The essence of power system operation is to maintain system security at minimal cost. The most important security requirements include maintaining instantaneous balance between generation and load power and keeping frequency close to its nominal value, i.e., 60 Hz in US. These security requirements must be satisfied at minimal operational cost, whether it is the cost for generator fuel consumption and emissions, or the cost to compensate for user discomfort when their electric loads are controlled by the system operator.

Traditional power system operations to fulfill these requirements primarily rely on controlling large generators. The underlying philosophy is to schedule the generators to match forecasted demand. However, this philosophy would not work in the future as the deepening penetration of intermittent renewable generation, such as wind and solar, introduces increasingly large and fast variations in power supply. In this case, the large generators may not provide adequate control capacity and respond sufficiently fast to balance the variations, which motivates us to control alternative resources, especially electric loads.

The idea of load management dates back to the late 1970s. Schweppe et al. advocated in a 1980 paper its deployment to “assist or even replace turbine-governed systems and spinning reserve.” They also proposed to use spot prices to incentivize the users to adapt their consumption to the true cost of generation at the time of consumption. Remarkably it was emphasized back then that such adaptive loads would “allow the system to accept more readily a stochastically fluctuating energy source, such as wind or solar generation” [1]. This point is echoed over the last decade, e.g., in [2–7], that argue for “grid-friendly” appliances, such as refrigerators, water or space heaters, ventilation systems, and air conditioners, as well as plug-in electric vehicles to help manage energy imbalance in power systems.

To exploit the full potential of load management, a set of important issues need to be addressed, including: (i) scalability and flexibility of the control system to support autonomous and plug-and-play operations of controllable loads; (ii) coordination between controllable loads, as well as coordination between loads and generators, to ensure a predictable and stable system behavior; (iii) optimization of comfort levels of controllable load users measured by utility functions. Addressing these issues calls for the transformation of power systems from a centralized, hierarchical control architecture to a distributed architecture.
Distributed Load Management

To facilitate the historic transformation discussed above, this chapter introduces our work on distributed load management. Specifically, Section 1.1 introduces a set of distributed protocols for electric vehicle (EV) charging, which provide analytical convergence guarantees of individual EV charging profiles without requiring the customers to share their charging constraints with the distribution utility company (or system operator).

These distributed protocols require messages to be exchanged among the distribution utility company and the customers regarding possible price profiles and desired charging profiles in response. This requires secure and low-latency two-way communication between the distribution utility company and the EV customers, which may be costly to achieve. In Section 1.2, we propose an online learning version of the algorithm that requires only one-way communication from the distribution utility company to the customers about the pricing profiles of previous days.

The EV charging schemes above are based on a simplified system model where the power network connecting the distribution utility company and EVs is not explicitly modeled. In Section 1.3, we extend these schemes in two aspects. First, we consider general controllable loads that not only include EVs, but also refrigerators, air conditioners, etc.; second, we consider the dynamic model of power network that connects the controllable loads, and focus on design of distributed feedback control and stability of dynamic power network under such control.

1.1 Distributed charging protocols for electric vehicles

Consider a scenario, also studied in [8–11], where an electric utility company negotiates with \(N\) electric vehicles (EVs) over \(T\) time slots of length \(\Delta T\) on their charging profiles. The utility is assumed to know (precisely predict) the inelastic base demand profile (aggregate non-EV demand) and aims to shape the aggregate charging profile of EVs to flatten the total demand (base demand plus EV demand) profile. Each EV can charge after it plugs in and needs to be charged a specified amount of electricity by its deadline. In each time slot, the charging rate of an EV is a constant. Let \(D(t)\) denote the base load in slot \(t\), \(r_n(t)\) denote the charging rate of EV \(n\) in slot \(t\), \(r_n := (r_n(1), \ldots, r_n(T))\) denote the charging profile of EV \(n\), for \(n \in \mathcal{N} := \{1, \ldots, N\}\) and \(t \in \mathcal{T} := \{1, \ldots, T\}\). Roughly speaking, this optimal control problem formalizes the intent of flattening the total demand profile, which is captured by the objective function

\[
L(r) = L(r_1, \ldots, r_N) := \sum_{t=1}^{T} U(D(t) + \sum_{n=1}^{N} r_n(t)).
\]  

In (1.1) and hereafter, \(r := (r_1, \ldots, r_N)\) denotes a charging profile of all EVs. The map \(U : \mathbb{R} \to \mathbb{R}\) is strictly convex.

The charging profile \(r_n\) of EV \(n\) is considered to take values in the interval \([0, \tau_n]\) for some given \(\tau_n \geq 0\). In order to impose arrival time and deadline
1.1 Distributed charging protocols for electric vehicles

constraints, \( \tau_n \) is considered to be time-dependent with \( \tau_n(t) = 0 \) for slots \( t \) before the arrival time and after the deadline of EV \( n \). Hence

\[
0 \leq r_n(t) \leq \tau_n(t), \quad n \in \mathcal{N}, \ t \in \mathcal{T}.
\]  

(1.2)

For EV \( n \in \mathcal{N} \), let \( B_n, s_n(0), s_n(T) \) and \( \eta_n \) denote its battery capacity, initial state of charge, final state of charge and charging efficiency. The constraint that EV \( n \) needs to reach \( s_n(T) \) state of charge by its deadline is captured by the total energy stored over time horizon

\[
\eta_n \sum_{t \in \mathcal{T}} r_n(t) \Delta T = B_n(s_n(T) - s_n(0)), \quad n \in \mathcal{N}.
\]  

(1.3)

Define the charging rate sum

\[
R_n := B_n(s_n(T) - s_n(0)) / (\eta_n \Delta T)
\]

for \( n \in \mathcal{N} \). Then, the constraint in (1.3) can be written as

\[
\sum_{t=1}^{T} r_n(t) = R_n, \quad n \in \mathcal{N}.
\]  

(1.4)

**Definition 1.1** Let \( U : \mathbb{R} \to \mathbb{R} \) be strictly convex. A charging profile \( r = (r_1, \ldots, r_N) \) is

1) **feasible**, if it satisfies the constraints (1.2) and (1.4);

2) **optimal**, if it solves the optimal charging (OC) problem

\[
\text{OC} \left\{ \begin{array}{ll}
\min_{r_1, \ldots, r_N} & \sum_{t=1}^{T} U \left( D(t) + \sum_{n=1}^{N} r_n(t) \right) \\
\text{s.t.} & 0 \leq r_n(t) \leq \tau_n(t), \quad t \in \mathcal{T}, n \in \mathcal{N}; \\
& \sum_{t=1}^{T} r_n(t) = R_n, \quad n \in \mathcal{N};
\end{array} \right.
\]  

(1.5)

3) **valley-filling**, if there exists \( A \in \mathbb{R} \) such that

\[
\sum_{n \in \mathcal{N}} r_n(t) = [A - D(t)]^+, \quad t \in \mathcal{T}.
\]

If the objective is to track a given demand profile \( G \) rather than to flatten the total demand, the objective function can be modified as

\[
\sum_{t=1}^{T} U \left( D(t) + \sum_{n=1}^{N} r_n(t) - G(t) \right).
\]  

(1.6)

As a means toward developing a decentralized charging protocol, [8, 9] established a series of properties of optimal charging profiles. We now review some of these properties without their proofs. For notational simplicity, for a given charging profile \( r = (r_1, \ldots, r_N) \), let

\[
R_r := \sum_{n \in \mathcal{N}} r_n
\]

denote its corresponding aggregate charging profile.
Property 1  If a feasible charging profile $r$ is valley-filling, then it is optimal.

Define $\mathcal{F}_n := \{r_n | 0 \leq r_n \leq r_n, \sum_{t \in \mathcal{T}} r_n(t) = R_n\}$ as the set of feasible charging profiles for EV $n$. Then,

$$\mathcal{F} := \mathcal{F}_1 \times \cdots \times \mathcal{F}_N$$

is the set of feasible charging profiles $r = (r_1, \ldots, r_N)$.

Property 2  If $\mathcal{F}$ is non-empty, optimal charging profiles exist.

Valley-filling is our intuitive notion of optimality. However, it may not be always achievable. For example, the “valley” in inelastic base demand may be so deep that even if all EVs charge at their maximum rate, it is still not completely filled, e.g., at 4:00 in Figure 1.1 (bottom). Besides, EVs may have stringent deadlines such that the potential for shifting the load over time to yield valley-filling is limited. The notion of optimality in Definition 1.1 relaxes these restrictions as a result of Property 2. Moreover, it agrees with the intuitive notion of optimality when valley-filling is achievable as a result of Property 1, illustrated in Figure 1.1 (top).

Definition 1.2  Two feasible charging profiles $r$ and $r'$ are equivalent, provided that $R_r = R_{r'}$, i.e., $r$ and $r'$ have the same aggregate charging profile. We denote this relation by $r \sim r'$.

It is easy to check that the relation $\sim$ is an equivalence relation. Define equivalence classes $\{r' \in \mathcal{F} | r' \sim r\}$ with representatives $r \in \mathcal{F}$, and

$$\mathcal{O} := \{r \in \mathcal{F} | r \text{ optimal}\}$$

as the set of optimal charging profiles.

Theorem 1.3  If $\mathcal{F}$ is non-empty, then $\mathcal{O}$ is non-empty, compact, convex, and an equivalence class of the relation $\sim$.

Corollary  Optimal charging profile may not be unique.

Theorem 1.4  The set $\mathcal{O}$ of optimal charging profiles does not depend on the choice of $U$. That is, if $r^*$ is optimal with respect to a strictly convex utility function, then $r^*$ is also optimal with respect to any other strictly convex utility function.

The optimal solution to problem OC provides a uniform means for defining optimality even when valley-filling is not achievable, and Theorem 1.4 implies that this optimality notion is intrinsic, independent of the choice of $U$.

Having established key properties of the optimal solutions for problem OC, we now follow the presentation of an algorithm from [8, 9] in order to solve the problem in a decentralized manner. While [8,9] developed both synchronous and asynchronous algorithms, we focus on the former.

Figure 1.3 shows the information exchange between the utility company and the EVs for the implementation of this algorithm. Given the “price” profile
1.1 Distributed charging protocols for electric vehicles

Figure 1.1 Base demand profile is the average residential load in the service area of Southern California Edison (SCE) from 20:00 on 02/13/2011 to 9:00 on 02/14/2011 [9]. Optimal total demand profile curve corresponds to the outcome of DCA (an algorithm introduced later in this chapter) with \( U(x) = x^2 \). With different specifications for EVs (e.g., maximum charging rate \( r_{\text{max}} \)), optimal charging profile can be valley-filling (top figure) or non-valley-filling (bottom figure). A hypothetical non-optimal curve is shown with dash-dot line.

broadcast by the utility, each EV chooses its charging profile independently, and reports back to the utility. The utility guides their behavior by altering the “price” profile. We assume \( U' \) is Lipschitz with the Lipschitz constant \( \beta > 0 \),

Figure 1.2 An example of equivalent charging profiles. In both top and bottom figures, the two regions correspond to the charging profiles of two different EVs. Since aggregate charging profiles in both figures equal, the charging profile $r_i$ in the top figure is equivalent to the charging profile $r'_i$ in the bottom figure.

\[ |U'(x) - U'(y)| \leq \beta|x - y| \]

for all $x, y$.

Figure 1.3 Schematic view of the information flow between the utility and the EVs. Given the “price” profile, the EVs choose their charging profiles independently. The utility guides their decisions by altering the “price” profile based on total demand profile.

Decentralized charging algorithm (DCA):
Given scheduling horizon $T$, the maximum number $K$ of iterations, error tolerance $\epsilon > 0$, base load profile $D$, the number $N$ of EVs, charging rate sum $R_n$
and charging rate upper bound \( \tau_n \) for EV \( n \in \mathcal{N} \), pick a step size \( \gamma \) satisfying
\[
0 < \gamma < \frac{1}{N \beta}.
\]

1. Initialize the “price” profile and the charging profile as
\[
p^0(t) := U'(D(t)), \quad r^0_n(t) := 0
\]
for \( t \in \mathcal{T} \) and \( n \in \mathcal{N} \), \( k \leftarrow 0 \).
2. The utility broadcasts \( \gamma p^k \) to all EVs.
3. Each EV \( n \in \mathcal{N} \) calculates a new charging profile \( r^{k+1}_n \) as the solution to the following optimization problem
\[
\min_{r_n} \sum_{t \in \mathcal{T}} \gamma p^k(t) r_n(t) + \frac{1}{2} (r_n(t) - r^k_n(t))^2 \tag{1.7}
\]
s.t. \( 0 \leq r_n(t) \leq \tau_n(t), \ t \in \mathcal{T} \);
\[
\sum_{t \in \mathcal{T}} r_n(t) = R_n,
\]
and reports \( r^{k+1}_n \) to the utility.
4. The utility collects charging profiles \( r^{k+1}_n \) from the EVs, and updates the “price” as
\[
p^{k+1}(t) := U' \left( D(t) + \sum_{n=1}^{N} r^{k+1}_n(t) \right) \tag{1.8}
\]
for \( t \in \mathcal{T} \).
   - If \( \|p^{k+1} - p^k\| \leq \epsilon \), return \( p^{k+1}, r^{k+1}_n \) for all \( n \).
5. If \( k < K \), \( k \leftarrow k + 1 \), and go to step (2).
   - Else, return \( p^K, r^K_n \) for all \( n \).

The “price” signal \( p \) is actually a control signal used by the utility company to guide the EVs in choosing their charging profiles, and is not necessarily the real electricity price.

In each iteration, the algorithm can be split into two parts. In the first part, EV \( n \) updates its charging profile to minimize its objective function as (1.7). There are two terms in the objective: the first term is the “cost” and the second term penalizes deviations from the profile computed in the previous iteration. The extra penalty term ensures convergence of DCA, and vanishes as \( k \to \infty \) (see Theorem 1.8). Hence, the objective function of each EV reduces to its “cost” as \( k \to \infty \). Intuitively, the smaller \( \gamma \) is, the more significant the penalty term becomes, the less \( r^{k+1}_n \) is going to deviate from \( r^k_n \), and the more likely DCA will converge. When \( \gamma < \frac{1}{N \beta} \), DCA converges to optimal charging profiles (see Theorem 1.7). The upper bound \( \bar{\gamma} := \frac{1}{N \beta} \) of \( \gamma \) is inversely proportional to the number \( N \) of EVs and the Lipschitz constant \( \beta \) of \( U' \). For large \( N \), each EV has to update its charging profile more slowly since the aggregate profile update is roughly \( N \) times amplified. Hence, the penalty term should be larger, and \( \bar{\gamma} \)
Distributed Load Management

should be smaller. When the Lipschitz constant \( \beta \) is larger, the same difference in total demand is going to cause a larger difference in \( U' \), or \( p \) according to (1.8). Hence, for the two terms in (1.7) to remain the same scale, \( \tau \) should decrease.

In conclusion, the upper bound \( \tau = \frac{1}{N\beta} \) agrees with our intuition. In the second part of the iteration, the utility updates the “price” profile according to (1.8). It sets higher prices for slots with higher total demand, to give EVs the incentive to shift their energy consumption to slots with lower total demand.

Let the superscript \( k \) for each variable denote its respective value in iteration \( k \). For example, \( r^k_n \) denotes the charging profile of EV \( n \) in iteration \( k \). Similarly, \( R_k := \sum_{n=1}^{N} r^k_n \) denotes the aggregate charging profile in iteration \( k \).

**Lemma 1.5** If the set \( \mathcal{F} \) of feasible charging profiles is non-empty, then the inequality
\[
\langle \gamma p^k, r^k_{n+1} - r^k_n \rangle \leq - \| r^k_{n+1} - r^k_n \|^2
\]
holds for \( n \in \mathcal{N} \) and \( k \geq 1 \).

**Lemma 1.6** If \( \mathcal{F} \) is non-empty, then for \( n \in \mathcal{N} \) and \( k \geq 1 \), \( r^{k+1}_n = r^k_n \) if and only if, for all \( r_n \in \mathcal{F}_n \),
\[
\langle p^k, r_n - r^k_n \rangle \geq 0.
\]

Recall that \( \mathcal{O} \) denotes the set of optimal charging profiles.

**Theorem 1.7** If \( \mathcal{F} \) is non-empty, then \( r^k \rightarrow \mathcal{O} \) as \( k \rightarrow \infty \).

**Corollary** A charging profile \( r \) is stationary for DCA, i.e., if \( r^k = r \) for some \( k \geq 0 \) then \( r^k = r \) for all \( k \geq k \), if and only if \( r \in \mathcal{O} \).

**Theorem 1.8** Let \( r^* \) be an optimal charging profile. If \( \mathcal{F} \) is non-empty, then
- the aggregate charging profile converges to that of \( r^* \), i.e.,
  \[ \lim_{k \to \infty} R^k = R_{r^*}; \]
- the price profile converges to that corresponding to \( r^* \), i.e.,
  \[ \lim_{k \to \infty} p^k = U'(D + R_{r^*}); \]
- for each EV \( n \), the difference between two consecutive charging profiles vanishes, i.e.,
  \[ \lim_{k \to \infty} \| r^{k+1}_n - r^k_n \| = 0. \]

Theorem 1.7 shows the convergence of \( r^k \) to the optimal set \( \mathcal{O} \) while Theorem 1.8 focuses on the convergence of \( R^k \) and \( p^k \) to the optimal value \( R_{r^*} \) and \( U'(D + R_{r^*}) \).
1.1 Distributed charging protocols for electric vehicles

We conclude this section with an example from [9] which compared DCA to another algorithm from [12] (referred to as “DAP” for “deviation from average penalty”).

We choose the average residential load profile in the service area of South California Edison from 20:00 on 02/13/2011 to 9:00 on 02/14/2011 as the average base demand profile per household. According to the typical charging characteristics of EVs in [13], we set $r_{\text{max}}^n = 3.3\text{kW}$. We assume that the charging rate $r_n(t)$ of EV $n$ at slot $t$ can take continuous values in $[0, r_{\text{max}}^n]$ after EV $n$ plugs in for charging and before its deadline. We consider the penetration level of $N = 20$ EVs in 100 households. The planning horizon is from 20:00 to 9:00 the next day, and divided into 52 slots of 15 minutes, during which the charging rate of an EV is a constant. The map $U : \mathbb{R} \to \mathbb{R}$ is taken to be $U(x) = \frac{1}{2}x^2$. As in [12], we choose the price function and parameters for Algorithm DAP as $p(x) = 0.15x^2$, $c = 1$, and $\delta = 0.15$.

![Figure 1.4](image)

**Figure 1.4** All EVs plug in at 20:00 with deadline at 9:00 on the next day, and have 10kWh charging capacity. Multiple dash-dot curves correspond to total demand in different iterations of Algorithm DAP.

**Homogeneous case**

Although DCA obtains optimal charging profiles, whatever the parameters $T_n(t)$ and $R_n$ are, we simulate the homogeneous case to compare with Algorithm DAP, since Algorithm DAP. In this example, all the EVs have the same deadline at 9:00 the next morning. Figure 1.4 shows the average total demand profile in each iteration of DCA and DAP in a homogeneous case. Both algorithms converge to a flat charging profile. Moreover, DCA converges with a single iteration, while DAP takes several iterations to converge.
Distributed Load Management

Figure 1.5 All EVs plug in at 20:00 with deadline at 9:00 on the next day, but have different charging capacities uniformly distributed in [0, 20] kWh.

**Non-homogeneous case**
Figure 1.5 shows the average total demand profiles at convergence of DCA and DAP in a non-homogeneous case where EVs have different charging capacities. DCA still converges to a flat charging profile in a few iterations while DAP no longer converges to a flat charging profile.

1.2 **An online-learning based implementation**

The algorithm considered above (and its various extensions) have analytical convergence guarantees and do not require the customers to share their charging constraints with the utility company. On the other hand, they require a series of messages to be exchanged among the utility company and the customers regarding possible price profiles and desired charging profiles in response. As the available power supply and the customer requirements for charging their EVs change from day to day, these messages need to be exchanged daily to calculate the charging profiles. This requires secure and low-latency two-way communication between the utility company and the EV customers, which may be costly to achieve. Instead, we propose an online learning version of the algorithm that requires only one-way communication from the utility company to the customers about the pricing profiles of previous days. The price that we pay for this one-way communication is a slower convergence of the charging profiles to the optimal ones, and a relaxation of the sense in which the charging profiles are optimal.

Some relevant references that apply regret minimization to demand response (DR) are [14–16]. In [14], real-time electricity pricing strategies for DR are de-
1.2 An online-learning based implementation

signed using regret minimization. However, the focus of that work is on optimizing the utility function for the utility company, and the customer behavior is assumed to be such that the load change is linear in the price variation. The objective of [15] is to design pricing policies for the customers having price responsive loads. The exact demand function of the customer is assumed to be unknown to the pricing policy maker. In [15], the utility company is the only decision maker, whereas in the discussion below, the utility company and the EV customers are all decision makers. In [16], regret minimization is used to learn the charging behavior of the EV customers. The price responsiveness for a community of customers is captured through a conditional random field model. The regret minimization algorithm is adopted to learn the parameters of the model.

We consider the same problem formulation as above. The specific difference is in the information flow. The utility company monitors the total load and publishes the price profile for the previous day as realized according to a fixed and known pricing policy. The customers decide on the charging schedules for the next day with access to these pricing profiles for the previous days. No other communication occurs between the utility company and the customers, or among the customers.

Online Learning Framework

We now adopt a regret minimization framework to solve both the optimization problem of the company and the customer. Let $L$ be a $\lambda$-strongly convex function with respect to a given norm $\| \cdot \|$. Let $D_L(\cdot, \cdot)$ denote the Bregman divergence [17] with respect to $L$. Let $\| \cdot \|_*$ denote the norm that is dual to $\| \cdot \|$. Let $\nabla L$ denote the gradient of $L$ and $\nabla L^{-1}$ denote the inverse mapping of $\nabla L$.

For the $i$-th EV customer, the decision variable is her charging profile on the $k$-th day and a charging profile is feasible if it the charging rate at every time satisfies the specified lower and upper bounds, and the customer has the required amount $S_i$ of charge at the end. We define $r_i^*$ as the optimal feasible charging profile that minimizes the price $p_i^k$ paid by the customer. Note that this profile does not change from one day to the next.

In the regret minimization framework, the notion of regret is used to measure the performance of an online algorithm [18]. For customer $i \in \mathcal{N}$, the customer regret after $K$ days $R_i$ is defined as the difference between the cumulative cost function value of the charging profiles $r_i^k$, $k = 1, \ldots, K$ generated by an online algorithm and the one generated by $x_i^*$, i.e.,

$$ R_i(K, r_i^k) := \sum_{k=1}^{K} p_i^k(r_i^k) - \min_{x_i \in \mathcal{X}_i} \sum_{k=1}^{K} p_i^k(x_i). $$  \hfill (1.11)

Notice that the suboptimal charging profile $x_i^*$ can only be calculated in hindsight after $K$ days have elapsed.

We adopt the optimistic mirror descent (OMD) algorithm [17] to generate
Distributed Load Management

the charging profile update which minimizes the regret (1.11). On each day, the regret minimization algorithm generates the charging profile update without knowing the current objective function (and its gradient). Specifically, the OMD algorithm iteratively applies the updates

\[ h_{i}^{k+1} = \nabla L_i^{-1}(\nabla L_i(h_i^k) - \eta_i \nabla p_i^k(r_i^k)) \]
\[ r_i^{k+1} = \arg\min_{r_i \text{ is feasible}} \eta_i r_i^T M_i^{k+1} + D_L(r_i, h_i^{k+1}), \]  

(1.12)

where \( \eta_i \in \mathbb{R} \) is an algorithm parameter, \( h_i^k \) is an intermediate update of the charging profile. For easy of presentation, for the vector \( h_i^k \in \mathbb{R}^T \), \( L_i(h_i^k) \) is set to \( L_i(h_i^k) = \frac{\|h_i^k\|^2}{2} \). The \( i \)-th customer may have a prediction \( M_i^k \) for the gradient of the cost function \( p_i^k \). Intuitively, the iteration (1.12) updates the charging profile toward the negative gradient direction and projects it onto the set of feasible charging profiles.

The regret minimization framework for the company can be similarly defined. We skip the details for conciseness.

Convergence Results

The following result summarizes the convergence of the charging profile updates generated by the OMD algorithm.

**Proposition 1.9** (Convergence of regret): For every \( r_i^* \) that is feasible, the iteration (1.12) converges in the sense that

\[ R_i(K, r_i^k) \leq \frac{1}{\eta_i} P_i + \frac{\eta_i}{2} \sum_{k=1}^{K} \| \nabla p_i^k(r_i^k) - M_i^k \|_2^2, \]

(1.13)

where

\[ R_i(K, r_i^k) := \sum_{k=1}^{K} p_i^k(r_i^k) - \sum_{k=1}^{K} p_i^k(r_i^*), \]
\[ P_i := \max_{r_i \text{ is feasible}} L_i(r_i) - \min_{r_i \in \mathcal{F}_i} L_i(r_i). \]

(1.14)

In particular, if \( \eta_i \) is chosen as \( O(1/\sqrt{K}) \), then the average regret, i.e., \( R_i(K)/K \), converges to zero as \( K \to \infty \). A similar convergence result holds for the OMD algorithms followed by the utility company as well.

Thus, as the number of days increases, the average performance of the charging profiles generated by the OMD algorithm approaches the performance that is obtained by the charging profiles \( r^* \) and \( r_i^* \), \( i \in \mathcal{N} \), respectively.

Design of the Pricing Function

There are no guarantees that the solutions \( r_i^* \), \( i \in \mathcal{N} \) obtained by the customers will sum up to the optimal solution obtained by the company. In fact, unless
the pricing function $p^k_i$ is carefully designed, these solutions will not be the same since the objectives of the utility company and the EV customers are different. In fact, the natural choice of $p^k_i$ as

$$p^k_i(r^k_i) = \left( \sum_{j=1}^{N} r^k_j + D^k \right)^T r^k_i$$

(1.15)

does not lead to the charging profiles $(r^*_1, ..., r^*_N)$ that reduce the regret of the utility company to zero. We now propose a choice of $p^k_i$ to ensure that when each customer minimizes her regret, the aggregated charging profile minimizes the utility company’s regret.

**Proposition 1.10** If $p^k_i$ is chosen as

$$p^k_i(r^k_i) = \left( \frac{1}{2} r^k_i + \sum_{j \neq i}^{N} r^k_j + D^k \right)^T r^k_i, \quad i \in \mathcal{N},$$

(1.16)

the customers adopt the iteration (1.12), and $\eta_u = \frac{1}{2} \eta_i$, then the average regret of the utility company converges to zero as the total number of days goes to infinity.

To update the charging profile on day $k$, the $i$-th customer needs to know $2r^k_{i-1} + \sum_{j \neq i}^{N} r^k_{j-1} + D^{k-1}$ or $\sum_{j \neq i}^{N} r^k_{j-1} + D^{k-1}$ depending on which pricing function is adopted. The utility company can simply publish the total load information for the previous day. The customers do not need to have full knowledge about how their consumption will map to a corresponding expenditure.

**Extensions**

The basic framework presented above can be extended in various directions.

**Regret with Respect to the Optimal Charging Profiles**

The regrets defined above measure the difference between the performance of the charging profiles generated by our algorithm and the performance that is obtained by the charging profiles $r^*$ and $r^*_i$, $i \in \mathcal{N}$ that are the optimal profiles that do not vary from one day to the next. In that sense, these are static regrets. Instead, we can define tracking regret that characterizes the difference between the cumulative cost of the charging profiles generated by our algorithm and the cumulative cost of executing the optimal charging profiles $r^{k*}$ that vary from one day to the next (but can be calculated only in hindsight).

**Theorem 1.11** If $\eta_u = O(1/\sqrt{K})$, the OMD algorithm yields that the tracking regret is of the order $O(\sqrt{K} [1 + \sum_{k=1}^{K} ||r^{k*} - r^{k+1*}||])$.

Note that this regret bound increases as the variation of the optimal sequence of decisions $\sum_{k=1}^{K} ||r^{k*} - r^{k+1*}||$ increases. If the optimal solution remains the same from one day to the next, then the tracking regret is of the order $O(\sqrt{K})$.

On the other hand, if the optimal solution varies significantly from one day to
the next, then the average tracking regret will not necessarily converge to zero. Note also that if the utility company has perfect prediction of the gradient of the cost function, then the utility company can set \( \eta \to \infty \) to ensure that the regret bound is zero.

**Presence of Inelastic Customers**

The discussion so far assumed that all customers were rational in the sense that they wanted to choose their charging profile to minimize their cost. Furthermore, they were elastic in scheduling their charging (within the pre-specified constraints). We now assume that some customers are either irrational or inelastic. Suppose that \( N_l \) out of \( N \) customers are inelastic. Denote the set of inelastic customers by \( \mathcal{N}_l \). For every inelastic customer \( i \in \mathcal{N}_l \), we assume that her charging profile remains the same from day to day and is not updated. We also set all customers’ predictions to zeros, i.e., \( M_k^i = 0, \ i \in \mathcal{N}_l, \ k \in \mathbb{N}_{>0} \). The update of the charging profile for inelastic customer \( i \in \mathcal{N}_l \) can thus be written as

\[
\begin{align*}
    h_{i}^{k+1} &= \nabla L_i^{-1} \left( \nabla L_i(h_i^k) - \eta_i \left( \nabla c_i^k(r^k) + \epsilon_i^k \right) \right), \\
    r_{i}^{k+1} &= \arg\min_{x_i \in \mathcal{F}_i} D_{L_i}(r_i, h_i^{k+1}), \quad (1.17)
\end{align*}
\]

where \( \epsilon_i^k \) is an error term that quantifies the inconsistency between the updates as desired by the utility company for each customer to execute and the inelastic customer’s behavior. Due to the presence of the inelastic customers, the ability of the aggregated solution to be valley-filling and hence to minimize the cost function in problem is decreased. We can quantify the performance loss. In particular, we find that the average regret converges to a constant whose size of this constant depends on the error terms \( \epsilon_i, \ i \in \mathcal{N}_l \) and the charging constraints of the inelastic customers.

**Numerical Examples**

We conclude with the following example from [19]. Assume that there are 20 customers. A time slot representing an interval of 30 minutes is used. There are \( T = 24 \) time slots. The starting time is set to 8:00 pm. For simplicity, we consider that all EV customers charge their EVs from the 9th to the 16th time slots. On the first day, the initial charging profiles are assumed to be uniformly distributed over the time slots. The maximum charging rate is set to \( r_{\text{up}}^i(t) = 2 \text{ kW}, \ i \in \mathcal{N} \) and the desired sum \( S_i = 10 \text{ kW}, \ i \in \mathcal{N} \). The simulation is carried out for total \( K = 200 \) days. We set the parameters \( \eta_i = 0.05/\sqrt{K}, \ i \in \mathcal{N} \). We first examine the convergence of the static regret. The base load profile is given in Figure 1.6. The prediction \( M_k^i, \ i \in \mathcal{N}, \ k \in \mathbb{N}_{>0} \) is set to \( M_k^i(r_i^k) = \frac{1}{k} \sum_{k=1}^{k-1} \nabla p_i^k(r_i^k), \ i \in \mathcal{N} \). Figure 1.7 shows that the average regrets converge to zero and the average regret with the prediction converges faster than the one without the prediction. Figure 1.8 shows the static regrets with and without the prediction. Figure 1.8 shows that the regrets are sublinear functions of the number of days.
1.2 An online-learning based implementation

Similar results for the case of a base load profile which does not remain the same from day to day and when inelastic customers are present are presented in [19].

Thus, we have designed a framework for distributed charging control of EVs using online learning and online convex optimization. The proposed algorithm can be implemented without low-latency two-way communication between the utility company and the EV customers, which fits in with the current communication infrastructure and protocols in the smart grid.
Distributed Load Management

1.3 Distributed feedback control of networked loads

This section extends the distributed EV charging schemes above in two aspects. First, we consider general controllable loads that not only include EVs, but also refrigerators, air conditioners, dryers, etc., which can temporarily adjust their power consumption to provide power balance, frequency regulation, and voltage regulation services to the grid. Second, we consider the dynamic model of power network that connects the controllable loads, and focus on design of distributed feedback control and stability of dynamic power network under such control.

In this section, we will introduce our work on load-side primary frequency control [20]. To the best of our knowledge, this work has the first network model and stability analysis of distributed feedback control of networked loads. References are also provided for other load-side control functions, such as secondary frequency control, that are developed by us.

The power network under consideration is modeled as a directed and connected graph \((\mathcal{N}, \mathcal{E})\) where \(\mathcal{N}\) is the set of buses (nodes) and \(\mathcal{E}\) is the set of lines connecting the buses. Directions of lines provide reference directions but not indicate actual directions of power flow, so the assignment of directions can be arbitrary. We use \((i, j) \in \mathcal{E}\) and \(i \to j\) interchangeably to denote a line directed from bus \(i\) to bus \(j\), and use “\(i : i \to j\)” and “\(k : j \to k\)” to denote the sets of buses that are predecessors and successors of bus \(j\), respectively.

The network has two types of buses: generator buses and load buses. A generator bus has an AC generator that converts mechanic power into electric power. A load bus has only electric loads but no generator. We denote the set of generator buses by \(\mathcal{G}\) and the set of load buses by \(\mathcal{L}\). The dynamic model of power network
1.3 Distributed feedback control of networked loads

is described by the following differential-algebraic equations:

\[
\dot{\omega}_j = -\frac{1}{M_j} \left( D_j \omega_j + d_j - P^m_j + \sum_{k \rightarrow j} P_{jk} - \sum_{i \rightarrow j} P_{ij} \right), \quad \forall j \in \mathcal{G} \tag{1.18}
\]

\[
0 = D_j \omega_j + d_j - P^m_j + \sum_{k \rightarrow j} P_{jk} - \sum_{i \rightarrow j} P_{ij}, \quad \forall j \in \mathcal{L} \tag{1.19}
\]

\[
\dot{P}_{ij} = B_{ij} (\omega_i - \omega_j), \quad \forall (i,j) \in \mathcal{E} \tag{1.20}
\]

where, for every bus \( j \in \mathcal{N} \), \( \omega_j \) is the local frequency deviation from its nominal value (60 Hz in US), \( d_j \) is the deviation of aggregate controllable load power from its nominal value or user-preferred value, \( P^m_j \) models uncontrolled changes in power injection, such as load startup, generator loss, or changes in nondispatchable renewable generation, \( P_{ij} \) is the deviation of active power flow on line \((i,j)\) from its nominal value. Positive constant \( M_j \) measures inertia of generator bus \( j \), and the term \( D_j \omega_j \), where \( D_j \) is a positive constant, models the damping effect of frequency sensitive loads, such as induction motors, at bus \( j \). Constant \( B_{ij} \) is given, and in practice it can be calculated from the nominal voltage magnitudes at buses \( i \) and \( j \), the phase angle difference between buses \( i \) and \( j \), and the inductance of line \((i,j)\).

An equilibrium point of the dynamical system (1.18)–(1.20) is a state \((\omega, P)\) where \(\dot{\omega}_j = 0\) for \( j \in \mathcal{G} \) and \(\dot{P}_{ij} = 0\) for \((i,j) \in \mathcal{E}\), i.e., where frequency at all the buses and power flow on all the lines are constant over time.

Given uncontrolled changes \( P^m = (P^m_j, j \in \mathcal{N}) \) of power injections. How should we adjust the controllable loads \( d_j \) to balance power generation and load in a way that minimizes the aggregate disutility for changing these loads? In general we can design state feedback controllers of the form \( d_j(t) := d_j(\omega(t), P(t)) \), prove the feedback system is globally asymptotically stable, and evaluate the aggregate disutility at the equilibrium point. Here we take an alternative approach by directly formulating our goal as an optimal load control (OLC) problem and derive feedback control as a distributed algorithm to solve OLC. Let \( \dot{d}_j := D_j \omega_j \) be the aggregate change of frequency-sensitive uncontrollable load at bus \( j \). Then OLC minimizes the total cost over \( d \) and \( \dot{d} \) while balancing generation and load across the network:

\[
\text{OLC:} \quad \min_{\underline{d} \leq d \leq \overline{d}} \sum_{j \in \mathcal{N}} \left( c_j(d_j) + \frac{1}{2D_j} \dot{d}_j^2 \right) \tag{1.21}
\]

subject to

\[
\sum_{j \in \mathcal{N}} (d_j + \dot{d}_j) = \sum_{j \in \mathcal{N}} P^m_j \tag{1.22}
\]

where \( \underline{d}_j \) and \( \overline{d}_j \) are given constants that bound the controllable load change \( d_j \). We assume the following condition:

**Condition 1.12** OLC is feasible. The cost functions \( c_j \) are strictly convex and twice continuously differentiable on \([\underline{d}_j, \overline{d}_j]\).
The choice of cost functions is based on physical characteristics of electric loads and how user comfort levels change with load power. Examples of cost functions can be found for EVs in [8,9,21] and air conditioners in [22]; see, e.g., [23,24] for other cost functions that satisfy Condition 1.12.

The objective function of the dual problem of OLC is

$$\sum_{j \in N} \Phi_j(\nu) := \sum_{j \in N} \min_{d_j \leq d_j \leq \hat{d}_j} \left( c_j(d_j) - \nu d_j + \frac{1}{2D_j} \nu^2 - \nu \hat{d}_j + \nu P_{jn} \right)$$

where the minimization can be solved explicitly as

$$\Phi_j(\nu) := c_j(d_j(\nu)) - \nu d_j(\nu) - \frac{1}{2} D_j \nu^2 + \nu P_{jn}$$

(1.23)

with

$$d_j(\nu) := \min \left\{ \max \left\{ c_{j-1}(\nu), \bar{d}_j \right\}, \underline{d}_j \right\}.$$  

(1.24)

This objective function has a scalar variable \(\nu\) and is not separable across buses \(j \in N\). Its direct solution hence requires coordination across buses. We propose the following distributed version of the dual problem over the vector \(\nu := (\nu_j, j \in N)\), where each bus \(j\) optimizes over its own variable \(\nu_j\) which are constrained to be equal at optimality:

**DOLC:** \[
\begin{align*}
\max_{\nu} \quad & \Phi(\nu) := \sum_{j \in N} \Phi_j(\nu_j) \\
\text{subject to} \quad & \nu_i = \nu_j, \quad \forall (i,j) \in E.
\end{align*}
\]

We have the following results that are proved in [20]. Instead of solving OLC directly, these results suggest solving DOLC and recovering the unique optimal point \((\hat{d}^*, \hat{d}^*_j)\) of OLC from the unique dual optimal \(\nu^*\).

**lemma 1.13** The objective function \(\Phi\) of DOLC is strictly concave over \(R^{|N|}\).

**lemma 1.14** 1. DOLC has a unique optimal point \(\nu^*\) with \(\nu^*_i = \nu^*_j = \nu^*\) for all \(i,j \in N\).

2. OLC has a unique optimal point \((\hat{d}^*, \hat{d}^*_j)\) where \(\hat{d}^*_j = d_j(\nu^*)\) and \(\hat{d}^*_j = D_j \nu^*\) for all \(j \in N\).

To derive a distributed solution for DOLC consider its Lagrangian

$$L(\nu, \pi) := \sum_{j \in N} \Phi_j(\nu_j) - \sum_{(i,j) \in E} \pi_{ij}(\nu_i - \nu_j)$$

(1.25)

where \(\nu \in R^{|N|}\) is the (vector) variable for DOLC and \(\pi \in R^{|E|}\) is the associated dual variable for the dual of DOLC. Hence \(\pi_{ij}\), for all \((i,j) \in E\), measure the cost of not synchronizing the variables \(\nu_i\) and \(\nu_j\) across buses \(i\) and \(j\). Using
(1.23)–(1.25), a partial primal-dual algorithm for DOLC takes the form
\[ \dot{\nu}_j = \gamma_j \frac{\partial L}{\partial \nu_j}(\nu, \pi) = -\gamma_j (d_j(\nu_j) + D_j \nu_j - P_j^m + \pi_j^\text{out} - \pi_j^\text{in}), \forall j \in \mathcal{G}, \]
\[ 0 = \frac{\partial L}{\partial \nu_j}(\nu, \pi) = - (d_j(\nu_j) + D_j \nu_j - P_j^m + \pi_j^\text{out} - \pi_j^\text{in}), \forall j \in \mathcal{L} \]
\[ \dot{\pi}_{ij} = -\xi_{ij} \frac{\partial L}{\partial \pi_{ij}}(\nu, \pi) = \xi_{ij}(\nu_i - \nu_j), \quad \forall (i, j) \in \mathcal{E} \]

where \( \gamma_j, \xi_{ij} \) are positive stepsizes and \( \pi_j^\text{out} := \sum_{k, j \rightarrow k} \pi_{jk}, \pi_j^\text{in} := \sum_{i, j \rightarrow i} \pi_{ij}. \)

We interpret (1.26)–(1.28) as an algorithm iterating on the primal variables \( \nu \) and dual variables \( \pi \) over time \( t \geq 0 \). Set the stepsizes to be:
\[ \gamma_j = M_j^{-1}, \quad \xi_{ij} = B_{ij}. \]

Then (1.26)–(1.28) become identical to (1.18)–(1.20) if we identify \( \nu \) with \( \omega \) and \( \pi \) with \( P \), and use \( d_j(\omega_j) \) defined by (1.24) for \( d_j \) in (1.18), (1.19). This means that the frequency deviations \( \omega \) and the branch flows \( P \) are respectively the primal and dual variables of DOLC, and the network dynamics together with frequency-based load control execute a primal-dual algorithm for DOLC.

For convenience, we collect system dynamics and load control equations:
\[ \dot{\omega}_j = -\frac{1}{M_j} (d_j + \dot{d}_j - P_j^m + P_j^\text{out} - P_j^\text{in}), \quad \forall j \in \mathcal{G} \]
\[ 0 = d_j + \dot{d}_j - P_j^m + P_j^\text{out} - P_j^\text{in}, \quad \forall j \in \mathcal{L} \]
\[ \dot{\omega}_{ij} = B_{ij} \omega_i - \omega_j, \quad \forall (i, j) \in \mathcal{E} \]
\[ d_j = D_j \omega_j, \quad \forall j \in \mathcal{N} \]
\[ d_j = \min \left\{ \max \left\{ c_j^{-1}(\omega_j), \frac{\dot{d}_j}{\dot{d}_j} \right\}, \frac{\dot{d}_j}{\dot{d}_j} \right\}, \quad \forall j \in \mathcal{N}. \]

The dynamics (1.29)–(1.32) are automatically carried out by the power system while the active control (1.33) needs to be implemented at controllable loads.

We have the following result regarding the load-controlled system (1.29)–(1.33).

**Theorem 1.15** Starting from any initial point, \((d(t), \dot{d}(t), \omega(t), P(t))\) generated by (1.29)–(1.33) converges to a limit \((d^*, \dot{d}^*, \omega^*, P^*)\) as \(t \to \infty\) such that

1. \(d^*, \dot{d}^*\) is the unique vector of optimal load control for OLC;
2. \(\omega^*\) is the unique vector of optimal frequency deviations for DOLC;
3. \(P^*\) is a vector of optimal line power flows for the dual of DOLC.

Detailed proof of Theorem 1.15 can be referred to [20], and here we only provide a sketch. First, we establish equivalence between the set of optimal points \((\omega^*, P^*)\) of DOLC and its dual and the set of equilibrium points of (1.29)–(1.33). Denote both sets by \(Z^*\). Then, we show that if \((\mathcal{N}, \mathcal{E})\) is a tree network, \(Z^*\) contains a unique equilibrium point \((\omega^*, P^*)\); otherwise (if \((\mathcal{N}, \mathcal{E})\) is a mesh network) \(Z^*\) has an uncountably infinite number (a subspace) of equilibria with the same \(\omega^*\) but different \(P^*\). Next, we use a Lyapunov argument to prove that
every trajectory \((\omega(t), P(t))\) generated by (1.29)–(1.33) approaches a nonempty, compact subset \(Z^+\) of \(Z^*\) as \(t \to \infty\). Hence, if \((N, E)\) is a tree network, any trajectory \((\omega(t), P(t))\) converges to the unique optimal point \((\omega^*, P^*)\); if \((N, E)\) is a mesh network, we show with a more careful argument that \((\omega(t), P(t))\) still converges to a point in \(Z^+\), as opposed to oscillating around \(Z^+\). Theorem 1.15 then follows from Lemma 1.14. The Lyapunov function we find is:

\[
U(\omega, P) = \frac{1}{2} (\omega_G - \omega_G^*)^T M_G (\omega_G - \omega_G^*) + \frac{1}{2} (P - P^*)^T B^{-1} (P - P^*). \tag{1.34}
\]

where \((\omega^*, P^*) = (\omega_G^*, \omega_L^*, P^*) \in Z^*\) is an arbitrary equilibrium point, \(M_G := \text{diag}(M_j, j \in G)\), and \(B := \text{diag}(B_{ij}, (i, j) \in E)\).

This result confirms that, with the proposed algorithm, frequency adaptive loads can balance power and synchronize frequency after a disturbance in the power network, just as the droop control of the generators currently does. Moreover, our design ensures minimal aggregate disutility at equilibrium caused by changes in loads from their nominal power usage. Our result has four important implications. First the local frequency deviation on each bus conveys exactly the right information about the global power imbalance for the loads themselves to make local decisions that turn out to be globally optimal. This allows a completely decentralized solution without explicit communication to or among the loads. Second the global asymptotic stability of the primal-dual algorithm of DOLC suggests that ubiquitous continuous decentralized load participation in primary frequency control is stable. Third we present a “forward engineering” perspective where we start with the basic goal of load control and derive the frequency-based controller and the swing dynamics as a distributed primal-dual algorithm to solve the dual of OLC. In this perspective the controller design mainly boils down to specifying an appropriate optimization problem (OLC). Fourth the opposite perspective of “reverse engineering” is useful as well where, given an appropriate frequency-based controller design, the network dynamics will converge to a unique equilibrium that inevitably solves OLC with an objective function that depends on the controller design.

We illustrate the performance of OLC through the simulation of the 68-bus New England/New York interconnection test system [25]. The single line diagram of the 68-bus system is shown in Fig. 1.9. We run the simulation on Power System Toolbox. The simulation model is more detailed and realistic than the analytic model used above. The detail of the simulation model including parameter values can be found in the data files of the toolbox.

In the test system there are 35 load buses with a total load power of 18.23 GW. We add three loads to buses 1, 7 and 27, each making a step increase of 1 pu (based on 100 MVA). We also select 30 load buses to perform OLC. In the simulation we use the same bounds \([d, d]\) with \(d = -\overline{d}\) for each of the 30 controllable loads, and call the value of \(30 \times \overline{d}\) the total size of controllable loads. We present simulation results below with different sizes of controllable loads.
1.3 Distributed feedback control of networked loads

**Figure 1.9** Single line diagram of the 68-bus test system.

**Figure 1.10** The (a) frequency and (b) voltage on bus 66 for cases (i) no PSS, no OLC; (ii) with PSS, no OLC; (iii) no PSS, with OLC; (iv) with PSS and OLC.

The disutility function of controllable load $d_j$ is $c_j(d_j) = d_j^2/(2\alpha)$, with identical $\alpha = 100$ pu for all the loads.

We look at the impact of OLC on both the steady state and the transient response of the system, in terms of both frequency and voltage. We present the results with a widely used generation-side stabilizing mechanism known as power system stabilizer (PSS) either enabled or disabled. Figures 1.10(a) and 1.10(b) respectively show the frequency and voltage on bus 66, under four cases: (i) no PSS, no OLC; (ii) with PSS, no OLC; (iii) no PSS, with OLC; and (iv) with PSS and OLC. In both cases (iii) and (iv), the total size of controllable loads is 1.5 pu. We observe in Fig. 1.10(a) that whether PSS is used or not, adding OLC always improves the transient response of frequency, in the sense that both...
the overshoot and the settling time (the time after which the difference between the actual frequency and its new steady-state value never goes beyond 5% of the difference between its old and new steady-state values) are decreased. Using OLC also results in a smaller steady-state frequency error. Cases (ii) and (iii) suggest that using OLC solely without PSS produces a much better performance than using PSS solely without OLC. The impact of OLC on voltage, with and without PSS, is qualitatively demonstrated in Fig. 1.10(b). Similar to its impact on frequency, OLC improves significantly both the transient and steady-state of voltage with or without PSS. For instance the steady-state voltage is within 4.5% of the nominal value with OLC and 7% without OLC.

To better quantify the performance improvement due to OLC we plot in Figures 1.11(a)–1.11(c) the new steady-state frequency, the lowest frequency (which indicates overshoot) and the settling time of frequency on bus 66, against the total size of controllable loads. PSS is always enabled. We observe that using OLC always leads to a higher new steady-state frequency (a smaller steady-state error), a higher lowest frequency (a smaller overshoot), and a shorter settling time, regardless of the total size of controllable loads. As the total size of controllable loads increases, the steady-state error and overshoot decrease almost linearly until a saturation around 1.5 pu. There is a similar trend for the settling time. In summary OLC improves both the steady-state and transient performance of frequency, and in general deploying more controllable loads leads to bigger improvement.

To verify the theoretical result that OLC minimizes the aggregate cost (disutility) of load control, Fig. 1.12 shows the cost of OLC over time, obtained by evaluating the quantity defined in (1.21) using the trajectory of controllable and frequency-sensitive loads from the simulation. We see that the cost indeed converges to the minimum.

Our extended work on OLC includes the design of distributed load-side secondary frequency control [26], which not only stabilizes frequency after a disturbance, but also restores frequency to the nominal value and satisfies other power network operational constraints such as inter-area power flow equal to scheduled value and thermal limits on line power flow. Stability of this distributed
load control scheme is also analyzed with a more realistic power network model, which includes nonlinear AC power flow and higher-order generator dynamic models [27].

In summary, the distributed load management schemes discussed in this chapter serve as a good complement to the existing generator control schemes in balancing the power system and maintaining appropriate frequency. They, with provable convergence and optimality and significant performance improvement demonstrated by numerical simulations, address a set of important issues in load management, including: scalability and flexibility of the control system to support autonomous and plug-and-play operations of controllable loads; coordination between controllable loads, as well as coordination between loads and generators, to ensure a predictable and stable system behavior; and optimization of comfort levels of controllable load users or minimization of aggregate disutility due to the change of load power.
References


