Energy Disaggregation via Adaptive Filtering

Roy Dong, Lillian J. Ratliff, Henrik Ohlsson, and S. Shankar Sastry

Abstract— The energy disaggregation problem is recovering device level power consumption signals from the aggregate power consumption signal for a building. We show in this paper how the disaggregation problem can be reformulated as an adaptive filtering problem. This gives both a novel disaggregation algorithm and a better theoretical understanding for disaggregation. In particular, we show how the disaggregation problem can be solved online using a filter bank and discuss its optimality.

I. INTRODUCTION

Power consumption data of individual devices have the potential to greatly decrease costs in the electricity grid. Currently, residential and commercial buildings account for 40% of total energy consumption [1], and studies have estimated that 20% of this consumption could be avoided with efficiency improvements with little to no cost [2], [3]. It is believed that the largest barrier to achieving these energy cost reductions is due to behavioral reasons [4].

The authors of [5] claim that the full potential of the new smart meter technology cannot be exploited if human factors are not considered; that is, we must recognize that the smart grid is a system with a human in the loop. Furthermore, the authors note that billions of dollars are being expended on the installation of smart meters, which can provide the utility company with high resolution data on a building's power consumption. However, this hardware currently only provides the aggregate power consumption data, and deployment is at a sufficiently advanced stage that a change in hardware is prohibitively expensive.

Disaggregation presents a way in which consumption patterns of individuals can be learned by the utility company. This information would allow the utility to present this information to the consumer, with the goal of increasing consumer awareness about energy usage. Studies have shown that this is sufficient to improve consumption patterns [6].

Outside of informing consumers about ways to improve energy efficiency, disaggregation presents an opportunity for

R. Dong, L. Ratliff, H. Ohlsson, and S. S. Sastry are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, CA, USA {roydong, ratliffl, ohlsson, sastry} @eecs.berkeley.edu

H. Ohlsson is also with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden.

utility companies to strategically market products to consumers. It is now common practice for companies to monitor our online activity and then present advertisements which are targeted to our interests. This is known as 'personalized advertising'. Disaggregation of energy data provides a means to similarly market products to consumers. This leads to the question of user privacy and the question of ownership with regards to power consumption information. Treatment of the issue of consumer privacy in the smart grid is outside the scope of this paper. However, this is discussed in [7].

Additionally, disaggregation also presents opportunities for improved control. Many devices, such as heating, ventilation, and air conditioning (HVAC) units in residential and commercial buildings implement control policies that are dependent on real-time measurements. Disaggregation can provide information to controllers about system faults, such as device malfunction, which may result in inefficient control. It can also provide information about energy usage which is informative for demand response programs.

Our aim in this paper is to formulate the disaggregation problem in the filter banks framework. In doing so we extend our previous work in which we developed a method that combines the use of generative models, e.g. linear dynamical models of devices, with a supervised approach to disaggregation [8]. In particular, we develop an algorithm for disaggregation of whole building energy data using dynamical models for devices and filter banks for determining the most likely inputs to the dynamical models. Under mild assumptions on the noise characteristics we are able to provide guarantees for when the algorithm recovers the disaggregated signal that most closely matches our observed data and priors.

In Section II, we discuss previous work on the topic of energy disaggregation. In Section III, we formally define the problem of energy disaggregation. In Sections IV-A to IV-B, we establish our framework for solving the problem of energy disaggregation. In Section V, we provide an online adaptive filtering algorithm for estimating individual device power consumption patterns, and in Section VI, we prove properties of this algorithm. In Section VII, we show energy disaggregation results from a small-scale experiment. Finally, in Section VIII, we give concluding remarks and describe plans for future work.

II. BACKGROUND

The problem of energy disaggregation, and the existing hardware for disaggregation, has been studied extensively in the literature (see [9], [10], for example). The goal of the current disaggregation literature is to present methods for improving energy monitoring at the consumer level

without having to place sensors at device level, but rather use existing sensors at the whole building level. The concept of disaggregation is not new; however, only recently has it gained attention in the energy research domain, likely due to the emergence of smart meters and big data analytics, as discussed in Section I.

Disaggregation, in essence, is a single-channel source separation problem. The problem of recovering the components of an aggregate signal is an inverse problem and as such is, in general, ill-posed. Most disaggregation algorithms are batch algorithms and produce an estimate of the disaggregated signals given a batch of aggregate recordings. There have been a number of survey papers summarizing the existing methods (e.g. see [11], [12]). In an effort to be as self-contained as possible, we try to provide a broad overview of the existing methods and then explain how the disaggregation method presented in this paper differs from existing solutions.

The literature can be divided into two main approaches, namely, supervised and unsupervised. Supervised disaggregation methods require a disaggregated data set for training. This data set could be obtained by, for example, monitoring typical appliances using plug sensors. Supervised methods assume that the variations between signatures for the same type of appliances is less than that between signatures of different types of appliances. Hence, the disaggregated data set does not need to be from the building that the supervised algorithm is designed for. However, the disaggregated data set must be collected prior to deployment, and come from appliances of a similar type to those in the target building. Supervised methods are typically discriminative.

Unsupervised methods, on the other hand, do not require a disaggregated data set to be collected. They do, however, require hand tuning of parameters, which can make it hard for the methods to be generalized in practice. It should be said that also supervised methods have tuning parameters, but these can often be tuned using the training data.

The existing supervised methods include sparse coding [13], change detection and clustering based approaches [14], [15] and pattern recognition [16]. The sparse coding approach tries to reconstruct the aggregate signal by selecting as few signatures as possible from a library of typical signatures. Similarly, in our proposed framework we construct a library of dynamical models and reconstruct the aggregate signal by using as few as possible of these models.

The existing unsupervised methods include factorial hidden Markov models (HMMs), difference hidden Markov models and variants [17], [18], [19], [20], [21] and temporal motif mining [22]. Most unsupervised methods model the on/off sequences of appliances using some variation of HMMs. These methods do not directly make use of the signature of a device and assume that the power consumption is piecewise constant.

All methods we are aware of lack the use of the dynamics of the devices. While the existing supervised methods often do use device signatures, these methods are discriminative and an ideal method would be able to generate a power consumption signal from a given consumer usage profile.

Both HMMs and linear dynamical models are generative as opposed to discriminative, making them more advantageous for modeling complex system behavior. In the unsupervised domain, HMMs are used; however, they are not estimated using data and they do not model the signature of a device.

In a previous paper we developed a method which combines the use of generative models, i.e. linear dynamical models of devices, with a supervised approach to disaggregation [8]. In this paper, we extend previous work by formalizing our method within an adaptive filtering framework. Specifically, we formulate hypotheses on the on/off state of the devices over the time horizon for which we have data. The on/off state corresponds to whether the input is activated or not. Using filter banks and the dynamical models we have for device behavior, we evaluate which is the most likely hypothesis on the inputs. We provide an algorithm for this process. Under mild assumptions on the noise characteristics we are able to provide guarantees for when the our algorithm results in an optimal solution. The filter bank framework is similar to HMM frameworks in the sense that both methods essentially formulate hypotheses on which devices are on at each time instant. However, in contrast to HMMs, in the filter bank framework we incorporate the use of dynamical models to capture the transients of the devices, which helps identify them.

III. PROBLEM FORMULATION

In this section, we formalize the problem of energy disaggregation.

Suppose we are given an aggregated power consumption signal for a building. We denote this data as $y[t]$ for $t = 0, 1, \ldots, T$, where $y[t]$ is the aggregate power consumption at time t. The entire signal will be referred to as y. This signal is the aggregate of the power consumption signal of several individual devices:

$$y[t] = \sum_{i=1}^{D} y_i[t] \text{ for } t = 0, 1, \ldots, T, \qquad (1)$$

where D is the number of devices in the building and $y_i[t]$ is the power consumption of device i at time t. The goal of disaggregation is to recover y_i for $i = 1, 2, \ldots, D$ from y.

To solve this problem, it is necessary to impose additional assumptions on the signals y_i and the number of devices D.

IV. PROPOSED FRAMEWORK

At a high level, our framework can be summarized as follows. First, in the training phase of our disaggregation framework, we assume we have access to a training set of individual device power consumption data that is representative of the devices in the buildings of concern. From this training data, we build a library of models for individual devices. With these models, the disaggregation step becomes finding the most likely inputs to these devices that produces our observed output, the aggregate power consumption signal.

A. Training phase

Suppose we have a training data set, which consists of the power consumption signals of individual devices. Let $z_i[t]$ for $t = 0, 1, \ldots, T_i$ be a power consumption signal for a device i. Then, $\{z_i\}_{i=1}^{D}$ is our training data. From this training data, we will learn models for individual devices.

For device i, we assume the dynamics take the form of a finite impulse response (FIR) model:

$$z_i[t] = \sum_{j=0}^{n_i} b_{i,j} u_i^z[t-j] + e_i[t], \tag{2}$$

where n_i is the order of the FIR model corresponding to device i, $b_{i,j}$ represent the parameters of the FIR model and $e_i[t]$ is white noise, i.e. random variables that are zero mean, finite variance, and independent across both time and devices. Furthermore, $u_i^z[t]$ represents the input to device i at time t in the training dataset, z.

We now make the following assumption:

Assumption IV.1. FIR models fed by piecewise constant inputs give a rich enough setup to model the energy consumed by individual appliances.

Firstly, many electrical appliances can be seen having a piecewise constant input. For example, the input of a conventional oven can be seen as $0°F$ if the oven is off, and $300°F$ if the oven is set to heat to $300°F$. Note that the input is not the actual internal temperature of the oven, but rather the temperature setting on the oven. Since the temperature setting is relatively infrequently changed, the input is piecewise constant over time. Many other appliances are either on or off, for example lights, and can be seen having a binary input with infrequent changes. This is also a piecewise constant input. For a washing machine, we have a discrete change between modes (washing, spinning, etc.) and this mode sequence can be seen as the piecewise constant input of the washing machine.

Secondly, a FIR model can fit arbitrarily complex stable linear dynamics. Assuming that FIR models fed by piecewise constant inputs give a rich enough setup to model the energy consumed by individual appliances is therefore often sufficient for energy disaggregation.

Thirdly, without any assumption on the inputs, the disaggregation problem later presented in Section IV-B is ill-posed; thus, Assumption IV.1, which assumes that changes in input are sparse, serves as a regularization which helps make the problem less ill-posed.

In most applications, we will not have access to any input data. Thus, our system identification step becomes estimation of both the input and the FIR parameters. This is known as a blind system identification problem, and is generally very difficult.

However, with the assumption that the inputs represent an on/off sequence, we can use simple change detection methods to estimate the binary input u_i^z. For more complicated inputs, we refer to [23].

Although n_i is not known a priori, we can select the value of n_i using criterion from the system identification and statistics literature. For example, one can use the Akaike information criterion (AIC) or the Bayesian information criterion (BIC). For more information on model selection, as well as other possible criteria for model selection, we refer the reader to [24].

Finally, we can succinctly rewrite (2) in vector form:

$$z_i[t] = \boldsymbol{\beta}_i^\top \boldsymbol{\xi}_i[t] + e_i[t], \tag{3}$$

where $\boldsymbol{\beta}_i$ are the FIR parameters:

$$\boldsymbol{\beta}_i = \begin{bmatrix} b_{i,0} & b_{i,1} & \ldots & b_{i,n_i} \end{bmatrix}^\top, \tag{4}$$

and $\boldsymbol{\xi}_i[t]$ are the regressors at time t:

$$\boldsymbol{\xi}_i[t] = \begin{bmatrix} u_i^z[t] & u_i^z[t-1] & \ldots & u_i^z[t-n_i] \end{bmatrix}^\top. \tag{5}$$

B. Energy disaggregation

Suppose we have estimated a library of models for devices $i = 1, 2, \ldots, D$. That is, we are given $\boldsymbol{\beta}_i$ for devices $i = 1, 2, \ldots, D$. Furthermore, we are given y. We wish to find y_i for $i = 1, 2, \ldots, D$. Now, we make the following assumption:

Assumption IV.2. The devices in our building are a subset of the devices $\{1, 2, \ldots, D\}$. Furthermore, these devices have dynamics of the form in (3).

Note here that we assume that all devices are modeled in our library, or, equivalently, all devices are represented in our training data. This is a common assumption in the disaggregation literature but we plan to relax this assumption in future work.

Now, this problem is equivalent to finding inputs to our devices that generate our observed aggregated signal. More explicitly, let:

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^\top & \boldsymbol{\beta}_2^\top & \ldots & \boldsymbol{\beta}_D^\top \end{bmatrix}^\top, \tag{6}$$

$$\boldsymbol{\psi}[t] = \begin{bmatrix} \boldsymbol{\psi}_1[t]^\top & \boldsymbol{\psi}_2[t]^\top & \ldots & \boldsymbol{\psi}_D[t]^\top \end{bmatrix}^\top, \tag{7}$$

where $\boldsymbol{\beta}_i$ are as defined in (4) for each device $i = 1, 2, \ldots, D$ and:

$$\boldsymbol{\psi}_i[t] = \begin{bmatrix} u_i[t] & u_i[t-1] & \ldots & u_i[t-n_i] \end{bmatrix}^\top. \tag{8}$$

Then, we have a model for the aggregate power signal:

$$y[t] = \boldsymbol{\beta}^\top \boldsymbol{\psi}[t] + e[t], \tag{9}$$

where $e[t] = \sum_{i=1}^{D} e_i[t]$ is still white noise. For simplicity, we assume zero initial conditions, i.e. $u_i[t] = 0$ for $t = -n_i, -n_i+1, \ldots, -1$. This assumption can easily be relaxed. Thus, the problem of energy disaggregation is now finding $u_i[t]$ for $t = 0, 1, \ldots, T$ and $i = 1, 2, \ldots, D$. Let:

$$\boldsymbol{u}[t] = \begin{bmatrix} u_1[t] & u_2[t] & \ldots & u_D[t] \end{bmatrix}^\top. \tag{10}$$

Recall that in the training phase we assumed that FIR models fed by piecewise constant inputs gave a rich enough setup for accurately modeling energy consumption of individual appliances. We will in the disaggregation step similarly

assume that $u_i[t], i = 1, \ldots, D$, are piecewise constant over time. It follows that the vector-valued function $\boldsymbol{u}$ is piecewise constant.

Let a segment be defined as an interval in which $\boldsymbol{u}$ is constant. Then, energy disaggregation becomes a segmentation problem. More formally, define a segmentation as $k^n = (k_1, k_2, \ldots, k_n)$ such that $0 \leq k_1 < k_2 < \cdots < k_n$. Here, both n and k_l, $l = 1, \ldots, n$, are unknown. For a segmentation k^n, we have that:

$$\boldsymbol{u}[s] = \boldsymbol{u}[t] \text{ for all } k_{l-1} < s, t \leq k_l, \tag{11}$$

with $k_0 = -1$.

Here, we will introduce some additional notation which will be helpful for the rest of this paper.

First, we introduce an alternative notation for segmentations. Let $\delta[t] = 1$ if $\boldsymbol{u}[t] \neq \boldsymbol{u}[t-1]$, and 0 otherwise. In other words, $\delta[t]$ is a binary variable that equals 1 if and only if the input changes between times $t-1$ and t. Thus, $k^n = (k_1, k_2, \ldots, k_n)$ and $\delta \in \{0, 1\}^T$ are equivalent representations of a segmentation. Throughout this paper we shall freely move between the two.

Next, suppose we are given a segmentation k^n. Then for each device i, we can define a function $\bar{u}_i : \{1, 2, \ldots, n\} \to \mathbb{R}$ such that $\bar{u}_i(l) = u_i[k_l]$, i.e. $\bar{u}_i(l)$ represents the input to device i in the lth segment. Then, let $\bar{\boldsymbol{u}} : \{1, 2, \ldots, n\} \to \mathbb{R}^D, l \mapsto (\bar{u}_1(l), \bar{u}_2(l), \ldots, \bar{u}_D(l))$. $\bar{\boldsymbol{u}}(l)$ represents the input to all devices in the lth segment.

Also, let y^t denote all measurements available at time t. That is, $y^t = (y[0], y[1], \ldots, y[t])$.

Let $p(\boldsymbol{u})$ denote a probability distribution on the user's input to the devices; that is, $p(\boldsymbol{u})$ is the likelihood of the input $\boldsymbol{u}$. This encapsulates our prior on user consumption patterns. For example, in residential buildings, power consumption tends to be low mid-day, while in commercial buildings, power consumption drops off after work hours. This knowledge can be represented in $p(\boldsymbol{u})$.

The disaggregation problem is to find the maximum a posteriori (MAP) estimate of $\boldsymbol{u}$ and, consequently, the power consumption of device i, given our observations. In Section V, we provide an adaptive filtering algorithm for solving this problem, and in Section VI, we provide theoretical guarantees of our proposed algorithm.

There are many criteria other than the MAP by which to select a segmentation. The best criteria for selection of a segmentation is an active topic of debate, and a thorough treatment of this question is outside the scope of this paper. We refer the interested reader to [25] for more details on segmentation.

V. ENERGY DISAGGREGATION VIA ADAPTIVE FILTERING

A. Algorithm definition

In this section, we provide a tractable algorithm to solve the problem posed in Section IV. Furthermore, this algorithm is defined recursively on measurements across time, so it can be run online.

We draw on results in the adaptive filtering literature. An adaptive filter is any filter that adjusts its own parameters based on observations. In our particular case, we use a filter bank approach to handle the problem presented in Section IV. A filter bank is a collection of filters, and the adaptive element of a filter bank is in the insertion and deletion of filters, as well as the selection of the optimal filter.

We will define a filter bank, and also the problem a filter bank solves. Suppose we are given measurements y^t. We wish to find the maximum a posteriori estimate of the input $\boldsymbol{u}$ given our measurements y^t: we wish to find $\boldsymbol{u}$ that maximizes $p(\boldsymbol{u}|y^t)$, which is equivalent to maximizing $p(y^t|\boldsymbol{u})p(\boldsymbol{u})$. Decomposing u into δ and $\bar{\boldsymbol{u}}$, we can again rewrite this as maximizing $p(y^t|\bar{\boldsymbol{u}}, \delta)p(\bar{\boldsymbol{u}}|\delta)p(\delta)$. Note that we can calculate:

$$p(\delta) = \int p(\bar{\boldsymbol{u}}, \delta) d\bar{\boldsymbol{u}}. \tag{12}$$

The final manipulation is that we wish to find a δ to maximize the following quantity:

$$\max_{\bar{\boldsymbol{u}}} \ p(y^t|\bar{\boldsymbol{u}}, \delta)p(\bar{\boldsymbol{u}}|\delta)p(\delta). \tag{13}$$

Now, our algorithm maintains a collection of filters, known as a filter bank. Let $\mathcal{F}$ denote this filter bank. Each filter $f \in \mathcal{F}$ corresponds to a segmentation $\delta_f \in \{0, 1\}^t$. Given each δ_f, we can calculate:

$$\bar{\boldsymbol{u}}_f = \underset{\bar{\boldsymbol{u}}}{\operatorname{argmax}} \ p(y^t|\bar{\boldsymbol{u}}, \delta_f)p(\bar{\boldsymbol{u}}|\delta_f)p(\delta_f). \tag{14}$$

$$p_f = \max_{\bar{\boldsymbol{u}}} \ p(y^t|\bar{\boldsymbol{u}}, \delta_f)p(\bar{\boldsymbol{u}}|\delta_f)p(\delta_f). \tag{15}$$

There are only finitely many possible δ. Thus, if we kept a filter f for every possible segmentation δ, we could easily find the MAP estimate of $\boldsymbol{u}$. However, the filter bank $\mathcal{F}$ would grow exponentially with time. Thus, we need to find methods to keep the size of $\mathcal{F}$ under control.

The process of finding the best segmentation can be seen as exploring a binary tree. That is, a segmentation δ can be thought of as a leaf node on a binary tree of depth t. This is visualized in Figure 1.

Limiting the growth of $\mathcal{F}$ can be done by deciding which branches to expand and which branches to prune. This sort of formulation lends itself very easily to an online formulation of the filter banks algorithm. In fact, it is more intuitive to think of the algorithm in an online fashion.

At time t, we choose to branch a filter only if it corresponds to one of the most likely segmentations. By branching, we refer to exploring both the 0 and 1 branches. This is depicted by the blue and green lines in Figure 1. Otherwise, we will merely extend the last segment of the segmentation, i.e. only follow the 0 branch. Additionally, at time t, we prune any paths that have sufficiently low likelihood. That is, we remove the filter f from $\mathcal{F}$ if $p_f < p_{\text{thres}}$, where p_{thres} is an algorithm parameter. This is depicted by the red dotted line in Figure 1.

Finally, we can exhibit our algorithm. It is presented in Figure 2.
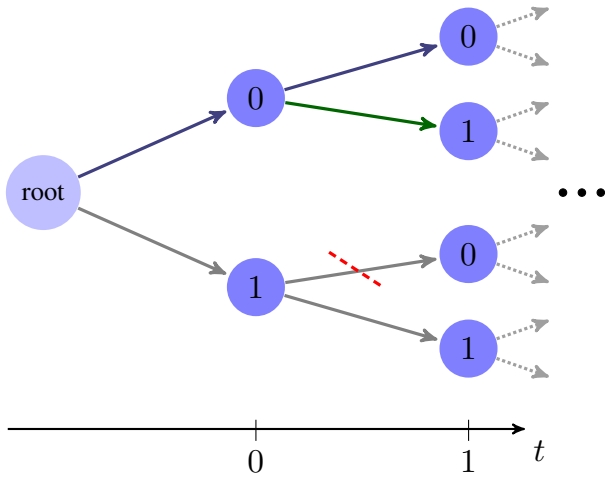
Fig. 1. A segmentation δ can be thought of as a leaf node on a binary tree of depth t. That is, δ corresponds exactly to one leaf node of this binary tree. If we choose to branch the blue path (0) at time 0, then we add the green path $(0,1)$ as well as the blue path $(0,0)$. If we choose not to branch (0), then only $(0,0)$ would be added to our filter bank. The red dotted-line depicts pruning; this would involve removing $(1,0)$ and all its children from the tree.

As presented currently, the algorithm give in Figure 2 is a high-level algorithm. We now discuss a specific implementation. First, we will add some assumptions.

Assumption V.1. In the true $\boldsymbol{u}$, each segment has length greater than or equal to $N = \max\{n_1, n_2, \ldots, n_D\}$.

This assumption places a minimum length of a segment for our piecewise constant input $\boldsymbol{u}$; it asserts that each device is in steady-state before a device changes state.

Let $y_{\mathrm{ss},i}(k^n, l)$ denote the steady state value of y_i, the power consumption for the ith device, in the lth segment of k^n. That is:

$$y_{\mathrm{ss},i}(k^n, l) = \sum_{j=0}^{n_i} b_{i,j} \bar{u}_i(l). \tag{16}$$

Then, let $y_{\mathrm{ss}}(k^n, l)$ denote the steady-state value of y in the lth segment of k^n; thus:

$$y_{\mathrm{ss}}(k^n, l) = \sum_{i=1}^{D} y_{\mathrm{ss},i}(k^n, l). \tag{17}$$

y_{ss} can be directly estimated from our observations y, independently of our values for $\bar{\boldsymbol{u}}$. We will assume $y_{\mathrm{ss}}(k^n, l)$ is known from this point onward. Additionally, let $y_{\mathrm{ss}}(k^n, 0) = 0$.

Suppose we are given a segmentation δ. We will convert the estimation of $\bar{\boldsymbol{u}}$ into the estimation of the change in $\bar{\boldsymbol{u}}$. To such end, define $\Delta \bar{\boldsymbol{u}}(l) = \bar{\boldsymbol{u}}(l) - \bar{\boldsymbol{u}}(l-1)$, the change in input from segment $l-1$ to segment l, with $\Delta \bar{\boldsymbol{u}}(0) = 0$. Then, with Assumption V.1, linearity implies that our dynamics take the following form:

$$y[t] - y_{\mathrm{ss}}(k^n, l-1) = \mathcal{L}_{k_{l-1}+1}(\Delta \bar{\boldsymbol{u}}(l), t) + e[t] \\ \text{for } k_{l-1} < t \leq k_l, \tag{18}$$

1: Initialize $t \leftarrow 0$, $f_0 \leftarrow \delta_{f_0} = (0)$, $f_1 \leftarrow \delta_{f_1} = (1)$, and $\mathcal{F} \leftarrow \{f_0, f_1\}$.
2: Pick algorithm parameter p_{thres}.
3: **while** TRUE **do**
4:     // *Find the filters that correspond to the most likely*
5:     // *segmentations given y^t.*
6:     $\mathcal{F}' \leftarrow \emptyset$.
7:     **for** $f \in \mathcal{F}$ **do**
8:         **if** $p_f = \max_{f' \in \mathcal{F}} p_{f'}$ **then**
9:             Add a copy of f to $\mathcal{F}'$.
10:         **end if**
11:     **end for**
12:     // *When available, update all filters in $\mathcal{F}$ with the*
13:     // *new measurement.*
14:     Wait for new measurement $y[t+1]$.
15:     **for** $f \in \mathcal{F}$ **do**
16:         Append 0 to δ_f. Recalculate $\bar{\boldsymbol{u}}_f$ and p_f.
17:     **end for**
18:     // *Branch the filters corresponding to the most likely*
19:     // *segmentations given y^t.*
20:     **for** $f' \in \mathcal{F}'$ **do**
21:         Append 1 to $\delta_{f'}$.
22:         Recalculate $\bar{\boldsymbol{u}}_{f'}$ and $p_{f'}$.
23:         Add f' to $\mathcal{F}$.
24:     **end for**
25:     // *Prune elements from the filter bank that have*
26:     // *unlikely segmentations.*
27:     **for** $f \in \mathcal{F}$ **do**
28:         **if** $p_f < p_{\mathrm{thres}}$. **then**
29:             Remove f from $\mathcal{F}$.
30:         **end if**
31:     **end for**
32:     $t \leftarrow t + 1$.
33: **end while**

Fig. 2. Algorithm for Online energy disaggregation via filter banks (OEDFB).

where $\mathcal{L}_{k_l-1}(\Delta \bar{\boldsymbol{u}}(l), t)$ is the value of the zero-state step response at time t of the aggregated system model in (9) to a step of $\Delta \bar{\boldsymbol{u}}(l)$ beginning at time $k_{l-1} + 1$. Note that this linear function can easily be calculated from $\boldsymbol{\beta}$.

The essential point of this equation is that, since all the devices are in steady state at the beginning of the lth segment, the actual values of $\bar{\boldsymbol{u}}(l-1)$ and $\bar{\boldsymbol{u}}(l)$ do not matter; the dynamics depend only on the change $\Delta \bar{\boldsymbol{u}}(l)$. Thus, we can estimate $\Delta \bar{\boldsymbol{u}}(l)$ separately for each segment l.

Furthermore, we consider the following prior. Suppose we have a bound on how much the input can change from segment to segment. That is, we know $\Delta \boldsymbol{u}_{\min}, \Delta \boldsymbol{u}_{\max}$ such that $\Delta \boldsymbol{u}_{\min} \leq \Delta \bar{\boldsymbol{u}}(l) \leq \Delta \boldsymbol{u}_{\max}$ for all l. Furthermore, $p(\bar{\boldsymbol{u}}|\delta)$ is a uniform distribution within these bounds.

Finally, if the noise term in (9) is Gaussian white noise with fixed variance σ^2, then the calculations of $\bar{\boldsymbol{u}}_f$ and p_f are relatively straightforward. Let y^l denote the portion of y

in segment l:

$$y^l = \begin{bmatrix} y[k_{l-1}+1] & y[k_{l-1}+2] & \ldots & y[k_l] \end{bmatrix}^\top. \quad (19)$$

By a slight abuse of notation, simply let $\mathcal{L}(\boldsymbol{u})$ denote the zero-state response of (9) to a step of $\boldsymbol{u}$. We can find $\Delta\bar{\boldsymbol{u}}(l)$ by solving the following least-squares problem:

$$\min_{\Delta\bar{\boldsymbol{u}}} \|y^l - y_{\text{ss}}(k^n, l-1) - \mathcal{L}(\Delta\bar{\boldsymbol{u}})\|_2^2 \\ \text{subj. to } \Delta\boldsymbol{u}_{\min} \le \Delta\bar{\boldsymbol{u}} \le \Delta\boldsymbol{u}_{\max}. \quad (20)$$

This will give us $\bar{\boldsymbol{u}}_f$. Let:

$$e[t] = y[t] - y_{\text{ss}}(k^n, l-1) - \mathcal{L}(\Delta\bar{\boldsymbol{u}}(l)) \text{ for } k_{l-1} < t \le k_l \quad (21)$$

We can also calculate:

$$p_f = cp(\delta) \prod_{s=0}^{t} \exp\left(-\frac{e[s]^2}{2\sigma^2}\right) \quad (22)$$

where c is a constant that is independent of δ and $\bar{\boldsymbol{u}}$.

VI. THEORY

One of the benefits of our framework is that it allows us to leverage results from adaptive filtering. In this section, we prove theorems relating to the algorithm presented in Section V.

Let $\widehat{\delta^t}$ denote any segmentation such that:

$$p(\widehat{\delta^t}|y^t) = \max_{\delta \in \{0,1\}^t} p(\delta|y^t), \quad (23)$$

for any $t \in \{0,1,\ldots,T\}$. That is, $\widehat{\delta^t}$ denotes a maximum a posteriori estimate of the segmentation δ at time t. We can now apply the following result:

Theorem VI.1. (*Optimality of partial MAP estimates* [25]) Let t be any arbitrary time in $\{0,1,\ldots,T\}$, and let t_0 be any time such that $0 \le t_0 \le t$. Let δ be any binary sequence of length t such that $\delta[t_0] = 1$. Let δ_1 denote the first $t_0 - 1$ elements of δ and δ_2 denote the last $t - t_0$ elements of δ. That is: $\delta = (\delta_1, 1, \delta_2)$.

If Assumption V.1 holds and if $\Delta\bar{\boldsymbol{u}}(l)$ and $\Delta\bar{\boldsymbol{u}}(m)$ are independent given δ for $l \ne m$, then:

$$p(\delta|y^t) \le p((\widehat{\delta^{t_0-1}}, 1, \delta_2)|y^t) \quad (24)$$

Proof. Note that our hypotheses together imply that $\Delta\bar{\boldsymbol{u}}(l)$ and $\Delta\bar{\boldsymbol{u}}(m)$ are independent given δ and y^t for $l \ne m$. Thus:

$$\begin{aligned} p(\delta|y^t) &= p((\delta_1, 1, \delta_2)|y^t, \delta[t_0]=1)p(\delta[t_0]=1|y^t) \\ &= p(\delta_1|y^t, \delta[t_0]=1)p(\delta_2|y^t, (\delta_1, 1)) \cdot \\ &\quad p(\delta[t_0]=1|y^t) \\ &= p(\delta_1|y^{t_0-1})p(\delta_2|y^t, \delta[t_0]=1)p(\delta[t_0]=1|y^t) \\ &\le p(\widehat{\delta^{t_0-1}}|y^{t_0-1})p(\delta_2|y^t, \delta[t_0]=1) \cdot \\ &\quad p(\delta[t_0]=1|y^t) \\ &= p(\widehat{\delta^{t_0-1}}|y^t)p(\delta_2|y^t, \delta[t_0]=1) \cdot \\ &\quad p(\delta[t_0]=1|y^t) \\ &= p((\widehat{\delta^{t_0-1}}, 1, \delta_2)|y^t) \end{aligned} \quad (25)$$

where, by a slight abuse of notation, $p(\delta_1|\delta_2)$ denotes the likelihood that the first $t_0 - 1$ elements of the true δ are

equal to δ_1 given that the last $t - t_0$ elements are equal to δ_2.

The first and second equalities utilize Bayes' law. Causality and independence of segments imply that δ_2 does not depend on δ_1 given that $\delta[t_0] = 1$ and that δ_1 does not depend on later measurements given that $\delta[t_0] = 1$. This gives us the third equality. The inequality follows from the definition of $\widehat{\delta^{t_0-1}}$, given in (23). The final equalities are similar to the first equalities. □

This theorem implies that, conditioned on a change at time t_0, the MAP sequence at time t must begin with the MAP sequence at time t_0. Also, note that under Assumption V.1, we have that $\Delta\bar{\boldsymbol{u}}(l)$ and $\Delta\bar{\boldsymbol{u}}(m)$ are independent given δ for $l \ne m$.

We can now assert the following claims about our algorithm:

Theorem VI.2. (*Optimality of the proposed algorithm's branching policy* [25]) Consider the algorithm given in Figure 2 with $p_{\text{thres}} = 0$. Suppose Assumption V.1 holds and $\Delta\bar{\boldsymbol{u}}(l)$ and $\Delta\bar{\boldsymbol{u}}(m)$ are independent given δ for $l \ne m$.

Fix any time t, and let $\mathcal{F}$ be the filter bank at time t. Then, there exists an $f \in \mathcal{F}$ such that $\delta_f = \widehat{\delta^t}$.

Proof. If $\widehat{\delta^t}[s] = 1$ for some s, then the first $s - 1$ elements of $\widehat{\delta^t}$ is a MAP estimate at time $s - 1$. This follows from Theorem VI.1. This means that, at time s, we only need to branch the most likely segmentations. □

Theorem VI.2 states that, in the case of no pruning, any MAP estimate will still be present in the filter bank. In other words, maximizing over the reduced set of filters in the filter bank will be equivalent to maximizing over every single possible segmentation.

This theorem also gives rise to our corollary. First, let $(\widehat{\delta^t})^s$ denote the first s elements of $\widehat{\delta^t}$. Then:

Corollary VI.3. (*Optimality of proposed algorithm's pruning policy*) Consider the algorithm given in Figure 2 with $p_{\text{thres}} > 0$. Suppose Assumption V.1 holds and $\Delta\bar{\boldsymbol{u}}(l)$ and $\Delta\bar{\boldsymbol{u}}(m)$ are independent given δ for $l \ne m$.

Fix any time t, and let $\mathcal{F}$ be the filter bank at time t. If $p((\widehat{\delta^t})^s|y^s) \ge p_{\text{thres}}$ for all $0 \le s < t$, then there exists an $f \in \mathcal{F}$ such that $\delta_f = \widehat{\delta^t}$.

Proof. If $p(\widehat{\delta^s}|y^s) \ge p_{\text{thres}}$ for all $0 \le s < t$, then $\widehat{\delta^t}$ will never be pruned. □

Corollary VI.3 states a condition for when an MAP estimate will still be present in the filter bank at time t.

VII. EXPERIMENT

A. Experimental setup

To test our disaggregation method, we deployed a small-scale experiment. To collect data, we use the emonTx wireless open-source energy monitoring node from OpenEnergyMonitor[1]. We measure the current and voltage of devices

[1]http://openenergymonitor.org/emon/emontx

with current transformer sensors and an alternating current (AC) to AC power adapter. For each device i, we record the root-mean-squared (RMS) current I_{RMS}^i, RMS voltage V_{RMS}^i, apparent power P_{VA}^i, real power P_W^i, power factor ϕ_{pf}^i, and a coordinated universal time (UTC) stamp. The data was collected at a frequency of 0.13Hz.

Our experiment focused on small devices commonly found in a residential or commercial office building. First, we recorded plug-level data z_i for a kettle, a toaster, a projector, a monitor, and a microwave. These devices consume anywhere from 70W to 1800W. For each device, we fit a fifth-order FIR model as outlined in Section IV-A.

Then, we ran an experiment using a microwave, a toaster, and a kettle operating at different time intervals. These measurements form our ground truth y_i, and we also sum the signals to get our aggregated power signal $y = \sum y_i$. The individual plug measurements are shown in Figure 3. It is worth commenting that the power consumption signals for individual devices are not entirely independent; one device turning on can influence the power consumption of another device. This coupling is likely due to the non-zero impedance of the power supply system. However, we found this effect to be negligible in our disaggregation algorithms.
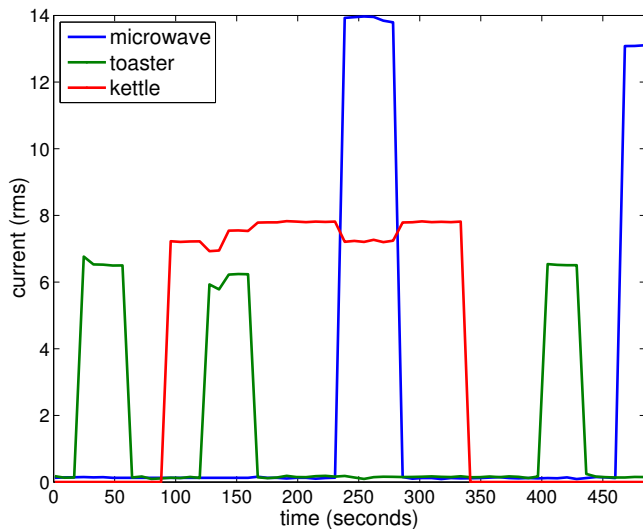

Fig. 3.   The measurements of individual plug RMS currents.

B. Implementation details

In practice, we observed that many devices seem to have different dynamics between when they switch on and when they switch off. For example, consider the root-mean-squared (RMS) current of a toaster in Figure 4. There is an overshoot when the toaster switches on, but the the dynamics when the device shuts off do not exhibit the same behavior. In fact, in all of the devices we measured, we found that when a devices switches off, the power consumption drops down to a negligible amount almost immediately. That is, we do not observe any transients when a device turns off. We modify the models from Section IV-A to encapsulate this observation.
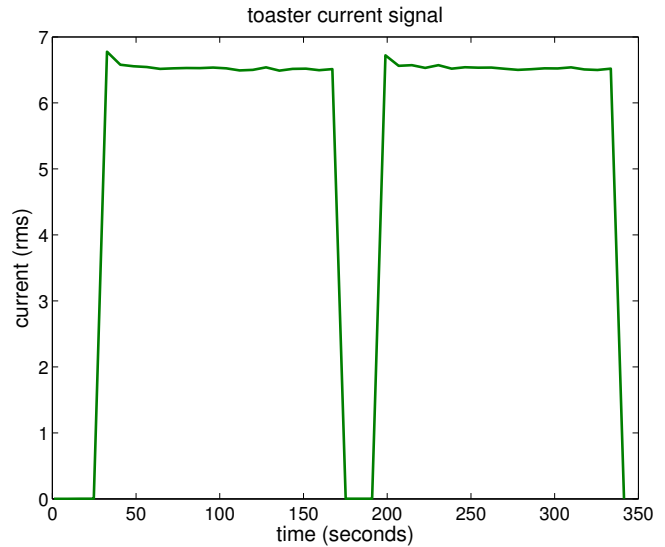

Fig. 4.   The measured RMS current signal for a toaster. Note that the on-switches display overshoot while the off-switches do not.

Several heuristics are used for pruning the binary tree depicted in Figure 1 that are specific to the task of disaggregation. First, we do not bother considering branches if the most likely segmentation explains the data sufficiently well. This greatly reduces the growth of the filter bank across time. Furthermore, we assume that at most one device switches on or off in any given time step. This unfortunately violates the assumptions of Theorem VI.1, but we find that it gives good results in practice.

C. Results

The disaggregation results are presented in Figure 5. We can see that the segmentation is correctly identified. Visually, the results also line up well.
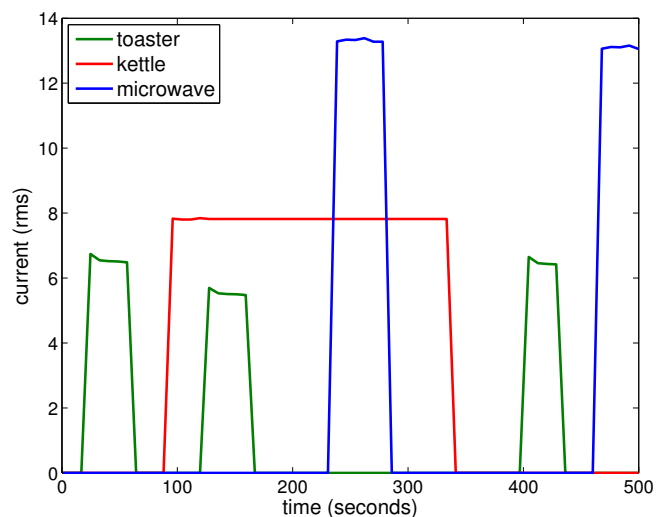

Fig. 5.   The estimated power consumption signals of each device.

We also note that it is not fair to compare results from our

small-scale experiment with many of the methods mentioned in Section II. Most of the methods listed are unsupervised methods which do not have a training set of data [12], [22], [18], [19]. Since these unsupervised methods do not learn from training data, they have many priors which must be tuned towards the devices in the library. Also, the sparse coding method in [13] requires a large amount of disaggregated data to build a dictionary.

VIII. CONCLUSIONS AND FUTURE WORK

In the work presented, we formalized the disaggregation problem within the filter banks framework. We provide an algorithm with guarantees on the recovery of the true solution given some assumptions on the data.

From the point of view of the utility company, the question of how to use this data to inform the consumer about their usage patterns and how to develop incentives for behavior modification is still largely an open one, which we are currently studying.

Another largely open question is the one concerning privacy. Given that energy data can be disaggregated with some degree of precision, how does this affect the consumer's privacy? The next natural step is to study how this data can be used in a privacy preserving way to improve energy efficiency. These privacy preserving policies may come in the form of selectively transmitting the most relevant data for a control objective, or incentive mechanisms for users to change their consumption behavior without direct transmission of their private information to the utility company. We are currently examining both approaches to the privacy issue.

REFERENCES

[1] L. Perez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, pp. 394–398, 2008.

[2] J. Creyts, A. Derkach, S. Nyquist, K. Ostrowski, and J. Stephenson, "Reducing U.S. greenhouse gas emissions: How much at what cost?" U.S. Greenhouse Gas Abatement Mapping Initiative, Tech. Rep., 2007.

[3] J. A. Ò. Laitner, K. Ehrhardt-Martinez, and V. McKinney, "Examining the scale of the behaviour energy efficiency continuum," in *European Council for an Energy Efficient Economy*, 2009.

[4] G. Crabtree, "Energy future report: 'energy future: think efficiency'," American Physical Society, Tech. Rep., 2008.

[5] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301421512007446

[6] K. Ehrhardt-Martinez, K. A. Donnelly, and J. A. Laitner, "Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities," American Council for an Energy-Efficient Economy, Tech. Rep., June 2010.

[7] A. A. Cardenas, S. Amin, G. Schwartz, R. Dong, and S. Sastry, "A game theory model for electricity theft detection and privacy-aware control in ami systems," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012.

[8] R. Dong, L. Ratliff, H. Ohlsson, and S. S. Sastry, "A dynamical systems approach to energy disaggregation," *arXiv*, Apr. 2013.

[9] M. E. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Enhancing electricity audits in residential buildings with nonintrusive load monitoring," *Journal of Industrial Ecology*, vol. 14, no. 5, pp. 844–858, 2010.

[10] M. Berges, E. Goldman, H. S. Matthews, and L. Soibelman, "Learning systems for electric consumption of buildings," in *ASCI International Workshop on Computing in Civil Engineering*, 2009.

[11] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics,*, vol. 57, no. 1, pp. 76–84, Feb. 2011.

[12] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proceedings of the SustKDD Workshop on Data Mining Appliations in Sustainbility*, 2011.

[13] J. Z. Kolter and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Neural Information Processing Systems*, 2010.

[14] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," *IEEE Computer Applications in Power,*, vol. 12, no. 4, pp. 47–51, 1999.

[15] D. Rahayu, B. Narayanaswamy, S. Krishnaswamy, C. Labbé, and D. P. Seetharam, "Learning to be energy-wise: discriminative methods for load disaggregation," in *Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012*. IEEE, 2012, pp. 1–4.

[16] L. Farinaccio and R. Zmeureanu, "Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses," *Energy and Buildings*, vol. 30, no. 3, pp. 245–259, 1999.

[17] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *SDM'11*, 2011, pp. 747–758.

[18] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2012.

[19] M. J. Johnson and A. S. Willsky, "Bayesian Nonparametric Hidden Semi-Markov Models," Massachusetts Institute of Technology, Tech. Rep. arXiv:1203.1365, 2012.

[20] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Nonintrusive load monitoring using prior models of general appliance types," in *26th AAAI Conference on Artificial Intelligence*, 2012.

[21] S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in *11th International Conference on Machine Learning and Applications (ICMLA), 2012*, vol. 2. IEEE, 2012, pp. 515–520.

[22] H. Shao, M. Marwah, and N. Ramakrishnan, "A temporal motif mining approach to unsupervised energy disaggregation," in *Proceedings of 1st International Non-Intrusive Load Monitoring Workshop*, 2012.

[23] H. Ohlsson, L. Ratliff, R. Dong, and S. S. Sastry, "Blind identification of ARX models with piecewise constant inputs," *arXiv*, Mar. 2013.

[24] L. Ljung, *System Identification – Theory for the User*. Prentice Hall, 1999.

[25] F. Gustafsson, *Adaptive Filtering and Change Detection*. John Wiley & Sons, Ltd, 2000.