Real-Time Deferrable Load Control: Handling the Uncertainties of Renewable Generation

Lingwen Gan
California Inst. of Tech.
lgan@caltech.edu

Adam Wierman
California Inst. of Tech.
adamw@caltech.edu

Ufuk Topcu
University of Pennsylvania
utopcu@seas.upenn.edu

Niangjun Chen
California Inst. of Tech.
ncchen@caltech.edu

Steven H. Low
California Inst. of Tech.
slow@caltech.edu

ABSTRACT

Real-time demand response is potential to handle the uncertainties of renewable generation. It is expected that a large number of deferrable loads, including electric vehicles and smart appliances, will participate in demand response in the future. In this paper, we propose a decentralized algorithm that reduces the tracking error between demand and generation by shifting the power consumption of deferrable loads to match the generation in real-time. At each time step within the control window, the algorithm minimizes the expected tracking error to go with updated predictions on demand and generation. It is proved that the root mean square tracking error of the algorithm vanishes as control window expands, even in the presence of prediction errors.

Keywords

smart grid, deferrable load control, demand response

1. INTRODUCTION

Real-time demand response seeks to induce dynamic load management in response to power supply conditions, e.g., by shifting the power consumption of deferrable loads, including plug-in electric vehicles and smart appliances, to compensate for the random fluctuations in renewable generation.

Work on deferrable load control falls into two categories: direct load control (DLC) [6, 7, 9, 11] which obtains reliable demand-side response and is used mainly for household loads; and pricing mechanisms [1, 2, 10] which allow consumers to implement their own control mechanisms and are used mainly for large industrial loads. The work summarized here focuses on direct load control, more specifically decentralized algorithms so as to handle the computational complexity introduced by a large number of deferrable loads.

This extended abstract summarizes the recent work from [8] where we propose a decentralized real-time deferrable load control algorithm to reduce the tracking error between generation and demand. At every time step, the algorithm minimizes the expected tracking error with updated predictions on future demand and generation. The algorithm has provable performance guarantees—the root mean square tracking error approaches the optimal value as time horizon expands. Finally, we compare the expected tracking error achieved by the proposed algorithm with that obtained by the optimal open-loop control and show that the proposed algorithm has larger improvement over open-loop control as control window expands.

The impact of prediction error is studied explicitly and analytically in the context of the proposed algorithm. Most prior works study prediction error via simulations, e.g., [4,5]; and the works with analytic results mostly take a “worst-case” perspective, e.g., [3, 12], as opposed to the “average-case” perspective of the work here.

2. PROBLEM FORMULATION

The work summarized here studies real-time algorithms for scheduling deferrable loads to compensate for the random fluctuations in renewable generation. Throughout, we consider a discrete-time model over a finite time horizon, which is divided into $T$ time slots $1, \ldots, T$ of equal length.

Assume $N$ deferrable loads $1, \ldots, N$ arrive over time, each requiring certain amount of electricity by certain deadline. Without loss of generality, assume that loads $1, \ldots, N(0) = 0$. Let $d_n(t)$ denote the power consumption of load $n$ at time $t$. Let $g$ denote the generation, and $e$ denote the tracking error between demand and generation.

The optimal deferrable load control (ODLC) problem seeks to minimize the root mean square tracking error (square root of (1a)), subject to power constraints (1c) and energy constraints (1d).

**ODLC:**

$$\min \frac{1}{T} \sum_{t=1}^{T} \left( e(t) - \frac{1}{T} \sum_{\tau=1}^{T} e(\tau) \right)^2$$

over $d_n(t), e(t), \forall n, t$

$$s.t.\ e(t) = \sum_{n=1}^{N} d_n(t) - g(t), \forall t; \quad (1b)$$

$$d_n(t) \leq d_n(t) \leq \bar{d}_n(t), \forall n, t; \quad (1c)$$

$$\sum_{t=1}^{T} d_n(t) = D_n, \forall n. \quad (1d)$$

Here $d_n, \bar{d}_n$ and $D_n$ are externally specified constants. Power constraints (1c) impose power to be bounded and energy...
constraints (1d) impose energy consumption to be constant. Finally, deferrable loads arrive randomly over time. Let
\[ a(t) := \sum_{n=N(\tau)+1}^{N(t)} D_n, \quad t = 1, \ldots, T \]
denote the energy request of deferrable loads that arrive at time \( t \), and assume \( \{a(t)\}_{t=1}^{T} \) to be a sequence of uncorrelated random variables with mean \( \lambda \) and variance \( \sigma^2 \). Further, let \( A(t) := \sum_{\tau=t+1}^{T} a(\tau) \) denote the total energy requested after time \( t \).

At time \( t \), real-time algorithms know 1) present deferrable loads, i.e., \( d_n, \overline{a}_n \), and \( D_n \) for \( n \leq N(t) \); 2) expectation of future deferrable load \( \mathbb{E}(A(t)) \); and 3) prediction \( g(t) \) of future deferrable load requests.

3. ALGORITHM DESIGN

Key contribution of this work is the following decentralized algorithm that solves the ODLC problem in real-time. The key idea is to introduce a pseudo deferrable load \( q \), simulated at the utility, to represent the power consumption of future deferrable loads. More specifically, solve the following problem at every time step \( t \).

**ODLC-t:** min \( \sum_{\tau=t}^{T} \left( e(\tau) - \frac{1}{T-t+1} \sum_{s=1}^{T} e(s) \right)^2 \)
over \( d_n(\tau), q(\tau), e(\tau), \ n \leq N(t), \tau \geq t \)
s.t. \( e(\tau) = \sum_{n=1}^{N(t)} d_n(\tau) + q(\tau) - g(\tau), \ \tau \geq t; \)
\( d_n(\tau) \leq d_n(\tau) \leq \overline{a}_n(\tau), \ n \leq N(t), \tau \geq t; \)
\( \sum_{\tau=t}^{T} d_n(\tau) = D_n(t), \ n \leq N(t); \)
\( g(\tau) \leq \overline{q}(\tau) \leq \overline{q}(\tau), \ \tau \geq t; \)
\( \sum_{\tau=t}^{T} q(\tau) = \mathbb{E}(A(t)) \).

where \( P_n(t) = P_n - \sum_{\tau=t}^{\tau-1} P_n(\tau) \) is the energy to be consumed at or after time \( t \) and \( g, \overline{q} \) are externally specified constants with \( g(\tau) = \overline{q}(\tau) = 0 \).

Algorithm 1 solves ODLC-t at every step \( t \). Specifically, let \( O(t) \) denote the set of optimal schedules \( (d,q) \) to ODLC-t, and define \( \text{dist}(d,Q(t)) := \min_{(d',q') \in O(t)} \| d - d' \| \) as the distance from \( d \) to \( O(t) \) for \( t = 1, \ldots, T \).

**Theorem 1.** At time \( t = 1, \ldots, T \), the deferrable load schedules \( d^{(k)} \) obtained by Algorithm 1 converge to optimal schedules to ODLC-t, i.e., \( \text{dist}(d^{(k)},O(t)) \to 0 \) as \( k \to \infty \).

There is a simple characterization of the solutions to ODLC-t. Specifically, at time \( t = 1, \ldots, T \), a feasible schedule \( (d,q) \) is called \( t \)-valley-filling, if there exists \( C(t) \in \mathbb{R} \) such that
\[ \sum_{n=1}^{N(t)} d_n(\tau) + q(\tau) - g(\tau) = C(t), \quad \tau = t, \ldots, T. \]

**Theorem 2.** At time \( t = 1, \ldots, T \), a \( t \)-valley-filling deferrable load schedule, if exists, solves ODLC-t. Furthermore, in such cases, all optimal schedules to ODLC-t have the same aggregate load.

This characterization serves as the basis for the performance analysis. Note that \( t \)-valley-filling schedules tend to exist if there is a large number of deferrable loads.

### Algorithm 1 Real-time decentralized load control

At time step \( t = 1, \ldots, T \).

**Input:** the utility knows \( g(t) \) and \( N(t) \), each deferrable load \( n \) knows \( D_n(t), \overline{a}_n \) and \( d_n \). The number \( K \) of iterations.

**Output:** \( d_n(t) \) for \( n = 1, \ldots, N(t) \).

1. Set \( k \leftarrow 0 \). Each deferrable load \( n \leq N(t) \) initializes its schedule \( (d_n^{(k)}(\tau), \overline{q}(\tau)) \) as
\[ d_n^{(0)}(\tau) = \begin{cases} d_n^K(\tau) & \text{if } n \leq N(t-1) \\ 0 & \text{if } n > N(t-1) \end{cases}, \quad \tau \geq t \]
where \( d_n^K(\tau) \) is the schedule in iteration \( K \) of the previous time step \( t-1 \).
2. The utility solves
\[ \min_{\tau=t}^{T} \sum_{n=1}^{N(t)} \left( \sum_{\tau=t}^{T} d_n^{(k)}(\tau) + q(\tau) - g(\tau) \right)^2 \]
over \( q(t), \ldots, q(T) \).

s.t. \( q(\tau) \leq \overline{q}(\tau), \ \tau \geq t; \)
\( \sum_{\tau=t}^{T} q(\tau) = \mathbb{E}(A(t)) \)
to obtain \( (q^{(k)}(\tau))_{t \geq t} \), calculates \( p^{(k)}(\tau) \) as
\[ p^{(k)}(\tau) \leftarrow \frac{1}{N(t)} \left( \sum_{n=1}^{N(t)} \left( d_n^{(k)}(\tau) + q^{(k)}(\tau) - d_n(\tau) \right) \right) \]
for \( \tau \geq t \), and broadcasts \( p^{(k)}(\tau) \) to deferrable loads.
3. Each deferrable load \( n \leq N(t) \) solves
\[ \min_{\tau=t}^{T} \sum_{\tau=t}^{T} p^{(k)}(\tau) d_n(\tau) + \frac{1}{2} \left( d_n(\tau) - d_n^{(k)}(\tau) \right)^2 \]
over \( d_n(T), \ldots, d_n(T) \).

s.t. \( d_n(\tau) \leq d_n(\tau) \leq \overline{a}_n(\tau), \ \tau \geq t; \)
\( \sum_{\tau=t}^{T} d_n(\tau) = D_n(t) \),
to obtain \( d_n^{(k+1)} \), and reports \( d_n^{(k+1)} \) to the utility.
4. Set \( k \leftarrow k + 1 \). If \( k < K \), go to Step (ii).
5. Deferrable load \( n \leq N(t) \) sets \( d_n(t) \leftarrow d_n^K(\tau) \) and \( D_n(t+1) \leftarrow D_n(t) - d_n(t) \).

4. PERFORMANCE EVALUATION

In this section, we study the impact of the uncertainties about generation and deferrable load arrival, on the mean square tracking error achieved by Algorithm 1; as well as the improvement of Algorithm 1 over open-loop control.

A generation prediction model is necessary to answer these two questions. In this paper, generation \( g \) is modeled as a random deviation \( \delta g \) around its expectation \( \overline{g} \) as illustrated in Figure 1. The process \( \delta g \) is further modeled as uncorrelated random variables \( e(1), \ldots, e(T) \), each with mean 0 and variance \( \sigma^2 \), passing through a causal filter with impulse response \( f \). Assume \( f(0) = 1 \), then \( f(\tau) = 0 \) for \( \tau < 0 \) and
\[ \delta g(\tau) = \sum_{s=1}^{\tau} e(s)f(\tau-s), \quad \tau = 1, \ldots, T. \]
Figure 1: Diagram of the notation and structure of the model for generation.

The prediction error \( g_t \) at time \( t = 1, \ldots, T \) is assumed to be

\[ g_t = g(\tau_t) + \sum_{s=1}^{t} e(s) f(\tau - s), \quad \tau = 1, \ldots, T. \]

Throughout, we assume that a \( t \)-valley-filling schedule exists at every time step \( t \), under which one has

\[ e(t) = \frac{1}{T - t + 1} \left( \sum_{n=1}^{N(t)} D_n(t) + E(A(t)) - \sum_{\tau=t}^{T} g_t(\tau) \right). \]

### 4.1 Expected mean square tracking error

The expected mean square tracking error

\[ E(V) = E \left[ \frac{1}{T} \sum_{t=1}^{T} e(t)^2 \right] \]

achieved by Algorithm 1 is given in the following theorem.

**Theorem 3.** The expected mean square tracking error \( E(V) \) obtained by Algorithm 1 is

\[ E(V) = \frac{s^2}{T} \sum_{t=2}^{T} \frac{1}{t} \sigma^2 \sum_{t=0}^{T-1} F(t) \frac{T - t - 1}{t + 1} \]

where \( F(t) = \sum_{s=0}^{t} f(s) \) for \( t = 0, \ldots, T \).

Theorem 3 explicitly states how the generation prediction error (\( \sigma \)) and deferrable load prediction error (\( s \)) affect the expected mean square tracking error \( E(V) \). It follows immediately that \( E(V) \) tends to 0 as the predictions get precise, i.e., \( \sigma \to 0 \) and \( s \to 0 \).

**Corollary 1.** The expected mean square tracking error \( E(V) \to 0 \) as \( \sigma \to 0 \) and \( s \to 0 \).

Besides, the impact of the time-correlation (given by \( f \)) in generation prediction error on \( E(V) \) is captured via \( F \).

**Corollary 2.** If \( f(t) \sim O(t^{-1/2-\alpha}) \) for some \( \alpha > 0 \), then \( E(V) \to 0 \) as \( T \to \infty \).

### 4.2 Improvement over open-loop control

In this section, we compare the expected mean square tracking error \( E(V) \) obtained by Algorithm 1, with that (denoted \( E(V') \)) obtained by the optimal open-loop control. Open-loop control uses predictions at the beginning of the time horizon. We assume all deferrable loads arrive at time 1, i.e., \( N(1) = N \), in this section since otherwise open-loop control cannot obtain a schedule for all deferrable loads.

The expected mean square tracking error \( E(V') \) obtained by optimal open-loop control is given in the following lemma.

**Lemma 1.** If \( N(1) = N \), then the expected mean square tracking error \( E(V') \) obtained by the optimal open-loop control is

\[ E(V') = \frac{s^2}{T^2} \sum_{t=0}^{T-1} \left( T(T - t) f^2(t) - F^2(t) \right). \]

Comparing \( E(V) \) and \( E(V') \) shows that Algorithm 1 always obtains a smaller expected mean square tracking error than the optimal open-loop control. Specifically,

**Corollary 3.** If \( N(1) = N \), then

\[ E(V') - E(V) = \frac{\sigma^2}{T} \sum_{t=2}^{T} \left( \sum_{m=0}^{t-1} \sum_{n=0}^{t-1} (f(m) - f(n))^2 \right) \geq 0. \]

Corollary 3 highlights that Algorithm 1 is guaranteed to obtain a smaller expected mean square tracking error than the optimal open-loop control. The next step is to quantify how much smaller \( E(V) \) is in comparison with \( E(V') \).

To do this we compute the ratio \( E(V') / E(V) \) for two representative impulse responses \( f(t) \), and the results are summarized in the following two corollaries.

**Corollary 4.** If \( N(1) = N \) and

\[ f(t) = \begin{cases} 1 & \text{if } 0 \leq t < \Delta \\ 0 & \text{otherwise} \end{cases} \]

for some \( \Delta > 0 \), then

\[ \frac{E(V')}{E(V)} = \frac{T/\Delta}{\ln(T/\Delta)} \left( 1 + O \left( \frac{1}{\ln(T/\Delta)} \right) \right). \]

**Corollary 5.** If \( N(1) = N \) and \( f(t) = a^t \) for some \( a \in (0, 1) \), then

\[ \frac{E(V')}{E(V)} = \frac{1 - a}{1 + a} T \ln T \left( 1 + O \left( \frac{1}{\ln(T/\Delta)} \right) \right). \]

5. REFERENCES


