Control Engineering Practice 89 (2019) 30–42

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Online end-use energy disaggregation via jump linear models

Valentina Breschi [a,*], Dario Piga [b], Alberto Bemporad [c]

[a] Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy
[b] IDSIA Dalle Molle Institute for Artificial Intelligence SUPSI-USI, 6928 Manno, Switzerland
[c] IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy

ARTICLE INFO

Keywords:
Energy disaggregation
Non-intrusive load monitoring
Jump models
Dynamic programming
Kalman filter

ABSTRACT

This paper presents two iterative algorithms for non-intrusive appliance load monitoring, which aims to decompose the aggregate power consumption only measured at the household level into the contributions of the individual electric appliances. The approaches are based on modelling the total power consumption as a combination of jump linear sub-models, each of them describing the behaviour of the individual appliance. Dynamic-programming and multi-model Kalman filtering techniques are used to reconstruct the power consumptions at the single-appliance level from the aggregate power in an iterative way.

1. Introduction

Retrieving residential power consumptions at the single-appliance level provides a useful starting point to design and assess the impact of energy efficiency programs, design customized energy demand management strategies, detect malfunctioning and increase consumers' awareness on their consumption habits, to ultimately provide potential savings and economic incentives, like replacing low-efficient devices or deferring the use of some appliances to peak-off hours.

A possible solution to acquire information on end-use energy consumption is to use smart appliances or to couple smart meters with every appliance in the house. Alternatively, energy disaggregation algorithms, also known as *Non-Intrusive Load Monitoring* (NILM), decompose the aggregate power demand gathered from a single-point smart metere into the individual consumption of each appliance. The advantages of using disaggregation methods are clearly the reduction of intrusiveness into consumers' houses and lower costs for installation, maintenance and replacement of the monitoring system.

The first algorithm for non-intrusive load monitoring was proposed in Hart (1992), where a signal processing method is used to detect on/off transitions of the appliances from the active and reactive power in the total load. After detecting a transition, typical consumption patterns of the individual appliances (commonly called *signatures*) are used to detect which device has been switched on or off. Since the pioneering work of Hart, several efforts have been spent to develop efficient methods for NILM, see Esa, Abdullah, and Hassan (2016), Zeifman and Roth (2011) and Zoha, Gluhak, Imran, and Rajasegarar (2012) for a detailed review on the subject. However, most of NILM algorithms available in the literature focus on particular scenarios or are able to reach a high level of accuracy only when specific assumptions are satisfied on the monitored appliances. For instance, the approaches in Giri and Berges (2015), Hart (1992) and Yang, Soh, and Yap (2015) are tailored to binary-state appliances, and thus they can be mainly used to detect if an appliance is on or off. Methods in Chang (2012), Liang, Ng, Kendall, and Cheng (2010) and Srinivasan, Ng, and Liew (2006) are based on the analysis of the sub-harmonics of the 50/60-Hz electric signal. However, to capture this information, a high-frequency sampling rate (larger than 1 kHz) is needed. In Mejari, Naik, Piga, and Bemporad (2018) and Suzuki, Inagaki, Suzuki, Nakamura, and Ito (2008), non-intrusive load monitoring is formulated as an integer programming problem, whose computational complexity increases with the number of appliances and the number of operating modes per appliance. An optimization-based approach is also proposed in Piga, Cominola, Giuliani, Castelletti, and Rizzoli (2016), where a fitting quadratic error is minimized together with a regularization term penalizing the switch of the operating mode of the individual appliances. Since convex optimization is used in Piga et al. (2016), its computational load is lower than (Suzuki et al., 2008). However, Piga et al. (2016) requires to carefully trade off between penalization of the mode transitions and minimization of the fitting error. Approaches based on *Factorial Hidden Markov Models* (FHMMs) are discussed in Bonfigli et al. (2017), Cominola, Giuliani, Piga, Castelletti, and Rizzoli (2017) and Kolter and Jaakkola (2012). The underlying idea of these methods is to describe the behaviour of the individual appliances with an *Hidden Markov Model* (HMM), where the hidden (*i.e.*, not observed) states represent the different operating conditions of the corresponding device. Such local HMMs are then combined into a factorial hidden Markov model, which describes the overall behaviour of the system. The most likely state sequence of the FHMM describes the evolution of the configuration of the whole

[*] Corresponding author.
E-mail addresses: valentina.breschi@polimi.it (V. Breschi), dario.piga@supsi.ch (D. Piga), alberto.bemporad@imtlucca.it (A. Bemporad).

https://doi.org/10.1016/j.conengprac.2019.05.011
Received 21 December 2018; Received in revised form 7 May 2019; Accepted 15 May 2019
Available online xxxx

system over time, based on which the power demand of each appliance is reconstructed. Methods using FHMMs have shown very promising results in energy disaggregation problems. However, since the states of the factorial hidden Markov model should describe all possible combination of all appliance's states, the computational complexity of FHMM-based approaches is exponential with the number of appliances and the associated operating regimes.

Many state-of-the-art NILM approaches requires a batch of data to perform energy disaggregation. To overcome this limitation, in this paper, two iterative algorithms for non-intrusive load monitoring are proposed. The first algorithm proposed in the paper uses dynamic programming, the second one is based on multiple-model Kalman filtering. Both of them process the flow of total consumption data in an iterative way, and thus they are suited for online monitoring of consumption of individual appliances. To reduce the computational time required to perform disaggregation, strategies for complexity reduction are further proposed for both the presented approaches, so to decrease the number of configurations of the devices evaluated at each time step. Differently from many existing approaches, *e.g.,* (Hart, 1992; Yang et al., 2015), the methods presented in the paper can handle multi-state appliances. Therefore, the proposed approaches allows not only for the detection of on/off states of the devices, but also for the reconstruction of the consumption pattern of the individual appliances. Additionally, the proposed approaches do not require to use high-frequency sampling devices, as they do not exploit information on the sub-harmonics of the electric signal. Indeed, as shown by the experimental results reported in the paper, energy disaggregation can be satisfactorily performed with 1 min power readings. Similarly to NILM methods using HMMs, the proposed approaches rely on jump models to describe the behaviour of each appliance. These models are learned from the appliance's signature through the coordinate-descent optimization algorithm recently proposed in Bemporad, Breschi, Piga, and Boyd (2018). Therefore, the individual appliance is modelled through a finite collection of linear sub-models, each one representing its behaviour at a specific operating condition. Unlike most of existing disaggregation approaches, the behaviour of the appliances at different operating conditions is described with static or dynamical linear sub-models, thanks to the flexibility of the learning method in Bemporad et al. (2018). Indeed, the use of dynamical sub-models is helpful in capturing transient behaviours, by providing useful information for energy disaggregation of devices that exhibit more complex consumption patterns than on/off and, thus, enhancing the quality of the reconstructed consumption patterns.

The paper is organized as follows. After formalizing the energy disaggregation problem in Section 2, Section 3 describes the training algorithms used to estimate jump linear models for each appliance based on its signature. The problems of learning static and dynamic models are discussed. The two NILM algorithms are presented in Sections 4.1 and 4.2, along with heuristics to reduce their computational complexity. In Section 5, the developed algorithms are tested against the AMPd dataset (Makonin, Popowich, Bartram, Gill, & Bajic, 2013), which contains power consumption readings in a residential house with a time resolution of 1 min. Concluding remarks and directions for future works are given in Section 6.

2. Problem formulation

Consider N different electrical appliances available in a house and connected to the electric power line. Let $y_i(t)$, with $i \in \{1,\dots,N\}$, be the power demand of the ith appliance at time t and $y(t)$ be the household aggregate power reading, *i.e.*,

$$y(t) = \sum_{i=1}^{N} y_i(t) + e(t), \tag{1}$$

where $e(t)$ is a modelling error, accounting for additional appliances connected to the line and measurement noise on the aggregate power reading.

The problem addressed in this paper is the following:

Energy disaggregation problem Given a sequence $\{y(t)\}_{t=1}^{T}$ of observations of the aggregate power signal, estimate the power demand $y_i(t)$ of the individual appliances at each time sample t. The problem is also known as non-intrusive appliance load monitoring. ■

3. Modelling appliance behaviour

The power demand of the ith appliance is described by K_i sub-models, with $K_i \in \mathbb{N}$ being a finite natural number, each one representing the consumption behaviour of the device at a different operating condition (or *mode*). Let $s_i(t) \in \{1,\dots,K_i\}$ be the active mode of the ith appliance at time t. The power consumption of the ith device is modelled as

$$y_i(t) = \mathbf{X_i}(t)\theta_{\mathbf{i}}^{\mathbf{s_i(t)}} + e_i(t). \tag{2}$$

where $\theta_{\mathbf{i}}^{\mathbf{j}}$ and $\mathbf{X_i}(t)$ are the model parameter and the feature vector, respectively, characterizing the power demand at mode j, and $e_i(t)$ is an intrinsic modelling error. N distinct datasets $\mathcal{M}_i = \{y_i(t)\}_{t=1}^{\bar{T}}$, for $i = 1,\dots,N$, comprising consumption profiles of each device are assumed to be available to estimate the parameter vectors $\boldsymbol{\Theta}_{\mathbf{i}} = (\theta_{\mathbf{i}}^{\mathbf{1}},\dots,\theta_{\mathbf{i}}^{\mathbf{K_i}})$, $i = 1,\dots,N$. The dataset $\mathcal{M}_i$ consists of the power demand of the ith appliance gathered over a short intrusive training period of length $\bar{T}$, where it is supposed that the power consumed by the ith appliance is directly accessible during data acquisition, for instance, by switching off all the other appliances. The length $\bar{T}$ of the training period should be as short as possible to reduce intrusiveness and costs, although long enough to allow estimating $\boldsymbol{\Theta}_{\mathbf{i}}$, and it may be different for each device.

Two different classes of sub-models (2) are used to describe the consumption pattern of the appliances at a given operating mode: (i) *static models* and (ii) *dynamical models*.

3.1. Learning static models

In case *static* models are used, the feature vector is $\mathbf{X_i}(t) = 1$, $\forall\, t = 1,\dots,\bar{T}$. Therefore, the sub-model in (2) becomes the piecewise constant signal

$$y_i(t) = \theta_{\mathbf{i}}^{\mathbf{s_i(t)}}. \tag{3}$$

The parameter $\theta_{\mathbf{i}}^{\mathbf{j}}$ thus represents the power consumption of the ith appliance when at mode j, with $j = 1,\dots,K_i$.

Eq. (3) shows that the estimation of the parameters $\boldsymbol{\Theta}_{\mathbf{i}}$ requires also to reconstruct the sequence of active modes $S_i = \{s_i(t)\}_{t=1}^{\bar{T}}$. Let $\mathbb{1}(\mathcal{A})$ be the indicator function of the logic condition $\mathcal{A}$, *i.e.*,

$$\mathbb{1}(\mathcal{A}) = \begin{cases} 1 & \text{if } \mathcal{A} \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The sub-model parameters Θ_i and the mode sequence S_i are jointly estimated through the jump model fitting approach proposed by Bemporad et al. (2018), based on the minimization over $\boldsymbol{\Theta}_{\mathbf{i}}$ and S_i of the *loss function*

$$\begin{aligned} J(\boldsymbol{\Theta}_{\mathbf{i}}, S_i) = &\sum_{t=1}^{\bar{T}-1} \left(\ell\left(y_i(t), s_i(t), \boldsymbol{\Theta}_{\mathbf{i}}\right) + \lambda_i(s_i(t))\mathbb{1}(s_i(t+1)\neq s_i(t))\right) \\ &+ \ell\Big(y_i(\bar{T}), s_i(\bar{T}), \boldsymbol{\Theta}_{\mathbf{i}}\Big), \end{aligned} \tag{5}$$

where $\ell\left(y_i(t), s_i(t), \boldsymbol{\Theta}_{\mathbf{i}}\right)$ is a *fitting cost* penalizing the mismatch between the measured and the model output. Among possible fitting costs, the squared fitting error has been chosen

$$\ell\left(y_i(t), s_i(t), \boldsymbol{\Theta}_{\mathbf{i}}\right) = \frac{1}{\bar{T}}\left(y_i(t) - \theta_{\mathbf{i}}^{\mathbf{s_i(t)}}\right)^2. \tag{6}$$

The term $\lambda_i(s_i(t))\mathbb{1}(s_i(t+1) \neq s_i(t))$ in (5) takes into account prior assumptions on the switch between different modes, penalizing by a factor $\lambda_i \geq 0$ a temporal change of the operating condition from $s_i(t)$ to

Algorithm 1 Learning static models

Input: Training data set $\mathcal{M}_i = \{y_i(t)\}_{t=1}^{\bar{T}}$; number K_i of sub-models; initial mode sequence $S_i^0 = \{s_i^0(1), \dots, s_i^0(\bar{T})\}$; parameters $\lambda_i(j)$, $j = 1, \dots, K_i$.

```
1. iterate for $h = 1, \dots$
    1.1. $\boldsymbol{\Theta}_i{}^h \leftarrow \arg\min_{\boldsymbol{\Theta}_i} \sum_{t=1}^{\bar{T}} \ell\left(y_i(t), s_i^{h-1}(t), \boldsymbol{\Theta}_i\right)$; (sub-model fitting)
    1.2. $S_i^h \leftarrow \arg\min_{S_i} J(\boldsymbol{\Theta}_i^h, S_i)$;            (mode sequence fitting)
2. until $S_i^h = S_i^{h-1}$.
```

Output: Estimated model parameters $\boldsymbol{\Theta}_i^\star = \boldsymbol{\Theta}_i^k$ and mode sequences $S_i^\star = S_i^k$.

a different mode $s_i(t+1)$. This reflects the hypothesis that each device rarely changes its operating regime over time, which is a reasonable assumption in energy disaggregation problems with power readings taken at high-time resolution (*e.g.*, 1 min).

The cost $J(\boldsymbol{\Theta}_\mathbf{i}, S_i)$ in (5) is minimized through Algorithm 1, a coordinate descent approach, originally proposed in Bemporad et al. (2018), that alternates minimization with respect to the sub-model parameters $\boldsymbol{\Theta}_\mathbf{i}$ (Step 1.1) and the mode sequence S_i (Step 1.2). Since the fitting cost ℓ is chosen as the quadratic function (6), Step 1.1 is solved analytically by least squares, while Step 1.2 is solved by discrete *dynamic programming* (DP). The initial mode sequences S_i^0 are chosen randomly, and the values for the parameters $\lambda_i(j)$ are set to 0 at the first iteration, while for the following runs they are chosen as $\lambda_i(j) = 0.9$, $\forall j \in \{1, \dots, K_i\}$. Technical details on the implementation of Algorithm 1 can be found in Bemporad et al. (2018).

3.2. Learning dynamic models

Since some appliances exhibit a transient behaviour, their power consumption patterns can be more accurately described by dynamical sub-models instead of static ones. For instance, a transient behaviour can be clearly observed in the consumption patterns of a fridge and a heat pump , which show typical dynamics of second and first-order *Linear Time-Invariant* (LTI) systems, respectively, as it can seen in Fig. 1.

Provided that the sequence of active modes S_i in (2) is given, the LTI dynamical sub-model associated to each mode can be estimated through standard system identification techniques (Ljung, 1999). Specifically, the power consumption $y_i(t)$ is modelled as (2), with regressor $\mathbf{X}_\mathbf{i}(t) = [\tilde{\mathbf{X}}_\mathbf{i}(t) \;\; 1]$, where $\tilde{\mathbf{X}}_\mathbf{i}(t)$ consists of past output samples, *i.e.*,

$$\tilde{\mathbf{X}}_\mathbf{i}(t) = [y_i(t-1) \;\; \dots \;\; y_i(t-n)], \tag{7}$$

with $n \in \mathbb{N}$ defining the dynamical order of the model. The parameters $\boldsymbol{\Theta}_\mathbf{i}$ are estimated by solving the simulation-error minimization problem

$$\min_{\boldsymbol{\Theta}_\mathbf{i}} \sum_{t=n+1}^{\bar{T}} \left(y_i(t) - \hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i(t))\right)^2, \tag{8}$$

where $\hat{y}_i(t)$ is the simulated output given by

$$\hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i(t)) = [\hat{y}_i(t-1, \boldsymbol{\Theta}_\mathbf{i}, s_i(t-1)) \;\; \dots \;\; \hat{y}_i(t-n, \boldsymbol{\Theta}_\mathbf{i}, s_i(t-n)) \;\; 1]\theta_\mathbf{i}^{\mathbf{s_{i(t)}}}. \tag{9}$$

Because of the nested dependence of $\hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i(t))$ on the model parameters $\boldsymbol{\Theta}_\mathbf{i}$, the optimization problem in (8) is non-convex and solved through Particle Swarm Optimization (Poli, Kennedy, & Blackwell, 2007). Since the sequence of active mode S_i in (8) is actually not known, Algorithm 1 is first run to estimate S_i using static sub-models.

The power consumption for a fridge and a heat pump estimated using static and second-order dynamical models are plotted in Fig. 2, along with the actual consumption profiles. The reported results show the capabilities of dynamical models to reconstruct the transient of the considered appliances. Similar performance is obtained for other electrical devices.

4. Disaggregation algorithms

Once the models of each appliance have been estimated, the energy disaggregation problem requires to detect the active mode $s_i(t)$ for each appliance at each time instant from the aggregate reading $y(t)$. In this section, two different algorithms for energy disaggregation are described, which process data iteratively and thus are suited for an online implementation.

In the following, the *joint active mode* $\mathbf{s}(t) \in \mathbb{N}^N$ is the vector stacking the appliances' modes at time t, *i.e.*, $\mathbf{s}(t) = [s_1(t), \;\; \dots, \;\; s_N(t)]$, $\mathcal{S}$ is the set of all possible combinations taken by $\mathbf{s}(t)$ and $|\mathcal{S}|$ indicates the cardinality of $\mathcal{S}$, *i.e.*, $|\mathcal{S}| = \prod_{i=1}^N K_i$.

4.1. Disaggregation based on dynamic-programming

The first algorithm for iterative energy disaggregation proposed in this paper is based on the minimization of the *loss function*:

$$J(t, \mathbf{s}(t)) = \ell(y(1), \mathbf{s}(1)) + \left[\sum_{\tau=2}^{t} \ell(y(\tau), \mathbf{s}(\tau)) + \lambda(\mathbf{s}(\tau-1))\mathbb{1}(\mathbf{s}(\tau) \neq \mathbf{s}(\tau-1))\right], \tag{10}$$

which penalizes, similarly to (5), the time switch of the joint mode, as well as the fitting error

$$\ell(y(t), \mathbf{s}(t)) = \left(y(t) - \sum_{i=1}^{N} \hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, \mathbf{s}_i(t))\right)^2$$

on the aggregate power measurement $y(t)$. Since disaggregation should be performed in real time, this formulation accounts for the fact that only the aggregate readings up to time t can be used to reconstruct the active mode $\mathbf{s}(t)$. The same assumption introduced in Section 3, namely the hypothesis that the appliances rarely change their operating mode over time, has been taken into account in penalizing transitions on the joint mode. The parameter $\lambda(d)$ in (10), with $d \in \mathcal{S}$, is proportional to the empirical probability of remaining in mode d for two consecutive time instants. Indeed, under the assumption that the appliances change their modes independently from each other, the value of $\lambda(d)$ is selected as

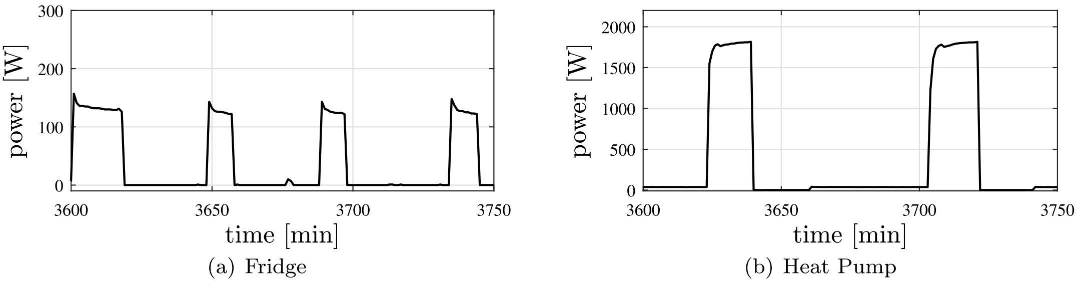
$$\lambda(d) = w \prod_{i=1}^{N} \lambda_i(d_i), \;\; w > 0, \tag{11}$$

where w is a tunable parameter and $\lambda_i(d_i)$ are the empirical probabilities

$$\lambda_i(j) = \frac{\sum_{t=1}^{\bar{T}-1} \mathbb{1}(s_i(t) = j \;\&\; s_i(t+1) = j) + 1}{\sum_{t=1}^{\bar{T}-1} \mathbb{1}(s_i(t) = j) + K_i^2}, \quad j = 1, \dots, K_i, \tag{12}$$

with $i = 1, \dots, N$ and $j = 1, \dots, K_i$. Unlike the minimization of (5), the cost (10) has not to be optimized with respect to the model parameters $\boldsymbol{\Theta}_\mathbf{i}$, that are known once the model of each appliance is learned as explained in Section 3, but only with respect to the active mode sequence $\{\mathbf{s}(\tau)\}_{\tau=1}^{t}$.

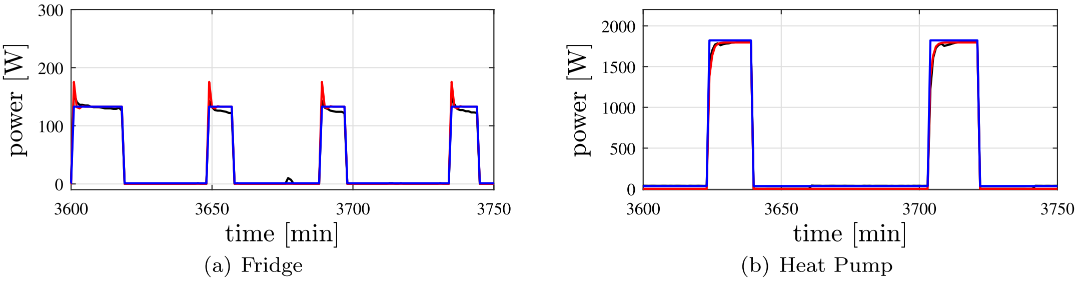
The minimization of the cost $J(t, \mathbf{s}(t))$ in (10) is performed as described in Algorithm 2. Specifically, for each possible joint mode $h \in \mathcal{S}$, the value of $J(1, h)$ is computed at Step 1, and the joint mode $\mathbf{s}(1)$ is selected as the one minimizing $J(1, h)$ over $h \in \mathcal{S}$ (Step 2). At time $t \geq 2$, the optimal costs $J^*(t, h)$, for $h \in \mathcal{S}$, are updated based on the previously computed optimal costs $J^*(t-1, d)$ and the current aggregate power measurement $y(t)$ (Step 3.1). The optimal joint mode $\mathbf{s}^*(t)$ is finally computed at Step 3.2.



(a) Fridge

(b) Heat Pump

Fig. 1. Power consumption profiles of a fridge and a heat pump.



(a) Fridge

(b) Heat Pump

Fig. 2. Power consumption profiles: true (black), estimated with static sub-models (blue), estimated with dynamic sub-models (red). Black and red lines almost overlap. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

```
Algorithm 2 Dynamic-programming based disaggregation
 Input: Aggregate output readings flow $y(1), y(2), \ldots$; model parameters $\boldsymbol{\Theta}_i$, $i = 1, \ldots, N$; parameters $\lambda(d)$, $d \in \mathcal{S}$.

  1. $J^*(1,h) \leftarrow \ell(y(1),h);\ \ h \in \mathcal{S}$;
  2. $\mathbf{s}^*(1) \leftarrow \arg\min_{h\in\mathcal{S}} J^*(1,h)$;
  3. iterate for $t = 2, \ldots$
       3.1. $J^*(t,h) \leftarrow \ell(y(t),h) + \min_{d\in\mathcal{S}} (J^*(t-1,d) + \lambda(d)\mathbb{1}(h \neq d))$, $h \in \mathcal{S}$;
       3.2. $\mathbf{s}^*(t) \leftarrow \arg\min_{h\in\mathcal{S}} J^*(t,h)$;

 Output: Estimated joint mode sequence $\mathbf{s}^*(t)$.
```

It is worth remarking that, in case static models are used, the dynamic programming approach in Algorithm 2 provides the optimal mode $\mathbf{s}^*(t)$ minimizing the cost $J(t, \mathbf{s}(t))$ in (10), given the information up to time t. If dynamical models are used, the fitting cost $\ell(y(t), h)$ is computed by approximating the individual appliance consumptions with the previous estimates $\hat{y}_i(t-1), \ldots, \hat{y}_i(t-n)$. This implies that the history up to time $t-1$ is embedded into $\hat{y}_i(t-1), \ldots, \hat{y}_i(t-n)$, thus leading to an approximation of the optimum of the objective function $J(t,h)$. The tth iteration of Algorithm 2 is further schematized in Fig. 3. It is worth remarking that the update of the cost $J^*(t,h)$ at Step 3.1 is recursively computed based on $J^*(t-1,d)$, $d \in \mathcal{S}$ and the new observation $y(t)$, without the need to store and reprocess past data. This makes Algorithm 2 suited for online disaggregation.

Note that, at each time step t, Algorithm 2 computes the "cost-to-go" $J^*(t,h)$ at mode h for all possible values of $h \in \mathcal{S}$, which requires to evaluate the cost $J^*(t-1,d) + \lambda(d)\mathbf{1}(h \neq d)$ for all possible values of $d \in \mathcal{S}$ (Step 3.1). Thus, the computational complexity of Algorithm 2 in detecting the joint mode $\mathbf{s}(t)$ is $O(|\mathcal{S}|^2)$. Since the number $|\mathcal{S}|$ of possible combinations of the joint mode $s(t)$ increases exponentially with the number of appliances N and the number of operating conditions K_i, $i = 1, \ldots, N$, the implementation of Algorithm 2 might not be practically feasible in case of large N and K_i, $i = 1, \ldots, N$.

Complexity reduction

A possible way of reducing the complexity of Algorithm 2 is to embed the past history of the appliances into the last estimate of the joint mode. Instead of evaluating the cost $J^\star(t-1,d) + \lambda(d)\mathbb{1}(h \neq d)$ for all possible $d \in \mathcal{S}$, Step 3.1 is thus approximated as

$$J^*(t,h) \leftarrow \ell(y(t),h) + J^*(t-1, \mathbf{s}^*(t-1)) + \lambda(\mathbf{s}^*(t-1))\mathbb{1}\left(h \neq \mathbf{s}^*(t-1)\right),\ h \in \mathcal{S} \tag{13}$$

where $\mathbf{s}^*(t-1)$ is the estimate of the joint mode at the previous time step $t-1$, given by

$$\mathbf{s}^*(t-1) = \arg\min_{h\in\mathcal{S}} J^*(t-1,h). \tag{14}$$

Since at time t only the past configuration $\mathbf{s}^*(t-1)$ is considered and $J^*(t,h)$ has to be evaluated for all possible $h \in \mathcal{S}$, the approach resulting from approximation (13), schematized in Fig. 4, leads to a reduction of the computational complexity from $O(|\mathcal{S}|^2)$ to $O(|\mathcal{S}|)$.

To further reduce the complexity of Algorithm 2, the cost $J^*(t,h)$ in Step 3.1 is not computed for all possible modes $h \in \mathcal{S}$, but only for ones satisfying at least one of the following conditions:

- (C_1) at most one appliance changes its operating condition;
- (C_2) all the appliances are in the operating condition corresponding to their minimal consumption energy (namely, all of them are off);
- (C_3) only one appliance is on;
- (C_4) the joint mode is equal to $s^*(t-1)$.

These assumptions are realistic in energy disaggregation problems with high-time resolution (*e.g.*, 1 min) power readings. Although (C_2)–(C_3) consider the cases where at most one appliance is on, condition (C_1) allows us to handle configurations where multiple appliances are consuming. When two devices are switched on simultaneously at time t, the operating mode of one of the two appliances cannot be correctly detected, as this configuration is not accounted for by (C_1)–(C_4). Nevertheless, the actual operating mode is expected to be retrieved at time $t+1$ thanks to condition (C_1). Since only the configurations satisfying (C_1), (C_2), (C_3) or (C_4) are considered, the computational complexity of the approach is further reduced and, in particular, it

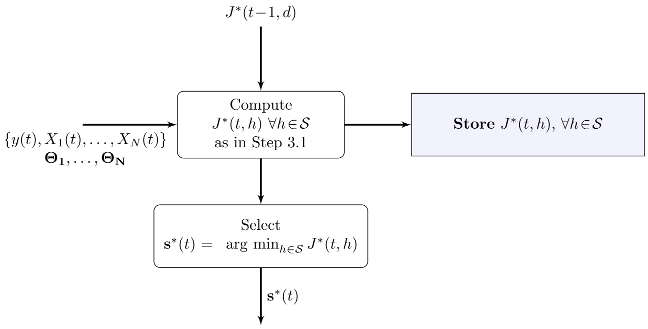

$J^*(t-1,d)$
$\{y(t), X_1(t), \ldots, X_N(t)\}$
$\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_N$
Compute
$J^*(t,h)\ \forall h \in \mathcal{S}$
as in Step 3.1
Store $J^*(t,h), \forall h \in \mathcal{S}$
Select
$\mathbf{s}^*(t) = \arg\min_{h\in\mathcal{S}} J^*(t,h)$
$\mathbf{s}^*(t)$

Fig. 3. Dynamic-programming based disaggregation: iteration at time t.

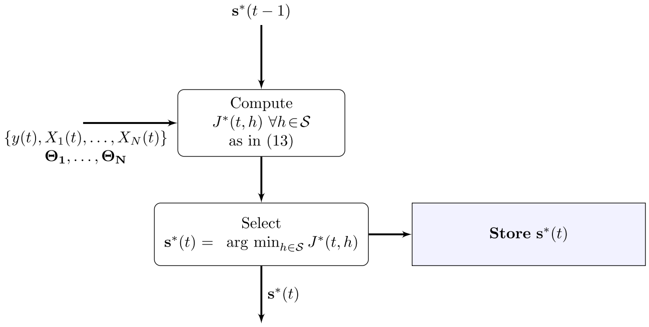

$\mathbf{s}^*(t-1)$
$\{y(t), X_1(t), \ldots, X_N(t)\}$
$\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_N$
Compute
$J^*(t,h)\ \forall h \in \mathcal{S}$
as in (13)
Select
$\mathbf{s}^*(t) = \arg\min_{h\in\mathcal{S}} J^*(t,h)$
Store $\mathbf{s}^*(t)$
$\mathbf{s}^*(t)$

Fig. 4. Schematic of the approximation used to reduce complexity of dynamic-programming based disaggregation from $O(|\mathcal{S}|^2)$ to $O(|\mathcal{S}|)$. All the past history up to time $t-1$ is embedded in $\mathbf{s}^*(t-1)$.

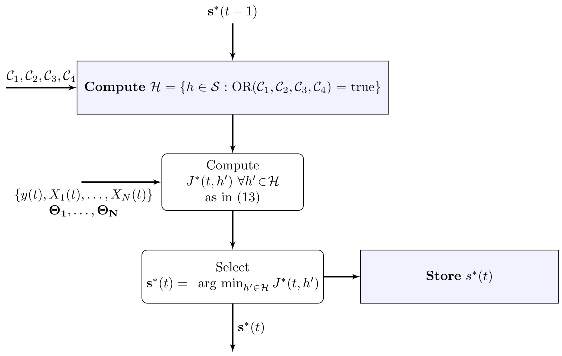

$\mathbf{s}^*(t-1)$
$\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$
Compute $\mathcal{H} = \{h \in \mathcal{S} : \mathrm{OR}(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4) = \text{true}\}$
$\{y(t), X_1(t), \ldots, X_N(t)\}$
$\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_N$
Compute
$J^*(t,h')\ \forall h' \in \mathcal{H}$
as in (13)
Select
$\mathbf{s}^*(t) = \arg\min_{h'\in\mathcal{H}} J^*(t,h')$
Store $s^*(t)$
$\mathbf{s}^*(t)$

Fig. 5. Schematic of the approximation used to further reduce complexity of dynamic-programming based disaggregation from $O(|\mathcal{S}|)$, where only configurations satisfying at least one of the conditions (C_1), (C_2), (C_3) or (C_4) are considered.

is upper-bounded by $O\left(2+\sum_{i=1}^{N} K_i\right)$. A schematic of the operations performed at time t is shown in Fig. 5.

4.2. Disaggregation via Kalman filtering

An alternative iterative approach is proposed, which is based on a reformulation of energy disaggregation as a state estimation problem for switching linear dynamical systems. Once the disaggregation problem is properly reformulated, it is then solved via multi-model Kalman filtering (Bar-Shalom, Kirubarajan, & Li, 2002, Chapetr 11).

A state-space representation of the switching linear model representing the household power consumption over time is given by:

$$\mathbf{x}(t+1) = \mathbf{A}[\mathbf{s}(t)]\mathbf{x}(t) + \mathbf{B}[\mathbf{s}(t)] + \mathbf{w}[\mathbf{s}(t)](t) \tag{15a}$$

$$y(t) = \mathbf{C}[\mathbf{s}(t)]\mathbf{x}(t) + v[\mathbf{s}(t)](t), \tag{15b}$$

where $y(t)$ is the measured aggregate power, $\mathbf{w}[\mathbf{s}(t)](t)$ and $v[\mathbf{s}(t)](t)$ are the process and measurement noise, respectively, which are supposed to be white and mutually independent. The noise terms $\mathbf{w}[\mathbf{s}(t)](t)$ and $v[\mathbf{s}(t)](t)$ are assumed to be generated by zero-mean Gaussian distributions with covariance matrices $\mathbf{Q}[\mathbf{s}(t)]$ and $R[\mathbf{s}(t)]$, *i.e.*, $\mathbf{w}[\mathbf{s}(t)](t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}[\mathbf{s}(t)])$ and $v[\mathbf{s}(t)](t) \sim \mathcal{N}(0, R[\mathbf{s}(t)])$. Similarly to Section 4.1, $\mathbf{s}(t) \in \mathcal{S}$ represents the joint mode at time $\mathbf{t}$. The continuous state $\mathbf{x}(t)$ and the matrices $\mathbf{A}[\mathbf{s}(t)]$, $\mathbf{B}[\mathbf{s}(t)]$, $\mathbf{C}[\mathbf{s}(t)]$ are properly defined based on the static/dynamic models of each device estimated as in Section 3. For example, in the case each appliance is described by static submodels (3), the state $\mathbf{x}(t)$ is the collection of the individual appliances' consumptions $\mathbf{x}(t) = [\, y_1(t) \ \ldots \ y_N(t)\,]'$ and

$$\mathbf{A}[\mathbf{s}(t)] = \mathbf{0}_{N,N} \qquad \mathbf{B}[\mathbf{s}(t)] = \begin{bmatrix} \theta_1^{s_1(t)} \\ \vdots \\ \theta_N^{s_N(t)} \end{bmatrix} \qquad \mathbf{C}[\mathbf{s}(t)] = \mathbf{1}_N',$$

with $\mathbf{0}_{N,N}$ being a zero squared matrix of size N, $\mathbf{1}_N$ being a unitary column vector of dimension N and $\mathbf{1}_N'$ indicating its transpose. The time evolution of the mode $\mathbf{s}(t)$ is described by a stationary *Markov Chain* with transition probabilities $P(\mathbf{s}(t) = h|\mathbf{s}(t-1) = d)$, with $h, d \in \mathcal{S}$. These transitions probabilities are approximated from the results of the training procedure described in Section 3.1, using the empirical probabilities with Laplace smoothing:

$$P(s_i(t) = h_i | s_i(t-1) = d_i) = \frac{\sum_{t=2}^{\bar{T}} \mathbb{1}(s_i(t) = h_i \ \&\ s_i(t-1) = d_i) + 1}{\sum_{t=2}^{\bar{T}} \mathbb{1}(s_i(t-1) = d_i) + K_i^2}, \tag{16}$$

for $i = 1, \ldots, N$. Under the hypothesis that the appliances change their mode independently from each other, the transition probabilities $P(\mathbf{s}(t) = h|\mathbf{s}(t-1) = d)$ are computed as

$$P(\mathbf{s}(t) = h|\mathbf{s}(t-1) = d) = \prod_{i=1}^{N} P(s_i(t) = h_i | s_i(t-1) = d_i). \tag{17}$$

Multi-model Kalman filtering techniques can be used to simultaneously estimate both the joint mode $\mathbf{s}(t)$ and the continuous state $\mathbf{x}(t)$ of the dynamical system in (15). In this work, the *first-order generalized pseudo-Bayesian* (GPB$_1$) algorithm (Bar-Shalom et al., 2002, Chapter 11) is employed. The main ideas behind GPB$_1$ are summarized in the rest of this section. Heuristics to reduce the computational complexity in applying GPB$_1$ to energy disaggregation problems are then described at the end of Section 4.2.

Let $\mathcal{I}^t$ be the set of available data up to time t, i.e., $\mathcal{I}^t = \{y(1), \ldots, y(t)\}$, $t \in \mathbb{N}$. At each time t, GPB$_1$ approximates the state conditional probability density function $p[\mathbf{x}(t)|\mathcal{I}^t]$ as:

$$\begin{aligned} p[\mathbf{x}(t)|\mathcal{I}^t] &= \sum_{h=1}^{|\mathcal{S}|} p[\mathbf{x}(t)|\mathbf{s}(t) = h, \mathcal{I}^t] P(\mathbf{s}(t) = h|\mathcal{I}^t) \\ &= \sum_{h=1}^{|\mathcal{S}|} p[\mathbf{x}(t)|\mathbf{s}(t) = h, y(t), \mathcal{I}^{t-1}] P(\mathbf{s}(t) = h|\mathcal{I}^t) \end{aligned}$$

$$\approx \sum_{h=1}^{|\mathcal{S}|} p[\mathbf{x}(t)|\mathbf{s}(t)=h, y(t), \hat{x}(t-1|t-1), \mathbf{P}_x(t-1|t-1)]\cdot \\ \cdot P(\mathbf{s}(t)=h|\mathcal{I}^t). \tag{18}$$

The idea behind this approximation is to embed the past information $\mathcal{I}^{t-1}$ on the system into the state estimate $\hat{\mathbf{x}}(t-1|t-1)$ computed at time $t-1$ (using only information up to $t-1$) and the associated covariance matrix $\mathbf{P}_x(t-1|t-1)$.

By assuming that the initial state $\mathbf{x}(0)$ is Gaussian with mean $\hat{\mathbf{x}}(0|0)$ and covariance $\mathbf{P}_x(0|0)$, the probability density function

$$p[\mathbf{x}(t)|\mathbf{s}(t)=h, y(t), \hat{\mathbf{x}}(t-1|t-1), \mathbf{P}_x(t-1|t-1)]$$

is Gaussian with mean $\hat{\mathbf{x}}[h](t|t)$ and covariance $\mathbf{P}_x[h](t|t)$, where $\hat{\mathbf{x}}[h](t|t)$ and $\mathbf{P}_x[h](t|t)$ are the output of the Kalman filter associated with the linear sub-model (15) for $\mathbf{s}(t)=h$.

Thus, $p[\mathbf{x}(t)|\mathcal{I}^t]$ in (18) is a Gaussian mixture, with weights

$$\alpha[h](t|t) = P(\mathbf{s}(t)=h|\mathcal{I}^t), \tag{19}$$

where $\alpha[h](t|t)$ represents the probability of being at mode h at time t, given the information up to time t. The weight $\alpha[h](t|t)$ can be equivalently expressed as

$$\alpha[h](t|t) = P(\mathbf{s}(t)=h|y(t),\mathcal{I}^{t-1}) \\ = \frac{p[y(t)|\mathbf{s}(t)=h,\mathcal{I}^{t-1}]P(\mathbf{s}(t)=h|\mathcal{I}^{t-1})}{\sum_{j\in\mathcal{S}} p[y(t)|\mathbf{s}(t)=j,\mathcal{I}^{t-1}]P(\mathbf{s}(t)=j|\mathcal{I}^{t-1})}, \tag{20}$$

where $P(\mathbf{s}(t)=h|\mathcal{I}^{t-1})$ is the probability of being at mode h at time t given the measurements up to time $t-1$, and it can be computed iteratively as

$$\alpha[h](t|t-1) = P(\mathbf{s}(t)=h|\mathcal{I}^{t-1}) = \sum_{d\in\mathcal{S}} P(\mathbf{s}(t)=h|\mathbf{s}(t-1)=d)\alpha[d](t-1|t-1), \tag{21}$$

with $P(\mathbf{s}(t)=h|\mathbf{s}(t-1)=d)$ given by (17), for all $h\in\mathcal{S}$.

By embedding the past information $\mathcal{I}^{t-1}$ into $\hat{\mathbf{x}}(t-1|t-1)$ and $\mathbf{P}_x(t-1|t-1)$ as in (18), the conditional likelihood $p[y(t)|\mathbf{s}(t)=h,\mathcal{I}^{t-1}]$ of the aggregate output $y(t)$ can be approximated as

$$p[y(t)|\mathbf{s}(t)=h,\mathcal{I}^{t-1}] \approx p[y(t)|\mathbf{s}(t)=h,\hat{\mathbf{x}}(t-1|t-1),\mathbf{P}_x(t-1|t-1)]. \tag{22}$$

Using the dynamical equations (15) and prior assumptions on the distributions of $\mathbf{x}(0)$, $\mathbf{w}[h](t)$ and $v[h](t)$, the approximated likelihood in (22) is Gaussian with mean $\mathbf{C}[h]\hat{\mathbf{x}}[h](t|t-1)$ and covariance $R[h]+\mathbf{C}[h]\mathbf{P}_x[h](t|t-1)\mathbf{C}[h]'$, where

$$\hat{\mathbf{x}}[h](t|t-1) = \mathbf{A}[h]\hat{\mathbf{x}}(t-1|t-1)+\mathbf{B}[h] \\ \mathbf{P}_x[h](t|t-1) = \mathbf{A}[h]'\mathbf{P}_x(t-1|t-1)\mathbf{A}[h]+\mathbf{Q}[h].$$

Summarizing, the weights of the Gaussian mixture $p[\mathbf{x}(t)|\mathcal{I}^t]$ in (18) are calculated using (20) and (22). The state estimate $\hat{\mathbf{x}}(t|t)$ and the associated covariance $\mathbf{P}_x(t|t)$ are then chosen as the expected value and covariance matrix of the random variable $\mathbf{x}\sim p[\mathbf{x}|\mathcal{I}^t]$, namely:

$$\hat{\mathbf{x}}(t|t) = \sum_{h\in\mathcal{S}} \hat{\mathbf{x}}[h](t|t)\alpha[h](t), \tag{23}$$

$$\mathbf{P}(t|t) = \sum_{h\in\mathcal{S}} \alpha[h](t|t)\left\{\mathbf{P}[h](t|t)+[\hat{\mathbf{x}}[h](t|t)-\hat{\mathbf{x}}(t|t)][\hat{\mathbf{x}}[h](t|t)-\hat{\mathbf{x}}(t|t)]'\right\}. \tag{24}$$

The active mode at time t is finally selected as

$$\mathbf{s}^*(t) = \operatorname*{argmax}_{h\in\mathcal{S}} \alpha[h](t|t), \tag{25}$$

and the final disaggregated power of each appliance is retrieved from the estimated state $\hat{\mathbf{x}}[\mathbf{s}^*(t)](t|t)$.

Algorithm 3 summarizes the iterations of the Kalman filter based disaggregation approach. If no prior on the initial mode probabilities $\alpha[h](0|0) = P(s(0)=h)$ is available, Algorithm 3 can be initialized by setting $\alpha[h](0|0) = \frac{1}{|\mathcal{S}|}$ for all $h\in\mathcal{S}$. This is equivalent to assume that the initial mode $\mathbf{s}(0)$ has a uniform probability distribution. The tth step of Algorithm 3 is further schematized in Fig. 6. Note that, at each iteration t, the computations in Steps 1.1–1.3 can be recursively performed based on $\hat{\mathbf{x}}(t-1|t-1), \mathbf{P}_x(t-1|t-1)$ and the new observation $y(t)$, without the need to store and reprocess past data. Thus, like Algorithm 2, also Algorithm 3 is suited for online disaggregation.

Algorithm 3 Kalman filter based disaggregation

Input: Aggregate output readings flow $y(1), y(2), \ldots$; models $\mathbf{A}[h], \mathbf{B}[h], \mathbf{C}[h]$ and noise covariance matrices $\mathbf{Q}[h], R[h]$, $h\in\mathcal{S}$; prior on the initial state $\hat{\mathbf{x}}(0|0)$, $\mathbf{P}_x(0|0)$, initial mode probabilities $\alpha[h](0|0)$, $h\in\mathcal{S}$.

```
1. iterate for $t = 1, 2, \ldots$
    1.1. update $\hat{\mathbf{x}}[h](t|t), \mathbf{P}_x[h](t|t)$, $h\in\mathcal{S}$, using linear Kalman filter;
    1.2. compute the likelihood $p[y(t)|\mathbf{s}(t)=h,\mathcal{I}^{t-1}]$, $h\in\mathcal{S}$, as in Eq. (22);
    1.3. update $\alpha[h](t|t)$, $h\in\mathcal{S}$, as in Eq. (20);
    1.4. $\mathbf{s}^*(t) \leftarrow \text{argmax}_{h\in\mathcal{S}}\, \alpha[h](t|t)$, $h\in\mathcal{S}$;
    1.5. compute $\hat{\mathbf{x}}(t|t)$ as in Eq. (23);
    1.6. compute $\mathbf{P}_x(t|t)$ as in Eq. (24);
```

Output: Estimated sequence of joint mode $\mathbf{s}^*(t)$ and optimal state $\hat{\mathbf{x}}[\mathbf{s}^*(t)](t|t)$.

Since Step 1.1 of Algorithm 3 should be performed for each $h\in\mathcal{S}$, it requires $|\mathcal{S}|$ Kalman filters to run in parallel. As the number $|\mathcal{S}|$ of possible joint modes $\mathbf{s}(t)$ increases exponentially with the number of appliances N and the number of corresponding operating conditions K_i, $i=1,\ldots,N$, the approach is limited to disaggregation problems with few devices and with a small number of operating regimes for each appliance.

Complexity reduction

To reduce the computational complexity of Algorithm 3 and, thus, the computational time needed to disaggregate energy, Step 1.1 (which requires to run $|\mathcal{S}|$ Kalman filters in parallel) is performed at time t only for the modes h' in $\mathcal{S}$ satisfying at least one of the following conditions:

($\mathcal{K}_1$) The probability $\alpha[h'](t|t-1) = P(\mathbf{s}(t)=h'|\mathcal{I}^{t-1})$ is larger than a threshold ϵ, *i.e.*,

$$\alpha[h'](t|t-1) \geq \epsilon.$$

A possible value for ϵ is $1/|\mathcal{S}|$;

($\mathcal{K}_2$) all the appliances are in the operating condition corresponding to their minimal consumption energy (namely, all of them are off);

($\mathcal{K}_3$) only one appliance is on;

($\mathcal{K}_4$) the joint mode is equal to $s^*(t-1)$.

Conditions ($\mathcal{K}_2$)–($\mathcal{K}_4$) are equal to conditions ($\mathcal{C}_2$)–($\mathcal{C}_4$) already used in Section 4.1 to reduce the computational complexity of the dynamic programming based approach. Condition ($\mathcal{K}_1$) allows us to discard the configurations which are predicted to be "unlikely". The operations to be performed at time t are summarized in Fig. 7.

5. Experimental tests

The proposed disaggregation algorithms are tested against the AMPds dataset (Makonin et al., 2013), which consists of the power readings of a house located in Canada, and it comprises the consumption profiles of 19 appliances, recorded over a year (from April 1, 2012 to March 31, 2013) at one-minute time resolution. The goal of the

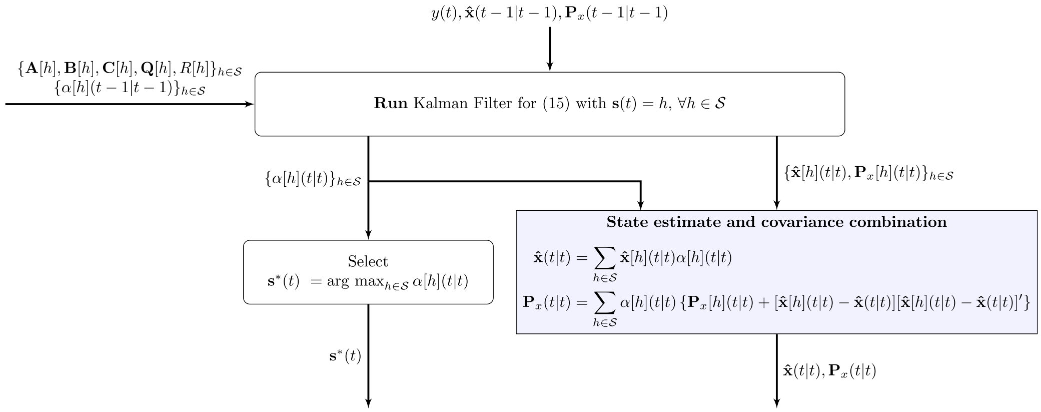

$y(t), \hat{\mathbf{x}}(t-1|t-1), \mathbf{P}_x(t-1|t-1)$
$\{\mathbf{A}[h], \mathbf{B}[h], \mathbf{C}[h], \mathbf{Q}[h], R[h]\}_{h\in\mathcal{S}}$
$\{\alpha[h](t-1|t-1)\}_{h\in\mathcal{S}}$
Run Kalman Filter for (15) with $\mathbf{s}(t) = h$, $\forall h \in \mathcal{S}$
$\{\alpha[h](t|t)\}_{h\in\mathcal{S}}$
$\{\hat{\mathbf{x}}[h](t|t), \mathbf{P}_x[h](t|t)\}_{h\in\mathcal{S}}$
State estimate and covariance combination
$\hat{\mathbf{x}}(t|t) = \sum_{h\in\mathcal{S}} \hat{\mathbf{x}}[h](t|t)\alpha[h](t|t)$
$\mathbf{P}_x(t|t) = \sum_{h\in\mathcal{S}} \alpha[h](t|t)\{\mathbf{P}_x[h](t|t) + [\hat{\mathbf{x}}[h](t|t) - \hat{\mathbf{x}}(t|t)][\hat{\mathbf{x}}[h](t|t) - \hat{\mathbf{x}}(t|t)]'\}$
Select
$\mathbf{s}^*(t) = \arg\max_{h\in\mathcal{S}} \alpha[h](t|t)$
$\mathbf{s}^*(t)$
$\hat{\mathbf{x}}(t|t), \mathbf{P}_x(t|t)$

Fig. 6. Kalman filter based disaggregation approach: iteration at step t.

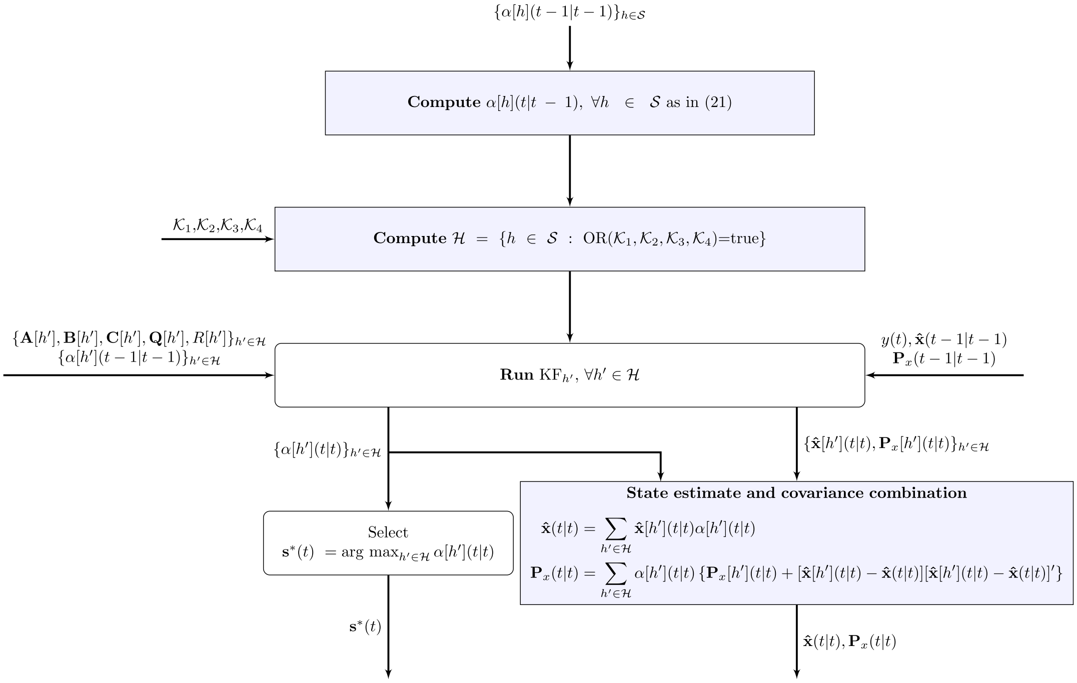

$\{\alpha[h](t-1|t-1)\}_{h\in\mathcal{S}}$
Compute $\alpha[h](t|t-1)$, $\forall h \in \mathcal{S}$ as in (21)
$\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4$
Compute $\mathcal{H} = \{h \in \mathcal{S} : \text{OR}(\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \mathcal{K}_4) = \text{true}\}$
$\{\mathbf{A}[h'], \mathbf{B}[h'], \mathbf{C}[h'], \mathbf{Q}[h'], R[h']\}_{h'\in\mathcal{H}}$
$\{\alpha[h'](t-1|t-1)\}_{h'\in\mathcal{H}}$
Run $\text{KF}_{h'}$, $\forall h' \in \mathcal{H}$
$y(t), \hat{\mathbf{x}}(t-1|t-1)$
$\mathbf{P}_x(t-1|t-1)$
$\{\alpha[h'](t|t)\}_{h'\in\mathcal{H}}$
$\{\hat{\mathbf{x}}[h'](t|t), \mathbf{P}_x[h'](t|t)\}_{h'\in\mathcal{H}}$
State estimate and covariance combination
$\hat{\mathbf{x}}(t|t) = \sum_{h'\in\mathcal{H}} \hat{\mathbf{x}}[h'](t|t)\alpha[h'](t|t)$
$\mathbf{P}_x(t|t) = \sum_{h'\in\mathcal{H}} \alpha[h'](t|t)\{\mathbf{P}_x[h'](t|t) + [\hat{\mathbf{x}}[h'](t|t) - \hat{\mathbf{x}}(t|t)][\hat{\mathbf{x}}[h'](t|t) - \hat{\mathbf{x}}(t|t)]'\}$
Select
$\mathbf{s}^*(t) = \arg\max_{h'\in\mathcal{H}} \alpha[h'](t|t)$
$\mathbf{s}^*(t)$
$\hat{\mathbf{x}}(t|t), \mathbf{P}_x(t|t)$

Fig. 7. Scheme of the reduced-complexity KF-based approach reporting the operations performed at step t.

test is to assess: (i) the capabilities of the proposed disaggregation algorithms in reconstructing the end-use power consumption from the aggregate readings; (ii) their robustness against modelling errors and against changes in the consumer behaviour due to seasonality; (iii) their performance with respect to state-of-the-art disaggregation methods; (iv) their computational complexity. When iterating Algorithm 2, the hyper-parameter w in (11) is set to 100. In running Algorithm 3 the covariance matrices $\mathbf{P}_x(0|0)$, $\mathbf{Q}[h]$ and $R[h]$ are chosen as diagonal matrices with non-zero entries equal to 1000, 10 and 800, respectively. The initial parameter $\hat{\mathbf{x}}(0|0)$ is a zero vector and the initial probabilities $\alpha[h](0|0)$ are set to $\frac{1}{|\mathcal{S}|}$, for all $h \in \mathcal{S}$. The threshold ϵ characterizing condition ($\mathcal{K}_1$) (Section 4.2) is equal to $\frac{1}{|\mathcal{S}|}$.

All tests are carried out on a MacBook Pro 2.8 GHz-Intel i7 in MATLAB R2018b.

5.1. Learning appliance behaviour

The data collected over the first 14 days (from April 1, 2012 to April 14, 2012) are used to construct training sets $\mathcal{M}_i$ of length $\bar{T} = 20160$, $i = 1, \dots, N$, needed to estimate the models for the individual devices. As discussed in Section 3, in the training phase it is assumed to have access to the consumption patterns of the individual appliances.

The following appliances are modelled: cloth dryer (CDE); dishwasher (DWE); fridge (FGE); heat pump (HPE) and basements plugs & lights (BME). First, static models (3) for each of the considered devices are estimated. The following parameters $\theta_{\mathbf{i}}^{\mathbf{j}}$ are obtained by using Algorithm 1.

$$\text{CDE}: \begin{bmatrix} 2.2 & 4586.6 \end{bmatrix},$$
$$\text{DWE}: \begin{bmatrix} 0.6 & 753.3 \end{bmatrix},$$
$$\text{FGE}: \begin{bmatrix} 1.1 & 135.5 \end{bmatrix},$$
$$\text{HPE}: \begin{bmatrix} 30.7 & 1769.1 \end{bmatrix},$$
$$\text{BME}: \begin{bmatrix} 7.2 & 343.9 \end{bmatrix}.$$

As discussed in Section 3.1, these parameters represent the estimate of the power consumption of the individual appliances at different operating regimes. For the fridge and the heat pump, second-order dynamical models are also estimated. Results regarding the quality of the estimated dynamical models have been already presented in Fig. 2.

5.2. Performance metrics

Disaggregation is performed on a dataset $\mathcal{D}_T$ of length T (disjoint from the training sets $\mathcal{M}_i$, $i = 1, \ldots, N$) which consists only of the aggregate power readings. The length T of the dataset $\mathcal{D}_T$ is equal to 129600, which corresponds to 90 consecutive days of observations (from January 1, 2013 to March 31, 2013). The available end-use profiles are employed only as ground-truth data to assess the quality of the disaggregated consumption patterns, which is measured with respect to the following metrics:

1. The F-score Fs (Batra et al., 2014)

$$Fs_i = 2\frac{PC_i \times RC_i}{PC_i + RC_i}, \tag{26}$$

where the indexes RC_i and PC_i in (26) are the so-called *recall* and *precision*, and they are defined as

$$RC_i = \frac{TP_i}{TP_i + FN_i}, \quad PC_i = \frac{TP_i}{TP_i + FP_i},$$

where TP_i, FP_i and FN_i are respectively: the number of events correctly classified when the appliance is on (true positive); the number of events classified as on when the appliance is actually off (false positive); the number of events classified as off when the appliance is actually on (false negative). The score Fs_i measures the capability of the disaggregation method to correctly classify the on/off state of the ith appliance. Since multiple-state appliances are used, 10 W is selected as a threshold to claim whether the appliance is on or off. The threshold is set to 50 W for the heat pump, as its estimated low-level power consumption is 33.7 W (see Section 5.1);

2. The *Estimated Energy Fraction Index* (EEFI)

$$\text{EEFI}_i = \frac{\sum_{t=1}^{T} \hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i^*(t))}{\sum_{i=1}^{N} \sum_{t=1}^{T} \hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i^*(t))}, \tag{27}$$

which represents the fraction of energy assigned to the ith appliance. This index is compared with the *Actual Energy Fraction Index* (AEFI)

$$\text{AEFI}_i = \frac{\sum_{t=1}^{T} y_i(t)}{\sum_{i=1}^{N} \sum_{t=1}^{T} y_i(t)}, \tag{28}$$

which indicates the actual fraction of energy consumed by the ith appliance. A similar value between EEFI_i and AEFI_i indicates that the contribution of the ith device on the total power consumption is correctly estimated.

3. The *Relative Square Error* (RSE)

$$\text{RSE}_i = \frac{\sum_{t=1}^{T} (y_i(t) - \hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i^*(t)))^2}{\sum_{t=1}^{T} (y_i(t))^2}. \tag{29}$$

The RSE_i index provides a normalized measure of the mismatch between the actual and the reconstructed consumption pattern for the ith appliance.

4. The R^2 coefficient;

$$R_i^2 = 1 - \frac{\sum_{t=1}^{T} (y_i(t) - \hat{y}_i(t, \boldsymbol{\Theta}_\mathbf{i}, s_i^*(t)))^2}{\sum_{t=1}^{T} (y_i(t) - \bar{y}_i)^2} \tag{30}$$

with $\bar{y}_i = \frac{1}{T} \sum_{t=1}^{T} y_i(t)$. As the RSE_i index, also R_i^2 measures the match over time between estimated and actual end-use power profiles.

The above metrics provide increasing levels of information on the end-use power consumptions. Indeed, the F-score only gives an indication on the capabilities of the disaggregation approach in detecting whether a device is on or off, while the EEFI_i index provides information on the power consumed by each appliance. This is more informative than the F-score to design customized feedbacks and demand management strategies. Finally, the RSE_i and R_i^2 indexes measure the quality of the reconstructed single-appliance power consumption trajectories over time, which is crucial to retrieve information about consumptions during peak hours.

Table 1
Achieved F-scores Fs. Results of the dynamic programming (DP)-based algorithm, the multi-model Kalman-filtering (KF)-based algorithm and their versions with reduced computational complexity.

	DP algorithm		KF algorithm	
	Complete	Reduced	Complete	Reduced
Clothes dryer	99.6%	99.6%	90.0%	90.1%
Dishwasher	99.0%	99.0%	89.4%	89.5%
Fridge	98.3%	98.4%	75.9%	76.0%
Heat Pump	99.9%	99.9%	85.3%	85.3%
Basement	94.6%	94.9%	94.6%	94.5%

Table 2
Relative Square Errors RSE and R^2 coefficients. Results of the dynamic programming (DP)-based algorithm and its version with reduced computational complexity.

	DP algorithm		DP with reduced complexity	
	RSE_i	R_i^2	RSE_i	R_i^2
Clothes dryer	14.4%	85.4%	13.4%	86.4%
Dishwasher	43.9%	55.0%	43.4%	55.5%
Fridge	20.4%	70.5%	21.6%	68.8%
Heat Pump	1.8%	97.9%	1.7%	98.0%
Basement	11.6%	85.6%	8.6%	89.3%

Table 3
Relative Square Errors RSE and R^2 coefficients. Results of the multi-model Kalman-filtering (KF)-based algorithm and its version with reduced computational complexity.

	KF algorithm		KF with reduced complexity	
	RSE_i	R_i^2	RSE_i	R_i^2
Clothes dryer	10.5%	89.3%	23.6%	76.0%
Dishwasher	31.2%	68.0%	46.5%	52.4%
Fridge	22.9%	67.0%	24.2%	65.0%
Heat Pump	1.4%	98.3%	7.2%	91.5%
Basement	11.6%	85.7%	10.8%	86.6%

5.3. Numerical results

The aggregate power readings $y(t)$ forming the validation dataset $\mathcal{D}_T$ is constructed by summing up the power consumptions of the five appliances specified in Section 5.1.

Estimated end-use profiles

To compare the performance of the proposed approaches and assess their robustness with respect to modelling errors, the aggregate readings $y(t)$ of dataset $\mathcal{D}_T$ are corrupted by a fictitious zero-mean Gaussian noise with standard deviation 4 W. Furthermore, the unmodelled consumption patterns of bedroom, garage and dining room are added on top of $y(t)$.

The obtained values of the performance metrics are provided in Tables 1–3, and in Figs. 8 and 9, while the disaggregated signals are plotted in Figs. 10–19. For the sake of visualization, only a portion of the disaggregated profiles is reported in the figures. The obtained results show that the dynamic-programming approach is more accurate in detecting on/off states of the appliances (see Table 1) and that both the dynamic-programming and the Kalman-filter-based approach accurately estimate the fraction of energy consumed by each appliance (see Figs. 8–9, where the EEFI and the AEFI indexes are compared). This good performance is mainly due to an accurate estimate of the

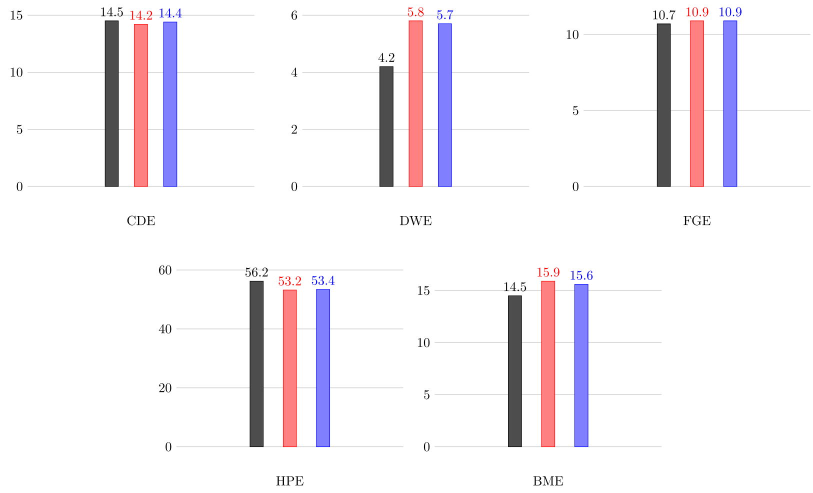


Fig. 8. Dynamic-programming-based approach. Actual Energy Fraction Index [%] (black); Estimated Energy Fraction Index (EEFI) [%] by complete version (red) and simplified version (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

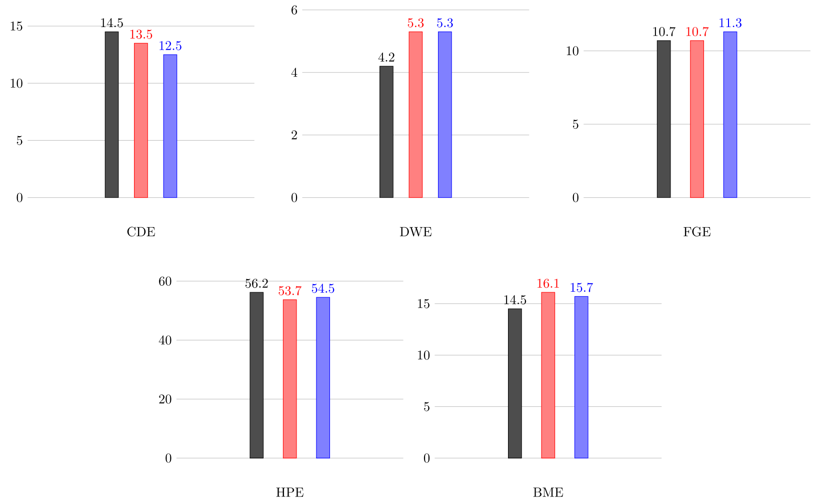
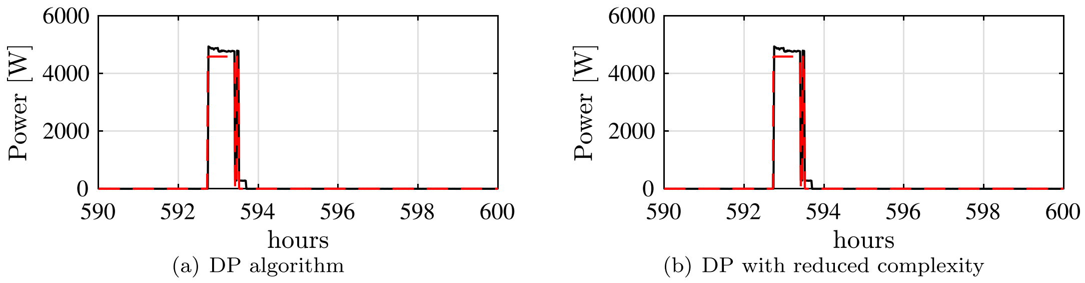


Fig. 9. Multi-model Kalman-filtering-based approach. Actual Energy Fraction Index [%] (black); Estimated Energy Fraction Index (EEFI) [%] by complete version (red) and simplified version (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

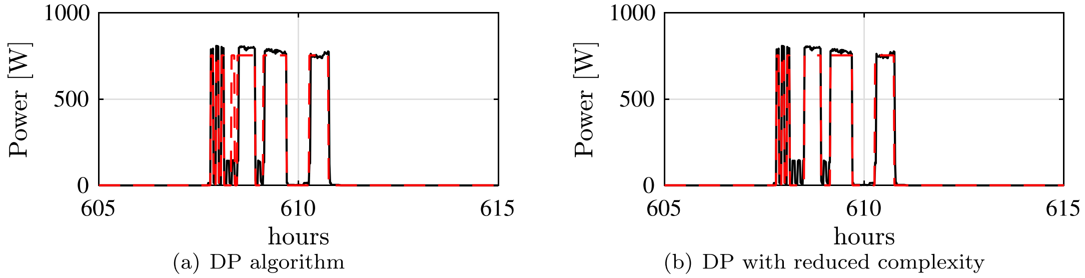


(a) DP algorithm

(b) DP with reduced complexity

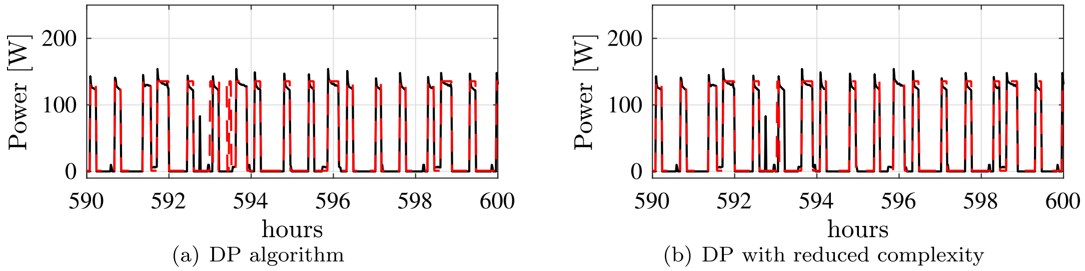
Fig. 10. Cloths dryer: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

disaggregated trajectories over time (as shown in Figs. 10–19, and quantified in terms of the RSE and R^2 indexes in Tables 2–3). It is interesting to note in Tables 1–3 that the reconstructed consumption patterns retrieved with the Kalman-filter-based approach are more accurate than the ones obtained with the DP-based method, due to the use of dynamic models to describe the appliances' behaviour. Furthermore, by comparing Tables 1 and 2, it can be noticed that the simplified version of the DP-based approach might even outperform the complete version of the algorithm. Indeed , because of conditions (C_1)–(C_4) (see Section 4.1), the simplified approach a-priori discards some configurations which are unlikely to happen in practice. On the other hand, the results obtained with the simplified version of the KF-based approach are always less accurate than the ones achieved with the complete version of the approach, but the use of the simplified KF-based algorithm guarantees a substantial reduction of the computational time required for disaggregation, as shown at the end of the section.



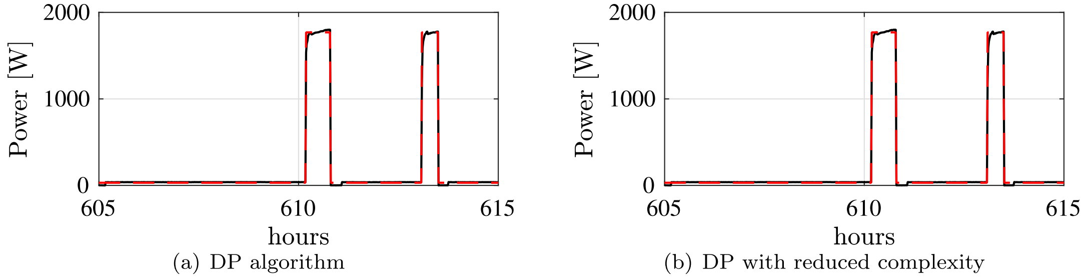
(a) DP algorithm (b) DP with reduced complexity

Fig. 11. Dishwasher: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



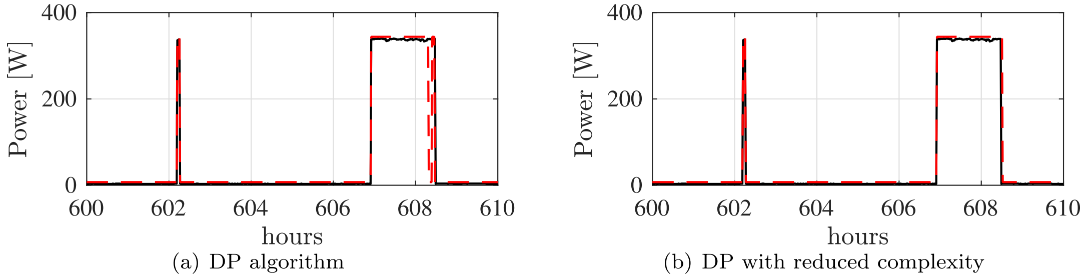
(a) DP algorithm (b) DP with reduced complexity

Fig. 12. Fridge: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



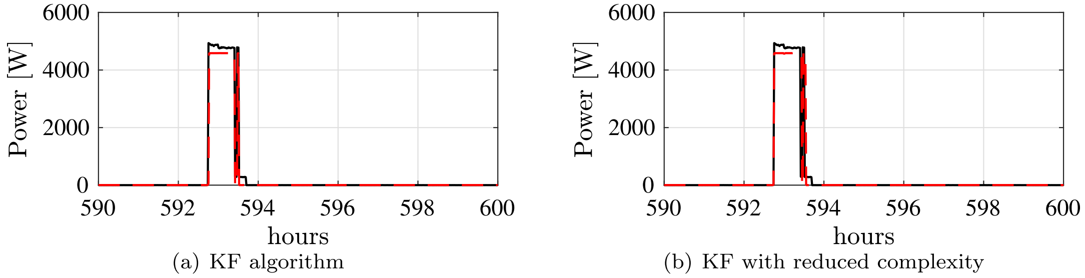
(a) DP algorithm (b) DP with reduced complexity

Fig. 13. Heat Pump: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



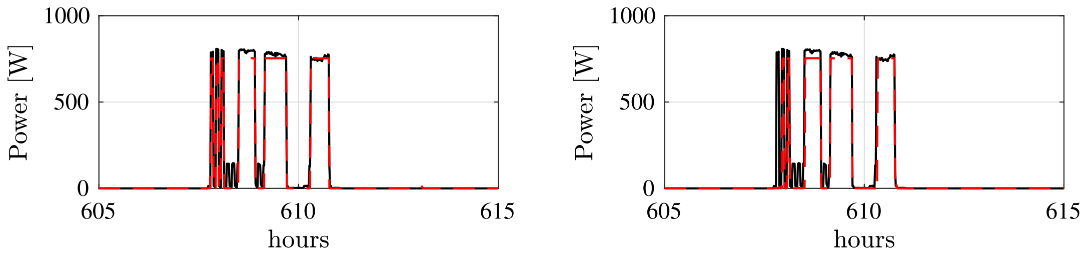
(a) DP algorithm (b) DP with reduced complexity

Fig. 14. Basement plugs & lights: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



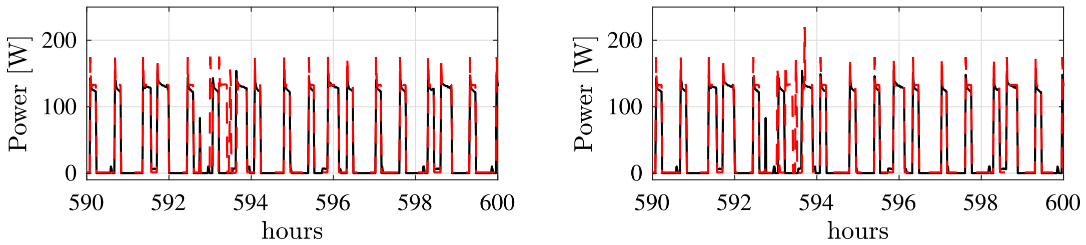
(a) KF algorithm (b) KF with reduced complexity

Fig. 15. Cloths dryer: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



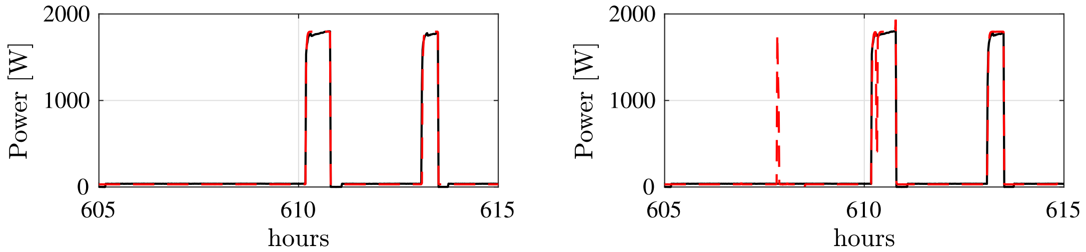
(a) KF algorithm (b) KF with reduced complexity

Fig. 16. Dishwasher: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



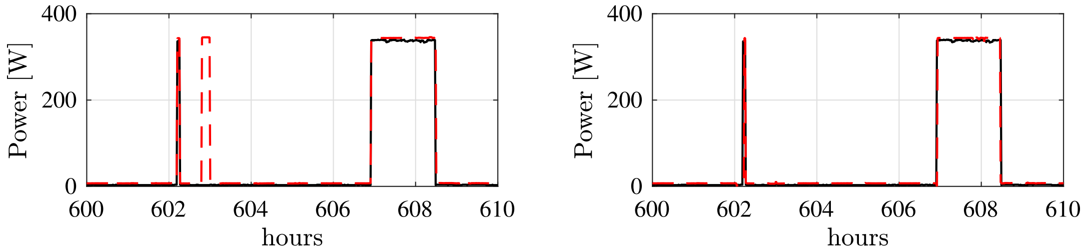
(a) KF algorithm (b) KF with reduced complexity

Fig. 17. Fridge: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) KF algorithm (b) KF with reduced complexity

Fig. 18. Heat Pump: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) KF algorithm (b) KF with reduced complexity

Fig. 19. Basement plugs & lights: True (black) vs estimated (red) power demands. Black and red lines are almost overlapped. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Effect of seasonality on the disaggregated profiles

To assess how changes in the consumer behaviour due to seasonality affect the performance of the proposed methods, disaggregation is additionally performed over a new dataset $\widetilde{\mathcal{D}}_T$, which comprises readings collected over 3 months (from June 1, 2012 to August 31, 2012). The obtained results are then compared with the ones obtained disaggregating $\mathcal{D}_T$ (constituted by measurements collected from January 1, 2013 to March 31, 2013). For a fair comparison, the aggregate readings build from $\mathcal{D}_T$ are not corrupted by fictitious noise. The F-scores and the R^2 indexes obtained for the different datasets are reported in Tables 4–5 and in Tables 6–7, respectively. As shown by the F-scores in Tables 4–5, changes due to seasonality do not significantly influence the capability of the methods in reconstructing the *on/off* states of the different devices. On the other hand, seasonality influences the accuracy of the reconstructed consumption patterns. Nonetheless, the achieved R^2 indexes reported in Tables 6–7 for $\mathcal{D}_T$ and $\tilde{\mathcal{D}}_T$ are generally comparable, with differences depending mainly on the accuracy of the individual appliances models in describing their consumption in each season.

Comparison with state-of-the-art disaggregation methods

The performance of the proposed approaches are then compared with two state-of-the-art disaggregation algorithms implemented in

Table 4
Achieved F-scores F_s. Results of the dynamic programming (DP)-based algorithm and its version with reduced computational complexity obtained disaggregating over $\mathcal{D}_T$ (Winter) and $\widetilde{\mathcal{D}}_T$ (Summer).

	DP algorithm		DP reduced complexity	
	Summer	Winter	Summer	Winter
Clothes dryer	99.7%	99.6%	99.7%	99.6%
Dishwasher	99.1%	99.1%	99.4%	99.0%
Fridge	98.1%	98.4%	98.4%	98.4%
Heat Pump	99.7%	99.9%	99.9%	99.9%
Basement	97.7%	94.6%	98.1%	94.9%

Table 5
Achieved F-scores F_s. Results of the multi-model Kalman Filter (KF)-based algorithm and its version with reduced computational complexity obtained disaggregating over $\mathcal{D}_T$ (Winter) and $\widetilde{\mathcal{D}}_T$ (Summer).

	KF algorithm		KF reduced complexity	
	Summer	Winter	Summer	Winter
Clothes dryer	93.2%	90.0%	93.4%	90.1%
Dishwasher	92.8%	89.4%	92.9%	89.5%
Fridge	73.1%	75.9%	73.0%	76.0%
Heat Pump	92.2%	85.4%	92.2%	86.0%
Basement	97.8%	94.6%	97.7%	94.5%

Table 6
R^2 coefficient. Results of the dynamic programming (DP)-based algorithm and its version with reduced computational complexity obtained disaggregating over $\mathcal{D}_T$ (Winter) and $\widetilde{\mathcal{D}}_T$ (Summer).

	DP algorithm		DP reduced complexity	
	Summer	Winter	Summer	Winter
Clothes dryer	83.4%	85.1%	84.5%	86.4%
Dishwasher	32.4%	57.0%	76.6%	55.4%
Fridge	70.6%	71.2%	70.2%	68.8%
Heat Pump	82.4%	97.9%	95.8%	98.0%
Basement	73.0%	85.9%	81.9%	89.3%

Table 7
R^2 coefficient. Results of the multi-model Kalman Filter (KF)-based algorithm and its version with reduced computational complexity obtained disaggregating over $\mathcal{D}_T$ (Winter) and $\widetilde{\mathcal{D}}_T$ (Summer).

	KF algorithm		KF reduced complexity	
	Summer	Winter	Summer	Winter
Clothes dryer	87.5%	89.1%	72.3%	76.3%
Dishwasher	68.6%	69.1%	62.2%	55.9%
Fridge	66.3%	67.1%	66.3%	65.5%
Heat Pump	94.6%	98.3%	88.8%	91.8%
Basement	78.8%	86.2%	77.7%	86.6%

Table 8
Achieved F-score F_s. Comparison between the results of the dynamic programming (DP)-based algorithm, the multi-model Kalman filter (KF)-based method and the 2-states Factorial Hidden Markov Model (FHMM) and Combinatorial Optimization (CO), implemented in (Batra et al., 2014).

	DP	KF	CO	FHMM
Clothes dryer	99.6%	90.1%	92.6%	95.7%
Dishwasher	99.0%	89.5%	93.3%	88.5%
Fridge	98.4%	76.0%	94.9%	92.6%
Heat Pump	99.9%	86.0%	99.5%	99.9%
Basement	94.9%	94.5%	85.7%	90.7%

Table 9
R^2 coefficient. Comparison between the results of the dynamic programming (DP)-based algorithm, the multi-model Kalman filter (KF)-based method and the 2-states Factorial Hidden Markov Model (FHMM) and Combinatorial Optimization (CO), implemented in (Batra et al., 2014).

	DP	KF	CO	FHMM
Clothes dryer	84.4%	76.3%	96.8%	98.5%
Dishwasher	55.4%	55.9%	8.3%	47.5%
Fridge	68.8%	65.5%	0%	24.0%
Heat Pump	98.0%	91.8%	92.1%	97.1%
Basement	89.0%	86.6%	0%	45.3%

Table 10
Average CPU time, in milliseconds, required to disaggregate the total power consumption at a given time instant. Results of the dynamic programming (DP)-based algorithm, the multi-model Kalman-filtering (KF)-based algorithm and their versions with reduced computational complexity.

	DP algorithm		KF algorithm	
	Complete	Reduced	Complete	Reduced
CPU time [ms]	0.006	0.006	0.451	0.166

the NILMTK toolkit (Batra et al., 2014) Specifically, the proposed approaches are compared with (i) the method for exact inference over Factorial Hidden Markov Models (FHMMs) with Gaussian probability of emission and (ii) combinatorial optimization. The FHMM and the signatures of each appliance are learned from the same datasets $\{\mathcal{M}_i\}_{i=1}^{5}$ used to train the jump models employed in the proposed approaches. Disaggregation is then performed on $\mathcal{D}_T$, without corrupting the aggregate readings. Table 8 shows the achieved F-scores, by considering only the simplified versions of the algorithms proposed in the paper. The reported F-scores show that the proposed algorithms perform comparably to state-of-the-art disaggregation methods in term of *on/off* state recognition. The lower values of F_s obtained for the fridge and the heat pump with the Kalman filter based approach can be attributed to the use of dynamic models to describe the consumption behaviour of these appliances. Nonetheless, both the DP-based and the KF-based approaches perform generally better than the considered state-of-the-art methods in reconstructing the appliances' consumption patterns. In particular, by looking at the R^2 indexes reported in Table 9, it can be noticed that both the proposed methods significantly outperform combinatorial optimization and FHMM in reconstructing the patterns for the fridge and the basement plugs and lights.

Computational complexity

The average CPU times required to compute the disaggregated signals at each time instant are reported in Table 10. Both the complete and the reduced-complexity DP-based approaches take 6 μs to perform a disaggregation step. Thus, in the considered application, the simplified method does not lead to improvements in terms of CPU time, because of the overhead time required to check conditions (C_1)–(C_4). On the other hand, the simplified KF-based approach is about 3× faster than the complete version. These results show the potentiality of the proposed algorithms for big-data processing, thanks also to their iterative nature. Compared with the approach in Piga et al. (2016), which in average takes around 3 minutes to disaggregate one-day consumption records, the proposed approaches take less than 8 milliseconds to disaggregate the same data record. Furthermore, the complete versions of the KF-based approach and the DP-based method require around 59 seconds and 0.7 s to disaggregate three months of data, respectively, with the CPU time required to disaggregate $\mathcal{D}_T$ dropping to 22 seconds when the KF-based approach with reduced complexity is used. Instead, the 2-state FHMM and combinatorial optimization (Batra et al., 2014) disaggregate three months of data in 20 seconds and 3 seconds on average, respectively, thus making the presented approaches competitive with state-of-the-art methods in terms of computational complexity. Nonetheless, it has to be remarked that, because of the recursive nature of the proposed algorithms, they are suited for real-time implementation. Furthermore, the complexity of their simplified versions does not increase exponentially with the number of appliances and the number of operating modes.

6. Conclusions

In this paper two iterative algorithms for real-time energy disaggregation have been presented, along with heuristics to reduce their computational complexity. Both the proposed algorithms use jump linear models, prior estimated from the devices' signatures, to describe the behaviour of the individual appliances.

Differently from many existing disaggregation approaches, the proposed methods can handle multiple-state devices, thus enabling the reconstruction of the appliances' consumption patterns rather than simply detecting on/off states. The proposed methods further allow the estimation of appliance-level consumptions from readings with a time resolution between 1 second and 1 minute, without the need to analyse the sub-harmonics of the 50/60-Hz electric signal. Therefore, the proposed approaches do not require high-frequency sampling devices. If 1 min data are not available, the proposed methods are expected to return satisfactory results, provided that the models of the individual appliances are identifiable (*i.e.,* the different operating modes can be distinguished from the consumption patterns) and that the devices are characterized by different consumption profiles, so that the contribution of the individual appliances is detectable from the aggregate data. It is worth remarking that this limitation is intrinsic to all existing disaggregation approaches. Another strength of the proposed methods is their computational efficiency that, combined with the iterative nature, makes them suited for big data processing. The main advantage of the proposed methods over existing ones is their recursive nature, which allows to process aggregate readings in real-time instead of requiring a batch of data to disaggregate power.

At the moment the approaches are not tailored to handle changes in the appliances behaviour at the different operating conditions, as the jump models describing the individual devices are not updated when disaggregation is performed. This limits the performance of the proposed approaches for long-term usage, where it is expected that the model of the appliances change due to multiple factors, *e.g.,* the aging of the device. To overcome this limitation, future research is devoted to the integration of real-time identification into the disaggregation algorithms, so to simultaneously reconstruct the disaggregated signals and recursively update the jump sub-models describing the individual appliances, without the need to recalibrate the models through intrusive experiments.

As a final remark, it is worth further stressing that algorithms for energy disaggregation allow consumers to obtain additional real-time information without replacing available appliances in the house and without installing costly sensing and communication infrastructures. Of course, the algorithms proposed in this paper for energy disaggregation would possibly lose their interest in the future, if residential houses are endowed with smart and interconnected plugs (or appliances). However, at least in the near future, this solution is likely to be still too costly, thus legitimizing research towards energy disaggregation methods.

Declaration of competing interest

None declared.

References

Bar-Shalom, Y., Kirubarajan, T., & Li, X.-R. (2002). *Estimation with applications to tracking and navigation*. New York, NY, USA: John Wiley & Sons, Inc.

Batra, N., Kelly, J., Parson, O., Dutta, H., Knottenbelt, W., Rogers, A., Singh, A., & Srivastava, M. (2014). Nilmtk: an open source toolkit for non-intrusive load monitoring. In *Proceedings of the 5th international conference on Future energy systems* (pp. 265–276). ACM.

Bemporad, A., Breschi, V., Piga, D., & Boyd, S. (2018). Fitting jump models. *Automatica*, *96*, 11–21.

Bonfigli, R., Principi, E., Fagiani, M., Severini, M., Squartini, S., & Piazza, F. (2017). Non-intrusive load monitoring by using active and reactive power in additive factorial hidden markov models. *Applied Energy*, *208*, 1590–1607.

Chang, H. (2012). Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses. *Energies*, *5*(11), 4569–4589.

Cominola, A., Giuliani, M., Piga, D., Castelletti, A., & Rizzoli, A. (2017). A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring. *Applied Energy*, *185*(P1), 331–344.

Esa, N., Abdullah, P., & Hassan, M. (2016). A review disaggregation method in non-intrusive appliance load monitoring. *Renewable & Sustainable Energy Reviews*, *66*, 163–173.

Giri, S., & Berges, M. (2015). An energy estimation framework for event-based methods in non-intrusive load monitoring. *Energy Conversion and Management*, *90*, 488–498.

Hart, G. W. (1992). Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, *80*(12), 1870–1891.

Kolter, J. Z., & Jaakkola, T. (2012). Approximate inference in additive factorial hmms with application to energy disaggregation. In *Proceedings of the fifteenth international conference on artificial intelligence and statistics 22* (pp. 1472–1482).

Liang, J., Ng, S., Kendall, G., & Cheng, J. (2010). Load signature study - part i: basic concept, structure, and methodology. *IEEE Transactions on Power Delivery*, *25*(2), 551–560.

Ljung, L. (1999). *System Identification: Theory for the User*. Englewood Cliffs, NJ.: Prentice-Hall.

Makonin, S., Popowich, F., Bartram, L., Gill, B., & Bajic, I. V. (2013). AMPds: a public dataset for load disaggregation and eco-feedback research. In *IEEE electrical power & energy conference (EPEC)* (pp. 1–6).

Mejari, M., Naik, V., Piga, D., & Bemporad, A. (2018). Energy disaggregation using piecewise affine regression and binary quadratic programming. In *2018 IEEE Conference on Decision and Control (CDC)*, Miami Beach, FL, 2018 (pp. 3116–3121).

Piga, D., Cominola, A., Giuliani, M., Castelletti, A., & Rizzoli, A. E. (2016). Sparse optimization for automated energy end use disaggregation. *IEEE Transactions on Control Systems Technology*, *24*(3), 1044–1051.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, *1*(1), 33–57.

Srinivasan, D., Ng, W., & Liew, A. (2006). Neural-network-based signature recognition for harmonic source identification. *IEEE Transactions on Power Delivery*, *21*(1), 398–405.

Suzuki, K., Inagaki, S., Suzuki, T., Nakamura, H., & Ito, K. (2008). Nonintrusive appliance load monitoring based on integer programming. In *2008 SICE annual conference* (pp. 2742–2747)..

Yang, C., Soh, C., & Yap, V. (2015). A systematic approach to on-off event detection and clustering analysis of non-intrusive appliance load monitoring. *Frontiers in Energy*, *9*(2), 231–237.

Zeifman, M., & Roth, K. (2011). Nonintrusive appliance load monitoring: review and outlook. *IEEE Transactions on Consumer Electronics*, *57*(1), 76–84.

Zoha, A., Gluhak, A., Imran, M. A., & Rajasegarar, S. (2012). Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, *12*(12), 16838–16866.