Applying Networked Estimation and Control Algorithms to Address Communication Bandwidth Limitations and Latencies in Demand Response

Gregory S. Ledva*, Evangelos Vrettos†, Silvia Mastellone‡, Göran Andersson† and Johanna L. Mathieu*
*Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, Michigan, USA
Email: gsledev AT umich.edu
†Power Systems Laboratory, ETH Zurich, Zurich, Switzerland
‡ABB Corporate Research Center, Baden, Switzerland

Abstract—Demand response can provide services to the power network; however, coordination of spatially distributed demand response resources generally requires coping with imperfect communication networks. This work investigates methods to manage communication constraints (e.g., delays and bandwidth limitations), faced by demand response aggregators who manipulate the on/off modes of residential thermostatically controlled loads (TCLs). We present two model predictive control (MPC) algorithms that exploit a priori knowledge of delay statistics. We also present three Kalman filter-based state estimation methods that handle measurements with heterogeneous delays that are known a posteriori. We simulate the closed loop system to quantify the error while the system tracks simplified power system signals of various frequencies. We find that the MPC algorithm incorporating the full delay distribution, versus only the mean delay, reduces the average tracking error 39%. Also, incorporating individual TCL models, identified on-line, within the state estimator versus only using a TCL aggregation model reduces the average estimation error 19%.

Keywords—Demand response; networked control; latency; model predictive control; Kalman filter; parameter identification

I. INTRODUCTION

Fluctuating renewable energy sources increase the need for power system services such as frequency regulation and load following [1]. Traditionally, these services are provided by generators; however, there is increasing interest in using flexible electric loads to provide these services [2] with the hope that loads can do so more effectively, at lower cost, and/or with less environmental impact than conventional generators. Residential thermostatically controlled loads (TCLs) such as air conditioners, space heaters, electric water heaters, and refrigerators are particularly well-suited to demand response because the timing of their power consumption is inherently flexible. However, coordinating the behavior of a large aggregation of TCLs is challenging because each TCL exhibits stochastic hybrid dynamics, i.e., each TCL has a continuous temperature state and a discrete on/off mode [3], [4]. Many recent papers, e.g., [5]–[11], have proposed approaches to model and control TCL aggregations to provide power system services while minimizing the impact on the consumer.

Controlling TCLs to provide regulation or load following requires communication between the spatially distributed, controllable loads and the aggregator. The use of a shared communication network, rather than dedicated point-to-point infrastructure, allows a flexible network architecture while reducing the installation and maintenance costs [12]. However, these networks are imperfect and introduce challenges such as bandwidth limitations, latencies, lost messages, and out-of-sequence message arrivals [13]. Bandwidth limitations exist due to the limited capacity of the communication channels and can lead to increased latencies and message loss as more capacity is used. Latencies, i.e., delays, mainly arise from the physical travel time and the queuing time in transmitting a message, which increases with network traffic. Lost messages arise due to excessive transmission latencies or failed transmissions. Finally, out-of-sequence message arrival arises due to time-varying latencies that can exceed one sampling interval.

In this paper, we develop methods to manage communication bandwidth limitations and latencies within demand response. In practice, curtailment signals were shown to experience delays averaging about 30 seconds [14] to over a minute [15]. We assume an aggregator provides services to the power system by manipulating TCL on/off modes, ensuring its actions are non-disruptive to consumers. However, we assume that control signals to the loads, state measurements from the loads, and TCL aggregate power measurements may be delayed or lost, and that state measurements are infrequent.

Numerous approaches exist for addressing delays and lost messages in communication systems [12], [13], [16]–[19]. However, to the best of our knowledge, these methods have not yet been applied to demand response, though [20] investigated the effect of delays in coordinating distributed batteries and [21] investigated the effect of lost messages (but not delays) on optimal load scheduling. Ref. [13] classified recent work on methods to manage communication...
delays into two frameworks: the robustness framework and the adaptation framework. The robustness framework applies methods from conventional time-delay systems to design a controller that is robust over the delays, but it does not use on-line delay information and is generally conservative. Alternatively, the adaptation framework uses the communication network’s capabilities and on-line information about the realized delays. Here, we propose methods consistent with the adaptation framework.

Recent work has also investigated the effect of bandwidth limitations on TCL tracking control performance. When state information is infrequent or unavailable, state estimators are used together with measurements/estimates of the aggregate power consumption of the TCL aggregation and models to estimate states [22]–[26]. However, none of this work has investigated the performance of these algorithms in systems with communications delays.

Our contribution is to modify networked state estimation and optimal control algorithms to mitigate communication limitations in demand response, and to compare their performance, identifying the most promising approaches. In addition, we conduct case studies to quantify how the performance of the estimators/controllers varies as a function of the mean delay and the frequency of the signal that the TCL aggregation tries to follow.

The remainder of this paper is organized as follows. Section II describes the communication and control setup. Section III presents the TCL models, Section IV the control algorithms, and Section V the state estimation algorithms. Section VI describes the case study scenarios and Section VII presents their results. Finally, Section VIII concludes.

II. COMMUNICATION AND CONTROL SETUP

We consider the problem of controlling the power consumption of an aggregation of hundreds to thousands of TCLs given a communication system with bandwidth limitations, communication delays, lost messages, and out-of-sequence arrivals. Figure 1 shows a block diagram of the control and communication system. The plant consists of a set of independent TCLs \( \mathcal{N}_{TCL} \), each indexed by \( i \), and each affected by an independent disturbance \( e^i_t \) at each time step \( t \in \mathcal{T} \). An aggregator, consisting of a state estimator and controller, broadcasts control inputs \( u^i_t \) to the plant which attempts to track a power system signal \( P^i_{\text{ref}} \). As in [22], the input is a vector consisting of on/off switching probabilities and TCLs respond to the element of \( u^i_t \) that corresponds to their current state, consisting of their (continuous) internal temperature \( \Theta^i_t \) and (discrete) on/off mode \( M^i_t \). The model and control input are more fully described in Section III. Unlike [22], the input travels through an imperfect communication network, leading to input delays \( \tau^i_{u,t} \), which are assumed i.i.d. and independent for each TCL. The power consumption of TCL \( i \) is \( P^i_t \).

As in [24], [25], we assume the following information is transmitted to the aggregator: i) frequent (e.g., every 2 seconds) noisy estimates of the aggregate power consumption of the TCL aggregation, \( P^\text{agg}_t = \sum_i P^i_t + P^\text{noise}_t \), and ii) infrequent (e.g., every 15 minutes) perfect measurements of each TCL’s state trajectory since the previous transmission, \( \Theta^i_t \) and \( M^i_t \), where \( \Theta^i_t \) is the temperature trajectory and \( M^i_t \) is the on/off mode trajectory. Noisy TCL aggregate power estimates can be computed by subtracting a forecast of the uncontrollable load served by the substation from the measured power consumption at the substation [22] and TCL state measurements can be obtained from smart meters and/or home energy management systems that communicate with both the TCLs and the aggregator. Let \( T^S \) and \( T^S_{\text{c}} \) denote the sets of state measurement sampling times and transmission times, respectively. Unlike [24], [25], we assume that measurements/estimates pass through imperfect communication networks leading to aggregate power measurement delays \( \tau^i_{x,t} \), which are assumed i.i.d., and TCL state estimate delays \( \tau^i_{\Theta,t} \), which are assumed i.i.d. in time and among the individual TCLs.

We assume all information is time-stamped and clocks are synchronous (clock drift is not considered), and so past delays and delay statistics are known. Delay realizations are generated from time-invariant, discrete, log-normal distributions \( \tau = \lfloor e^d \rfloor \) where \( d \) is sampled from a normal distribution with mean \( \mu \) and variance \( \sigma^2 \). The operator \( \lfloor \cdot \rfloor \) rounds its contents down to the nearest integer. We also assume that multiple values (e.g., control vectors, state measurement time series) can be transmitted within one message [13].

The aggregator uses both aggregate system models and individual TCL models for state estimation and control. Similar to [22], we use state estimation algorithms based on Kalman filtering; however, the approaches in [22] do not explicitly handle delays. Therefore, we propose three state estimation methods that use measurements with known delays due to time-stamping to estimate the aggregate system state \( \hat{x}_t \). Input delays cause uncertainty in the realized inputs, and so the estimator uses an estimate of the expected input \( \bar{u}_t \). We also develop two controllers that use knowledge of delay statistics. Like [8], we use model predictive control
(MPC), but the approaches in [8] do not explicitly handle delays. Since control inputs are delayed, we transmit the entire horizon of open-loop control inputs rather than sending only the first input [13].

III. TCL MODELING

We detail two methods for modeling TCLs. Section III-A describes a stochastic hybrid system model of the thermal dynamics of an individual TCL, which was developed in [3], [4], [27], [28]. This model is used to represent each TCL within the plant, and is also used within one proposed estimation algorithm. Section III-B describes a linear time invariant model of a TCL aggregation, which was developed in [8], [22] and is similar to the models in [7], [10], [29]–[32]. This model is more amenable to estimator/controller design, and it is used within all proposed estimation and control algorithms.

A. Individual TCL Model

The evolution of a cooling TCL’s internal temperature and on/off mode can be modeled as [3], [4], [27], [28]

\[ \theta_{t+1} = a^i \theta_t + (1-a^i)(\theta_{t+1}^{a^i} - m_t \theta_{t+1}^{b^i}) + e_t \]

\[ m_t^{i+1} = \begin{cases} 1 & \theta_{t+1}^{a^i} > \theta_{t+1}^{b^i} / 2 \\ m_t^{i-1} & \text{otherwise} \end{cases} \]

where \( a^i = \exp \left( \frac{-\Delta t}{\alpha_{a^i} \theta_{t+1}^{a^i}} \right) \), \( \theta_{t+1}^{b^i} = \alpha_{b^i} \theta_{t+1}^{b^i} \), and the remaining parameters are defined in Table I. Note that \( m_t^0 = 1 \) when TCL \( t \) is drawing power (‘on’) and \( m_t^0 = 0 \) when it is in standby (‘off’). Also note that three-state hybrid models that include the temperature of the internal thermal mass more accurately capture TCL dynamics [33], [34]; however, the two-state model has been validated against real TCL populations [35] and is commonly used in the literature [6]–[8], [11].

B. Aggregate TCL Population Model

A Markov model that captures the dynamics of a TCL aggregation was developed in [8], [22]. Each TCL within an aggregation maps its dead-band to a single, normalized dead-band that is divided into \( \frac{N_{\text{bin}}}{2} \) temperature intervals. Two state bins are defined for each interval – one associated with the on mode and one with the off mode. The resulting state vector \( x_t \in \mathbb{R}^{N_{\text{bin}}} \) contains the fraction of TCLs within each discrete state bin where the first \( \frac{N_{\text{bin}}}{2} \) elements correspond to the off mode and the remaining correspond to the on mode. Assuming \( \theta^{a^i} \) is constant, the system dynamics are

\[ x_{t+1} = Ax_t + Bu_t + w_t \]

where the state transition matrix \( A \in \mathbb{R}^{N_{\text{bin}} \times N_{\text{bin}}} \) is a transposed Markov Transition Matrix that captures the probability of TCLs moving from one state bin to another. It can be constructed in several ways [8]; here we use historical state measurements and compute the transition probabilities directly. The input vector \( u_t \in \mathbb{R}^{N_{\text{bin}}} \) specifies the fraction of the total number of TCLs in the aggregation that should switch in each temperature interval. Positive elements switch TCLs on and negative switch TCLs off. Matrix \( B \) ensures TCLs transition between on/off modes within the same temperature interval. Since individual TCLs cannot act on \( u_t \) without knowing the states of all other TCLs, we convert \( u_t \) into a vector of switch probabilities by dividing each input element by its corresponding state estimate element. The switch probabilities are broadcast to the TCLs, and each TCL determines which probability to act upon based on its current state. TCLs then carry out random processes locally to determine whether or not to switch. TCLs outside of their dead-band ignore the control signal, ensuring the control is non-disruptive to the consumer. Variable \( w_t \) represents process noise, including plant-model mismatch.

The output \( y_t \) is dependent on the available measurements. When only the aggregate power is available then \( y_t \in \mathbb{R} \), and when the TCL states and the aggregate power are available then \( y_t \in \mathbb{R}^{N_{\text{bin}}+1} \). The output equation is

\[ y_t = \begin{cases} C P x_t + v^P_t & \text{if } t \in T^S \\ [(C S)^T (C P)^T] x_t + [(v^P_t)^T (v^S_t)^T] & \text{if } t \in T^S \end{cases} \]

where \( C^P = P_{\text{on}}^{n_{\text{TCL}}} \) is an \( N_{\text{bin}} \times 1 \) identity matrix, \( v^P_t \in \mathbb{R} \) is the noise associated with the aggregate power measurement, and \( v^S_t \in \mathbb{R}^{N_{\text{bin}}} \) is the noise associated with the state measurements, which arises in the calculation of the aggregate state from the noise-free, individual TCL states due to measurement delays. Variable \( P_{\text{on}} \) is the average power demand of the TCLs in the on mode, calculated during steady state operation, and \( n_{\text{TCL}} \) denotes the number of TCLs in the population. We also denote the power and state outputs within (3) as \( y^P_t = C^P x_t + v^P_t \) and \( y^S_t = C^S x_t + v^S_t \), respectively. Note that [33] developed a Markov modeling approach capturing the aggregate system dynamics of TCLs represented with three-state individual TCL models. Using this approach, the methods presented within this paper could be extended to systems in which three-state models are more appropriate.

IV. CONTROL ALGORITHMS

We develop MPC-based controllers that rely on the aggregate TCL model. First, we present the core algorithm.
Then, we develop two methods to incorporate knowledge of the input delay statistics: one that uses only the average delay and one that uses the full knowledge of delay statistics.

A. Core MPC Formulation

A modified MPC scheme transmits entire horizons of open-loop inputs [13]. We assume each TCL: i) stores all input horizons that arrive; ii) selects an input, corresponding to the current time-step, from the horizon that has arrived most recently; and iii) applies a switching probability of zero when no valid input is available.

The MPC prediction horizon length \( N^{\text{MPC}} \) is chosen such that the probability of an input delay exceeding the horizon length is less than a threshold \( p^{\text{max}} \). At each calculation time \( t \), the controller solves

\[
\min_{u, \delta} \sum_{k=t}^{t+N^{\text{MPC}}-1} c^y (y^\text{err}_k)^2 + c^u (u_k^T u_k) + c^\delta (\delta_{k+1}^+ - \delta_{k+1}^-)
\]

s.t.

\[
x_k = Ax_k + Bu_k
\]

\[
y^\text{err}_k = P^\text{ref}_{k+1} - C^p x_k
\]

\[
0 \leq \delta_{k+1}^- \leq x_k \leq 1 + \delta_{k+1}^+
\]

\[
-1 \leq u_k \leq 1
\]

\[
x_t = \hat{x}_t
\]

where \( c^y, c^u, \) and \( c^\delta \) are cost coefficients, and \( \delta^-_{k+1}, \delta^+_{k+1} \) enable soft bounds for \( x_k \) in (9). Equality constraint (5) includes the TCL population’s aggregate dynamics, the tracking error \( y^\text{err}_k \) is defined in (6), and the algorithm is initialized in (10). The soft bounds allow a feasible solution regardless of the initial value, \( \hat{x}_t \), provided by the unconstrained estimator, and \( c^\delta \) is set so the controller does not actively violate the soft bounds. We assume the controller has no information about future values of \( P^\text{ref}_{k} \) and uses a persistent forecast over the horizon.

B. Controller 1: Including the Average Input Delay

A standard approach within stochastic programming is to use the expected value of a stochastic variable. We use state-space augmentation to include the average delay within the MPC’s model of the system dynamics [36]. The original model of the system dynamics and tracking error within the MPC formulation are

\[
x_{k+1} = Ax_k + Bu_k
\]

\[
y^\text{err}_{k+1} = P^\text{ref}_{k+1} - C^p x_{k+1}
\]

where \( u_{k-} \) indicates the input is delayed by \( T^\text{err} \), the average delay. We can write this as an augmented system

\[
x_{k+1} = Ax_k + Bu_k
\]

\[
y^\text{err}_{k+1} = P^\text{ref}_{k+1} - C^p x_{k+1}
\]

\[\text{Figure 2. The input selection process at the TCLs for time-step } k = t.\]

where \( A, B, C^p \), and \( x_k \) are the augmented matrices and vectors that include the mean input delay within the dynamics. The dynamics and output in (13) and (14) replace (5) and (6) within the MPC algorithm in Section IV-A. The MPC algorithm now determines new inputs starting from \( k = t \) to \( t + N^{\text{MPC}} - 1 \) taking into account previously broadcast inputs from \( k = t - T^\text{err} \) to \( t - 1 \) that have not influenced the system due to their expected delay.

The predetermined inputs from \( k = t - T^\text{err} \) to \( t - 1 \) within the augmented state vector and the newly generated inputs from \( k = t \) to \( t + N^{\text{MPC}} - 1 \) are broadcast to the TCLs, and if the expected delay occurs, the first newly generated input at \( k = t \) is used.

C. Controller 2: Including a Finite Set of Delayed Inputs

We now use the known probability distribution of delays to form an expected input from the possible inputs at a given time-step. The following algorithm was adapted from [37], which used a plant composed of a single component and estimated the implemented control sequence using interacting multiple models. We adapt this algorithm to account for our plant: a TCL aggregation consisting of numerous independent components. Rather than estimating the individual input sequence for each TCL, we estimate the expected input of the population \( u_k \) as a linear combination of the possible inputs at time-step \( k \) weighted by their probability of being implemented

\[
\hat{u}_k = (P^T_k U_k)^T. \tag{15}
\]

The value \( U_k \in \mathbb{R}^{N^{\text{MPC}} \times N^{\text{MPC}}} \) indicates the possible inputs at time-step \( k \), and \( P_k \in \mathbb{R}^{N^{\text{MPC}}} \) is each input’s probability being implemented.

Figure 2 depicts sets of possible inputs, \( U_k \), where \( N^{\text{MPC}} = 3 \); the \( U_t \) values on the left side of the figure depict the open-loop input sequence broadcast from the controller at time \( t; u_{k|t} \) is an input corresponding to time-step \( k \) within \( U_t \). Each \( U_k \) is composed of inputs corresponding to time-step \( k \) that were broadcast at \( t = k - N^{\text{MPC}} + 1 \) to \( k \). For time-step \( k = t \), this corresponds to

\[
U_k = [u_{k|t} \ u_{k|t-1} \ \cdots \ u_{k|t-N^{\text{MPC}}+1}]^T. \tag{16}
\]

Each element of the probability vector \( P_k \) is the probability that a TCL uses the corresponding input within \( U_k \), i.e., for time-step \( k = t \)

\[
P_k = [p_{k|t} \ p_{k|t-1} \ \cdots \ p_{k|t-N^{\text{MPC}}+1}]^T. \tag{17}
\]
At each $k$, TCLs implement the most recently broadcast input that has arrived, and so we construct each probability within $P_k$ from two events: i) the corresponding input within $U_k$ must have arrived by $k$, and ii) more recently broadcast inputs must not have arrived. As a result, each element within $P_k$ consists of two independent probabilities $p_{kl|t} = p_{k_1|t} p_{k_2|t}$ where $p_{k_1|t}$ is the probability of the first event, and $p_{k_2|t}$ is the probability of the second event.

The first event requires that an input $u_{k|t}$ broadcast in $U_t$ for time-step $k$ has a delay $\tau_k$ less than or equal to $k - t$

$$p_{k_1|t} = p(\tau_k \leq k - t).$$  \hfill (18)

Let $t^*$ denote broadcast times between $t$ and $k$. The second event requires that delays $\tau_k$ for inputs broadcast at all time-steps $t^*$ greater than $k - t^*$

$$p_{k_2|t} = \prod_{t < t^* < k} p(\tau_k > k - t^*).$$  \hfill (19)

Including (15) within the MPC formulation results in a $U_k$ for each time-step within the horizon, for example $U_{k=1}$, $U_{k=2}$, and $U_{k=2}$ within Fig. 2. The controller uses input sequences that have previously been broadcast ($U_{t-1}, U_{t-2}$, etc.) to compute optimal current and future inputs ($U_t, U_{t+1}$, etc.), but only broadcasts the current input sequence $U_t$.

In summary, we modify the core MPC formulation in Section IV as follows: (8) is modified to apply to each input within $U_{k,t}$, (15) is added as an additional constraint, and $u_{k,t}$ within (5) is replaced by $\hat{a}_{k,t}$.

The above calculation of $P_k$ assume independent delays, but the process can be modified to handle correlated delays. Assuming a Markov Transition Matrix captures the correlation within a discrete, finite set of delays and an additional component estimates the current delay distribution, then the event probabilities can be computed from the estimated delay distribution, the delay correlations, and the required delays for given input elements to be implemented.

V. STATE ESTIMATION ALGORITHMS

A networked, multi-sensor, Kalman filter was developed in [38] that accounts for delayed, lost, and out-of-sequence measurement arrivals. We use this Kalman filter; however, in contrast to [38], we use an estimated rather than a known input. First, we assume that all TCL states sampled at a given time-step experience the same delay, which allows us to use a known measurement noise covariance associated with the constructed aggregate state measurement. We then consider heterogeneous delays and propose three extensions to the core formulation.

A. Core Kalman Filter Formulation

The Kalman filter combines the power and aggregate state observations into a single observation. Creating a single observation requires higher dimension matrix operations within the Kalman filter but allows the optimal incorporation of separate measurements into the a posteriori state estimate. Due to the recursive nature of the Kalman filter, previously calculated estimates and error covariances must be updated within a horizon $N_k^{P}$ to fully include the delayed measurements into the present estimate. The oldest newly arrived measurement determines the horizon length, i.e., $N_k^{P}$ is time-varying. The Kalman filter incorporating delayed measurements updates previous estimates within $N_k^{P}$ whereas a standard Kalman filter in the delay-free case does not. For this reason, the modified Kalman filter requires some memory whereas a standard Kalman filter does not.

Indicators for the aggregate state and power measurements $\gamma_k^{S} \in \{0, 1\}$ and $\gamma_k^{P} \in \{0, 1\}$ describe whether the corresponding measurement, sampled at time-step $k$, is available at the estimator during the Kalman filter computation at time $t$. The arrival indicator is 0 if the measurement has not arrived and 1 otherwise. Within the Kalman filter, the aggregate demand observation, the aggregate state observations, and their respective noises are noted as $y_k^{P}|t$, $y_k^{S}|t$, $v_k^{P}|t$, and $v_k^{S}|t$. An observation is known once it arrives at the estimator, otherwise a dummy value is used for an unarrived observation within the Kalman filter calculation at time $t$. Using the arrival indicators within the augmented observation ensures the dummy value is neglected within the Kalman filter calculation. Finally, the measurement delays $\tau_k^{S}$ and $\tau_k^{P}$ are considered infinite until the measurement arrives, and then the delay is known.

We define the augmented observer output, output matrix, measurement noise, and noise covariance matrix as

$$y_k|t = \begin{bmatrix} \gamma_k^{S} \ y_k^{S} \end{bmatrix}^T \begin{bmatrix} \gamma_k^{P} \ y_k^{P} \end{bmatrix}^T$$ \hfill (20)

$$C_k|t = \begin{bmatrix} \gamma_k^{S} \ C_k^{S} \ C_k^{P} \end{bmatrix}^T$$ \hfill (21)

$$u_k|t = \begin{bmatrix} \gamma_k^{S} \ v_k^{S} \end{bmatrix}^T \begin{bmatrix} \gamma_k^{P} \ v_k^{P} \end{bmatrix}^T$$ \hfill (22)

$$R_k|t = \text{diag} \begin{bmatrix} \gamma_k^{S} \ R_k^{S} \ gamma_k^{P} \ R_k^{P} \end{bmatrix}^T$$ \hfill (23)

where $R^P$ and $R^S$ are the power and state measurement’s noise covariances. The arrival indicators ensure rows corresponding to unarrived measurements are zero within the augmented observation values. We remove these rows to reduce the dimension of the matrices, and denote the resulting matrices as $\tilde{y}_k|t$, $\tilde{C}_k|t$, $\tilde{v}_k|t$, and $\tilde{R}_k|t$.

The Kalman filter is initialized at each time $t$ based on the estimates from the previous computation that are unaffected by the newly arrived measurements

$$\tilde{x}_{t-N_k^{P}+1|t} = \tilde{x}_{t-N_k^{P}+1|t-1}$$ \hfill (24)

$$H_{t-N_k^{P}+1|t} = H_{t-N_k^{P}+1|t-1}$$ \hfill (25)

where $H_{k|t}$ corresponds to the estimate of the error covariance matrix. The aggregate demand dynamics in (2) produce
the a priori estimate according to the standard Kalman filter
\[ \hat{x}_{k|t} = A\hat{x}_{k-1|t} + Bu_{k-1|t} \]  
(26)
\[ H_{k|t} = AH_{k-1|t}A^T + Q \]  
(27)
where \( Q \) corresponds to the covariance of the underlying system’s process noise. We determine \( \hat{u}_{k|t} \) based on the MPC implementation: it equals the actuated input assuming the mean delay is realized when using the method in Section IV-B, and it equals (15) when using the method in Section IV-C.

The Kalman gain is calculated at each time-step. The observation values, which depend on arrival indicators, include only the received measurements in the update
\[ K_{k|t} = H_{k|t}^T(\bar{C}_{k|t}^T H_{k|t} \bar{C}_{k|t} + \bar{R}_t)^{\dagger} \]  
(28)
where \( \dagger \) denotes the pseudo-inverse to include the general case of a non-square argument. Finally, the calculated gain and available measurements update the a posteriori estimates
\[ \hat{x}_{k|t} = \hat{x}_{k|t} + K_{k|t}(y_{k|t} - \bar{C}_{k|t}\hat{x}_{k|t}) \]  
(29)
\[ H_{k|t} = H_{k|t} - K_{k|t}\bar{C}_{k|t} H_{k|t} \]  
(30)
The horizon \( k = t - N^f + 1 \) to \( t \) applies to (26)-(30). The Kalman filter is optimal for zero-mean, Gaussian process and measurement noise, but the plant-model mismatch within our underlying system is non-Gaussian, resulting in a non-Gaussian process noise and a suboptimal filter.

The assumption of synchronous TCL state measurement arrivals is relaxed in the following sections. In each case, the Kalman filters incorporate the delayed aggregate power measurements according to the scheme above.

B. Estimator 1: The Sampling Window Method

The sampling window method benefits from algorithmic simplicity, but it only incorporates the most recent TCL state measurements. It also requires that TCLs are sampled at the same time to allow the construction of an aggregate state measurement that corresponds to a given time-step.

The state estimator uses a sampling window of length \( \tau_{\text{agg}} \) and collects individual TCL measurements from time \( t \) until \( t + \tau_{\text{agg}} \). It constructs an aggregate state measurement from the arrived individual TCL state measurements once the sampling window expires or all measurements arrive. This, effectively, artificially delays the arrived TCL state measurements to \( \tau_{\text{agg}} \). Measurements arriving after this sampling window are discarded, which is justified by only needing partial TCL information for state estimation [22].

The accuracy of the constructed aggregate state measurement can degrade if too few TCL state measurements are used. As a result, the sampling window length is set so that a high percentage of the TCL state measurements arrive within \( \tau_{\text{agg}} \), allowing accurate aggregate state measurements to be constructed, and allowing the aggregate state’s measurement noise to be considered zero, i.e., that \( R_S = 0 \). This delayed aggregate state measurement is incorporated within the Kalman filter in (24)-(30).

C. Estimator 2: Parallel Filter Method

The parallel filter method incorporates individual TCL state measurements into the aggregate state estimate as they arrive and accepts TCL state measurements sampled at different times. Drawbacks of this algorithm include that it only uses the most recent TCL measurements and it is more computationally expensive. We ran a benchmark case with an average delay of 20 seconds and standard deviation of 0.5. Its runtime was under 15 minutes for a 3 hour simulation, which is 14.07 times longer than that of the sampling window method.

This method uses one copy of the Kalman filter from Section V for each of the \( n_{\text{TCL}} \) TCLs and combines the individual estimates into an aggregate state estimate. This is done by averaging the estimates from the Kalman filters
\[ \hat{x}_t = \frac{1}{n_{\text{TCL}}} \sum_{i=1}^{n_{\text{TCL}}} \hat{x}^i_t \]  
(31)
where the elements of \( \hat{x}^i_t \) are now interpreted as probabilities that TCL \( i \) is within a specific bin. As measurements arrive from individual TCLs, the corresponding Kalman filters immediately incorporate these, influencing the resulting aggregate estimate. A TCL’s state bin is known exactly at times when its measurement is available, and its state estimate evolves according to the dynamics of the aggregate model until a new TCL measurement arrives. The aggregate state is a linear combination of the individual TCL’s bin-based states, and the Kalman filter is a linear filter. This implies the parallel filter method and the sampling window method should perform identically in the absence of delays, and the results in Section VII-B support this claim.

D. Estimator 3: Identified Individual Model Method

The identified individual model method accepts histories of TCL measurements and immediately uses delayed TCL state measurements (which may be sampled at different times). The method uses both the aggregate TCL model and individual, identified TCL models. The computational complexity of the Kalman filter itself is similar to that of
the sampling window method, but additional computation is needed for the parameter identification. This averages 0.007 seconds per TCL when provided 450 state measurements. The framework of this method can be seen in Fig. 3.

TCL state histories $\Theta_t^i$ and $M_t^i$ are used within a nonlinear least squares problem relying on the thermal dynamics in (1) to identify $a_{t+1}^{k,i}$ and $a_{t+1}^{c,i}$. The estimator uses the identified parameters within (1), neglecting the disturbance term, to simulate individual TCLs, with the most recent TCL state measurement reinitializing the corresponding TCL model. Finally, identified models generate aggregate state predictions at each time-step, which the Kalman filter treats as aggregate state measurements $y_{t}^S$.

The prediction’s noise covariance, $R^S$ is formed by comparing previous aggregate state predictions against the received measurements. The prediction noise may be time-varying and may not be zero-mean and Gaussian, which results in a suboptimal filter.

VI. CASE STUDY SET-UP

Before quantifying the performance of each algorithm, we present the simulation parameters and the performance metrics used to evaluate the simulations.

A. Simulation Parameters

We simulate 10,000 air conditioners with time-invariant parameters selected from uniform distributions with ranges provided in Table I. The disturbance $c_t^\ell$ is sampled from a zero-mean, Gaussian distribution with standard deviation of $5 \times 10^{-4}$. The number of state bins $N_{\text{bin}}$ is set to ten. The noise on the power measurement, $P_{t}^{\text{noise}}$, is sampled from a zero-mean, Gaussian distribution. The standard deviation is set to 5% of the total distribution substation load, and as in [22], we assume the load is 17MV A per substation with 1,000 TCLs per substation. The corresponding power measurement noise is scaled to the total load of ten substations.

Within the control algorithms, the sampling window $\tau_{\text{agg}}$ is set to 3 minutes to ensure a suitable number of measurements arrive. The delay threshold, $\tau_{\text{max}}^\text{c}$, of the MPC horizon is set to 0.999, and cost coefficients $c^p, c^c = 1$ with $c^p = 1.01$. The time-invariant TCL parameters are identified before the simulation using data from one TCL state trajectory transmission collected during forced TCL operation.

Discrete time-steps of $\Delta t = 2$ seconds are used, and the number of time-steps $\tau_{\text{steps}}$ is set to 900 when testing a controller and 5400 when testing an estimator, corresponding to durations of 30 minutes and 3 hours respectively. The substation power and TCL states are sampled at every time-step while the TCL state trajectories are transmitted to the central controller at 15 minute intervals.

Section VII uses combinations of delay distributions and power system signals (referred to as “reference signals”) to characterize the control algorithms’ performance across a range of cases. We use delay distributions with mean values, $\mu$, ranging from 0 to 20 seconds at 2 second increments. The corresponding standard deviation, $\sigma$, is 0 in the case delay-free case (i.e., $\mu = 0$) and 0.5 otherwise. We use sinusoidal reference signals as simplified power system signals to characterize the algorithms’ performance as a function of the ratio between the reference signal frequency and average delay. The sinusoid’s period ranges from 20 seconds to 200 seconds at 20 second increments. The amplitude of the sine wave is set to 20% of the steady state TCL aggregation demand with an average reference value equal to the steady state TCL aggregation demand.

B. Performance Metrics

We use several metrics to determine the relative performance of the algorithms: the root mean square error (RMSE), a relative amplitude, and a time lag. The RMSE is

$$P_{\text{RMSE}} = \sqrt{ \frac{1}{\tau_{\text{steps}}} \sum_{t \in T} \Delta P_t^2 } \quad (32)$$

where $\Delta P_t$ indicates a generic error. The aggregate power tracking error, $\Delta P_t = P_{t}^{S} - P_{t}^{\text{ref}}$, is used when testing MPC algorithms where the total TCL demand is $P_{t}^{S} = \sum_i P_{t}^{i}$ with $i \in N_{\text{TCL}}$. The aggregate power estimation error, $\Delta P_t = P_{t}^{\hat{S}} - P_{t}^{S}$, is used when testing state estimation algorithms. The RMSE value indicates the overall effectiveness of following the reference, but we develop two additional metrics to quantify the type of error.

The relative amplitude, $P_{\text{amp}}$, is the RMS value of the estimated or realized TCL demand divided by the RMS value of the reference signal where the mean is subtracted from both signals. A value of 1 indicates the desired RMS content was provided exactly whereas a value of 0.5 indicates the realized signal only produces half of the desired power.

Last, we define a lag $r_{\text{lag}}$ for the controller simulations that is the time that delay maximizes the cross correlation between the desired and realized total TCL power demand

$$r_{\text{lag}} = \max_{\tau} \left( r_{\tau}^{\text{ref}} \ast P_{t+\tau}^{S} \right) \quad (33)$$

where $\ast$ is the cross correlation operator and $P_{t+\tau}^{S}$ is the realized total TCL demand with an additional time shift. Due to the use of purely periodic reference signals, the lag is limited to plus or minus half the reference’s period. The lag is not defined for the estimator simulations because the measurement delays influence the RMSE of the estimate but do not result in a lag between the real and estimated TCL demand.

VII. RESULTS

We now report the results for all combinations of reference signals and delay distributions described in Section VI-A. Our goal is to characterize the effect of delays when providing demand response services at various time-scales, i.e., with different periods of the sinusoidal reference
We investigate the performance of the MPC and state estimation algorithms in isolation, specifically the MPC algorithms receive a perfect state estimate at every time-step, and no input delays are used when evaluating the state estimation methods. Section VII-A presents the results related to the MPC performance, and Section VII-B presents the results related to the state estimator performance.

### A. Comparison of MPC Algorithms

Figure 4 shows each controllers’ RMSE as a function of the reference signal period and mean delay, Fig. 5 shows the relative amplitudes, Fig. 6 shows the time lags, and Fig. 7 shows time series for each controller using a reference period of 200 second and an average delay of 18 seconds.

Figure 4 indicates Controller 1 achieves requested amplitudes more effectively than Controller 2. Specifically, Controller 2 cannot achieve the requested amplitude when the delay length becomes a large portion of the reference period. This is due the controller refining the effective input for a given time-step in subsequent MPC calculations. Using the persistent reference within the formulation causes portions of the TCL population to actuate both upward and downward requests, and the overall demand change becomes less aggressive. This effect is less prevalent as the reference period increases. Overall, Controller 2 more effectively fulfills the requested demand changes, as can be seen from the lower RMSE values of Fig. 4.

Controller 1’s lag and relative amplitudes result in RMS tracking errors that generally increase for a given reference as the delay increases. The exception to this trend in the upper left region of Controller 1’s plots where delay becomes larger than half of the reference signal’s period. Since the reference signal is perfectly periodic, the realized demand begins to realign with the following period of the reference signal. This results in negative lag within Fig. 6 and appears as reduced tracking error within Fig. 4.

### B. Comparison of State Estimation Algorithms

The RMSEs and relative amplitudes for the state estimation algorithms are presented in Fig. 8 and Fig. 9, respectively. Figure 10 provides time series for each estimator using a reference period of 160 seconds and average delays of 20 seconds. The measurement delays result in a degradation in the aggregate power estimate accuracy and do not result in delays within the state estimate. As a result, the time lag results are not included.

Comparing the RMSEs in Fig. 8 we find that Estimator 3 achieves reduced estimation errors by including state predictions generated by individual, identified TCL models. These errors generally decrease as the reference signal’s period increases, indicating that the estimator may be more effective when supplying services at longer time-scales. The RMSE does not significantly vary as the delays increase, indicating that incorporating identified models makes the estimator relatively robust to measurement delays. Finally, Fig. 9 indicates that the amplitude of Estimator 3’s aggregate power estimate very accurately corresponds to the reference signal’s amplitude.

Alternatively, Estimator 1’s amplitude is not consistent with the reference signal amplitude, but it should be noted that the error is relatively small at 2-3%. Estimator 1’s RMSE tends to increase with increasing delays, which indicates it is more sensitive to measurement delays than Estimator 3. The RMSE tends to increase with increasing reference periods, indicating Estimator 1 may do poorly while providing services with longer time-scales.
Estimator 2 provides moderate improvements over Estimator 1. Specifically, the estimated power demand amplitude is closer to the desired value, and the RMSE is reduced at the higher delay levels. Generally, as the delay length increases, the improvement over Estimator 1 should increase. Section V-C claimed Estimators 1 and 2 should perform equivalently when no measurement delays are present. The raw RMSE for Controller 1 and 2 are equal in the delay-free cases, supporting this claim.

VIII. CONCLUSION

In this paper we presented state estimation and control algorithms to address communication issues that may arise within demand response. The algorithms take advantage of digital communication network characteristics to mitigate the effect of these communication issues while attempting to track a simplified power system signal. We investigated the performance of two MPC-based controllers and three Kalman filter-based estimators used to track purely sinusoidal reference signals of varying periods and under varying delays. Simulation results indicate that Controller 2, which uses knowledge of input delay probability distribution to estimate the expected input, provides more effective tracking in most scenarios. Estimator 3 provides the most accurate aggregate TCL demand estimates using individual models identified on-line to make state predictions that are used in a Kalman filter.

There are numerous suggestions for future work related to this research. Non-Gaussian noise is prevalent within the system, and its effect should either be characterized or the algorithms should be modified to address this. The use of a persistent forecast within the MPC is also a drawback in the controller formulation. The effect of a priori knowledge or predictions of the reference signal (e.g., based on historical data) should be investigated. Estimator 3’s performance is dependent on the accuracy of the identified models, and its performance should be characterized when exposed to more extreme disturbances. Finally, the algorithms developed within this work should be tested while tracking realistic power system signals.

ACKNOWLEDGMENT

The authors thank Stephan Koch for initial conversations.

REFERENCES


