e)

It is straightforward to see that (A.2) is equivalent to (A.3).

In addition, (2) is a convex optimization problem and Slater’s condition holds, which completes the proof.

APPENDIX D

OPTIMALITY ANALYSIS OF THE IMPLEMENTATION

Proof. In steady state, we have \(\omega_i = \omega_j = \omega^*\), \(\forall i,j \in \mathcal{N}\), and \(\frac{\partial \mathcal{L}_{ac}}{\partial \omega} = 0, \forall i \in \mathcal{N}_{ac}\). By (6a) – (6d) and Definition 1, we have
\[ 0 = \sum_{j \in \mathcal{N}_i} (\mu_i^* - \mu_j^*) + D_i \omega^*, \quad i \in \mathcal{N}_{ac} \] (D.1a)
\[ 0 = \sum_{j \in \mathcal{N}_i} (\mu_i^* - \mu_j^*), \quad i \in \mathcal{N}_{ac} \] (D.1b)

From (D.1a) and (D.1b), we have
\[ r_1 \mu^* + D_i \omega^* = 0 \] (D.2a)
\[ r_{1|\mathcal{N}_{ac}} \mu^* + D_i \omega^* = 0 \] (D.2b)
\[ r_{1|\mathcal{N}_{ac}+1} \mu^* = 0 \] (D.2c)
\[ \vdots \]
\[ r_{1|\mathcal{N}} \mu^* = 0 \] (D.2d)

where \(r_i\) is the \(i\)th row of Laplacian matrix \(L\), and \(r_1 + r_2 + \cdots + r_{|\mathcal{N}|} = 0\). Thus, we have
\[ \omega^* \sum_{i \in \mathcal{N}_{ac}} D_i = 0 \] (D.2e)

This implies that \(\omega^* = 0\), implying \(P_j^{\theta_i} = \hat{P}_i\). Then, we have \(\mu_i^* = \mu_j^* = \mu^*\) with a constant \(\mu^*\). Other analysis is similar to that of Appendix A, which is omitted here.