Abstract—The need for energy disaggregation increases with the need for a more detailed understanding and more accurate estimates about the energy usage. One of the main approaches for energy disaggregation is Non-Intrusive Load Monitoring (NILM). NILM refers to the analysis of the aggregate power consumption of electric loads in order to recognize the existence and the consumption profile of each individual appliance. While there exist non-event-based NILM systems, many NILM systems follow the event-based approach in the sense that they rely mainly on the detection and classification of events in the aggregate electrical signal. In this paper, we describe our work on developing an event detector suitable for unsupervised NILM systems. The proposed event detector is capable of accurately defining the times limits of each transition interval in the power signal. This feature is very important specially for NILM systems that depend on transient features. The detector is tested on the publicly available BLUED dataset and shows event detection results more than 98%. Test results of a complete unsupervised NILM system using the proposed detector are also provided and show possible disaggregation up to 92% of the energy. Moreover, the system has been utilized in an energy-disaggregation competition held by Belkin and achieved a score within the top ten results with disaggregation value of 93.41% of the total time.

Index Terms—Inverse Load Reconstruction, Unsupervised Non-Intrusive Load Monitoring (NILM), Model Order Estimation, Mean-Shift Clustering, BLUED Dataset

I. INTRODUCTION

Energy disaggregation becomes more and more important not only to residential consumers but also to power companies as well as appliance manufacturers. Many residential consumers lack a good understanding of their usage of energy or even the consumption of individual appliances [1][3]. Power companies require accurate estimates about future energy usage in order to handle more efficient energy generation strategies such as load-dependent energy generation, smartgrids, dynamic pricing models, or even to find more efficient energy conservation approaches. Appliance manufacturers can benefit from a detailed usage pattern of their appliances in order to provide more energy-efficient appliances or new power applications such as home automation, activity sensing, and health care.

Electrical loads can be monitored either in a distributed approach where each appliance has its own sensor or by disaggregating the building-level energy consumption profile. The latter approach is also referred to as Non-Intrusive appliance Load Monitoring (NILM) [7], or inverse load reconstruction [8]. NILM systems disaggregate the electrical signal measured from a single metering point, thus, providing more reliability as a result of the reduced metering points and less cost due to the reduction in the utilized hardware. A good review of existing NILM systems can be found in [6][9][10].

NILM systems are categorized into event-based and non-event-based approaches [11]. Event-based NILM systems rely mainly on the detection and classification of events within the aggregate electrical signal. Furthermore, in [12] NILM systems are categorized into supervised and non-supervised approaches depending on whether or not they require a training process prior to deployment on a target building. The training process required by the supervised NILM systems imposes a complex installation process or constraints on adding new appliances to the monitored circuit. In contrast, unsupervised NILM systems are expected to have a wider applicability and even less intrusion.

In this paper, we describe our work on the development of an event detector suitable for energy disaggregation. The detector is developed among the work on a completely unsupervised NILM system. Therefore, we provide test results of the event detector as well as complete disaggregation results from the NILM system. The system is tested on the publicly available Building-Level Fully labeled Electricity Disaggregation (BLUED) dataset [13]. Moreover, we describe our participation in the energy disaggregation competition held by Belkin [14] on Kaggle’s platform. A good review about event detection techniques for NILM systems is found in [11].

This paper is organized as follows. Section II provides a detailed description of the proposed event detector. Section III briefly describes the unsupervised NILM system within which the proposed event detector is tested. Section IV provides results of testing the event detector on the BLUED dataset together with disaggregation results of both BLUED and Belkin’s power datasets. Finally, Section V concludes this paper.

II. EVENT DETECTION

In the event detection stage, the electrical signal is segmented into transient and steady-state sections. In contrast to the conventional change-point detection, the proposed event detector is capable of accurately defining the time limits of each transition interval. Accurate detection of the change interval is crucial for extracting appliances’ signatures from their transient behavior. Moreover, the proposed detector does
not utilize any filtering technique in order not to distort the transient sections during the detection.

The event detector also features dynamic adaptation to noise without the need for a pre-training process. Noise in an electrical signal comes from both the power source and the operating appliances. Therefore, noise pattern can vary widely during the monitoring period depending on the number and types of connected appliances.

In order to enhance performance, the detection process is applied on the logarithmically transformed signals $P_l(t)$ and $Q_l(t)$ of the raw real $P(t)$ and reactive $Q(t)$ power signals based on the function

$$X_l = \begin{cases} \ln(X) & X > 0 \\ 0 & X = 0 \\ -\ln(-X) & X < 0 \end{cases}$$

where $X \in \{P, Q\}$. With this transform, the event detection is performed on a narrower power range resulting in a higher performance.

In the following, the event detector is described using a two-dimensional signal of the real $P$ and reactive $Q$ powers. However, the detection algorithm is applicable to any dimension and it has also been tested on a one-dimensional signal of the real power $P$ only.

The event detector consists of three main steps, namely the coarse event detection, the refining step, and the validation. However, a model order estimation algorithm is introduced first as it is the core of the coarse event detection step.

### A. Model order estimation

Given an interval $[t_i, t_{i+1}]$, the transformed real $P_l(t)$ and reactive $Q_l(t)$ power signals are projected on the $P_lQ_l$-Plane. In the resulting $P_lQ_l$-Plane, steady-states $\Pi$ are represented as clusters while transients $\Psi$ as well as noise are found as scattered points or outliers. The role of this step is to estimate the number of clusters. Worth noting is that we do not assume any constraints on the distribution of clusters.

First, the $P_lQ_l$-Plane is converted to a two dimensional histogram by dividing it into rectangular bins with the bin size of $(b_P, b_Q)$. Each instance $(P_l(\tau), Q_l(\tau))$ of the logarithmically transformed signals, where $\tau \in [t_i, t_{i+1}]$, increments the value of the bin $b_{ij}$ where

$$i = \left\lfloor \frac{P_l(\tau)}{b_P} \right\rfloor \quad \text{and} \quad j = \left\lfloor \frac{Q_l(\tau)}{b_Q} \right\rfloor$$

**Figure 1** shows an example of the two-dimensional histogram built by dividing the $P_lQ_l$-Plane into bins. Each set of non-zero, neighboring bins based on the eight-neighborhood rule (i.e., bins that share at least one vertex) are considered one object. The object’s occurrence value $\kappa$ is the sum of all bin values that belong to this object. We impose a constraint that a steady-state must be active for a minimum of $N_{th}^\Pi$ samples while the transient duration is not limited. The minimum steady-state length is initially set to

$$N_{th}^\Pi = \begin{cases} \lfloor 1.5 \cdot f_s \rfloor & f_s \geq 2 \\ \frac{3}{f_s} & f_s < 2 \end{cases}$$

where $f_s$ is the sampling frequency in Hertz.

Objects with occurrence value $\kappa \geq N_{th}^\Pi$ are considered clusters while the remaining are noise or belong to the transient section. The estimated model order $M$ is the number of clusters in the $P_lQ_l$-Histogram.

The initial value of the bin size is set based on the required minimum detectable change. The bin size is then dynamically adjusted during the detection in order to account for wider clusters as explained in the coarse event detection step. Worth noting is that we do not assume any constraints on the distribution of clusters.

### B. Coarse event detection

The term coarse event is used here to refer to a segment of the signal that contains exactly one transient section $\Psi_i$ together with at least $N_{th}^\Pi$ samples from each of its surrounding steady-states $\Pi_i$ and $\Pi_{i+1}$.

The algorithm starts with an initial window with a size equal to the minimum detectable steady-state length $N_{th}^\Pi$. The window is then widened iteratively with the application of the model order estimation in each step until the detection of two clusters. In each iteration the window is widened by a detection step $D_s$ samples from its right end. The detection step $D_s$ merely represents a trade-off between accuracy and performance and is defined as

$$D_s \in \left\{ x \in \mathbb{N} \mid 1 \leq x \leq N_{th}^\Pi \right\}$$

The smaller the detection step $D_s$ is, the slower the detection becomes but with higher accuracy.

Once two clusters are detected, the window is narrowed iteratively by $D_s$ from its left end while applying the model order estimation until one cluster is detected. The window is
then widened once by $D_s$ samples from its left end. The final window represents the current coarse event.

The algorithm of the coarse event detection can be described as follows:

**Step 1:** initialize window $[t_i, t_{i+1}]$, where $t_{i+1} = t_i + \frac{N^\Pi}{f_s}$.

**Step 2:** receive $n = D_s$ data samples, $t'_{i+1} = t'_{i} + \frac{n}{f_s}$.

**Step 3:** estimate the model order $M$ on $[t_i, t_{i+1}]$.

**Step 4:** if $M < 2$, return to step 2.

**Step 5:** mark last right end point $t_{i+1}$.

**Step 6:** delete $n = D_s$ samples from the left end as $t'_{i} = t'_i + \frac{n}{f_s}$.

**Step 7:** estimate the model order $M$.

**Step 8:** if $M > 1$, return to step 6.

**Step 9:** mark last left end point as $t'_i = t'_i - D_s$.

The final window $[t_i, t_{i+1}]$ represents the current coarse event. Figure 2 shows the coarse events of a sample signal from the BLUED dataset with the sampling frequency of $60$ Hz. The output is highlighted in the shaded windows which define the time limits of each coarse event. Each window contains only one transient $\Psi_i$ and at least $N_{Q_i}^\Pi$ samples from each of the surrounding steady-states $\Pi_i$ and $\Pi_{i+1}$. Therefore, if any of the coarse events is individually plotted on the $P(q)$-Plane, a model order estimation should always result in a model order of $M = 2$. Each coarse event is then fed to the refining step.

**C. Fine event detection**

In the fine event detection, each detected coarse event is fed to a more refining detection algorithm one at a time in order to accurately define the time limits of each transient section. In this step, the unsupervised Expectation Maximization (EM) clustering algorithm is applied on each coarse event. Each coarse event has an estimated number of modes $\hat{M} = 2$ where it contains exactly two steady-states together with noise. The value of $\hat{M}$ is then incremented by one to account for noise and samples that belong to the transient interval. The utilized implementation of the EM algorithm is the one found in the MATLAB function `gmdistribution.fit` [15]. The output from the clustering algorithm is a vector of cluster indexes for all samples within the coarse transition where each sample belongs to either the right steady-state, the left steady-state, or the transient interval.

Figure 3 shows the same power signal as in Figure 2 after applying the refining step on each event separately. As a result, the signal is accurately segmented into steady and transient sections while preserving all characteristic spikes for accurate feature extraction.

**D. Verification**

In this step, not only detected events are verified according to certain constraints but also feedback is provided to the previous detection steps for parameter adjustment. Such parameter adjustment leads to a more adaptive event detector that does not require manual calibration on each deployment. In the following, we describe two of the constraints used to verify each event, namely the off-on simultaneous events and noisy steady-states.

Given two consecutive events $\Psi_j$ with the time limits $[t_i, t_{i+1}]$ and $\Psi_{j+1}$ with the time limits $[t_{i+2}, t_{i+3}]$, the two events are considered simultaneous if

$$t_{i+2} - t_{i+1} < \frac{N_{Q_j}^\Pi}{f_s}$$

(5)

If the first event $\Psi_j$ represents a decrease in the real power $P$ while the second is an increase in the real power, we refer to them as off-on simultaneous events. Since the steady-state between the two events $\Psi_{j+1}$ is less than $N_{Q_j}^\Pi$ samples long, the two events together with the steady-state in between are detected as one event with the time limits $[t_i, t_{i+3}]$. Each detection is checked for the off-on simultaneous events according to

$$\min \left[ \Psi^P_k(0), \Psi^P_k(N_k^\Psi - 1) \right] - \Psi^P_k \geq \Delta P_{\min}$$

(6)

where $\Psi^P_k$ is the real power signal of the $k^{th}$ transient section, $N_k^\Psi$ is the number of samples in the $k^{th}$ transient section, and

$$\Psi^X_k = \min_n [\Psi^X_k(n)] \quad 0 \leq n < N_k^\Psi$$

(7)
If the condition in Equation 6 is satisfied, the event detection algorithm is triggered on the same event again with a reduced value for the minimum detectable steady-state length \( N_{\Pi} \) until the detection results in two off-on separate events. Separating off-on simultaneous events occurs in the event detection stage as long as the condition in Equation 6 is held. Otherwise, it is achieved in the transition matching stage whose description is beyond the scope of this paper.

In the \( P_i Q_i \)-Histogram, the synthesis of non-zero bins into objects is highly dependent on the bin size. A small bin size can incorrectly divide one object into several sub-objects resulting in a false detection as a result of the increase in the estimated model order. On the other hand, a large bin size may combine two or more objects into one leading to missed events. In order to avoid manual calibration of the bin size, we take the following approach.

Initially, the bin size is set to a suitably low value according to the required minimum detectable change. After each detection, the following condition is checked. Given two steady-states \( \Pi_j \) and \( \Pi_{j+1} \) (which are objects in the \( P_i Q_i \)-Histogram) with time limits \([t_i, t_{i+1}]\) and \([t_{i+2}, t_{i+3}]\) respectively, if

\[
\begin{align*}
t_{i+2} - t_i & \leq 0.5 \cdot \frac{N_{\Pi}}{f_s} \quad (8) \\
t_{i+3} - t_{i+1} & \leq 0.5 \cdot \frac{N_{\Pi}}{f_s} \quad (9)
\end{align*}
\]

then the bin size is increased iteratively until these two objects are combined. The new bin size is held until the detection of a new event. This process ensures that if the signal contains high fluctuations the bin size is increased to accommodate for the noise as long as the causing appliance is operating.

III. SYSTEM VIEW

In this section, we provide a brief description of the unsupervised NILM system in which the event detector has been tested. A detailed description of each stage is, however, beyond the scope of this paper.

![Figure 4](image)

**Figure 4**: A block diagram of the developed unsupervised NILM system. All stages utilize only unsupervised techniques in order to disaggregate the individual appliances from the given real and reactive power signals.

IV. EXPERIMENTS AND RESULTS

In this section we provide test results of the event detector applied on residential power datasets together with disaggregation results of the complete NILM system. The event detector is tested on the publicly available BLUED dataset [13] and the power dataset provided by the consumer electronics manufacturer Belkin in its energy disaggregation competition [14].

Table I shows the event detection results of both phases of the BLUED dataset. The True Positive Percentage (TPP) and the False Positive Percentage (FPP) represent the second detection metric from [11] and are defined as

\[
TPP = \frac{TP}{E}
\]

**Table I: Event detection results**

<table>
<thead>
<tr>
<th></th>
<th>TPP</th>
<th>FPP</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase A</td>
<td>98.5%</td>
<td>0.55%</td>
<td>886</td>
</tr>
<tr>
<td>Phase B</td>
<td>70.5%</td>
<td>8.75%</td>
<td>1579</td>
</tr>
</tbody>
</table>
Fig. 5: The application of the event detector on 6Hz power signal from Belkin energy disaggregation competition on Kaggles platform [14] with $N_{th} = 10$. Highlighted in blue circles are the detected events. Shown on the left is an example of detected off-on event. On the right is an example of noisy steady-states and their effect on the detection.

\[ FPP = \frac{FP}{E} \]  

(11)

where TP is the number of true positives or valid event detections, FP is the number of false positives or the incorrectly detected events, and $E$ is the total number of actual events taken from the ground truth. Phase A of the BLUED dataset consists mainly of on-off and Finite State Machine (FSM) appliances. This explains the higher detection results of Phase A over Phase B which contains several continuously variable appliances.

\[ \text{Figure 5} \] shows a sample signal from Belkin’s dataset. The figure also shows the detected events (highlighted with blue circles) upon application of the event detection algorithm with $f_s = 6$Hz and $N_{th} = 10$ samples. The first half of the signal shows an example of off-on simultaneous events. Off-on simultaneous events are detected even though in some cases the steady-state length is 600 ms (i.e. $N_{th} \sim 4$ samples).

The second half shows an example of varying steady-states where $\Pi_{12}$ and $\Pi_{14}$ follow a sinusoidal behavior while $\Pi_{13}$ has a wide and changing noise pattern. The figure shows the advantage of the dynamic bin size adaptation in handling these steady-states. Worth noting, however, is that high noise values ($\pm 50$ Watts) led to an inaccurate detection as observed in $\Psi_{12}$.

\[ \text{Figure 6a} \] shows disaggregation results of the NILM system on the BLUED dataset. The BLUED dataset has 7-day long measurements. In the figure, we projected all operation intervals of disaggregated appliances into a single 24-hours day. Shown results belong to phase A only and has 14 detected appliances where shaded green areas represent their intervals of operation. A02 represents two lights, bed room lights and bathroom downstairs lights because the system was not able to disaggregate these two load due to the similarity in their signatures. The figure also shows the low-activity during the time period [0, 7] hours as expected. Such low-activity periods are utilized in self-training the NILM system using individually operating appliances. The total disaggregation represents 92% of the total energy.

Appliance A01 is the refrigerator. As observed, its operation does not depend on the time of the day simply because it is a background appliance. \[ \text{Figure 6b} \] shows a single-day disaggregation of the refrigerator. The figure shows the clear periodic behavior of the load which directly indicates that it has an on-off controller. Using this information together with characteristics from the power signals (for example being resistive, capacitive, or inductive) can lead to an identification of the category of appliance. Therefore, behavioral analysis of
disaggregated appliances is among our planned future work in order to develop an unsupervised NILM system with appliance identification.

Finally, we participated in the energy disaggregation competition held by Belkin on the Kaggle platform using the developed NILM system with minor modifications. The disaggregation results were in the 5th position when evaluated on the public folder, and the 6th on the private folder on the last day of the competition 30th of October, 2013. Results showed a successful disaggregation of 93.41% of the total time.

V. CONCLUSION

We have developed a completely unsupervised event-based NILM system that consists of five stages. In this paper, we introduced the event detection stage together with test results of both the detector and the complete NILM system. The detector shows successful disaggregation up to 98% of the total events while the complete NILM system disaggregated 92% of the total energy of the BLUED dataset.

REFERENCES


