NONINTRUSIVE LOAD MONITORING FOR VERIFICATION AND DIAGNOSTICS

BY

CHRISTOPHER E. REEG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Thomas J. Overbye

ABSTRACT

Nonintrusive load monitoring (NILM) is a method of detecting the current energy consumption of a building, using a single set of sensors on the main building supply. This approach is in contrast to intrusive monitoring where end-use devices are sensed. Building on techniques of previous works, it will be shown that NILM can be implemented on commercially available devices with capabilities similar to modern smart meters and can provide meaningful feedback to both the user and supplying utility. Limitations of inexpensive commercial devices, such as resolution and measurement sample rate, will be addressed. Using clustering and a Hidden Markov Model approach, data about the state of the devices in a building can be determined. This information can then be used to verify the effectiveness of smart-grid initiatives such as VAr control and demand-side management in addition to other energy-saving measures such as weatherproofing and installing energy-efficient appliances. In addition, this information can be used to select devices for distributed control by analyzing not only the type of device but also the real-world operating characteristics.

To my parents for their continuous support and my grandparents for encouraging me to pursue engineering

ACKNOWLEDGMENTS

This research was supported by the Department of Energy under award number DE-OE0000097 as part of the Trustworthy Cyber Infrastructure for the Power Grid (TCIPG) program.

I would also like to acknowledge Erik Johnson and Professor Doug Jones for collaborating on the digital signal processing side of this project. Erik provided expert guidance during key parts of the post-processing development.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AMI	Advanced Metering Infrastructure
DSM	Demand-Side Management
FSM	Finite State Machine
HMM	Hidden Markov Model
NILM	Nonintrusive Load Monitoring

CHAPTER 1

INTRODUCTION

In the age of the smart grid, information is king. Data are being collected at the generating station, transmission, distribution and consumption stages. One developing area is residential load monitoring and control. Through improvements in smart-meter technology, the usage in an individual home can be tracked with high resolution. While this advanced metering infrastructure (AMI) is currently used for simply billing the customer via remote meter measurements, this technology has the potential to have wide-ranging effects. Such opportunities include time-of-use pricing, remote service disconnect, and, most promising, nonintrusive load monitoring [1].

1.1 Nonintrusive Load Monitoring

Nonintrusive load monitoring (NILM) is the process of collecting data about the load inside a building without implementing a submeter infrastructure. This configuration is in contrast to an *intrusive load monitoring* scheme which requires sensors on all of the devices of interest. These sensors then communicate back to a central aggregation hub, which may be part of the residential smart meter.

Instead of having multiple sensors spread out on the appliances of interest, NILM uses only a single sensor suite located at the building service entrance. Complex algorithms are then used to disaggregate the measured usage into individual devices. This type of configuration offers increased hardware reliability due to decreasing the overall number of sensors and auxiliary hardware. In addition, higher-quality sensors can be utilized without significantly increasing the cost of the system. As usage is measured at the service entrance, this technology can also be integrated into the smart meter itself. The trade-off is that NILM requires much more signal processing and analysis than a distributed metering scheme would need, especially if the system is autonomously learning about the loads inside the home without human intervention.

Currently, commercial devices are being produced that can be installed by the homeowner in the main distribution panel [2], [3], [4]. These devices have similar capabilities to those of a typical smart meter[1] and can be used to test the effectiveness of various NILM algorithms in a real-world test environment. While these devices have several restrictions that are not encountered in lab-quality sensing measurement devices, they more accurately represent the potential deployment environment in an AMI.

1.2 Research Overview

In order to correctly implement a NILM system on a commercial device, several steps had to be completed. The first was to simply collect data to be analyzed. This collection process included utilizing a commercially available home-monitoring device and interfacing a data collection mechanism to it. As part of this process, an evaluation was done to test the ability of the specific device to accurately measure usage under a wide range of conditions. After data had been collected, it was preprocessed to remove artifacts and

[1]See the hardware chapter for details.

extract useful features from the data. An evaluation of different feature-extraction methods was done to identify the best way to obtain meaningful information from the measurement data. The identified features were then used to build device profiles which were in turn used to identify devices from the aggregate data measurement. Applications for this information were also investigated and will be discussed later.

CHAPTER 2

LITERATURE REVIEW

The field of nonintrusive load monitoring is not a recent development. Active development of NILM has been progressing since the early 1980s with preliminary work done by Kern and Brown in 1983. In response to energy-supply constrictions in the early 1970s, an interest in energy independence grew. The Electric Power Research Institute (EPRI) sponsored several research programs to study ways to lower peak electrical demand which was being supplied by inefficient gas generators. One of the first proposed solutions to this problem was created by Brown, who tried to identify the consumption of appliances by nonintrusive methods. The idea then, as now, was to identify devices that had a significant impact on peak load that could be used for control [5]. A more complete history of NILM development has been compiled by George Hart which includes many published works in the field up to 1995 [6].

Large advancements were made by George Hart in 1992 [7]. In this paper, Hart proposes a basic framework for monitoring primarily commercial appliances and other large building loads such as chillers and fans. Three load models are proposed that can be combined to form a complete load model. The first is a simple on/off model which may represent the behavior of a refrigerator, pump, or other two-state device. The second is a load that follows a specific set of operations in the form of a finite state machine (FSM). An example of this may be a dishwasher or clothes dryer. These state machines may be dynamic, as appliances may have different "options" that may be selectively enabled. These first two models are relatively easy to detect either by manual profile construction or by automatic detection. The last load model, however, is a continuously variable model which is much harder to detect, as a result of not presenting any sharp changes in power consumption. This model is applicable in a variable-speed drive on an air conditioner or several other modern residential appliances [8].

Norford and Leeb also describe the difficulties in applying NILM [9]. In addition to the variable-speed drives, modern advancement in power electronics has yielded power factor correction power supplies. In these power supplies, the reactive power of the device is internally compensated such that the device consumes no reactive power from the distribution network. As a result, reactive power can not be used to characterize a device, only the real power. This characteristic reduces the number of dimensions that can be used to identify a device and increases the difficulty of disaggregating the load. Another difficulty is nonsymmetric on/off changes in consumption. In a simple resistive device, such as a lightbulb, the change in power consumption when the device turns on will be symmetric to the change when the device turns off. This symmetry is not necessarily the case with motors which may have a large starting transient that slowly decays. When the motor finally turns off, the power usage will have settled back down to a steady-state value which will most likely be less than the turn-on value. This effect was observed in the case of a refrigerator which would routinely show this characteristic.

Hart also discusses two types of learning methods: manual and automatic, with the former being simple and relatively easy to implement and the latter being the topic of machine learning enthusiasts. In the manual method, individual device characteristics are input by the user to define the devices

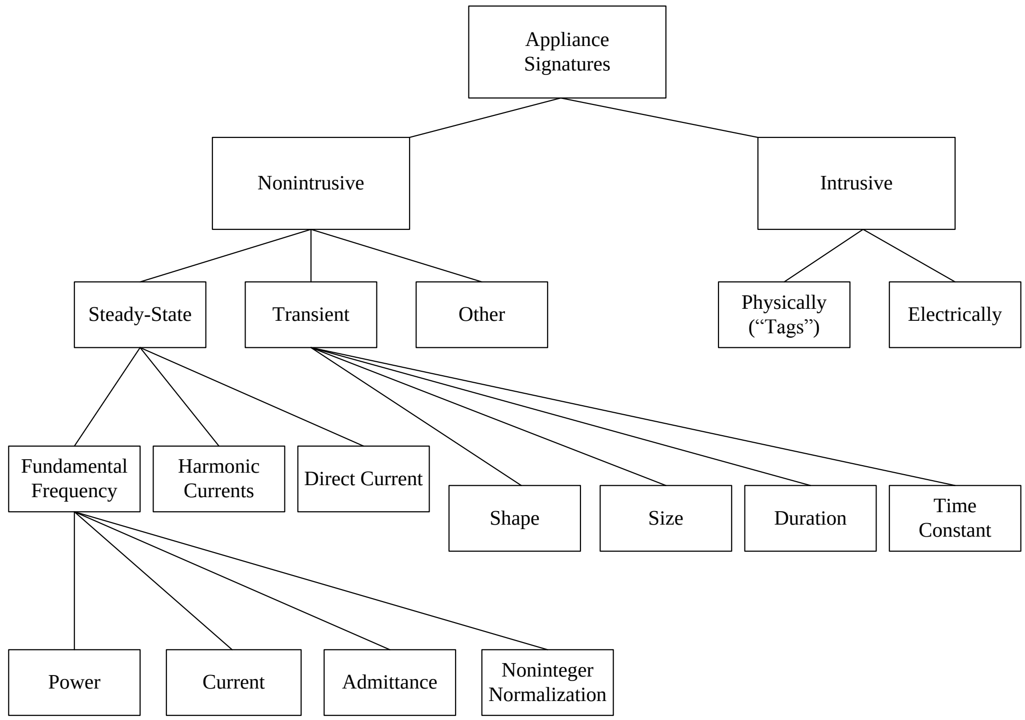


Figure 2.1: Signature Taxonomy

that the NILM system should look for. In an automatic system, the NILM system builds the device profiles as usage patterns and device behaviors are discovered. A hybrid option, which will be detailed in full later, uses automatic detection of features from devices that are manually separated from the aggregate load. In this way, the system can learn on its own what it should be looking for without involving the user in the number crunching.

The issue of which specific features to focus on in the NILM implementation is also investigated by Hart. In particular, features are divided into two groups, intrusive and nonintrusive. These two groups are further divided up into several other categories. The figure 2.1 shows the taxonomy of these different features.

While Hart takes the approach of observing changes in steady-state power consumption, Norford and Leeb take a transient pattern recognition approach. In the transient recognition case, line frequency measurements (60-

Hz sample rate) are taken and used to identify the turn-on characteristics of devices in the system. Sampling quickly also reduces the chance that two devices will have overlapping events inside the sample window. At 1-Hz sample rate, the chance that two devices will turn on at the same time is not trivial.

Another method which is used to detect devices uses harmonic analysis [10], [11], [12]. In this method, very high frequency measurements (several times line frequency) are taken in order to perform frequency analysis on the current waveforms. Using this technique, additional measurement dimensions can be added which increases the accuracy of event isolation. In particular, Laughman provides an example in which two devices, an incandescent lightbulb and a small computer, look similar in the Δ P/Q, 2D space but are significantly separated in the Δ 3rd Harmonic/P/Q, 3D space. It was also shown by Laughman that variable-speed drives can be isolated by observing that they produce harmonic currents in synchronization with the first harmonic.

Hart describes a method of learning finite state machine (FSM) load models using automatic methods. This method however assumes that every other load in the system is not changing so the individual states of a single device can be observed. In this case, each observation is treated as a transition of the FSM. Automatic learning of patterns is the holy grail of NILM but is very difficult due to a fundamental problem. In general, the NILM problem is underdetermined, i.e., there is not enough information in the raw measurement data to solve the problem. This underdetermination doesn't mean that the problem is intractable, but it limits the solution to probabilistic solutions instead of absolute deterministic solutions.

The overall algorithm, depicted in figure 2.2, which was originally proposed by Hart has not changed significantly over the past two decades.

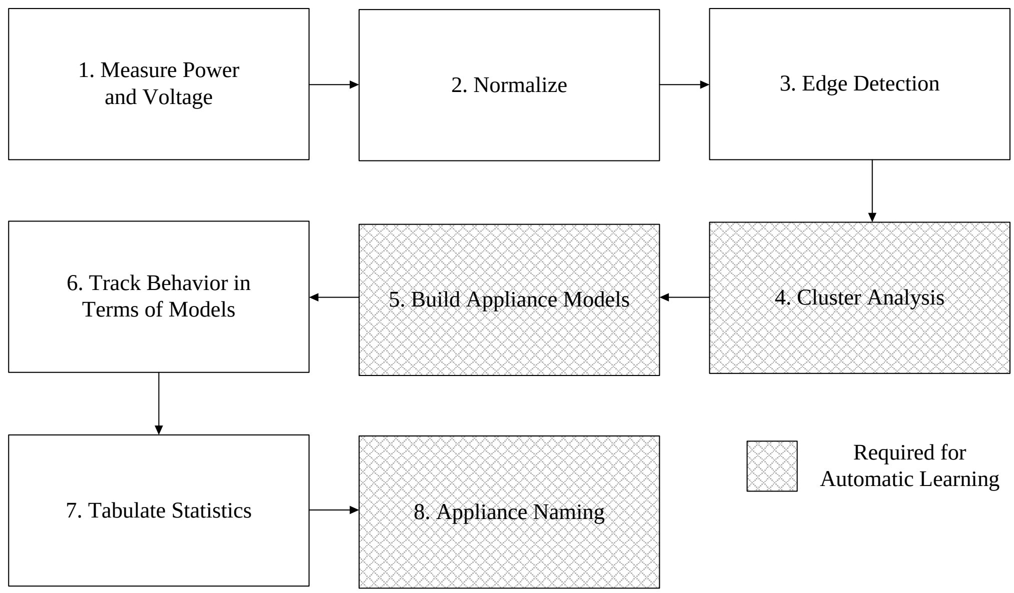


Figure 2.2: Original NILM Flowchart Proposed by Hart

Advances have been made in almost every step of this flowchart but the overall idea remains the same. Information flows down from the measured values, through preprocessing, feature detection, device profiling, tracking, and summary. Using this structure, several different algorithms can be applied to the measured data, such as the steady-state analysis done by Hart but also harmonic analysis and other methods detailed in figure 2.1.

The advantages primarily include increased hardware reliability, the general nonintrusive nature of monitoring device installation, and the ability to increase the number of devices sensed without increasing hardware requirements. In contrast, an intrusive load monitoring system requires several individual sensors which together decrease the overall reliability of the system. If another device is to be added to the system, additional hardware would be required rather than simply expanding the profile library as in a NILM system. Intrusive load monitoring systems will also require a communication back to a central data collection hub, which implies using additional wires or wireless bandwidth.

Table 2.1: List of Common Household Loads

Appliance	Power (watts)
Clock Radio	10
Coffeemaker	900-1200
Clothes Washer	350-500
Clothes Dyer	1800-5000
Dishwasher	1200-2400
Dehumidifier	785
Personal Computer	60 - 270
Laptop	50
Incandescent Lightbulb	40-150
Refrigerator	725
Television (27 inch)	113
Water Heater	4500-5500
Water Pump (Well)	250-1100

The disadvantages of NILM are also clear. As was discovered in field tests, NILM is not useful for all types of loads. Specifically, small loads such as home lighting, electronic loads such as entertainment systems or desktop computers, and loads that are always on. Fortunately, most loads of interest in the domestic environment are not below this "small" load threshold. This fact is particularly true in the application of NILM to demand-side load management which would not be interested in controlling small, insignificant loads but is more interested in controlling large appliances and motors. Table 2.1 shows a listing of common household loads as listed on the U.S. government Energy Savers website [13]. As can be seen from this chart, most of the loads of interest are above 200 watts and are therefore relatively easy to detect when the device turns on or off. As most of the processing work is done with algorithms and not direct measurement, the potential for error is also increased over an intrusive monitoring system.

CHAPTER 3

HARDWARE OVERVIEW

The first step in a nonintrusive load monitoring system is to collect the raw usage data from the power line. This task is primarily done using a current transformer or Hall-Effect sensor with an optional voltage probe. Using only the current probe, information about real and reactive power division is lost. When a voltage probe is also added to the sensor suite, the angle between the two waveforms can be obtained and used to calculate real and reactive power usage.

$$\text{Apparent Power: } S = V * I^* \tag{3.1}$$

$$\text{Real Power: } P = real(S) \tag{3.2}$$

$$\text{Reactive Power: } Q = imag(S) \tag{3.3}$$

3.1 Sensor Suite

The particular device chosen for this study was the TED 5000 provided by The Energy Detective [2]. This device consists of two current clamps for the A and B phases of the typical U.S. residential power connection, in addition to A and B phase voltage probes.

This device can measure power at a maximum of once per second (1-Hz sample rate). It also has a maximum 1-watt resolution. Unfortunately, it was found that at this resolution the measurement is often too noisy to be useful, so a lower change threshold was used: about 5 to 15 watts. A integrated circuit in the measurement device takes readings during the sample period at line frequency and then averages the result to be outputted [14]. As a result of averaging over the entire sample period, interesting artifacts arise when multiple electrical events occur during the sample window. These effects are addressed later.

The TED 5000 was chosen as a result of its having characteristics similar to that of a smart meter in that it has capabilities that are well within the range of modern smart meter capabilities. In fact, smart meter sensor suites are expected to have better accuracy as a result of needing to perform revenue class metering[1] [15]. Processing power and bandwidth requirements are minimized as a result of recording only significant events and sending data off-meter for actual processing. The alternative is to perform all processing and data recording on-meter, but this would require significant resources.

As a result of the 1-Hz sample rate, the probability that two devices will overlap is not trivial (see Appendix A). However, since only a few devices in the system are of interest, we can assume that the probility of two "interesting" devices having events at the same time is below a safe threshold. Even if an insignificant and significant device overlap, since the insignificant devices are assumed to have a small power signature relative to the significant devices, the error in event detection will be small.

[1]Cited meter has the capability of measuring harmonics; therefore, the capability of performing at least a 1-Hz sample rate is implied.

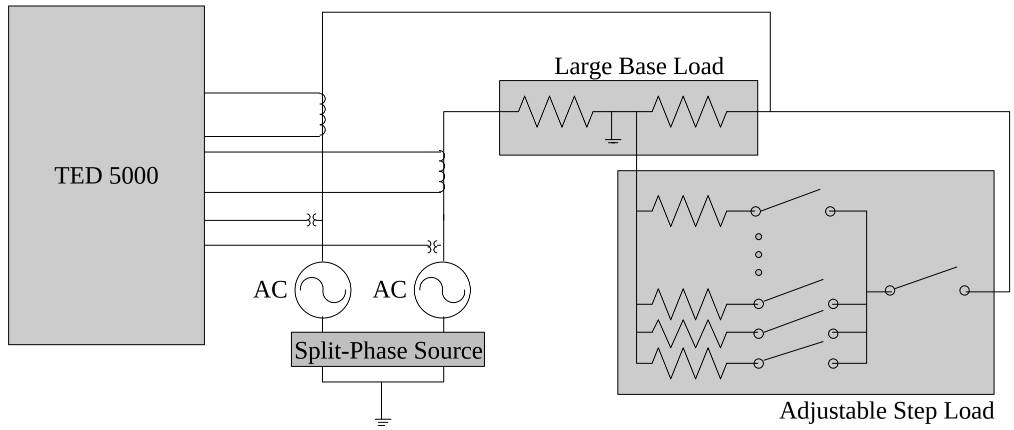


Figure 3.1: Experiment Schematic (lab meters not shown)

3.2 Current Transformer Saturation Study

As part of evaluating the accuracy of the measuring device, a lab experiment was designed to test for current transformer (CT) calibration and saturation. During routine measurement collection in a home, it was observed that measurements for a given device had some variance that depended on the base load of the system. From this observation, it was concluded that perhaps the current transformers were becoming slightly saturated and resulting in a change in measurement magnitude.

To test the ability of the measuring device to record accurate measurements over a series of base loads, a test load was set up that could be stepped on top of different base load levels. Figure 3.1 shows the schematic of the experiment.

First a baseline measurement was performed to test the fine-grained measurement accuracy of the TED system. In this case, the base load was turned off and the load was stepped in near 60-W increments. This process was repeated at five different load levels: 60, 120, 180, 240, 300. This sequence of step loads was then repeated with a base load of 2960 W and 3580 W. The voltage level during this experiment was nominally 240 V. Table 3.1 shows

Table 3.1: Results of Lab Experiment

Base Load (W)	Nominal Step Size (W)	Lab Meter (W)	TED Meter (W)
0	60	60	68
0	120	120	130
0	180	177	194
0	240	235	256
0	300	295	316
2961	0	2961	3074
2961	60	3004	3122
2961	120	3047	3156
2961	180	3087	3158
2961	240	3130	3196
2961	300	3172	3282
3586	0	3586	3700
3586	60	3627	3744
3586	120	3646	3744
3586	180	3686	3812
3586	240	3728	3808
3586	300	3772	3878

the results of the experiment as measured by the TED and by lab-quality meters.

As can be seen in figure 3.2, the current transformers show no saturation under significant base load. While the base load used in the experiment was not close to the maximum possible power available in a residential installation, it does accurately represent the load levels encountered during field data collection. This experiment served to show that the CTs are indeed accurate and the source of the error was not caused by saturation. As the CTs were rated for 100 amps, it was unlikely this was the cause of the error; but confirming the measurement system as a whole helped eliminate the possibility of measurement error. This experiment prompted further investigation into the cause of the error, the results of which will be discussed later.

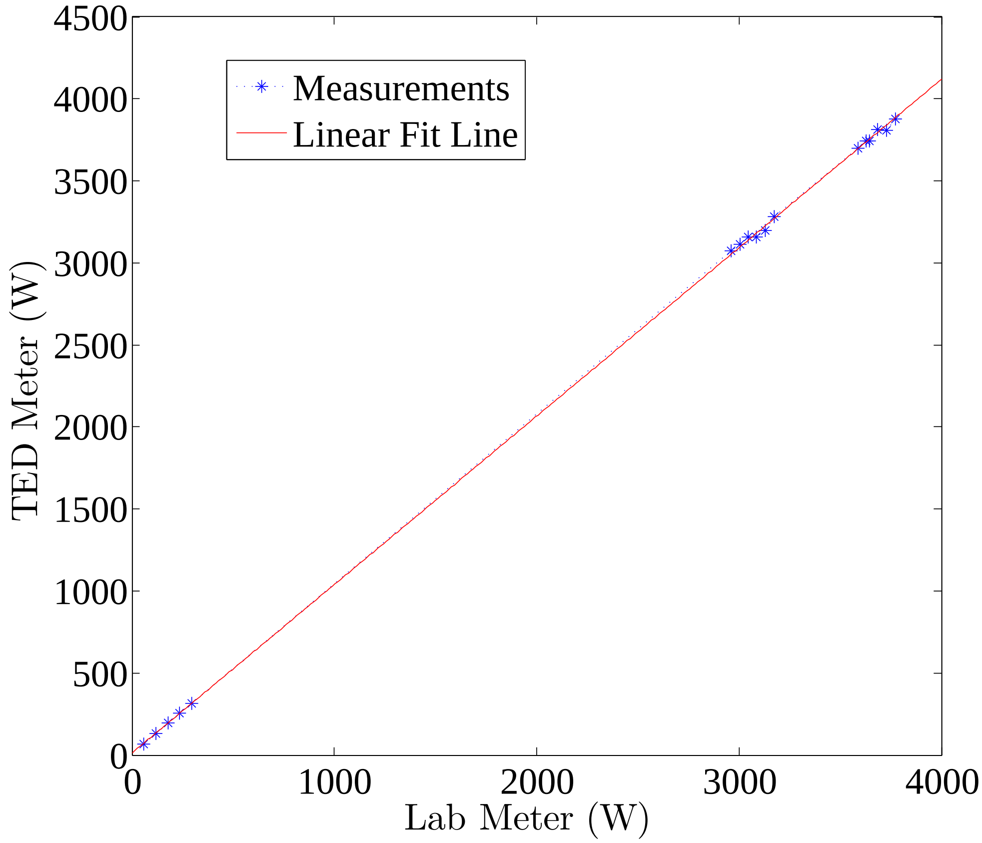


Figure 3.2: Current Transformer Test Results

3.3 Data Collection and Storage

In order to collect data from the TED at 1-Hz resolution, a custom computer setup had to be constructed. At the time of data collection, the TED recorded only real power and voltage at 1-second resolution, making it impossible to calculate the reactive component of the measurement. To overcome this limitation, a computer was set up to poll the device's API every second. The API provided access to real, apparent, and voltage measurements. Using this information, the reactive power quantity could be calculated. Figure 3.3 shows the data collection setup.

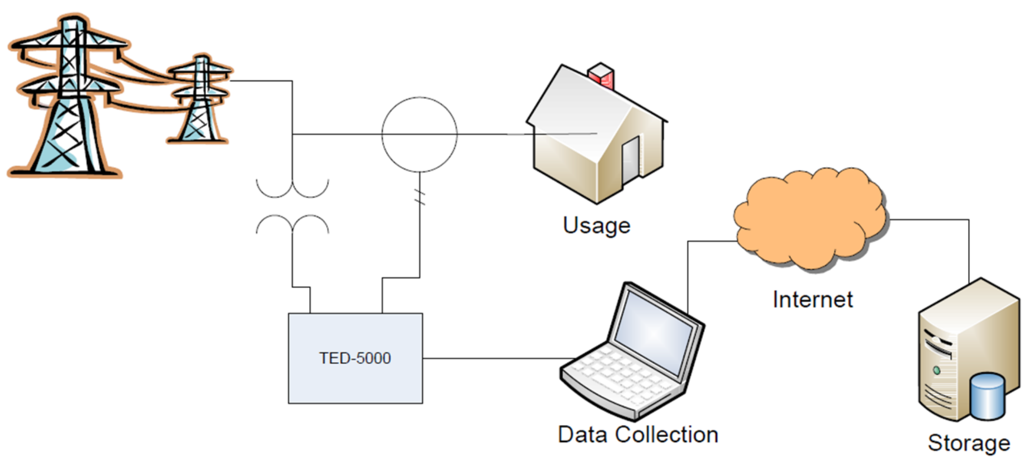


Figure 3.3: TED 5000 Data Collection

Data collected from the TED were stored locally on the polling computer until uploaded to a central server every 10 minutes. This scheme allowed the data collection computer to remain installed at a location for several weeks at a time without someone having to return to the location to retrieve the data. In an actual implementation, only the record of events will be uploaded to a central location rather than the entire measurement data stream. This compression serves to limit bandwidth usage and minimize storage requirements. In this case, the entire waveform was stored for research purposes. An important consideration in data collection is the security of the data. Data must be stored, transferred, and analyzed securely at all times. These requirements imply storing the data locally in a secure database, transmitting the data using an encrypted computer, and analyzing the data only on secure computers. Implications of obtaining NILM data and how to obscure usage data has been investigated and should be investigated before implementing a widely deployed system [16], [17].

CHAPTER 4

PREPROCESSING

When data are first collected direct from the measurement unit, they are in a raw form which requires some preprocessing. There are a few different artifacts that must be corrected before the data become useful. They include

- Transient startup artifacts
- Measurement noise
- Varying voltage
- Multi-sample events (sloped edges)

4.1 Normalization

When differences in event magnitude were first noticed in field data collection, it was assumed that the current transformers were saturating. In reality, the differences were caused by differences in system voltage during the events. When the system was heavily loaded, the overall system voltage would be decreased from when it was lightly loaded. System voltage is also dependent of several other factors, such as the state of the local power grid and time of day (see figure 4.1). This voltage difference affected the power consumed by devices as a result of most of the devices following a constant impedance load model. While the standard U.S. residential supply is constrained to vary

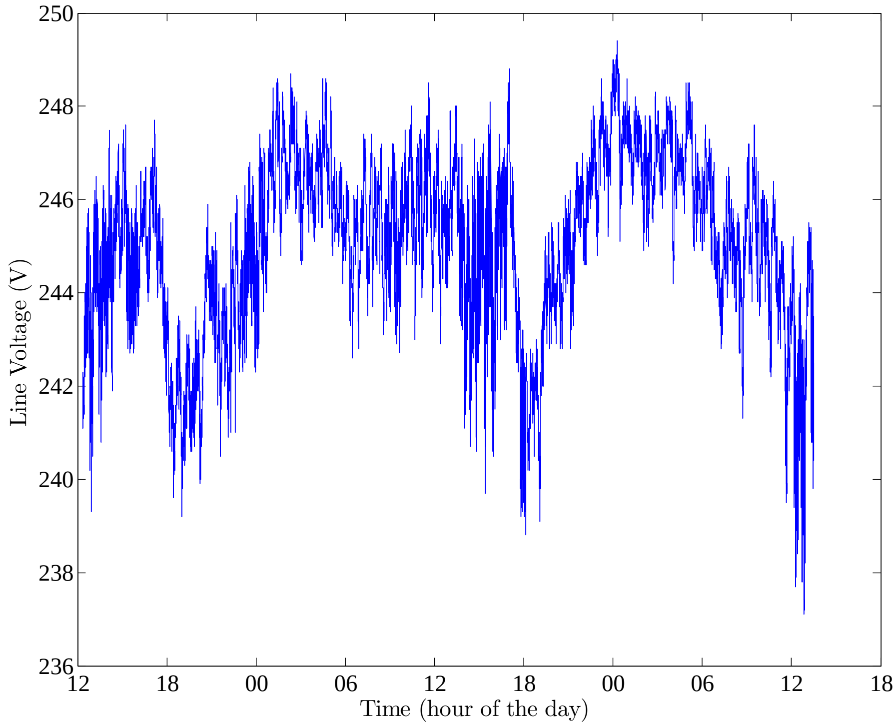


Figure 4.1: Example of voltage variation as measured at a test location over 48 hours, nominally 240 V

$\pm 10\%$, this would cause the power measurements to vary $\pm 20\%$. The solution to this variation is to normalize the power signals based on admittance [7].

$$Y(t) = \frac{P(t)}{V(t)^2} \tag{4.1}$$

$$P_{norm}(t) = 120^2 * Y(t) = (\frac{120}{V(t)})^2 * P(t) \tag{4.2}$$

The equation for admittance is shown in (4.1). As it is more intuitive to compare power usage instead of changes in admittance, this normalized admittance is then referred back to real power values in equation (4.2). A similar calculation can be performed for reactive power by normalizing the susceptance.

4.2 Filtering

After the varying system voltage had been addressed, two other artifacts needed to be corrected: transients and measurement noise. These two challenges can be corrected by filtering. The first attempt at filtering was to use a simple low-pass, finite-impulse response (FIR) filter. This filter design helped remove measurement noise but was significantly affected by start-up transients or events that had large changes. As a result of being a linear filter, edges were actually softened instead of being made more clear.

The solution to dealing with start-up transients and poorly defined events was solved by Norford and Leeb by using a nonlinear median filter [9]. A median filter takes a window of samples from a signal and then chooses the median value of the window as the value for the center of the window. The window then slides down the signal until all samples have been processed. The result is that edges in a signal become more defined and sort-lived transient events are removed. In testing this filter, a fast matlab implementation was used to process data collected in the field [18]. Figure 4.2 shows the dramatic change once the median filter is applied to the raw data. The result is even more dramatic when the clustering of the edges is examined. One of the analytic techniques for this data is to perform cluster analysis on the magnitude of the events [9], [12]. While this method was later found to be not the best method for analysis, it does demonstrate the effectiveness of the median filter. Figure 4.3(a) shows how the event magnitudes cluster together without filtering, while figure 4.3(b) shows the result when a median filter is applied. As can be seen by the differences in the plots, the median filter improves consistancy and allows for more accurate device profiles to be built.

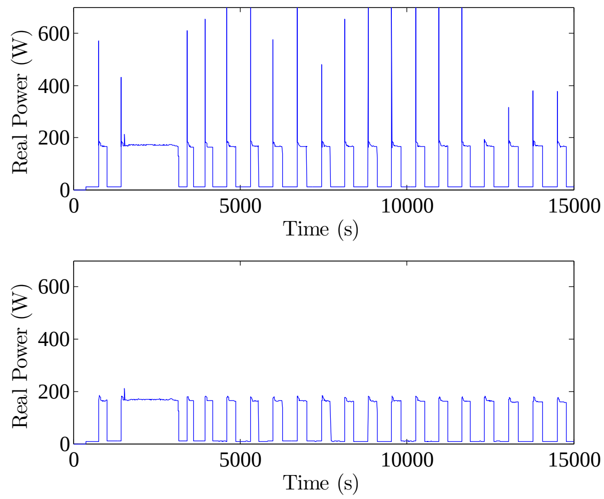
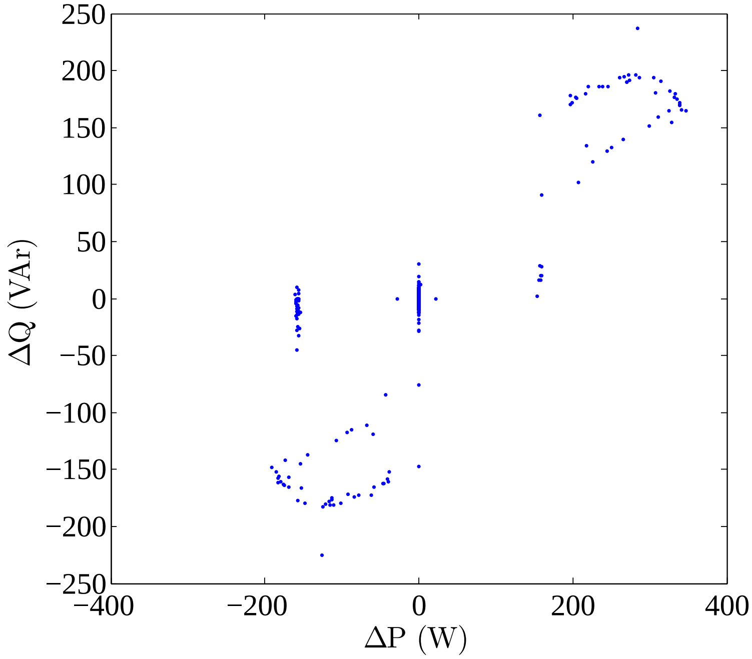


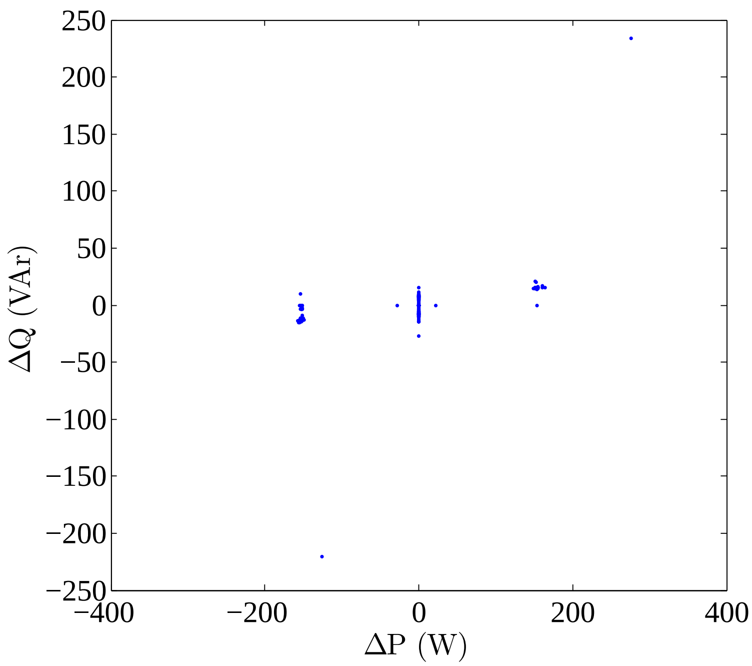
Figure 4.2: Top: Unfiltered Data; Bottom: Median Filtered Data

4.3 Threshold Processing

In examining figure 4.3, observe that an abnormally high number of events cluster around zero. These events are the result of noise and other small devices that were being measured on the system. There are also events that have no real power change while showing a small reactive power change. These measurements are not interesting and serve only to complicate the disaggregation process. To simplify the readings, real power measurements below 50 watts are ignored and any reactive power change that does not have a corresponding change in real power is also ignored. As was seen in table 2.1, most loads of interest are near 200 watts and above, therefore this threshold process is validated. Removing the high-density cluster near the origin also assists clustering algorithms to find meaningful groupings.



(a) Unfiltered $\Delta P/\Delta Q$ clustering



(b) Median filtered $\Delta P/\Delta Q$ clustering

Figure 4.3: Effect of Applying a Median Filter to Clustering

CHAPTER 5

EDGE DETECTION

In order to reduce storage requirements and improve the computational efficiency of this algorithm, an edge detection process was implemented to select key features from the measurement data. The method stores only large changes in the measurement rather than then entire waveform. This process also reduces the communication bandwidth needs, as only important features are transmitted. Building device profiles from edge information also simplifies the model for each device, reducing processing requirements. The edge detection itself is most likely to be the most processor-intensive operation of the system, as a result of having to touch every incoming sample.

Three methods will be described which have been experimentally tested on field data, one manually defined method and two automatic detection methods. Using a manually defined method is not completely out of the question if there is sufficient information about the devices on the system and if they operate in a strictly repeatable fashion. Automatic methods are preferred when the characteristics of devices change over time or are not generally uniform.

5.1 Methods

5.1.1 Absolute Differencing

Absolute differencing is a method in which the user defines a specific (ΔP, ΔQ) change that identifies an edge of interest. In this case, the set of edges that can be detected is limited to the number of entries that are provided by the user. If the system is simple, with relatively few devices, this method can be used to detect edges with very little processing requirements. This algorithm also limits the amount of information being passed to the next processing step, which again decreases the processing power needs. The algorithm simply compares the current sample to the previous sample. If the measured difference matches one of the predefined changes, then the sample is marked as a edge. The advantage of this method is that is has excellent temporal resolution as a result of having no inherent filtering effects, thus preserving the original edges.

The disadvantage of this method is that is requires a large initial setup effort by the user. The user must know how each of the devices of interest changes its real and reactive power consumption when it turns on and off. In addition, if the devices drift slightly in their characteristics, the events they produce may land outside the tolerance of the predefined edges. This would result in none of the edges being detected or otherwise being labeled incorrectly.

5.1.2 Windowed Derivative

The next method attempted was an automatic edge detection method. This method detects edges without having any information about the edges themselves. The only defining parameters are the window in which to look for edges and a threshold to cut out small edges. Utilizing an optimization algorithm, these two parameters were tuned to yield a collection of edges that captured the significant behavior of the system without crowding the later processing stages with superfluous information. The algorithm is as follows:

1. Derivative
   (a) Select a window of consecutive samples from the measurement data.
   (b) Perform a linear fit line calculation on the window.
   (c) Take the slope of the line to be the derivative of the middle sample.
2. Find local maxima of the derivative signal; ignore maxima smaller than a given threshold.

The local maxima of the derivative signal represent the location of the edges in the original signal. To find the magnitude of the edges, the difference in average value between the previous edge and subsequent edge is used. This technique requires waiting for the next edge to occur or waiting a predefined number of samples before calculating the magnitude of the current edge. This requirement is not seen as a limitation, however, as events occur quite regularly in a real-world system and the delay in determining the magnitude of a particular edge is not significantly long. Figure 5.1 shows an application of this algorithm.

The disadvantage of this method is that the strict window size does not

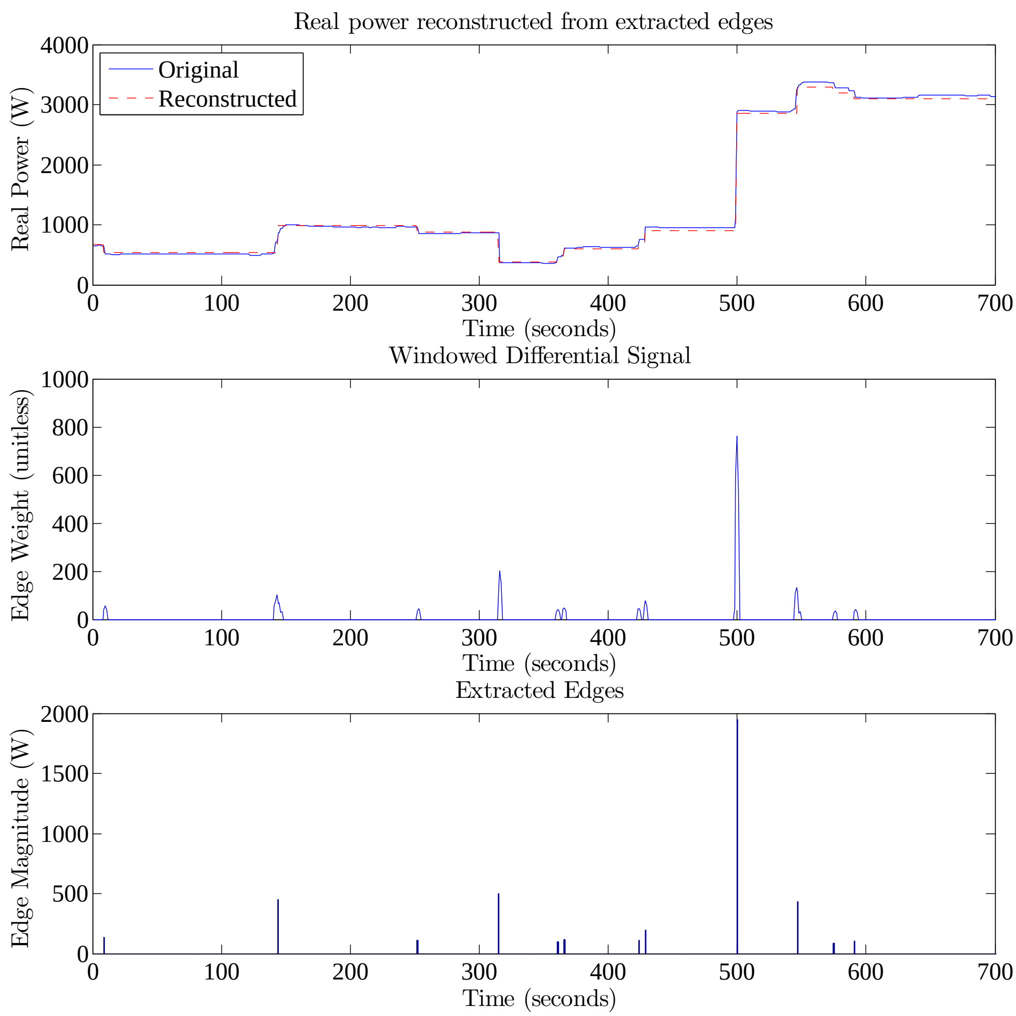


Figure 5.1: Application of Windowed Derivative Algorithm

handle well events that are close together. Events that are captured in the same window are most likely going to be combined. Selecting a smaller window helps differentiate events from one another but results in lower noise rejection.

5.1.3 Canny Edge Detection

To further improve the edge detection process, methods from image processing were investigated. Edge detection in images has been widely used in computer vision, particularly for recognizing changes in depth and orientation of a pictured object. In the 1-dimensional case, these algorithms can be applied to power signal processing. The particular method that was selected was developed by John Canny in 1986 [19].

In this method, the derivative of a Gaussian curve is convolved with the original signal. The result is a signal which has local maxima centered on the location of edges in the signal. This effect is similar to the windowed derivative method described before. Figure 5.2 shows how different functions can be convolved with the original signal to produce an edge detection signal. As before, this method detects only the location of edges and not the magnitude; therefore, a similar method of average differencing with neighboring edges is applied.

The advantage of this method is that it is well vetted by the image processing community and is very robust. The only tuning parameters to this method are the width of the Gaussian function and the threshold level to filter out insignificant edges. When combined with the median filter from the previous chapter, this method is highly accurate at locating edges. The figure 5.3 shows the results of applying this algorithm to collected field data.

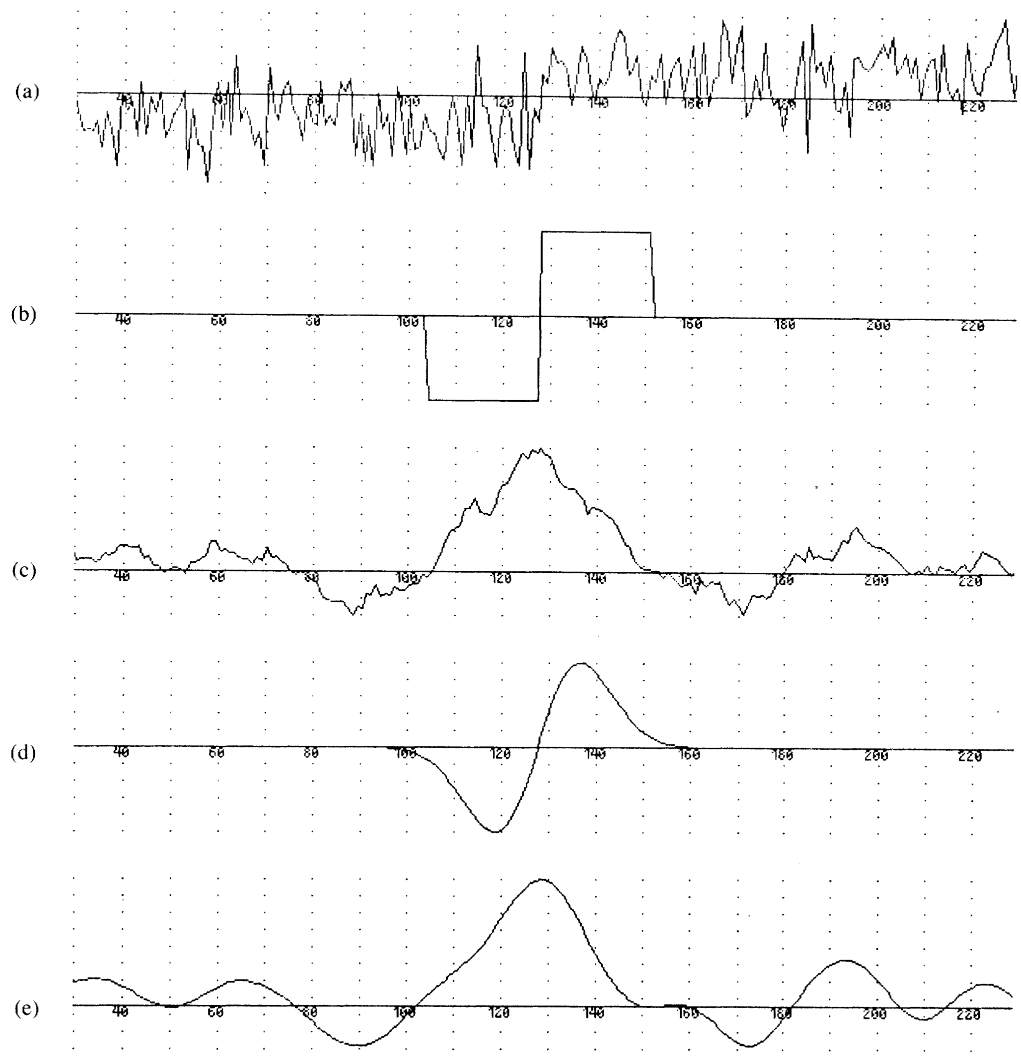


Figure 5.2: (a) A noisy step edge. (b) Difference of boxes operator. (c) Difference of boxes operator applied to the edge. (d) First derivative of Gaussian operator. (e) First derivative of Gaussian applied to the edge.

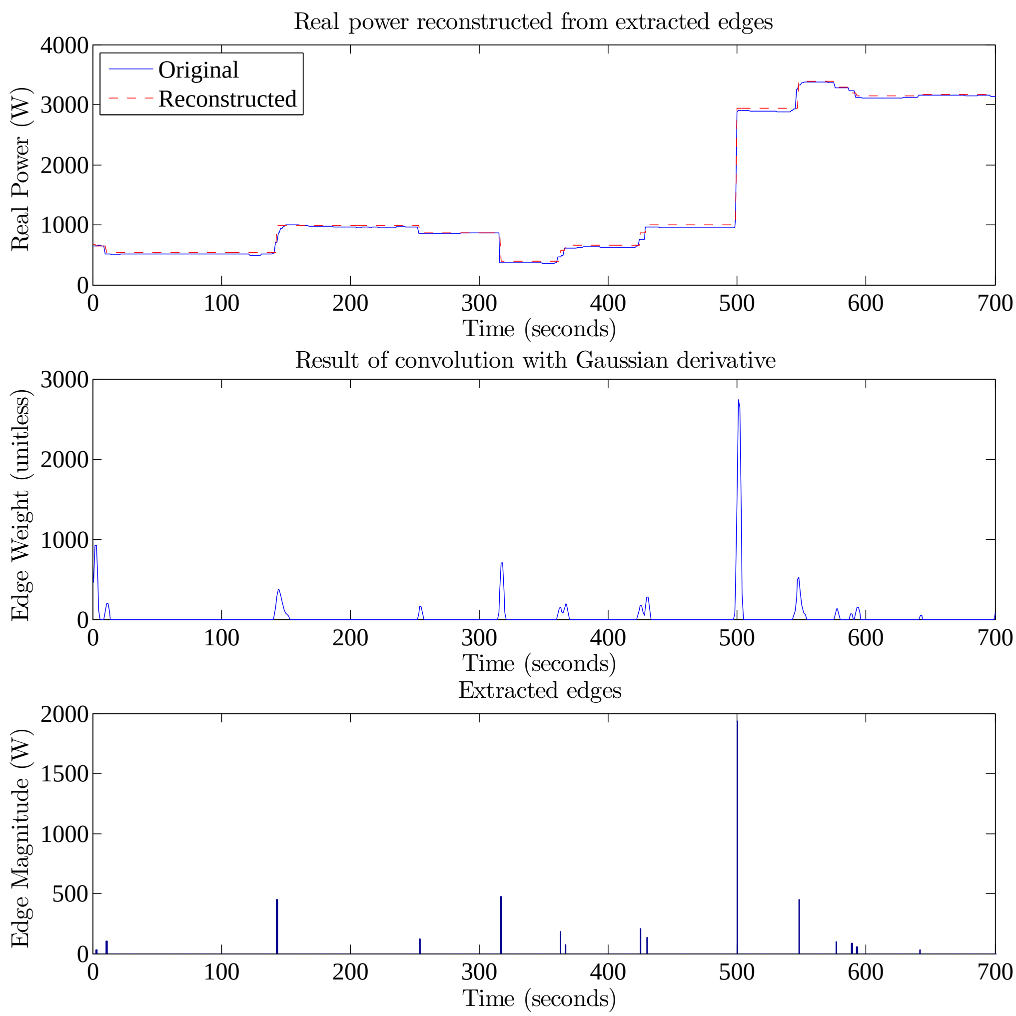


Figure 5.3: Application of the Canny Edge Detection Algorithm

5.2 Edge Matching

Often after extracting the edges from the measurement signals, events may be offset from each other from 0 to 2 samples. This effect results from edges that cover multiple sample periods. If the real power edge is detected on the front of the event and the reactive power event is detected on the back of the event, then a single event can be separated into two distinct edges. In order to build a uniform P/Q description for an event, these two edges have to be matched up to each other. In general, the real power edges are taken as the actual events and the reactive power edges are adjusted to the best real power edge. This prioritization was done because the real power values are being directly recorded from the measurement device, while reactive power is a calculated value which may present with modified edge characteristics from an actual reactive power measurement.

To perform edge matching, each edge in the real power collection is examined. If a reactive power edge is found within a specified range around the real power edge and no other real edge could claim this reactive edge, the reactive edge is aligned to the real edge. The process continues until all of the real edges have been processed. Often, a real power edge will occur without a significant reactive power edge. This type of event is the reason for looking only in the neighborhood of a real power edge for a corresponding reactive edge and not simply aligning the "closest" edges.

CHAPTER 6

POST-PROCESSING

There are many ways to extract information from data. Up to this point, data have been collected and manipulated, but information has not been extracted. In order to learn about the devices on the system, two approaches were attempted. The first attempt was to use clustering methods to match collected events with a particular device. In this case, when an event occurred, the event was labeled with a device name depending on the best matching event cluster. The second approach was to use a Hidden Markov Model (HMM). In this method, information about the current device states, collected measurements, and observation probabilities are taken into account to form a more complete state model.

6.1 Clustering

Clustering is the process of assigning a grouping to a set of data. The input dataset is divided into one or more clusters depending on certain characteristics that define how "close" or "far" a data point is from other points. Data can be divided in several ways, including partitioning, density-based clustering, and hierarchical clustering, among others. This particular problem lends itself primarily to density-based clustering (DBSCAN) as a result of exhibiting nonuniform clusters. However, k-means, a partioning method, did prove useful in applications with the HMM approach. QT-clustering [20] was also considered but did not prove to have any particular advantage over the previous methods.

6.1.1 Methods

K-Means

K-means is a partitioning-based clustering method. This method will divide a set of data into k distinct clusters without leaving any data points without a label. This approach is in contrast to a density-based method which leaves outliers without a label. K-means needs to be told how many clusters to partition the data into. As a result, in the particular application to power monitoring, the number of devices must be known ahead of time before clustering, assuming that each device has a distinctive on/off ΔP, ΔQ signature. The general k-means algorithm is as follows [21]:

1. Randomly seed the sample space with k distinct cluster "centers."
2. Assign each data point in the sample space to the closest cluster "center."
3. Move the k cluster "centers" to the mean of their member data points.
4. Go to step 2, until cluster assignments no longer change or the maximum number of iterations is reached.

As can be seen in step 1 of the algorithm, the process starts out with a random seeding. This step causes the process not to be strictly repeatable. A different initial seeding may result in different cluster assignments. To improve clustering, the process may be repeated several times, and the result with the least total distance sum between the centers and their member data points is used. Another artifact that may arise is that a cluster center may not be assigned any data points, as a result of being too far from other clusters. In this case, the cluster center is either dropped or the iteration is terminated before total convergence. K-means has the tendency to find circular clusters (if a Euclidean distance measure is used) or square clusters (if a city block distance is used). If the data does not fit one of these shapes, data may be clustered unnaturally.

DBSCAN

DBSCAN (Density-Based Scan Algorithm with Noise) is a clustering method that groups data points based on the distance between neighboring points rather than overall partitioning [22]. As a result, often data points are left without a cluster assignment if they fall outside a particular distance measure. As with other density-based methods, DBSCAN does not require knowledge about the number or shape of the clusters. Instead it requires information about the maximum distance, called the Eps-neighborhood, around a given point to search for other group members, and the minimum number of points to be considered a cluster. The complete algorithm is found below:

```
DBSCAN(SetOfPoints, Eps, MinPts)
  ClusterId := nextId(NOISE);
  For i from 1 to SetOfPoints.size Do
    Point := setOfPoints.get(i);
    If Point.ClId = UNCLASSIFIED Then
      If ExpandCluster(SetOfPoints,Point,ClusterId,Eps,MinPts) Then
        ClusterId := nextId(ClusterId);
      End If
```

```
    End If
  End For
End

ExpandCluster(SetOfPoints,Point,ClId,Eps,MinPts): Boolean
  seeds := SetOfPoints.regionQuery(Point,Eps);
  If seeds.size<MinPts THEN //not a core point
    SetOfPoint.changeClId(Point,NOISE);
    return False;
  else
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    While seeds != Empty Do
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,eps);
      If result.size >= MinPts then
        For i from 1 to result.size Do
          resultP := result.get(i);
          If resultP.ClId in {UNCLASSIFIED,NOISE} Then
            If resultP.ClId == UNCLASSIFIED Then
              seeds.append(resultP);
            End If;
           SetOfPoints.changeClId(resultP,ClId);
          end if;
        end for;
      end if
      seeds.delete(currentP);
```

```
    end while;
    return true;
  end if;
end;
```

The advantage of DBSCAN is that it can identify clusters with irregular shapes. Since the neighborhood around each point is evaluated instead of the distance to a cluster center, clusters can be built that follow an arbitrary shape. In the case of power data, clusters are often elongated as a result of measurement errors. A k-means method may partition a cluster such as this into two or more pieces, but DBSCAN would only create one. In addition, any point that is not placed is a cluster is considered an outlier, making this method highly noise tolerant.

The disadvantage with this method is that if distinct clusters are too close to each other, then they will be combined. Since k-means is a partitioning method, any two arbitrarily close clusters will be divided into two groups, although the division may not necessarily be meaningful. Selecting a smaller Eps radius will help solve this problem but may result in more clusters of fewer points, which will result in unnecessary cluster divisions.

6.1.2 Clustering Analysis

Applying clustering methods was the first attempt at producing meaningful information from the measurement data that was collected. The edge detection algorithm provided a set of $\Delta P/\Delta Q$ measurements. These edges were then clustered using the different algorithms. The first attempt was with the k-means algorithm. As mentioned before, it is necessary to know the number of clusters that should be found in the data. As it was difficult to determine

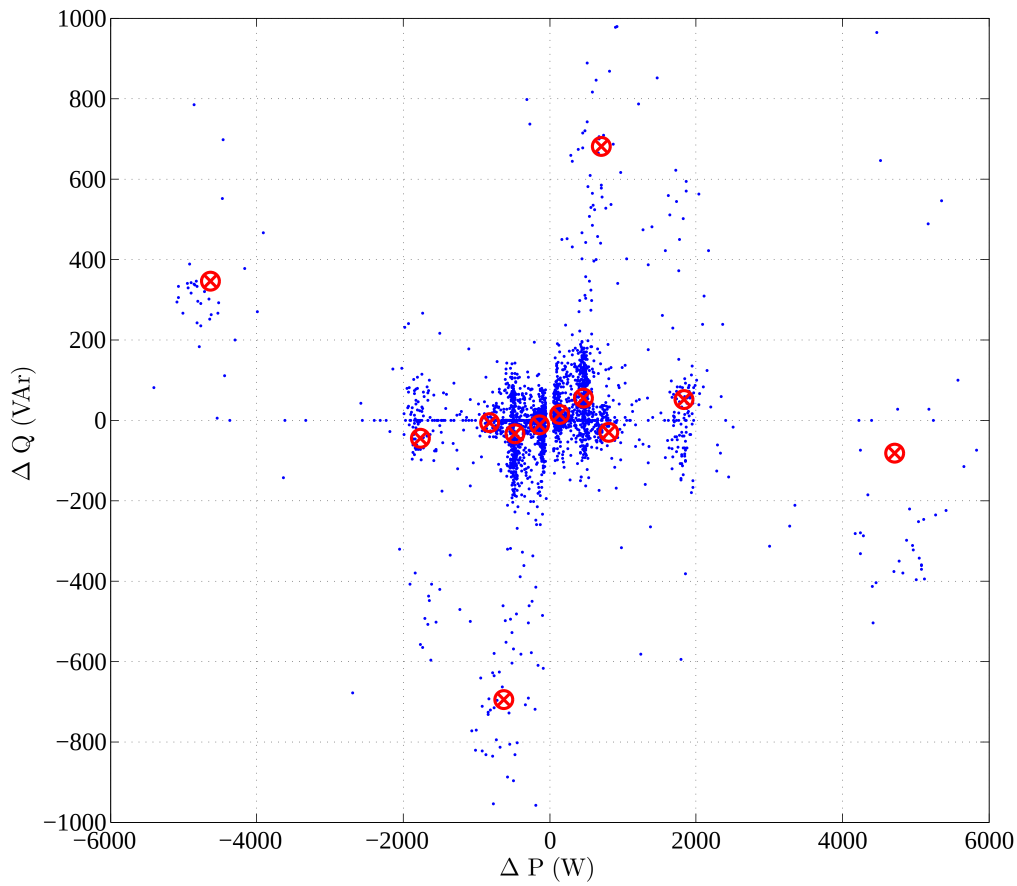


Figure 6.1: K-Means Clustering Algorithm Applied to 48 Hours of Data, k=12 Clusters

how many distinct devices would appear in the data, several different cluster numbers were tested. Figure 6.1 shows the results of applying this algorithm to the collected data. This figure shows a 48-hour sample of an entire house, looking for 12 partitions. Note that the measurement thresholding is also applied to this data, which results in small events and mismatched reactive power events being ignored.

As can be seen in figure 6.1, four large clusters are centered in the middle of the P/Q space. These most likely represent two refrigerators that were in the home. The cluster centered near ±2000 watts is a set of baseboard heaters. The sparse clusters near ±5000 watts belong to another baseboard heater but was not measured uniformly, resulting in poor grouping. Other devices such as computers, entertainment systems, and lights are not well represented by this clustering algorithm as a result of being lost in the measurement noise.

The next algorithm to be tested was the DBSCAN method. It was assumed that because the clusters were elongated and not in natural shapes that a density-based method would produce better results. Difficulty arose in choosing the two parameters for the algorithm, MinPts and the Eps radius. This selection was mostly done by trial and error. The larger the Eps radius, the less distinct the clusters will be; therefore, it is favorable to have a low eps radius. However, with a low eps radius, devices that resulted in sparse measurement readings would be discounted as outliers. In addition, if the event is infrequent, a high MinPts parameter will lead to ignoring these devices. After testing, it was determined that a small MinPts parameter and an Eps Radius of 12 to 15 worked well for the collected data.

Figure 6.2 shows the results of applying the DBSCAN algorithm to the same test data as the k-means demonstration. Points in black indicate outliers and all other data points are colored to identify the cluster they belong to. Note how the clusters are elongated and not necessarily in straight lines or circles. These clusters demonstrate the strength of the DBSCAN method in handling measurement variation. The downside, as evidenced in the figure, is that a significant portion of the data was discounted as outliers. This effect was particularly evident in the case of the 5 KW heater, which operated infrequently but was still present in the data. As a result of measurement inconsistency and infrequent operation, this device was outside the detection threshold of the algorithm.

The DBSCAN method proved difficult to tune in order to capture all of the desired devices while leaving out the noise. K-means was useful in find-

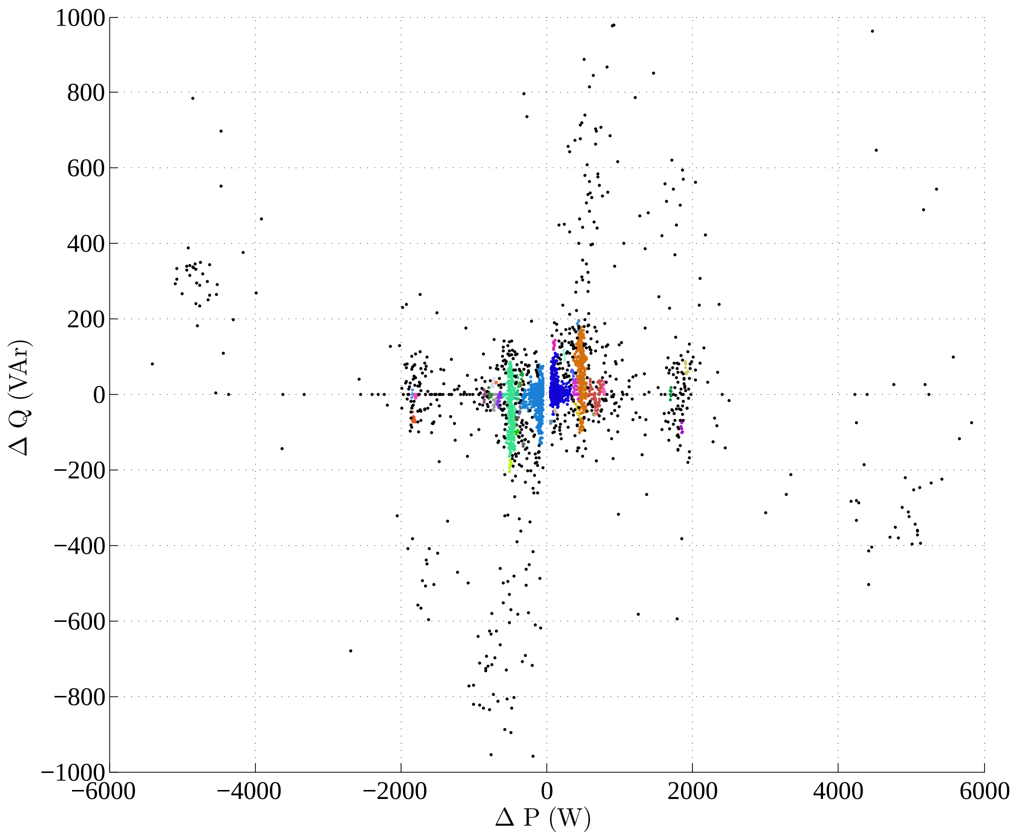


Figure 6.2: DBSCAN Algorithm Results, MinPts = 3, Eps = 15, Applied to 48 Hours of Data

ing infrequent events but did not provide useful clustering for devices with measurements spread out in nonuniform manners. Unfortunately, a problem inherent to all clustering methods was discovered: overlapping clusters. Some devices on the system, due in part to slight variation in operating characteristics and due to measurement variance, appeared to be very similar in the P/Q space. This case can be easily demonstrated in the case where two identical appliances are in the same house, for instance, two window air-conditioner units. These types of devices are too important to leave out of the analysis or lump together and so must be separated in another manner.

Clustering provided a good initial analysis of the collected data but could not provide enough distinction between different devices in order to build a profile of an individual device. There is significant information, however, about the loads in the home. Using this clustering information, general information about the home can be extracted. In this case, clustering can be used to identify the frequency of different load levels in the home. In examining figure 6.2, it can be seen that the most frequently operating loads in the home consume less than 1 KW and that loads greater that 4 KW rarely operate. From this, it can be inferred that the home most likely has a gas hot water heater and stove instead of electric models which would show frequent high-power changes.

In addition, this data shows that the reactive power requirements for devices in this home are generally less than 200 VAr. This information could be used by utilities to accurately plan for the amount of compensation needed on a distribution feeder.

This data can also be used for verification of installing energy-efficient appliances and weatherproofing. If the number of member data points in a given cluster decreases after installing an energy-efficient appliance, it can be inferred that the device is operating less frequently and therefore most likely consuming less energy. The same method can be applied to weatherproofing. If the addition of insulation and weather stripping has indeed had an effect on the energy usage of the home, the overall density of points in a given cluster should decrease. It is not necessary to directly isolate the furnace from the rest of the data collected as long as it is approximately known which cluster the furnace belongs to.

6.2 Hidden Markov Models

To solve the problem of overlapping events in the P/Q space, more information needed to be added to the analysis. When clustering algorithms were applied to the data, all information about the timing of the events was lost. To restore this dimension, timing and state information needed to be considered. A Hidden Markov Model approach proved to be one of the best ways to keep track of the state of the system while handling the uncertainty associated with the measurements being collected.

6.2.1 Overview

Hidden Markov Models (HMM) are a particular specialization of Markov Models [23]. In a general Markov Model, states of a system are arranged with transition rates between each of the states. The probability of being in one state and transitioning into the next state is a direct function of the transition rates between the states. In the general Markov Model, the states are directly observable. For example, take a 100 W lightbulb that has two states: on and off. The state of the lightbulb can be directly observed and the transition rate between on and off easily figured.

The *hidden* part of HMM comes in when the states of the system are not directly observable, but instead observations of certain qualities can be made when the system is in a certain state. In the case of the previous example, the lightbulb can be covered with a metal bucket. Since the state of the lightbulb can not be directly observed, the underlying model is hidden from view. Instead, the temperature of the bucket can be measured. When the light is on, the temperature will be "high." When the light is off, the temperature will be "low." There is some uncertainty in the measurement, however, possibly as a result from the room temperature varying or the time it takes to heat and cool the metal. When the system changes state, the observations of the temperature will change accordingly.

If a sequence of measurements is taken, the challenge is to determine what sequence of states best matches the observations. This task is met by the Viterbi algorithm [24]. Originally proposed by A. J. Viterbi in 1967, this algorithm is used to determine the maximum likelihood path through a set of state transitions. This process assumes that the structure of the underlying model has been discovered and the state transitions are known. If this information is not known, the Baum-Welch algorithm [25] can be used to identify the transition and emission parameters of a HMM given a set of training data.

6.2.2 Advantages

The problem of power usage disaggregation applies itself particularly well to Hidden Markov Models. In the case of a home power system, the states of the devices in the home are unknown and can not be directly observed. Instead, the devices in the home produce different emissions depending on

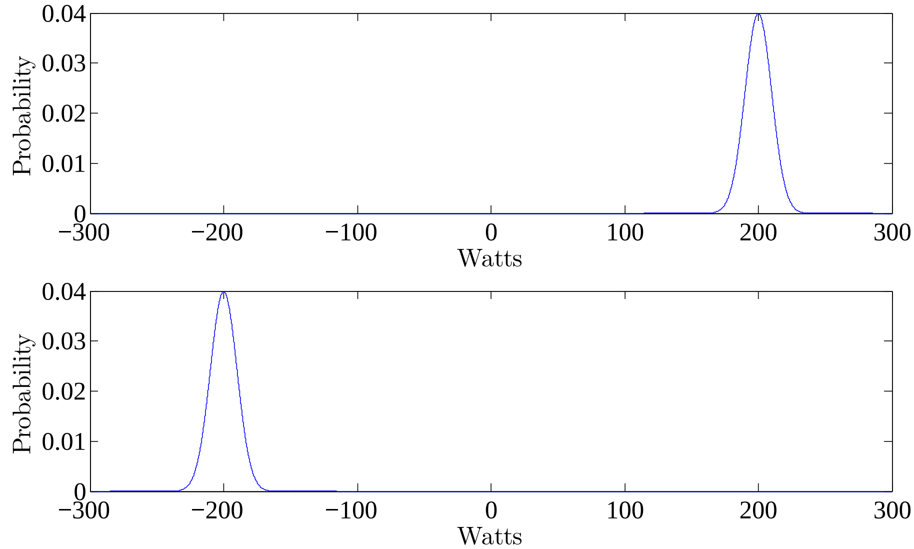


Figure 6.3: Top: Emission PDF of a Device Turning On; Bottom: Emission PDF of a Device Turning Off

the state they are in. In this case, real and reactive power consumption will change in a correlated manner depending on what device changed state. As seen before, there can often be variations in the measurements which can be accounted for by assigning emission probability density functions (pdf) to each of the devices. Figure 6.3 shows an example of the emission probabilities for a generic refrigerator.

This method also takes into account the current state of the system when evaluating what measurements belong to which devices. In the HMM solution, if two similar events were observed one after the other, it would not be assumed that they came from the same device but instead that they were from two different devices, even though the characteristics seemed similar. Utilization of state information is difficult in cluster methods as a result of eliminating time relational information.

6.2.3 State Machine Implementations

There are several ways to construct the underlying state machine for the model. Two of these methods were investigated for use. The first involves modeling the entire system using a single-state machine with many states representing all combinations of on/off states that the loads in the home can take. The other method actually breaks the HMM into several individual state models, one for each device in the home.

Single, many-state machine

The single-machine, many-state model offers a complete picture of the state of the devices of interest in the home. This model quickly becomes very complicated as devices are added to the system. The advantage of this model is that it can utilize the full strength of the Viterbi algorithm. As a result of tracking the state of all of the devices, the most likely path through the state transitions can be easily evaluated.

Issues arise when devices are added to the system. As devices are added, the model becomes exponentially more complex. As each state in the HMM emits a unique signal that is then measured by the sensor, several states arise for each combination of device states. For instance, in a two-device case, there are two ways the "all off" state can be reached: by turning A off or B off, with the opposite device already in the off state. Since the emission when device A turns off may be different from the emission when device B turns off, it is necessary to have two "all off" states. Figure 6.4 shows a connection diagram for a two-device, single-state machine example.

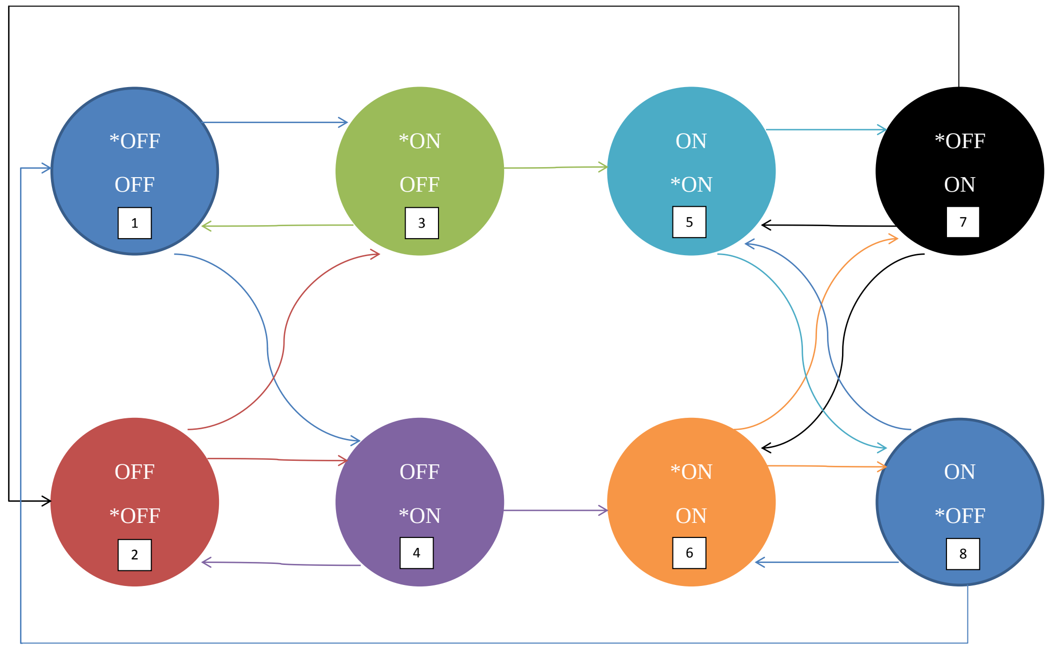


Figure 6.4: Two-Device, Single-Model Example (*Denotes device that changed state)

Multiple, single-device machines

The alternative to a single-state machine model is to divide the system into several individual state machines. Each device on the system is modeled by two (or more, depending on the device) states that are not connected to any other device states. When a new observation is received, it is checked against all of the individual state machines to determine which device most likely emitted that event. In this manner, current state and time relational information can still be utilized but without the complexity of a combined model.

The disadvantage of modeling the system in this manner is that the full strength of the Viterbi algorithm can not be utilized. As each of the observations must be considered individually without relation to the previous observations, it is not guaranteed that the generated state sequence is the most likely. If the observations were considered in relation to each other as they are in the many-state model, then the most likely state sequence is guaranteed to be most likely. While the state information is being updated after each emission is evaluated, only one step through the state matrix is being considered instead of the many steps that are evaluated in the full Viterbi algorithm.

6.2.4 Experimentation and Results

To apply a Hidden Markov Model approach to this problem, the first step is to develop emission probabilities for each of the devices that is to be detected. While this could be done using a Baum-Welch algorithm, it is more efficient to independently measure the device of interest for a short sample time and then analyze the isolated device. A limited number of events is required to build a profile; but if more samples are included, the accuracy of the model will be improved. This type of information can be collected in a "survey" of the devices in the home or, more practically, pulled from a library of existing devices. A survey of devices in the test collection home was completed in a single afternoon. During the same time the devices were being profiled in an isolated manner, data on the entire house were also collected.

To reduce processing requirements, discrete emission profiles were created for the devices instead of the continuous profile shown in figure 6.5. To reduce storage requirements, the manner in which measurements tended to cluster in certain groups was taken advantage of. As seen in figure 6.1, much of the sample space is unoccupied or sparsely occupied. It would be a waste of storage space to divide up the sample space into evenly sized discrete bins. Instead, using the k-means method, the sample space was divided up into 50 non-overlapping subsets.[1] In this manner, the dense measurement clusters

[1] 50 was chosen by experimentation to achieve an acceptable level of division while still

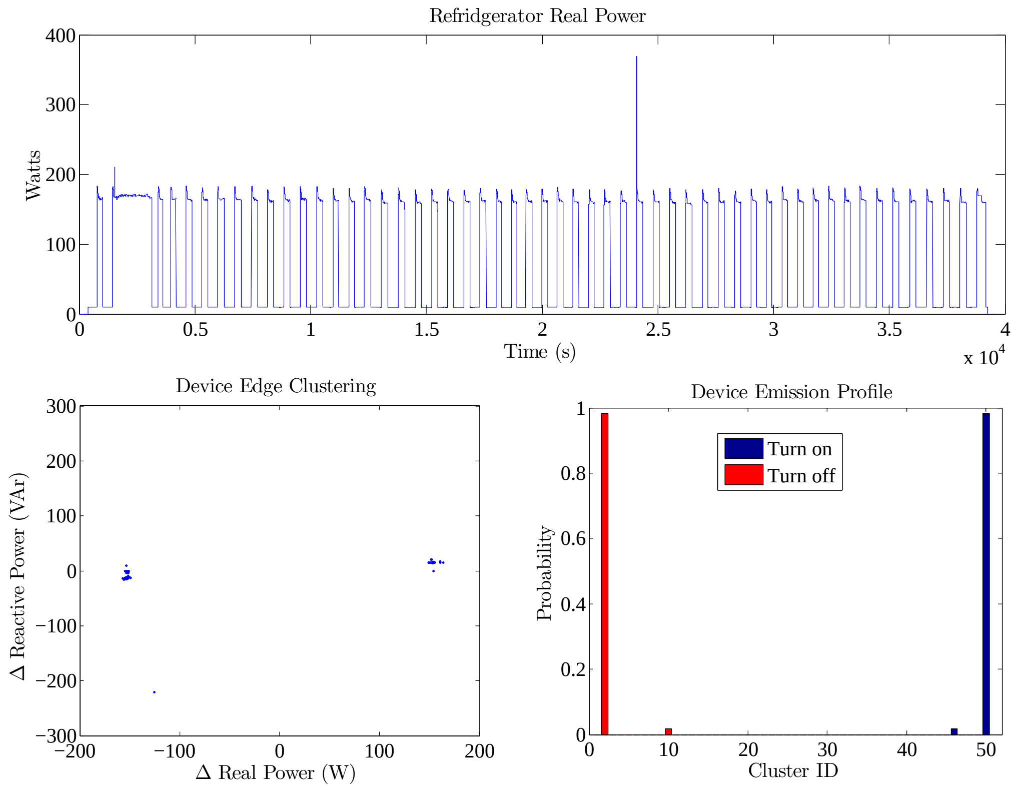


Figure 6.5: Creation of an Emission Profile for a Refrigerator

are divided up into clusters of greater detail, while the sparse regions are covered by more general labels. The DBSCAN method was not used in this case because the number of desired divisions was known and it was desired to cover the entire sample space without an outlier classification.

Using the binning that was developed using the k-means method, the edges detected for a individual device could be labeled as a certain "class" of edge. By counting the number of edges in each class, an emission profile for each of the devices could be developed. Figure 6.5 shows an isolated sample for a refrigerator and how its edges were classified into a "turn on" and "turn off" profile.

Most of the devices in the home followed a simple two-state model: on

maintaining a relatively low number of clusters to evaluate.

and off. The transition rates between these states were very simple to derive. When the device is off, it will always transition to on if an "on" event is observed and vice versa. Therefore the transition matrix for each of the two-state models is stated in equation (6.1).

$$T = \begin{array}{cc|ll} \multicolumn{2}{c|}{\text{Current State}} & & \\ \text{On} & \text{Off} & & \\ \hline 0 & 1 & \text{On} & \multirow{2}{*}{Next State} \\ 1 & 0 & \text{Off} & \end{array} \tag{6.1}$$

By evaluating which device a given emission belongs to, the most likely device can be chosen and its state updated. The remaining devices maintain their current state. Initially all of the devices are in an indeterminate state, having a 50% probability of being on or off. After an event is received and assigned to a device, the probability of the device states are adjusted to reflect the new information.

Experiment on Test Data

To test the functionality of the algorithm, a set of test data was developed that was clean and uniform. This test set was functioning under the assumption that data collected in the field could be processed sufficiently to meet these conditions. As shown in the chapter on preprocessing, these assumptions are not far from what can currently be accomplished.

In this simulation, five two-state devices were placed on the system. Two of these devices had similar emission profiles and could be considered copies of the same device, i.e., a home with two refrigerator units. The third device was a model of a induction motor on a HVAC unit. The forth and fifth devices represented electric resistive base-board heaters of 2 KW and 5 KW

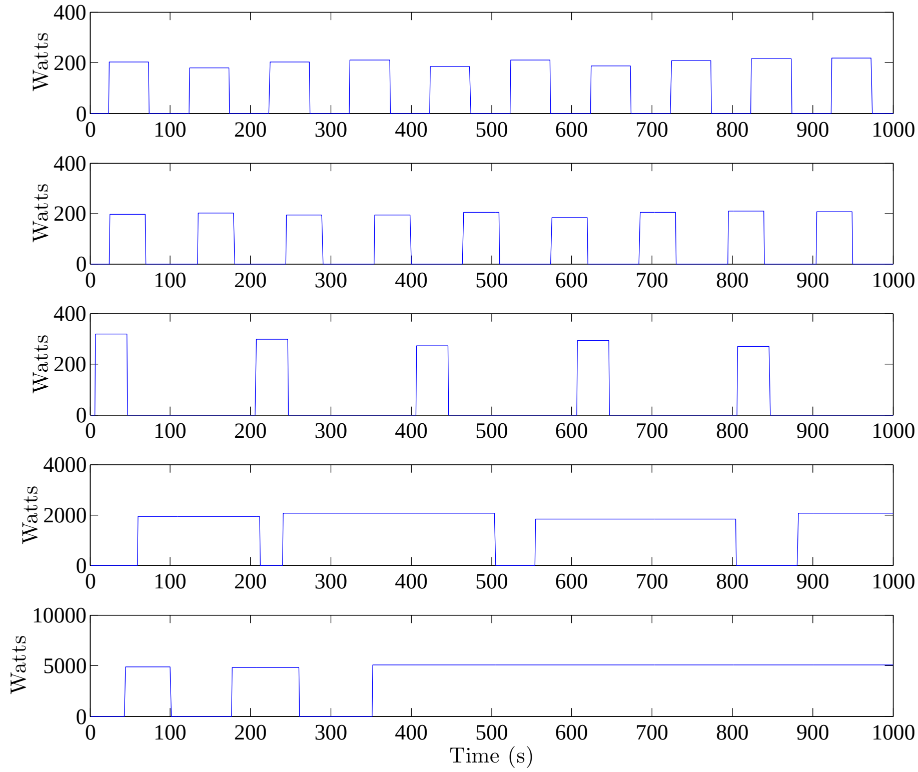


Figure 6.6: Simulated Devices

ratings. A measurement variance of ±10% was used to vary the simulated events. It was assumed that all of the individual electrical events could be correctly isolated and the magnitude correctly recorded. Figure 6.6 shows the input data to the simulation. Note that none of the events in this simulation were overlapping.

This input data was applied to a HMM model which consisted of five two-state machines. Using the emission probability matrix that was built from the training data and a simple state transition matrix, the state of the devices could be estimated using the stream of "measured" events. Figure 6.7 shows the probability of the different devices being on or off according to the measurement data. Note that before a matching event is detected, the state of the device is undetermined with an on probability of 50%. It can also be seen that the last two "on" events for device 1 and 2 are swapped with each other: a result of both devices being equally likely when a measurement was received. Device 1 was arbitrarily chosen as the device that was assigned to the event. While not currently implemented, this type of error could be reduced by taking advantage of the fact that many of these devices follow a certain duty cycle. The emission matrix can therefore be modified to reflect the probability of the device changing state.

Other information can also be added to improve the accuracy of the state estimation. The more dimensions that can be added, the more accurate the model. Absolute time of day information can be used to more easily identify devices that show a pattern of being used during a certain time of the day. For instance, a coffeemaker that is used only in the morning may be missed because it is an isolated event. Utilizing the time of use information, the emission matrix for this device could be adjusted to boost the probability of a matching electrical event being assigned to this device.

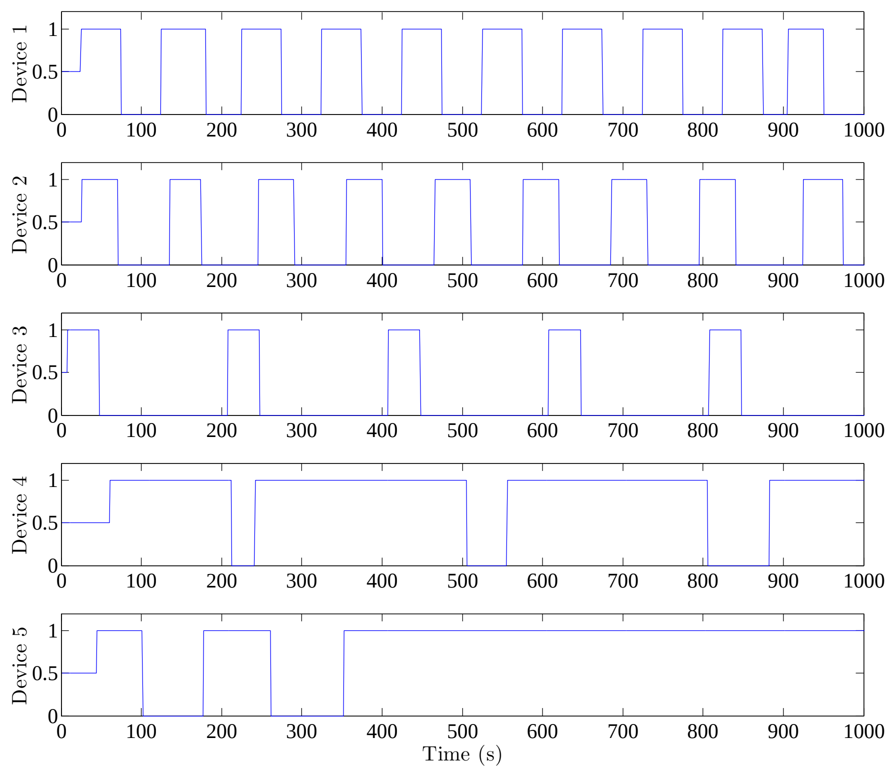


Figure 6.7: Results of the HMM Simulation: Probability of a Device Being in the “On” State

CHAPTER 7

APPLICATIONS AND STUDIES

After information about the loads in the home has been collected, it can be applied to several functions. As information is the basis for the "smart grid," this type of low-level, granular information can be very useful to monitoring and control schemes. For instance, utilities could know not only the current load level but also the composition of the load. Consumers can directly use this information as an incentive to install energy-efficient appliances and receive measurable results back. Another benefit gained from this type of information is identifying devices for control. Utilities could then use this information for targeted advertising or other outreach programs to improve demand-side management programs.

7.1 Smart-Grid Control Verification

One application for device-level data is to verify that a certain control measure has been effective. In particular, this can be applied to load shed verification at the device level [26]. If a command has been issued by the utility but the NILM system still detects the device, then the device may be damaged, overridden, or otherwise disconnected from the demand-side management system.

This type of diagnostic test would have not been available if only an aggregate house-level measurement was taken. While a reduction in consumption may have been observed, it is difficult to determine if the result was incidental or if the desired control action was taken, especially if multiple devices are being controlled in the house. If only one of three devices respond to the control, it is difficult to determine which one responded and which did not. Utilizing device-level information allows for discrete evaluation of the performance of a demand-side management program.

7.2 Identifying Devices for Control

Following along the same lines as verification is the need to expand control of the grid. While demand-side management (DSM) is not a new technology, it is showing a recent resurgence due to improved communication structures and consumer cost benefits [27]. These programs have often proved not to be cost-effective for utilities due to unfavorable market conditions and consumer dissatisfaction, among other factors. In order to maximize the effect of DSM programs, devices with high impact need to be chosen for control. These devices must also be controlled in a minimally intrusive manner so as to limit consumer disruptions. This information is exactly the type provided by a NILM system.

Information from an NILM system can be used to select devices that have certain characteristics favorable to DSM. For instance, devices that are operated infrequently during the day are most likely of high importance to the consumer and are most likely not available for control. However, infrequent devices that occur during the night are probably available for control. While controlling devices at night may not be of particular use for peak-curtailing, this demonstrates that information about when a device is used can be utilized when selecting potential control candidates.

Devices that have regularly repeating cycles can also be detected using the technology. In the home, cyclic devices are most likely thermostatic devices such as refrigerators, heat pumps, and water heaters. These types of devices often have enough thermal mass to withstand skipping an active cycle or two without having the temperature deviate significantly. These devices often represent a significant load on the system and can therefore provide a large relief when controlled effectively (see table 2.1). Without device-level information, it may be difficult to identify cyclic devices. With NILM, these devices can be easily isolated and examined to see if cycles can be skipped.

Many of the loads in a home are inductive motor loads. Most pumps, fans, and compressors are driven by induction motors. In a low-voltage scenario, during a fault, induction motors are detrimental to the system, as they promote further voltage sag [28]. Using NILM to identify devices that have a low power factor, these devices can be selected for passive control schemes during blackout restoration. In this type of scheme, identified devices would be forcefully disconnected from the grid for a random amount of time after the distribution system is energized. After the given time, devices can be switched back in and allowed to start, minimizing the reactive power demand when energizing a distribution feeder. Without NILM it would be very difficult to identify individual devices that consume a high amount of reactive power. With this technology, control devices can be efficiently placed on targeted appliances.

7.3 Energy-Efficiency Confirmations

One of the most difficult aspects of implementing energy-efficient measures in the home is the lack of positive feedback to the consumer. For instance, a large investment may be made in new insulation in the attic of a house, improving the overall energy usage of the house, even though the temperature difference may be hard to notice. If the consumer instead looked at the behavior of the HVAC unit in the home, it would be more apparent that unit is spending less time on and is more lightly loaded.

Using the performance data from specific devices, improvements in the building envelope such as new windows, insulation, or weather stripping can be more easily recognized. When the performance of each of the conditioning elements in the house can be identified, a more detailed cause and effect relationship can be realized. In addition, the monetary payback for energy-efficiency improvements can be more closely monitored and incentivized. For instance, an insulation manufacturer may guarantee a certain duty cycle reduction for a given system or pay the difference if it doesn't meet the goal. This offer would encourage consumers to make purchases they normally would not because of the lack of tangible results. While this type of information can be gained though other methods, it is given in a cost-effective manner in an NILM installation. Utilizing one set of sensors, several different conditioning units can be monitored. As these loads often represent a significant portion of the aggregate load, they are easily identified and tracked over time.

CHAPTER 8

CONCLUSIONS

Utilizing an approach of effective data collection, clustering, and Hidden Markov Modeling, it was possible to implement the necessary parts of a nonintrusive load monitoring scheme using constraints similar to the capability of modern smart meters. Challenges identified earlier, such as limited bandwidth, limited computing power, and inaccurate sensors were effectively dealt with. Unfortunately, a complete system that combined all of the various components of the algorithm has not been completed and is left to future work. While the simulation was successful, it was not possible to implement the algorithm using real-world data. It is assumed that the obstacles holding back the design can be overcome relatively quickly.

8.1 Current Approach, Advantages and Limitations

The system as a whole is divided into three modules: data collection, preprocessing, and post-processing. The data collection method was successfully implemented on real hardware and allowed for secure collection of real-time usage data. This setup closely represented the resolution and sample rate of a "smart meter" device. Data was also securely stored and transfered to a central storage location for analysis. This "store and forward" method of data collection minimized the bandwidth requirement and allowed for long-term storage of data.

The primary limitation of this part of the system was the reliability of the measurement device. The TED 5000 proved to be not very reliable when used in this type of data collection. The device would often reset or simply fail completely. These problems made it very hard to collect continuous measurements for more than a few days at a time. It would have been better to collect data in monthlong blocks to observe long-term patterns. As part of the profiling process, data were collected from a main, whole-house, monitoring device and also from a secondary, device-monitoring level. Unfortunately, the two devices were not calibrated to each other, which resulted in different waveforms being observed at the two locations. Normally, absolute measurements are not strictly required for device identification but since different devices were being used for profiling and long-term data collection, the profile did not match the aggregate data stream. The alternative explanation is that there is a secondary effect that changes the measurement result, depending on the base load level that is being measured. This possibility, however, was ruled out by the experiment detailed in the hardware chapter.

After data were collected, individual devices were profiled and the aggregate data were partitioned into individual "bins." As explained, this part of the algorithm started as a method for finding devices in the aggregate signal; but when this proved intractable, the clustering methods were used for pre-processing the data. The signal normalization method suggested in [7] proved to eliminate the measurement variation which was noted in the hardware evaluation. To eliminate noise from the signal while sharpening edges, a median filter was applied. While a custom method for edge detection was developed, the Canny edge detector method proved to be more rigorous and reliable.

As with most filtering algorithms, information is lost after passing through the filter. In general, the lost information is "garbage" information anyway and serves only to complicate the signal; but in a few cases, the lost information is valuable and should not have been discarded. In applying the median filter to the power signals, most of the noise was eliminated. However, if events occurred close to each other in time, the event was either completely filtered out or its magnitude detrimentally altered. While the median filter did a much better job at filtering than any of the linear filters applied, other filter designs need to be considered. In addition, once the filter has done its job, it should be a relatively simple process to identify edges in the signal. This precise filtering would eliminate the need for a separate edge detection module. Ideally, after filtering, edges will be sharp and single-valued instead of spread out over several samples. In this case, edges could easily be detected by comparing each sample to the previous one. In addition, this characteristic also helps determine event magnitude, as the difference between neighboring samples can be considered the actual event magnitude.

Finally, the Hidden Markov Model, which does most of the heavy lifting in this algorithm was used as the post-processing structure. This method provided excellent support of measurement variance, state tracking, and secondary effects such as absolute time dependence and duty cycle influence. By implementing the HMM using a time-varying emission matrix, much of the heuristic knowledge about a certain device can be included in the model. This additional knowledge improves the overall accuracy and robustness of the model. This type of model adjustment is not currently implemented, but the concept has been tested using a simple simulation. In the simulation, devices were assumed not to be rapidly acting, so the modeled device was not allowed to change for 10 seconds after changing state. This assumption successfully allowed more devices to be tracked in a tight time frame than the original capability.

One drawback of the current HMM implementation is that it utilizes the multi two-state machine model instead of the single multi-state model. This design limits the capability of the Viterbi algorithm and prevents the system from taking advantage of several other features of HMM. For instance, if a single model were used, the Baum-Welch [25] algorithm could be applied to enable the system to learn about the devices in the system. This configuration would limit the need for active device profiling and instead turn learning into a passive process. Information about the system would still be required but would be significantly decreased from the current profiling method.

8.2 Further Research

This research represents the preliminary work required to develop a full non-intrusive load monitoring system for the smart grid. Several areas of research can be continued from this work. The first area would be to continue this research in order to create a fully functional system. As mentioned before, several assumptions prevented this algorithm from functioning on real-world data with any acceptable level of accuracy. To correct this limitation, the process will need to be cleaned from the bottom up, starting with accurate device profiling. Ideally this profiling process would not be required, but for now, it remains a necessary step. Different collection devices should be explored, including using actual smart meter devices. Unfortunately, getting real data from a utility smart meter may prove rather difficult. A range of devices used for home energy monitoring should be able to effectively take the place of a smart meter. It is important to identify the limitations of different brands of meters to test to robustness of the overall algorithm.

Efficient methods of data storage and transfer should also be tested. The current method of data transfer should scale well, but an actual implementation should be done to test the algorithm under real-world conditions. In addition, different methods of securing the data should be tested.

Another area of potential development is the filtering stage. As mentioned above, the median filter proved to be a valuable tool to take noise out of the signal and sharpen the edges. However, this process is not perfect and should continue to be further refined. Utilizing a custom nonlinear filter, it should be possible to identify edges that are close together, while still maintaining a high degree of noise isolation. Once a filter is developed with these properties, it could be simply plugged into the existing NILM algorithm.

The current Hidden Markov Model approach has been effective, but it is still limited. As discussed before, the current model uses multiple two-state machines instead of the ideal case of one multi-state machine. Different underlying methods of modeling devices in the home should be investigated to determine the most compact yet accurate model. By leveraging different features of HMM, such as autonomous learning and future state prediction, the model can become more accurate and its usefulness improved.

One of the largest parts of this project is what to do with the data after individual devices have been isolated on the system. As discussed in the studies chapter, the information provided from these devices can be utilized in a number of ways. As a result of the limitation of the data collection system, no long-term studies were performed using this NILM algorithm. A long-term data collection and analysis should be performed to determine the limitations of this NILM system in the context of annual data analysis. Other applications of NILM data should be investigated to determine its worth not only to the consumer but also to utilities and device manufacturers.

APPENDIX A

PROBABILITY OF SIMULTANEOUS ELECTRICAL EVENTS

It is well known that the probability of two independent events occurring together is given by equation (A.1).

$$P(A \text{ and } B) = P(A) * P(B) \tag{A.1}$$

Since the measurements are discrete in the NILM applications, the meaning of events occurring at the same time is modified to mean events occurring in the same sample period. In this particular case, simultaneous events are events that occur within less than 1 second of each other.

Let us now consider the probability of a given device emitting an electrical event during any measurement period. For instance, a refrigerator with a 50% duty cycle and 10-minute period will emit two events events every 10 minutes. This behavior can be represented as a emission rate as in equation (A.2). A probability can be derived from a given rate by multiplying by the sampling window time as in equation (A.3).

$$\lambda_{fridge} = \frac{\text{2 Events}}{\text{10 Minutes}} = \frac{1}{300}\text{events/s} \tag{A.2}$$

$$P_{fridge} = \lambda_{fridge} * \Delta T = \frac{1}{300}\text{events/s} * 1s = \frac{1}{300} \tag{A.3}$$

Other devices such as lightbulbs and other electronic devices do not follow a

Table A.1: Selected average device emission probabilities

Device	Probability
LCD TV	0.025
Microwave	0.00012
Desktop Computer	0.0022
General Home Lighting	0.0028
Refrigerator	0.0033
Mini Fridge	0.0042

strict duty cycle. By observing the usage of various devices in a test residence over a 24-hour period, the average probability of the devices emitting an event was calculated and is shown in table A.1.

The probability that any of the devices are on at the same time is given by equation (A.4).

$$\begin{aligned} P_{any} =& 1 - \sum_{i=1}^{n}(1 - P_i) \\ =& 1 - (0.975 * 0.99988 * 0.9978 * 0.9972 * 0.9967 * 0.9958) \\ =& 0.03725 \end{aligned} \tag{A.4}$$

The probability that only one device is on, with all other devices off, is given by equation (A.5).

$$\begin{aligned} P_{one} =& \sum_{i=1}^{n}(P_i * \prod_{\substack{j=1 \\ j\neq i}}^{n}(1 - P_j)) \\ =& 0.03476 \end{aligned} \tag{A.5}$$

Utilizing Bayes Theorem shown in equation (A.6), it can be seen that whenever any event occurs on the system, there is a 93% probability that is was only a single event. Unfortunately, this also means that there is a 7% probability that a simultaneous event occurred. This probability is acceptibly small, especially considering that the most frequent event emitters are of small magnitide, which does not significantly affect the accuracy of the system.

$$\begin{aligned} P(A|B) &= \frac{P(B|A) * P(A)}{P(B)} \\ &= \frac{1 * 0.03476}{0.03725} \\ &= 0.93315 \end{aligned} \tag{A.6}$$

Where:

$P(A)$ = Probability of only 1 event occurring

$P(B)$ = Probability of any event occurring

$P(B|A)$ = Probability of any event occurring when 1 event occurs

REFERENCES

[1] "Advanced metering infrastructure (AMI)," February 2007, Electric Power Research Institute (EPRI). [Online]. Available: http://www.ferc.gov/eventcalendar/Files/20070423091846-EPRI%20-%20Advanced%20Metering.pdf

[2] "The energy detective," February 2011. [Online]. Available: http://www.theenergydetective.com/

[3] "Envi whole home energy monitors by powersave," February 2011. [Online]. Available: http://www.currentcost.net/

[4] "Powerhouse dynamics — electric energy monitoring system for home and office," March 2011. [Online]. Available: http://www.powerhousedynamics.com/

[5] D. Brown, "Identification of appliances in residential electricity consumption," M.I.T. Dept. EE/CS BS Thesis, Massachusetts Institute of Technology, Cambridge, January 1983.

[6] "Nonintrusive appliance load monitor published references," February 2002. [Online]. Available: http://www.gerogehart.com/research/nalmrefs.html

[7] G. Hart, "Nonintrusive appliance load monitoring," in *Proceedings of the IEEE*, vol. 80, no. 12, December 1992, pp. 1870–1891.

[8] "Variable speed drives in home appliances to double by 2013," November 2009, IMS Research. [Online]. Available: http://imsresearch.com/news-events/press-template.php?pr_id=1177

[9] L. Norford and S. Leeb, "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms," *Energy and Buildings*, vol. 24, pp. 51–61, September 1996.

[10] S. Shaw et al., "Instrumentation for high performance nonintrusive electrical load monitoring," *Transactions of the ASME. Journal of Solar Energy Engineering*, vol. 120, no. 3, pp. 224–229, August 1998.

[11] S. Leeb, S. Shaw, and J. Kirley, "Transient event detection in spectral envelope estimates for nonintrusive load monitoring," *IEEE Transactions on Power Delivery*, vol. 10, no. 3, pp. 1200–1210, July 1995.

[12] C. Laughman et al., "Power signature analysis," *IEEE Power and Energy Magazine*, pp. 56–63, March/April 2003.

[13] "Energy savers: Estimating appliance and home electronic energy use," February 2011. [Online]. Available: http://www.energysavers.gov/your_home/appliances/index.cfm /mytopic=10040

[14] "CS5461 data sheet," February 2011. [Online]. Available: http://www.cirrus.com/en/products/eol/detail/EOL313.html

[15] "The encompass electronic meter family," February 2011, kv2cb_emf.pdf. [Online]. Available: http://www.gepower.com/prod_serv/products/metering/en/downloads/

[16] G. Hart, "Residential energy monitoring and computerized surveillance via utility power flows," *IEEE Technology and Society Magazine*, pp. 12–16, June 1989.

[17] T. Nicol and T. Overbye, "Toward technological defenses against load monitoring techniques," in *North American Power Symposium (NAPS)*, 2010, pp. 1–5.

[18] M. Little and N. Jones, "Sparse Bayesian step-filtering for high-throughput analysis of molecular machine dynamics," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4162–4165.

[19] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, November 1986.

[20] L. Heyer, S. Kruglyak, and S. Yooseph, "Exploring expression data: Identification and analysis of coexpressed genes," *Genome Research*, vol. 9, pp. 1106–1115, 1999.

[21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fith Berkeley Symposium on Math, Statistics and Probability*, vol. 1, 1967, pp. 281–297.

[22] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[23] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77, no. 2, February 1989, pp. 257–286.

[24] G. Forney, "The Viterbi algorithm," in *Proceedings of the IEEE*, vol. 61, no. 3, March 1973, pp. 268–278.

[25] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The Annals of Matematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[26] D. Bergmand et al., "Nonintrusive load-shed verification," *IEEE Pervasive Computing Special Issue on Smart Energy Systems*, vol. 10, no. 1, pp. 49–57, Jan. - Mar. 2011.

[27] D. Loughran and J. Kulick, "Demand-side management and energy efficiency in the united states," *The Energy Journal*, vol. 25, no. 1, pp. 19–43, 2004.

[28] C. W. Taylor, *Power System Voltage Stability.* New York, NY: McGraw-Hill, Inc., 1994.