Generation Following with Thermostatically Controlled Loads via Alternating Direction Method of Multipliers Sharing Algorithm

Eric M. Burger, Scott J. Moura

Energy, Control, and Applications Lab, University of California, Berkeley

Abstract

A fundamental requirement of the electric power system is to maintain a continuous and instantaneous balance between generation and load. The intermittency and uncertainty introduced by renewable energy generation requires expanded ancillary services to maintain this balance. In this paper, we examine the potential of thermostatically controlled loads (TCLs), such as refrigerators and electric water heaters, to provide generation following services in real-time energy markets (1 to 5 minutes). Previous research in this area has primarily focused on the development of centralized control schemes with an aggregate TCL model. An objective of our approach is to enable each TCL to model and control its dynamics independently and to use distributed convex optimization techniques to allow a central aggregator to influence, but not directly control, the behavior of the population. To control the non-linear dynamics of hysteretic dead-band systems in a manner suitable for convex optimization, we introduce an alternative control trajectory representation of the TCLs and their discrete input signals. This approach allows us to approximate the control of a TCL as a convex program and to produce a solution that can be interpreted stochastically for implementation. To perform distributed optimization across large populations of TCLs, we apply a variation of the alternating direction method of multipliers (ADMM) algorithm. The objective of the distributed optimization algorithm is to enable an aggregator to coordinate with a population of TCLs and to increase or decrease the total power demand according to a control signal. We include experimental results in which different populations of TCLs with varying levels heterogeneity are optimized to provide 5-minute ahead generation following services. We numerically demonstrate the algorithm's potential for controlling a TCL population's power demand within a definable error tolerance.

Keywords: Smart grid, Distributed optimization, Alternating Direction Method of Multipliers (ADMM), Ancillary services, Generation following, Thermostatically Controlled Loads (TCL)

1. Introduction

1.1. Background and Motivation

The variability of renewable energy resources, particularly wind and solar, poses a challenge for power system operators. Namely, as renewable penetration increases it will be necessary for operators to procure more ancillary services, such as regulation and load following, to maintain balance between generation and load [1][2][3][4][5]. Researchers have proposed a number of solutions for employing residential demand response to shift flexible loads based on a price signal, helping to reduce the need for load following [6][7][8]. In the long-term, grid-scale storage technologies (e.g. flywheels, batteries, etc.) are sure to play a major role in providing these ancillary services [9][10][11]. In the near-term, responsive thermostatically controlled loads (TCLs) have a high potential for providing such ancillary services [12][13].

This paper investigates the challenge of controlling a *heterogeneous* TCL population to perform an ancillary service, specifically 5-minute ahead gener-

*Corresponding author: Eric M. Burger,
Email: ericburger@berkeley.edu
**Affiliation Address: Energy, Control, and Applications Lab (eCAL), 611 Davis Hall, Department of Civil and Environmental Engineering, University of California, Berkeley, Berkeley, CA 94720, USA.

ation following. For experimental purposes, we define generation following as the complement of load following whereby loads are employed to smooth the power generation from renewable energy sources.

The advantages of responsive TCLs over large storage technologies include: 1) they are well-established technologies; 2) they are distributed throughout the power system thus providing spatially and temporally distributed actuation; 3) they employ simple and fast local actuation well-suited for real-time control; 4) they are robust to outages of individuals in the population; and 5) they, on the aggregate, can produce a quasi-continuous response despite the discrete nature of the individual controls [13][14][15]. These characteristics make TCLs suitable for both direct load management programs, in which a utility can actuate TCLs to meet objectives like peak demand reduction or emergency situation handling, and indirect load management programs, in which utilities use price signals, rebates, and subsidies to incentivize the shifting or reduction of TCL power demands [3]. We refer the reader to [3] for a comprehensive study of utility-scale load management, to [13] for a discussion of the advantages and disadvantages of TCLs compared to grid-scale storage technologies, and to [14] for a look into the potential costs and revenues of demand response with TCLs.

Additionally, because TCLs are controlled according to a temperature setpoint, customers are generally indifferent to precisely when energy is consumed as long as the temperatures are maintained within a dead-band range. This natural flexibility makes TCLs a promising candidate for participating in power system services.

1.2. Contributions

Novel contributions of this work include:

- The alternative control trajectory representation – a novel approach for representing the control of agents with non-convex constraints as a convex program. The resulting convex program provides a solution that can be interpreted stochastically for implementation.
- The application of an alternating direction method of multipliers (ADMM) sharing algorithm for the distributed convex optimization of TCLs. Each TCL agent optimizes a private objective function, while the central aggregator iteratively updates an incentive variable to drive the population towards a global objective, such as generation following. By distributing the computation using ADMM, each TCL is able to optimize its objective in parallel and the population can efficiently converge to a global solution.
- By applying the alternative control trajectory representation and alternating direction method of multipliers sharing algorithm, this paper demonstrates the control of a population of systems with integer states using a convex algorithm. This is a fundamental gap that we bridge.

1.3. Literature Review

1.3.1. Early TCL Modeling and Cold Load Pickup

Research into the modeling and control of TCLs began with applications to peak shaving and cold load pickup in power systems. Cold load pickup is a phenomenon which occurs in a distribution network due to the restoration of power after an extended outage. Normally, the power demand of thermostatically controlled loads is desynchronized. However, following outages, TCLs will simultaneously demand full power, contributing to the cold load pickup peak. To address this problem, researchers focused on methods for modeling and reducing TCL demand during cold load pickup events as well as peak demand hours.

The earliest examples of such work include the Ihara and Scwheppe paper on space conditioning during cold load pickup [16] and the Chong and Debs paper on individual and aggregation load models [17], both of which used individual TCL models to describe load dynamics. In [18], Mortensen and Haggerty develop a discrete-time TCL model, which was later adapted by Ucak to model heterogeneous TCL populations [19]. In [20], Pahwa and Brice describe the modeling and parameter estimation of residential air conditioning loads as well as a basic aggregation method. Malhame and Chong's study [21] is among the first reports to use stochastic analysis to develop an aggregate model of a TCL population. The resulting coupled Fokker-Planck equations, derived in [21], define the aggregate behavior of a homogeneous population.

While efforts were made in these early works to model the aggregate demand of a TCL population and to propose control schemes for reducing demand during peak hours and cold load pickup events, the most meaningful contributions focused on the modeling and parameter estimation of individual TCLs.

1.3.2. Aggregate TCL Modeling and Centralized Control

Recent research efforts have focused on the modeling of TCL populations using aggregation methods. A key objective of this research is to develop and evaluate methods for characterizing the temperature density evolution of a TCL population. By incorporating centralized control strategies, aggregated TCL populations are able to provide ancillary power system services like load following and regulation rather than just load reduction. In [15], one of the first papers to develop a modeling and control strategy that allows TCLs to perform ancillary services, Callaway uses a linearized Fokker-Planck model to describe the aggregated behavior of a TCL population. Direct load control is achieved by broadcasting a single time-varying setpoint temperature offset signal to every agent. Numerical results demonstrate how small perturbations to the setpoint can enable TCLs to perform wind generation following. Later work builds upon concepts in [15] by considering sliding mode control [22], proportional-integral control [23], linear quadratic regulators [24], and switching rate broadcast actuation [25].

In [14] and [26], Mathieu, Koch, and Callaway propose a proportional controller which, at each time step, broadcasts a switching probability, η, to all the TCLs in the population. If $\eta < 0$, all TCLs that are on must switch off with a probability of η and if $\eta > 0$, TCLs that are off switch on with a probability of η. In [27], Koch et al. employ a linear time-invariant (LTI) representation of a TCL population. As in [22], a "state bin" modeling framework is used and the aggregate probability mass is allowed to move through these bins. A Markov Chain-based approach is used to predict the evolution of the heterogeneous TCL population.

Similar work can be found in [28], [29], and [30] where Zhang et al. use a state bin concept to represent the evolution of the TCLs and introduce clustering to better account for heterogeneity. In [28], a second-order aggregate model for a heterogeneous population of TCLs is developed. To address the high state-space dimensionality of this model, a complexity reduction method and reduced-order model is proposed in [29]. In [30], the second-order aggregate model is used to simulate a population of heating, ventilation, and air-conditioning (HVAC) systems and a novel method for incorporating minimum dwell time is proposed. Specifically, Zhang et al. define a state which represents the number of off TCLs that are "locked" and will not turn on in response to the central control signal. Thus, the individual TCLs are able to locally enforce dwell times and the aggregator is able to adjust the control signal to account for locked TCLs.

A significant body of research has grown out of the above literature in response to open challenges around aggregate model efficacy and efficiency, modeling and control framework limitations, and unaddressed system constraints. In [31], Moura et al. develop a diffusion-advection partial differential equation (PDE) model and a parameter identification scheme for an aggregated population of heterogeneous TCLs, alleviating the need for prior knowledge of TCL parameters. In [32], Ghaffari et al. develop a deterministic hybrid PDE-based model capable of representing a heterogeneous TCL population and apply a uniform dead-band shifting strategy for control. In [33], Vrettos and Anderson research the aggregation of TCLs to simultaneously provide frequency and voltage regulation services, recognizing that solving these problems separately can produce suboptimal solutions. Iacovella et al. introduce the use of tracer TCLs in [34]. These virtual tracer devices represent the state density distribution of a cluster of heterogeneous TCLs. The approach enables the use of reduced-order aggregate models with control achieved via a single broadcasted signal.

In [35] and [36], Mathieu et al. build upon previous work in [14][26] to employ a state bin modeling framework with a "non-disruptive" approach in which the TCL's temperature is maintained within the existing dead-band. Hao et al. also consider a non-disruptive approach in [37] using a battery model of the TCL population and a priority stack strategy to determine which TCLs to control at a given time step.

1.3.3. Decentralized TCL Control for Frequency Services

Recognizing that system frequency is a universally available indicator of supply-demand imbalance, a number of researchers have developed fully decentralized techniques for performing frequency services with TCLs. In [38], Short et al. show the suitability of TCLs to perform frequency services using system frequency as a control signal and the potential for a population of TCLs to respond to a sudden loss of generation. This demand response capability reduces the dependence of grid operators on rapidly deployable backup generation.

In [2], Xu et al. develop a TCL model in which devices adjust their setpoints linearly according to the system frequency, allowing the population to act as a fast frequency controlled reserve. To address problems of long-term instability, Angeli and Kountouriotis develop a decentralized stochastic controller in [39] that is capable of maintaining desynchronization among the TCLs while regulating overall power consumption. In [40], Tindemans et al. present a stochastic controller whereby each TCL in the population independently targets a reference power profile. The result is a stable and fully decentralized system that requires only the locally available control signals of frequency and time.

1.4. A Distributed Approach

There are a number of advantages to the modeling and control approaches described above. Firstly, the aggregated models are based upon linear representations of TCL dynamics. This makes the aggregated models well suited for a variety of established control and optimization techniques. Moreover, these models are good at prediction and control over small time scales (i.e. seconds and milliseconds), making them ideal for producing fast short-term responses (e.g. frequency regulation) [26][35].

A limitation of these aggregate models is low model fidelity and the inability to incorporate device specific dynamics. Note the literature is rich with techniques for multi-state thermal modeling of heating, ventilation, and air-conditioning (HVAC) systems in buildings including solar gain estimation and multi-zone state estimation [41][42][43][44][45]. Because aggregate models are not amenable to the incorporation of device specific, nonlinear, or nonparametric models, they are incapable of leveraging the work of these and other researchers. At larger time scales (i.e. minutes and hours), higher model fidelity becomes very important for the accurate forecasting of TCL power demand. By employing basic linear models, particularly when modeling the complex dynamics of HVAC systems in buildings, aggregated TCL modeling approaches are poorly suited for producing accurate long-term responses (e.g. load-shifting) [42][43][44]. Hao et al. [37], for example, derive a "generalized battery model" to predict aggregate TCL flexibility. Even with a simple single-state TCL model, summing the set of flexible trajectories involves an arduous Minkowski sum that they approximate through bounding sets. Recent work by Tindemans et. al. pursues a stochastic single TCL model that can be distributed [40]. However, this model is mathematically formulated as a partial differential equation that fundamentally relies on a single state to represent temperature. In this manuscript, we pursue a method extendible to the multi-state models that characterize data collected from real-world TCLs [42][43][44].

An additional limitation of linear models is that they permit the TCLs to short-cycle. Short-cycling is a behavior in which a TCL turns on and/or off for a short amount of time. This behavior is produced by linear controllers and optimization techniques when it is optimal for the temperature to oscillate around a point, such as the edge of the dead-band or the temperature setpoint. Over time, this short-cycling will reduce the efficiency and operational life of the hardware within a TCL. Efforts to prevent short-cycling, such as preferential binning, priority/preferential switching, and lockout estimation, are made in [35][36][30][37]. However, the preferential techniques employed in [35][36] cannot guaranteed the prevention of short cycling and the lockout estimation in [30][37] requires centralized knowledge of the minimum dwell times of every agent in the population.

A key advantage of decentralized TCL control methods is the reduced or eliminated need for communication infrastructure. However, by relying on system frequency as the control signal, applications are limited to frequency regulation and real-time load shaping. To produce long-term responses (e.g. load-shifting), it is necessary for a grid entity to define the service objective, to forecast network states, and to coordinate or otherwise control the TCL population to meet the objective. Thus, the control paradigm shifts from decentralized to centralized or distributed control.

To control a TCL population to produce long-term responses in a manner that is agnostic of the individual TCL models (e.g. device specific, nonlinear, nonparametric) and that enables the incorporation of locally defined constraints (e.g. short-cycling), this manuscript presents a novel TCL modeling technique and distributed control approach. This work diverges from the above literature in the following respects:

- This paper presents a distributed control scheme with a centralized aggregator via

ADMM. Related distributed control schemes use consensus coordination [46], distributed model predictive control [47][48], iterative load profile aggregation [49], multi-agent systems [6], and game-theory [8][50].

- In this paper, all TCL parameters, objectives, and constraints remain private. Each TCL is simulated locally and independently of the population. The only information that a TCL communicates with the central aggregator is its predicted power trajectory. Therefore, if necessary, TCL parameter identification can be performed locally [51].
- We do not employ an aggregate model of the TCL population. Thus, rather than modeling the entire population, the central aggregator is only responsible for updating an incentive variable that drives the population towards a desired behavior.
- There is no requirement that each TCL in the population employs the same model structure or local control scheme. The only requirement is that the TCL is able to produce predictions of its power demand under multiple alternative control scenarios. While we employ a hybrid state TCL model in this manuscript, this is not restrictive and the distributed optimization technique is compatible with a variety of different TCL modeling approaches.
- We do not use continuous setpoint control. In this paper, all temperature setpoint offsets are integer valued and therefore easily implementable.
- Individual TCLs are not required to participate at every time step. Because the TCL population is not centrally modeled, the distributed scheme is robust to an arbitrarily large loss or acquisition of agents.
- Our proposed modeling and control approach is capable of honoring non-convex constraints, such as minimum dwell time - a critically important practical constraint that eliminates compressor short-cycling.
- Our proposed modeling and control approach is directly extendible to multi-state and nonlinear TCL models that characterize many TCLs in practice, as shown by the system identification studies in [42][43][44].

For the distributed optimization of a TCL population, we present a variant of the alternating direction method of multipliers (ADMM) algorithm known as sharing ADMM [52]. Due to its parallelizability and convergence characteristics, the sharing ADMM algorithm is generally applicable to the minimization of distributed agents. Furthermore, past research on the application of ADMM to the balancing of generators, fixed loads, deferrable loads, and storage devices has demonstrated the suitability of ADMM to efficiently solve large convex optimization problems in parallel [53]. In this paper, we develop a formulation of the ADMM algorithm to enable a TCL population to perform 5-minute power generation following. Under our proposed control scheme, each TCL optimizes its behavior according to both a private objective function (which primarily enforces feasibility) and a shared objective function (which follows a generation signal). Optimization is achieved by iteratively updating a shared incentive variable, which is calculated and broadcast by a central aggregator, until the population converges to a feasible solution.

1.5. Paper Outline

This paper is organized as follows. Section 2 discusses the TCL model and the alternative control trajectory representation. Section 3 overviews the sharing ADMM algorithm. Section 4 formulates sharing ADMM for distributed TCL control. Section 5 provides numerical examples of our proposed algorithms and highlights its applicability to highly heterogeneous populations. Finally, Section 6 summarizes key results. Nomenclatures and notation used in this paper are defined in the Appendix.

2. TCL Model and Optimization

2.1. Hybrid State Model

Each TCL is modeled using the hybrid state discrete time model [15][16][18]

$$T^{n+1} = \theta_1 T^n + (1-\theta_1)(T^n_\infty + \theta_2 m^n) + \theta_3$$
$$m^{n+1} = \begin{cases} 1 & \text{if } T^{n+1} < T_{set} - \frac{\delta}{2} \\ 0 & \text{if } T^{n+1} > T_{set} + \frac{\delta}{2} \\ m^n & \text{otherwise} \end{cases} \tag{1}$$

where state variables $T^n \in \mathbf{R}$ and $m^n \in \{0,1\}$ denote the temperature of the conditioned mass and the discrete state (on or off) of the mechanical system, respectively. Additionally, $n = 1, 2, \ldots, N_t$ denotes the integer-valued time step, $T_\infty^n \in \mathbf{R}$ the ambient temperature (°C), $T_{set} \in \mathbf{R}$ the temperature setpoint (°C), and $\delta \in \mathbf{R}$ the temperature dead-band width (°C).

In this paper, we define the time elapsed between each time step as $h = 1/60$ (hours). The parameter θ_1 represents the thermal characteristics of the conditioned mass as defined by $\theta_1 = \exp(-h/RC)$ where C is the thermal capacitance (kWh/°C) and R is the thermal resistance (°C/kW), θ_2 the energy transfer to or from the mass due to the systems operation as defined by $\theta_2 = RP$ where P is the rate of energy transfer (kW), and θ_3 is an additive process noise accounting for energy gain or loss not directly modeled. We assume that θ_3 is normally distributed with variance $h\sigma^2$ (bulk units of °C^2). In this paper, we assume a noise standard deviation σ of 0.01°C/$\sqrt{sec}$ or 0.6°C/$\sqrt{hr}$ [13].

The power demand of a TCL at each time step is defined by

$$p^n = \frac{|P|}{COP} m^n \tag{2}$$

where $p^n \in \mathbf{R}$ is the electric power demand (kW) and COP the coefficient of performance.

The sign conventions in (1) assume that the TCL is providing a heating load and that P (and thus θ_2) is positive. Therefore, we expand the m-update statement to account for both heating and cooling loads. Additionally, in this paper, the optimal control of each TCL is based on setpoint manipulation. In other words, at each time step n, a TCL will either enforce T_{set} or move the setpoint by u^n. While we define u^n such that the setpoint may be adjusted at each time step, in practice, we employ a single adjustment over multiple consecutive time steps. The TCL model can now be expressed as

$$T^{n+1} = \theta_1 T^n + (1 - \theta_1)(T_\infty^n + \theta_2 m^n) + \theta_3$$

$$m^{n+1} = \begin{cases} 1 & \text{if } \theta_2 > 0 \text{ and} \\ & \quad T^{n+1} < T_{set} - \frac{\delta}{2} + u^n \\ 0 & \text{if } \theta_2 > 0 \text{ and} \\ & \quad T^{n+1} > T_{set} + \frac{\delta}{2} + u^n \\ 1 & \text{if } \theta_2 < 0 \text{ and} \\ & \quad T^{n+1} > T_{set} + \frac{\delta}{2} + u^n \\ 0 & \text{if } \theta_2 < 0 \text{ and} \\ & \quad T^{n+1} < T_{set} - \frac{\delta}{2} + u^n \\ m^n & \text{otherwise} \end{cases} \tag{3}$$

where $u^n \in \mathbf{R}$ is the setpoint change at time step n. While u^n may, by definition, take on any value in $\mathbf{R}$, in this paper we will only consider integer changes to the temperature setpoint (i.e. $u^n \in \mathbf{Z}$).

As noted in [15][18], the discrete time model implicitly assumes that all changes in mechanical state occur on the time steps of the simulation. In this paper, we will assume that this behavior reflects the programming of the systems being modeled. In other words, we will assume that the TCLs have a thermostat sampling frequency of $1/h$ Hz or once per minute.

Finally, in this paper, we will emphasize *heterogeneous* TCLs populations and thus vary R, C, P, and COP for each agent in the population, as discussed in Section 4. Because R, C, and P define the thermal mass and rate of heat transfer, the parameters govern the system dynamics. The COP parameter does not impact the system dynamics but rather scales the magnitude of the electricity power demand.

2.2. Alternative Control Trajectory Representation

In this section, we consider the optimization of a TCL represented by the hybrid state model above. While the model presents an intuitive representation of a dead-band control system, the discrete and piece-wise nature of the m-update statement poses a numerical challenge for optimal control. In particular, if the TCL's temperature is near the setpoint (i.e. away from the upper and lower bound), then the mechanical state m^{n+1} is dependent upon the previous state m^n.

This dependency, as well as the binary on/off state, makes the system combinatorial and therefore non-convex. There are optimization approaches, such as dynamic programming and genetic algorithms, that are well suited for solving such a non-convex problem to identify an optimal control strategy. However, these approaches are poorly suited for distributed optimization problems because the number of optimization variables is intractable for real-time control.

Therefore, we introduce a novel approach for representing the control of non-linear systems in a manner suitable for linear/convex programming. Put simply, we simulate the system under multiple feasible alternative control inputs in order to generate a discrete set of output trajectories. These alternative control trajectories can be incorporated into a convex program as a linear constraint, thereby enforcing feasibility.

To begin, we define N_a alternative control inputs for N_t time steps

$$u_j = (u_j^1, u_j^2, \dots, u_j^{N_t}) \quad \forall\, j = 1, \dots, N_a \tag{4}$$

with variable $u_j \in \mathbf{R}^{N_t}$ and $u_j^n \in S_u$ for $n = 1, ..., N_t$, where $S_u \subset \mathbf{Z}$ is the constraint set of feasible/allowed setpoint changes. Note that the number of alternative control inputs and the method for generating each u_j will depend on the application (see Section 4.1 for the method used in this paper).

Next, for each control input u_j, we simulate the TCL model defined in (3) while imposing any additional physical, mechanical, or numerical constraints, such as a minimal dwell time. Given the simulation results, we generate N_a feasible alternative trajectories as defined by the state variables T and m. Since the power demand p^n is linearly related to the mechanical state m^n, we can also define the set of alternative power demand trajectories.

$$\begin{aligned} T_j &= (T_j^2, T_j^3, \dots, T_j^{N_t+1}) \\ m_j &= (m_j^2, m_j^3, \dots, m_j^{N_t+1}) \\ p_j &= (p_j^2, p_j^3, \dots, p_j^{N_t+1}) \\ &\forall\, j = 1, \dots, N_a \end{aligned} \tag{5}$$

The input and output variables can be expressed compactly as

$$\begin{aligned} \mathbf{U} &= (u_1, u_2, \dots, u_{N_a}) \\ \mathbf{T} &= (T_1, T_2, \dots, T_{N_a}) \\ \mathbf{M} &= (m_1, m_2, \dots, m_{N_a}) \\ \mathbf{P} &= (p_1, p_2, \dots, p_{N_a}) \end{aligned} \tag{6}$$

with variables $\mathbf{U}$, $\mathbf{T}$, $\mathbf{M}$, and $\mathbf{P}$ representing the set of all u_j, T_j, m_j, and p_j sets for $j = 1, \dots, N_a$. Naturally, we can also view $\mathbf{U}$, $\mathbf{T}$, $\mathbf{M}$, and $\mathbf{P}$ as matrices $\in \mathbf{R}^{N_a \times N_t}$ such that the rows represent the alternative trajectories and the columns represent the time step n. It should be noted that the function defined by (3) is not one-to-one (i.e. a function f such that $f(u_j) = m_j$ is not injective). In other words, the distinctness of u_j does not guarantee the distinctness of T_j, m_j, and p_j. Thus, for computational efficiency, if T_j, m_j, or p_j are equal to any previously generated output for $j = 2, \dots, N_a$, then each set u_j, T_j, m_j, and p_j should be excluded from $\mathbf{U}$, $\mathbf{T}$, $\mathbf{M}$, and $\mathbf{P}$. We define the number of *distinct* alternative control trajectories as N_d such that $N_d \in \{1, \dots, N_a\}$.

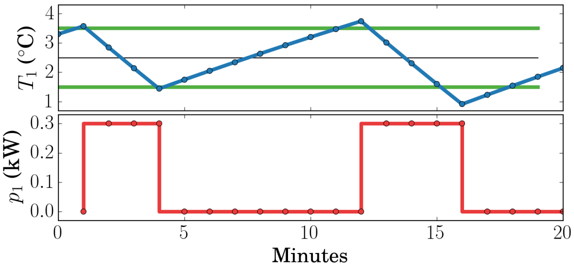

Figure 1: Examples of alternative temperature t_1 and power p_1 trajectories given input u_1

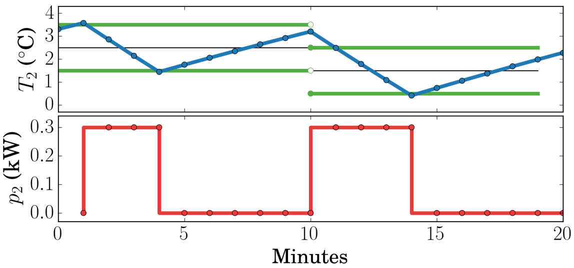

Figure 2: Examples of alternative temperature t_2 and power p_2 trajectories given input u_2

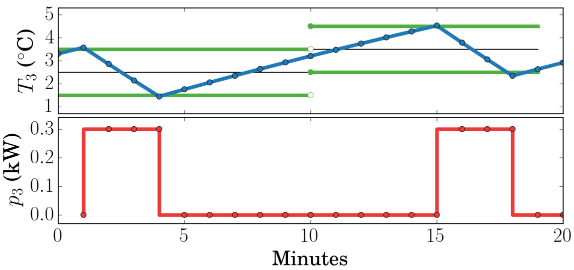

Figure 3: Examples of alternative temperature t_3 and power p_3 trajectories given input u_3

Figures 1, 2, and 3 illustrate an example of a TCL (specifically, a refrigerator) with $N_a = 3$ alternative trajectories. In the example, each alternative input u_j for $j = 1, 2, 3$ is $\in \{0, -1, 1\}^{20}$. For trajectory $j = 1$, $u_1^n = 0$ for $n = 1, \dots, 20$. For trajectory $j = 2$, $u_2^n = 0$ for $n = 1, \dots, 10$ and $u_2^n = -1$ for $n = 11, \dots, 20$. For trajectory $j = 3$, $u_3^n = 0$ for $n = 1, \dots, 10$ and $u_3^n = 1$ for $n = 11, \dots, 20$.

The TCL has been simulated using (3) with a default setpoint T_{set} of 2.5°C, a dead-band width δ of 2°C, an initial temperature T^1 of 3.3°C, and an initial mechanical state m^1 of 0. Figures 1, 2, and 3 present the T_j and p_j trajectories corresponding to each input u_j for $j = 1, 2, 3$. The mechanical state trajectories m_j can be inferred from the T_j and p_j trajectories. As illustrated by the figures, each distinct input u_j produces a distinct T_j, m_j, and p_j. Therefore, in this example, $N_d = N_a = 3$.

In summary, we have produced a representation of the system's dynamics under multiple alternative control trajectories. This representation can be incorporated into a convex program, as described in the next section. To the authors' knowledge, this is the first paper to introduce such an approach. While we have developed the method with the intention of enforcing non-linear system constraints in TCLs (such as minimum compressor on/off dwell times), we have found that the approach is well suited for the aggregated control of energy systems in general. By abstracting the system inputs, dynamics, and constraints into the $\mathbf{U}$ and $\mathbf{P}$ matrices, we can also model the aggregated optimization of heterogeneous energy systems such as residential solar panels, battery storage, and electrified vehicles.

2.3. Convex Optimization

In this section, we detail how the alternative control trajectory representation described above can be introduced into a convex program. To begin, we will introduce a variable $w \in \{0,1\}^{N_d}$ such that

$$w_j = \begin{cases} 1 & \text{if trajectory } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$
$$\forall\ j = 1, \ldots, N_d$$

Thus, if $j = 1$ is the selected trajectory (i.e. $w_1 = 1$)

$$\begin{aligned} \mathbf{U}^T w &= u_1 \\ \mathbf{T}^T w &= T_1 \\ \mathbf{M}^T w &= m_1 \\ \mathbf{P}^T w &= p_1 \end{aligned}$$

The integer program below demonstrates how $\mathbf{P}$, $\mathbf{T}$, and w can be introduced to solve for the optimal trajectory

$$\begin{aligned} \underset{w}{\text{minimize}}\ & F(\mathbf{P}^T w) + G(\mathbf{T}^T w) \\ \text{subject to}\ & \textstyle\sum w_j = 1 \\ & w \in \{0,1\}^{N_d} \end{aligned} \qquad (8)$$

where $F : \mathbf{R}^{N_t} \to (-\infty, \infty]$ and $G : \mathbf{R}^{N_t} \to (-\infty, \infty]$ are closed convex functions. Function F represents the utility of a power demand trajectory. This could be a cost function for electricity, a penalty function for deviating from a predefined profile, or a regularization function that flattens the power demand. Function G represents the utility of a temperature trajectory. For heating and air conditioning systems, G could represent the thermal comfort/discomfort of occupants. For TCLs like refrigerators or water heaters, G could quantify the willingness of a customer to allow deviations from the setpoint.

The above program is an example of the generalized assignment problem (GAP). If feasible, the integer program (8) guarantees that only one component of minimizer w^* is non-zero. However, because the integer program (8) is combinatorial and potentially intractable for large scale problems, it is unsuitable for many applications. In particular, distributed convex optimization methods require linearity or convexity in the agents [52]. By relaxing the binary constraint such that $\hat{w} \in \mathbf{R}^{N_d}$, we can express the convex program as

$$\begin{aligned} \underset{\hat{w}}{\text{minimize}}\ & F(\mathbf{P}^T \hat{w}) + G(\mathbf{T}^T \hat{w}) \\ \text{subject to}\ & \textstyle\sum \hat{w}_j = 1 \\ & \hat{w} \geq 0 \\ & \hat{w} \in \mathbf{R}^{N_d} \end{aligned} \qquad (9)$$

Due to the linear constraints, minimizer $\hat{w}_j^* \in [0,1]$ for $j = 1, \ldots, N_d$ and in practice, can be interpreted as the probability of selecting control trajectory j. In other words, we allow the convex program to form linear combinations of the alternative control trajectories. Once the program has converged to an optimal solution, we implement a single trajectory based on the discrete probability distribution $\hat{w}^*$. Expressed mathematically, we can generate a discrete random variable $X \in \{1, \ldots, N_d\}$ such that $\hat{w}_j^* = \Pr(X = j)$ for $j = 1, \ldots, N_d$. The value of X represents the index of the probabilistically selected control trajectory. Thus, we can define a variable $\tilde{w} \in \{0,1\}^{N_d}$, representing the probabilistic solution of (9), as

$$\tilde{w}_j = \begin{cases} 1 & \text{if } X = j \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$
$$\forall\ j = 1, \ldots, N_d$$

To reiterate, the optimal solution to (8) is physically realizable (i.e. only one component of w^* is non-zero) but not solvable using convex optimization. By contrast, (9) is convex but the optimal solution is not realizable (i.e. all components of $\hat{w}^*$ may be non-zero). Using (10), we can transform

$\hat{w}^*$ into $\tilde{w}$, which is realizable (i.e. only one component of $\tilde{w}$ is non-zero). Additionally, w^* and $\hat{w}^*$ are guaranteed to be optimal solutions to (8) and (9), respectively. However, $\tilde{w}$ may be an optimal or sub-optimal solution to both (8) and (9).

It should be noted that w^* is only optimal with respect to the N_d alternative control trajectories as represented by **U**, **T**, **M**, and **P**. If **U**, **T**, **M**, and **P** define the set of all feasible trajectories which satisfy the constraints of the system, then w^* is globally optimal. Otherwise, if **U**, **T**, **M**, and **P** define a subset of the feasible trajectories, then there is no guarantee of global optimality.

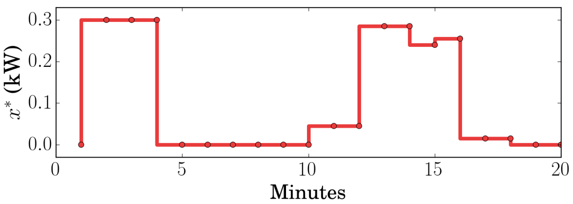

Figure 4: Example Solution to Convex Program

By way of example, we again refer to the alternative trajectories illustrated by Figures 1, 2, and 3. If we assemble the trajectories into the **T** and **P** matrices and solve (8), we might produce the solution $w^* = (1, 0, 0)$. In other words, the program selects trajectory $j = 1$. If we solve (9), we might produce the solution $\hat{w}^* = (0.8, 0.15, 0.05)$. In this case, the program selects a linear combination of the 3 trajectories. The resulting power demand trajectory $x = \mathbf{P}^T\hat{w}^*$ is illustrated in Figure 4. Finally, if we apply (10), there are 3 possible outcomes for $\tilde{w}$,

$$\Pr(\tilde{w} = (1,0,0)) = 80\%$$
$$\Pr(\tilde{w} = (0,1,0)) = 15\%$$
$$\Pr(\tilde{w} = (0,0,1)) = 5\%$$

Throughout this paper, we refer to the optimal power demand profile ($p = \mathbf{P}^T w$) produced by (8) as the *discrete* solution ($w^* \in \{0,1\}^{N_d}$), by (9) as the *continuous* solution ($\hat{w}^* \in \mathbf{R}^{N_d}$), and by (9) and (10) as the *probabilistic* solution ($\tilde{w} \in \{0,1\}^{N_d}$).

3. Alternating Direction Method of Multipliers

In this section, we briefly cover the *alternating direction method of multipliers* (ADMM) algorithm for convex optimization. We refer the reader to [52][54] for a more complete description of the algorithm. Next, we discuss a special case of block separable problems referred to as *sharing* ADMM [52]. We derive a formulation of the sharing ADMM algorithm suitable for the distributed optimization of TCLs and present primal and dual residual equations and stopping criteria not found in [52].

3.1. ADMM

The *alternating direction method of multipliers* is a common splitting method for solving problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array} \tag{11}$$

with variables $x \in \mathbf{R}^{N_x}$ and $z \in \mathbf{R}^{N_z}$, where $f : \mathbf{R}^{N_x} \to (-\infty, \infty]$ and $g : \mathbf{R}^{N_z} \to (-\infty, \infty]$ are closed convex functions, $A \in \mathbf{R}^{N_c \times N_x}$ and $B \in \mathbf{R}^{N_c \times N_z}$ are linear operators, and $c \in \mathbf{R}^{N_c}$ is a vector. ADMM is a variant of the augmented Lagrangian approach which uses partial updates of the dual variables at each iteration. The algorithm optimizes the coupled problem (11) by solving the uncoupled unscaled steps

$$x^{k+1} = \underset{x}{\text{argmin}}\ f(x) + \langle \lambda^k, Ax \rangle + \frac{\rho}{2}\|Ax + Bz^k - c\|_2^2 \tag{12a}$$

$$z^{k+1} = \underset{z}{\text{argmin}}\ g(z) + \langle \lambda^k, Bz \rangle + \frac{\rho}{2}\|Ax^{k+1} + Bz - c\|_2^2 \tag{12b}$$

$$\lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{12c}$$

where variable $\lambda \in \mathbf{R}^{N_c}$ is the dual variable, constant $\rho > 0$ is the augmented Lagrangian parameter, also referred to as the penalty parameter, and k is the integer valued iteration of the ADMM algorithm.

The necessary and sufficient optimality conditions for the ADMM problem (12) are given by the primal feasibility,

$$Ax^* + Bz^* - c = 0 \tag{13}$$

and dual feasibility,

$$0 = \nabla f(x^*) + A^T\lambda^* \tag{14}$$

$$0 = \nabla g(z^*) + B^T\lambda^* \tag{15}$$

assuming f and g are differentiable.

The convergence of (12) can be summarized by

- Objective Convergence: $f(x^k)+g(z^k) \to J^*$ as $k \to \infty$ where J^* denotes the optimal value of (11)

- Primal Residual Convergence: Residual $r^k \to 0$ as $k \to \infty$ where $r^k = Ax^k + Bz^k - c$

- Dual Variable Convergence: Variable $\lambda^k \to \lambda^*$ as $k \to \infty$

We refer the reader to [52][54] for a discussion of the augmented Lagrangian, scaled form, primal and dual residuals, and convergence rates.

3.2. Sharing ADMM

In this paper, we consider an ADMM-based method for solving the generic *sharing* problem using distributed optimization, as presented in [52]. In this section, we demonstrate how the *sharing* problem can be represented as a special case of (11) where f and A have a separable structure that we can exploit. The method is well suited for solving problems of the form

$$\text{minimize} \quad \sum f_i(x_i) + g(\sum x_i) \tag{16}$$

with variables $x_i \in \mathbf{F}_i^{N_x}$, the decision variable of agent i for $i = 1, \ldots, N$, where $\mathbf{F}_i$ represents the convex constraint set of agent i, N the number of agents in the network, N_x is the length of x_i, f_i is the cost function for agent i, and g is the shared objective function of the network. The function g takes as input the sum of the individual agent's decision variables, x_i. The sharing problem allows each agent in the network to minimize its individual/private cost $f_i(x_i)$ as well as the shared objective $g(\sum x_i)$.

By introducing variable $z_i \in \mathbf{R}^{N_x}$, a term that copies the x_i decision variable of each agent, the sharing problem can be written in an ADMM-compatible form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum f_i(x_i) + g(\sum z_i) \\ \text{subject to} \quad & x_i - z_i = 0, \; i = 1, \ldots, N \end{aligned} \tag{17}$$

with variables $x_i \in \mathbf{F}_i^{N_x}$, $z_i \in \mathbf{R}^{N_x}$, $\sum z_i \in \mathbf{G}^{N_x}$ for $i = 1, \ldots, N$ where $\mathbf{G}^{N_x}$ represents the convex constraint set of the shared objective. Therefore, the unscaled form of sharing ADMM is

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\text{argmin}} \; f_i(x_i) && \text{(18a)} \\ &\quad + \langle \lambda_i^k, x_i \rangle + \frac{\rho}{2}\|x_i - z_i^k\|_2^2 \\ z^{k+1} &= \underset{z}{\text{argmin}} \; g(\sum z_i) && \text{(18b)} \\ &\quad + \sum (\langle \lambda_i^k, -z_i \rangle + \frac{\rho}{2}\|x_i^{k+1} - z_i\|_2^2) \\ \lambda_i^{k+1} &= \lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1}) && \text{(18c)} \end{aligned}$$

with variable $z = (z_1, \ldots, z_N)$ and augmented Lagrangian parameter $\rho > 0$. Unlike (12), where there is a single globally defined dual variable λ, in (18), each agent has its own λ_i. Thus, the x_i-update and λ_i-update steps can be executed by each agent $i = 1, \ldots, N$ independently and in parallel. The z-update step is executed by a *collector* or *aggregator* with knowledge of each agent's decision variable x_i.

3.3. Sharing ADMM Residuals

Next, we define the sharing ADMM residuals. The necessary and sufficient optimality conditions for the sharing ADMM algorithm and derivation of the residuals are presented in the Appendix. The primal residual is defined as

$$r_i^{k+1} = x_i^{k+1} - z_i^{k+1} \tag{19}$$

and the dual residual as

$$s_i^{k+1} = -\rho(z_i^{k+1} - z_i^k) \tag{20}$$

3.4. Stopping Criteria

We define the stopping criteria as presented in [52] by

$$\|r^k\|_2 \leq \epsilon^{primal} \text{ and } \|s^k\|_2 \leq \epsilon^{dual} \tag{21}$$

where $r^k = (r_1^k, \ldots, r_N^k)$, $s^k = (s_1^k, \ldots, s_N^k)$, and $\epsilon^{primal} > 0$ and $\epsilon^{dual} > 0$ are feasibility tolerances for the primal and dual conditions (44) and (45). In this paper, we set $\epsilon^{primal} = \epsilon^{dual} = 1$.

3.5. Averaged Sharing ADMM

As written, the sharing ADMM algorithm (18) requires the local calculation of a z_i^k, λ_i^k, and r_i^k term for each agent $i = 1, \ldots, N$ in the network. Next, we will simplify the algorithm by introducing global variables $\bar{x}^k$, $\bar{z}^k$, and $\bar{\lambda}^k$ representing the arithmetic mean of all x_i^k, z_i^k, and λ_i^k, respectively. The unscaled form of the averaged sharing ADMM algorithm is given below. The derivation of the averaged sharing ADMM algorithm is presented in the Appendix.

$$x_i^{k+1} = \underset{x_i}{\operatorname{argmin}}\ f_i(x_i) + \langle \bar{\lambda}^k, x_i \rangle + \frac{\rho}{2}\|x_i - x_i^k + \bar{x}^k - \bar{z}^k\|_2^2 \tag{22a}$$

$$\bar{z}^{k+1} = \underset{\bar{z}}{\operatorname{argmin}}\ g(N\bar{z}) + \langle \bar{\lambda}^k, -N\bar{z} \rangle + \frac{N\rho}{2}\|\bar{x}^{k+1} - \bar{z}\|_2^2 \tag{22b}$$

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + \rho(\bar{x}^{k+1} - \bar{z}^{k+1}) \tag{22c}$$

With this averaged sharing ADMM form, the individual agents no longer update their own λ_i variable. Instead, a single aggregator updates $\bar{\lambda}$, along with $\bar{x}$ and $\bar{z}$, and reports these global variables to every agent in the network. We refer the reader to [52][54] for a further discussion of the averaged sharing ADMM algorithm and convergence characteristics.

3.6. Averaged Sharing ADMM Residuals

In order to apply the stopping criteria (21), we must redefine the primal and dual residuals for the averaged form. The derivation of the averaged residuals is presented in the Appendix. The averaged primal residual is defined as

$$r_i^{k+1} = \bar{x}^{k+1} - \bar{z}^{k+1} \tag{23}$$

and the averaged dual residual as

$$s_i^{k+1} = \rho((\bar{x}^{k+1} - \bar{x}^k) - (x_i^{k+1} - x_i^k) - (\bar{z}^{k+1} - \bar{z}^k)) \tag{24}$$

The corresponding ℓ_2-norms of the stopping criteria are therefore

$$\begin{aligned} \|r^k\|_2 &= N\|\bar{x}^k - \bar{z}^k\|_2 \\ \|s^k\|_2 &= \sum \|s_i^k\|_2 \end{aligned} \tag{25}$$

4. Distributed TCL Optimization For Generation Following

In this section, we describe the application of the sharing ADMM algorithm to the distributed optimization of TCLs with the objective of providing 5-minute ahead generation following ancillary services. Specifically, we define the optimization program for the individual TCLs and the aggregator. Then, we describe the sharing ADMM algorithm for the TCL population. Finally, we detail the infrastructure required for communication, computation, and control as well as the execution of the ADMM-based control method. Results from multiple studies are described in the next section. Our formulation is based on the following assumptions:

- Each TCL is capable of (i) manipulating its setpoint by a discrete/integer amount, (ii) accurately monitoring and forecasting its power demand, (iii) solving convex programs, and (iv) communicating with a central aggregator (representing a load-serving entity such as an electric utility).
- The consumer is indifferent to the relative energy costs of the alternative control trajectories. In other words, either the consumer does not pay for energy used by the TCL or the compensation for participating in the demand response program is such that the change in energy cost is negligible. This does not imply that each alternative trajectory is of equal utility.
- At each ADMM iteration and time step, a TCL's decision variable and selected power demand trajectory is shared with only the aggregator. The TCL's characteristics and decision making, including the $\mathbf{P}$ matrix, remain private to that TCL.

4.1. TCL Optimization

In this paper, we consider four types of thermostatically controlled loads: refrigerators, electric water heaters, heat pumps, and electric baseboard heaters. Each TCL is simulated using model (3) with published parameter ranges, given in Table 1 and adopted from [14]. To generate a population, parameters are randomly drawn from a uniform distribution between the maximum and minimum values shown in the table. For heat pumps and baseboard heaters, the C parameter is multiplied by the number of zones, an integer randomly drawn from the range given. Additionally, for the ambient temperature T_∞^n of the heat pumps and baseboard heaters, we utilize weather data for Berkeley, California from the morning of 3/19/2015, shown in

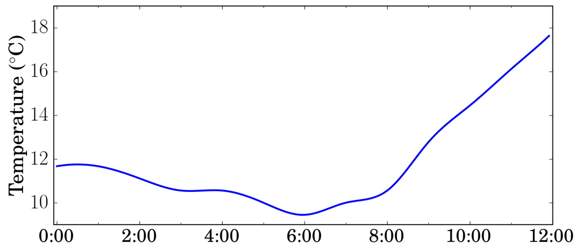

Figure 5: Ambient Temperature Data for Berkeley, CA, on the Morning of 3/19/2015

Figure 5 [55]. The electric power demand of the TCL at each time step is given by (2).

TCL control takes the form of setpoint manipulation. Rather than considering the full set of feasible control inputs, we only consider a small subset of the feasible set. Specifically, we define $N_a = 3$ control inputs for each TCL in a population. The first control input applies no change to the temperature setpoint and corresponds to the default or normal operation of the TCL. The second input applies a setpoint change that will cause the system to turn on or stay on and is therefore expected to increase the average power demand of the TCL relative to normal operation. Conversely, the third input applies a setpoint change that will cause the system to turn off or stay off and is expected to decrease the average power demand of the TCL relative to normal operation.

To generate these control inputs, we define a discrete set of feasible/allowed setpoint changes, represented by S_u. Though we simulate the TCLs using a one minute time scale ($h = 1/60$ hours), we apply all setpoint changes over 5 consecutive time steps ($N_t = 5$). Thus, for a refrigerator with $S_u = \{0, -2, 1\}$,

$$\begin{aligned} u_1 &= (0, 0, 0, 0, 0) \\ u_2 &= (-2, -2, -2, -2, -2) \\ u_3 &= (1, 1, 1, 1, 1) \end{aligned}$$

Therefore, the refrigerator has a maximum of $N_a = 3$ alternative control trajectories. As stated previously, each distinct input u_j is not guaranteed to produce a distinct output T_j, m_j, or p_j. For a given TCL, the number of *distinct* alternative control trajectories, N_d, is in the discrete set $\{1, \ldots, N_a\}$.

The zero input u_1 represents the default TCL input and is always first in the set of alternative control trajectories. If $N_d = 1$, we describe the TCL as fixed or inflexible. In other words, the TCL is at a point in its cycle such that setpoint manipulation does not impact the temperature trajectory. If $N_d = 2$ and the mean of p_2 is greater than the mean of p_1, then the TCL is only capable of increasing demand; if $N_d = 2$ and the mean of p_2 is less than or equal to the mean of p_1, then the TCL is only capable of decreasing demand. If $N_d = 3$, then the TCL is flexible and capable of increasing or decreasing demand. This classification is used to interpret results in Section 5.

Using the alternative control trajectory representation, we can simulate a TCL using $\mathbf{U}$ and (3) to output $\mathbf{T}$, $\mathbf{M}$, and $\mathbf{P}$ matrices such that $\mathbf{U}$, $\mathbf{T}$, $\mathbf{M}$, and $\mathbf{P} \in \mathbf{R}^{N_d \times N_t}$. Now, the individual TCL's optimization problem can be defined as a constrained least-squares fit.

$$\begin{aligned} \underset{\hat{w}}{\text{minimize}} \; & \alpha_x \|\mathbf{T}^T \hat{w} - T_{set}\|_2^2 \\ \text{subject to} \; & \textstyle\sum \hat{w}_j = 1 \\ & \hat{w} \geq 0 \end{aligned} \tag{26}$$

with variables $\mathbf{T} \in \mathbf{R}^{N_d \times N_t}$, representing the set of distinct temperature trajectories, $\hat{w} \in \mathbf{R}^{N_d}$, representing the optimal linear combination of trajectories and/or the discrete probability distribution of selecting control trajectory j for $j = 1, \ldots, N_d$, $T_{set} \in \mathbf{R}^{N_t}$ the TCL's temperature setpoint, N_t the number of time steps simulated, N_d the number of control trajectories, and α_x a weighting term for the TCL's objective. As previously described, the *continuous* solution for the power demand profile is determined by $x^* = \mathbf{P}^T \hat{w}^*$. Given $\hat{w}^*$ and (10), we denote the *probabilistic* solution as $\tilde{p} = \mathbf{P}^T \tilde{w}$. Because $\tilde{w} \in \{0, 1\}^{N_d}$, $\tilde{p}$ is in the feasible set of power trajectories defined by $\mathbf{P}$. As previously stated, $\hat{w}^*$ and x^* are guaranteed to be optimal, but $\tilde{w}$ and $\tilde{p}$ may be sub-optimal.

It should be noted that the TCLs could be simulated and controlled with time steps of less than one minute without impacting the computational requirements of the distributed optimization algorithm. For example, we could simulate a TCL with a time scale of one second. To produce the alternative temperature and power trajectories required for the optimization, we would use the minute-wise averages of the simulated temperature and power demand of the TCL. In this way, the time scale used for optimization is uniform over the population while the time scale used for simulation and control is determined by the individual TCLs.

Parameter	Refrigerator	Water Heater	Heat Pump	Baseboard Heater
Thermal resistance, R (°C/kW)	[80, 100]	[100, 140]	[1.5, 2.5]	[1.5, 2.5]
Thermal capacitance, C (kWh/°C)	[0.4, 0.8]	[0.2, 0.6]	[0.15, 0.25]	[0.15, 0.25]
Energy transfer rate, P (kW)	[-1, -0.2]	[4, 5]	[14, 25.2]	[0.5, 1.5]
Coefficient of performance, COP	2	1	3.5	1
Temperature setpoint, T_{set} (°C)	[1.7, 3.3]	[43, 54]	[15, 24]	[15, 24]
Dead-band width, δ (°C)	[1, 2]	[2, 4]	[0.25, 1]	[0.25, 1]
Ambient temperature, T_∞ (°C)	20	20	variable	variable
Number of zones	1	1	[5,10]	[1,2]
Number of trajectories, N_a	3	3	3	3
Allowed setpoint changes, S_u (°C)	{0, -2, 1}	{0, 5, -5}	{0, 1, -2}	{0, 1, -2}

Table 1: TCL parameter ranges adopted from [14]

4.2. Aggregator Objective

In this paper, the aggregator, representing a load-serving entity, will influence the behavior of the TCLs so as to perform 5-minute power generation following. To demonstrate this potential, we consider 5 minute ahead forecasts of wind and solar generation retrieved from the California Independent System Operator (ISO) [56]. Figure 6 presents the wind and solar power generation for the morning of 3/19/2015. The center plot shows a smooth polynomial fit of the total renewable generation. The error between the actual generation and the smooth fit will serve as our exemplary 5-minute generation following signal in this paper, shown in the bottom plot.

Ideally, 5-minute generation following is a zero net energy service. Accordingly, the mean of the control signal is 1.229×10^{-7} MW. Considering that the signal is on the order of 10 MW and that TCLs are on the order of 1 kW loads, in this paper, we will utilize the TCLs to respond to 1% of the signal shown in Figure 6. Additionally, we are simulating the TCL's using a one minute time scale but the signal is on a five minute time scale. Thus, we will treat the signal as a piecewise constant function. It is possible to interpolate between the current and previous control signal to produce a smooth or piecewise linear signal. Nonetheless, we are electing to use a piecewise constant interpretation.

To perform generation following, the aggregator's objective function can be defined as an un-

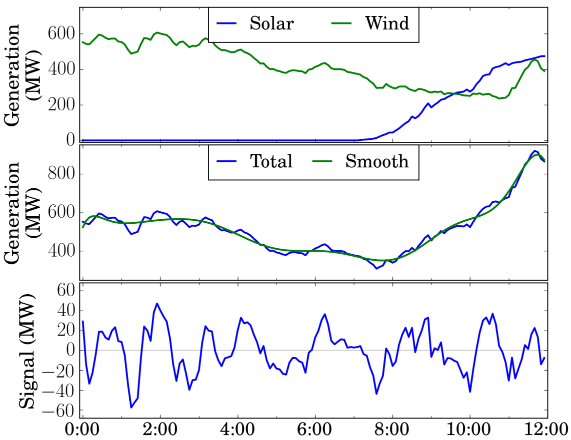

Figure 6: California ISO Wind and Solar Generation 5-Min Forecasts for 3/19/2015 (Top), Smooth Polynomial Fit of Total Generation (Center), and exemplary 5-minute Generation Following Signal (Bottom)

constrained least-squares fit.

$$\text{minimize } \alpha_z \|\textstyle\sum x_i - d\|_2^2 \tag{27}$$

with variables $d \in \mathbf{R}^{N_t}$, the aggregator's desired power demand given the generation following signal $y \in \mathbf{R}^{N_t}$, and $x_i \in \mathbf{R}^{N_t}$, the power demand of TCL i for $i = 1, \ldots, N$, where N represents the number of TCLs in the network and $N_t = 5$ is the number of time steps in d and x_i. Lastly, α_z is a weighting term for the aggregator's objective.

We calculate the desired power demand d by adding the current generation following signal y to the power demand of the population in the previous time step (i.e. $d^n = \sum_i \tilde{p}_i^{n-1} + y^n$ for $n = 1, \ldots, N_t$). Since the value of the signal only changes once every 5 minutes, we optimize the aggregated power demand over a horizon of $N_t = 5$ time steps and thus,

$$d^n = \begin{cases} \sum_i \tilde{p}_i^{n-1} + y^n & \text{if } n = 1 \\ d^{n-1} & \text{otherwise} \end{cases} \qquad (28)$$
$$\forall\ n = 1, \ldots, N_t$$

4.3. TCL Sharing ADMM

Given the TCL and aggregator optimization programs (26) and (27), we can now define the sharing ADMM algorithm for power generation following using a population of TCLs.

$$\begin{aligned} \hat{w}_i^{k+1} &= \underset{\hat{w}_i}{\text{argmin}}\ \alpha_{x,i} \|\mathbf{T}_i^T \hat{w}_i - T_{set,i}\|_2^2 && \text{(29a)} \\ &\quad + \langle \bar{\lambda}^k, \mathbf{P}_i^T \hat{w}_i \rangle + \frac{\rho}{2} \|\mathbf{P}_i^T \hat{w}_i - x_i^k + \bar{r}^k\|_2^2 \\ &\quad \text{s. to} \quad \sum \hat{w}_j = 1, \quad \hat{w} \geq 0 \\ x_i^{k+1} &= \mathbf{P}_i^T \hat{w}_i^{k+1} && \text{(29b)} \\ \bar{z}^{k+1} &= \underset{\bar{z}}{\text{argmin}}\ \alpha_z \|N\bar{z} - d\|_2^2 + \langle \bar{\lambda}^k, -N\bar{z} \rangle && \text{(29c)} \\ &\quad + \frac{N\rho}{2} \|\bar{x}^{k+1} - \bar{z}\|_2^2 \\ \bar{r}^{k+1} &= \bar{x}^{k+1} - \bar{z}^{k+1} && \text{(29d)} \\ \bar{\lambda}^{k+1} &= \bar{\lambda}^k + \rho(\bar{r}^{k+1}) && \text{(29e)} \end{aligned}$$

In our implementation, the ADMM algorithm is run once every 5 minutes to determine the optimal power demand of the TCL population over the next 5 minutes at a 1 minute time scale. For simplicity, we report the power demand of the TCLs as a 5 minute average. For fixed TCLs (i.e. $N_d = 1$), the power demand profile is reported to the aggregator before the first ADMM iteration. The N and d parameters are adjusted accordingly and the ADMM algorithm run on the remaining population.

4.4. Generation Following Algorithm, Distributed Network Structure, and Communication

To achieve distributed control of a TCL population, we assume a certain amount of existing infrastructure for communication, computation, and control. Our assumptions are comparable to those made in [33][36][48] and include:

- Bi-directional communication between the individual TCLs and the aggregator via wired or wireless links.
- Sufficient local computation and hardware for solving convex programs and measuring TCL states.
- A local TCL model whose parameters are either known a priori or identified using a parameter estimation technique [44][45][51].

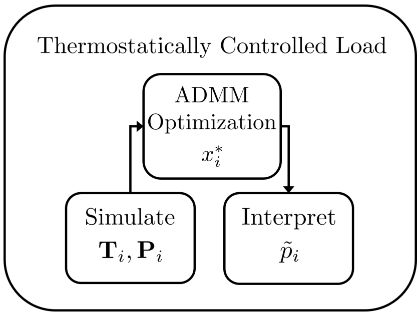

Figure 7: TCL Model and Optimization Structure. Each TCL i in the population will simulate its dynamics to produce the alternative control trajectories, coordinate with an aggregator using the ADMM algorithm to produce a continuous solution, and finally interpret the discrete probability distribution to produce a probabilistic solution.

In this manuscript, we assume a simple network structure with bi-directional communication between the aggregator and each TCL. The inputs required by our ADMM algorithm are presented in Table 2 and the outputs of the optimization in Table 3. The internal ADMM variables are listed in Table 4 and the parameters in Table 5.

The execution of the generation following algorithm can be summarized by the follow 4 steps:

1. Aggregator Preparation: Every 5 minutes, the aggregator receives the signal y and produces the desired power profile d.
2. TCL Simulation: Each TCL i in the population simulates its dynamics to produces a set of alternative temperature trajectories $\mathbf{T}_i$ and power trajectories $\mathbf{P}_i$.
3. Optimization via ADMM: For each iteration k until the stopping criteria are met:
   (a) Broadcast Signal: The aggregator reports the mean primal residual $\bar{r}^k$ (i.e. the difference between $\bar{x}^k$ and $\bar{z}^k$) and the mean dual incentive variable $\bar{\lambda}^k$ to each TCL i in the population.

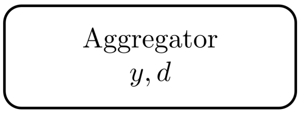

Figure 8: Aggregator Preparation Step. The aggregator receives the signal y and produces the desired power profile d.

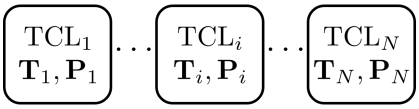

Figure 9: TCL Simulation Step. Each TCL i in the population simulates its dynamics to produce the alternative control trajectories.

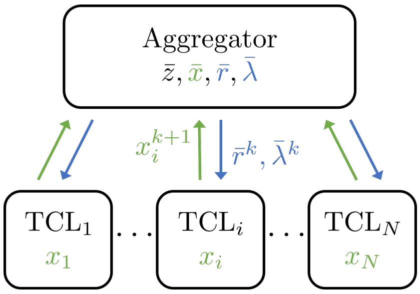

Figure 10: Optimization Step. For each iteration k of the ADMM algorithm, the aggregator reports the mean primal residual $\bar{r}^k$ and the mean dual incentive variable $\bar{\lambda}^k$ to each TCL i in the population. Each TCL i then reports its updated x_i^{k+1} to the aggregator.

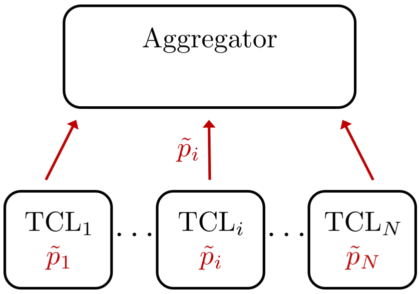

Figure 11: Interpretation Step. Each TCL i interprets the discrete probability distribution $\hat{w}_i^*$ to select a power trajectory $\tilde{p}_i$ from the set $\mathbf{P}_i$ and reports the probabilistic solution $\tilde{p}_i$ to the aggregator.

   (b) Local Optimization: Each TCL i optimizes (29a) and reports x_i^{k+1} (29b) to the aggregator.
   (c) Aggregator Optimization: Given the mean TCL power profile $\bar{x}^{k+1}$, the aggregator optimizes (29c) and updates the mean primal residual $\bar{r}^{k+1}$ (29d) and the mean dual incentive variable $\bar{\lambda}^{k+1}$ (29e).

4. Interpretation: Each TCL i interprets the discrete probability distribution $\hat{w}_i^*$ to select a power trajectory $\tilde{p}_i$ from the set $\mathbf{P}_i$ and reports the probabilistic solution $\tilde{p}_i$ to the aggregator.

Figure 7 outlines the steps performed by each TCL i in the population. At each time step, the TCL simulates its dynamics to produce the alternative control trajectories as represented by $\mathbf{T}_i$ and $\mathbf{P}_i$, coordinates with an aggregator via ADMM to produce a continuous solution x_i^*, and finally interprets the discrete probability distribution $\hat{w}_i^*$ to produce a probabilistic solution $\tilde{p}_i \in \mathbf{P}_i$.

The 4 steps of the generation following algorithm, as well as the structure of the distributed system, are illustrated in Figures 8, 9, 10, and 11. In particular, the figures indicate for each step of the algorithm which variables are defined locally and which are communicated between the aggregator and the TCLs in the population.

In our algorithm, each TCL i reports the power demand profile x_i^{k+1} to the aggregator but not to the other TCLs in the network. Each TCL's $\mathbf{T}$, $\mathbf{P}$, and $\hat{w}^k$ remain private. In addition to the stopping criteria (21), we impose a limit on the absolute value of $\bar{\lambda}$ (i.e. stop if $|\bar{\lambda}^n| \geq \lambda_+$ for $n = 1, \ldots, N_t$). This limit is empirically selected and serves as a means of detecting if the population of TCL's is able to match the signal within a certain tolerance. As defined by (27), any power demand is feasible, but in practice, we only want to perform generation following if the aggregate continuous solution $N\bar{x}^*$ is within a certain error tolerance, ϵ^{error}, of the control signal d (i.e. $\max(|N\bar{x}^* - d|) < \epsilon^{error}$). Therefore, if the ADMM algorithm does not converge to a solution within this tolerance, the population has failed to perform generation following and each TCL implements some default behavior. In this paper, the default behavior is to return to the original temperature setpoint by implementing the control trajectory $u_1 = (0, 0, 0, 0, 0)$.

At optimality, the power demand profile x_i^* represents the TCL's *continuous* solution and is not directly implementable. While it is conceptually possible to cluster complementary TCLs or to incorporate energy storage so as to directly achieve the continuous solution, we assume no such coordination in this paper. Instead, each TCL in the population will implement a single control trajectory given the discrete probability distribution $\hat{w}_i^*$. The TCLs' states are updated and the resulting power demand profile, referred to as the *probabilistic* solution $\tilde{p}_i$, is reported to the aggregator. The potential for error between the continuous and probabilistic solution is addressed in Section 4.6 below.

4.5. ADMM Parameter Selection

The parameters of the ADMM algorithm are presented in Table 5. These parameters have been empirically selected to fit the characteristics of our application. Specifically, because the behavior of each TCL in the population is constrained by its alternative control trajectories, the goal of the generation following algorithm is primarily to shape the aggregate load in response to a signal. This is expressed in the ADMM parameters by selecting an aggregator coefficient α_z that is larger than the TCL coefficient α_x for each TCL in the population. Decreasing α_z or increasing α_x will cause less emphasis to be placed on the global objective. Therefore, the TCLs will choose to optimize their local objectives rather than optimizing the aggregate power demand (a further discussion of this behavior is presented in Section 5.7).

The primal and dual feasibility tolerances are positive values which define the stopping criteria of the ADMM algorithm. In our application, the mean primal residual $\bar{r}^k$ is the difference between $\bar{x}^k$, the mean power demand based on the continuous solutions reported by the TCL population, and $\bar{z}^k$, the mean power demand based on the solution to the aggregator's objective function. The primal feasibility tolerance ϵ^{primal} is a measure of the primal residual that we are willing to accept. Based on the relative weighting of the aggregator and TCL objectives and the error tolerance $\epsilon^{error} = 10\text{kW}$, ϵ^{primal} can effectively be any positive value less than $\sqrt{10}$. We have selected $\epsilon^{primal} = 1$ based on empirical observations that the value produces aggregate continuous solutions within the error tolerance ϵ^{error} within a modest number of ADMM iterations (i.e. <50).

The dual residual s_i^k of each TCL i is a measure of the change in the continuous solution x_i^k and in the primal residual $\bar{r}^k$ between ADMM iteration k and $k+1$. Thus, $\|s^k\|$ is a measure of the rate of change in the solutions of the aggregator and TCL population. A large dual feasibility tolerance ϵ^{dual} will cause the ADMM algorithm to stop once the primal feasibility criterion is met while a small tolerance will cause the algorithm to continue until the solutions of the aggregator and TCLs no longer change from one iteration to the next. We have empirically chosen a dual feasibility tolerance ϵ^{dual} of 1 such that the dual feasibility criterion is met a few iterations (i.e. <10) after the primal feasibility criterion.

The Lagrangian penalty has been tuned to be sufficiently large such that the ADMM algorithm converges relatively quickly but sufficiently small so as to avoid oscillatory behaviors in the ADMM updates as the algorithm begins to converge. Lastly, we have observed that when the desired power profile d is outside the feasible power demand range of the TCL aggregation, the absolute values of $\bar{\lambda}$ increase dramatically as the ADMM algorithm attempts to drive the TCL population toward an infeasible solution so as to reduce the aggregator's objective function. To detect this behavior and stop the ADMM algorithm, we impose a limit of $\lambda_+ = 50$ on the absolute value of $\bar{\lambda}$.

4.6. Divide and Conquer

At optimality, the solution x_i^* represents the *continuous* solution of the relaxed form of the general assignment problem, as described in (9). While this relaxation is essential for distributed convex optimization, the continuous solution is not directly implementable. Instead, we employ the *probabilistic* solution $\tilde{p}_i$ and thereby introduce the potential for error between the solution returned by the ADMM algorithm and the actual power demand of the TCLs. For highly homogeneous populations of TCLs, we have observed that the aggregated continuous and probabilistic solutions are comparable (i.e. have similar errors with respect to the signal). The logical explanation is that due to the homogeneity, many TCLs converge to similar solutions. Thus, their probabilistic solutions are complementary such that the aggregated power demand is close to the continuous solution returned by the ADMM algorithm. For highly heterogeneous populations, however, this is not the case.

To address this, we investigated the introduction of a sparsity-inducing weighted ℓ_1 norm [57] into the TCL's objective function to drive the probabilities towards 0% or 100% (Due to the non-negativity constraint in (9), tradition ℓ_1-regularization is ineffective). However, we found that sparsity came at the cost of slower convergence and higher errors between the continuous solution and the signal.

Our solution is a relatively brute force, divide and conquer approach. Stated simply, we run ADMM on the entire population of TCLs. Upon convergence, we fix a certain number of the TCLs (10-

Generation following signal	y
Desired power demand profile	d
Temperature setpoint of TCL i	$T_{set,i}$
Temperature trajectories of TCL i	$\mathbf{T}_i$
Power trajectories of TCL i	$\mathbf{P}_i$

Table 2: Optimization Inputs

Final power demand profile of TCL i (probabilistic solution)	$\tilde{p}_i$

Table 3: Optimization Outputs

20% of the total population) using the probabilistic solution. These TCLs are them removed from the population being optimized and the N and d parameters are adjusted accordingly. Next, we repeat the ADMM algorithm to find the continuous solution of the remaining population using the previous value of λ and adjusted values of $\bar{x}$ and $\bar{z}$ as a warm start. This process is repeated until all TCLs are fixed. For successive ADMM runs, we decrease the number of ADMM iterations as the problem becomes more constrained. Numerical examples are provided next.

5. Experimental Results

In this section, we present results for 4 experimental studies. In each experiment, we model a population of TCLs to follow 1% of the signal described in Figure 6. This 5-minute generation following is achieved by running the sharing ADMM algorithm every 5 minutes between midnight and noon for the morning of 3/19/2015. In the first experiment, we consider a large, highly homogeneous population of refrigerators. Second, a small, heterogeneous population of refrigerators. Third, a highly heterogeneous population of refrigerators, water heaters, heat pumps, and baseboard heaters. Fourth, a highly heterogeneous population of refrigerators, water heaters, heat pumps, and baseboard heaters using the divide and conquer approach described above.

For each study, we employ the ADMM parameters in Table 5. For refrigerators and water heaters, $\alpha_x = 0$ indicating that the consumer is indifferent to the selection of a control trajectory. Thus,

Probability distribution of TCL i at iteration k	$\hat{w}_i^k$
Power demand profile of TCL i at iteration k (continuous solution)	x_i^k
Mean TCL power demand profile at iteration k (continuous solution)	$\bar{x}^k$
Mean aggregator power demand profile at iteration k (continuous solution)	$\bar{z}^k$
Mean primal residual	$\bar{r}^k$
Mean dual variable	$\bar{\lambda}^k$

Table 4: ADMM Variables

Lagrangian Penalty	ρ	10
Aggregator Coefficient	α_z	20
TCL Coefficient (Refrigerator)	α_x	0
TCL Coefficient (Water Heater)	α_x	0
TCL Coefficient (Heat Pump)	α_x	1
TCL Coefficient (Baseboard Heater)	α_x	1
Primal Feasibility Tolerance	ϵ^{primal}	1
Dual Feasibility Tolerance	ϵ^{dual}	1
Error Tolerance	ϵ^{error}	10 kW
$\bar{\lambda}$ Limit	λ_+	50

Table 5: ADMM Parameters

the TCL's objective function (26) is constant and weakly convex. At each iteration, the TCL enforces feasibility and adjusts its power demand according to the incentive signal λ. For heat pumps and baseboard heaters, $\alpha_x = 1$ indicating that the consumer would prefer to keep the temperature near the setpoint. The weight α_x is not large enough to prevent the selection of any alternative control trajectory, but rather numerically incentives the utilization of more cooperative/responsive refrigerators and water heaters before heat pumps and baseboard heaters. Lastly, S_u defines a set of 3 allowed change in setpoint values. Thus, each TCL has a maximum of $N_a = 3$ alternative control trajectories.

For each of the experimental studies, we present the aggregated power demand and response of the population for the respective experiment. The aggregated continuous and probabilistic power *demand* are presented as the mean of the total power demand over each $N_t = 5$ minute interval.

$$x^k_\Sigma = \frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N}(x^n_i)^* \tag{30}$$

$$p^k_\Sigma = \frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N}\tilde{p}^n_i \tag{31}$$

where variables $x^k_\Sigma, p^k_\Sigma \in \mathbf{R}$, N is the number of TCLs in the population, and k denotes the integer valued time step of each ADMM run (i.e. each $N_t =$ 5 minute interval between midnight and noon).

The continuous and probabilistic *responses* of the population denote the change in power demand, and are respectively given by

$$x^k_\Delta = x^k_\Sigma - p^{k-1}_\Sigma \tag{32}$$

$$p^k_\Delta = p^k_\Sigma - p^{k-1}_\Sigma \tag{33}$$

Because x^*_i is not directly realizable, x^k_Δ is calculated relative to the previous probabilistic demand p^{k-1}_Σ.

For each time step k, we also present the minimum and maximum power demand that the population of TCLs *could* have achieved given the set of power trajectories $\mathbf{P}_i$ for each TCL. For each TCL i, we denote the trajectories with the minimum and maximum mean power demand as $p^{\min}_i \in \mathbf{P}_i$ and $p^{\max}_i \in \mathbf{P}_i$, respectively. Therefore, the minimum and maximum mean power demand of the population is

$$p^k_{\min\Sigma} = \frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N}(p^n_i)^{\min} \tag{34}$$

$$p^k_{\max\Sigma} = \frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N}(p^n_i)^{\max} \tag{35}$$

Thus, the maximum up or down response of the population is given by

$$p^k_{\min\Delta} = p^k_{\min\Sigma} - p^{k-1}_\Sigma \tag{36}$$

$$p^k_{\max\Delta} = p^k_{\max\Sigma} - p^{k-1}_\Sigma \tag{37}$$

where variable $p^k_{\min\Delta}$ corresponds to demand decrease and $p^k_{\max\Delta}$ to demand increase (from the perspective of the load). In the case that $p^k_{\min\Delta} > 0$ or $p^k_{\max\Delta} < 0$, the population is incapable of decreasing or increasing its power demand, respectively.

5.1. Highly Homogeneous Population

To begin, we present the results using a highly homogeneous population of refrigerators. Specifically, we have modeled and controlled a population of $N = 20{,}000$ refrigerators with identical parameters (the mean of the parameter ranges in Table 1). We have limited the number of ADMM iterations to 10.

Figure 12 presents the results from the homogeneous experiment. The top plot shows how well the continuous responses x^k_Δ and the probabilistic responses p^k_Δ compare to the signal y^k for each 5 minute interval between midnight and noon. To reiterate, the continuous response is the difference between the aggregated solution to the ADMM algorithm and the power demand in the previous time step. The probabilistic response is the difference between the aggregated probabilistically selected TCL trajectories and the power demand in the previous time step. The RMSEs of the continuous and probabilistic responses are 0.11 kW and 14.25 kW, respectively. The ADMM algorithm only failed to converge to a continuous solution within the error tolerance of 10 kW during two intervals at 10:00 and 10:05 AM, resulting in a generation following success rate of 98.6% over the time period studied.

The second plot in Figure 12 shows the probabilistic p^k_Σ, the minimum $p^k_{\min\Sigma}$, and the maximum $p^k_{\max\Sigma}$ power demand of the population at each time interval. The third plot shows the corresponding minimum $p^k_{\min\Delta}$ and maximum $p^k_{\max\Delta}$

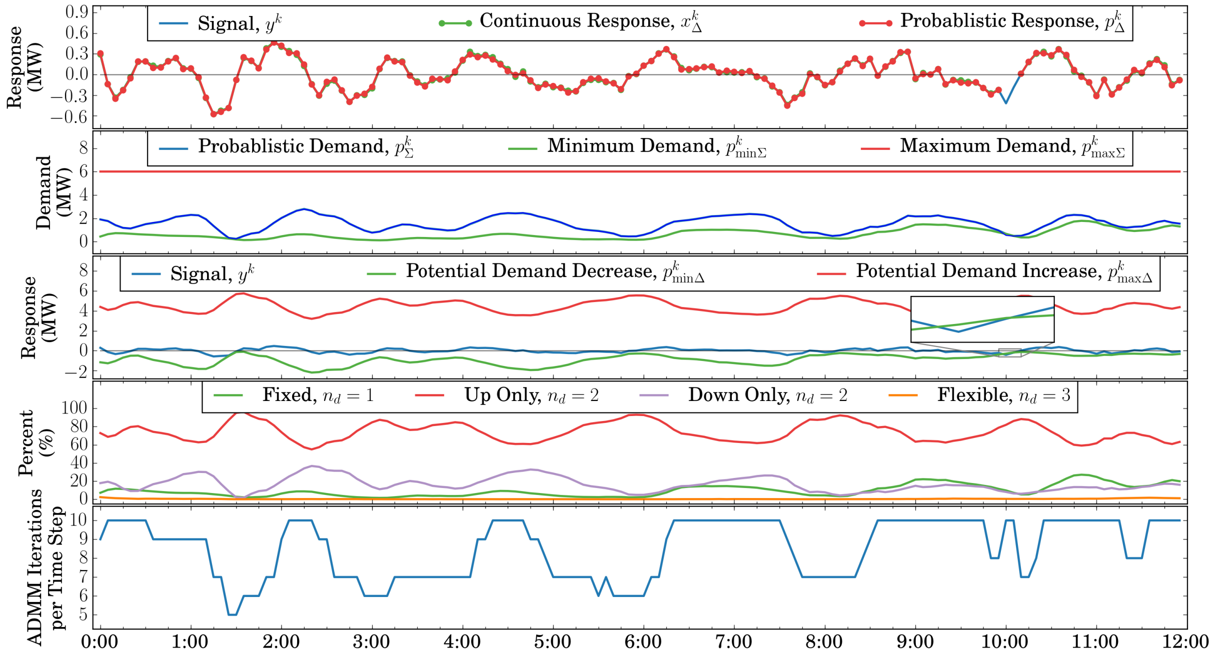

Figure 12: Highly Homogeneous Population

potential (i.e. the difference between the minimum or maximum power demand and the demand in the previous time step). While it is possible for the aggregator to discern these minimum and maximum values by manipulating $\bar{\lambda}$ to drive the TCLs to their extremes, we have assumed no such behavior in our implementation. Thus, the aggregator can only determine if the signal and the feasible up or down responses are within the specified error tolerance after the ADMM algorithm converges. The only exception is if $\bar{\lambda}$ violates the λ_+ limit, indicating that the ADMM algorithm is attempting to drive the population toward an infeasible solution so as to reduce the aggregator's objective function (though the TCLs will guarantee that the solution at each iteration is feasible).

The fourth plot shows the percentage of the population that is either fixed, flexible, or capable of only up or down responses. From midnight to 6:00, we observe that the TCLs move between up only and down only conditions, with the percent of fixed and flexible TCLs remaining small. After 6:00, the TCLs in the up only population begin to move to the down only or fixed populations. In the fifth plot, which shows the number of ADMM iterations executed before stopping, we see that the ADMM algorithm has more difficulty finding a solution in these later time intervals and begins hitting the iterations limit of 10. This trend represents a decline in the capability of the population to perform generation following.

5.2. Homogeneous Population with Dwell Time

In this study, we demonstrate the suitability of the control framework to honor minimum dwell time constraints. We consider a homogeneous population of $N = 20{,}000$ refrigerators with identical parameters (the mean of the parameter ranges in Table 1). The TCLs are controlled such that a minimum dwell time of 5 minutes is enforced (i.e. if a TCL turns on or off, it must remain in the new state for at least 5 minutes). Again, we have limited the number of ADMM iterations to 10.

The minimum dwell time constraint is applied at the Simulate TCLs step of the generation following algorithm. Specifically, if a TCL simulation produces a mechanical state trajectory m_j such that the minimum dwell time of 5 minutes would be violated if the trajectory was implemented, the trajectory is discarded by excluding the corresponding u_j, T_j, m_j, and p_j from $\mathbf{U}$, $\mathbf{T}$, $\mathbf{M}$, and $\mathbf{P}$.

The results, presented in Figure 13, show a generation following success rate of 100.0% over the time period studied. The RMSEs of the continuous and

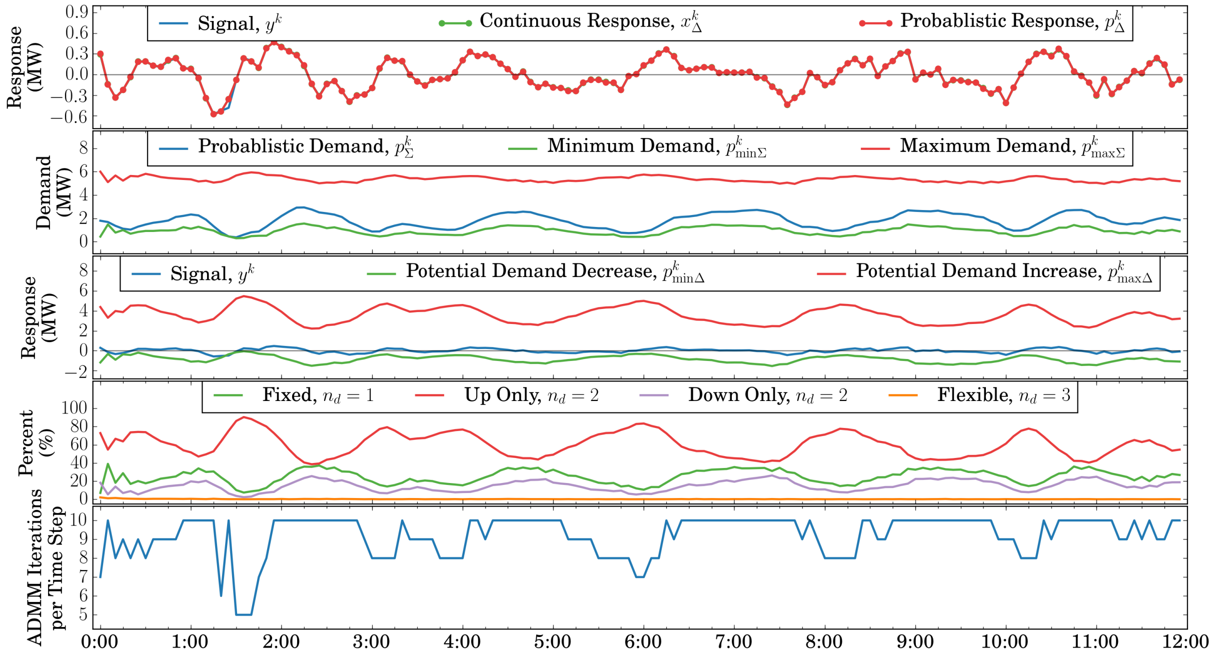

Figure 13: Homogeneous Population with 5 Minute Dwell Time

probabilistic responses are 8.13 kW and 11.80 kW, respectively. Note that this study employs the same number of TCLs with the same parameter values as those in the previous homogeneous study. However, due to the enforcement of the dwell time constraint, we observe a greater percentage of the population in the fixed and down only conditions. In the previous homogeneous study, the means of the fixed, up only, and down only populations over the 12 hours were 9.35%, 74.09%, and 16.23%, respectively. With the enforcement of the dwell time, the mean percentages are 24.87%, 60.22%, and 14.74%, respectively. Due in part to the increase in the fixed population, more ADMM iterations are required to find a solution.

5.3. *Heterogeneous Population*

To begin introducing heterogeneity, we have modeled the control of N = 10,000 refrigerators with parameters randomly drawn from the uniform distributions in Table 1. We have also raised the ADMM iterations limit to 40. The results from this study are presented in Figure 14 and show a success rate of 95.8% over the time period studied. The RMSEs of the continuous and probabilistic responses are 8.81 kW and 17.84 kW, respectively.

In this study, we have significantly decreased the population size and thus the potential for increasing demand. The second and third plots indicate that as we approach noon, we experience a decline in the maximum feasible power demand $p^k_{\max\Sigma}$ and the demand increase potential $p^k_{\max\Delta}$. The fourth plot shows the percentage of the population that is either fixed, flexible, or capable of only up or down responses and presents some insight into the loss of demand increase potential. Between midnight and 7:00, we observe that the TCLs generally oscillate between up only and down only, with the percent of fixed and flexible TCLs remaining small. After 7:00, the TCLs in the down only population begin to become fixed. Finally, the TCLs begin switching between up only and fixed, making it more difficult to perform generation following and driving up the number of ADMM iterations.

5.4. *Highly Heterogeneous Population*

In this study, we consider a highly heterogeneous population of refrigerators, water heaters, heat pumps, and baseboard heaters with parameters randomly drawn from the uniform distributions in Table 1. We model 3,000 refrigerators, 2,000 water heaters, 1,800 heat pumps, and 1,800 baseboard heaters for a total of N = 8,600 TCLs. We set the ADMM iterations limit to 20.

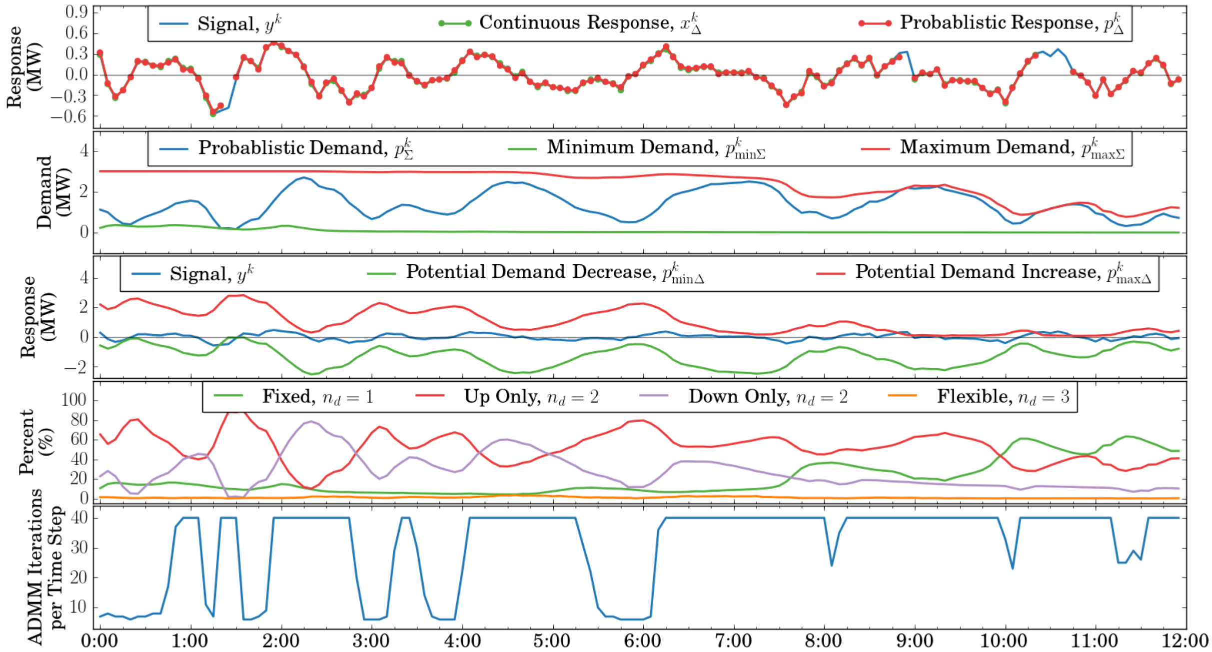

Figure 14: Heterogeneous Population

The results, presented in Figure 15, show a generation following success rate of 91.0% over the time period studied. Based on the fifth plot, we observe that for 96.5% of the time steps, the stopping criteria were not met and the ADMM algorithm hit the iterations limit of 20. However, in 90.3% of these time steps, the error was within the tolerance of 10 kW. The RMSEs of the continuous and probabilistic responses are 4.39 kW and 81.78 kW, respectively. This increase in the error of the probabilistic response can be attributed to the increased heterogeneity of the TCL population.

The fourth plot in Figure 15 shows that at each time interval, the percentage of fixed TCLs remained over 40%. Nonetheless, the potential for increasing the demand remained near 8 MW, declining slightly after 9:00 due to the rise in ambient temperatures (and thus a loss in demand increase potential from heat pumps and baseboard heaters). Overall, the population suffered from an insufficient potential for decreasing demand. This could be addressed by better conditioning the TCLs so that more remain in a flexible or down only condition or by extending the forecasting horizon beyond the next 5 minutes, allowing the aggregator and TCLs to better prepare for future signals.

5.5. *Heterogeneous Population with Divide and Conquer*

To address the error between the probabilistic response p^k_Δ and the signal y^k, we have re-simulated the highly heterogeneous population of $N = 8{,}600$ TCLs using the divide and conquer approach. In other words, we have run the ADMM algorithm 5 times. After each run, we fixed 20% of the total population so that after the final run, all 8,600 TCLs are fixed. Additionally, between each run, the N and d parameters are adjusted according to the results of the newly fixed TCLs. Lastly, as a warm start, the previous value of λ and adjusted values of $\bar{x}$ and $\bar{z}$ are employed to initialize the next ADMM run. If the error tolerance is violated at the end of an ADMM run, the algorithm is terminated. For the first ADMM run, the iteration limit is set to 20. For successive ADMM runs, the limit is 10.

To improve the performance of the algorithm, we have sorted the TCLs such that those with the highest power demand are fixed first and those with the lowest are fixed last. In other words, the order of consideration is heat pump, electric water heater, electric baseboard heater, and refrigerator.

The test results are presented in Figure 16. While we have increased the total number of ADMM iterations at each time interval, the RMSEs of the

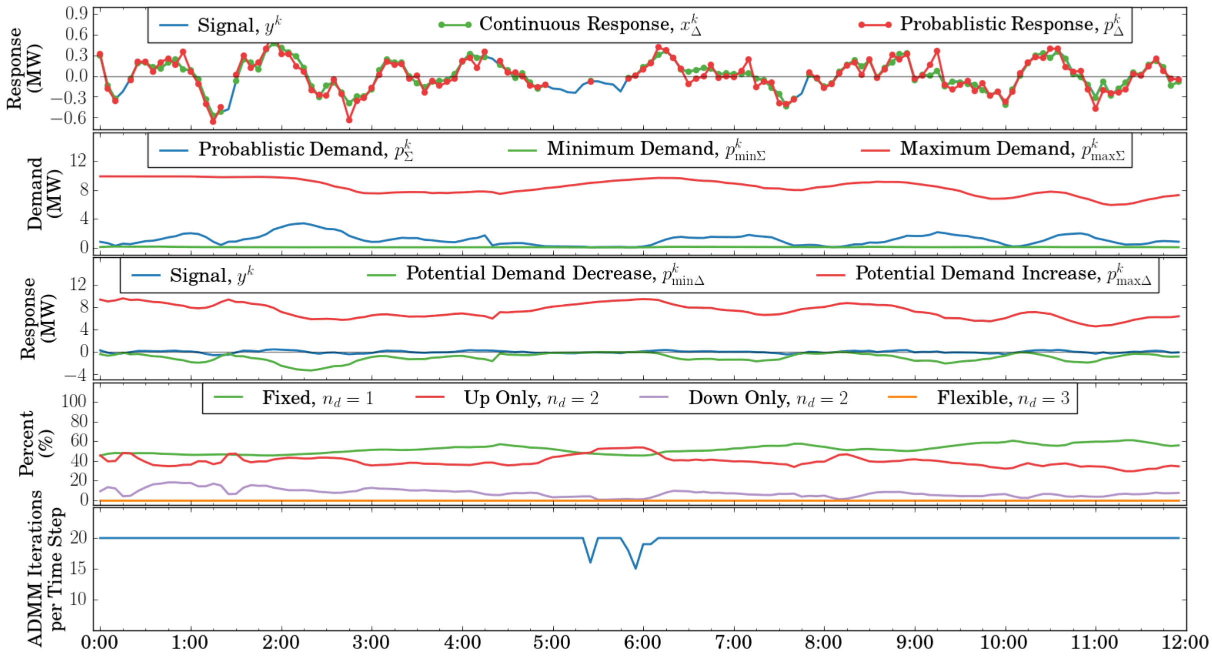

Figure 15: Highly Heterogeneous Population

continuous and probabilistic responses are now significantly reduced to 7.19 kW and 9.56 kW, respectively. This demonstrates that the TCLs can be controlled such that the probabilistic response p^k_Δ is within the error tolerance of 10 kW. The success rate for the time period simulated is 88.9%. Once again, the population struggles to match the required demand decrease. In the previous studies, the failed attempts terminated at 40 ADMM iterations, the upper limit. In this study, failed attempts are terminated after the first ADMM run of 20 iterations.

Lastly, because we are simulating each TCL with a one minute time step, we can reproduce the power demand for every minute, as shown in Figure 17. Because of the piecewise constant interpretation of the signal and the formulation of the aggregator's objective function, the electric power demand of the TCL population has step-like appearance.

5.6. *Increasing Population Size*

To test the impact of population size on the number of ADMM iterations, we have designed an experiment in which TCL populations of varying size are employed to respond to the load following signal. Each population is comprised of homogeneous refrigerators with identical parameters (the mean of the parameter ranges in Table 1).

To account for the variation in population size, the percentage of the signal followed by the aggregator is scaled such that the per TCL signal remains constant across the different populations. The same is done with the error tolerance of the aggregator. Specifically, the percentage of the signal is defined as 10^{-4}% per TCL and the error tolerance is 10^{-4} kW or 0.1 W per TCL. Therefore, 100 TCLs are employed to follow 0.01% of the signal with an error tolerance of 0.01 kW and 1,000,000 TCLs are employed to follow 100% of the signal with an error tolerance of 100 kW.

In this experiment, we generate populations of 100, 500, 1,000, 5,000, 10,000, 50,000, 100,000, 500,000, and 1,000,000 TCLs and employ each population to follow the first hour (i.e. first 12 time steps) of the signal. Additionally, the ADMM algorithm is stopped once the aggregate power demand of the population is within the error tolerance. This can be viewed as a relaxation of the stopping criteria in (21).

The results of this experiment are presented in Figures 18 and 19. Figure 18 shows the number ADMM iterations at each time step for the 100, 1,000, 10,000, 100,000, and 1,000,000 TCL populations. Note that for the 10,000, 100,000, and 1,000,000 TCL populations, the iteration numbers

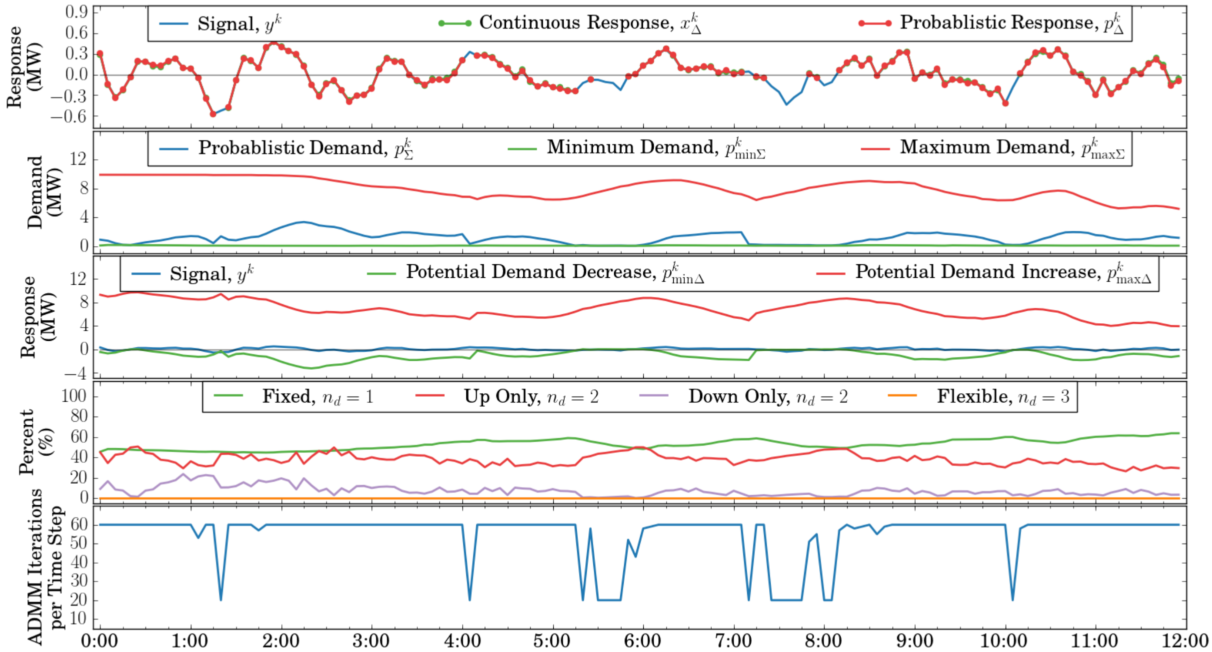

Figure 16: Highly Heterogeneous Population with Divide and Conquer

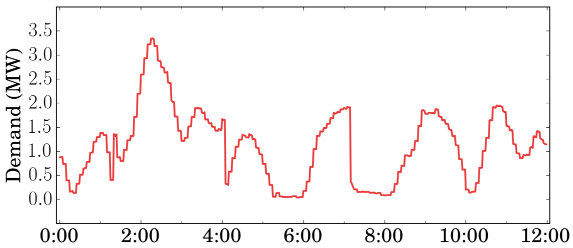

Figure 17: Power Demand on 1 Minute Time Scale

at each time step are equal.

Figure 19 shows the mean number of iterations in the the first hour of generation following for each of the nine TCL populations. The results suggest that the number of ADMM iterations is independent of the population size. Therefore, increasing the number of the TCLs in the population does not directly increase the number of ADMM iterations required to perform generation following.

5.7. Increasing α_x

The weighting term α_x represents the willingness of a TCL to permit temperature drift away from the setpoint. To test the impact of α_x on the number of ADMM iterations, we have designed an experiment in which TCL populations with varying α_x

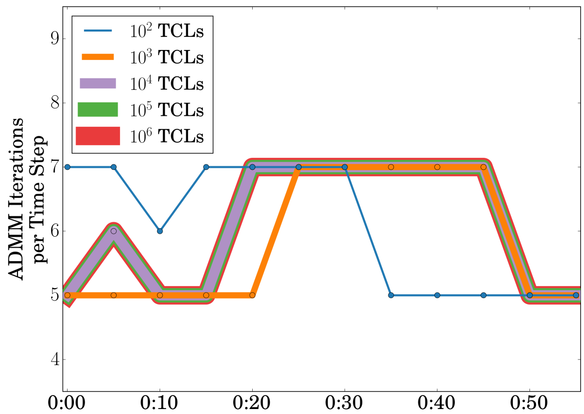

Figure 18: ADMM iterations at each time step with TCL populations of varying size

respond to 1% of the load following signal. Each population is comprised of 10,000 homogeneous refrigerators with identical parameters (the mean of the parameter ranges in Table 1) and is employed to follow the first hour (i.e. first 12 time steps) of the signal. We limit the number of ADMM iterations to 40.

The results of this experiment are presented in Figures 20, 21, and 22. Figure 20 shows the number ADMM iterations at each time step for the different

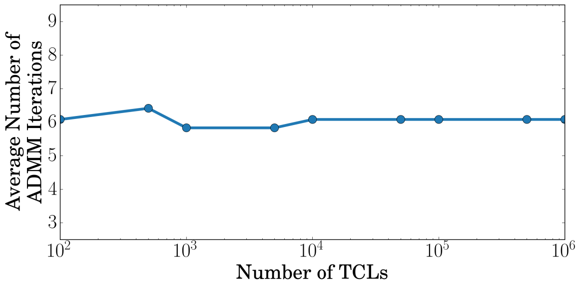

Figure 19: Mean number of ADMM iterations for each TCL population over first hour of generation following

values of α_x and Figure 21 shows the mean number of iterations in the first hour of generation following. As shown, when α_x is small, fewer ADMM iterations are required to find a solution. This can be attributed to the weighting of the aggregator objective function relative to the objective function of the individual TCLs in the population. In other words, the TCLs seek to minimize the global objective of generation following and allow their temperatures to drift from the setpoint. As α_x increases (and thus the relative weighting of the aggregator objective decreases), the TCLs become less cooperative and more iterations are required to find a solution.

Eventually, α_x increases to a point where the optimal solution of the distributed ADMM algorithm is to minimize the objectives of the individual TCLs rather than the aggregator objective. In other words, the refrigerators in the population choose to minimize the deviation of their internal temperatures from the setpoint rather than participating in the generation following aggregation. As a result, the average number of ADMM iterations increases to the limit of 40, as shown in Figures 20 and 21, and we observe an increase in the RMSE of the continuous response, as shown in Figure 22.

6. Conclusions

In this paper, we have presented an alternative control trajectory representation. This representation allows for the modeling of a TCL as a generalized assignment problem and fully recognizes the non-convex constraints of hysteretic dead-band systems. By relaxing the binary constraint, the problem becomes convex and the optimal solution can be interpreted as both a continuous and probabilistic solution.

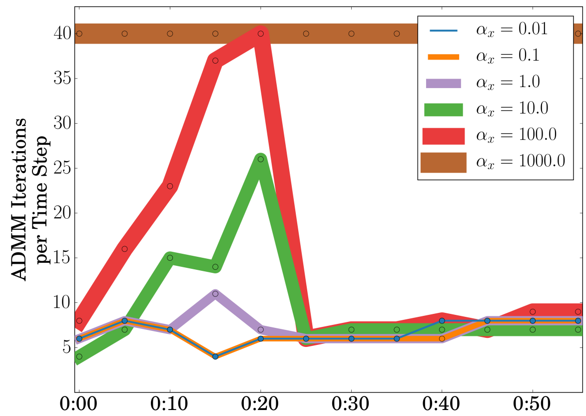

Figure 20: ADMM iterations at each time step with varying values of α_x

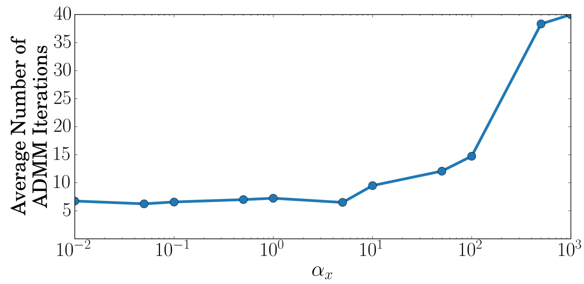

Figure 21: Mean number of ADMM iterations for each value of α_x over first hour of generation following

We have also presented a formulation of the sharing ADMM algorithm suitable for the distributed optimization of TCLs. The formulation is highly parallelizable and requires the broadcasting of only λ^k and $(\bar{x}^k - \bar{z}^k)$. Given the objective function of every agent is convex, the algorithm is guaranteed to converge to an optimal solution.

Finally, we have applied the sharing ADMM algorithm with TCL alternative control trajectory representation to the problem of 5-minute ahead renewable energy generation following. Findings of this paper include:

- Using actual wind and solar generation forecasts, ambient temperature records, and published TCL parameters, we have demonstrated how populations of TCLs can be optimized to perform power system services.

- By applying the alternative control trajectory representation to TCLs, we have shown how a population of systems with integer states can

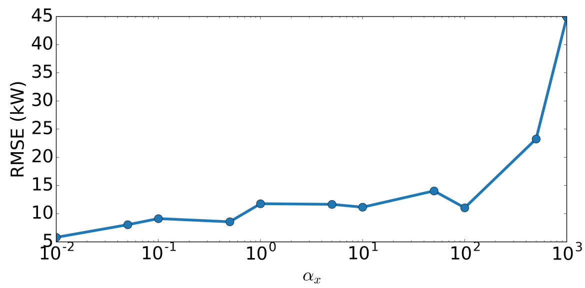

Figure 22: RMSE of continuous response for each value of α_x over first hour of generation following

be controlled using a convex algorithm.

- By distributing the computation using the sharing ADMM algorithm, we have demonstrated that the generation following algorithm can be scaled to large populations of TCLs without increasing the number of ADMM iterations.
- For highly heterogeneous TCL populations, we have shown that a divide and conquer approach can be employed to minimize the error between the probabilistic solution and the signal.

There are a number of advantages to the distributed TCL control method presented in this manuscript. Firstly, each TCL models its dynamics locally and there is no requirement that TCLs all employ the same model structure or control scheme. Individual TCLs can incorporate higher fidelity or device specific models and still participate in the distributed optimization. Secondly, TCLs can prevent short-cycling. For example, a TCL could exclude any alternative trajectories that violate a minimum dwell time. Thirdly, due to the bi-directional communication, the aggregator can have perfect knowledge of the population's future power demand. There is no need to estimate the power demand if the TCLs are capable of committing to the solution of the optimization algorithm. Quantifying and qualifying the advantages of these characteristics will be the focus of future research.

A challenge not addressed in this manuscript is that, because we are not centrally modeling the TCL population, the aggregator does not know the current state or generation following potential of the population. Methods for better understanding and maintaining the generation following potential, which is related to the average temperature of each TCL, will be the subject of future work. Similarly, understanding the impact of seasonal and regional weather conditions on the performance of the TCL aggregation will be future work.

Using our sharing ADMM algorithm, we have demonstrated the potential for TCLs to help maintain a continuous and instantaneous balance between generation and load by participating in real-time ancillary service markets. The deployment of such responsive load will be essential for maintaining the stability of power systems with high renewable energy penetration.

Appendix

6.1. Notation

To simplify equations, we employ the following notation and abbreviations throughout the paper.

ℓ_1-norm:

$$\|x\|_1 = \sum_{i=1}^{N} |x_i| \tag{38}$$

ℓ_2-norm:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{N} x_i^2} \tag{39}$$

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{40}$$

Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{41}$$

Sum:

$$\sum x_i = \sum_{i=1}^{N} x_i \tag{42}$$

Inner product:

$$\langle \lambda, x \rangle = \lambda^T x \tag{43}$$

with variable x, $\lambda \in \mathbf{R}^N$.

Temperature, °C	T
Discrete state, on/off	m
Power demand, kW	p
Temperature setpoint change, °C	u
Set of allowed setpoint changes	S_u
Ambient temperature, °C	T_∞
Temperature setpoint, °C	T_{set}
Temperature dead-band width, °C	δ
Thermal capacitance, kWh/°C	C
Thermal resistance, °C/kW	R
Energy transfer rate, °C/kW	P
Coefficient of performance	COP
Temperature trajectory	T_j
State trajectory	m_j
Power trajectory	p_j
Input trajectory	u_j
Set of temperature trajectories	$\mathbf{T}$
Set of state trajectories	$\mathbf{M}$
Set of power trajectories	$\mathbf{P}$
Set of input trajectories	$\mathbf{U}$
Number of control trajectories	N_a
Number of distinct control trajectories	N_d
Discrete solution	w^*
Continuous solution	$\hat{w}^*$
Probabilistic solution	$\tilde{w}$
Mean aggregator solution	$\bar{z}$
Mean TCL solution	$\bar{x}$
Continuous power demand solution	x^*
Probabilistic power demand solution	$\tilde{p}$
Aggregated continuous demand	x_Σ
Aggregated probabilistic demand	p_Σ
Aggregated continuous response	x_Δ
Aggregated probabilistic response	p_Δ
Dual variable	λ
Augmented Lagrangian parameter	ρ
Primal residual	r
Dual residual	s
Generation following signal	y
Desired power demand profile	d
Aggregator Coefficient	α_z
TCL Coefficient	α_x
Primal Feasibility Tolerance	ϵ^{primal}
Dual Feasibility Tolerance	ϵ^{dual}
Error Tolerance	ϵ^{error}
$\bar{\lambda}$ Limit	λ_+

Table 6: Nomenclature: Variables, parameters, and sets used throughout the paper

6.2. Sharing ADMM Optimality and Residuals

In this section, we derive the sharing ADMM residuals, which are required to define the stopping criteria. The necessary and sufficient optimality conditions for the sharing ADMM problem (18) are given by the primal feasibility,

$$x_i^* - z_i^* = 0 \tag{44}$$

and dual feasibility,

$$0 = \nabla f_i(x^*) + \lambda_i^* \tag{45}$$

$$0 = \nabla g(\textstyle\sum z_i^*) - \sum \lambda_i^* \tag{46}$$

for $i = 1, \ldots, N$ assuming f_i and g are differentiable.

Since z^{k+1} minimizes (18b) by definition, we can show that z^{k+1} and λ^{k+1} always satisfy (46),

$$\begin{aligned} 0 &= \nabla g(\textstyle\sum z_i^{k+1}) - (\sum \lambda_i^k + \sum \rho(x_i^{k+1} - z_i^{k+1})) \\ &= \nabla g(\textstyle\sum z_i^{k+1}) - \sum(\lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1})) \\ &= \nabla g(\textstyle\sum z_i^{k+1}) - \sum \lambda_i^{k+1} \end{aligned}$$

Therefore, optimality is achieved by satisfying (44) and (45). From (44), we can define the primal residual as

$$r_i^{k+1} = x_i^{k+1} - z_i^{k+1} \tag{47}$$

Since x_i^{k+1} minimizes (18a) by definition, we can show

$$\begin{aligned} 0 &= \nabla f_i(x_i^{k+1}) + \lambda_i^k + \rho(x_i^{k+1} - z_i^k) \\ &= \nabla f_i(x_i^{k+1}) + \lambda_i^k + \rho(x_i^{k+1} - z_i^k + z_i^{k+1} - z_i^{k+1}) \\ &= \nabla f_i(x_i^{k+1}) + (\lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1})) + \rho(z_i^{k+1} - z_i^k) \\ &= \nabla f_i(x_i^{k+1}) + \lambda_i^{k+1} + \rho(z_i^{k+1} - z_i^k) \end{aligned}$$

Thus, we can define the dual residual as

$$s_i^{k+1} = \nabla f_i(x_i^{k+1}) + \lambda_i^{k+1} = -\rho(z_i^{k+1} - z_i^k) \tag{48}$$

6.3. Averaged Sharing ADMM

In this section, we derive the averaged form of the sharing ADMM algorithm. The sharing ADMM algorithm (18) requires the local calculation of a z_i^k, λ_i^k, and r_i^k term for each agent $i = 1, \ldots, N$ in the network. Next, we will show that we can simplify the algorithm by introducing global variables $\bar{x}^k$, $\bar{z}^k$, and $\bar{\lambda}^k$ representing the arithmetic mean of all x_i^k, z_i^k, and λ_i^k, respectively.

We begin by introducing $\bar{z}^k$ into the z-update equation (18b), which can be rewritten as

$$\min_{z,\bar{z}} g(N\bar{z}) + \sum\left(\langle\lambda_i^k, -z_i\rangle + \frac{\rho}{2}\|x_i^{k+1} - z_i\|_2^2\right) \quad \text{s.t. } \bar{z} = \frac{1}{N}\sum z_i \tag{49}$$

or in augmented Lagrangian form

$$\begin{aligned}\mathscr{L}(z,\bar{z},\mu) &= g(N\bar{z}) + \sum\langle\lambda_i^k, -z_i\rangle \\ &+ \sum\left(\frac{\rho}{2}\|x_i^{k+1} - z_i\|_2^2\right) \\ &+ \mu^T\left(\bar{z} - \frac{1}{N}\sum z_i\right)\end{aligned}$$

Thus, for every iteration of the sharing ADMM algorithm, the optimal value of each z_i is

$$\begin{aligned}0 &= \frac{\partial\mathscr{L}}{\partial z_i}(z_i^*, \bar{z}^*, \mu^*) \\ &= \lambda_i^k + \rho(x_i^{k+1} - z_i^*) + \frac{\mu^*}{N} \\ &= \frac{1}{\rho}\left(\lambda_i^k + \frac{\mu^*}{N}\right) + x_i^{k+1} - z_i^* \\ z_i^* &= \frac{\mu^*}{N\rho} + \frac{\lambda_i^k}{\rho} + x_i^{k+1}\end{aligned} \tag{50}$$

Finally, we can calculate the optimal value of $\bar{z}$

$$\begin{aligned}\bar{z}^* &= \frac{1}{N}\sum z_i^* \\ &= \frac{1}{N}\sum\left(\frac{\mu^*}{N\rho} + \frac{\lambda_i^k}{\rho} + x_i^{k+1}\right) \\ &= \frac{1}{N}\left(\frac{\mu^*}{\rho} + \frac{1}{\rho}\sum\lambda_i^k + \sum x_i^{k+1}\right) \\ &= \frac{\mu^*}{N\rho} + \frac{\bar{\lambda}^k}{\rho} + \bar{x}^{k+1}\end{aligned} \tag{51}$$

Thus, substituting $\mu^*/N\rho$ from (51) into (50),

$$z_i^* = \bar{z}^* - \frac{\bar{\lambda}^k}{\rho} - \bar{x}^{k+1} + \frac{\lambda_i^k}{\rho} + x_i^{k+1} \tag{52}$$

or equivalently

$$z_i^{k+1} = \bar{z}^{k+1} + (x_i^{k+1} - \bar{x}^{k+1}) + \frac{1}{\rho}(\lambda_i^k - \bar{\lambda}^k) \tag{53}$$

Next, we can replace z_i^{k+1} in the λ_i-update equation (18c)

$$\begin{aligned}\lambda_i^{k+1} &= \lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1}) \\ &= \lambda_i^k \\ &+ \rho(x_i^{k+1} - (\bar{z}^{k+1} + x_i^{k+1} - \bar{x}^{k+1})) \\ &- (\lambda_i^k - \bar{\lambda}^k) \\ &= \bar{\lambda}^k + \rho(\bar{x}^{k+1} - \bar{z}^{k+1})\end{aligned} \tag{54}$$

which shows that the dual variables λ_i^k are all equal to the global $\bar{\lambda}^k$ and thus

$$z_i^{k+1} = \bar{z}^{k+1} + (x_i^{k+1} - \bar{x}^{k+1}) \tag{55}$$

Therefore, we can express the unscaled form of the averaged sharing ADMM algorithm as presented in (22).

With this averaged sharing ADMM form, the individual agents no longer update their own λ_i variable. Instead, a single aggregator updates $\bar{\lambda}$, along with $\bar{x}$ and $\bar{z}$, and reports these global variables to every agent in the network.

6.4. Averaged Sharing Residuals

In order to apply the stopping criteria, we must redefine the primal and dual residuals for the averaged form. We can substitute z_i^{k+1} from (55) into (47) and (48) in order to define the primal residual r_i^k and dual residual s_i^k in terms of $\bar{z}$, as shown in (23) and (24), respectively. The corresponding ℓ_2-norms of the stopping criteria are presented in (25).

7. References

[1] Y. V. Makarov, C. Loutan, J. Ma, P. De Mello, Operational impacts of wind generation on california power systems, Power Systems, IEEE Transactions on 24 (2) (2009) 1039–1050.

[2] Z. Xu, J. Ostergaard, M. Togeby, C. Marcus-Moller, Design and modelling of thermostatically controlled loads as frequency controlled reserve, in: Power Engineering Society General Meeting, 2007. IEEE, IEEE, 2007, pp. 1–6.

[3] K. Kostková, L. Omelina, P. Kyčina, P. Jamrich, An introduction to load management, Electric Power Systems Research 95 (2013) 184–191.

[4] K. G. Boroojeni, M. H. Amini, A. Nejadpak, S. Iyengar, B. Hoseinzadeh, C. L. Bak, A theoretical bilevel control scheme for power networks with large-scale penetration of distributed renewable resources, in: Electro Information Technology (EIT), 2016 IEEE International Conference on, IEEE, 2016, pp. 0510–0515.

[5] K. G. Boroojeni, S. Mokhtari, M. Amini, S. Iyengar, Optimal two-tier forecasting power generation model in smart grids, arXiv preprint arXiv:1502.00530.

[6] M. H. Amini, B. Nabi, M.-R. Haghifam, Load management using multi-agent systems in smart distribution network, in: IEEE PES General Meeting, Vancouver, BC, Canada, 2013.

[7] M. Amini, J. Frye, M. D. Ilić, O. Karabasoglu, Smart residential energy scheduling utilizing two stage mixed integer linear programming, in: North American Power Symposium (NAPS), 2015, IEEE, 2015, pp. 1–6.

[8] F. Kamyab, M. Amini, S. Sheykhha, M. Hasanpour, M. M. Jalali, Demand response program in smart grid using supply function bidding mechanism, IEEE Transactions on Smart Grid 7 (3) (2016) 1277–1284.

[9] B. J. Kirby, Frequency regulation basics and trends, ORNL/TM 2004/291, Oak Ridge National Laboratory.

[10] E. Hirst, B. Kirby, Separating and measuring the regulation and load-following ancillary services, Utilities Policy 8 (1999) 75–81.

[11] M. Bragard, N. Soltau, S. Thomas, R. W. De Doncker, The balance of renewable sources and user demands in grids: Power electronics for modular battery energy storage systems, IEEE Transactions on Power Electronics 25 (12) (2010) 3049–3056.

[12] G. Strbac, Demand side management: Benefits and challenges, Energy Policy 36 (12) (2008) 4419–4426.

[13] D. S. Callaway, I. Hiskens, et al., Achieving controllability of electric loads, Proceedings of the IEEE 99 (1) (2011) 184–199.

[14] J. L. Mathieu, M. Dyson, D. S. Callaway, Using residential electric loads for fast demand response: The potential resource and revenues, the costs, and policy recommendations, ACEEE Summer Study on Energy Efficiency in Buildings.

[15] D. S. Callaway, Tapping the energy storage potential in electric loads to deliver load following and regulation, with application to wind energy, Energy Conversion and Management 50 (5) (2009) 1389–1400.

[16] S. Ihara, F. C. Schweppe, Physically based modeling of cold load pickup, IEEE Transactions on Power Apparatus and Systems 100 (9) (1981) 4142–4250.

[17] C. Chong, A. Debs, Statistical synthesis of power system functional load models, in: 1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes, no. 18, 1979, pp. 264–269.

[18] R. E. Mortensen, K. P. Haggerty, A stochastic computer model for heating and cooling loads., IEEE Transactions on Power Systems 3 (3) (1998) 1213–1219.

[19] C. Ucak, R. Çağlar, The effects of load parameter dispersion and direct load control actions on aggregated load, in: Power System Technology, 1998. Proceedings. POWERCON'98. 1998 International Conference on, Vol. 1, IEEE, 1998, pp. 280–284.

[20] A. Pahwa, C. Brice III, Modeling and system identification of residential air conditioning load, Power Apparatus and Systems, IEEE Transactions on (6) (1985) 1418–1425.

[21] R. Malhame, C.-Y. Chong, Electric load model synthesis by diffusion approximation of a high-order hybrid-state stochastic system, Automatic Control, IEEE Transactions on 30 (9) (1985) 854–860.

[22] S. Bashash, H. K. Fathy, Modeling and control insights into demand-side energy management through setpoint control of thermostatic loads, in: American Control Conference (ACC), 2011, IEEE, 2011, pp. 4546–4553.

[23] C. Perfumo, E. Kofman, J. H. Braslavsky, J. K. Ward, Load management: Model-based control of aggregate power for populations of thermostatically controlled loads, Energy Conversion and Management 55 (2012) 36 – 48. doi:10.1016/j.enconman.2011.10.019.

[24] S. B. S. Kundu, N. Sinitsyn, I. Hiskens, Modeling and control of thermostatically controlled loads, in: Proceedings of the 17th Power Systems Computation Conference, Stockholm, Sweden, 2011.

[25] L. C. Totu, R. Wisniewski, J. Leth, Modeling populations of thermostatic loads with switching rate actuation, arXiv preprint arXiv:1411.2864.

[26] J. L. Mathieu, S. Koch, D. S. Callaway, State estimation and control of electric loads to manage real-time energy imbalance, Power Systems, IEEE Transactions on 28 (1) (2013) 430–440.

[27] S. Koch, J. L. Mathieu, D. S. Callaway, Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services, in: Proc. PSCC, 2011, pp. 1–7.

[28] W. Zhang, K. Kalsi, J. Fuller, M. Elizondo, D. Chassin, Aggregate model for heterogeneous thermostatically controlled loads with demand response, in: Power and Energy Society General Meeting, 2012 IEEE, IEEE, 2012, pp. 1–8.

[29] W. Zhang, J. Lian, C.-Y. Chang, K. Kalsi, Y. Sun, Reduced-order modeling of aggregated thermostatic loads with demand response, in: 2012 IEEE 51st Annual Conference on Decision and Control, 2012, pp. 5592–5597.

[30] W. Zhang, J. Lian, C.-Y. Chang, K. Kalsi, Aggregated modeling and control of air conditioning loads for demand response, Power Systems, IEEE Transactions on 28 (4) (2013) 4655–4664.

[31] S. Moura, V. Ruiz, J. Bendsten, Modeling heterogeneous populations of thermostatically controlled loads using diffusion-advection PDEs, in: ASME 2013 Dynamic Systems and Control Conference, American Society of Mechanical Engineers, 2013, pp. V002T23A001–V002T23A001.

[32] A. Ghaffari, S. Moura, M. Krstić, Modeling, control, and stability analysis of heterogeneous thermostatically controlled load populations using partial differential equations, Journal of Dynamic Systems, Measurement, and Control 137 (10) (2015) 101009.

[33] E. Vrettos, G. Andersson, Combined load frequency control and active distribution network management with thermostatically controlled loads, in: Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on, IEEE, 2013, pp. 247–252.

[34] S. Iacovella, F. Ruelens, P. Vingerhoets, B. Claessens, G. Deconinck, Cluster control of heterogeneous thermostatically controlled loads using tracer devices, IEEE Transactions on Smart Grid (2015) 1–9.

[35] J. L. Mathieu, M. Kamgarpour, J. Lygeros, D. S. Callaway, Energy arbitrage with thermostatically controlled loads, in: Control Conference (ECC), 2013 European, IEEE, 2013, pp. 2519–2526.

[36] J. L. Mathieu, M. Kamgarpour, J. Lygeros, G. Andersson, D. S. Callaway, Arbitraging intraday wholesale energy market prices with aggregations of thermostatic loads, Power Systems, IEEE Transactions on 30 (2) (2015) 763–772.

[37] H. Hao, B. M. Sanandaji, K. Poolla, T. L. Vincent, Aggregate flexibility of thermostatically controlled loads, Power Systems, IEEE Transactions on 30 (1) (2015) 189–198.

[38] J. A. Short, D. G. Infield, L. L. Freris, Stabilization of grid frequency through dynamic demand control, Power Systems, IEEE Transactions on 22 (3) (2007) 1284–1293.

[39] D. Angeli, P.-A. Kountouriotis, A stochastic approach to dynamic-demand refrigerator control, Control Systems Technology, IEEE Transactions on 20 (3) (2012) 581–592.

[40] S. H. Tindemans, V. Trovato, G. Strbac, Decentralized control of thermostatic loads for flexible demand response, Control Systems Technology, IEEE Transactions on 23 (5) (2015) 1685–1700.

[41] M. Wetter, Modelica-based modelling and simulation to support research and development in building energy and control systems, Journal of Building Performance Simulation 2 (2) (2009) 143–161.

[42] S. Goyal, P. Barooah, A method for model-reduction of non-linear thermal dynamics of multi-zone buildings, Energy and Buildings 47 (2012) 332–340.

[43] H. Okuyama, Y. Onishi, System parameter identification theory and uncertainty analysis methods for multi-zone building heat transfer and infiltration, Building and Environment 54 (2012) 39–52.

[44] P. Radecki, B. Hencey, Online building thermal parameter estimation via unscented kalman filtering, in: American Control Conference (ACC), 2012, IEEE, 2012, pp. 3056–3062.

[45] E. M. Burger, H. E. Perez, S. J. Moura, Piecewise Linear Thermal Model and Recursive Parameter Estimation of a Residential Heating System (January 2015).
URL http://escholarship.org/uc/item/8kx450mg

[46] H. Xing, Y. Mou, Z. Lin, M. Fu, Fast distributed power regulation method via networked thermostatically controlled loads, Vol. 19, Cape Town, South Africa, 2014, pp. 5439 – 5444.

[47] M. Liu, Y. Shi, Distributed model predictive control of thermostatically controlled appliances for providing balancing service, in: 2014 IEEE 53rd Annual Conference on Decision and Control (CDC), 2014, pp. 4850–4855.

[48] L. C. Totu, J. Leth, R. Wisniewski, Control for large scale demand response of thermostatic loads*, in: American Control Conference (ACC), 2013, IEEE, 2013, pp. 5023–5028.

[49] A. Safdarian, M. Ali, M. Fotuhi-Firuzabad, M. Lehtonen, Domestic ewh and hvac management in smart grids: Potential benefits and realization, Electric Power Systems Research 134 (2016) 38–46.

[50] D. Guo, R. Zheng, Z. Lin, G. Yan, A game-theoretic approach to decentralized control of heterogeneous load population, Electric Power Systems Research 140 (2016) 552–559.

[51] E. M. Burger, S. J. Moura, Recursive Parameter Estimation of Thermostatically Controlled Loads via Unscented Kalman Filter, Sustainable Energy, Grids and Networks 8 (2016) 12–25.
URL http://escholarship.org/uc/item/7t453713

[52] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.

[53] M. Kraning, E. Chu, J. Lavaei, S. Boyd, Message passing for dynamic network energy management, arXiv preprint arXiv:1204.1106.

[54] T. Goldstein, B. O'Donoghue, S. Setzer, R. Baraniuk, Fast alternating direction optimization methods, UCLA Computational and Applied Mathematics Report (2012) 12–35.

[55] Weather Underground Web Service and API.
URL wunderground.com/weather/api/

[56] California independent system operator.
URL oasis.caiso.com

[57] E. J. Cands, M. Wakin, S. Boyd, Enhancing sparsity by reweighted l1 minimization, Journal of Fourier Analysis and Applications 14 (5) (2008) 877–905.