**UNIVERSITÀ DI PADOVA** **Dipartimento di Ingegneria dell'Informazione**

TESI DI LAUREA

# SELF-LEARNING ALGORITHM FOR ENERGY DISAGGREGATION

**Laureando**: Giulia De Poli

**Relatori**: Stefano Tomasin , Alessio Filippi

## Corso di Laurea Magistrale in Ingegneria delle Telecomunicazioni

Anno Accademico 2011/2012

# Contents

# List of Figures

# List of Tables

# Abstract

Electric energy consumption is becoming a relevant topic in the last years due to environmental and economic reasons. One of the field of interests in this energy problem is the reduction of the consumption in a house environment. Nowadays people only have information about the total energy consumption of their homes while a detailed report of the appliances individual behaviour would be useful to identify which appliances are effectively consuming more energy. In this way it will be possible to make a decision to reduce the total consumption based on the real consumption of the single appliance. A practical method to measure each appliance consumption is necessary to achieve this purpose.

The non-intrusive appliance load monitor (NALM) approach measures aggregate power energy use as power enters the home.reverses the traditional balance, with simple hardware but complex software to analyse the collected data. It is necessary to install only one measurement unit, which permits very easy installation, removal, and maintenance compared to traditional intrusive load monitoring techniques. Then complex software has to disaggregate the overall measured data in the single appliance.

The aim of the project is the design of a self-learning algorithm that automatically identifies the appliances in a NALM system, determining itself the significant signatures and the associated appliances without any external information. The problem is divided into two parts: event detection and cluster analysis. The first part extracts from the overall data significant signatures that characterise the appliances. The extracted data are called events. The second part analyses the extracted events to find frequent patterns which identify the appliances.

The implemented system is described in detail and validated in the final chapter. The first part presents the demonstrator we use to collect and analyse data and shows some examples which validate the model. The second part estimates the parameter of the model. Finally, in the third part, the model is applied in a real environment.

# Introduction

Electric energy consumption is becoming a relevant topic in the last years due to environmental and economic reasons. The interest in energy efficiency is motivated by the growing concern on climate change and the desire to reduce the energetic costs. One of the field of interests in this energy problem is the reduction of the consumption in a house environment. Nowadays people only have information about the total energy consumption of their homes while a detailed report of the appliances individual behaviour would be useful to identify which appliances are effectively consuming more energy. In this way it will be possible to make a decision to reduce the total consumption based on the real consumption of the single appliance. A practical method to measure each appliance consumption is necessary to achieve this purpose. The traditional approach measures every load in the home separately using complex instrumentation systems that individually meter each device energy consumption. This method involves complex hardware but simple software. It is necessary to install a monitoring point at each appliance and interior wires to connect each one to a central point. A new approach was introduced in 1989 by G. Hart [8]. This method uses a single non-intrusive appliance load monitor (NALM) to measure aggregate power energy use as power enters the home. The NALM approach reverses the traditional balance, with simple hardware but complex software to analyse the collected data. It is necessary to install only one measurement unit, which permits very easy installation, removal, and maintenance compared to traditional intrusive load monitoring techniques. Then complex software has to disaggregate the overall measured data in the single appliance.

The aim of the project is the design of a self-learning algorithm that identifies the appliances in a NALM system. There are two different types of set-up when a NALM system is installed: Manual Setup (MS) and Automatic Setup (AS). The first one requires a one-time intrusive period. During the installation period, when the appliances are switched on and off a signature is observed. The observed signature is manually named as the corresponding appliance. The ”Automatic Setup” determines itself the significant signatures and the associated appliances without any external information. It only uses a priori information about the possible appliance characteristics. The AS-NALM system is preferable for its completely non-intrusive approach. This project proposes a solution to the ”Automatic-Setup” problem. The problem is divided into two parts: event detection and cluster analysis. The first part extracts from the overall data significant signatures that characterise the appliances. The extracted data are called events. The second part analyses the extracted events to find frequent patterns which identify the appliances.

The thesis is organized as follows.

Chapter 1 describes the system model, presenting the original model proposed by Hart. In particular the architecture of the system, the electrical model, the various steps of the NALM algorithm and the possible signatures that characterise the devices are described. Then the problem we want to solve is described in details. The state of the art for the ”Automatic-Setup” and the improvements we want to achieve are presented.

Chapter 2 describes our solution for the problem of the event detector. The chapter is divided into three parts. The first part gives a theoretical definition of the event. Then the tools and the algorithm we use to extract the events are introduced. The third part estimates the parameters of our model by looking at the output in an ideal environment.

Chapter 3 describes our solution to the cluster analysis. The chapter is divided into two parts. The first part describes a known algorithm we use to create the clusters and the modifications we made to make it more suitable to our specific case. The second part introduces a new approach to the cluster algorithm that allows us to recognise more complex types of devices.

Chapter 4 validates the model we presented in the previous chapters. The chapter is organised into three parts. The first part presents the demonstrator we use to collect and analyse data and shows some examples which validate the model. The second part estimates the parameter of the model. Finally, in the third part, the model is applied in a real environment.

# Chapter 1

# System model

This chapter presents the architecture of a NALM system. The chapter is organised into two parts. The first part is an overview of the system model. The general concepts and the state of the art of a NALM system are presented. The second part defines the problem we want to solve, the choices we made and the improvements we want to add to the actual solutions.

## 1.1 System model

A non-intrusive appliance load monitor (NALM) [8] determines the energy consumption of the appliances by analysing both current and voltage measured in a single point. This approach permits a detailed analysis of the energy consumption without installing a measure point at each appliance of interest. The current and voltage are measured in a single point. Complex software for signal processing and analysis extracts from the overall data information on the consumption of the single devices. It is possible to estimate the number and type of the appliances, their energy consumption and other statistics. The NALM system looks for certain ”signatures” which give information about the operating status of a particular device in the load. For example, if we measure a 600 W step increase in the overall power and we know that there is a device, e.g. a hair dryer, consuming the same power, this step indicates that the hair dryer had been switched on. The change in power is an example of a particular electrical signature. The NALM system has to choose certain signatures which describe the appliances in the environment. A signature has to be characteristic of the electrical behaviour of the appliance. It has to be very different for different appliances in order to recognise them. The NALM system is organised as an algorithm in eight steps. During these steps the data are collected and analysed, the final output is a detailed analysis of the appliances in the load and a statistic of their consumptions.

There are two main options in a NALM system:

- **”Manual Setup”**: A MS-NALM requires a one time intrusive period. During this period the appliances are switched on and off and the observed signature is named as the corresponding appliance. This period is necessary to create the database of the appliances. After this period no intrusive access is needed and the system estimates by itself the consumptions and statistics of the single appliances.

- **”Automatic Setup”**: An AS-NALM sets itself the database of the appliances without any external information. It only uses a priori information on the possible type of device. It has to detect significant signatures and relate them to the corresponding appliances.

The MS-NALM system has been the basis to the developing of the AS-NALM system and it will be useful in situations where the AS-NALM fails. The AS-NALM is technically more ambitious because of its total non-intrusiveness.

The possible applications of a NALM system can be residential, commercial or industrial. These classes are considered separately because of the different types of appliance that are present. The implementation and field test have focused more on residential loads because there is more difficult to use intrusive techniques. Some examples of applications are the monitor of single appliances consumption to give a detailed report of the energetic consumptions. Suppose to install in a home environment a NALM system during one month. At the end of the month the system can give a detail report of the consume of the single appliances. The disaggregate energetic bill can be useful to suggest ways to reduce the consumption. The costumer knows which appliances are effectively consuming more energy and can take actions to reduce the total energy consumption. Another example is the monitoring for failure analysis or security purpose. Failures can be detected if there is an unusual power consumption of some appliances. For example we observe that a device , e.g. a freezer, that normally operates with ON-OFF cycles is always in the ON-state at maximum power. The system can give a feedback of the detected failure. An example of the use of a NALM system for security purpose is the application in vacation homes. The system is installed in a home that is unoccupied for long periods. If there is a failure or unexpected consumptions the owner would be notified so it is possible to take actions to solve the problems. Another class of applications where the NALM system can be useful are the situations where it is not possible to get physical access to the individual appliances. In these situations it is not possible to use an intrusive measurement.

In a NALM system a single sensor unit measures the total load to find the signatures that give information about the working devices. The collected data is sent to a central processor which disaggregates the data to characterise each device. The system model description is divided into three parts. The first part presents the electrical model. The second part is an overview of the possible signatures that are generally used to characterise the devices. The third part presents the most important steps of the algorithm of a NALM system.

### 1.1.1 Electrical model

The electricity network, Fig. 1.1, is modelled as a set of appliances connected in parallel to an ideal sinusoidal voltage generator $v_0(t)$. The voltage generator operates at the fundamental frequency $f_0$ (50 Hz in Europe) and has an internal impedance $Z_0$. It is possible to measure the power, the current or the admittance of the total load. As the appliances are connected in parallel these values measured at the end point are the sum of the respective values measured at each appliance. The measured power and current depend on the line voltage $V$ that is supposed to be constant. In reality this value is time varying $V(t)$ due to variations like rapid fluctuations that can affect the total measure. For these reasons the admittance is a preferable signature for the

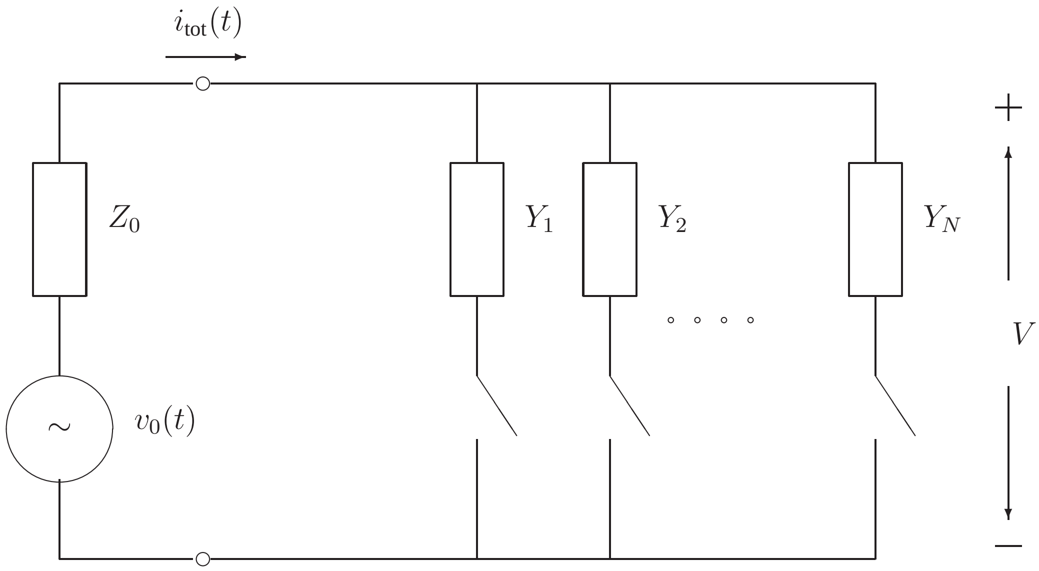



**Figure 1.1:** *Equivalent electrical scheme of a single phase house.*

devices. The admittance is voltage independent and is additive when the devices are connected in parallel.

### Phasor values

The total load admittance $Y(k)$ can be calculated from the measured power $S(k)$ and the RMS voltage $V_{RMS}(k)$:

$$Y(k) = \frac{S(k)}{V_{RMS}^2(k)} \tag{1.1}$$

To better understand the value assumed by the admittance we write it as the normalised power. The normalised power is the value the power is supposed to have with constant voltage $V_{ref}$:

$$\tilde{S}(k) = V_{ref}^2 Y(k) = \left(\frac{V_{ref}}{V_{RMS}}\right)^2 S(k) \tag{1.2}$$

where $V_{ref} = 230[V]$.

The normalised power can be written as sum of the normalized power of the devices that are ON at time $k$:

$$\tilde{S}(k) = \tilde{P}(k) + j\tilde{Q}(k) = V_{ref}^2 \sum_{n=1}^{N} b_{k,n} Y_n(k) \tag{1.3}$$

where $b_{k,n}$ indicates the devices that are ON at time $k$:

$$b_{k,n} = \begin{cases} 1 & \text{if appliance n is ON at time } k \\ 0 & \text{if appliance n is OFF at time } k \end{cases} \tag{1.4}$$

and $N$ is the total number of appliances.

In this analysis we have used the phasor notation. Here the values of voltage, current and complex power are obtained.

We can write the voltage as:

$$v_{0,k}(t) = \sqrt{2}V_{RMS}(k)\sin(\omega t + \alpha(k)) \tag{1.5}$$

where

$$f_0 = \frac{\omega}{2\pi} = 50Hz \tag{1.6}$$

is the fundamental frequency.

The index $k$ is a discrete time index that indicates the number of the period and $\alpha(k)$ is the phase that can change every period $k$.

The Root Mean Square (RMS) value at instant k of the voltage is defined as:

$$V_{RMS}(k) = \sqrt{\frac{1}{T}\int_{t_k}^{t_k+T} v^2(t)dt} \tag{1.7}$$

where $T = \frac{1}{f_0}$ is the period of the waveform and $t_k$ is an arbitrary time instant.

The voltage phasor is defined as:

$$V_0(k) = V_{RMS}(k)e^{j\alpha(k)} \tag{1.8}$$

We can write the overall current as:

$$i_{tot,k}(t) = \sqrt{2}I_{RMS,tot}(k)\sin(\omega t + \beta(k)) \tag{1.9}$$

where

$$I_{RMS}(k) = \sqrt{\frac{1}{T}\int_{t_k}^{t_k+T} i^2(t)dt} \tag{1.10}$$

and $\beta(k)$ is the phase.

The equivalent current phasor is:

$$I_{tot}(k) = I_{RMS,tot}(k)e^{j\beta(k)}. \tag{1.11}$$

Finally the instantaneous power is defined:

$$p_{tot,k}(t) = v_{1,k}(t)i_{tot,k}(t) \tag{1.12}$$

where $v_{1,k}$ is the voltage seen by the devices:

$$v_{1,k} = v_{0,k}(t) - Z_0 i_{tot,k}(k). \tag{1.13}$$

We can assume $Z_0 = 0$ so $v_{1,k} = v_{0,k}$. We can rewrite the instantaneous power as:

$$p_{tot,k}(t) = V_{RMS}(k)I_{RMS,tot}(k)\cos(\phi(k)) - V_{RMS}(k)I_{RMS,tot}(k)\cos(2\omega t + 2\alpha(k) - \phi(k)) \tag{1.14}$$

where $\phi(k) = \alpha(k) - \beta(k)$ is the difference between the phase of the voltage and the phase of the current.

The first term of (1.14) corresponds to the real power:

$$P(k) = \frac{1}{T}\int_{t_k}^{t_k+T} p(t)dt = V_{RMS}(k)I_{RMS,tot}(k)\cos(\phi(k)) \tag{1.15}$$

The complex power $S(k)$ in the phasor notation is defined as:

$$S(k) = V(k)I_{tot}(k) \tag{1.16}$$

We can write it as the sum of the real and reactive power:

$$S(k) = P(k) + jQ(k) \tag{1.17}$$

where the real power $P(k)$, as defined in (1.15), is:

$$P(k) = V_{RMS}(k)I_{RMS,tot}(k)\cos(\phi(k)) \tag{1.18}$$

and the reactive power $Q(k)$ is:

$$Q(k) = V_{RMS}(k)I_{RMS,tot}(k)\sin(\phi(k)). \tag{1.19}$$

### 1.1.2 Signature

The appliance signatures are the essence of the NALM system. To characterise the appliances we need to choose a unique electrical feature which represents the appliances. This feature can be defined as an appliance signature. An appliance signature is a measurable parameter of the total load that gives information about the nature or operating state of an individual appliance in the load.

The signature disaggregation can be intrusive or non-intrusive. The intrusive signatures require a physical or electrical intrusion to measure directly the behaviour of a device.

We are interested in the non-intrusive approach. A non-intrusive signature is the one which can be measured at the central load point. Within the non-intrusive signatures there is a separation in steady state, transient or others signatures. The difference between steady state and transient signatures is the time of extraction of the information that characterises the signature. A transient signature considers only the information during the transition from one state to the other. Transient signatures are more difficult to detect because they are present only for a small period of time. They can be useful to detect appliances that have the same steady signatures but are characterised by a different transient.

On the contrary a steady state signature is continuously present after the transition during the operational time of the appliance. These characteristics make the steady state signatures easier to detect. The steady state signatures derive from the difference between the two operating states before and after the transition. Another characteristic is that they are addictive. If two appliances change status at the same moment the total steady state signature is the sum of the the

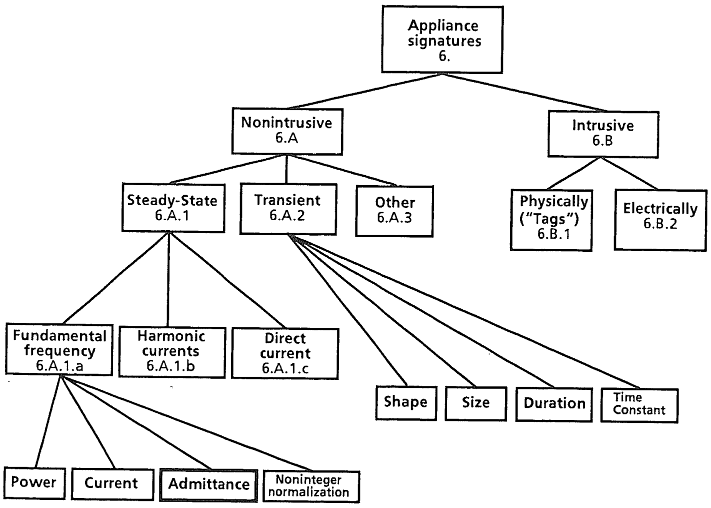



**Figure 1.2:** *Signature*

two signatures if they were operated separately. This property is not always true for the transient signatures.

The steady state signatures can be divided into three groups: fundamental frequency, harmonic currents and direct current. The first group includes all the signatures that can be measured at the utility fundamental frequency (50 Hz in Europe). There are three possible measurable signatures of the total load: power, current or admittance. The three parameters can be considered equivalent as they are related to each other by the line voltage. This is true only in theory while in a real environment the line voltage is not constant but is affected by small fluctuations. For this reason the admittance is more stable than the other two signatures. The admittance is independent from the voltage and is addictive. A common signature to represent the admittance is the normalized power (1.2). This signature is the measured admittance multiplied by a constant scaling factor. This value represents the expected power value if the line voltage was constant.

The fundamental frequency signatures consider all the devices as operating linearly with respect to the current. There are some devices that present non linear components. These components can be detected to characterise the devices in a more complete way. It is possible to differentiate devices with the same signatures at the fundamental frequency but different behaviour at higher frequencies. Two examples are the signatures using harmonic frequency or direct current. The harmonic frequency signatures use the current components that some appliances generate at higher frequency than the fundamental one. The direct current signatures is another non linear property that is present in some device and can be used to characterise them.

### 1.1.3 Algorithm model

The algorithm model introduced by Hart [8] divides the problem in eight steps (Fig. 1.3) described below.

A. **Measure Power and Voltage.** The power is measured at the utility interface. The mea-

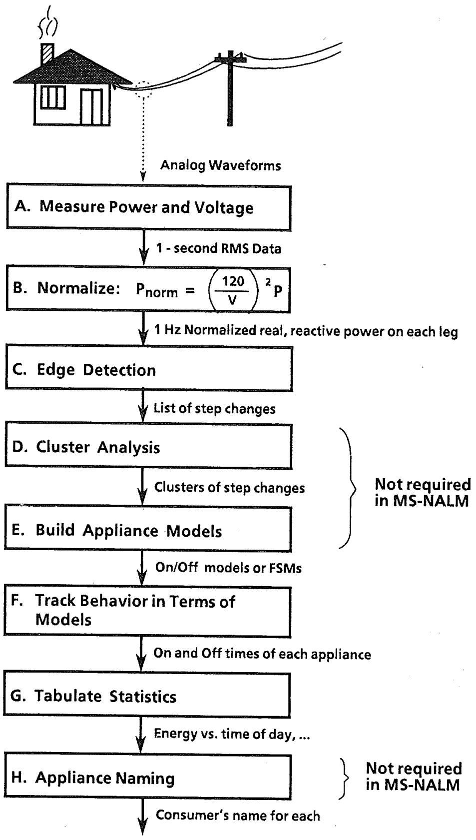



**Figure 1.3:** *Algorithm model*

sured values are averaged over a time window. The choice of the window length is important to determine the separation time between different events. It is not possible to separate events that happen inside the same time window. If the window is shorter it is possible to separate more events.

B. **Calculate Normalised Power.** The normalised power (1.2) is used to extract the events as it is more stable than the complex power because it is not affected by the voltage fluctuations.

C. **Edge Detection.** The edge detection algorithm analyses the normalised power. It extracts the time and size of the rapid changes in the normalized power. There are many techniques to find these values. The most used are signal processing techniques such as filtering, differentiating, and peak detection.

D. **Cluster Analysis.** The size of the changes extracted in the edge detection step are the input for the cluster analysis. The time stamps are not used for the moment. The observed changes define a scatter plot in PQ-plane. The points in the plane are grouped into clusters, that ideally have to represent the state change of one appliance. The cluster size can be different with respect to the appliance state it is represented. Very consistent appliances, for example resistive heaters, are represented by small sized clusters while appliances in which the start up load can be very variable are represented by large sized clusters.

E. **Build Appliance Models.** There are three classes of appliances models: ON/OFF, FSM (finite state machines) for devices with a finite number of states and continuously variable for devices with an infinite number of states. During this step these models for each appliance in the load are created from the cluster output. For example, the ON/OFF model can be constructed coupling the clusters that are symmetrically placed with respect to the origin.

F. **Track Behaviour.** A decoding approach is used to identify the appliances. The appliances are represented by models generated in the previous step.

G. **Tabulate Statistics.** It is possible to tabulate a lot of different statistics about operating power and energy consumed for each device.

H. **Appliance Naming.** A final task for the AS-NALM is to give a name to each detected appliance. The statistics calculated in the previous step can help to differentiate the devices.

### 1.1.4 State of the art

In our analysis we are interested in the state of the art of the C and D steps of the algorithm defined by Hart [8]. The C step identifies the events while the D step creates the clusters to identify the devices. [12] gives an update of the structure and methodology used to solve the NALM problem.

The presented event detector can only extract variations higher than 100 W [12]. This is a strong limitations to identify smaller devices. An overview of the signatures is presented in [9] The appliance usages can be recognized using the real and reactive power features [8] - [5]. More complex device that have similar P-Q characteristics can be identified using other features. In [3] the event detector uses the energy of the transient to separate devices with the same real and reactive power. [17] proposes an approach where the events are identified looking at the shape of the transient. In [11] is also shown how is possible to detect some events when their transient periods are overlapping. In [10], the current waveforms are used to characterize the appliances in a more specific way than the P-Q characteristic. Another problem is the detection of events that do not present a sharp or particular shape transient. A solution to the problem of more complex events is presented in [4]. The transient is approximated as a slow ramp.

The problem of disaggregation presents numerous solutions from different theoretical fields as optimization problem or pattern recognition. An approach using genetic algorithms is presented in [13]. Several studies use neural networks to disaggregate the power [2]- [1]. Another interesting approach using the detection theory of communication system is presented in [7] Most state of the art algorithms for cluster analysis consider clusters in the PQ-plane from the events extracted before. In [16] a self learning process to detect and classify devices based on their electrical behaviour is presented. An approach to detect devices at a circuit level is presented in [14].

## 1.2 Problem definition

The aim of the project is the design of a self-learning algorithm to identify the appliances in a NALM system. Initially, we have to choose which signatures to use to characterise the devices. A signature has to represent the characteristic of an appliance to differentiate each appliance from the others. Then the self-learning algorithm procedure is divided into three parts: steps C, D and E of the algorithm described in section 1.1.3. Step C, edge (or event) detector, identifies appliances changing status and extracts the chosen signatures. The detected changes are called events. Step D, cluster analysis, analyses the identified events to find common patterns representing the appliances. Step E, Build Appliance Model, identifies clusters representing different state of the same device. In our analysis we only consider the ON/OFF model.

### 1.2.1 Signatures choice

The first step in our analysis is the choice of a signature which characterises the device. The choice of the signature is an important step and determines the next steps of the algorithm, event detector and cluster analysis. In our analysis we focus on the steady state signatures, Fig. 1.2. A signature can be of two types: a snapshot form or a delta form. We consider the delta form which is the difference between two consecutive snapshot form signatures. We consider two different signatures to characterise the appliances. The first signature is the delta form of the normalized power $\Delta\tilde{S}$ (1.2). The space of all the normalized power signatures is a two dimensional space. One dimension is the delta form of the real power $\Delta\tilde{P}$, the second one is the delta form of the reactive power $\Delta\tilde{Q}$. We consider an event that begins at the time instant $k_1$ and finishes at the time instant $k_2$. We can write the normalized power signature of the event:

$$\Delta\tilde{S} = \tilde{S}(k_2) - \tilde{S}(k_1) = \Delta\tilde{P} + j\Delta\tilde{Q} \tag{1.20}$$

where $\Delta\tilde{P} = \tilde{P}(k_2) - \tilde{P}(k_1)$ and $\Delta\tilde{Q} = \tilde{Q}(k_2) - \tilde{Q}(k_1)$. This signature utilises only information at the fundamental frequency.

The second signature we utilise is a delta form of the current waveform (1.9). Some devices present a non linear behaviour. The current waveform does not always have a sinusoidal waveform but sometimes has other components at higher frequencies. The delta form of the current waveform is:

$$\Delta i(t) = i_{tot,k_2}(t) - i_{tot,k_1}(t) \tag{1.21}$$

where $k_1$ and $k_2$ represent the starting and ending time of the event and $i_{tot,k}$ is the overall current 1.9 at period $k$. The current waveform is represented by $N_{current} = \frac{F}{f_0}$ samples, where $f_0$ is the fundamental frequency ($50$ Hz in Europe) and $F$ is the sampling frequency. We use a sampling frequency $F = 10000$ Hz, so the current waveform is represented by $N_{current} = 200$ samples. $N_{current}$ is the dimension of the current signature space. Using this signature it is possible to identify devices that have the same power value but different current waveforms.

### 1.2.2 Problems and improvements

The difficulty in the design of the system is to recognise devices with very different electrical behaviours using the same tools. The event detector has to extract significant events which are analysed by the cluster algorithm. We consider significant the events caused by a device that is changing status but we do not want to consider events with changes in power caused by operating status of the devices. Most state-of-the-art algorithms can only detect very large rapid changes in power. This is a strong limitation for the detection of devices which have more complex transients. For example some devices, e.g. vacuum cleaner, switches on as a sequence of steps. The event detector as defined in literature identifies a sequence of state changes and not an unique event as desired. Other devices, e.g. some lamps, switch off very slowly without an instantaneous change in power which make very difficult their detection.

The aim of our project is to develop a self-learning algorithm that allows us to detect and recognise not only devices characterized by simple ON OFF events but also the ones which have a more complex type of transient. The idea to recognise more complex devices is to identify them using the cluster procedure. The event detector has to be able to detect simple step changes in power. The cluster algorithm has to analyse the extracted events to identify more complex patterns. After coupling together clusters representing different states of the same device, the final output of the system is the identification of the devices present in the environment.

The second chapter describes the solution to the event detector problem. The third chapter describes the cluster algorithm and proposes a solution to identify more complex type of devices.

# Chapter 2

# Event detector

The aim of the event detector is to extract significant signatures that characterise the appliances. The chosen signatures are two steady state delta form signatures: the normalised power $\Delta\tilde{S}$ and the current waveform $\Delta i(t)$. These signatures are the difference of the respective overall values in two different instants. The event detector has to identify two steady state instants representing a device changing status and extract the two signatures from the measured voltage and current. These changes are called events.

We calculate the normalised power from the measured voltage and current and we use it as input to the event detector to identify a change. The overall normalised power is the sum of the normalised power of the active devices. If a single device changes power consumption by, e.g. $\Delta\tilde{S}$, the overall power would change by the same amount $\Delta\tilde{S}$. Therefore the device characteristic features can be extracted by observing the variations in the overall normalised power. The key challenges are in coupling the variations of the normalised power to the variation of the status of a device. The difficulty of the problem lies in the different behaviour of different devices. The switching on of a complex device, e.g. a TV, generates a sequence of variations (or small events). These small events can be interpreted as changing status of small devices, e.g. lamps. Another problem is the duration of the changing states. Some devices, e.g. some lamps, do not present a clear step in the power change. The change can be very slow and difficult to detect.

This chapter presents our solution to the problem of the event detection. The chapter is organised into three parts. The first one gives a definition of event. The second one presents the tools and the algorithm that we use to extract the events. The third part evaluates the output of the algorithm on some examples of ideal events. This analysis allows us to estimate the theoretical parameters of the model.

## 2.1 Event definition

The aim of the project is the identification of the devices. We identify the presence of a device when it changes its status. The difficulty in the detection of these status changes consists in different behaviours of different devices. It is not possible to use a unique event detector to identify all the device changing status. The idea is that the event detector has to extract the most

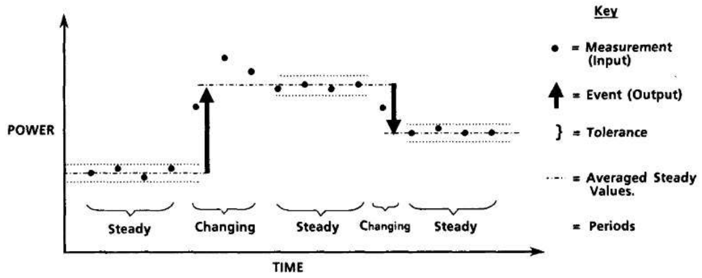



**Figure 2.1:** *Example of the subdivision into stable and changing period by Hart*

simple features. Then the classifier has to analyse them to describe the device changing status.

Fig. 2.1 shows the subdivision in interval representing different periods done by Hart [8].

The input $\tilde{S}(n) : \mathbb{Z} \longrightarrow \mathbb{C}$ to the event detector is the measured overall normalized power at time instant $nT$ where $T$ is the sampling period of the normalised power. We use the normalised power to identify the changes. We introduce two parameters: $N$ and $K$. The first parameter $N$ defines the length of the observation window: $L = 2N$. The input signal is analysed on a sliding window of length $L$. The second parameter $K$ defines the observed value as stable or not. It is used to separate variations due to changing periods from small variations due to noise. We consider the signal in a window centred in $n$. We split the input signal into two parts and calculate the mean of the two parts.

$$\Delta S_{in}(n) = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{S}(n-m-1) \tag{2.1}$$

is the mean of the first part while

$$\Delta S_{fin}(n) = \frac{1}{N} \sum_{m=0}^{N-1} \tilde{S}(n+m) \tag{2.2}$$

is the mean of the second part. We look at the absolute value of the difference of the two mean values and compare this result with the threshold parameter $K$:

$$\Big| \Delta S_{fin}(n) - \Delta S_{in}(n) \Big| > K \tag{2.3}$$

We define an interval of length $L = 2N$ centred in the time instant $n$ a transient if (2.3) is true. A transient interval represents a time period during which a device is changing status. It is characterised by its center value $n$ and the two mean values: $\Delta S_{fin}(n)$ and $\Delta S_{in}(n)$. Different devices are described by different sequences of transients with different mean values. The idea is to use this information as input for the classification. The classifier has to analyse sequences of transients to find out common patterns which represent the device changing status.

However, $\Delta S_{fin}$ and $\Delta S_{in}$ are sensitive to the sampling phase. Transients of the same device can assume very different values. To avoid this limitation, we define an event as the longest sequence of consecutive transients. This feature is less sensitive to the sampling phase.

The output of the event detector is a sequence of events characterized by four values: two signature values, a starting time and a time duration. Let $n_{min}$ be the first time instant of the sequence and $n_{max}$ the last one. The first time instant $n_{min}$ is the time instant where the first transient interval is centred. The last time instant $n_{max}$ is the time instant where the last transient interval is centred.

The signature value $\Delta\tilde{S}$ is equal to the difference of the mean values of the two windows centred after and before the event:

$$\Delta\tilde{S} = (\Delta S_{fin}(n_{max}+1) + \Delta S_{in}(n_{max}+1)) - (\Delta S_{fin}(n_{min}-1) + \Delta S_{in}(n_{min}-1)). \quad (2.4)$$

This value is less sensitive to the sampling phase. The signature value $\Delta i$ is extracted in the same way:

$$\Delta i(t) = (\Delta i_{fin,n_{max}+1}(t) + \Delta i_{in,n_{max}+1}(t)) - (\Delta i_{fin,n_{min}-1}(t) + \Delta i_{in,n_{min}-1}(t)) \quad (2.5)$$

where

$$\Delta i_{fin,n}(t) = \frac{1}{N}\sum_{m=0}^{N-1} i_{tot,n+m}(t) \quad (2.6)$$

and

$$\Delta i_{in,n}(t) = \frac{1}{N}\sum_{m=0}^{N-1} i_{tot,n-m-1}(t). \quad (2.7)$$

The event starting $n_{start}$ is equal to the first time instant of the sequence: $n_{min}$. The time duration $\Delta t_{dur}$ is equal to the length of the sequence:

$$\Delta t_{dur} = n_{max} - n_{min} + 1. \quad (2.8)$$

These four values characterise an event. The events are used to identify the devices in the classifier.

## 2.2 Filter design

In this section we define the functions and the algorithm that are used to extract the events from normalised power $\tilde{S}(n) : \mathbb{Z} \longrightarrow \mathbb{C}$ using the two parameters $(N, K)$. The overall current $i_{tot,n}(t)$ is used as input to calculate the current signature. We define a filter $h_N(n)$:

$$h_N(n) = \frac{1}{N}R_N(n+N-1) - \frac{1}{N}R_N(n-1) \quad (2.9)$$

where $R_N$ is the discrete rect function:

$$R_N(n) = \begin{cases} 1 & \text{if } 0 \leq n < N \\ 0 & \text{elsewhere} \end{cases} \quad (2.10)$$

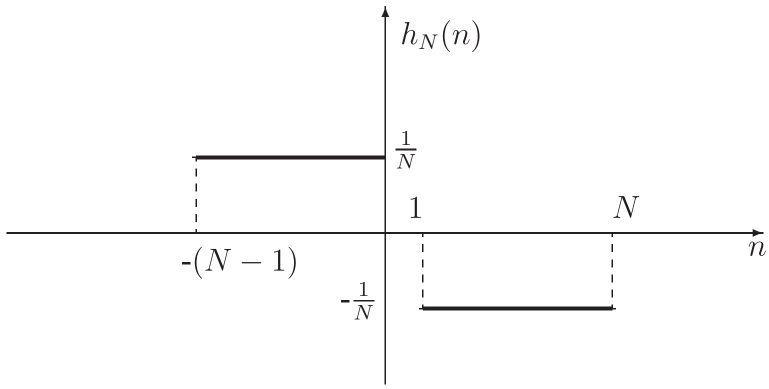



**Figure 2.2:** *Filter* $h_N(n)$

The output of the filter:

$$y_N(n) = (\tilde{S} * h_N)(n) = \sum_{m=-\infty}^{\infty} \tilde{S}(m) h_N(n-m) \tag{2.11}$$

represents the difference between the two values $\Delta S_{fin}(n)$ and $\Delta S_{in}(n)$ in the observation interval centred in $n$. If $|y_N(n)| > K$ the interval $[\tilde{S}(n-N), \tilde{S}(n+N-1)]$ is a transient (2.3). Otherwise it is stable. An event is the longest sequence of consecutive values such that $|y_N(n)| > K$.

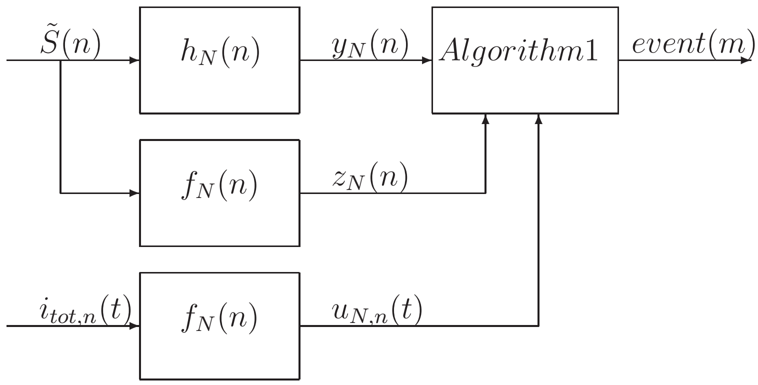



**Figure 2.3:** *Event detection*

The procedure to extract the events is described in Algorithm 1 (Extract events). The algorithm to extract the events operates as follows. If $|y_N(n)| \leq K$, $n$ belongs to a stable period. An event is starting when we find the first value $n_1$ such that $|y_N(n_1)| > K$. The event lasts until

**Algorithm 1** Extract events

$n \leftarrow 1$ {n time instant}
$m \leftarrow 0$ {m total number of detected events}
$event \leftarrow \emptyset$ {detected events}
{M total number of samples} {K threshold value} {L=2N filter length}
**while** $n \leq M$ **do**
  **if** $|y_N(n)| > K$ **then**
    $m \leftarrow m + 1$
    $S_{iniz} \leftarrow z_N(n-1)$
    $i_{iniz}(t) \leftarrow u_{N,n-1}(t)$
    $n_{start}(m) \leftarrow n$
    $\Delta t_{dur}(m) \leftarrow 1$
    $n \leftarrow n + 1$
    **while** $|y_N(n)| > K$ **do**
      $n \leftarrow n + 1$
      $\Delta t_{dur}(m) \leftarrow \Delta t_{dur}(m) + 1$
    **end while**
    $\Delta\tilde{S}(m) \leftarrow z_N(n) - S_{iniz}$
    $\Delta i_m(t) \leftarrow u_{N,n}(t) - i_{iniz}(t)$
    $event(m) = \{\Delta\tilde{S}(m), \Delta i_m(t), n_{start}(m), \Delta t_{dur}(m)\}$ {values of the m-event}
  **end if**
**end while**

is found a final value $n_2$ such that $|y_N(n_2)| \leq K$. The value $n_2$ belongs to a steady period. The starting time of the event $n_{start} = n_1$ and the time duration $\Delta t_{dur} = n_2 - n_1$. To extract the event value we introduce a new filter $f_N(n)$:

$$f_N(n) = \frac{1}{2N} R_{2N}(n+N) \tag{2.12}$$

This filter is a mean function and it is used to reduce the noise in the steady period when the step change of the event is calculated.

The output

$$z_N(n) = (\tilde{S} * f_N)(n) = \sum_{m=-\infty}^{\infty} \tilde{S}(m) f_N(n-m) \tag{2.13}$$

is the mean of the input signal $\tilde{S}(n)$ on intervals of length $L_{f_N} = 2N$. The power signature is calculated as the difference $\Delta\tilde{S}(1) = z_N(n_2) - z_N(n_1 - 1)$. The output

$$u_{N,n}(t) = \sum_{m=-\infty}^{\infty} i_{tot,m}(t) f_N(n-m) \tag{2.14}$$

is the mean of the current $i_{tot,n}(t)$. The current signature is calculated as the difference $\Delta i_1(t) = u_{N,n_2}(t) - u_{N,n_1-1}(t)$.

When the event signatures are extracted we can proceed to find the next event. The next value $n_3$ such that $|y(n_3)| > K$ is the starting time of the second event. The end of the event and its values are extracted in the same way described for the first event. Then the procedure follows detecting the next events. This procedure identifies all the events and their characteristic values.

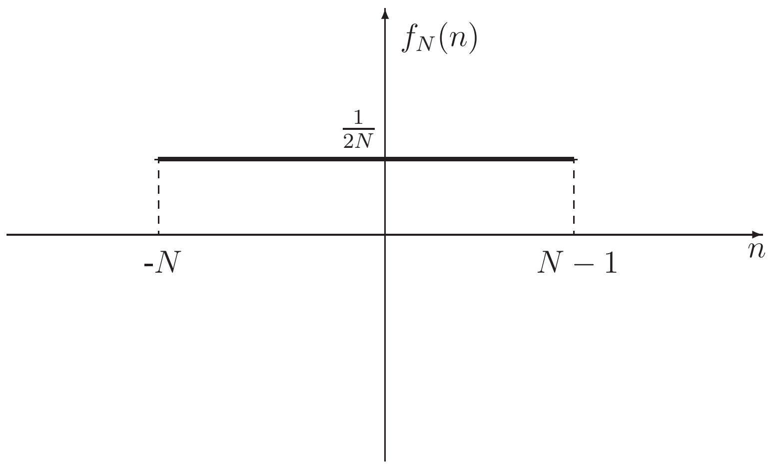



**Figure 2.4:** *Filter* $f_N(n)$

The final outputs of the event detector are the four characteristic values of each detected event. The event detector scheme is shown in Fig. 2.3.

## 2.3 Ideal input

In this section we analyse the output of the event detector for three different ideal input signals. The output result depends on the choice of the two parameters $(N, K)$. The comparison of the different outputs, changing these two parameters, allows us to estimate a theoretical choice for the two parameters $N$ and $K$.

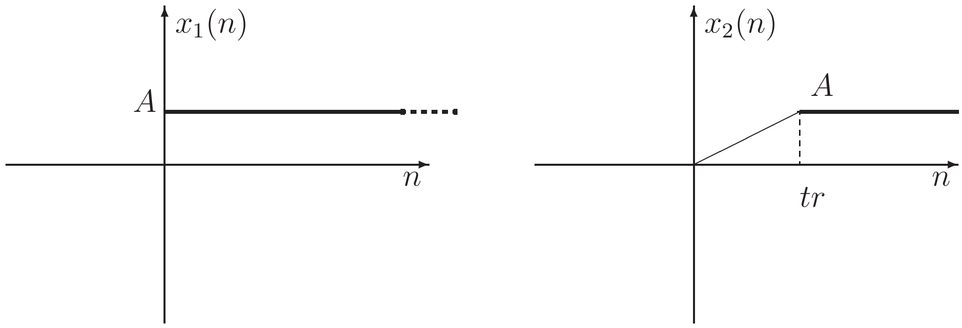



**Figure 2.5:** *Input* $x_1(n)$ *and* $x_2(n)$

We consider three ideal inputs to describe the switching on of the devices. More complex examples of switching on can be modelled as sequences of these three inputs. We analyse only the switching on because the switching off can be modelled as the opposite of these signals. The theoretical analysis for the switching off is symmetric.

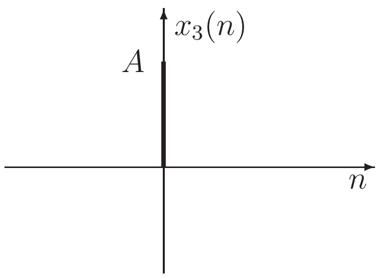



**Figure 2.6:** *Input* $x_3(n)$

The first input, in Fig. 2.5, is a step of amplitude $A$:

$$x_1(n) = Au(n), \tag{2.15}$$

where $u(n)$ is the discrete step, defined as:

$$u(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{if } n < 0 \end{cases} . \tag{2.16}$$

The second one, in Fig. 2.5, is a ramp with slope of $\frac{A}{tr}$:

$$x_2(n) = \begin{cases} A & \text{if } n \geq tr \\ \frac{A}{tr}n & \text{if } 0 \leq n < tr \\ 0 & \text{if } n < 0 \end{cases}, \tag{2.17}$$

where $tr$ is the length of the changing period.

$$x_3(n) = A\delta(n) = \begin{cases} A & \text{if } n = 0 \\ 0 & \text{if } n \neq 0 \end{cases}. \tag{2.18}$$

The third example (2.18) analyses the problem of an event that we do not want to detect. We propose an example of a very sharp jump that lasts only for one sample, figure 2.6.

We are interested in the detection of the event. We do not look at the event signatures. The expected results for the first input $x_1(n)$ are: $n_{start} = 0$ and time duration $\Delta t_{dur} = 1$. The expected results for the second input $x_2(n)$ are: $n_{start} = 1$ and time duration $\Delta t_{dur} = tr$. Both inputs have a final amplitude $\Delta\tilde{S} = A$. The expected result for the third input $x_3(n)$ is no detection of an event.

### 2.3.1 Output results

We compare the different outputs in relation with the two parameters $(N, K)$. This analysis allows us to make a choice on the values of $(N, K)$ with respect to the events we want to detect. The output $y_N(n)$ only depends on the parameter $N$, where $L_{h_N} = 2N$ is the length of the filter $h_N(n)$.

The output for the first input is:

$$y_{1,N}(n) = (x_1 * h_N)(n) = \sum_{m=-\infty}^{\infty} x_1(m)h_N(n-m) = A\text{tri}\Big(\frac{n}{N}\Big) \tag{2.19}$$

where

$$\text{tri}(n) = \begin{cases} 1 - |n| & \text{if } |n| \leq 1 \\ 0 & \text{elsewhere} \end{cases}. \tag{2.20}$$

In Figure 2.7 we can observe that as $N$ increases as the support of the output becomes larger. The maximum is in $n_{max} = 0$ for all the outputs $y_{1,N}(n)$ for $N = 1, .., \infty$. The maximum value is the same for all the output functions and is equal to the amplitude of the step function $A$. This value gives the starting time of the event $n_{start} = n_{in} = 0$ and the amplitude $y(n_{max}) = A$. The event is identified for all the possible values of $N$ if $K < A$.

The output $y_{2,N}(n)$ of the second input $x_2(n)$ is a symmetric function centred at the time instant $n_{max} = \frac{tr+1}{2}$. The output $y_{2,N}(n)$ has its maximum value in $n_{max}$. If $tr$ is an even number we have at least two maximum values situated in $n_{1,max} = \frac{tr}{2}$ and in $n_{2,max} = \frac{tr}{2} + 1$. This is depending only on the sampling rate, so we can consider $tr$ as an odd number to have only

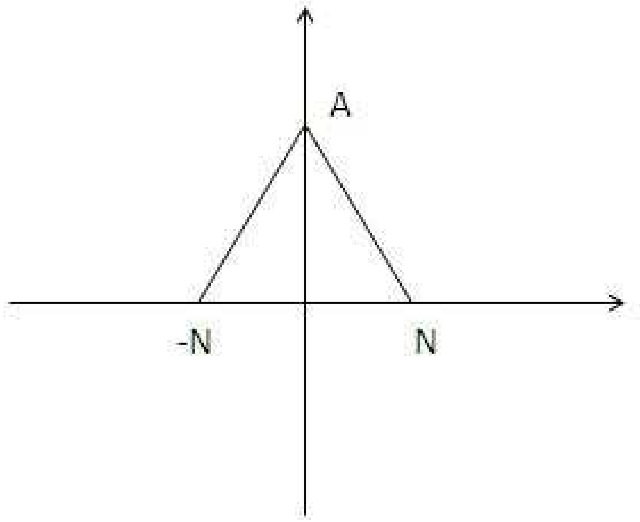



**Figure 2.7:** *Output $y_{1,N}(n)$ for the first input $x_1(n)$.*

| $N$ | $y(n_{max})$ | Number maximum values $L_{max}$ | Length support $L_{supp}$ |
|---|---|---|---|
| $< \frac{tr}{2}$ | $\frac{A}{tr}N$ | $tr - 2(N-1)$ | $2N + tr - 2$ |
| $> \frac{tr}{2}$ | $A\left(1 - \frac{(tr-1)(tr+1)}{4Ntr}\right)$ | 1 | $2N + tr - 2$ |

**Table 2.1:** *Output values for the second input $x_2(nT)$*

one maximum value to simplify the result. We consider two different values of $L_{h_N} = 2N$ filter length in comparison with the transient length $tr$. The first one is when the filter length is shorter that the transient: $N < \frac{tr}{2}$. The second one is when it is longer: $N > \frac{tr}{2}$. If $N < \frac{tr}{2}$ the output does not have a unique maximum value. It will have a sequence of maximum of length $L_{max} = tr - 2(N-1)$ with maximum value $y(n_{max}) = \frac{A}{tr}N$. As the filter length $L_{h_N}$ increases the maximum value increases and the number of maximum values decreases. We find only one maximum value if $N > \frac{tr}{2}$. We calculate the maximum value, the number of maximum values and the length of the support. The results are shown in Table 2.1. The event is detected if $K < y(n_{max})$. If the filter length is longer than the transient there is only one maximum value. The maximum approaches $A$ with the increasing of $N$. The length of the support is $L_{supp} = 2N + tr - 2$. Another observation about the filter length is that the longer the filter the higher the support of the output function increases. This can be a problem in the overlapping of events that happen close together. This problem will be analysed in the Section 2.3.2.

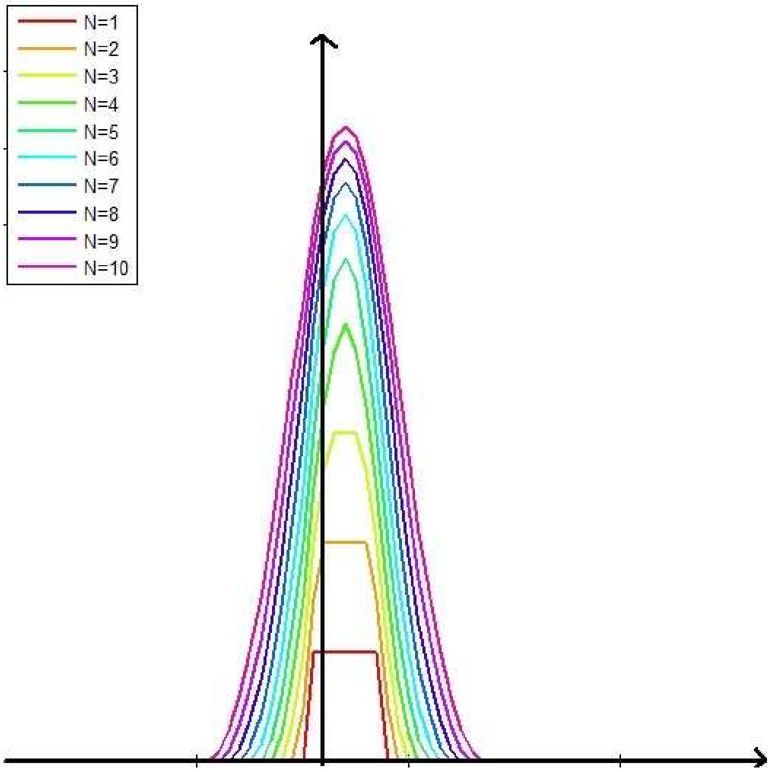



**Figure 2.8:** *Output $y_{2,N}(n)$ for a ramp of slope $\frac{100}{7}$*

In Fig. 2.8 the output $y_{2,N}(n)$ is shown for an input signal with amplitude $A = 100$ and transient of length $tr = 7$. The outputs are obtained from the different filters for $N = 1, ..., 10$. We can observe that the green output ($N = 4$) is the first having only one maximum value, $N = 4 > \frac{tr}{2}$. As $N$ increases the maximum approaches to the amplitude value $A$.

The last example $x_3(n)$ is an input signal that we do not want to identify as an event because this signal does not imply a persistent change in the overall power.

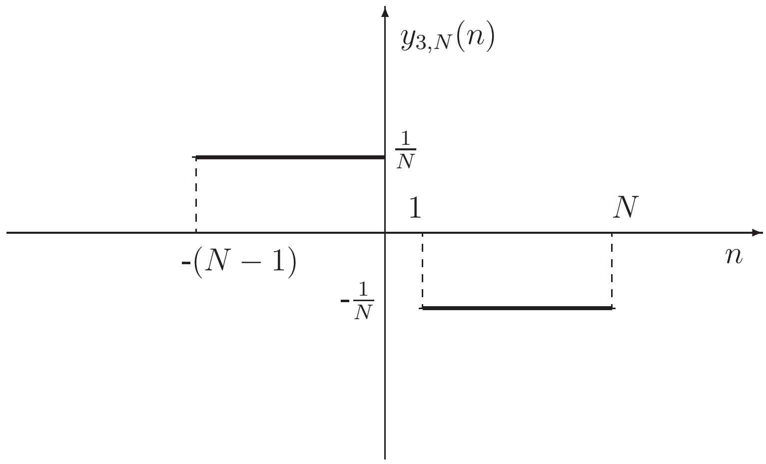



**Figure 2.9:** *Output* $y_{3,N}$

$$y_{3,N}(n) = (x_3 * h_N)(n) = Ah_N(n) \tag{2.21}$$

The output of the filter $y_{3,N}(n)$ changing the parameter $N$ is reported in (2.21). The filter analysis shows that as the filter length increases the maximum value decreases (fig. 2.9). The event is not detected if $K > \frac{A}{N}$

### 2.3.2 Separation of different events

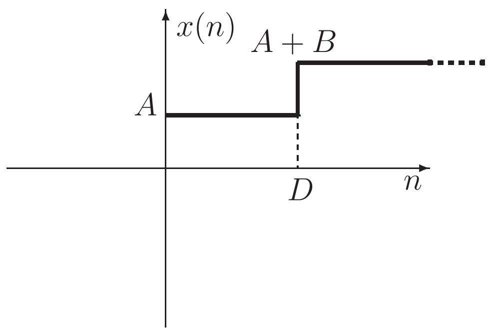



**Figure 2.10:** *Input* $x(n)$

Until now we have considered only events that happen separately from each other. The previous results show that if the filter is longer we can detect more events and do not detect very short jumps. A problem we have not considered is the separation of events that happen close to each other. When two events are close together it is usually better to use a short filter to detect them. If the filter is too long the event detector identifies only one event.

We consider a simple example: two events that have a step transient (Fig. 2.10):

$$x(n) = \begin{cases} 0 & \text{if } n < 0 \\ A & \text{if } 0 \leq n < D \\ A + B & \text{if } n \geq D \end{cases} \tag{2.22}$$

The first event has an amplitude $\Delta\tilde{S}_1 = A$ and starts at time $n_1 = 0$. The second has an amplitude $\Delta\tilde{S}_2 = B$ and starts at time $n_2 = D$. The time distance between them is $D$ samples.

The output function $y_N(n)$, in Fig. 2.11, is equal to:

$$y_N(n) = (x * h_N)(n) = A\text{tri}\left(\frac{n}{N}\right) + B\text{tri}\left(\frac{n-D}{N}\right) \tag{2.23}$$

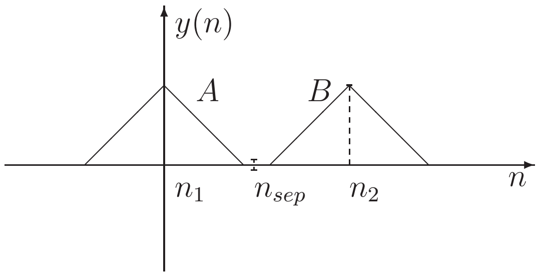



**Figure 2.11:** *Output* $y(n)$

We want to investigate how the filter length can change the detection of the two events. We consider different values of the filter length $L_{h_N} = 2N$ in comparison with the time distance between the two events $D$. To simplify the mathematical analysis we consider D as an even number. We evaluate the result in $n_{sep} = \frac{D}{2}$. The expected result is a very low value in $n_{sep}$ and two peaks in $n_1$ and $n_2$. The expected values of the two peaks are the amplitude values: $y(n_1) = A$ and $y(n_2) = B$.

| $N$ | $y(n_1)$ | $y(n_2)$ | $y(n_{sep})$ |
|---|---|---|---|
| Expected values | $A$ | $B$ | 0 |
| $< \frac{D}{2}$ | $A$ | $B$ | 0 |
| $= \frac{D}{2}$ | $A$ | $B$ | $\frac{A+B}{2} = \frac{2(A+B)}{D}$ |
| $> \frac{D}{2}$ | $A$ | $B$ | $\frac{A+B}{N}\left(N - \frac{D}{2}\right)$ |
| $> D$ | $> A$ | $> B$ | $\frac{A+B}{N}\left(N - \frac{D}{2}\right)$ |

**Table 2.2:** *Output changing the filter length* $L_{h_N} = 2N$

The results in table 2.2 show the differences in the output values changing the filter length. If the length of the filter is lower than the time difference $D$, then the two events are separable and recognizable for all the values of $K$. If the value is higher than $2D$ than the events are not separable and the values in the peak points are different from the original jump. If the filter length is between $D$ and $2D$ it is not always possible to separate the events. The result depends on the choice of the second parameter $K$. The two events are separable if $K > y(n_{sep})$:

$$K > (A + B)\Big(1 - \frac{D}{2N}\Big) \tag{2.24}$$

### 2.3.3 Parameters N-K

The choice of the two parameters $(N, K)$ determines which events are detected with respect to the amplitude of the jump and the type of event. In the previous part we have analysed the different output results for three types of transient: a step, a ramp and an impulse $\delta(.)$. We determine which events are detected with respect to the choice of $(N, K)$.

The first input $x_1(n)$ is an ideal step of amplitude $A$. We can detect all the events of the first type with amplitude $A > K$ for all the possible values of $N$. The detection of this type of events is independent of the filter length $L_{h_N} = 2N$. The parameter $K$ determines the smallest amplitude we can detect. This first example suggests to choose a small value of $K$ to detect also small variations due to small devices. It does not give any information about the length of the filter $L = 2N$. We define:

$$amplitude_{min} = K \tag{2.25}$$

as the minimum step amplitude that is detectable fixed the values $N$ and $K$.

The second input $x_2(n)$ is a ramp of slope $\frac{A}{tr}$. $A$ is the final amplitude and $tr$ is the duration of the changing period in samples. We divide these events in two big categories with respect to the length of the filter $L_{h_N} = 2N$. The first one is composed by short events where $tr < 2N$ and the second one is composed by long events where $tr > 2N$. The events of the first group are detected if:

$$K < A\Big(1 - \frac{(tr - 1)(tr + 1)}{4Ntr}\Big). \tag{2.26}$$

The events of the second group are detected if:

$$K < \frac{A}{tr}N. \tag{2.27}$$

In this second example the detection depends also on the choice of the parameter $N$. This example suggests to choose a big value of $N$ to detect also events characterised by a long transient. The value of the threshold $K$ has to be small like the previous example. We define:

$$slope_{min} = \frac{K}{N} \tag{2.28}$$

as the minimum slope we can detect fixed the values $N$ and $K$.

The last example $x_3(n)$ determines a lower bound for $K$. This example shows a type of event we do not want to detect. We do not detect an event of the third type with amplitude A only if:

$$K > \frac{A}{N} \tag{2.29}$$

This example fixes a lower bound for the threshold value $K$. It also suggests to choose a large value of $N$ like the previous examples. We define:

$$impulse_{max} = KN \tag{2.30}$$

as the maximum amplitude an impulse event can assume that is not detectable. This analysis shows that the best choices are large values for the parameters $N$ and $K$.

We want to choose a value of $K$ small enough to detect also the small variations of the first type, but large enough to do not detect a sharp instantaneous event of the third type. The three parameters $amplitude_{min}$, $slope_{min}$ and $impulse_{max}$ give an indication on which events are detectable for given parameters $(N, K)$.

This first analysis let us think that if we choose a large value of $N$ we can detect all the relevant events, also the ones that are characterised by a very long transient and discard the sharp jumps that last only for a very short time. On the other hand, if we choose a very large value of $N$ it is difficult to separate two events temporally close together. We investigate the minimum distance necessary between two events to detect them as two separate events. The distance between two events is equal to the distance between the end of the first event and the beginning of the second event. We analyse the result of two step transients described in the previous section. The two events are separable if

$$K > \frac{A+B}{N}\Big(N - \frac{D}{2}\Big). \tag{2.31}$$

When the two parameters $(N, K)$ are fixed the minimum required distance to separate two events is:

$$D_{min} = 2N\Big(1 - \frac{K}{A+B}\Big) \tag{2.32}$$

This result gives an upper bound to the parameter $N$ and a lower bound to the parameter $K$. If we choose a smaller value for the filter length $L_{h_N} = 2N$ it is possible to separate events that are closer to each other. This example shows the result when the two input functions have a step transient. The distance $D$ is different if the events do not have a step transient. We can write the distance $D$ in the case that the two events are two ramps with transient lengths $tr_1$ and $tr_2$. The output has two maxima centred in $n_1$ and $n_2$, the middle of the two ramps. The distance $D$ for two ramps transient is:

$$D_{ramp} = \Big(n_2 - \frac{tr_2+1}{2}\Big) - \Big(n_1 + \frac{tr_1+1}{2}\Big). \tag{2.33}$$

The values of the amplitude $(A + B)$ changes in $\frac{A}{tr_1} + \frac{B}{tr_2}$.

These results determine the detectable events with the choice of the two parameters $(N, K)$. This choice also determines the minimum distance necessary to separate two events. The final choice suggests an upper bound for the filter length $L_{h_N} = 2N$ with respect to the minimum distance to separate two events. The threshold value $K$ has to be small enough to detect also variations due to small loads but big enough to not detect events that are not significant and to separate events that are close one to the other.

## 2.4 Conclusions

The event detector analysis gives a theoretical background to the possible events that we can detect. The choice of the two parameters $(N, K)$ determines which events are detectable. The choice of the two parameters determines four values: $amplitude_{min}$ the minimum instantaneous amplitude that is detectable, $slope_{min}$ the slower slope that is detectable, $impulse_{max}$ the maximum impulse value that is tolerable and $D_{min}$ the minimum distance between two events that is necessary to separate them.

# Chapter 3

# Cluster analysis

The goal of the second part of our analysis is to find frequent patterns in the extracted events to identify the appliances. We detect an event when there is a change in the total power consumption. The change is supposed to be caused by a device changing status. The cluster analysis looks for similar patterns in the extracted event. We want to group together similar events. Each group (cluster) has to represent a device changing status.

The chapter is divided into three parts. The first part gives a short overview of the clusters algorithm and characteristics. The second part introduces the chosen algorithm and the modifications we add to it. The last part introduces an approach to identify more complex devices.

## 3.1 Event Clustering

Clustering tries to find natural grouping in a set of data. There are several ways to define the concept of natural grouping. Here we give an overview of the four most significant definitions.

The first one, well-separated clusters, is an ideal definition of clusters. A cluster is defined as a set of objects in which each object is more similar to every other object in the cluster than to any object not in the cluster. This definition is applicable only when the data have a structure where each cluster is quite far from the others. The second one, prototype-based, uses a prototype element to define a cluster. A cluster is defined as all the data objects that are closer to the prototype that defines the cluster than to any other prototype. The prototype is often the mean of the elements of the cluster. The third one, contiguity-based, uses the relations between objects. A cluster is composed by all the points that are closer to at least one point of the cluster than to any other point in another cluster. These clusters are sensitive to noise. Noise can create connections between different clusters. The fourth one, density- based, uses density to define a cluster. A cluster is defined as a dense region of objects that is surrounded by a region of low density. This definition is useful when the clusters have an irregular shape and in presence of noise. Noise is modelled as a low density region.

There are other properties in the cluster definitions that express the relation between clusters and between an object and the clusters. Here we describe the main differences. We can distinguish various types of clustering. The most common distinction is if the set of clusters is hierarchical or partitional. A partitional clustering is a division of the set of data into disjoint clusters such that each data object is in exactly one subset. A hierarchical clustering is obtained if clusters have subclusters. The structure of a hierarchical cluster is a tree where each node, except for the leaf nodes, is a cluster. The leaf nodes are the data objects and each cluster is the union of its children (subclusters). The root of the tree is the cluster containing all the data objects. Furthermore a cluster can be exclusive, overlapping or fuzzy. In an exclusive cluster each data object belongs only to one cluster. On the contrary if a data object can belong to more than one group we have an overlapping clustering. In a fuzzy clustering every object has a weight related to each cluster. The weight is between 0 and 1 and expresses the probability to belong to that cluster. Finally we can distinguish in complete or partial clustering. While a complete clustering assigns every object to a cluster the partial does not. The partial cluster is used if there is noise or outliers in the data set.

The previous definitions use the concept of similarity between objects to define the clusters. A problem, in the clustering analysis, is how to define the similarity between two data objects. There are several measure to compare the objects. The most common measure is the distance between the data objects.

### 3.1.1 Choices

The events extracted from the event detector are the input of our cluster algorithm. The event detector identifies changes in the normalized power. The events are not always representative of a significant change. For example two devices that switch on at the same time cause a change in the normalized power that is the sum of their expected values. This value is considered as an outlier. It will happen only few times and we want to discard it as an outlier. Another characteristic of the input data is the difference of the data distribution for data representing different devices. The extracted data can be a little bit different for different devices. This is due to their electrical behaviour. In the cluster analysis this causes clusters of different shapes. These are the main reasons to choose a density based clustering approach. In a density based approach we divide the input data space in high and low density regions. We can handle the problem of outliers that are modelled as low density regions. We can create clusters of different shapes for different devices.

In relation to the property of the cluster, we choose a partitional, exclusive, partial clustering. We choose a partitional cluster because we want each cluster to be representative of only one device status. In the exclusive cluster each element belongs only to one cluster. We choose the exclusive cluster because we assume that the elements of a cluster have to be relevant. If they are not representative of that cluster we prefer to discard them. Finally we choose a partial cluster to deal with noise. Using a partial cluster we can discard elements we do not consider relevant and keep only the data we are interested in.

The used metric is the euclidean distance.

## 3.2 Clustering algorithm

We use a density based approach and a structure based on graphs to represent the density regions. The nodes are the data objects. A cluster can be defined as a group of data objects that are connected to each other. Elements in different clusters are not connected together. The similarity measure can be used to create the connections between nodes. For example, if we use the euclidean distance as measure of similarity, we can define a connection if the euclidean distance between two data objects is less than a given threshold. In general we define $d(p, q)$ as the distance between two nodes $p$ and $q$ according to the metric we choose.

We give some definitions of the relations between objects in a graph structure [6].

There are several methods to define density. We use the center-based approach where density is estimated for each point in the data set by counting the number of points inside the circular region of radius $\epsilon$ centred in the point we want to estimate its density. The neighbours of a point $p$ given the radius $\epsilon$ are the elements of the set:

$$N_\epsilon(p) = \{q|d(p, q) \leq \epsilon\}. \tag{3.1}$$

A point is defined belonging to a dense region if the number of neighbours inside the circular region of radius $\epsilon$ is more than a given number $P$:

$$N_\epsilon \geq P \tag{3.2}$$

The density will depend on the choice of the radius $\epsilon$ and the threshold $P$.

These two definitions give an indication of the type of regions a point belongs to. Now we introduce other definitions that express the relation between different points.

The first definition defines which points are connected to a point in a dense region.

**Definition 1 (Directly density-reachable)** *A point p is directly density-reachable form a point q given, $\epsilon$, $P$ if:*

- $p \in N_\epsilon(q)$
- $N_\epsilon(q) \geq P$

The second definition, density reachable, determines the connections between two points using a chain of directly density reachable points between them. Two density-reachable points are shown in Fig. 3.1.

**Definition 2 (Density-reachable)** *A point p is density-reachable from a point q given $\epsilon$, $P$ if there is a chain of points $p_1, ..., p_n, p_1 = q, p_n = p$ such that $p_{i+1}$ is directly density-reachable from $p_i$.*

Finally, the definition of density-connected points defines the connection between two points if they are both density reachable from a common point.

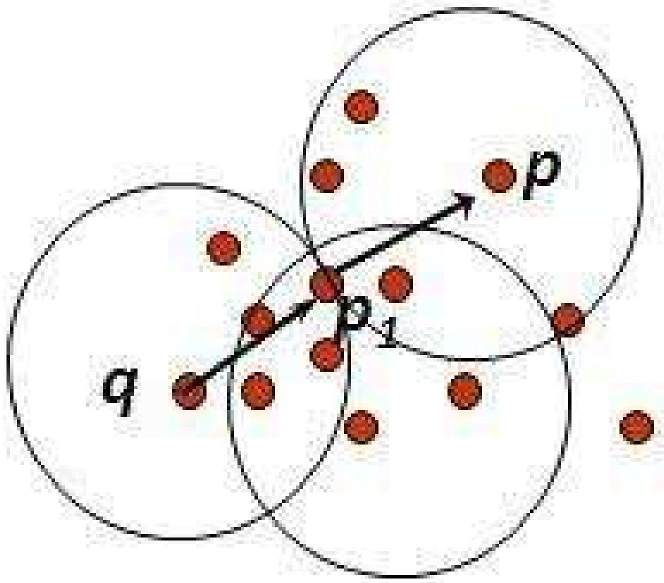



**Figure 3.1:** *Density reachable*

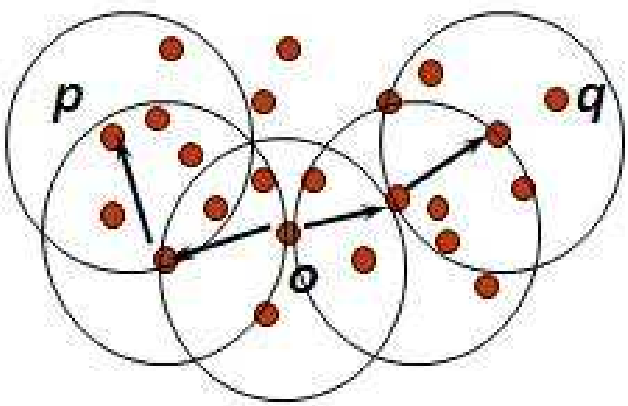



**Figure 3.2:** *Density connected*

**Definition 3 (Density-connected)** *A point p is density-connected to a point q given* $\epsilon$, $P$ *if there is a point o such that both, p and q are density-reachable from o given* $\epsilon$ *and* $P$.

This definition allows to create group of points in dense region that are connected together. The density-connected relation is shown in Fig. 3.2.

We describe an algorithm that uses this structure to create clusters.

### 3.2.1 DBSCAN Algorithm

DBSCAN (for Density-Based Spatial Clustering of Applications with Noise) is a density based clustering algorithm. It was proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996 [6]. In a density based algorithm low density regions separate high density regions.

The algorithm is divided into two parts. In the first part, the points belonging to a dense region are identified and the connection between points are created. In the second part, the connections between points are analysed to create the clusters.

The algorithm uses two parameters to define density: $Eps$ and $MinPts$. Two points are neighbours if the distance between them is less than $Eps$. A point belongs to a dense region if it has at least $MinPts$ neighbours in a circular region of radius $Eps$ centred on itself, as shown in (3.3).

The points of the data set can be classified into three categories: core, border or noise points based on the density of the region they belong to:

- Core points: A point is a core point if the number of points within a radius $Eps$ exceeds the threshold $MinPts$. It is a core point if it satisfies condition (3.2).
- Border points: A border point is not a core point but it is a neighbour of a core point.
- Noise points: A noise point is a point that is neither a core point nor a border point.

Each node is labelled as core, border or noise point.

After classifying the points, the first part of the algorithm creates the connections between points.

The graph related to the algorithm uses direct edges to connect the points. An edge can start only from a core point. An edge is created between each core points and all its neighbours.

The second part uses the structure to create the clusters. The DBSCAN algorithm puts in the same cluster the core points that are connected by an edge. The border points are put in the cluster of the closest core point connected to them. The noise points are discarded as outliers.

Using the definitions of the previous sections, a cluster is defined as the maximal set of density-connected points.

A cluster $C$ in the data set $D$ (with parameters $Eps$ and $MinPts$) is a set of points of $D$ such that:

- $\forall p, q \in D$ if $p \in C$ and $q$ is density reachable from $p$ wrt. $Eps$ and $MinPts$ then $q \in C$

- $\forall p, q \in C$ $p$ is density connected to $q$

The algorithm procedure is described in Algorithm 2.

**Algorithm 2** DBSCAN

Label all points as core, border or noise points
Eliminate noise points
Put an edge between all core points that are within $Eps$ of each other
Make each group of connected core points into a separate cluster
Assign each border point to one of the clusters of its associated closest core points

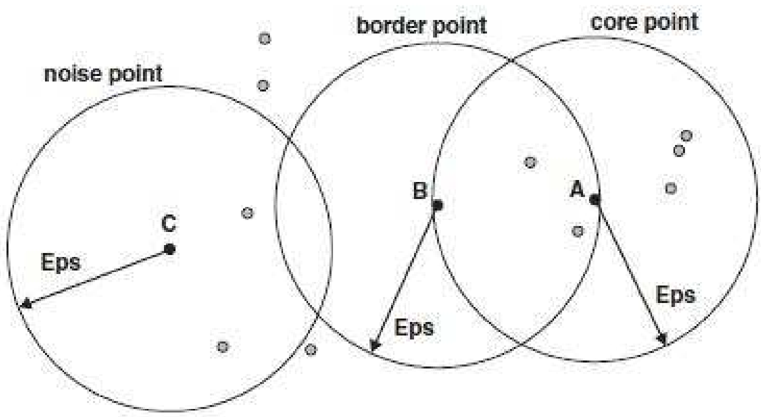



**Figure 3.3:** *Core, border and noise points*

In Fig. 3.4 the cluster formed by the different labelled nodes is shown.

#### 3.2.1.1 Complexity

The time complexity of the DBSCAN algorithm depends on the time to find the neighbours of a point. The algorithm executes that query exactly one time for each point. The time complexity is $O(n^2)$, where $n$ is the number of points, with a proper indexing structure, such as kd-trees, the complexity can be reduced to $O(n \log n)$.

The memory required is $O(n)$ because for each point it is only necessary to save few data such as assigned cluster and the identification of a point as core, border or noise point.

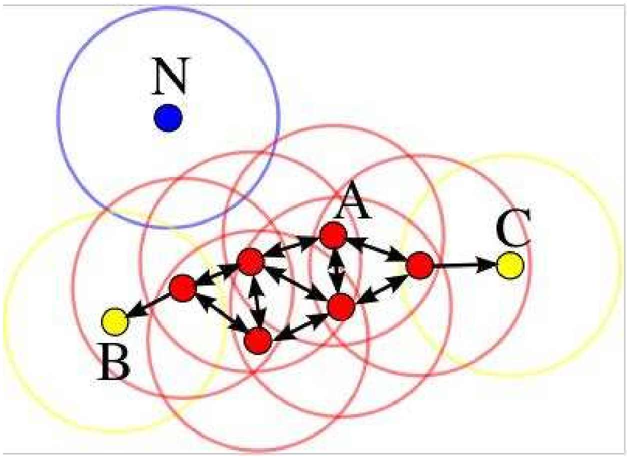



**Figure 3.4:** *DBSCAN Cluster: red are core points, yellow are border point and blue are noise point*

### 3.2.2 Parameters

The metric $d(p,q)$ used to define the distance between two points $p$ and $q$ in the dataset $D$ is the euclidean distance:

$$d(p,q) = ||p-q|| = \sqrt{(\sum_{k=1}^{S_d} |p_k - q_k|^2)} \tag{3.3}$$

where $S_d$ is the dimension of the element $p \in D$.

The parameters that we have to choose are the radius $Eps$ and the minimum number of points $MinPts$. Two points $p, q$ are connected if:

$$||p-q|| < Eps \tag{3.4}$$

A common problem in the algorithms based on density is to find a common radius $Eps$ to define regions with different density. The DBSCAN does not solve this problem. The radius $Eps$ is the same for all the clusters. In our data set we notice that the elements in a cluster with larger absolute mean value are more spread than the ones within a cluster with lower mean value. This characteristic is due to the electrical behaviour of the devices. We introduce a new parameter $\alpha$ to give a different weight to the radius, based on the input values. Using this new parameter the radius is not a fixed value but it depends on the input absolute value. Two points $p, q$ are classified as neighbours if:

$$||p-q|| < Eps + \alpha \left|\left|\frac{p+q}{2}\right|\right| \tag{3.5}$$

The second parameter $MinPts$ determines the label we associate to each node: core, border or noise. We introduce another normalization factor based on the dimension of the data set. The idea is that if the data set increases we need a region to be more dense to be considered as a new cluster. If we do not use a normalization based on the total number of elements it is possible that outliers values create new clusters. We introduce a normalization factor $\beta$.

$$MinPts = MinPts_{in} + \beta * M \tag{3.6}$$

where $M$ is the total number of identified events. We fix the value of $MinPts_{in} = 3$ and $\beta = 0.01$. The values $Eps$ and $\alpha$ depend on the characteristics of the input data set.

### 3.2.3 Input-Output results

The input to the clustering algorithm is the output of the event detector. The event detector, as described in Chapter 2, extracts two signature values, a starting time and a time duration for each identified event.

The input to the clustering algorithm is only a signature value. The cluster algorithm runs separately on the two signature values as input. We define:

$$\Delta Sign = \begin{cases} \Delta\tilde{S} & \text{if we use the power signature} \\ \Delta i(t) & \text{if we use the current signature} \end{cases} \tag{3.7}$$

We create two different dataset $D$ with respect to the chosen signature $\Delta Sign$. The dataset contains all the signature values $\Delta Sign$ of each extracted event.

The output of the cluster algorithm is a set of clusters. Each cluster represents a device changing status. For example if we have a lamp switching on and off the expected result is two different clusters. One cluster is composed by all the event values extracted when the lamp switches on. The other cluster is composed by the events extracted during the switching off. The extracted events that are not representative of these two states are supposed to be discarded.

The two signature values give two different output results. The current signature is more sensitive to variations due to operating status of the devices but on the other hand is more accurate as it also considers the shape of the waveform.

## 3.3 Device model

The second step of the classification procedure aims at identifying the devices that are present in the environment.

There are three classes of appliance models: ON/OFF, Finite State Machine (FSM) and continuously variable. The ON/OFF model considers the devices as composed only by two possible states: ON or OFF. The FSM model considers the presence of other discrete states and discrete transitions. The continuously variable model considers the devices as composed by an infinite number of states.

In our analysis we are interested in the simplest model: ON/OFF. Fig. 3.5 shows an example of an ON/OFF model. The two states represent the device ON and OFF states. The transition from the OFF to the ON state is caused by a positive change in power $\Delta\tilde{S}$. This transition identifies an element in a cluster representing the device switching ON. The transition from the

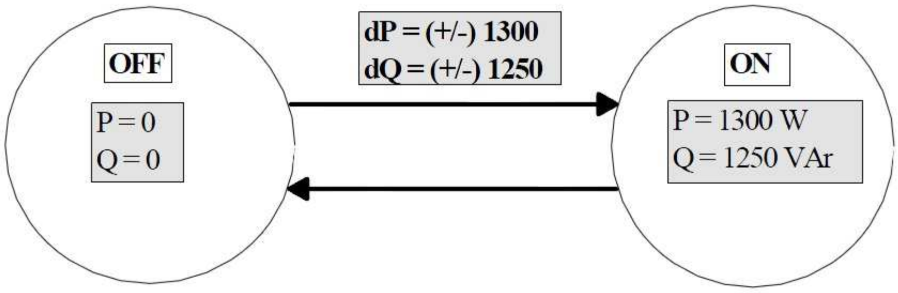



**Figure 3.5:** *Device ON/OFF model*

ON to the OFF cluster is caused by a negative change in power $-\Delta\tilde{S}$ with opposite value with respect to the OFF/ON transition. This value identifies an element in a cluster representing the device switching OFF. We want to couple together the two clusters representing the same device that is switching on and off. This procedure is done by coupling together clusters that have opposite mean value.

## 3.4 Identifying complex devices

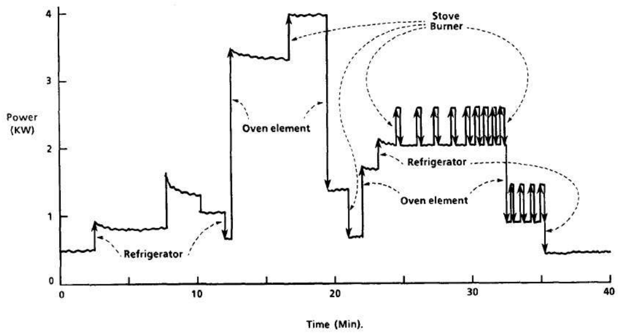



**Figure 3.6:** *Example of devices with a periodic behaviour during their operating status*

The last part of this analysis focuses on recognizing periodic patterns in the extracted events. Some devices, e.g. stove burner in fig. 3.6, have some cycles during their operating status. The detection of this periodic behaviours allows us to identify the presence of these particular devices. Another periodic pattern is due to complex device, e.g. some lamps in fig. 3.7, which change status as a sequence of events. The event detector identifies a sequence of different events but it cannot relate these events to the same appliance. The detection of these frequent sequences

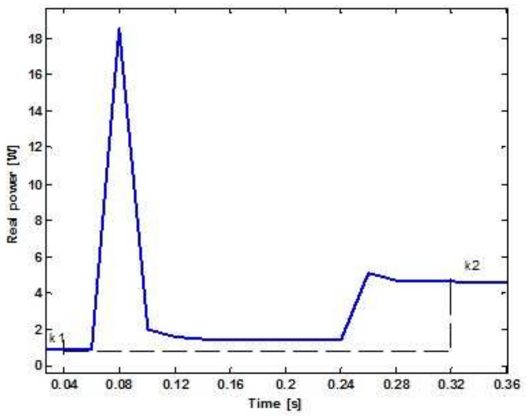



**Figure 3.7:** *Example of devices, lamp with nominal power 4W, with a complex on transient: detected as two steps*

allows us to describe more complex types of device changing status.

The idea is to detect these periodic behaviours as a sequence of events that happen at a certain time distance between them.

The input data (events), where we look for periodic patterns, is a sequence of extracted signatures at different time instants. There are two problems in the detection of the periodic patterns. The first problem is the time distance between events. Different periodic sequences have different time distances between their events. We cannot look for frequent sequences of signature values but we have to look for frequent sequences of signatures at the same time distances between them. The second problem is the length of the periodic sequence. The length of the sequence, which is the number of events that compose it, is not known a priori. We do not know how many events compose a sequence.

### 3.4.1 Time-Clustering

The idea to identify these periodic sequences is to use the same cluster algorithm adding information about the time distance between the events. This allows us to group together events with similar signature that happen at a similar time distance.

The second problem to solve is the length of the periodic sequence. We initially suppose to know the length of the periodic sequence. For example we suppose that the length of the sequence is equal to two. We want to find frequent sequences of two events at a certain time distance.

We call $S_d$ the dimension of the event signature. $S_d$ is equal to two for the normalised power signature and it is equal to 200 for the current waveform signature.

The cluster algorithm as explained before looks for clusters in the $S_d$ dimensional space. It can identify frequent signature values but it does not have any information about frequent sequences. To identify sequences of length two we look for clusters in a space of dimension: $Dim_2 = 2 * S_d + 1$ . The elements in this space are the signatures of two events and the time distance between them. The dense regions in this space represent frequent sequences of two events at a certain time distance. Adding the time information in the input space where we look for dense regions allows us to determine frequent sequences of events of different time distances. For example we can identify two different clusters representing two different sequences which have the same signature values but different time distances between them.

This procedure identifies all the sequences of length two. The procedure to identify frequent sequences of three events is the same as described before. The difference is the input space where we look for dense regions. The input space has dimension $Dim_3 = 3 * S_d + 2$ : three event signatures and two time differences between the events. The first time difference is the time distance between the first and the second event, the second time difference is the one between the second and the third event. The clusters found represent frequent sequences of three events.

Generally, the input space to identify a sequence of length $l$ has dimension:

$$Dim_l = l * S_d + (l - 1). \tag{3.8}$$

The clusters found represent frequent sequences of $l$ events.

The procedure is shown in figure 3.8. Each level $l$ represents the detected sequences of length $l$.

The second problem of identifying a sequence is that we do not know a priori from how many events is composed.

The idea to identify the length of a frequent sequence is to apply the cluster algorithm to successive level until no cluster is found. If there is no cluster we know that the cluster found at the previous level was the longest most frequent sequence.

The final algorithm starts looking for sequences of length one (only signature values without time information). Then sequences of length two are identified and so on until no cluster is found. The algorithm stops the maximum cluster layer $l_{max}$ when new clusters are not identified.

### 3.4.2 Algorithm

**Algorithm 3** Sequence-l

$r \leftarrow 1$
**for** $r = 1 \to M - l + 1$ **do**
  **for** $t = 1 \to l$ **do**
    $Y(r, t) = \Delta Sign(r + t - 1)$
  **end for**
  **for** $t = 1 \to l - 1$ **do**
    $Y(r, t + l) = K_{temp}(n_{start}(r + t) - n_{start}(r + t - 1))$
  **end for**
**end for**

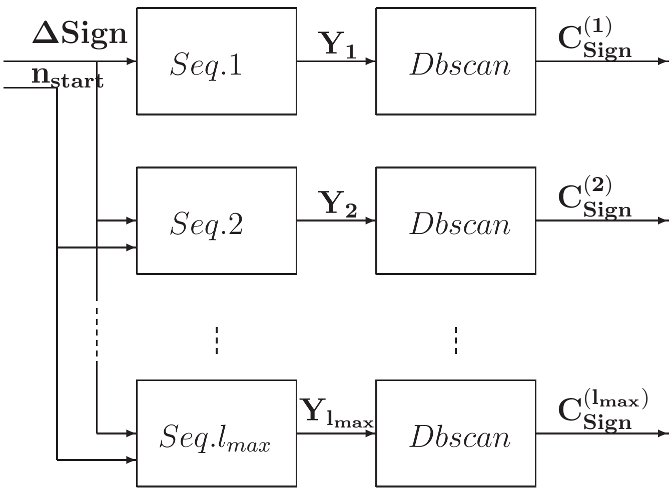



**Figure 3.8:** *Time-Clustering*

In this section the algorithm is explained. The implementation of the algorithm looks only to sequence of consecutive events. The procedure to identify sequences of events, also no consecutive, is the same but it requires an higher computational time to compare more sequences.

The vector $\mathbf{\Delta Sign}$ and $\mathbf{n_{start}}$, in Fig. 3.8, are the inputs to the time clustering algorithm. The vector $\mathbf{n_{start}}$ contains the values of the starting time of the events and its dimension is $M$ where $M$ is the total number of extracted events. The vector $\mathbf{\Delta Sign}$ contains the event signature value. It can be the power $\Delta\tilde{S}$ or the current waveform $\Delta i(t)$signature of the detected event.

$$\Delta Sign = \begin{cases} \Delta\tilde{S} & \text{if we use the power signature} \\ \Delta i(t) & \text{if we use the current signature} \end{cases} \tag{3.9}$$

The dimension $S_d$ of the signature vector is equal to 2 for the power signature $\Delta\tilde{S}$ and 200 for the current signature $\Delta i$.

The first part creates the input to the cluster algorithm. Each level corresponds to a different sequence length of events. For example the $l$ level corresponds to a sequence of $l$ consecutive events. Each row of $Y_l$ is an element of the dataset of level $l$, a sequence of $l$ events. The block which creates the sequence is explained in Algorithm 3. The first layer is the case without time described in section 3.2.3. The vector $\mathbf{Y_1}$ contains only the event values $\mathbf{\Delta Sign}$:

$$Y_1(r) = \Delta Sign(r). \tag{3.10}$$

$\mathbf{Y_1}$ has dimension $(M, S_d)$ The output of the first layer $\mathbf{C}^{(1)}_{Sign}(r)$ is equal to the cluster assigned to the element $Y_1(r)$.

The second layer input is the matrix $\mathbf{Y_2}$ where the $r$ element, $r$-line is the vector:

$$Y_2(r, 1 : 2 * S_d + 1) = (\Delta Sign(r), \Delta Sign(r + 1), K_{temp}(n_{start}(r + 1) - n_{start}(r)). \tag{3.11}$$

We introduce a parameter $K_{temp}$ to normalize the time values to the corresponding signature value. $\mathbf{Y_2}$ has dimension $(M-1, 2*S_d+1)$. The output of the second layer $\mathbf{C}^{(2)}_{Sign}(r)$ is equal to the cluster assigned to the $r$-element $Y_2(r,:)$.

The $l$-layer input is the matrix $\mathbf{Y_l}$ where the $r$ element, $r$-line is the vector:

$$Y_l(r, 1 : l*S_d+l-1) = (\Delta Sign(r), .., \Delta Sign(r+l-1), K_{temp}(n_{start}(r+1) - n_{start}(r)), ..., K_{temp}(n_{start}(r+l-1) - n_{start}(r+l-2))). \tag{3.12}$$

$\mathbf{Y_l}$ has dimension $(M-l+1, l*S_d+l-1)$. The output of the $l$-layer $\mathbf{C}^{(l)}_{Sign}(r)$ is equal to the cluster assigned to the $r$-element, $Y_l(r,:)$, which is the sequence beginning with $\Delta Sign(r)$ of length $l$. The sequence is composed by $l$ consecutive events:

$$\Delta Sign(r), ..., \Delta Sign(r+l-1) \tag{3.13}$$

The $l_{max}$-layer is the maximum layer. The $l_{max}$ input is the matrix $\mathbf{Y_{l_{max}}}$ where the $r$ element, $r$-line is the vector

$$Y_{l_{max}}(r, 1 : l_{max}*S_d+l_{max}-1) = (\Delta Sign(r), .., \Delta Sign(r+l_{max}-1), K_{temp}(n_{start}(r+1) - n_{start}(r)), .., K_{temp}(n_{start}(r+l_{max}-1) - n_{start}(r+l_{max}-2))). \tag{3.14}$$

$\mathbf{Y_{l_{max}}}$ has dimension $(M-l_{max}+1, l_{max}*S_d+l_{max}-1)$. No cluster is identified in this layer so the algorithm stops.

This procedure identifies all the frequent sequences of events of different lengths. The algorithm stops when we find the longest sequences. All the periodic patterns are identified.

## 3.5 Conclusions

The cluster algorithm identifies the devices that are present in the environment. It is divided into two parts. The first part uses only the signature information. The second part adds information about time to look for periodic patterns to identify more complex devices. The output of the algorithm is a set of clusters. The clusters representing the same device are coupled together. The detected devices are assigned to different groups based on their extracted electrical characteristics.

# Chapter 4

# System Performance

This chapter describes the implementation of the system model. The chapter is organised into four parts. The first part describes the demonstrator and the devices we use to test the system. The second part shows some examples which validate the model and the implementation of the algorithm in the demonstrator. The third part estimates the parameters of the model. Finally, in the fourth part, the algorithm is applied in a real environment.

## 4.1 Demonstrator

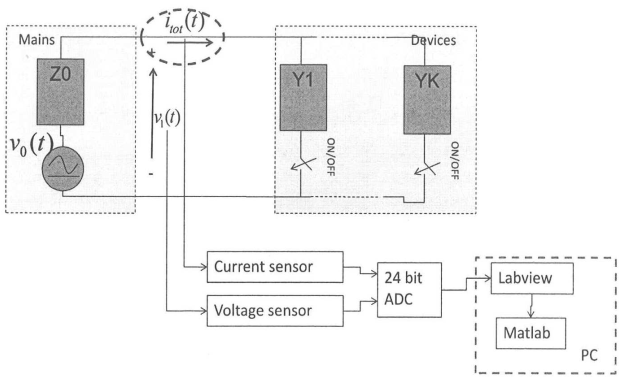



**Figure 4.1:** *Architecture system*

The set-up is fed by mains that behave as a single distribution channel that delivers electricity to the house. A differential voltage sensor is used for the demonstrator. The voltage sensor is located just after the current sensor. The voltage sensor maps the voltage into 0-5 volts range to ensure safe input to the AD converter. We use an Agilent current probe to sense the electric current. The current probe operates at 0.1 V/Amps configuration and it is linear within a high dynamic range from milli-Amperes to tens of Amperes. The voltage and the current values are fed into a 24 bits national Instrument AD converter that interfaces with LabVIEW. The 24 bits

ADC with 10 Volts range provides a granularity of approximately 0.6 $\mu V$. With the 0.1 V/A division range of current probe and 1/200 division range of voltage amplifier, 0.6 $\mu V$ granularity projects to 6 $\mu A$ for current and 0.12 mV for voltage measurement. We made these choices to be able to distinguish also very small loads ($\sim$ 1W) and to discard the effects of quantization on the performance of the algorithms. The AD converter runs at 10 KHz sampling rate.

LabVIEW takes the data from the ADC and shares them with Matlab where we implemented the algorithms. The data is provided in batches of 1k samples. This makes the communication process faster compared to single data transfer. One thousand samples correspond to 5 periods of current signature, i.e. a period has 200 samples, as the sampling frequency and line frequency are 10 kHz and 50 Hz, respectively.

**Figure 4.2:** *Demonstrator*

Fig. 4.2 and fig. 4.3 show the demonstrator.

### 4.1.1 Devices

The devices used in the experiment are listed in table 4.1. Some devices have more than one status, e.g. hair dryer and vacuum cleaner have two different ON status. In our model we consider only devices characterised by two status: ON and OFF. We model the second ON status as a different device. For example we have one hair dryer with two different stages of respective nominal power 720 W and 1400 W. We consider to have two different hair dryers the first one (Device 8 in tab.4.1) with nominal power 720 W and the second one (Device 9 in tab. 4.1) with nominal power 1400 W. Table 4.1 reports the nominal power of the devices. Fig. 4.4 shows some devices we used in the experiments. The second signature we consider is the current waveform.

In figures 4.5 and 4.6, we report the current waveforms of the devices. These values give

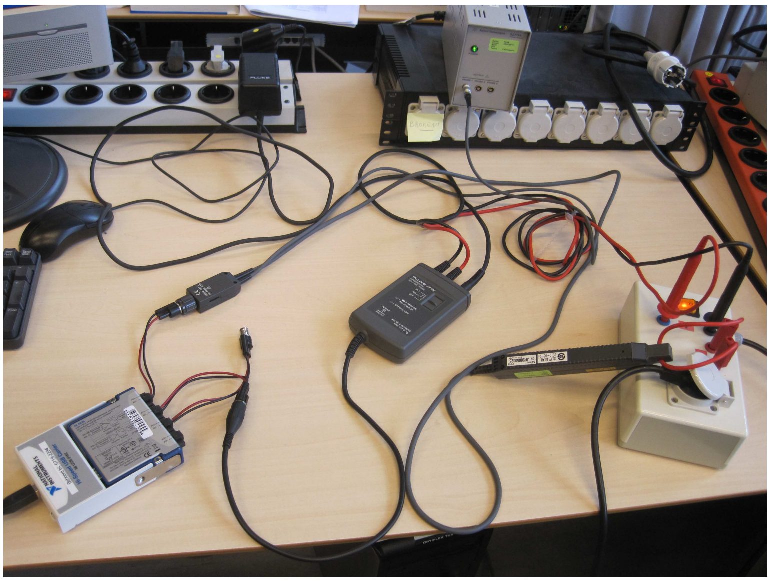


**Figure 4.3:** *Electrical connections in the demonstrator*

| Device number | Device name | Nominal power |
|---|---|---|
| Device 1 | Led Lamp | 4W |
| Device 2 | CFL | 5W |
| Device 3 | Lamp | 14W |
| Device 4 | CFL | 20W |
| Device 5 | Incandescent Lamp | 40W |
| Device 6 | Lamp | 400W |
| Device 7 | Water cooker | 2400W |
| Device 8 | Hair dryer | 720W (stage1) |
| Device 9 | Hair dryer | 1400W (stage2) |
| Device 10 | Vacuum cleaner | 400W (minimum) |
| Device 11 | Vacuum cleaner | 1250W(maximum) |

**Table 4.1:** *Device used in the experiments*



**Figure 4.4:** *Some devices used in the experiments*

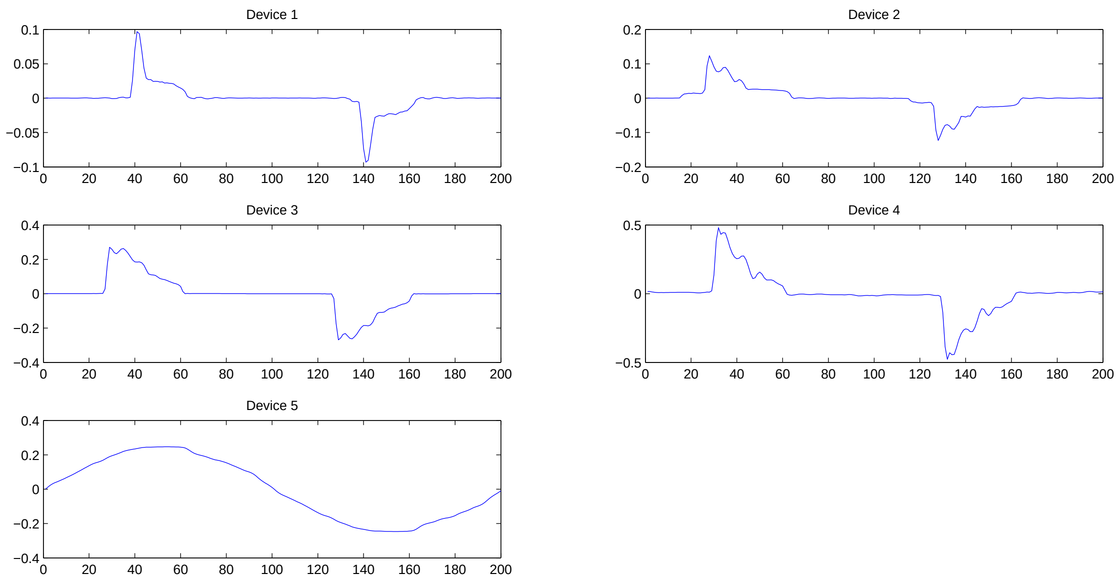




**Figure 4.5:** *Current waveforms of the devices 1-5. The x-axis represent the time in samples. 200 samples correspond to one period of 0.02s. The y-axis represents the current amplitude in Amperes*

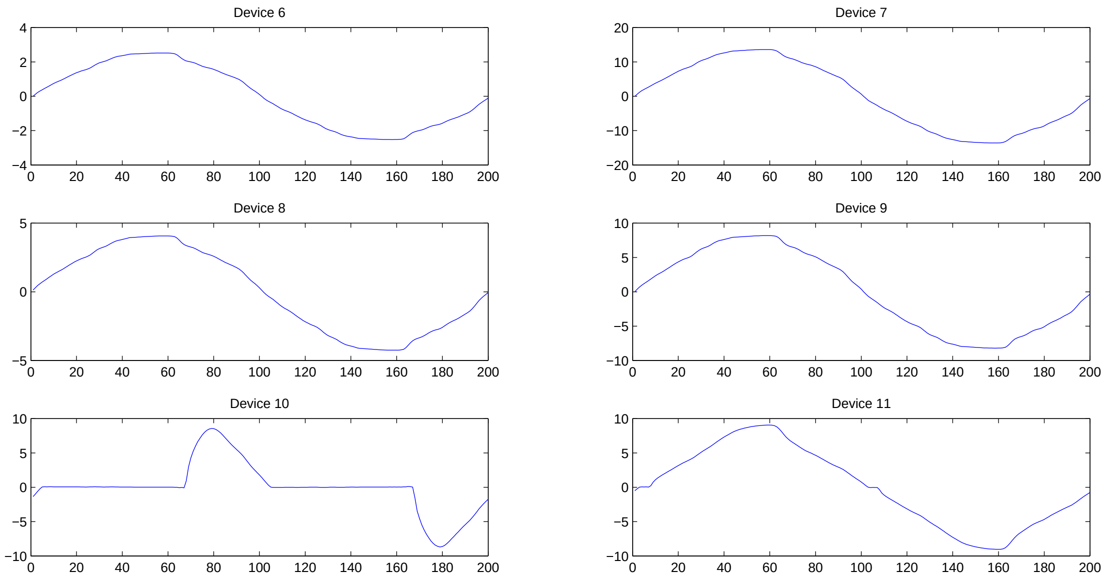
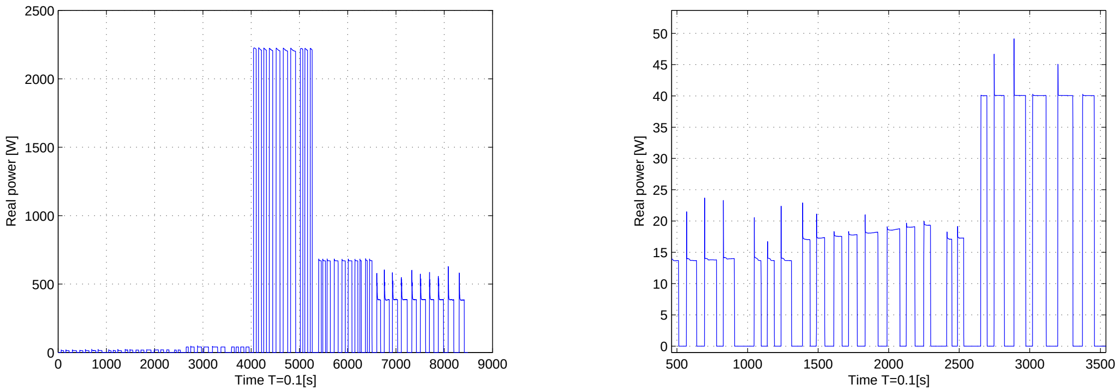



**Figure 4.6:** *Current waveforms of the devices 6-12. The x-axis represent the time in samples. 200 samples correspond to one period of 0.02s. The y-axis represents the current amplitude in Amperes*

the testing database we used. We suppose not to know it and the final goal of the system is to identify the appliances reconstructing this database. The event detector has to extract these signatures and the cluster analysis has to identify the corresponding devices states. The device are identified coupling the respective ON/OFF clusters.

## 4.2 System demonstrator

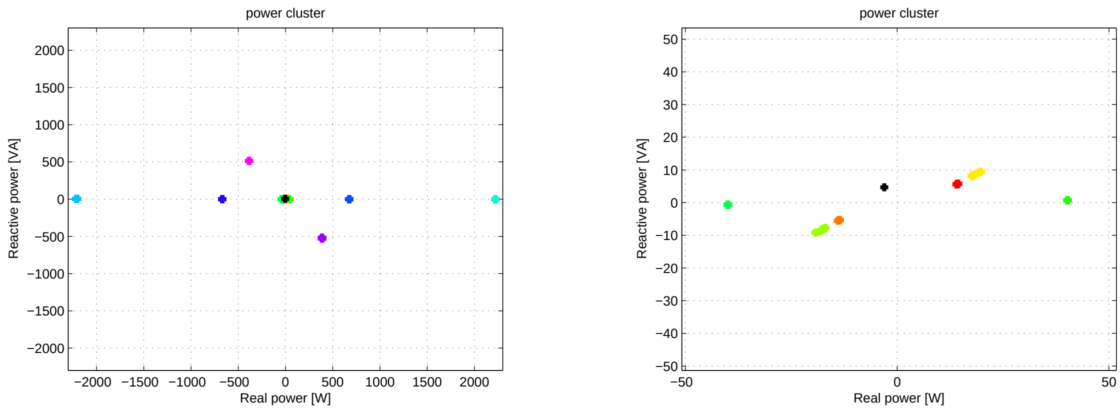
We report an example of the system.In this example we only show the output of the cluster algorithm for the first layer. We do not look for frequent sequences of device. We run an experiment on six devices. The six devices are the devices number 3, 4, 5, 7, 8, 10 in table 4.1. The sampling rate for the normalised power is $T = 0.1[s]$. We switch on and off each device separately from the others ten times each one. The parameters we use in the event detector are: $N = 1$ and




(a) Plot (b) Zoom first plot

**Figure 4.7:** *Real normalised power for six devices switched on ten times each one. x-axis* $T = 0.1[s]$, *y-axis* $\tilde{P}(n)$

$K = 2$. The expected number of detected events is $N_{real} = 120$. The number of detected events is $N_{det} = 121$. The cluster algorithm is run separately using the two different signatures. The parameters of the cluster algorithm are: $MinPts = 3$ and $\beta = 0.01$. Firstly, the cluster algorithm is applied using the power signature. The parameters we use are: $R = 0.8$ and $\alpha = 0.06$. The results of the cluster algorithm using the power signature are shown in Fig. 4.8. These pictures represent the signature power space. It is a two-dimensional space where the x-axis is the real normalised power signature $\Delta\tilde{P}$ and the y-axis is the reactive normalised power signature $\Delta\tilde{Q}$. The first picture shows the cluster result values. The second picture is the zoom around zero to show clusters of small devices. The dots in the picture represent the signature value of the detected event. The color of the dot identify to which cluster the event belongs to. Events with the same color belongs to the same cluster while different colors represent different clusters. Black dots are outliers. The results of the cluster algorithm are reported in Table 4.2. The total number of events in the clusters is $N_{clust} = 120$. One event is discarded as outlier. The identified




(a) Cluster result (b) Zoom clusters

**Figure 4.8:** *Output first layer cluster $C_S^{(1)}$ using power signature for six devices input*

| No. Cluster | Device detected (status) | Mean value [W] | No. of elements |
|---|---|---|---|
| 1 | Device 3 (Lamp ON) | $14 + 5i$ | 10 |
| 2 | Device 3 (Lamp OFF) | $-14 - 6i$ | 10 |
| 3 | Device 4 (CFL ON) | $18 + 8i$ | 10 |
| 4 | Device 4 (CFL OFF) | $-18 - 9i$ | 10 |
| 5 | Device 5 (Incandescent lamp ON) | $40$ | 10 |
| 6 | Device 5 (Incandescent lamp OFF) | $-40 - i$ | 10 |
| 7 | Device 7 (Water cooker ON) | $2223 - 4i$ | 10 |
| 8 | Device 7 (Water cooker OFF) | $-2209 + 2i$ | 10 |
| 9 | Device 8 (Hair dryer ON) | $675 - 2i$ | 10 |
| 10 | Device 8 (Hair dryer OFF) | $-671$ | 10 |
| 11 | Device 10 (Vacuum cleaner(min) ON) | $387 - 527i$ | 10 |
| 12 | Device 10 (Vacuum cleaner(max) OFF) | $-386 + 512i$ | 10 |

**Table 4.2:** *Output values first layer cluster algorithm $C_S^{(1)}$ using power signature*

clusters do not represent a device. They represent a device state. In our model we only consider device characterised by two states: ON and OFF. It is possible to couple together clusters which have opposite mean values that represent the same device switching ON and OFF. The identified

| Device identified | No. cluster ON event | No. cluster OFF event |
|---|---|---|
| Lamp (Device 3) | 1 | 2 |
| CFL lamp (Device 4) | 3 | 4 |
| Incandescent lamp (Device 5) | 5 | 6 |
| Water cooker (Device 7) | 7 | 8 |
| Hair dryer (Device 8) | 9 | 10 |
| Vacuum cleaner (Device 10) | 11 | 12 |

**Table 4.3:** *Identified devices after coupling ON and OFF clusters using power signature*

devices are listed in table 4.3. Each device is characterised by two clusters. The two clusters represent the ON and OFF events. This is the output of the first layer using the power signature. The cluster algorithm is run separately using the current waveform signature. The parameters we choose are: $R = 0.28$ and $\alpha = 0.06$. The mean of the current waveform we extract for each

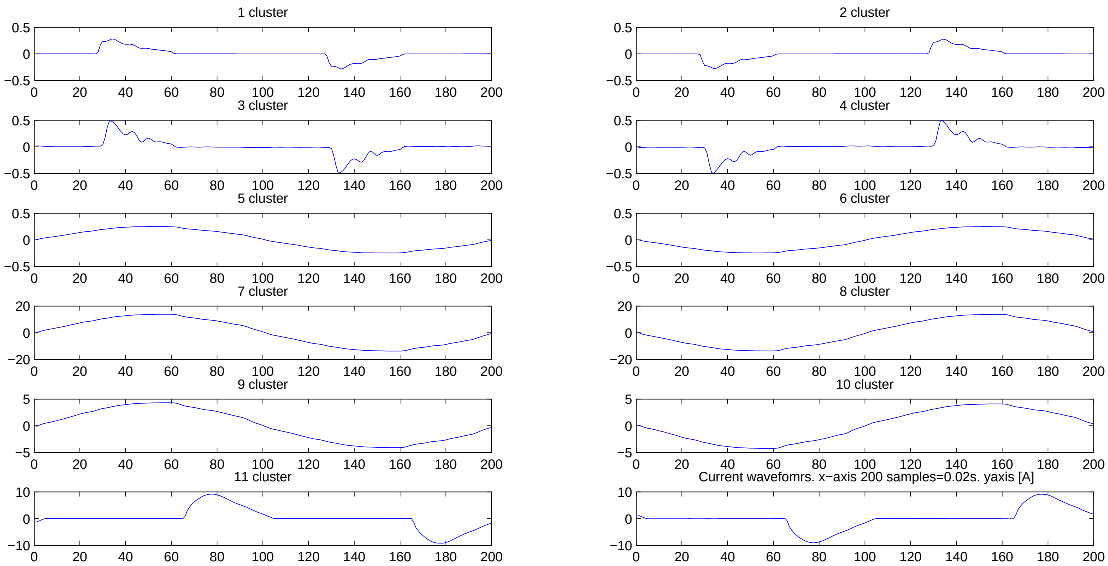



**Figure 4.9:** *Mean value current waveform for each cluster. x-axis 200 samples corresponds to one period of 0.02[s]. y-axis: current amplitude [A]*

cluster is reported in fig. 4.9. Table 4.4 reports the identified clusters using current signature. We can couple together clusters which present opposite waveforms to identify clusters representing the ON and OFF state of the same device. Table 4.5 reports the identified devices after the

| No. Cluster | Device detected (status) | $I_{RMS}$ Mean value [A] | No. of elements |
|---|---|---|---|
| 1 | Device 3 (Lamp ON) | 0.094 | 10 |
| 2 | Device 3 (Lamp OFF) | 0.0913 | 10 |
| 3 | Device 4 (CFL lamp ON) | 0.1337 | 10 |
| 4 | Device 4 (CFL lamp OFF) | 0.131 | 10 |
| 5 | Device 5 (Incandescent lamp ON) | 0.1748 | 10 |
| 6 | Device 5 (Incandescent lamp OFF) | 0.1739 | 10 |
| 7 | Device 7 (Water cooker ON) | 9.6676 | 10 |
| 8 | Device 7 (Water cooker OFF) | 9.6029 | 10 |
| 9 | Device 8 (Hair dryer ON) | 2.9385 | 10 |
| 10 | Device 8 (Led lamp OFF) | 2.9144 | 10 |
| 11 | Device 10 (Vacuum cleaner ON) | 3.8045 | 10 |
| 12 | Device 10 (Vacuum cleaner OFF) | 3.7529 | 10 |

**Table 4.4:** *Output values first layer cluster algorithm $C_i^{(1)}$ using current signature*

| Device identified | No. cluster ON event | No. cluster OFF event |
|---|---|---|
| Lamp (Device 3) | 1 | 2 |
| CFL lamp (Device 4) | 3 | 4 |
| Incandescent lamp (Device 5) | 5 | 6 |
| Water cooker (Device 7) | 7 | 8 |
| Hair dryer (Device 8) | 9 | 10 |
| Vacuum cleaner (Device 10) | 11 | 12 |

**Table 4.5:** *Identified devices after coupling ON and OFF clusters using current signature*

coupling of the ON and OFF clusters. This example shows the self learning algorithm results in our environment. We can identify all the devices using both signatures: power and current waveform. In the next subsections we analyse the difference between the two signatures and the higher layers of the cluster algorithm.

### 4.2.1 Detection of devices using current waveform

The example in this section shows the difference between the power and current signatures. The current signature has more information than the power. Using the current signature it is possible to separate devices which have a close value in power but different current waveforms. In this example we switch ON and OFF separately ten times each one two devices. The devices are two lamps with respective nominal power of 4W and 5W, devices 1 and 2 in table 4.1. These devices have a very close value in power but their current waveforms do not have a sinusoidal shape. The difference in the current shape allows us to identify the two devices using the current signature. Figure 4.10 shows the plot of the real normalised power. The total number of expected

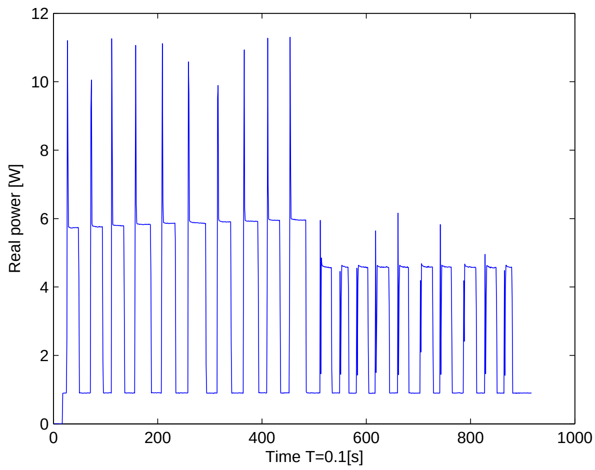



**Figure 4.10:** *Real normalised power device 2 (lamp 5W) and device 1 (lamp 4W) switched ON and OFF ten times each one separately. x-axis $T = 0.1[s]$, y-axis $\tilde{P}(n)$*

events is $N_{real} = 40$. The number of detected events is $N_{det} = 43$. We detect for three times over ten the device 2 (5W lamp) switching on as a sequence of 2 events, in total 6 events. We are not interested now in the detection of sequence of events which is analysed in subsection 4.2.3. We want to consider these six detected events as outliers. So the total number of detected events $N_{det} = 37 + 6$ where 37 is the number of events we want to classify and 6 are outliers. The expected result is shown in table 4.6. Firstly we run the cluster algorithm using the power signature. The output result is shown in fig. 4.11 and table 4.7. The output of the cluster algorithm using the power signature identifies only two clusters. The first cluster represents the switching off event and it is composed by 23 elements. This cluster is the union of the OFF

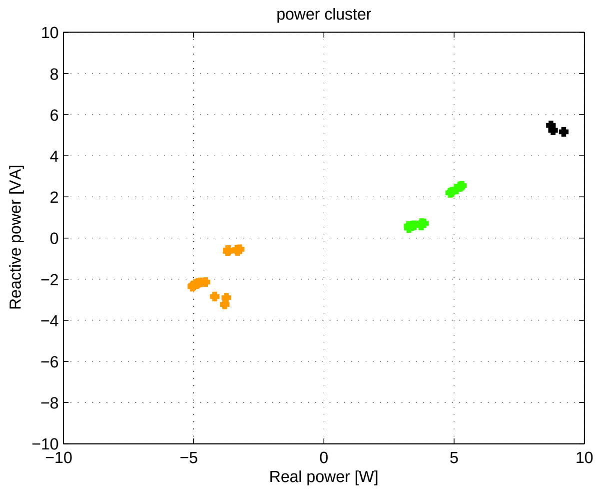



**Figure 4.11:** *Output values first layer cluster algorithm $C_S^{(1)}$ using power signature*

| No. Cluster | Device status | Nominal power [W] | No. of elements |
|---|---|---|---|
| 1 | Device 1 (on) | 4 | 10 |
| 2 | Device 1 (off) | -4 | 10 |
| 3 | Device 2 (on) | 5 | 7 |
| 4 | Device 2 (off) | -5 | 10 |

**Table 4.6:** *Expected output result*

| No. Cluster | Device detected (status) | Mean value [W] | No. of elements |
|---|---|---|---|
| 1 | Device 1-2 (off) | $-5-2i$ | 23 |
| 2 | Device 1-2 (on) | $4+i$ | 17 |

**Table 4.7:** *Output values first layer cluster algorithm $C_S^{(1)}$ using power signature*

events of the two devices and three outliers. The second cluster represents the switching ON and it is composed by 17 elements. This cluster is the union of the ON events of the two devices. Only three outliers over six are labelled as outliers. The cluster result using the power signature can only identify two clusters representing the ON/OFF states of a single device. We apply

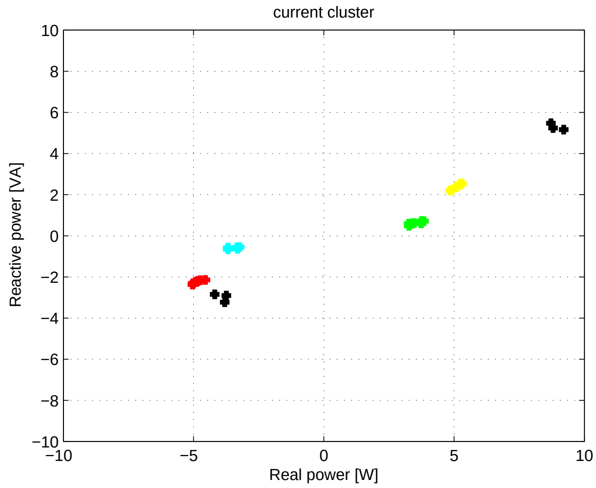
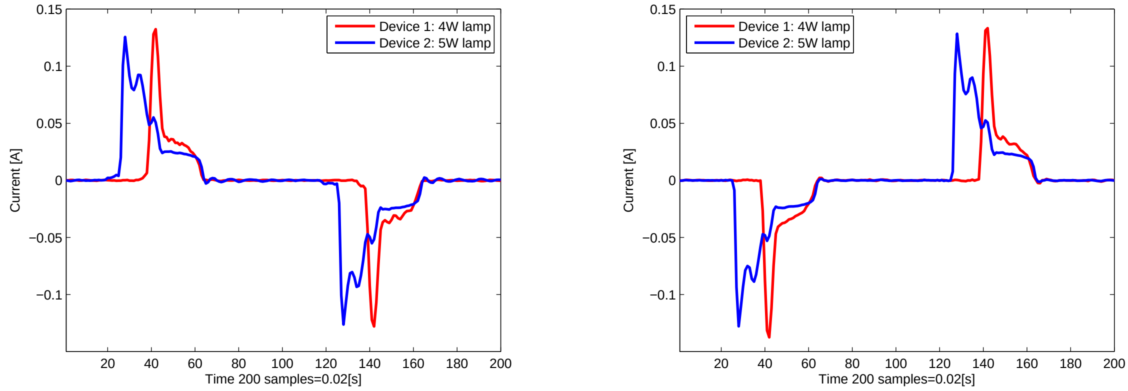



**Figure 4.12:** *Cluster output using current device 1 and 2*

| No. Cluster | Device detected (status) | $I_{RMS}$ [A] | No. of elements |
|---|---|---|---|
| 1 | Device 2 (off) | 0.034 | 10 |
| 2 | Device 2 (on) | 0.0349 | 7 |
| 3 | Device 1 (off) | 0.0278 | 10 |
| 4 | Device 1 (on) | 0.0284 | 10 |

**Table 4.8:** *Output values first layer clustering $C_i^{(1)}$ using current signature for 4W and 5W lamps*

the cluster algorithm using the current signature. The result is shown in fig. 4.12 and table 4.8. Using the current signature we can identify the clusters of the expected result, tab. 4.6. The cluster representing the OFF event using the power signature (cluster No.1 in tab. 4.7) is split into three clusters: device 1 (4W lamp) OFF (cluster No.3 in tab. 4.8), device 2 (5W lamp) OFF (cluster No.1 in tab. 4.8) and three outliers. The cluster representing the on event using the power signature (cluster No. 2 in tab. 4.7) is split into two clusters: device 1 (4 W lamp) ON (cluster No. 4 in tab. 4.8) and device 2 (5W lamp) ON (cluster No.2 in tab. 4.8). The difference in the shape of the two current waveforms allows us to separate the events due to different devices using the current signature. The power signature does not give enough information to divide them. The difference between the current waveforms is shown in fig. 4.13. The current signature also allows the separation of the three outliers that are close in power to the off events. The current signature is useful when we want to separate devices which have a close value in power but different current waveforms.




(a) Current waveform on events (b) Current waveform off events

**Figure 4.13:** *Current waveforms Device 1 and 2: lamp 4 and 5 W*

### 4.2.2 Example of devices working together

In the previous examples the data are collected switching ON and OFF each device separately from the others. One device is switched ON when all the other devices are OFF. In this section we want to identify a device even if there are other devices ON at the same moment. The example we choose is the detection of the device 5 (40 W incandescent lamp) in presence of other devices working at the same time. The chosen devices are the CFL lamp (device 4), the lamp (device 6), the hair dryer (device 8) and the water cooker (device 7). Figure 4.14 shows the plot of the

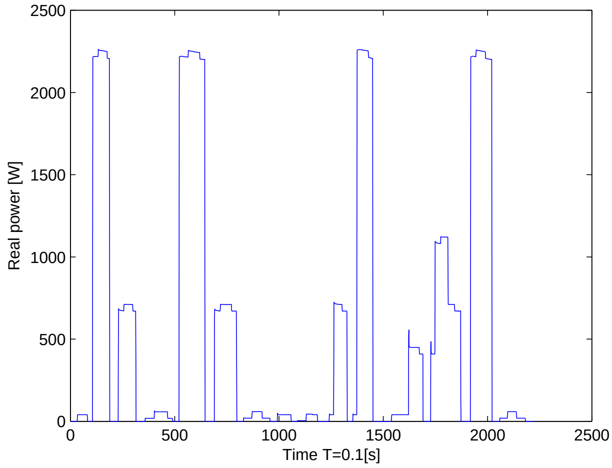
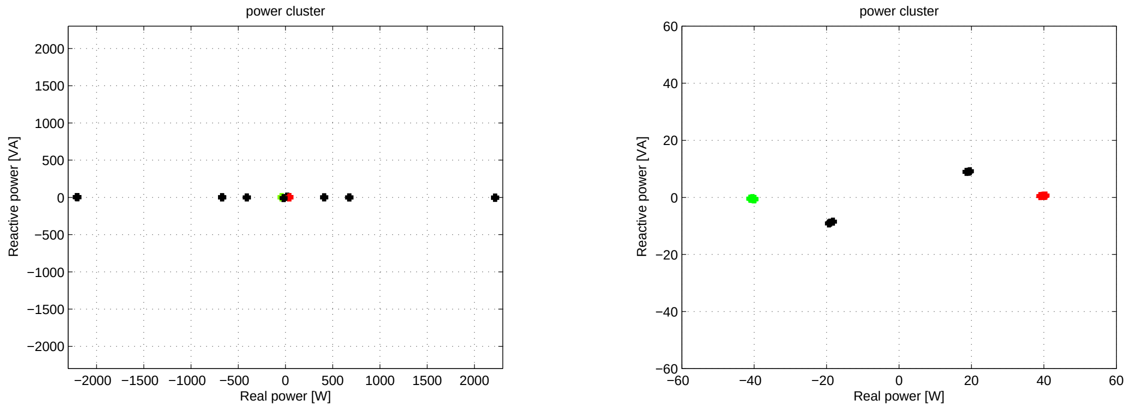



**Figure 4.14:** *Real normalised power of device 5 (lamp 40W) and other devices. x-axis* $T = 0.1[s]$, *y-axis* $\tilde{P}(n)$

real normalised power. The lamp is switched ON and OFF 15 times with different backgrounds.

The cluster result using the power signature is shown in fig. 4.15. The black dots represent the




(a) Cluster result (b) Zoom clusters

**Figure 4.15:** *Output first layer cluster $C_S^{(1)}$ using power signature for device 5 (40W lamp)*

outliers. The devices used to create a different background are not switched ON and OFF enough times to create their own new cluster. Table 4.9 reports the values of the detected clusters. The

| No. Cluster | Device detected (status) | Mean value [W] | No. of elements |
|---|---|---|---|
| 1 | Device 5 (Incandescent lamp on) | 39 | 15 |
| 2 | Device 5 (Incandescent lamp off) | -41-i | 15 |

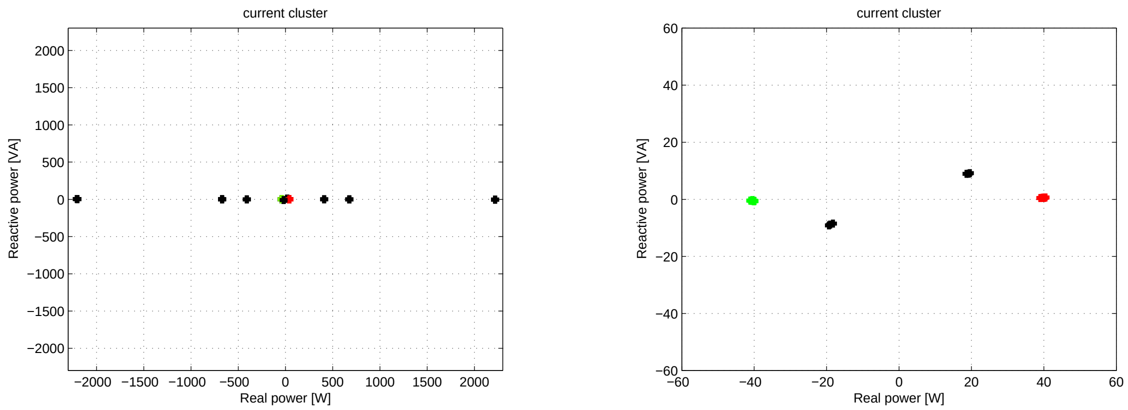
**Table 4.9:** *Output values first layer clustering $C_i^{(1)}$ using current signature for device 5 (40W lamp)*

two clusters represent the switching ON and the switching OFF of the lamp. The result using the current signature are shown in fig. 4.16. The values of the clusters are listed in table 4.10.

| No. Cluster | Device detected (status) | $I_{RMS}$ Mean value [A] | No. of elements |
|---|---|---|---|
| 1 | Device 5 (Incandescent lamp On) | 0.1736 | 15 |
| 2 | Device 5 (Incandescent lamp Off) | 0.1755 | 15 |

**Table 4.10:** *Output values first layer clustering $C_i^{(1)}$ using current signature for device 5 (40W lamp)*

Figure 4.17 shows the current waveform of the two identified clusters. This example shows that we can detect and identify a device even if we are in presence of bigger loads working at the same moment. The device is detectable using both power or current signature.




(a) Cluster result (b) Zoom clusters

**Figure 4.16:** *Output first layer cluster $C_i^{(1)}$ using current signature for device 5 (40W lamp)*

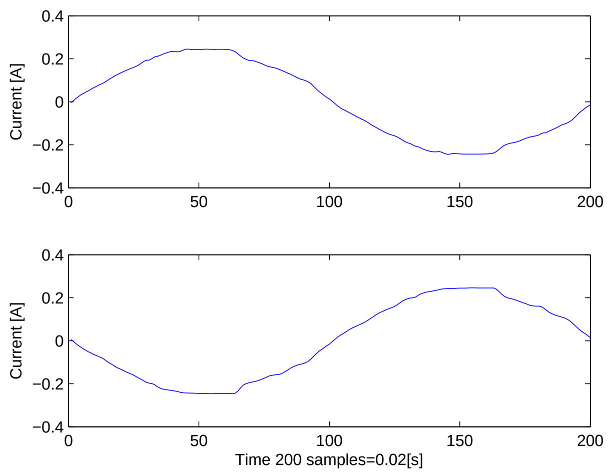



**Figure 4.17:** *Current waveforms clusters device 5 (40W lamp)*

### 4.2.3 Cluster of sequence of events

The successive layers of the cluster analysis look for periodic patterns, frequent sequences of events. In this section we show an example of a frequent sequence caused by a device switching ON which is detected as a sequence of two events. In the previous example, section 4.2.1, the device 2 (5W lamp) sometimes switches ON as a sequence of two events. If we apply the cluster algorithm we can not identify these sequences as a unique device changing status. In this section we show an example of how the second layer of the cluster algorithm solves this problem. We

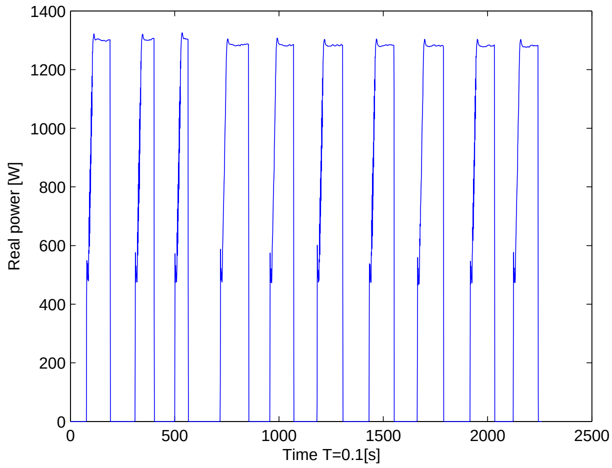
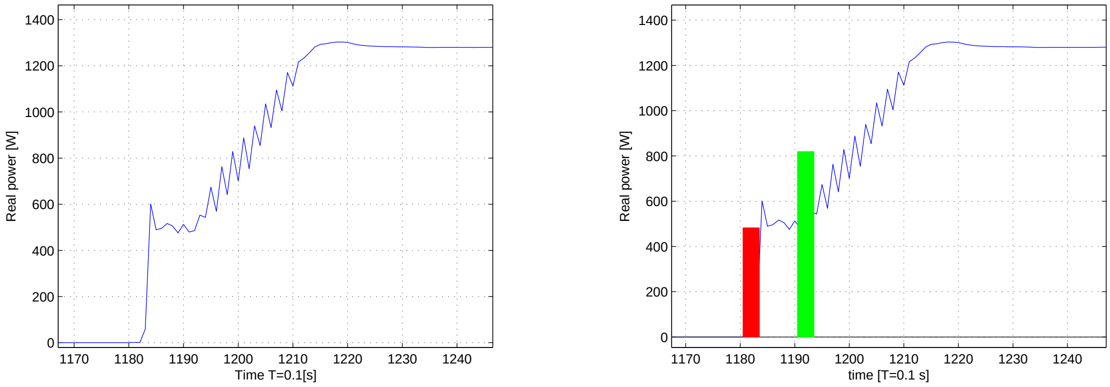



**Figure 4.18:** *Real normalised power of device 11 (Vacuum cleaner maximum power). x-axis* $T = 0.1[s]$*, y-axis* $\tilde{P}(n)$

choose to analyse the device 11 in table 4.1, the vacuum cleaner, when is switched ON and OFF using the maximum power level. We switch ON and OFF the vacuum cleaner ten times. No other devices are used in this experiment. The normalised real power is shown in fig. 4.18. In fig. 4.19 the zoom of the switching on of the vacuum cleaner is shown. The first picture shows the real power during the vacuum cleaner switching ON. The vacuum cleaner switches ON slowly and there is a stable period during the switching ON. The second picture shows the output of the event detector. The event detector identifies two different jumps. The jumps are represented with different colors. The color is related to the corresponding cluster in the output of the first layer without using time information. In fig. 4.20 and table 4.11 the output of the first layer of the cluster algorithm using power signature is shown. The cluster algorithm identifies three clusters. Two clusters represent the switching ON and one cluster represents the switching OFF. As shown in fig. 4.19 the switching ON of the vacuum cleaner is detected as a sequence of two events. The first event belongs to the first cluster (red) and the second to the second cluster (green). There is no relation between the two events. We want to identify them as belonging to the same device. We apply the second layer of the cluster algorithm adding time information. We look for frequent sequence of two events at the same time distance. In tables 4.12 and 4.13 the output of the cluster algorithm second layer is shown. The output of the second layer identifies




(a) Switching on vacuum cleaner (b) Detection events

**Figure 4.19:** *Zoom plot of the vacuum cleaner switching ON and output of the event detector. x-axis* $T = 0.1[s]$*, y-axis* $\tilde{P}(n)$

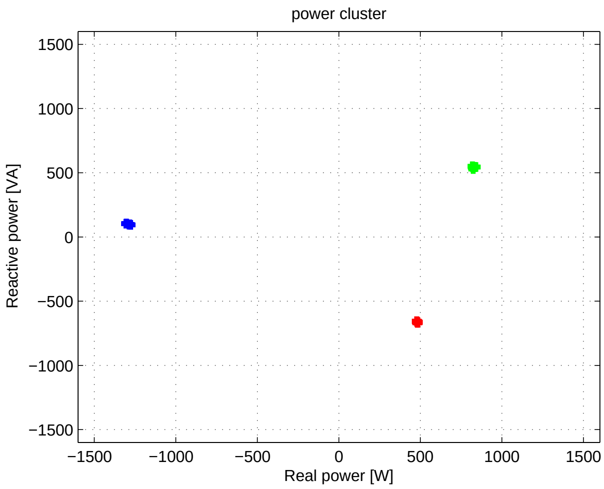



**Figure 4.20:** *Output first layer cluster* $C_S^{(1)}$ *using power signature for device 11 (Vacuum cleaner max)*

| No. Cluster | Device detected (status) | Mean value [W] | No. of elements |
|---|---|---|---|
| 1 | Device 11 (Vacuum on-partial) | $480 - 663i$ | 10 |
| 2 | Device 11 (Vacuum on-partial) | $826 + 543i$ | 10 |
| 3 | Device 11 (Vacuum off) | $-1290 + 98i$ | 10 |

**Table 4.11:** *Output values first layer clustering* $C_S^{(1)}$ *using current signature for device 11 (Vacuum cleaner max)*

| No. Cluster | Device detected (status) | No. of elements |
|---|---|---|
| 1 | Device 11 (Vacuum on-partial) | 10 |

**Table 4.12:** *Output second layer cluster $C_S^{(2)}$ using power signature for device 11 (Vacuum cleaner max)*

| No. Cluster | Mean value [W] first | Mean value [W]second | Time difference [s] |
|---|---|---|---|
| 1 | 480-663i | 826+543i | 1 |

**Table 4.13:** *Output values second layer clustering $C_S^{(2)}$ using power signature for device 11 (Vacuum cleaner max)*

the presence of one frequent sequence of two events at a time distance of one second between them. The number of element is ten like the number of switching ON. This result shows that all the events representing a sequence of switching ON are coupled together. We do not know a priori by how many elements is composed a frequent sequence. The cluster algorithm is applied then to sequences of three events. No cluster is found at this layer. When no cluster is found the algorithm stops. The second layer is the maximum layer where we can find sequence of events. All the switching on are identified and detected as belonging to the same device. The same result is obtained if we use the current signature adding time information. In table 4.14 and fig. 4.21

| No. Cluster | Device detected (status) | $I_{RMS}$ [A] | No. of elements |
|---|---|---|---|
| 1 | Device 11 (Vacuum on-partial) | 4.7411 | 10 |
| 2 | Device 11 (Vacuum on-partial) | 5.6253 | 10 |
| 3 | Device 11 (Vacuum off) | 5.6586 | 10 |

**Table 4.14:** *Output values first layer clustering $C_i^{(1)}$ using current signature for device 11 (Vacuum cleaner max)*

the results of the first layer of the cluster algorithm using the current signature are shown. As expected we find three different cluster. Two cluster represent the two steps of the ON event and one cluster represents the OFF event. We apply the second layer of the cluster algorithm looking for frequent sequences of two events. The output of the second layer identifies a sequences of two events which represent the switching ON of the vacuum cleaner. The results are listed in tables 4.15 and 4.16. The sequence of events caused by the vacuum cleaner switching ON is identified all the times using the current signature too. This example shows that is possible to identify periodic patterns using power or current signature.

## 4.3 Parameter estimation

This section estimates the parameters of the model. Firstly the metric we used are presented. Then the metric are used to estimate the parameters. This procedure is done for the event detector and for the cluster analysis separately.

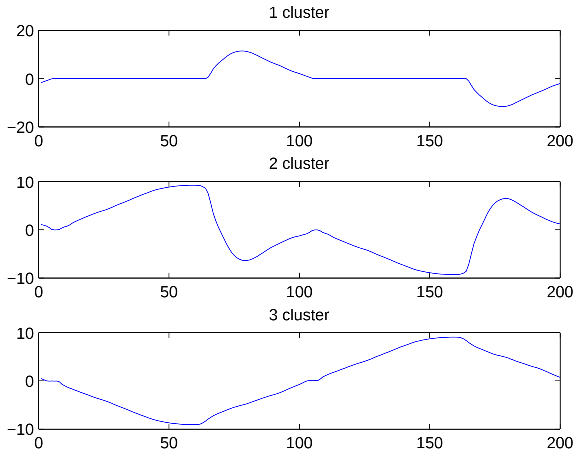



**Figure 4.21:** *Current waveforms of the three detected clusters using current signatures. x-axis time 200 samples=0.02 [s], y-axis current [A]*

| No. Cluster | Device detected (status) | No. of elements |
|---|---|---|
| 1 | Device 11 (Vacuum on-partial) | 10 |

**Table 4.15:** *Output second layer cluster $C_i^{(2)}$ using current signature for device 11 (Vacuum cleaner max)*

| Cluster number | $I_{RMS}$ [A] first | $I_{RMS}$ [A]second | Time difference [s] |
|---|---|---|---|
| 1 | 4.7411 | 5.6253 | 1 |

**Table 4.16:** *Output values second layer clustering $C_i^{(2)}$ using current signature for device 11 (Vacuum cleaner max)*

### 4.3.1 Event detector

The two parameters, that characterise the event detector, are $N$ and $K$. The first parameter $N$ determines the filter length: $L_{h_N} = 2N$. This parameter is related to the stability of the original input signal. The second parameter $K$ is a threshold to determine when a time instant can be considered part of an event. Now we want to estimate the parameters in a real environment. As described in chapter 2, the event detector algorithm extracts from the original signal a certain number of events. Two signature values, a time instant representing the beginning of the event and a time duration are calculated for each event.

We consider the estimation as a problem of binary classification. Each element is labelled as positive (p) or negative (n). There are four possible outcomes from a binary classifier:

- TP (True Positive): if both the outcome from the prediction and the actual value are p
- FP (False Positive): if the predicted value is p however the actual value is n
- TN (True Negative): if the predicted value is n while the actual value is p
- FN (False Negative): if both the predicted value and the actual value are n

In our analysis we consider the two values: false positive (FP) and false negative (FN). An element is labelled as positive (p) if it is considered an event (detected or real). Otherwise it is labelled as negative (n).

The first group (False positive FP) represents an event that is detected by the event detector of parameters $(N, K)$ but is not a real event. It is not a real event means that there is a change in the power that we do not want to consider as a device changing state. This change can be caused by the operating state of a device so we do not want to detect it.

Each detected event has to be labelled as belonging to the first group (FP) or not. The detected event is labelled belonging to the group FP if there is not a real event happening during the same duration time. Otherwise it is labelled as not belonging to it.

The second group TN (True Negative) represents the real events that are not detected by the event detector. Each real event has to be labelled as belonging to the second group or not. It belongs to the second group (TN) if it is not identified by the event detector.

We define two ratios:

- True Positive Rate=$1 - \frac{TN}{N_{real}}$
- False Discovery Rate=$\frac{FP}{N_{det}}$

where $N_{real}$ is the total number of events and $N_{det}$ is the total number of detected events.

The first ratio determines the rate of identified events in the total number of real events.

The second ratio determines the rate of detected events that do not represent a real event in the total number of detected events.

These two ratios are used to estimate the parameters of our model.

#### 4.3.1.1 Experiment and results

The experiment to estimate the parameters of the event detector is characterized by six devices with different electrical behaviour. The chosen devices are devices number 1, 2, 4, 5, 7 and 8 in table 4.1. Each device is switched on and off ten times. The total number of expected events is $N_{det} = 120$. Each device is switched on and off separately from the others. A device is switched on only if all the other devices are OFF at that moment. The two parameters we have to determine are $(N, K)$. $N$ determines the filter length: $L_{h_N} = 2N$. $K$ is the threshold value.

| N | K | FP | TN | $N_{det}$ |
|---|---|---|---|---|
| 1 | 1 | 38 | 0 | 179 |
| 1 | 2 | 5 | 2 | 144 |
| 1 | 5 | 0 | 102 | 104 |
| 2 | 1 | 35 | 120 | 175 |
| 2 | 2 | 6 | 120 | 146 |
| 2 | 5 | 3 | 102 | 107 |
| 3 | 1 | 30 | 120 | 170 |
| 3 | 2 | 11 | 120 | 151 |
| 3 | 5 | 7 | 103 | 120 |

**Table 4.17:** *Numerical results*

| N | K | False discovery 'rate' | True positive 'rate' |
|---|---|---|---|
| 1 | 1 | 0.21 | 1 |
| 1 | 2 | 0.0347 | 0.98 |
| 1 | 5 | 0 | 0.85 |
| 2 | 1 | 0.2 | 1 |
| 2 | 2 | 0.04 | 1 |
| 2 | 5 | 0.028 | 0.85 |
| 3 | 1 | 0.18 | 1 |
| 3 | 2 | 0.07 | 1 |
| 3 | 5 | 0.058 | 0.86 |

**Table 4.18:** *Results rate*

Table 4.17 shows the output values of false positive and true negative events. Table 4.18 contains the ratios of False Discovery and True Positive. The chosen values are $N = 2$ and $K = 2$. We choose these values because they are the ones which give the best True Positive Rate with a low enough value of the False Discovery Rate.

### 4.3.2 Classifier

The evaluation measures that characterized the cluster validity are classified into two different groups [15]:

- *Unsupervised*. These measures utilise only internal information. The clusters are evaluated looking at the data that characterise a cluster. For this reason these measures are called internal indices. There are two large classes: measures of cluster cohesion, which determine how similar are the objects inside a cluster, and measures of cluster separation, which determine how different are the elements from different clusters.

- *Supervised*. These measures utilise external information. The data points are labelled as belonging to a given category. These measures estimate how well the clustering output is similar to the assigned category structure.

In our analysis we use a supervised measure. There are several types of measures belonging to this class. We consider the ones based on the mapping of each item between its cluster and category.

Being $C$ the set of clusters to be evaluated, $L$ the set of categories and $M$ the number of items, the precision of a cluster $C_n$ for a given category $L_m$ is:

$$\Pi(C_n, L_m) = \frac{|C_n \cap L_m|}{|C_n|} \tag{4.1}$$

The precision, $\Pi$, of a cluster related to a given category expresses the fraction of the cluster that consists of the elements of the given category. The expected value is equal to one for only one category and zero for all the others categories. This implies that the cluster is composed only by elements of one category.

A similar measure is the recall. The recall for a cluster $C_n$ for a given category $L_m$ is:

$$Recall(C_n, L_m) = \frac{|C_n \cap L_m|}{|L_m|} \tag{4.2}$$

The recall of a cluster expresses how many elements of the given category are present in that cluster. The expected value is one for one category and zero for all the others. That implies that all the elements of the category (that assumes a value equal to one) are present in that cluster.

We give two more general definitions to compare the cluster results. The first is called purity and uses the precision values defined in (4.1). The purity value is calculated taking the weighted average of the maximal precision values:

$$Purity = \sum_n \frac{|C_n|}{M} \max_m \Pi(C_n, L_m) \tag{4.3}$$

In the same way we can define the inverse purity. We use the recall value defined in (4.2). The inverse purity looks at the cluster with maximum recall for a given category.

$$InversePurity = \sum_m \frac{|L_m|}{M} \max_n Recall(C_n, L_m) \tag{4.4}$$

These two measures are used to evaluate the parameters of the cluster.

We fix the two values of $MinPts = 3$ and $\beta = 0.01$ and estimate the two parameters: radius $R$ and normalization factor $\alpha$.

In the cluster algorithm we can use two different data input: the normalised power step or the current waveform. The parameters $R$ and $\alpha$ will assume different values in these two different cases.

#### 4.3.2.1 Experiment and results

We run a similar experiment to estimate the parameters of the cluster algorithm. The chosen devices are devices number 3, 4, 5, 7, 8 and 10 in table 4.1. Each device is switched on and off ten times separately from the others. We estimate the two parameters $R$ and $\alpha$ separately for the normalised power signature and for the current waveform signature.

| R | $\alpha$ | Purity | Inverse purity |
|---|---|---|---|
| 1 | 0 | 1 | 0.72 |
| 3 | 0 | 1 | 0.94 |
| 5 | 0 | 0.79 | 0.98 |
| 1 | 0.03 | 1 | 1 |
| 3 | 0.03 | 0.89 | 0.99 |
| 5 | 0.03 | 0.8 | 1 |
| 1 | 0.06 | 1 | 1 |
| 3 | 0.06 | 1 | 1 |
| 5 | 0.06 | 0.8 | 1 |

**Table 4.19:** *Results normalized power signature*

The best results are obtained for the couples $(R, \alpha)$: $(1, 0.03)$, $(1, 0.06)$ and $(3, 0.06)$. We choose the couple $(1, 0.06)$ because the single values of $R = 1$ and $\alpha = 0.06$ are the ones which give best results. This choice, a smaller value for $R$ and bigger for $\alpha$, allows us to create clusters of devices which have a big mean value, even if they are spread, while we can separate small devices with a close value in power because they are less affected by the value of $\alpha$.

The best results are obtained for the couples $(R, \alpha)$: $(0.5, 0.03)$ and $(0.5, 0.06)$ We choose the couple $(0.5, 0.06)$ for the same reason explained before.

## 4.4 Real environment

The previous examples, in section 4.2, show the validation of the system. The data are collected using the demonstrator, so we know which appliances are working in the environment. In this section we show an example where the algorithm is applied to real data. We do not know what we are looking for and we analyse the output of the system to identify the devices. The data are collected during one day on the lamps of a floor in the Philips offices.

| R | $\alpha$ | Purity | Inverse purity |
|---|---|---|---|
| 0.3 | 0 | 1 | 0.7 |
| 0.5 | 0 | 1 | 0.84 |
| 1 | 0 | 0.76 | 0.84 |
| 0.3 | 0.03 | 1 | 0.96 |
| 0.5 | 0.03 | 1 | 1 |
| 1 | 0.03 | 0.8 | 1 |
| 0.3 | 0.06 | 1 | 0.99 |
| 0.5 | 0.06 | 1 | 1 |
| 1 | 0.06 | 0.8 | 1 |

**Table 4.20:** *Results current waveform signature*

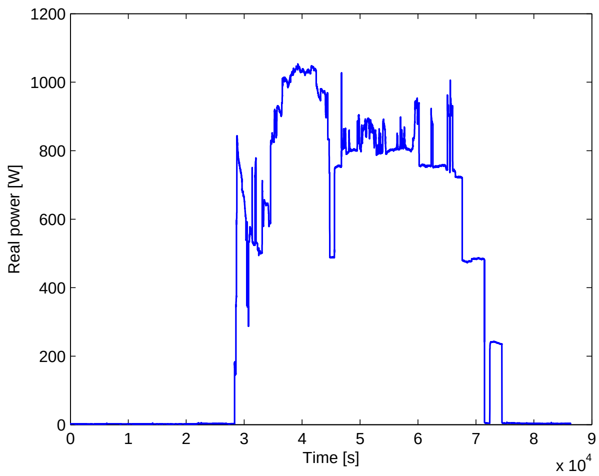



**Figure 4.22:** *Real normalised power lamps measured during one day. x-axis* $T = 1[s]$, *y-axis* $\tilde{P}(n)$

Fig. 4.22 shows the plot of the real power. The sampling period for the normalised power is higher: $T = 1[s]$. The algorithm is applied using the power signature.

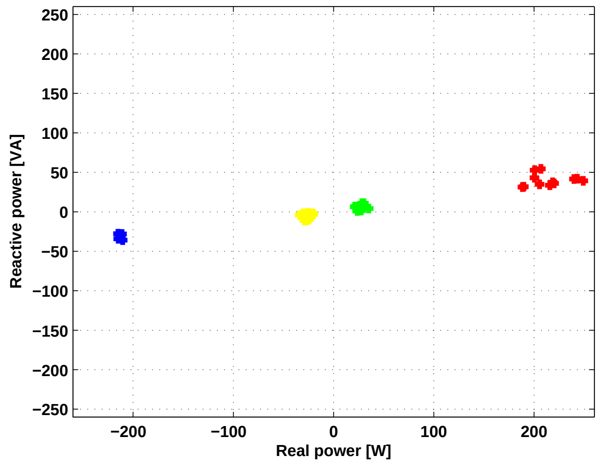



**Figure 4.23:** *Output first layer cluster $C_S^{(1)}$ using power signature*

| No. Cluster | Mean value [W] | No. of elements |
|---|---|---|
| 1 | $215 + 40i$ | 12 |
| 2 | $-27 - 7i$ | 19 |
| 3 | $26 + 5i$ | 10 |
| 4 | $-213 - 30i$ | 6 |

**Table 4.21:** *Output values first layer clustering $C_S^{(1)}$ using power signature*

In figure 4.23 and table 4.21 the results of the cluster algorithm are reported. We identify four clusters. Two opposite clusters with absolute mean values around 200W and two opposite clusters with absolute mean values around 30W. This first result shows the presence of two devices.

We apply the second layer of the cluster algorithm adding time information.

| No. Cluster | Mean value [W] first | Mean value [W] second | Time [s] | No. of elements |
|---|---|---|---|---|
| 1 | $-213 - 28i$ | $-27 - 10i$ | 9.75 | 4 |

**Table 4.22:** *Output values second layer clustering $C_S^{(2)}$ using power signature and time information*

Table 4.22 shows the output of the second layer. Fig. 4.24 is the total output of the first layer adding the new cluster identified in the second layer. The second layer identifies one cluster

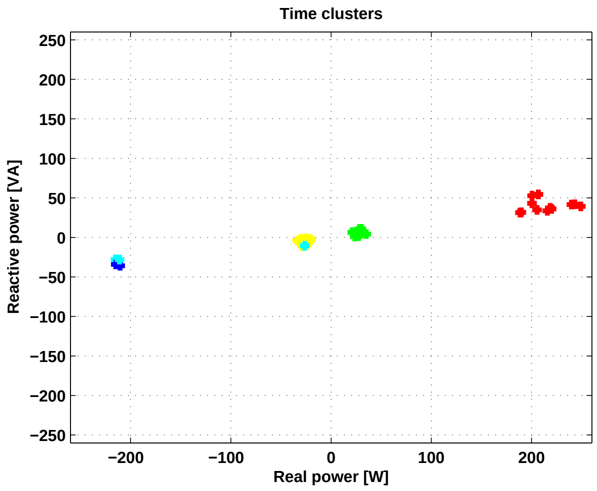



**Figure 4.24:** *Output second layer cluster $C_S^{(2)}$ using power signature and time information*

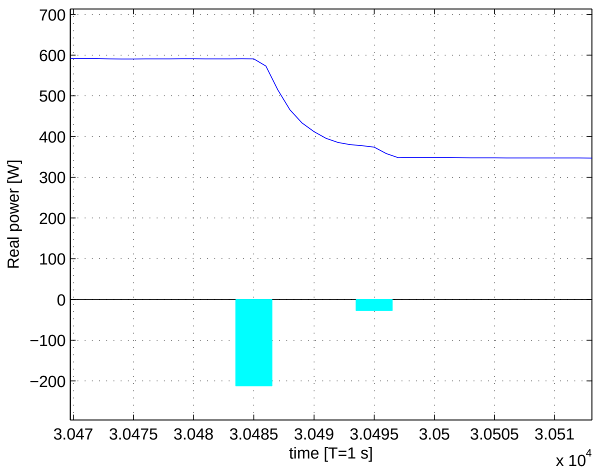



**Figure 4.25:** *Example lamp switching OFF, identified cluster 2 layer*

representing a lamp which switches off very slowly. The event detector identifies two different events during the switching off. The switching off is plotted in fig. 4.25. The third layer does not identify any cluster so the algorithm stops.

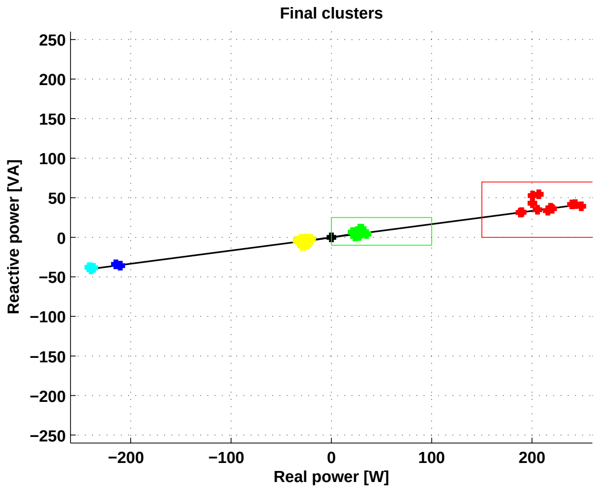



**Figure 4.26:** *Final output cluster algorithm: first and second layer*

| Device identified | Mean value ON event | Mean Value OFF event |
|---|---|---|
| Lamp (red) | $215 + 40i$ | $-212 - 35i$ |
| Lamp (red) | $215 + 40i$ | $-239 - 38i = (-212 - 28i) + (-27 - 10i)$ |
| Lamp (green) | $26.5 + 5i$ | $-26 - 6i$ |

**Table 4.23:** *Identified devices after coupling ON and OFF clusters*

In fig. 4.26 the final output is plotted. The new identified cluster is the sum of the two events that were identified as a sequence from the second layer of the cluster algorithm. The signature space has two different regions: the red one represents lamps of nominal power around 200 W, the green one represents smaller devices. The five final clusters are coupled together identifying three different types of devices. Two are lamps that have the ON event in the red cluster but they have different OFF events. One lamp switches off instantaneously while the other present a very long switching off (identified as two events). The third element is a smaller element belonging to the green region. The results are reported in table 4.23.

This result shows that we are able to identify the presence of different lamps using the power signature. The time information added to the cluster algorithm allows us to detect more complex devices, for example lamps switching off slowly, which are not detectable using only the first layer of the cluster algorithm.

## 4.5 Conclusions

This chapter shows the validation of the system model. We show the demonstrator and the implementation of the algorithm on it. We report some examples of the validation of the model using the data collected with the demonstrator. The algorithm is then applied to a real environment with satisfactory results.

# Conclusions

The non-intrusive appliance load monitoring system (NALM) provides information about the electrical consumption of the appliances measuring voltage and current in a single point. This system is preferable to the traditional approach because of the easy of installation, maintenance and low cost. In this thesis we design an Automatic Setup NALM system (AS-NALM). This system is less intrusive then the classic MS-NALM system. It learns by itself about information on the appliances that are present in the environment and their consumptions.

Firstly, an event detector was developed. It detects changes in the normalised power which exceed a given threshold. When a change is identified the event detector extracts two signatures to characterised the corresponding device. One signature is the normalised power step, the other is the current waveform difference.

Secondly, a cluster algorithm analysed the output of the event detector to find out frequent patterns which identify a device status. The cluster analysis is composed by different layers to find out frequent sequences of events. These sequences identify periodic patterns in the extracted events.

Finally a device model is created. The cluster representing different states (on and off) of the same device are coupled together. Then a name is given to the identified devices looking at their electric behaviour.

The system was tested in a demonstrator to validate the model we proposed. Then it was applied to a real environment where the number and characteristic of the devices was unknown. The final result shows that the system is able to identify the devices.

# Bibliography


[1] H.H. Chang, P.C. Chien, L.S. Lin, and N. Chen. Feature extraction of non-intrusive load-monitoring system using genetic algorithm in smart meters. In *IEEE International Conference and E-Business Engineering*, pages 299–304, 2011.

[2] H.H. Chang and C.L. Lin. A new method for load identification of nonintrusive energy management system in smart home. In *Proc. IEEE International Conference and E-Business Engineering*, pages 351–357, 2010.

[3] H.H. Chang, C.L. Lin, and H.T. Yang. Load recognition for different loads with the same real power and reactive power in non-intrusive load monitoring systems. In *Proc. 12th International Conference on Computer Supported Cooperative Work in Design*, 2008.

[4] A.I. Cole and A. Albicki. Data extraction for effective non-intrusive identification of residential power loads. In *Proc. Instrumentation and Measurements Technology Conference*, volume 2, pages 812–815, 1998.

[5] S. Drenker and A. Kader. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, 12(4):47–51, Oct. 1999.

[6] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, August 1996.

[7] A. Filippi, A. Pandharipande, A. Lelkens, R. Rietman, T. Schenk, Y. Wang, and P. Shrubsole. Multi-appliance power disaggregation an approach to energy monitoring. In *Proc. IEEE International, Energy Conference and Exhibition (EnergyCon)*, pages 91–95, 2010.

[8] G. W. Hart. Non-intrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, Dec. 1992.

[9] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power Energy Magazine*, 1(2):56–63, Mar/Apr 2003.

[10] W.K. Lee, G.S.K. Fung, H.Y. Lam, and Lucente M. Chan, F.H.Y. Exploration on load signatures. In *Proc. International Conference on Electrical Engineering*, 2004.

[11] S.L. Leeb, S.R. Shaw, and J.L. Kirtley. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Transactions on Power Delivery*, 10(3):1200–1210, July 1995.

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

[13] Y.H. Lin, M.S. Tsai, and C.S. Chen. Applications of fuzzy classification with fuzzy c-means clustering and optimization strategies for load identification in nilm system. In *IEEE International Conference on Fuzzy System*, pages 859–866, 2011.

[14] A. Marchiori, D. Hakkarinen, Han Qi, and L. Earle. Circuit-level load monitoring for household energy management. *IEEE Pervasive Computing*, 10(1):40–48, 2011.

[15] T. Pang-Ning, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.

[16] A.G. Ruzzelli, C. Nicolas, A. Schoofs, and G.M.P. O'Hare. Real-time recognition and profiling of appliances through a single electricity sensor. In *Proc. Seventh Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON10*, pages 279–287, 2010.

[17] S.R. Shaw, S.B. Leeb, L.K. Norford, and R.W. Cox. Nonintrusive load monitoring and diagnostics in power systems. *IEEE Transactions on Instrumentation and Measurement*, 57(7):1445–1454, July 2008.