Title

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

Authors

Burger, Eric M.
Moura, Scott J.

Publication Date

2015-10-06

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

Eric M. Burger, Scott J. Moura

***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). 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. To perform distributed optimization across large populations of TCLs, we apply a variation of the alternating direction method of multipliers (ADMM) algorithm. We numerically demonstrate the algorithm's potential for controlling a TCL population's total power demand within an error tolerance of 10 kW.**

I. Introduction

A. 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]. 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 [3][4]. In the near-term, responsive thermostatically controlled loads (TCLs) have a high potential for providing such ancillary services [5][6].

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

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 [6][7].

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 deadband range. This natural flexibility makes TCLs a promising candidate for participating in power system services.

B. Literature Review

Past literature on modelling and control of TCL populations has generally focused on aggregation methods with centralized control. Malhame and Chong's study [8] 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 [8], define the aggregate behavior of a homogeneous population. More recently, [9] developed a diffusion-advection partial differential equation (PDE) model and parameter identification scheme for an aggregated population of heterogeneous TCLs. In [10], the authors present a deterministic hybrid PDE-based model for heterogeneous TCL populations, and apply a uniform deadband shifting strategy for control.

In [11], the author uses a linearized Fokker-Planck model to describe the aggregated behavior of a TCL population. Direct 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 [11] by considering sliding mode control [12], proportional-integral control [13], and linear quadratic regulators [14].

In [15], the authors employ a linear time-invariant (LTI) representation of a TCL population. As in [12], a "state bin" modelling 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

Corresponding Author: Eric M. Burger, Email: ericburger@berkeley.edu, Phone: 317-385-1768

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.

heterogeneous TCL population. Similarly, [16] uses a state-bin concept with clustering to account for TCL heterogeneity. In [7], the authors 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.

This work diverges from the above literature in multiple respects:

- All the aforementioned research presents fully centralized control schemes. This paper presents a distributed control scheme with a centralized aggregator via ADMM. Related distributed control schemes use consensus coordination [17] or distributed model predictive control [18].
- 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 load trajectory. Therefore, if necessary, TCL parameter identification can be performed locally.
- We do not employ an aggregate model of the TCL population. Thus, rather than modelling the entire population, the central aggregator is only responsible for updating an incentive variable that drives the population towards a desired behavior.
- We do not use continuous setpoint control. In this paper, all temperature setpoint offsets are integer valued and therefore easily implementable on legacy hardware.
- In practice, a TCL is not required to participate at every time step. Because the TCL population is not centrally modelled, the distributed scheme is robust to an arbitrarily large loss or acquisition of agents.
- Our proposed modelling and control approach is capable of honoring highly non-convex constraints, such as minimum dwell time - a critically important practical constraint that eliminates compressor short-cycling.

For the distributed optimization of a TCL population, we present a variant of the ADMM algorithm known as sharing ADMM [19]. Due to its parallelizability and convergence characteristics, the sharing ADMM algorithm is generally applicable to the minimization of distributed agents. 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 forecast). 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.

C. Paper Outline

This paper is organized as follows. Section II discusses the TCL model and the alternative control trajectory representation. Section III overviews the sharing ADMM algorithm. Section IV formulates sharing ADMM for distributed TCL control. Section V provides numerical examples of our proposed algorithms, and highlights its applicability to highly heterogeneous populations. Finally, Section VI summarizes key results and the Appendix defines mathematical notation.

II. TCL Model and Optimization

A. Hybrid State Model

Each TCL is modeled using the hybrid state discrete time model [11][20][21]

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

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, $T_a^n \in \mathbf{R}$ the ambient temperature (°C), $T_{set} \in \mathbf{R}$ the temperature setpoint (°C), and $\delta \in \mathbf{R}$ the temperature deadband 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 $^\circ\mathrm{C}^2$). In this paper, we assume a noise standard deviation σ of $0.01^\circ\mathrm{C}/\sqrt{sec}$ or $0.6^\circ\mathrm{C}/\sqrt{hr}$.

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

$$T^{n+1} = \theta_1 T^n + (1-\theta_1)(T_a^n + \theta_2 m^n) + \theta_3$$
$$m^{n+1} = \begin{cases} 1 & \text{if } \theta_2 > 0 \text{ and} \\ & T^{n+1} < T_{set} - \frac{\delta}{2} + u^n \\ 0 & \text{if } \theta_2 > 0 \text{ and} \\ & T^{n+1} > T_{set} + \frac{\delta}{2} + u^n \\ 1 & \text{if } \theta_2 < 0 \text{ and} \\ & T^{n+1} > T_{set} + \frac{\delta}{2} + u^n \\ 0 & \text{if } \theta_2 < 0 \text{ and} \\ & T^{n+1} < T_{set} - \frac{\delta}{2} + u^n \\ m^n & \text{otherwise} \end{cases} \quad (3)$$

As noted in [11][21], 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 IV. 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.

B. 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 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

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

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.

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.

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

The input and output variables can be expressed compactly as

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

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 $j = 1, \ldots, N_a$. 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 $j = 2, \ldots, N_a$, 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 $N_d \in \{1, \ldots, N_a\}$.

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 $n = 1, \ldots, 20$. For trajectory $j = 2$, $u_2^n = 0$ for $n = 1, \ldots, 10$ and $u_2^n = -1$ for $n = 11, \ldots, 20$. For trajectory $j = 3$, $u_3^n = 0$ for $n = 1, \ldots, 10$ and $u_3^n = 1$ for $n = 11, \ldots, 20$.

The TCL has been simulated using (3) with a default setpoint T_{set} of 2.5°C, a deadband 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,

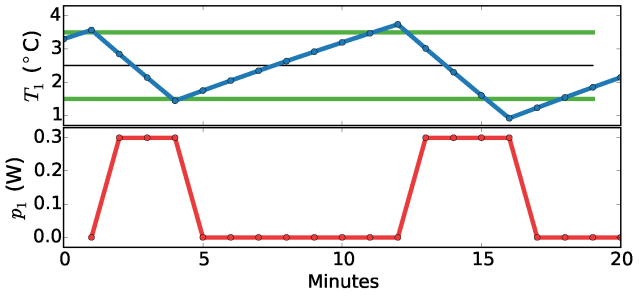

Fig. 1: Example Alternate Trajectories given u_1

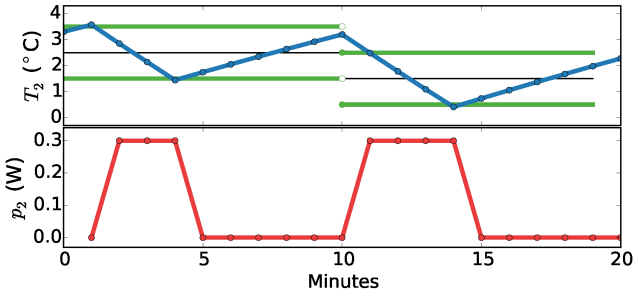

Fig. 2: Example Alternate Trajectories given u_2

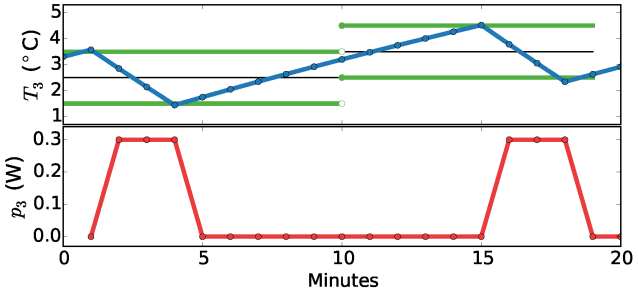

Fig. 3: Example Alternate Trajectories given u_3

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.

C. 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

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

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

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

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, (8) guarantees that only one component of minimizer w^* is non-zero. However, since the program is non-convex and NP-complete, it is unsuitable for many applications. By relaxing the binary constraint such that $\hat{w} \in \mathbf{R}^{N_d}$, we can express the convex program as

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

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

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

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). It should be noted that 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).

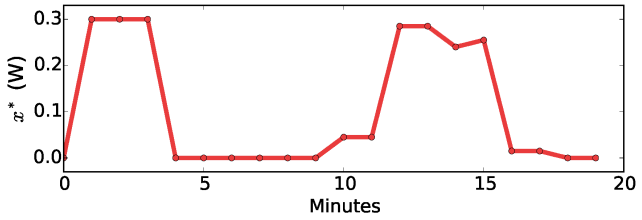

Fig. 4: Example Solution to Convex Program

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 $\mathbf{T}$ and $\mathbf{P}$ 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 $\tilde{w}$,

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

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 ($\tilde{w} \in \{0, 1\}^{N_d}$).

III. 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 [19][22] for a more complete description of the algorithm. Next, we discuss a special case of block separable problems referred to as *sharing* ADMM [19]. 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 [19].

A. ADMM

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

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

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: Dual $\lambda^k \to \lambda^*$ as $k \to \infty$

We refer the reader to [19][22] for a discussion of the augmented Lagrangian, scaled form, primal and dual residuals, and convergence rates.

B. Sharing ADMM

In this paper, we consider an ADMM-based method for solving the generic *sharing* problem using distributed optimization, as presented in [19]. 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

$$x_i^{k+1} = \underset{x_i}{\text{argmin}} \ f_i(x_i) + \langle \lambda_i^k, x_i \rangle + \frac{\rho}{2}\|x_i - z_i^k\|_2^2 \tag{18a}$$

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

$$\lambda_i^{k+1} = \lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1}) \tag{18c}$$

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.

C. Sharing Optimality and Residuals

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

$$x_i^* - z_i^* = 0 \tag{19}$$

and dual feasibility,

$$0 = \nabla f_i(x^*) + \lambda_i^* \tag{20}$$

$$0 = \nabla g(\sum z_i^*) - \sum \lambda_i^* \tag{21}$$

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 (21),

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

Therefore, optimality is achieved by satisfying (19) and (20). From (19), we can define the primal residual as

$$r_i^{k+1} = x_i^{k+1} - z_i^{k+1} \tag{22}$$

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

$$s_i^{k+1} = \nabla f_i(x_i^{k+1}) + \lambda_i^{k+1} = -\rho(z_i^{k+1} - z_i^k) \tag{23}$$

D. Stopping Criterion

We define the stopping criterion as presented in [19] by

$$\|r^k\|_2 \leq \epsilon^{primal} \text{ and } \|s^k\|_2 \leq \epsilon^{dual} \tag{24}$$

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 (19) and (20). In this paper, we set $\epsilon^{primal} = \epsilon^{dual} = 1.0$.

E. Averaged Sharing ADMM

As written, our sharing ADMM algorithm 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

$$\begin{aligned} \min_{z,\bar{z}} \ & g(N\bar{z}) + \sum \left(\langle \lambda_i^k, -z_i \rangle + \frac{\rho}{2}\|x_i^{k+1} - z_i\|_2^2 \right) \\ \text{s.t.} \ & \bar{z} = \frac{1}{N}\sum z_i \end{aligned} \tag{25}$$

or in augmented Lagrangian form

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

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

$$\begin{aligned} 0 = \frac{\partial \mathscr{L}}{\partial z_i}(z_i^*, \bar{z}^*, \mu^*) &= \lambda_i^k + \rho(x_i^{k+1} - z_i^*) + \frac{\mu^*}{N} \\ &= \frac{1}{\rho}(\lambda_i^k + \frac{\mu^*}{N}) + x_i^{k+1} - z_i^* \\ z_i^* &= \frac{\mu^*}{N\rho} + \frac{\lambda_i^k}{\rho} + x_i^{k+1} \end{aligned} \tag{26}$$

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

$$\begin{aligned}\bar{z}^* &= \frac{1}{N}\sum z_i^* \\ &= \frac{1}{N}\sum(\frac{\mu^*}{N\rho} + \frac{\lambda_i^k}{\rho} + x_i^{k+1}) \\ &= \frac{1}{N}(\frac{\mu^*}{\rho} + \frac{1}{\rho}\sum\lambda_i^k + \sum x_i^{k+1}) \\ &= \frac{\mu^*}{N\rho} + \frac{\bar{\lambda}^k}{\rho} + \bar{x}^{k+1}\end{aligned} \tag{27}$$

Thus, substituting $\mu^*/N\rho$ from (27) into (26),

$$z_i^* = \bar{z}^* - \frac{\bar{\lambda}^k}{\rho} - \bar{x}^{k+1} + \frac{\lambda_i^k}{\rho} + x_i^{k+1} \tag{28}$$

or equivalently

$$z_i^{k+1} = \bar{z}^{k+1} + (x_i^{k+1} - \bar{x}^{k+1}) + \frac{1}{\rho}(\lambda_i^k - \bar{\lambda}^k) \tag{29}$$

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

$$\begin{aligned}\lambda_i^{k+1} &= \lambda_i^k + \rho(x_i^{k+1} - z_i^{k+1}) \\ &= \lambda_i^k + \rho(x_i^{k+1} - (\bar{z}^{k+1} + x_i^{k+1} - \bar{x}^{k+1})) \\ &\quad - (\lambda_i^k - \bar{\lambda}^k) \\ &= \bar{\lambda}^k + \rho(\bar{x}^{k+1} - \bar{z}^{k+1})\end{aligned} \tag{30}$$

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

$$z_i^{k+1} = \bar{z}^{k+1} + (x_i^{k+1} - \bar{x}^{k+1}) \tag{31}$$

Therefore, the unscaled form of the averaged sharing ADMM algorithm is

$$\begin{aligned}x_i^{k+1} &= \underset{x_i}{\text{argmin}}\ f_i(x_i) + \langle\bar{\lambda}^k, x_i\rangle \\ &\quad + \frac{\rho}{2}\|x_i - x_i^k + \bar{x}^k - \bar{z}^k\|_2^2 \end{aligned} \tag{32a}$$

$$\begin{aligned}\bar{z}^{k+1} &= \underset{\bar{z}}{\text{argmin}}\ g(N\bar{z}) + \langle\bar{\lambda}^k, -N\bar{z}\rangle \\ &\quad + \frac{N\rho}{2}\|\bar{x}^{k+1} - \bar{z}\|_2^2 \end{aligned} \tag{32b}$$

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + \rho(\bar{x}^{k+1} - \bar{z}^{k+1}) \tag{32c}$$

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.

F. Averaged Sharing Residuals

In order to apply the stopping criterion, we must redefine the primal and dual residuals for the averaged form. We can substitute z_i^{k+1} from (31) into (22) and (23) in order to define the primal residual r_i^k and dual residual s_i^k in terms of $\bar{z}$

$$r_i^{k+1} = \bar{x}^{k+1} - \bar{z}^{k+1} \tag{33}$$

$$\begin{aligned}s_i^{k+1} &= \rho((\bar{x}^{k+1} - \bar{x}^k) - (x_i^{k+1} - x_i^k) \\ &\quad - (\bar{z}^{k+1} - \bar{z}^k))\end{aligned} \tag{34}$$

and the corresponding ℓ_2-norms of the stopping criterion

$$\begin{aligned}\|r^k\|_2 &= N\|\bar{x}^k - \bar{z}^k\|_2 \\ \|s^k\|_2 &= \sum\|s_i^k\|_2\end{aligned} \tag{35}$$

IV. 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 final sharing ADMM algorithm for the TCL population. Results from multiple studies are described in the next section. Our formulation is based on the following assumptions:

1) 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).
2) 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.
3) 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.

A. 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 (3) with published model parameter ranges, given in Table I and adopted from [7]. 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_a^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 [23]. The electric power demand of the TCL at each time step is given by (2).

TCL optimal control will take the form of setpoint manipulation. For each TCL type, 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 $S_u = \{0, -2, 1\}$,

	Fridge	Water Heater	Heat Pump	Baseboard Heater
R	[80, 100]	[100, 140]	[1.5, 2.5]	[1.5, 2.5]
C	[0.4, 0.8]	[0.2, 0.6]	[0.15, 0.25]	[0.15, 0.25]
P	[-1, -0.2]	[4, 5]	[14, 25.2]	[0.5, 1.5]
COP	2	1	3.5	1
T_{set}	[1.7, 3.3]	[43, 54]	[15, 24]	[15, 24]
δ	[1, 2]	[2, 4]	[0.25, 1]	[0.25, 1]
T_a	20	20	variable	variable
$Zones$	1	1	[5,10]	[1,2]
N_a	3	3	3	3
S_u	{0, -2, 1}	{0, -5, 5}	{0, -2, 1}	{0, -2, 1}

TABLE I: TCL Parameter Ranges adopted from [7]

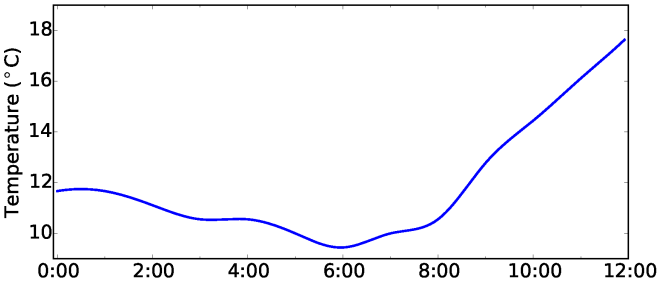

Fig. 5: Ambient Temperature Data for Berkeley, CA, on the Morning of 3/19/2015

$$u_1 = (0, 0, 0, 0, 0)$$
$$u_2 = (-2, -2, -2, -2, -2)$$
$$u_3 = (1, 1, 1, 1, 1)$$

In other words, 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. Thus, for any 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 V.

Thus, 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.

$$\begin{aligned} \underset{\hat{w}}{\text{minimize}} \ & \alpha_x \|\mathbf{T}^T \hat{w} - T_{set}\|_2^2 \\ \text{subject to} \ & \textstyle\sum \hat{w}_j = 1 \\ & \hat{w} \geq 0 \end{aligned} \tag{36}$$

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 $x_i^* = \mathbf{P}^T \hat{w}_i^*$. Given $\hat{w}_i^*$ and (10), we denote the *probabilistic* solution as $\tilde{p}_i = \mathbf{P}^T \tilde{w}_i$. As previously stated, $\hat{w}_i^*$ and x_i^* are guaranteed to be optimal, but $\tilde{w}_i$ and $\tilde{p}_i$ may be sub-optimal.

B. 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) [24]. 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 unconstrained least-squares fit.

$$\text{minimize} \ \alpha_z \|\textstyle\sum x_i - d\|_2^2 \tag{37}$$

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

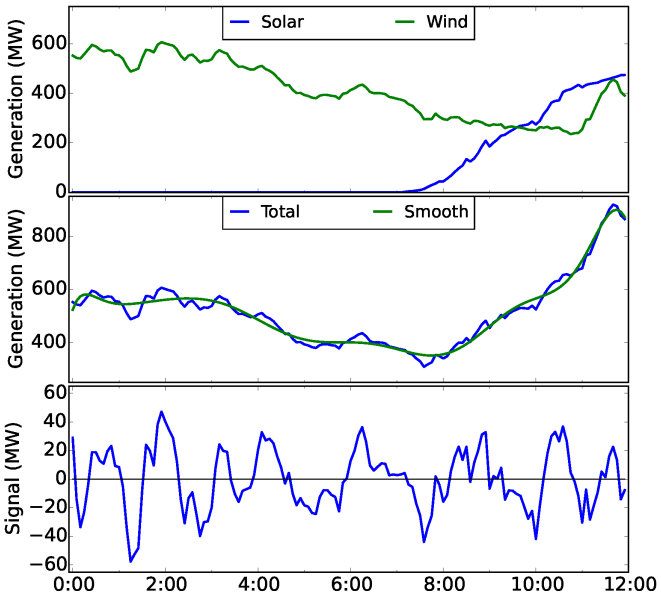

Fig. 6: California ISO Wind and Solar Generation 5-Min Forecasts for 3/19/2015 (Top), Smooth Polynomial Fit of Total Generation (Center), and exemplary 5-minute Generation Following Signal (Bottom)

$N_t = 5$ 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 $n = 1, \dots, N_t$). 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,

$$d^n = \begin{cases} \sum_i \tilde{p}_i^{n-1} + y^n & \text{if } n = 1 \\ d^{n-1} & \text{otherwise} \end{cases} \quad \forall\ n = 1, \dots, N_t \tag{38}$$

C. TCL Sharing ADMM

Given the TCL and aggregator optimization programs (36) and (37), we can now define the sharing ADMM algorithm for power generation following using a population of TCLs.

$$\hat{w}_i^{k+1} = \underset{\hat{w}_i}{\text{argmin}}\ \alpha_{x,i}\|\mathbf{T}_i^T \hat{w}_i - T_{set,i}\|_2^2 + \langle \bar{\lambda}^k, \mathbf{P}_i^T \hat{w}_i \rangle + \frac{\rho}{2}\|\mathbf{P}_i^T \hat{w}_i - x_i^k + \bar{r}^k\|_2^2 \tag{39a}$$

$$\text{s. to} \quad \sum \hat{w}_j = 1, \quad \hat{w} \geq 0$$

$$x_i^{k+1} = \mathbf{P}_i^T \hat{w}_i^{k+1} \tag{39b}$$

$$\bar{z}^{k+1} = \underset{\bar{z}}{\text{argmin}}\ \alpha_z \|N\bar{z} - d\|_2^2 + \langle \bar{\lambda}^k, -N\bar{z} \rangle + \frac{N\rho}{2}\|\bar{x}^{k+1} - \bar{z}\|_2^2 \tag{39c}$$

$$\bar{r}^{k+1} = \bar{x}^{k+1} - \bar{z}^{k+1} \tag{39d}$$

$$\bar{\lambda}^{k+1} = \bar{\lambda}^k + \rho(\bar{r}^{k+1}) \tag{39e}$$

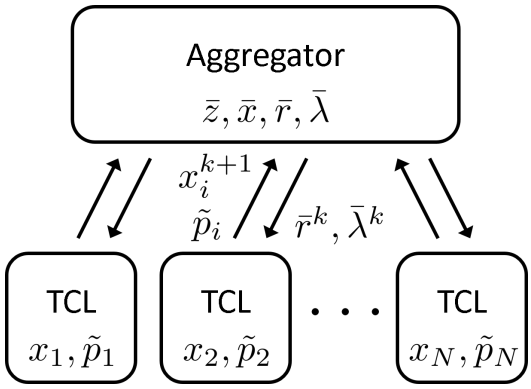

Fig. 7: Distributed TCL Optimization Structure

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.

Figure 7 illustrates the structure of the distributed system. At each ADMM iteration, the TCLs report their decision variables x_i^{k+1} to the aggregator but not to the other agents in the network. At the end of the iteration, the aggregator reports the updated incentive variable $\bar{\lambda}^{k+1}$ and the mean primal residual $\bar{r}^{k+1}$ (i.e. the difference between $\bar{x}^{k+1}$ and $\bar{z}^{k+1}$) to the TCL population. Each TCL's $\mathbf{T}$, $\mathbf{P}$, and $\hat{w}^k$ remain private. In addition to the stopping criteria (24), we impose a limit on the absolute value of $\bar{\lambda}$ (i.e. stop if $|\bar{\lambda}^n| \geq \lambda_+$ for $n = 1, \dots, N_t$). 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 (37), any power demand is feasible, but in practice, we only want to perform generation following if we are within a certain error tolerance, ϵ^{reg}.

At optimality, 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, refereed 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 the next section.

D. 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 solu-

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}	1.0
Dual Feasibility Tolerance	ϵ^{dual}	1.0
Error Tolerance	ϵ^{reg}	10 kW
$\bar{\lambda}$ Limit	λ_+	50

TABLE II: ADMM Parameters

tion 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 [25] 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 (10-20% 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.

V. 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 II. For refrigerators and water heaters, $\alpha_x = 0$ indicating that the consumer is indifferent to the selection of a control trajectory. Thus, the TCL's objective function (36) 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 feasible 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.

$$x_\Sigma^k = \frac{1}{N_t} \sum_{n=1}^{N_t} \sum_{i=1}^{N} (x_i^n)^* \tag{40}$$

$$p_\Sigma^k = \frac{1}{N_t} \sum_{n=1}^{N_t} \sum_{i=1}^{N} \tilde{p}_i^n \tag{41}$$

where variables $x_\Sigma^k, p_\Sigma^k \in \mathbf{R}$, 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

$$x_\Delta^k = x_\Sigma^k - p_\Sigma^{k-1} \tag{42}$$

$$p_\Delta^k = p_\Sigma^k - p_\Sigma^{k-1} \tag{43}$$

Because x_i^* is not directly realizable, x_Δ^k is calculated relative to the previous probabilistic demand p_Σ^{k-1}.

For each time step k, we also present the minimum and maximum power demand that the population of TCLs *could*

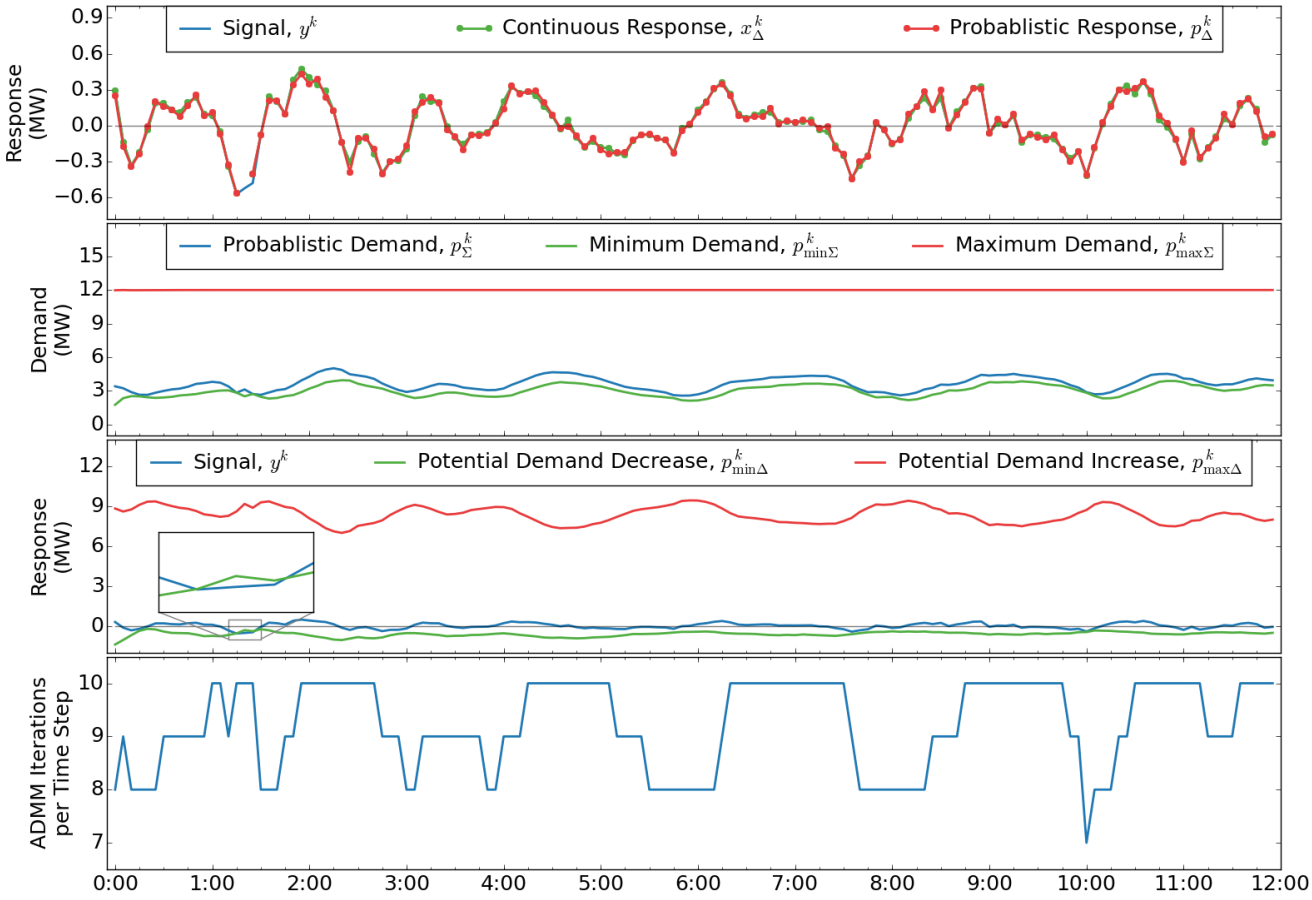

Fig. 8: Highly Homogeneous Population

have achieved given the set of feasible 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 $p_i^{\min} \in \mathbf{P}_i$ and $p_i^{\max} \in \mathbf{P}_i$, respectively. Therefore, the minimum and maximum mean power demand of the population is

$$p^k_{\min\Sigma} = \frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N}(p_i^n)^{\min} \tag{44}$$

$$p^k_{\max\Sigma} = \frac{1}{N_t}\sum_{n=1}^{N_t}\sum_{i=1}^{N}(p_i^n)^{\max} \tag{45}$$

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

$$p^k_{\min\Delta} = p^k_{\min\Sigma} - p^{k-1}_{\Sigma} \tag{46}$$

$$p^k_{\max\Delta} = p^k_{\max\Sigma} - p^{k-1}_{\Sigma} \tag{47}$$

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.

A. *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 $N = 40{,}000$ refrigerators with identical parameters (the mean of the parameter ranges in Table I). The peak feasible power demand of the population is 12 MW, significantly more than the maximum signal of 0.47 MW (1% of the signal in Figure 6). Because we have such a large population for this experiment, we have limited the number of ADMM iterations to 10.

Figure 8 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 6.61 kW and 26.62 kW, respectively. The algorithm only failed to converge to a solution within the error tolerance of 10 kW during one interval at 1:20 AM, resulting in a generation following

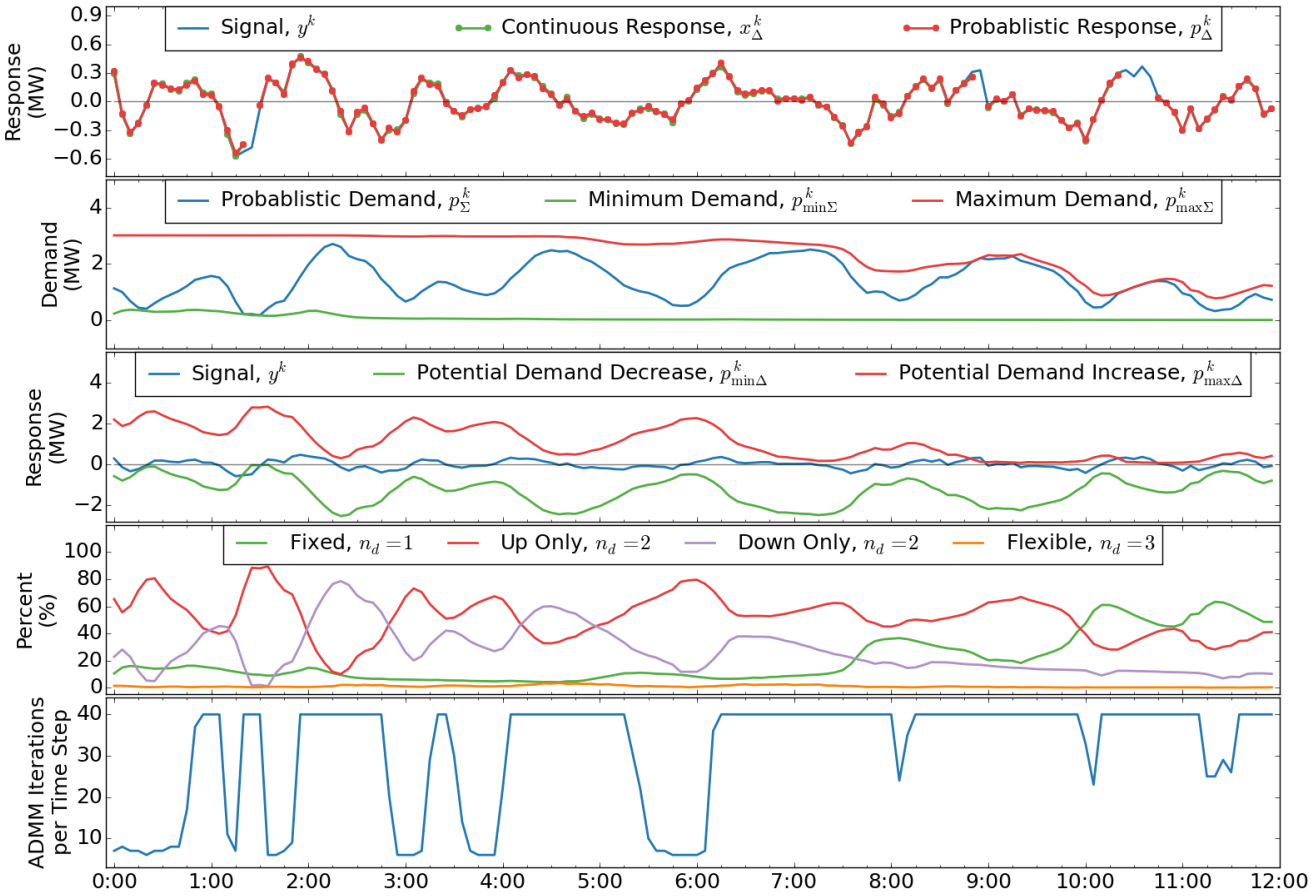

Fig. 9: Heterogeneous Population

success rate of 99.3% over the time period studied.

The second plot in Figure 8 shows the probabilistic p_Σ^k, the minimum $p_{\min\Sigma}^k$, and the maximum $p_{\max\Sigma}^k$ power demand of the population at each time interval. The third plot shows the corresponding minimum $p_{\min\Delta}^k$ and maximum $p_{\max\Delta}^k$ 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).

Examples of this behavior are shown at the 1:15, 1:20, and 1:25 intervals. In each case, the signal is slightly outside the feasible region, as shown in the third plot. In the fourth plot, which shows the number of ADMM iterations executed before stopping, we see that the ADMM algorithm hit the iterations limit of 10 for these time intervals. For the 1:15 and 1:25 intervals, the continuous response returned by the ADMM algorithm was within the error tolerance despite the fact that the stopping criteria were not met. For the 1:20 interval, however, the tolerance was not met and thus load following was not performed. In this case, each TCL returned to its default behavior.

B. 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 I. We have also raised the ADMM iterations limit to 40. The results from this study are presented in Figure 9 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 $p_{\max\Sigma}^k$ and the demand increase potential $p_{\max\Delta}^k$. The fourth plot shows the percentage of the population that is either fixed, flexible, or capable 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

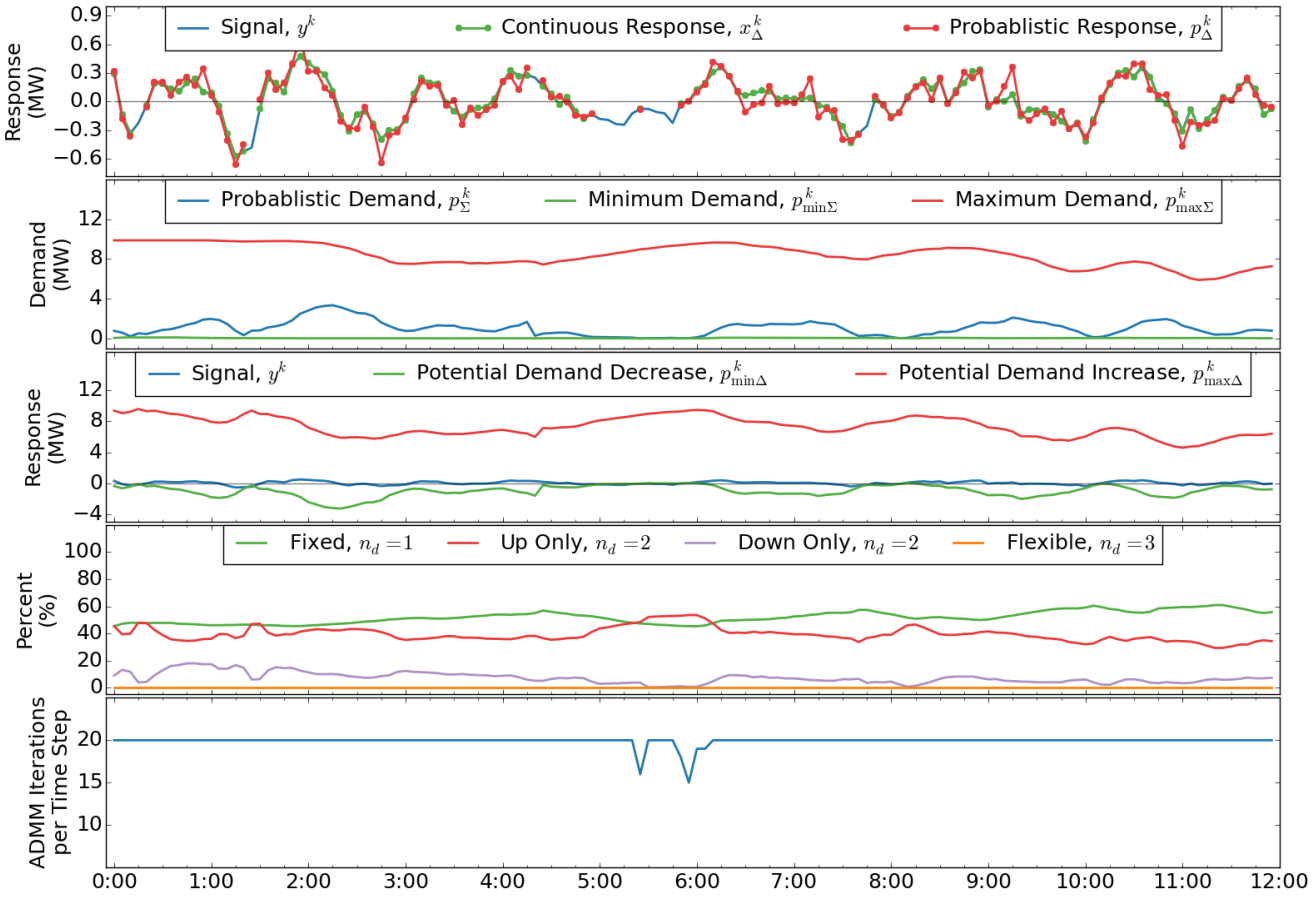

Fig. 10: Highly Heterogeneous Population

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.

C. 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 I. 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.

The results, presented in Figure 10, 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 criterion 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.

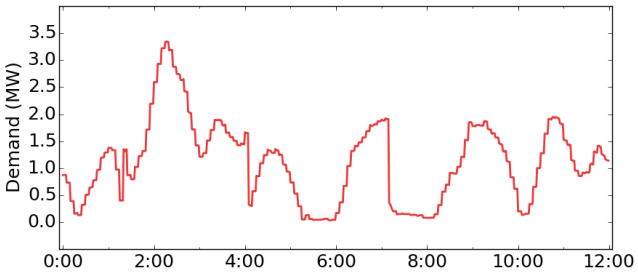

Fig. 12: Power Demand on 1 Minute Time Scale

The fourth plot in Figure 10 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.

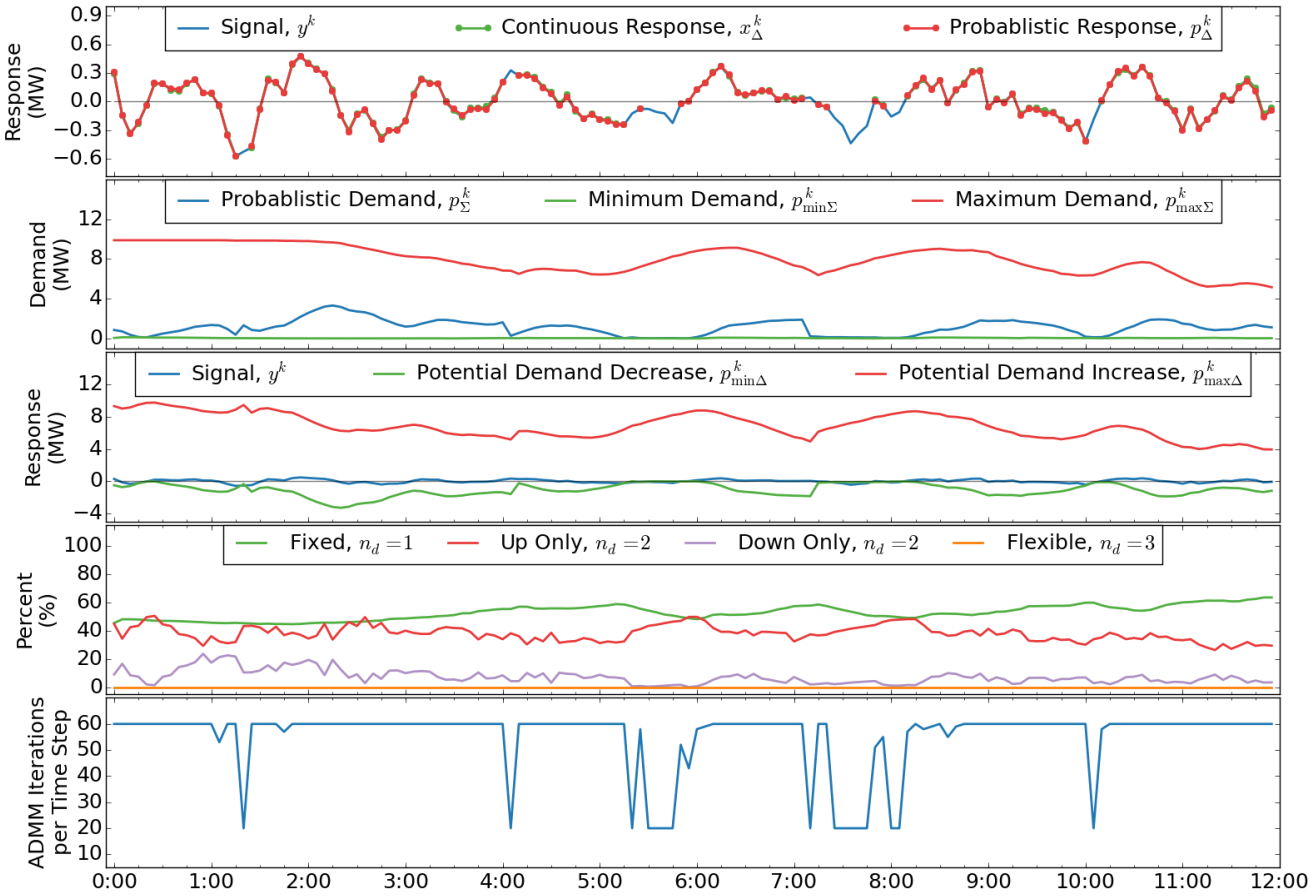

Fig. 11: Highly Heterogeneous Population with Divide and Conquer

D. Heterogeneous Population with Divide and Conquer

To address the error between the probabilistic response p^k_Δ and the signal r^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 11. While we have increased the total number of ADMM iterations at each time interval, the RMSEs of the 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 12. Because of the piece-wise 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.

VI. Conclusions

In this paper, we have presented an alternative control trajectory representation. This representation allows for the modelling of a TCL as a generalized assignment problem and fully recognizes the nonconvex 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.

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 renewable energy generation following. Using actual wind and solar generation forecasts, ambient temperature records, and TCL parameters, we have demonstrated how heterogeneous populations of TCLs can be optimized to perform fast power system services. For highly heterogeneous populations, we have shown that a divide and conquer approach can be employed to minimize the error between the probabilistic solution and the signal.

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

A. Notation

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

ℓ_1-norm:

$$\|x\|_1 = \sum_{i=1}^{N} |x_i| \tag{48}$$

ℓ_2-norm:

$$\|x\|_2 = \sqrt{\sum_{i=1}^{N} x_i^2} \tag{49}$$

Root Mean Squared Error:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{50}$$

Mean:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{51}$$

Sum:

$$\sum x_i = \sum_{i=1}^{N} x_i \tag{52}$$

Inner product:

$$\langle \lambda, x \rangle = \lambda^T x \tag{53}$$

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

REFERENCES

[1] Y. V. Makarov, C. Loutan, J. Ma, and P. De Mello, "Operational impacts of wind generation on california power systems," *Power Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 1039–1050, 2009.

[2] Z. Xu, J. Ostergaard, M. Togeby, and 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] B. J. Kirby, "Frequency regulation basics and trends," *ORNL/TM 2004/291, Oak Ridge National Laboratory*, December 2004.

[4] E. Hirst and B. Kirby, "Separating and measuring the regulation and load-following ancillary services," *Utilities Policy*, vol. 8, pp. 75–81, 1999.

[5] G. Strbac, "Demand side management: Benefits and challenges," *Energy Policy*, vol. 36, no. 12, pp. 4419–4426, 2008.

[6] D. S. Callaway, I. Hiskens *et al.*, "Achieving controllability of electric loads," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 184–199, 2011.

[7] J. L. Mathieu, M. Dyson, and 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*, 2012.

[8] R. Malhame and C.-Y. Chong, "Electric load model synthesis by diffusion approximation of a high-order hybrid-state stochastic system," *Automatic Control, IEEE Transactions on*, vol. 30, no. 9, pp. 854–860, 1985.

[9] S. Moura, V. Ruiz, and 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.

[10] A. Ghaffari, S. Moura, and M. Krstic, "Pde-based modeling, control, and stability analysis of heterogeneous thermostatically controlled load populations," *Journal of Dynamic Systems, Measurement, and Control*, 2015.

[11] 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*, vol. 50, no. 5, pp. 1389–1400, May 2009.

[12] S. Bashash and 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.

[13] C. Perfumo, E. Kofman, J. H. Braslavsky, and J. K. Ward, "Load management: Model-based control of aggregate power for populations of thermostatically controlled loads," *Energy Conversion and Management*, vol. 55, pp. 36 – 48, 2012.

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

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

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

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

[18] M. Liu and 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)*, Dec 2014, pp. 4850–4855.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, January 2011.

[20] S. Ihara and F. C. Schweppe, "Physically based modeling of cold load pickup," *IEEE Transactions on Power Apparatus and Systems*, vol. 100, no. 9, pp. 4142–4250, 1981.

[21] R. E. Mortensen and K. P. Haggerty, "A stochastic computer model for heating and cooling loads." *IEEE Transactions on Power Systems*, vol. 3, no. 3, pp. 1213–1219, Aug 1998.

[22] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *UCLA Computational and Applied Mathematics Report*, pp. 12–35, 2012.
[23] Weather Underground Web Service and API. [Online]. Available: wunderground.com/weather/api/
[24] California independent system operator. [Online]. Available: oasis.caiso.com
[25] E. J. Cands, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.