Title
Energy Saving in Home Entertainment Systems via Dynamic Modeling

Permalink
https://escholarship.org/uc/item/02m83243

Author
Wang, Xiao

Publication Date
2015

Peer reviewed|Thesis/dissertation
UNIVERSITY OF CALIFORNIA,
IRVINE

Energy Saving in Home Entertainment Systems via Dynamic Modeling

THESIS

submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in Electrical Engineering

by

Xiao Wang

Thesis Committee:
Professor G.P.Li, Chair
Assistant Professor Mark Bachman
Assistant Professor Marco Levorato

2015
DEDICATION

To

my dear parents
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRACT OF THE THESIS</td>
<td>viii</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>CHAPTER 1: Background</td>
<td>6</td>
</tr>
<tr>
<td>On-market Energy Saving Solutions</td>
<td>7</td>
</tr>
<tr>
<td>CHAPTER 2: Data Acquisition</td>
<td>12</td>
</tr>
<tr>
<td>Analysis</td>
<td>13</td>
</tr>
<tr>
<td>User Interface</td>
<td>13</td>
</tr>
<tr>
<td>CHAPTER 3: CalPlug SIM Lab</td>
<td>16</td>
</tr>
<tr>
<td>General Testing Conditions</td>
<td>17</td>
</tr>
<tr>
<td>Configuration</td>
<td>18</td>
</tr>
<tr>
<td>General Testing Procedures</td>
<td>19</td>
</tr>
<tr>
<td>Sample Test Results</td>
<td>20</td>
</tr>
<tr>
<td>CHAPTER 4: Density histogram</td>
<td>25</td>
</tr>
<tr>
<td>Model Selection and Training</td>
<td>28</td>
</tr>
<tr>
<td>EM Algorithm for Gaussian Mixture Model</td>
<td>38</td>
</tr>
<tr>
<td>Result and Accuracy Demonstration</td>
<td>39</td>
</tr>
<tr>
<td>Conclusion</td>
<td>40</td>
</tr>
<tr>
<td>CHAPTER 5: Real Time Sample Demonstration</td>
<td>42</td>
</tr>
<tr>
<td>Filter Design</td>
<td>43</td>
</tr>
<tr>
<td>Different Games Study</td>
<td>44</td>
</tr>
<tr>
<td>CHAPTER 6: Conclusion and future work</td>
<td>46</td>
</tr>
<tr>
<td>REFERENCES (OR BIBLIOGRAPHY)</td>
<td>48</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.1</td>
<td>Control procedure of an APS with IR sensor</td>
</tr>
<tr>
<td>2.1</td>
<td>System set up</td>
</tr>
<tr>
<td>2.2</td>
<td>Web-Modeling UI</td>
</tr>
<tr>
<td>2.3</td>
<td>The detailed pop-over page</td>
</tr>
<tr>
<td>3.1</td>
<td>CalPlug SIM LAB</td>
</tr>
<tr>
<td>3.2</td>
<td>Testing configuration</td>
</tr>
<tr>
<td>3.3</td>
<td>Power consumption for playing game</td>
</tr>
<tr>
<td>3.4</td>
<td>power consumption for Game Idle</td>
</tr>
<tr>
<td>3.5</td>
<td>Power consumption for XBOX stay on main menu</td>
</tr>
<tr>
<td>3.6</td>
<td>Power consumption for streaming video from DVD</td>
</tr>
<tr>
<td>3.7</td>
<td>Power consumption for streaming music from CD</td>
</tr>
<tr>
<td>4.1</td>
<td>Percentage Histogram for XBOX Model 0</td>
</tr>
<tr>
<td>4.2</td>
<td>Percentage Histogram for XBOX Model 1</td>
</tr>
<tr>
<td>4.3</td>
<td>Percentage Histogram for XBOX Model 2</td>
</tr>
<tr>
<td>4.4</td>
<td>Percentage Histogram for XBOX Model 3</td>
</tr>
<tr>
<td>4.5</td>
<td>Percentage Histogram for XBOX Model 4</td>
</tr>
<tr>
<td>4.6</td>
<td>Gaussian distribution works good for one sample data set</td>
</tr>
<tr>
<td>4.7</td>
<td>Gaussian Distribution for XBOX Model 0</td>
</tr>
<tr>
<td>4.8</td>
<td>Gaussian Distribution for XBOX Model 1</td>
</tr>
<tr>
<td>4.9</td>
<td>Gaussian Distribution for XBOX Model 2</td>
</tr>
<tr>
<td>4.10</td>
<td>Gaussian Distribution for XBOX Model 3</td>
</tr>
</tbody>
</table>
Figure 4.11    Gaussian Distribution for XBOX Model 4 33
Figure 4.12    Using two Gaussian distributions to train Model 1 34
Figure 4.13    Gaussian Mixture Model for XBOX Model 0 35
Figure 4.14    Gaussian Mixture Model for XBOX Model 1 35
Figure 4.15    Gaussian Mixture Model for XBOX Model 2 36
Figure 4.16    Gaussian Mixture Model for XBOX Model 3 36
Figure 4.17    Gaussian Mixture Model for XBOX Model 4 37
Figure 5.1     Loading power consumption feature for Game 1 45
Figure 5.2     Another Loading power consumption feature for Game 1 45
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table 3.1</th>
<th>Example of LabVIEW output data</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.1</td>
<td>Result for XBOX Model 0</td>
<td>39</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Result for XBOX Model 1</td>
<td>39</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Result for XBOX Model 2</td>
<td>40</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Result for XBOX Model 3</td>
<td>40</td>
</tr>
<tr>
<td>Table 4.5</td>
<td>Result for XBOX Model 4</td>
<td>40</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Real time sample</td>
<td>42</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Result for real time sample</td>
<td>42</td>
</tr>
<tr>
<td>Table 5.3</td>
<td>Filtered result</td>
<td>44</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my committee chair and advisor, Professor G.P.Li, without his guidance and persistent help this thesis would not have been possible. And I sincerely thank him to offer me a chance to become his Ph.D. student.

I would like to thank my mentor Sergio for his kindest help and suggestions, he is strict with me on research but I know he is doing this all for my own good.

I would like to thank my committee members, Professor Marco Levorato and Professor Mark Bachman. Thank you Professor Marco Levorato for providing so many ideas in energy efficiency and I hope I can learn more from you in my Ph.D. study. And thank you Professor Mark Bachman to be my committee member and I hope we could have more cooperations in the future.

I thank my team members Michael Chao, Anchit Roy, Corey Vu and Shixuan Cai in CalPlug for assisting me with testing, writing and helping me a lot with coding problems and it's really good to have you around.
ABSTRACT OF THE THESIS

Energy Saving in Home Entertainment Systems via Dynamic Modeling

By

Xiao Wang

Master of Science in Electrical Engineering
University of California, Irvine, 2015

Professor G.P.Li, Chair

We propose a new system which can recognize and classify the different operational states of individual devices in home entertainment systems based on their power consumption models with the goal of improving energy management systems, providing information for energy saving, and reducing energy wasting. Modern advanced power strips can provide energy saving solutions by applying different kinds of sensors to monitor the user activities and make the decision to cut off the power source of the controlled outlet. However, normally this kind of solution needs a long trigger time and has less accuracy due to the differences in user behavior. Our system is able to effectively identify a device’s class and its operational states to provide a more efficient solution for home entertainment systems’ energy saving. We use LabVIEW and power analyzer to collect real time power consumption data for the different states of individual devices every second. Tracking and understanding this data will let us establish the dynamic models of the operational states in home entertainment systems. We devise recognition
techniques based on those dynamic models by characterizing these devices’ signatures and their power consumption distributions to build an algorithm to recognize the operational states of different devices. We designed a prototype integrated with our algorithm and models that demonstrated up to 60% wasted energy could be saved. This system would enable home entertainment systems to become more energy efficient and further reduce energy waste by combining user behavior tracking and prediction.
INTRODUCTION

A growing amount of over 3 billion plug load devices (PLDs) represent over 10% of the U.S. residential electricity usage, and this percentage is expected to reach 40% by 2030, as emerging technologies are introduced to the market[1]. Home entertainment systems have especially become the main driver for advancement in electronics and network science and technology in the past decade. The concept of home entertainment has evolved from individual PLDs into a networked system. A typical home entertainment system includes a television set, DVD/Blu-ray player, set-top box (service provider or retail), game console (e.g. XBOX one, PlayStation 4, and Wii), and audio system. Nowadays, home entertainment systems occupy Americans for an average of five hours per day, and these systems consume on average between 700 to 800 KWhr per household annually, which translates to over 9TWhr of energy consumption in California[2].

Huge energy loss follows the large home entertainment systems’ user community. There are two inevitable situations wasting our energy and money. Firstly, vampire power, normally called standby power, represents the energy loss while devices are switched off but not unplugged or switched to a standby mode. The leaking electricity by vampire power is different but normally low; however, multiplied by the billions of PLDs in American households, the wasted energy is 5% of the national residential electricity consumption. In other words, it is equivalent to the output of 18 typical power stations[3]. This vampire power also produces 1% of the world’s carbon dioxide emissions[4]. Secondly, before PLDs switched to a standby mode or off mode, due to differences in user
behavior, PLDs are wasting energy while they are still on because of how users are absent. The wasted energy seems inevitable because PLDs or modern solutions all need a trigger time to switch PLDs into a standby mode or off mode, which still lead to vampire power wasting. Therefore, we propose a new system which can recognize and classify the different operational states of individual devices in home entertainment systems based on their power consumption models with the goal of improving modern energy management systems, providing information for energy saving, and reducing energy wasting. Realizing the huge energy waste and the growing trend of energy prices, energy management systems which reduce energy wasting in household have demonstrated its importance in modern life: advanced power strips, smart meters, intelligent sockets, and so on, have become popular in consumers’ houses to save energy. However, effectively identifying devices’ operational states have not been reported yet. Moreover, user behavior on their PLDs usage cannot be easily tracked and analyzed. Modern energy management systems are collecting user activities information from various kinds of sensor and historic power consumption data from smart meters. However, normally this kind of solution needs a long trigger time and has less accuracy due to the differences in user behavior. Our system is able to effectively identify a device’s class and its operational states to provide a more efficient solution for home entertainment systems’ energy saving. By detecting devices’ idle states, we can cut off their power source in a shorter trigger time to reduce energy wasting. In our system, collecting devices’ power consumption data plays a critical role. Tracking and understanding this data will let us establish the dynamic models of the operational states in home entertainment systems. Also, collecting real time power consumption data
provides us an effective way to monitor state changes to make the decision to cut off the power source when we apply this system. The California Plug Load Research Center (CalPlug) at the University of California at Irvine (UCI) provides its SIM (simulation integration majority) lab to help me in collecting data from their typical American household living room. We conducted an independent study about power consumption of PLDs’ different states by collecting data from power analyzer and recording it by LabVIEW every one second. A standardized testing condition and procedure was used at CalPlug SIM lab, and different participants behaved as they would when they are at home to generate power consumption data along with more user behavior.

In the CalPlug SIM lab, the home entertainment system includes a television, a DVD player, a set-top box, an XBOX 360, and an audio system. The specificity of our development is we invited groups of volunteers from UCI to participate into our data collection progress. We collected power consumption data for these five devices under different conditions with different users. Over 100 hours of data was collected to establish power consumption distributions and models for PLDs’ classes and their operational states classification. We devised recognition techniques based on those dynamic models by characterizing these devices’ signatures and their power consumption distributions to build an algorithm to recognize the operational states of different devices. The first challenge we met is selecting proper pattern recognition model to generate our dynamic models from the first hand power consumption data collected from SIM lab. After trial and error, we decided to use Gaussian Mixture Model (GMM) as power consumption distributions are always nonlinear but follow Gaussian distribution in partial range. A GMM can provide richer information than a single one. For the second challenge, we employ Expectation-maximization (EM)
algorithm to find the maximum likelihood that the real time power consumption samples match the different models. However, at times, the power consumption differences in different states showed slight variation, which caused the maximum likelihood algorithm to readout incorrectly. Therefore, we designed a filter to eliminate as many possible error detections—this is also included in our work. The sampling rate is also considered in our algorithm; our data collection instrumentations can record power consumption data every one second, and different sampling rates samples could help us find the latent probabilistic models which is also an optional way to get more accuracy result.

With the algorithm and dynamic models, we designed a prototype to evaluate how much energy it can save. This prototype uses the same configuration to collect data but this data was uploaded to a MySQL database in real time, which allows remote access and processing. A web font user interface was developed to allow users to graphically learn the power consumption from real-time charts (watts versus time) and devices’ states generated by algorithm displayed to demonstrate the accuracy of the algorithm.

However, by only using power consumption data to recognize the states of PLDs in home entertainment system can still be improved. This system would enable home entertainment systems to become more energy efficient and further reduce energy waste by combining user behavior tracking and prediction. Finally, our goal is that recognizing the states of individual devices by monitoring the power consumption data in real time and make decision for consumer based on their habit or behavior on their PLDs’ usage.

This paper is organized in the following manner. In Chapter 1, I give a detailed description about the home entertainment system’s background and summary of on market energy management solutions with their pros and cons. In Chapter 2, a full description about our
system will be introduced. In Chapter 3, the proposed data collection configuration and test condition and procedure observed in CalPLug SIM lab will be claimed. In Chapter 4, we train Gaussian Mixture models for XBOX’s different states, apply EM algorithm to get demonstrate the performance for the algorithm. In Chapter 5, we use real time experiment and statistically determine the accuracy between projected states and PLDs’ actual states and introduce some methods to improve the accuracy. In Chapter 6, we describe conclusions and future.
Chapter 1

1. Background

Over the years, the market for plug load devices has been rapidly increasing, and along with it, the market for home entertainment systems. Almost every home in America now includes at least a TV, DVD/Blu-ray player, Set-top box, game console, or audio system. More specifically, 80% of American households with at least one game console in 2015 [5]. In U.S. the average power burned by home entertainment system in 2014 is 96 billion kWhr and shared 7% of total consumption [6]. So energy saving in home entertainment system has been a popular issue in recent years and needs to push forward to reach the more and more efficient target.

By January 2015, the 8th generation game console sales for XBOX one is 11 million and PS4 is 19 million [7, 8]. And based on earlier research, home entertainment system consumers spent an average of 6.3 hours per week on their video games, however 68% energy used by game console is during idle, and this amount represents a 10.8 TWhr power consumption or $1.24 billion which could be saved if there is an opportunity for an effectively energy management system applied [9, 10]. This wasting is so huge that most users didn’t realize that an idle device at home could waste so much money because there was limited energy information feedback to consumer directly.

Vampire power is the leaking electricity or phantom load when plug load devices are switched to off but are still plugged in to the power supply. On market energy efficiency solutions are targeting to eliminate this waste as much as possible but due to the differences in consumer’s behavior and the mobility and diversity of PLDs, it is very complicated to design global solutions for home entertainment system. There are several
solutions like advanced power strip, smart socket and etc. could effectively reduce power wasting but obviously there is still space to improve. Our system is providing a novelty method to save more energy by training PLDs’ states model and identifying PLDs’ classes and operational states.

2. On-market Energy Saving Solutions

2.1 Sleep/standby mode

Sleep/standby mode is the most common solution for PLDs in home entertainment system to save energy. Normally sleep mode is treated as a low power consumption mode compared to keeping a device fully functional, and it can reduce the reboot time for user compared to turn on devices from completely off. In home entertainment system, each individual device could save energy by switching to a sleep mode after a period of inactive time. However, vampire power consumption during the sleep mode can’t be ignored, and as mentioned above, this huge amount of waste energy has become a main drive for energy management development. And to reduce the reboot time for user, devices need to be ready to switch back to fully functional in a short time and this makes the sleep mode power consumption stays on a relatively high level. Power wasted during sleep mode is 5% of the national residential electricity consumption, and the vampire power produced 1% of the world’s carbon dioxide emissions. For example, based on our data collection trials, the average power consumption for our Set-top box (STB) fully functional is 18.4 Watts, however, after four hours inactivity, the STB automatically switched to sleep mode and consumes an average of 17.6 Watts, which shows a tiny discrepancy. So develop new
technologies that could cut off the vampire power leaking and save significant energy has been major challenges on the market.

2.2 Advanced power strips

Advanced power strips are energy-saving devices which can automatically eliminate vampire consumption of inactive PLDs. They are effectively solutions to cut off power source by monitoring PLDs’ power consumption and user activities which enable more energy saving than switch devices to sleep mode. Numerous advanced power strip have been developed and tested to monitor user activities on their PLDs which is plugged in the controlled outlets on the strip. By cutting off the power source of the controlled outlets, advanced power strip can reduce the vampire power of PLDs when they are not being used. To monitor user activities, advanced power strips are integrated with motion sensor to detect moving objects or IR sensor to detect specific frequency infrared signal from remote controls. Also, scheduled timers and logical relationship among a networked configuration are applied in advanced power strip design. A typical control procedure of an advanced power strip with IR sensor is shown below:

Plug load device which is under testing is plugged in the controlled outlet on the advanced power strip, the IR sensor is tracking 38KHz infrared signals from remote control. If there is no infrared signal received in a trigger time (could set to 1, 2, 4, or 8 hours), the advanced power strip will assume user is absent and the device should be turned off. The controlled outlet power source will be cut off and simply reduce the vampire power consumption.
From this procedure we can learn that modern advanced power strips work primarily on sensor inputs or scheduled timer without knowing devices’ states or the consumers’ usage on their devices, which inevitably cause error detection. So most advanced power strips with a sensor unit try to increase the accuracy by applying a long trigger time to cut off the power source of controlled outlets since sensors don’t receive any inputs.

To help advanced power strip save more energy, we need more accurate input about user's activities from knowing PLDs’ classes and their operational states and so a shorter trigger time is available. And we believe our system will help most advanced power strips become more energy efficient.

2.3 Smart socket with a mobile APP

With all new technologies offered to mobile devices manufactures, it has been a trend to carry a mobile device such as smart phone, tablet or laptop for both young and old. In today’s society, people are relying on mobile devices to communicate with freely accessible
information, and it makes the mobile device a perfect media to schedule and control devices in home entertainment system and help prevent devices wasting energy. A smart socket sometimes called smart plug is an intelligent cloud based system which could manage your household energy consumption and achieve home automation to help you save energy. Normally, a single smart socket has one or two outlets integrated with smart meter which could allow users to access historic global consumption data and generate periodically energy information to help user to better schedule their usage on the plugged device. It works with WI-FI, ZigBee, or any other wireless network to enable local and remote control. The outlet can cut off power source based on scheduled timer or user command to reduce energy wasting and totally eliminate the vampire power leaking. But with the increasing amount of plug loads in consumers’ house, one single smart socket could not satisfy consumers’ demand. So the concept of Internet of Things (IoT) has been introduced to smart socket market, to allow a center Hub to control all smart sockets in a consumer’s house and allow them communicate among each other. An Android or IOS app is the user interface to let users set scheduled timer or plan for their plug loads and control the power supply remotely with the help of notification that their devices are overheating or idle for a setting time.

Smart socket with an APP seems like a simple, easy to use solution to let consumers control and monitor their home entertainment systems from mobile devices. However, similar with advanced power strip we mentioned above, the monitored information from the smart meter integrated in the smart socket is only power consumption value without further analyzing. Though notification to consumer’s mobile device could help the smart socket make the right decision to save energy, it’s still a challenge for automation energy
management. And also consumer needs too many sockets as the increasing amount of plug load and it makes this solution expensively.

Our system is using a web-front user interface which integrated with our real time testing automation configuration and PLDs’ classes/operational states classification algorithm to enable future advanced power strip and smart socket to better identify users’ activities on their devices in home entertainment system to improve PLDS’ energy efficiency and reduce more wasting consumption behaviors.
Chapter 2

This section provides a description of the system. The system consists of three main components: data acquisition, analysis, and a user interface.

![System set up](image)

**Figure 2.1  System set up**

1. **Data Acquisition**

The data acquisition (DAQ) process is comprised of a power analyzer and a computer running a LabVIEW virtual instrument (VI). The power analyzer measures various types of analog parameters. The parameters measured for this experiment include the following: power consumption, power factor, RMS current, RMS voltage, signal frequency, and amplitude of the voltage. Further discussion of the power analyzer can be found in Chapter
3. Next, the LabVIEW VI is set up so that it reads measurements from the power analyzer, and then writes the measurements into an Excel spreadsheet. To implement real-time data processing, the VI is also set up to work with a Python script to simultaneously write the measurements into a MySQL database.

2. Analysis

The analysis component is primarily comprised of a MATLAB script that performs Gaussian mixture model (GMM) analyses, expectation–maximization (EM) calculations, and data filtering for error detection. The GMM analyses utilize a built-in MATLAB function that creates an object file that defines a multivariate distribution that consists of a mixture of one or more multivariate Gaussian distribution components. From the created object, a mixture model can be derived and plotted, and the mean and covariance of each of the components can be found. Next, the EM calculation is a good solution to find the maximum likelihood for a sample data belong to model data sets. This is the first step in attempting to classify device status. The next step involves the data filtering stage, which consists of procedures to ensure the prior classification attempts are not incorrect; comparisons between the sample data set’s standard deviation and the measured data sets’ standard deviations are made in order to properly classify device status. Further discussion of the GMM method and EM calculations can be found in Chapter 4.

3. User Interface

The user interface (UI) utilizes PHP to transmit real-time, server-side data to JavaScript charts and gauges. As mentioned in Section 1, the system utilizes a Python script to write
power analyzer measurements into a MySQL database. Hence, the PHP script reads the measurements from the MySQL database in real-time, encodes the measurements into JavaScript Object Notation (JSON) format, and then outputs the JSON data whenever the PHP script is called. In this experiment, the web UI utilizes JavaScript spline charts and gauges. The spline charts are updated in real-time to illustrate trends in the device’s power consumption. The gauges are utilized to numerically display the power consumption in real-time. And furthermore, based on the analysis function, real time data set will generate device’s real time state on the web UI to inform consumers their usage on home entertainment system, and also a suggestion to help them save energy.

![Dynamic Modeling](image)

**Figure 2.2  Web-Modeling UI**

The previous graph is the UI for a Dynamic Modeling web page that allows users to understand the power consumption of home entertainment devices. The interesting aspect about this UI is that the user will be able to see the amount of power that is used by any given device, and will allow users to understand how much power is used at current time.
Based on the graphs seen it seems that all devices are using the same amount of power because all the data shown is in reference specifically to Xbox power consumption. The following graph on the other hand serves as a popover, to provide even more data on the given device. As illustrated the popover is comprised of a chart showing the power consumption of the device in real time, and a gauge working as a power meter, it also shows the state that the device is in at a given time based on our analysis algorithm, and also shows how much power is used by other devices when that specific device is on. For future reference, and use, we will utilize more data collection set-up to allow us to read data for multiple devices at a given time, which will provide the user an even better understanding of the power consumption of their home entertainment system.

Figure 2.3   The detailed pop-over page
Chapter 3

1. CalPlug SIM Lab

The California Plug Load Research Center (CalPlug) at the University of California at Irvine (UCI) was established to improve energy efficiency in the use and design of appliances and consumer electronic devices. CalPlug addresses challenges in plug load efficiency for both residential and commercial buildings by collaborating closely with utilities, manufacturers, advocacy groups, research institutions, Government, and energy policy makers. We are using the CalPlug SIM (Simulation, Integration, Majority) Lab to collect first hand, unbiased energy data for plug load devices in home entertainment system.

There are two ways to collect power consumption data: field trials and lab tests. Field trials can provide more accurate data because it comes from a consumer’s home and the data relying on the consumer’s behavior in that home. However, lab test may not as such accurate as a laboratory environment may not provide real world data. Because of this, the CalPlug SIM lab simulates the typical American household living room to make sure the devices would be tested under the same conditions as they would be used by consumers, and different groups of students or faculties from UCI are invited to participate in our testing and acted just like they are at home to make sure the accuracy and universality of collected data. The SIM Lab is furnished with the five most common home entertainment appliances (TV, audio system, DVD player, Set-top box, and game console) and industry standards instrumentations. The testing procedures are from accredited organizations, such as Energy Star, IEC and CEA, and the LabVIEW software is also employed in our testing methodology. Electrical properties such as average power consumption, power
rating, power factor, and power states are characterized and reported objectively by LabVIEW every second. Deemed savings can also be tested in both laboratory and simulated environments.

Figure 3.1 CalPlug SIM LAB

2. **General Testing Conditions:**[11]

- Use of a stable power supply (<2% harmonics).
- Stable ambient test room conditions.
- Digital power meter with fundamental active power accuracy of 0.5% or better.
- Calibrate the power meter using LabVIEW.
- LabVIEW will have data logging capabilities.
- The power analyzer will sample at intervals every second.
An AC power source shall be used to provide input voltage and frequency of 115± 1% at 60 Hz to the PLD.

3. Configuration

This section includes a detailed explanation of the equipment used in our tests and how they are set up with the devices.

The break-out box and power analyzer are both connected to the AC power source. Devices are then connected to the break-out box depending on the device being tested. Information from the power analyzer is sent to the computer and compiled into a spreadsheet through LabVIEW. Those results are then uploading to the database in real time. Every second, LabVIEW could collect up to 23 different power and energy parameters from the power analyzer and compiles it into a spreadsheet that is sent to a database in real time. This data compiled by LabVIEW can be accessed in two different ways:
• In real time as the data is being sent to database
• As a fully compiled report after the test is completed

An example of the LabVIEW data is shown below:

<table>
<thead>
<tr>
<th>Vrms</th>
<th>Arms</th>
<th>Watt</th>
<th>VA</th>
<th>VAr</th>
<th>Freq</th>
<th>PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.20E+02</td>
<td>1.0721</td>
<td>7.05E+01</td>
<td>1.28E+02</td>
<td>1.07E+02</td>
<td>6.00E+01</td>
<td>5.49E-01</td>
</tr>
<tr>
<td>1.20E+02</td>
<td>1.0719</td>
<td>7.05E+01</td>
<td>1.28E+02</td>
<td>1.07E+02</td>
<td>6.00E+01</td>
<td>5.50E-01</td>
</tr>
<tr>
<td>1.20E+02</td>
<td>1.072</td>
<td>7.05E+01</td>
<td>1.28E+02</td>
<td>1.07E+02</td>
<td>6.00E+01</td>
<td>5.49E-01</td>
</tr>
<tr>
<td>1.20E+02</td>
<td>1.0727</td>
<td>7.06E+01</td>
<td>1.28E+02</td>
<td>1.07E+02</td>
<td>6.00E+01</td>
<td>5.49E-01</td>
</tr>
</tbody>
</table>

The specific models listed are the ones we chose for our tests; actual results might vary with different choices.

**Audio System:** Onkyo HT-S3500 5.1-Channel Home Theater Speaker/Receiver

**DVD Player:** Sony DVPSR510H DVD Player (Upscaling)

**Game Console:** Microsoft Xbox 360

**Television:** VIZIO E390i-A1 39-Inch 1080p 120Hz Smart LED HDTV

**Set-Top Box:** DirecTV HR44

**AC Power Source:** Chroma Programmable AC Source Model 61601

**Break-Out Box:** Voltech Universal Break-Out Box

**Power Analyzer:** Voltech PM1000+ Power Analyzer

4. **General Testing Procedures:**

The testing procedure for testing the multiple states of one device (active, idle, main menu, etc.) is as follows:

1) Set up electrical equipment as depicted in the figure above.
2) Connect the desired testing device to the break-out box.

3) Volunteers are invited to the SIM living room to test the devices. In order to accurately simulate normal household behavior, the volunteers are encouraged to do whatever they would normally do when using the devices at their homes.

4) Tests span 0.5-2 hours to preserve accuracy and collect data on all states of the devices. One engineer will also gather user behavior data on the volunteers, noting, for example, when users are disengaged but the Xbox is still left active and record the moment when users changed devices’ states.

5) Data is sent from the power analyzer and compiled and stored by LabVIEW into the spreadsheet, the engineer noted user behavior will generate another report about the timeline indicates the time range of different states.

6) At the end of the tests, a questionnaire is given to the participant to collect data on user behavior, experience and improvement.

5. Sample Test Results

The testing results span a total of over 100 hours in two month. Over 10 UCI students are participated in our tests and the total data was classified based on its category. We then use the classified data to train our dynamic models for devices under different states. In this thesis, we focus on XBOX dynamic modeling as it has more states than STB, TV, DVD and audio system. However, the method we are using to train the model can be applied equally for all devices. Examples of the MATLAB power consumption data for the various Xbox power states are shown below:
Figure 3.2 is the power consumption for user play “Assassin’s Creed”, this set of data will be used to train the model “Playing game”.

Figure 3.3  Power consumption for playing game

Figure 3.4  power consumption for Game Idle
Figure 3.3 is the power consumption for “Assassin’s Creed” Idle, user was absent for 30 minutes. This set of data will be used to train the model “Game Idle”. It has an obviously drop in this curve because for this game after 15 minutes idle time, the state has changed a little but will keep on this state for the rest of time.

Figure 3.4 is the power consumption for XBOX stays on main menu, this state is very important because it’s the watershed state for XBOX, to change state among different games, play DVD, off to on and play music, XBOX has to change to main menu first.

Figure 3.5 Power consumption for XBOX stay on main menu
Figure 3.6  Power consumption for streaming video from DVD

Figure 3.5 is the power consumption when streaming DVD using an XBOX, it has been a trend that people use XBOX to play DVD instead of a DVD player.

Figure 3.7  Power consumption for streaming music from CD
Figure 3.6 is the power consumption for streaming music from CD. From these five samples, we have seen that different states have overlap power consumption values, so it’s not possible to make a distinction just based on the power consumption values. In next chapter, we will introduce the different kinds of distributions for these models and the maximum likelihood for a specific data match these models. Five models will be learned respectively to demonstrate the accuracy for our methods:

- Model 0: Play Game
- Model 1: Game Idle
- Model 2: Main menu
- Model 3: Play DVD
- Model 4: Play Music
Chapter 4

1. Density Histogram

From last Chapter we already known that just based on the range of power consumption for different states, it's not possible to identify the operational states. So we are going to use probability theory to solve this problem. The input for this processing is the raw data we collected by our testing instrumentations, and the output should be different states for XBOX. However, the raw data we collected is just a time-sequential record of power factors. We have to do some preprocessing to make the raw data useful, in our task, raw data has to transform to valuable data form that indicates features for different states so that we could characterize theses states' signature and then build the algorithm for classification.

Density histogram, also called histogram was first introduced by Karl Pearson, is a graphical representation of the distribution of numerical data[12]. It is an estimate of the probability distribution of a continuous variable and in this thesis paper the continuous variable is the range of power consumption value. To construct a histogram, for example, the histogram for XBOX Model 0, the first step is to combine all XBOX Model 0 data together, then divide the entire range of power consumption values into a series of 0.1 watt intervals, this series of bins will be the X-axis for this histogram and the Y-axis could be in two formats, we can count how many values fall into each bin or calculate the frequency/percentage. In our research, we use the percentage density histogram to find the features of the power consumption for different states. And the mathematic definition of percentage density histograms could be defined as:

$$\sum_{i=1}^{k} \frac{m_i}{n} = 1$$
In this equation, \( n \) is the total number of power consumption data and \( k \) is the total number of bins, \( \frac{m_i}{n} \) is the percentage histogram and the sum of them will equal to 1.

Figure 4.1   Percentage Histogram for XBOX Model 0
Figure 4.2 Percentage Histogram for XBOX Model 1

Figure 4.3 Percentage Histogram for XBOX Model 2

Figure 4.4 Percentage Histogram for XBOX Model 3
From above histograms, we can find that Model 3 Play DVD has unique feature that make it easy to classify. However, for Model 0 Play game and Model 1 Game Idle, it has overlaps and also Model 2 Main menu and Model 4 Play Music. So we need to find a classifier that could use the raw data as an input and make the decision which model it is based on information provided by histograms.

2. Model Selection And Training

2.1 Bayesian Probabilities

The percentage density histograms were created by finite data collected in CalPlug SIM Lab, they show the characteristic of different states for XBOX. However, in the field of model classification, a general method will be more helpful. Thomas Bayes’ theory, the Bayesian classification, will be helpful to classify power consumption samples based on probability
theory and could help us find the maximum probability option among all these models. The Bayes’ formula is known as [13]:

$$P(F_j \mid E) = \frac{P(EF_j)}{P(E)} = \frac{P(E \mid F_j)P(F_j)}{\sum_{i=1}^{n} P(E \mid F_i)P(F_i)}$$

This formula will help us to find the maximum likelihood Model for different samples. Assume we have \( j \) states \( F_j \) and \( E \) is the sample power consumption set of unknown states. So \( P(F_j \mid E) \) is the probability that given sample \( E \) belong to state \( F_j \) which also called as posterior probabilities for Bayesian Classification. \( P(E \mid F_j) \) is the probability density function (PDF) of state \( F_j \) in the sampled power consumption range which is also called “likelihood function”. And we can get a value from the frequency histograms but these histograms are modeled by finite tests and if we want to more general results, we need to train new models for infinite values. The probability \( P(F_i) \) is called “priori probability”, which means the probability of the state before knowing the sample. In this research, this probability could represent as percentage time people spend on each states, but it’s not a simple situation, this issue is so complicated that we will treat each state has the same priori probability, so that \( P(F_j) = \frac{1}{j} \). The denominator in formula equation is the normalization constant, which ensures that the posterior distribution on the left-hand side is a valid probability density and normally integrates to one. In our case, we could use \( \sum_{i=1}^{j} P(E \mid F_i)P(F_i) \) to calculate this constant as a scaling factor. So the Bayes’ theory can be stated in words [14]:
The highest posterior probability makes the decision about which model the sample belong to. But the risk of error detection will always exist, and minimize the error risk will be our guideline to select the model. And the major problem is to find the likelihood function \( P(E | F_i) \) which could tell the distribution of sample data set and find the highest posterior probability describes the XBOX model based on the Bayes’ formula. We started from Gaussian distribution.

2.2 Gaussian Distribution

Gaussian distribution is the most important probability distribution for continuous variables. We define that \( X \) is a Gaussian distribution or simply that \( X \) is normally distributed, with parameters \( \mu \) and \( \sigma^2 \) if the density of \( X \) is given by[13]:

\[
f(x) = \frac{1}{\sqrt{2\pi \sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}
\]

The parameter \( \mu \) and \( \sigma^2 \) of a Gaussian distribution are called mean and variance, and \( \mu \) also represents the expected value.

With the help of MATLAB, we apply Gaussian distributions on our sample sets and compare them with the histogram as mentioned in section 1. Gaussian distribution could effectively help us to find the likelihood needed for Bayesian classification. An example is shown below, this data set was collected for XBOX play DVD and it shows very similar feature with Gaussian distribution, and we can find the frequency for different power consumption value. Suppose that we have a data set of another testing for XBOX play DVD

\[ S = \{ p_1, p_2, p_3, \ldots, p_n \}; \quad p_1, p_2, p_3, \ldots, p_n \text{ represent } n \text{ power consumption data value and we can find the } f(p) \text{ based on the Gaussian distribution formula. Notice } p_1, p_2, p_3, \ldots, p_n \text{ are } \]
independent and identically distributed on the Gaussian distribution in Figure 4.6, so the joint probability of sample $S$ can be calculated by the product of the marginal probabilities for each event separately[13]. So the likelihood function we are looking for is given below:

$$p(S) = \prod_{i=1}^{n} f(p_i)$$

Figure 4.6  Gaussian distribution works good for one sample data set

However as discussed before, our goal is to use finite data set to predict the feature for infinite data. This 2-hour data set demonstrate Gaussian distribution could be a solution for us to find the likelihood function however, when we apply Gaussian distribution to Figure 4.1 to Figure 4.5 in Section 1, we have the following Gaussian distributions, these Gaussian distributions are showing partial information as power consumption range limited:
Figure 4.7  Gaussian Distribution for XBOX Model 0

Figure 4.8  Gaussian Distribution for XBOX Model 1

Figure 4.9  Gaussian Distribution for XBOX Model 2

Figure 4.10  Gaussian Distribution for XBOX Model 3
Compare these distributions with percentage histograms, Gaussian distribution works fine for Models with less features, but for an example Model 1 has two obviously features in two ranges of the power consumption value, by using Gaussian distribution, the likelihood function will have severe error results and it will be too complicate to distinct Model 0 and Modle1. So Gaussian Mixture distribution was considered as a compensatory method to provide richer information than single Gaussian distribution, we will discuss the detail in next part.

2.3 Gaussian Mixture distribution

Though Gaussian distribution provides several properties for our XBOX models, it has too many limitations when the Model has complex features and it can result an error detection for a real-time input data set. For example in Figure 4.2 the histogram for XBOX game Idle Model, it has two dominant ranges, and single Gaussian distribution couldn’t represent all features for this model. But if we use two Gaussian distributions for each dominant range in Figure 4.12, we will have a better characterization of the data set.
Compare with Figure 4.8, using the superposition of two Gaussian distributions provides us more accurate information. Such superpositions, formed by taking linear combinations of more basic distributions such as Gaussians, can be formulated as probabilistic models known as mixture distributions. By tying different number of Gaussians, and by adjusting their means and variances as well as the coefficients in the linear combination, almost any continuous density can be approximated to arbitrary models [14].

The superposition of $K$ Gaussian distribution can be defined by this form:

$$p(x) = \sum_{k=1}^{K} a_k f(x_k)$$

$f(x_k)$ is the single Gaussian distribution with mean $\mu_k$ and variance $\sigma_k^2$ and always called as “component”. The parameter $a_k$ is the mixing coefficients, normally $0 \leq a_k \leq 1$.

Figure 4.13 ~ Figure 4.17 are the final Models trained by Gaussian mixture distributions with the help of MATLAB:
Figure 4.13  Gaussian Mixure Model for XBOX Model 0

Figure 4.14  Gaussian Mixure Model for XBOX Model 1
Figure 4.15  Gaussian Mixture Model for XBOX Model 2

Figure 4.16  Gaussian Mixture Model for XBOX Model 3
By reading these figures we immediately see that using Gaussian Mixture Model, we can get richer information about XBOX different states. In our thesis, we decided to use Gaussian Mixture distribution to train Models for devices’ states in home entertainment system. For Gaussian Mixture distribution, we have known that it’s the superposition of single Gaussian distributions, but the likelihood function is not simply a summation. In some books, the log of the likelihood function is given by [14]:

\[
\ln p(a, \mu, \sigma) = \sum_{n=1}^{N} \ln \left( \sum_{k=1}^{K} a_k f(x_k) \right)
\]

This likelihood is so complicated than single Gaussian’s. It has \( K \) summations inside the logarithm. Also MATLAB can help us easily calculate the likelihood function, we prefer to employ a general expectation maximization algorithm to find the likelihood function which will discuss in next section.
3. EM Algorithm For Gaussian Mixture Model

Given a Gaussian mixture model, there is a standard procedure to maximize the likelihood function:

1) Initialize the means $\mu_k$ and variances $\sigma_k^2$ and mixing coefficients $a_k$, and evaluate the initial value of the log likelihood.

2) E STEP. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{a_k f(x_n)}{\sum_{j=1}^{K} a_j f(x_j)}$$

3) M STEP. Re-estimate the parameters using the current responsibilities

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$
$$\sigma_k^{2 new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$
$$a_k^{new} = \frac{N_k}{N}$$

Where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

The sample data are written as vector format.

4) Evaluate the log likelihood

$$\ln p(a, \mu, \sigma) = \sum_{n=1}^{N} \ln \left( \sum_{k=1}^{K} a_k f(x_k) \right)$$

And check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied, return to step 2[14].

By using MATLAB this process could be easily finished.
By now we have the Gaussian Mixture Model and EM algorithm to find the likelihood function, and also assumption for prior probability, recall the Bayesian classification:

\[ \text{posterior} \propto \text{likelihood} \times \text{prior} \]

We have a complete progress to find the posterior probability to help us make the decision. With help of MATLAB script, we will show the result in next section.

*In this section, the procedure for EM algorithm is introduced in *PATTERN RECOGNITION AND MACHINE LEARNING* by Christopher M. Bishop, P438.

4. Result And Accuracy Demonstration

In this section we will apply Gaussian Mixture and EM algorithm to different samples. Each sample is collected when XBOX is under unique Model. The following tables should give the accuracy when the algorithm applied to play game, game idle, main menu, play DVD and play music samples.

<table>
<thead>
<tr>
<th>Table 4.1 Result for XBOX Model 0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model 0</strong></td>
</tr>
<tr>
<td>Sample data quantity</td>
</tr>
<tr>
<td>Sample rate</td>
</tr>
<tr>
<td>Accuracy</td>
</tr>
<tr>
<td>Error detection type</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4.2 Result for XBOX Model 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model 1</strong></td>
</tr>
<tr>
<td>Sample data quantity</td>
</tr>
<tr>
<td>Sample rate</td>
</tr>
<tr>
<td>Accuracy</td>
</tr>
</tbody>
</table>
### Table 4.3  Result for XBOX Model 2

<table>
<thead>
<tr>
<th>Error detection type</th>
<th>40/667 are Model 0(Playing game)</th>
</tr>
</thead>
</table>

#### Model 2  
- **XBOX Main menu**
- **Sample data quantity**: 600
- **Sample rate**: 5
- **Accuracy**: $\frac{118}{120} = 98.3\%$
- **Error detection type**: 2/120 are Model 4(Playing Music)

### Table 4.4  Result for XBOX Model 3

<table>
<thead>
<tr>
<th>Error detection type</th>
<th>N/A</th>
</tr>
</thead>
</table>

#### Model 3  
- **XBOX Play DVD**
- **Sample data quantity**: 9478
- **Sample rate**: 5
- **Accuracy**: $\frac{1895}{1895} = 100\%$
- **Error detection type**: N/A

### Table 4.5  Result for XBOX Model 4

<table>
<thead>
<tr>
<th>Error detection type</th>
<th>180/900 are Model 2(Main Menu)</th>
</tr>
</thead>
</table>

#### Model 4  
- **XBOX Play Music**
- **Sample data quantity**: 4500
- **Sample rate**: 5
- **Accuracy**: $\frac{718}{900} = 80\%$

### 5. Conclusion

Based on section 4 we have the conclusion that Gaussian Mixture Model and EM algorithm work pretty good for steady state samples. For Model Playing Game and Model Game Idle, from the histogram and Gaussian mixture distribution we have discussed that they have overlap so majority error detection for Model 0 and 1 are always each other. So the most
essential reason for error detection is that the distribution couldn’t distinct the overlap. In next Chapter when we use the real time multi-models sample, this issue will become even worse. We will introduce some methods to reduce error detection in next Chapter, the idea is we need to learn the error detection rules and then design proper filter algorithm to eliminate error detection as many as possible.
Chapter 5

1. Real Time Sample Demonstration

In last Chapter, we have demonstrated the accuracy when the algorithm applied to specified states and it shows very good performance. Now we process some real time samples include almost all states in a long period to see how well the algorithm can work for much more complex situation.

Table 5.1  Real time sample

<table>
<thead>
<tr>
<th>Period (second)</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ~ 80</td>
<td>Main Menu</td>
</tr>
<tr>
<td>81 ~2010</td>
<td>Play Game</td>
</tr>
<tr>
<td>2011 ~3135</td>
<td>Game Idle</td>
</tr>
<tr>
<td>3136 ~ 3365</td>
<td>Main Menu</td>
</tr>
<tr>
<td>3366 ~4175</td>
<td>Play DVD</td>
</tr>
<tr>
<td>4176 ~ 4200</td>
<td>Main Menu</td>
</tr>
<tr>
<td>4201 ~5105</td>
<td>Play Music</td>
</tr>
<tr>
<td>5106 ~ 5725</td>
<td>Main Menu</td>
</tr>
<tr>
<td>5726 ~ 7800</td>
<td>Play Game</td>
</tr>
</tbody>
</table>

This sample covered all states of XBOX and Table 5.1 records its timeline for each state. Apply the classification algorithm to this real time sample, we get the result below:

Table 5.2  Result for real time sample

<table>
<thead>
<tr>
<th>States</th>
<th>Accuracy</th>
<th>Main Error type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>N/A</td>
<td>Play Music</td>
</tr>
<tr>
<td>Play Game</td>
<td>80%</td>
<td>Game Idle/Play Music</td>
</tr>
<tr>
<td>Game Idle</td>
<td>70%</td>
<td>Play Game</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>------------------</td>
</tr>
<tr>
<td>Play DVD</td>
<td>95%</td>
<td>Change State</td>
</tr>
<tr>
<td>Play Music</td>
<td>75%</td>
<td>Main Menu</td>
</tr>
</tbody>
</table>

From Table 5.2 we immediately find that the result is not as accurate as steady state. The reason is that compared with steady state, real time data has a very complex extra situation, change states between two steady states. Actually most error detection occur during the short time for changing state, and Modeling this situation is a too complex issue and we leave it for further study. And the other reason is still the overlap among different states have similar power consumption. And we have a basic assumption that XBOX needs to change to Main Menu state first before it changes states. So we have problem to classify main menu because user always stays on main menu for a very short time, and prefer to changing to another state, it makes the main menu state becomes the boundary for changing states and hard to detect. And we need to consider this and train new model for it. To reduce the error detection, several properties for different states are studied and will help to build a filter in next section.

2. **Filter Design**

Based on Figure 3.2 and Figure 3.3, we found that when XBOX is under play game Model, the power consumption has a wider range, more complex changes and bigger standard deviation, however, when game is idle the power consumption stays on a relatively stable level and the range and standard deviation are much smaller. This property will help us design the filter to distinct Play Game and Game Idle Models.
Based on Table 5.1 and our testing, the first rule is if you don’t change your state to Main Menu, you can’t change state. In the classification results, a lot errors are occurred during one specific state and we can eliminate other error states because without changing back to Main Menu, these error states can’t be there. Also user shouldn’t get game idle state if there is not play game model detected in the previously classification. These logical relegations will help us reduce a lot error detections.

And sample rates are also considered in our filter design, different sample rates can improve the algorithm efficiency and could be a reference to make the right identification for states. We use sample rates 5, 15 and 60 to generate 3 results and compared them to help improve the accuracy. As a result, the filtered result shows:

<table>
<thead>
<tr>
<th>States</th>
<th>Accuracy</th>
<th>Main Error type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Menu</td>
<td>70%</td>
<td>Play Music</td>
</tr>
<tr>
<td>Play Game</td>
<td>90%</td>
<td>Play Music</td>
</tr>
<tr>
<td>Game Idle</td>
<td>85%</td>
<td>Play Game</td>
</tr>
<tr>
<td>Play DVD</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>Play Music</td>
<td>85%</td>
<td>Main Menu</td>
</tr>
</tbody>
</table>

For Main Menu and Play Music error detection, to help improve the accuracy in real time sample, standard deviation and frequency domain properties are learned. However, we still need a lot times to learn the change states power consumption profile and train it as a new model because when the state Main Menu and Play Music steady, we can easily identify them from the real time data.
3. Different Games Study

Identify different games is such a complicate issue because there are so many games on market and each of them has a totally different power consumption distribution. It’s not possible for us to train the model for every game but we have learned the loading time for several games and get an immaturity theory that same game has similar loading power consumption distribution and different games have distinct loading power consumption distribution. This issue will study in the future.

Figure 5.1  Loading power consumption feature for Game 1

Figure 5.2  Another Loading power consumption feature for Game 1
1. Conclusion

In previous Chapters, we have introduced the system, the data Acquisition configuration, Gaussian Mixture Model and EM algorithm, and also the result. Though our system has a good performance on XBOX states identification, we still need a lot works to improve it and may finally put it on market.

Set-top box, DVD, audio system and TV can also use the systemic process to classify their states and use the results in home entertainment system energy efficient. As we know, 68% power consumption was wasted during home entertainment systems idle model. So by applying this method we could help consumer to get immediately power consumption data and devices’ states to help them save energy. As we could detect devices’ idle states, so we could cut off the power source after a shorter trigger time. In our system, we set it to 15 minutes.

If modern APS could be integrated with our algorithm, it can significantly reduce its trigger time and has a more accurate and effective way to detect user activities. And eventually our system could help modern home energy management system to reach more energy saving goal.

2. Future Work

The future work will include:

1) Train Model for XBOX change states.
2) Distinct different games
3) Learn the logical relationships of networked systems

4) Track user behavior on their devices to find the prior probability in Chapter 4

5) Modeling user behavior to help building an intelligent algorithm

6) Run more tests to make the Gaussian Mixture Model cover more situations and could provide more accurate information
Reference

