6-1-2016

Energy Management Control Center

Francis Estacio
Santa Clara University

Ren Hirokawa
Santa Clara University

Devon Quaternik
Santa Clara University

Matthew Salmanpour
Santa Clara University

SANTA CLARA UNIVERSITY
Department of Electrical Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED
UNDER MY SUPERVISION BY

Francis Estacio, Ren Hirokawa, Devon Quaternik, Matthew Salmanpour

ENTITLED

ENERGY MANAGEMENT CONTROL CENTER

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

BACHELOR OF SCIENCE
IN
ELECTRICAL ENGINEERING

6/1/2016

Thesis Advisor ______________________ Date

6/1/2016

Department Chair ______________________ Date

ENERGY MANAGEMENT CONTROL CENTER

By

Francis Estacio, Ren Hirokawa, Devon Quaternik, Matthew Salmanpour

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Electrical Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Electrical Engineering

Santa Clara, California

2016

Energy Management Control Center

Francis Estacio, Ren Hirokawa, Devon Quaternik, Matthew Salmanpour

Department of Electrical Engineering
Santa Clara University
2016

ABSTRACT

Home energy management is a booming market and is a problem that a lot of people are working on all across the world. The “Energy Management Control Center,” henceforth referred to as EMC^2, is a project built upon the Smart Thermostat Senior Design project that was completed last year. We took the smart thermostat system and integrated it with an AC unit through a central hub. We will take this concept and extend it to other electrical appliances, essentially creating a home hub that can intelligently control different home appliances.

Keywords: Fuzzy Logic Control, Smart House, Energy Efficiency, Internet of Things

Acknowledgments

The authors would like to thank the Santa Clara School of Engineering for its support in providing all software and equipment used in development and testing as well as funding to support our project. We would also like to thank the Smart Thermostat group from last year for providing their software and support.

They would especially like to thank Dr. Maryam Khanbaghi for all of her guidance and support with throughout the project.

Contents

1.0 Introduction

1.1 Problem Statement

This senior design project is focused on reducing high energy bills for low-income homes. The goal is to afford families with peace of mind due to a fully automated system. A control system was designed to manage and optimize energy use of appliances without sacrificing comfort within the home. The design is centered on the use of fuzzy logic to optimize energy use.

1.2 Project Background and Motivation

The Senior Design Project is something that engineering students very much look forward to as they approach their Senior Year. When it comes time to choose which actual project to work on, the electrical engineers observe presentations on numerous projects in their Junior Design class. As it turned out, the members of our group were all interested in the Smart Thermostat design, a project advised by Dr. Maryam Khanbaghi the previous year. The idea of integrating home life and comfort with concepts we have learned in electrical engineering, especially in control systems, was very appealing to us because it seems like something of great value to society today. The group wanted to take the project further than the Smart Thermostat went last year by expanding it to the whole home.

The group considered simply improving the Smart Thermostat, but the idea of having our own project and expanding energy efficiency to the whole home is something that was more appealing to use from the beginning. After a lot of brainstorming with our advisor, the group developed the idea of a control center that would manage the power loads of different appliances in the home and distribute energy based on the budget and comfort of the user. A simulation based project was agreed upon, as an actual hardware build would not fit within the scope of this project within the given timeline of a school year.

1.3 Existing Market

The Home Energy Management System (HEMS) is any product or service that monitors, controls, or analyzes energy in the home. Utility industries today are working toward cheaper ways to connect customers to a smart grid using a HEMS. Around

2008 and 2009, the HEMS started to hit the market. For the next few years, the growth of HEMS will result in a number of company acquisitions which will lead to millions of homes subscribing to HEMS services and product offerings that are more affordable and desirable for consumers.

From our research, there are three products today that are working on HEMS services and products that we found to be most applicable to what the project. The first and most popular of the three is the Nest Thermostat. It is a smart user-based thermostat system that learns user's behavior through a machine learning algorithm and adjusts the temperature from that data. The second is the VeraEdge Smarter Home Control which is a one-app control for cameras, door locks, sensors, thermostats and other home automation systems. It is priced for around $99.00 and uses a Wi-Fi connection to communicate. The third is the Insteon Connected Home Automation Kit which gives remote control access to the home from anywhere.

There is one service today that is particularly relevant to our project. The Wiser Home Management is a HEMS that puts the control of a home's energy into a device. It has access to real-time home energy output information through a smart thermostat, in-home display unit, computer, or smart device. This service is very similar to our project which monitors the energy and temperature of a home and finds the best way to spend energy to promote savings without sacrificing comfort to a home. The main difference, however, is that our project strives for a low cost implementation.

2.0 Design Considerations

2.1 System Level Requirements

To create this solution, certain requirements had to be set up. HVAC and appliance use were manipulated and controlled to reduce home energy bills. The specification we set up for this was to reduce the home energy bill by 5%. No home energy management system would be effective if it didn't take into account user comfort, and so the specification was also set up for our system to keep the room within +/- 5 degrees of the user selected temperature. The last specification set up was to have the user have complete control of everything going on in the home at all times, so an override command was implemented that trumped all other control suggestions.

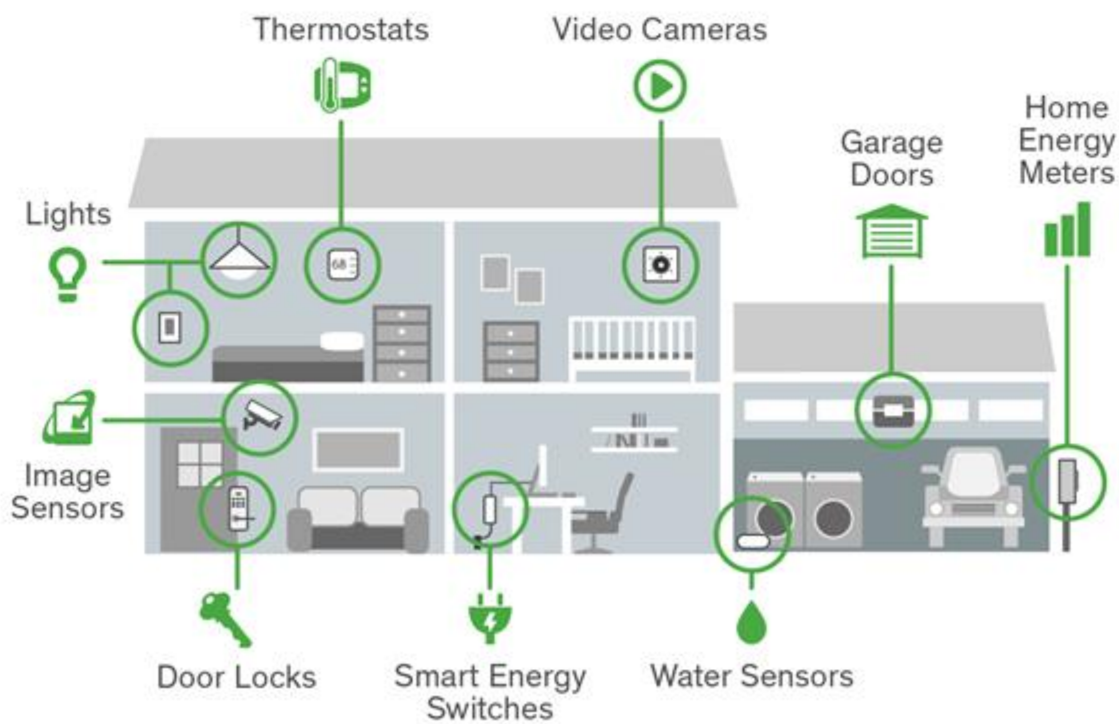

Figure 1: Home Energy Management System (HEMS) integrates technologies throughout the home that serve different functions, to effectively manage energy consumption by residents in an intelligent manner.

2.2 Customer Needs

Home energy management is a problem that a lot of people all over the world are tackling right now. Because of that, there are a lot of potential customers for the system, all of whom have very different homes. As can be seen from Figure 2, there are so many different variables in a home that could vary from home to home. This makes the customer needs be very broad and varied. So, to satisfy the customer needs, the system needs to be very adaptable and modular. It accomplishes this by the Smart Plug feature, which would go in between any appliance and the wall outlet, which would allow the system to take into account power consumption by any type of appliance. For more information about Smart Plugs, see Section 5.5, and Section 7.

2.3 Design Solution

The EMC² system is designed around manipulating the HVAC system and appliance use in order to reduce power consumption during peak electricity rate times as seen in Figure 1. This ensures consumption is higher in the off peak and partial peak points. As a result, the electricity bill per month is reduced. Also, the system prioritizes user comfort, while maximizing savings.

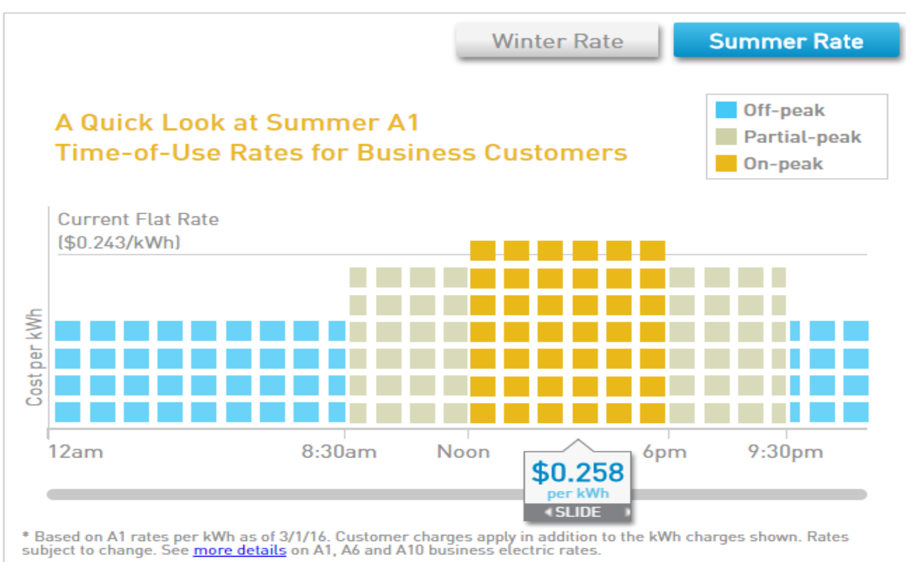

Figure 2: Time-Of-Use Rates taken from the PG&E website that detail different different electricity rates at different times of the day. These rates are set by energy demand at that time.

Source: http://www.pge.com/en/mybusiness/rates/tvp/toupricing.page?WT.mc_id=Vanity_TOU

2.4 Physical Components

Although we focused mainly on the simulation for this project, we did do research on possible hardware components to be used if there was time for a physical build or if a future Senior Design Group wanted to continue this project. Some physical components that we researched about is a Raspberry Pi 2 to download our main controller onto. This would serve as the central hub to our control strategy. Some other hardware that we need is at least 5 if not more of the Satechi IQ Smart Plug. These Smart Plugs will serve as the connection between our central hub and the appliances to monitor power consumption. Sensors that would need to be used are the DS18B20 Thermometer for temperature sensing purposes both in the home and outside of the home, and the MG-811 CO_2 Sensor and multiple HC-SR501 Motion Sensors, to have the presence of CO_2 and Motion in the home act as checks for people actually being home. Physical relays would also be needed to turn and off different components of our system, such as the Smart Plugs or sensors when they are not necessary. Relays we looked at were the JQX-15F Magnetic Relay and the S108T02 Solid State Relay.

3.0 Fuzzy Logic Control

Fuzzy Logic is what is used in EM2 as the main control strategy. It uses IF-THEN statements to make rules based on common sense. These rules form the knowledge base you can see at the top of Figure 3. The fuzzy logic will take information from the process, the models in this Simulink system, and inputs it into the fuzzification interface. There the information is turned into words to describe the state. It then takes this and feeds it to the inference engine, which looks at the rules already set up and decides on an action, in words, to take. It outputs this to the defuzzification interfaces, which takes the words and turns it into a number a computer can use which is then fed to a non-fuzzy controller to create an action in the process.

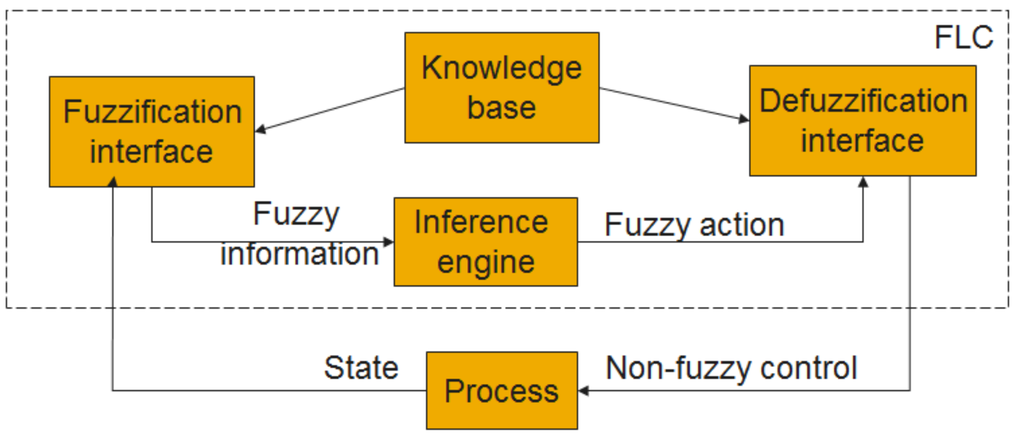

Figure 3: This figure shows a high level overview of the Fuzzy Logic Controller and its actions. Everything in the dotted line is part of the fuzzy logic.

3.1 How Fuzzification and Defuzzification work

Fuzzification and defuzzification are accomplished using what are known as membership functions. Figure 4, below, shows the Time of Day input membership function to the EMC^2 system. Weights are used to decide the significance of a membership function and come into play with the overlap of the functions.

$$C_m = \frac{\sum_{i=1}^{m} \mu_z(x_i)x_i}{\sum_{i=1}^{m} \mu_z(x_i)}$$

Figure 4: The equation for a membership function is shown in this figure.

The functions here are a mirror of the PG&E graph shown earlier in Figure 2. They divide the times into off-, partial-, and peak times. There are multiple functions for creating the shape of a membership function, those here are based on gauss2, but there are trimf, zmf, and others. The shape decides how quickly things change from one zone to the next. The lack of overlap between off-peak and partial ensures the system is not turning appliances on too late in the morning or too early in the evening to help save on costs. There is a lot of overlap with partial-peak and peak times to the same effect. The equation in Figure 4 shows how fuzzification and defuzzification creates or changes a number into words. □represents the weight of the membership function and □is the place you are on the x-axis. If you are not in a membership function, □will be zero on. This creates a weighted average of all the membership functions, taking into account for the previously mentioned overlap. There are a couple of other methods for creating this number, but weighted average allows for simple weighting of the membership functions.

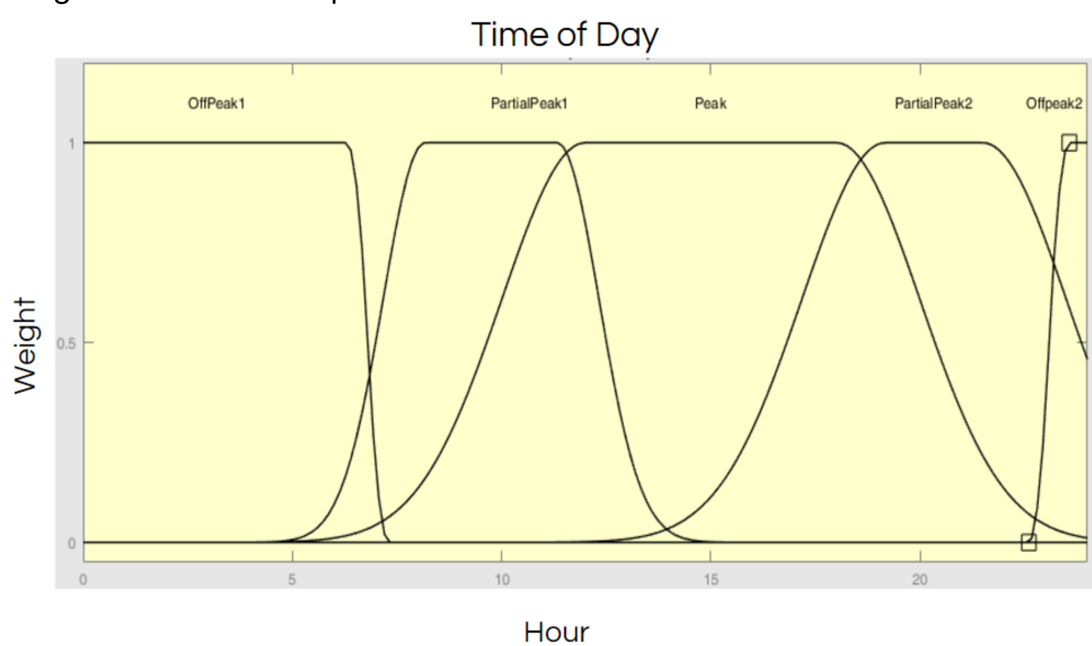

Figure 5: Examples of membership functions are displayed in this figure.

3.2 Rules in Fuzzy Logic

An example of a rule used by EMC² fuzzy logic is shown below in Figure 6. It gives a typical case that has already gone through fuzzification. A visual of the circumstances can be seen below in Figure 7. The red line indicates where in the membership functions the value lies and how filled the graphs are indicating the weight. The yellow is input and blue is output. For Output Command and Thermostat outputs weight does not matter as they are either on or off.

If (PowerUsage is High) and (BudgetGiven is Strict) and (TimeOfDay is PartialPeak) and (Override is OFF) then (OutputCommand is OFF) (Suggestion is Can)(Thermostat is Add1)

Figure 6: 1 of 168 rules in our Fuzzy Logic Controller is displayed here.

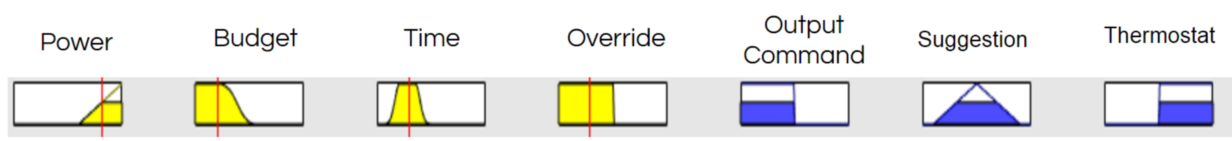

Figure 7: The states of different inputs for a particular rule are displayed here.

You must create a certain number of rules based on the inputs and number of membership functions within each input. The number of rules grows exponentially for number of inputs and linearly for number of cases. The system had a total of 168 rules spread across the smart thermostat as well as EMC² fuzzy logic, which can be found in code form in appendix B. All of these rules allowed the system to intelligently save energy and cost.

3.3 Application of Fuzzy Logic Control

The system uses fuzzy logic to mimic human intuition. With each rule the decision was made of how someone would be comfortable in a certain situation. By following the minimum rule equation, the system covers all cases that could be considered from given inputs. Through this, the system gained intelligence and was able to make inferences similar to that of a person trying to save money throughout the day. There are four inputs to the system. The first is a power input, which was taken in as watts. Next is budget given, a sliding scale from 1 to 10 that corresponds to a percentage amount of savings. A '1' is a strict budget and corresponds to approximately 4%

savings. A ‘10’ is flexible and will be less than 1% savings. Time of day is the third, and the clock is used to determine the peak times. Lastly, override gives the user full control of EMC^2. The outputs are fewer with only 3. An output command, used to hold off appliances or turn them on, in this system it was specific only to the dishwasher, but can be used for other cyclic appliances as well. Second, a suggestion for the user reaching for the override so they know whether or not it is a good time to turn things on. For the last one, there is the thermostat output that changes the input temperature to the smart thermostat proactively, rather than reactively, to turn on or off the HVAC when appliances are adding heat into the home.

4.0 Simulink

The entire project was created through Matlab's Simulink program. It consists of many blocks that simulate the power and heat of a 10 x 20 x 8 house and it also consists of two fuzzy logic blocks representing EMC^2 and the smart thermostat. It is made up of two major sections: EMC^2 and the smart thermostat. The smart thermostat is an input to the EMC^2 .

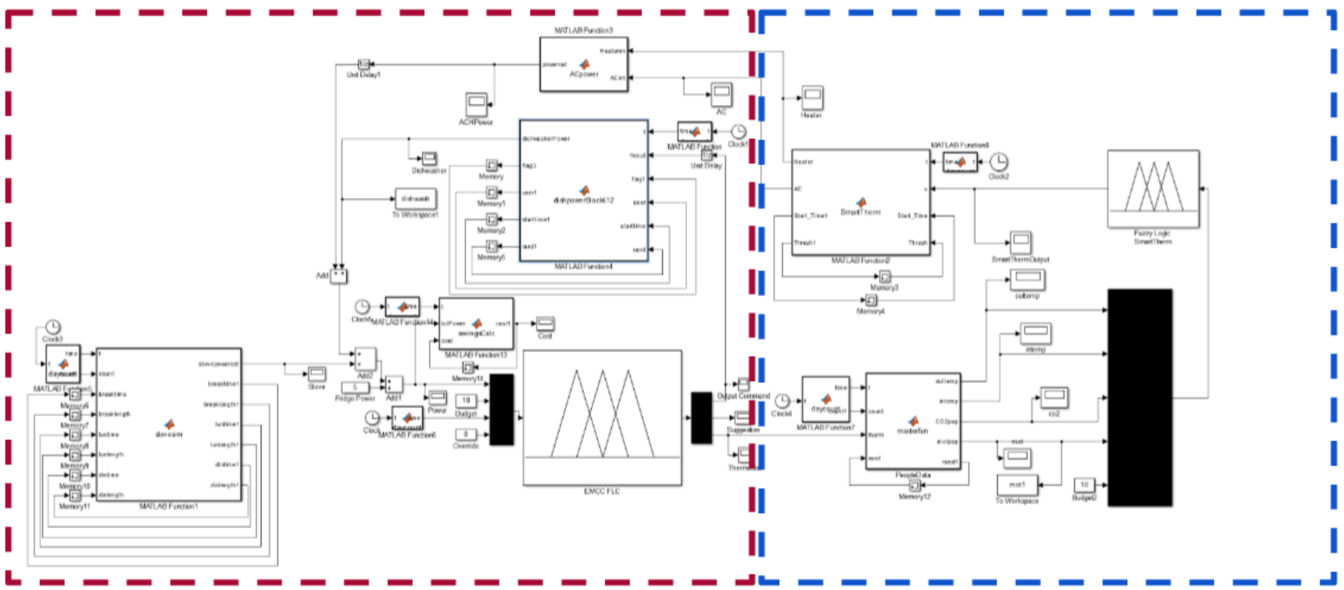

Figure 8: Entire Simulink diagram of EMC^2 and Smart Thermostat.

4.1 Simulink Model - Energy Management Control Center

EMC^2 is broken up into three major sections: the fuzzy logic controller, uncontrolled power uses, and controlled power uses. The uncontrolled power uses are power uses that the fuzzy logic controller is not able to control and the controlled power uses are power uses that the fuzzy logic controller is able to control. The fuzzy logic controller for EMC^2 takes in four inputs: total power usage, budget, time of day, and override. Override is an input given by the user, either zero or one. Zero represents an override off meaning that the fuzzy logic controller (EMC^2) is in control of the device and a one represents override on meaning that EMC^2 is not in control of the device. The time of day is an input from the clock that inputs into EMC^2 to represent the peak, off-peak, and partial peak times. The budget is an input given by the user's flexibility and strictness of their budget. It is a scale from one to ten in which a one represents a strict budget where comfortability will be sacrificed and a ten represents a flexible budget in which comfortability will be high priority. With all of these inputs, the fuzzy logic controller will have three outputs: output command, suggestion, and thermostat. The output command is an output from EMC^2 which will take in the inputs and turn on controlled power uses depending on rules set by the fuzzy logic. The suggestion is based on the output command to give the user readable text in the GUI in case the user is considering or is in override. The thermostat output command from EMC^2 inputs into the smart thermostat to add a degree depending on whether an appliance with significant temperature change is on.

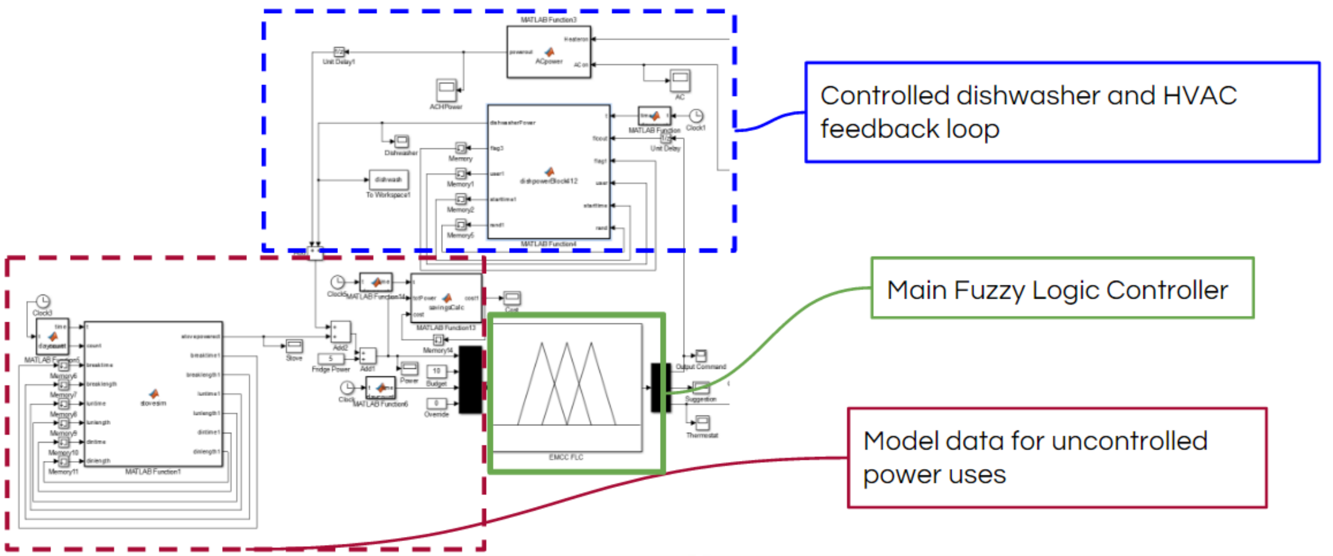

Figure 9: EMC^2 portion of the Simulink diagram.

There are two major appliances of the uncontrolled power uses: fridge power and stove power. The fridge is a simple constant of 5 Watts/minute. Most fridges on the market today maintain a certain temperature and thus maintain a certain power usage. When the stove is on, it uses a constant power usage and once the stove is off, the power usage goes to zero. Because of this, the stove will be best represented with a rectangular function.

$$rect(t) = \Pi(t) = \begin{cases} 0 \; if \; t = \; outside \; range \\ A \; if \; t > \; 5{:}00 \; PM \\ A \; if \; t < 5{:}30 \; PM \end{cases}$$

Figure 10: Example of a rect function used in our Simulink models.

In order to make the simulation more representative of real life, the randomness of the stove's on times had to be implemented. This is done with Matlab's rand() function. All of the outputs feedback into the inputs in order maintain randomness for multiple days. However, this randomness will be constrained to be between average breakfast times and dinner times.

There are two major appliances for the controlled power uses: dishwasher power and HVAC power. The dishwasher power will take in the output of the fuzzy logic and turn on depending on EMC^2 output. Before this, the dishwasher will first need an input from the user to signify that it can turn on. This is represented in the dishwasher function in which user input is randomized throughout the day. Again, the outputs of the model will feedback into the inputs in order to maintain randomness for multiple days. The dishwasher will output a constant rectangular function once-a-day to represent its power use to the fuzzy logic controller. The HVAC power will take in the smart thermostat's output command to turn on depending on certain inputs of the smart thermostat which will be talked about later. All of the controlled and uncontrolled power uses will then add up as an input of total power to the EMC^2 fuzzy logic.

4.2 Simulink Model - Smart Thermostat

The smart thermostat, much like EMC^2, is broken into three major sections: the fuzzy logic controller, uncontrolled inputs, and controlled outputs. The fuzzy logic controller (smart thermostat) takes in five inputs: indoor temperature, outdoor temperature, carbon dioxide, motion, and budget. The indoor and outdoor temperatures are represented as sinusoidal functions in which temperature is high during the middle of the day and temperature is low during the beginning and ends of the day. The carbon dioxide and motion are also sinusoids in which during the beginning and end of the day are high when people are around and low during the middle of the day when people are not around. The budget is an input given by the user's flexibility and strictness of their budget. It is a scale from one to ten in which a one represents a strict budget where comfort will be sacrificed and a ten represents a flexible budget in which comfort will be high priority. With all of these inputs, the fuzzy logic controller will have one output: the output command of when the HVAC system should turn on. The output is a number from 0-1. A number between 0 and 0.4 turns on the AC and a number between 0.6 and 1 turns on the heater. A number between 0.4 and 0.6 turns off the HVAC system.

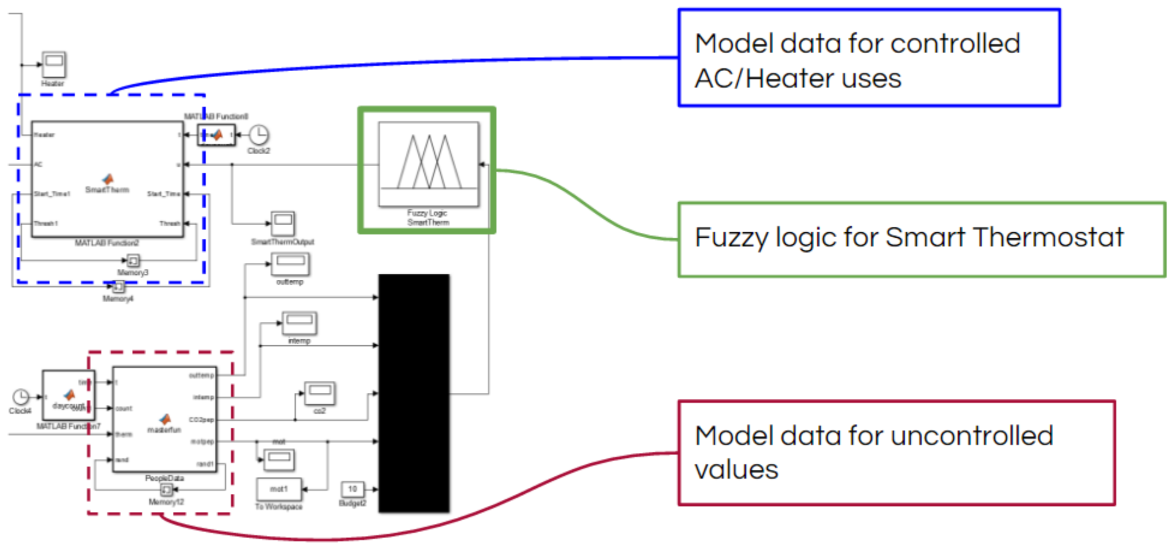

Figure 11: Smart Thermostat portion of the Simulink diagram.

There is one major appliance controlled by the smart thermostat which is the HVAC system. The HVAC system will take in the output command of the smart thermostat and either turn on or turn off the HVAC system. The power from the HVAC system will then feed into EMC^2. This whole process from EMC^2 to the smart thermostat will repeat for every run of our simulation.

4.3 Graphical User Interface (GUI)

The graphical user interface (GUI) will allow the user to input their flexibility and turn on or off the override. The graphical user interface allows the user to read the suggestion EMC^2 gives if the user is in override or considering override. The top of the GUI gives the user selection for how strict or flexible the user wants their budget. It is a scale from one to ten in which a one is estimated to give approximately 4% savings and a ten is estimated to approximately give a 1% savings. The stricter the budget, the more the user would have to sacrifice comfort. The middle section of the GUI is the override button. A green override button represents “Override Off” in which the fuzzy logic controllers are functioning and manipulating devices. A red override button represents “Override On” in which the user is in full control of his/her device. At the bottom of the GUI, a text box shows a suggestion if the user is considering override or is in override. For example, a suggestion can show that the current time is acceptable to turn devices on.

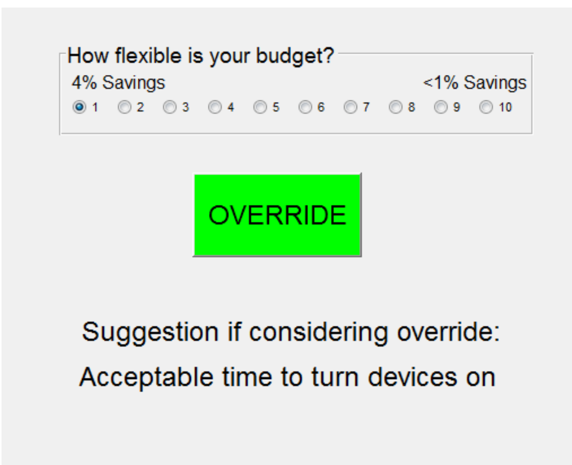

Figure 12: GUI with override off.

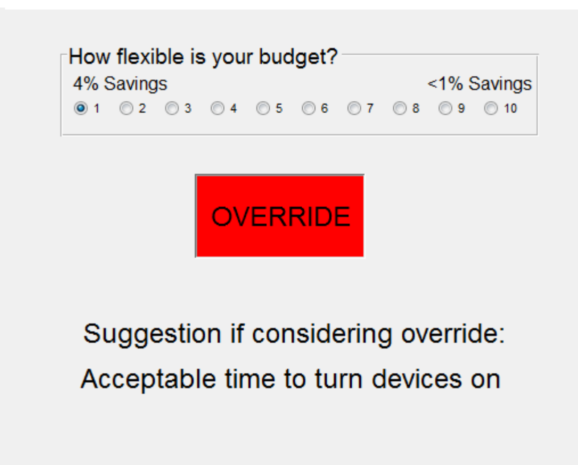

Figure 13: GUI with override on.

5.0 Testing and Analysis

5.1 Appliance Models

EMC^2 takes in information from three different appliances. These appliances were picked on the way in which power is consumed. A stove represents an appliance that turns on based on user preference and cannot be controlled by EMC^2. A refrigerator represents an appliance that is always on with a constant power use and cannot be controlled by either the user or EMC^2. A dishwasher is an appliance that turns on for a set period of time and can be controlled by EMC^2.

The first appliance listed is the stove, shown in Figure 14 below. The stove uses power only when turned on by the user and it is an uncontrolled device. This is modeled as a rect function where the amplitude corresponds to the power used by the stove, which was calculated as 33.33 Watts/min. The width of the rect function is determined by the user as the length of time that the stove is on. Figure 14 models a period of four days, with each day including a rect function that represents breakfast and dinner. Lunch was excluded from this graph to show realism that the typical home is usually vacant during the middle of the day as residents are either at work or school. Randomness was also included to allow for different times when the stove was turned on, as well as the length of time the stove was on.

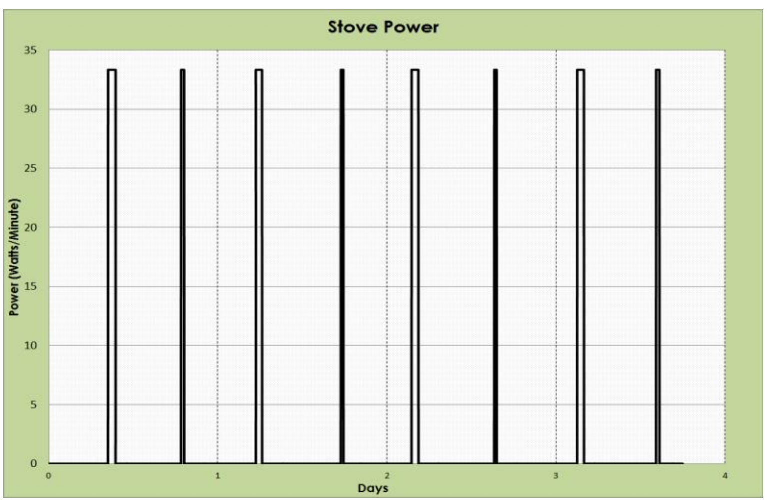

Figure 14: Stove models over multiple days, showing randomness in on-time and duration.

The second appliance model is the dishwasher, shown in Figure 15. This model uses the same basic principle as the stove. The dishwasher uses a rect function with an amplitude that corresponds to the power used, which was calculated at 22.5

Watts/min. The width of the rect function for the dishwasher is set to an average cycle duration of two hours. Randomness is also factored into the model to allow for varying times throughout the day when the user decides to turn the dishwasher on.

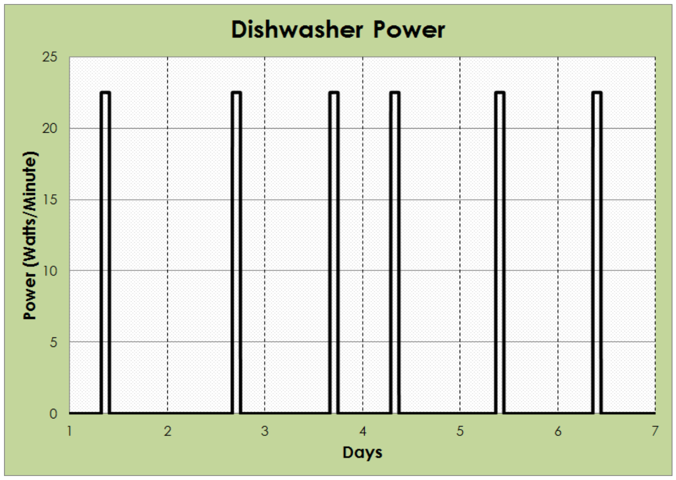

Figure 15: Dishwasher cycles over multiple days, showing random start time.

The third appliance model is the refrigerator, shown in Figure 16. This model is a constant value at all times. The power consumption is based on an average refrigerator that uses 5 Watts/min.

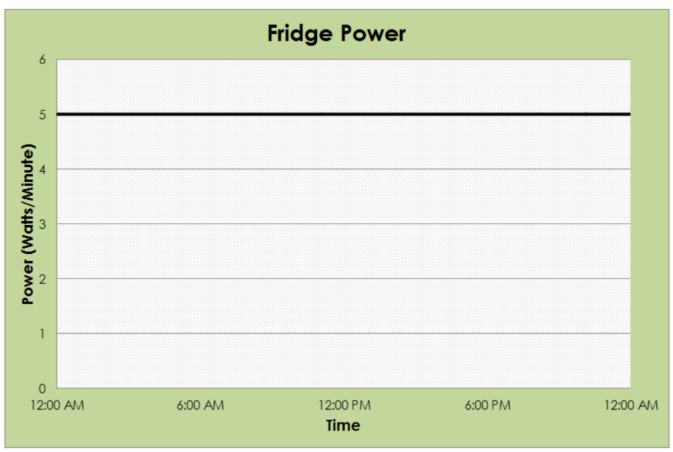

Figure 16: Fridge power modeled as a constant.

5.2 Thermostat Models

Just as EMC^2 had some input models, the Smart Thermostat has input models as well. Because real sensor data was not used for this part of the project, MATLAB models were created that mimicked real life so that the system would be able to control the HVAC system in a home, in the Smart Thermostat's case. As can be seen in Figure 18, the outdoor temperature varies a lot from day to day, because a lot of different days were desired to be represented so that the system can be shown to be robust and able to take in a lot of different types of inputs. The indoor temperature, seen in Figure 17, follows the outdoor temperature, but is scaled down due to insulation in the home. There are also some small peaks a few times a day for the indoor temperature, and those happen because empirical data was taken in a 10’ by 20’ by 8’ studio/kitchen space that shows that having the stove on for 30 minute windows while cooking actually increase the ambient temperature by 1 degree, and so the models take that into account too.

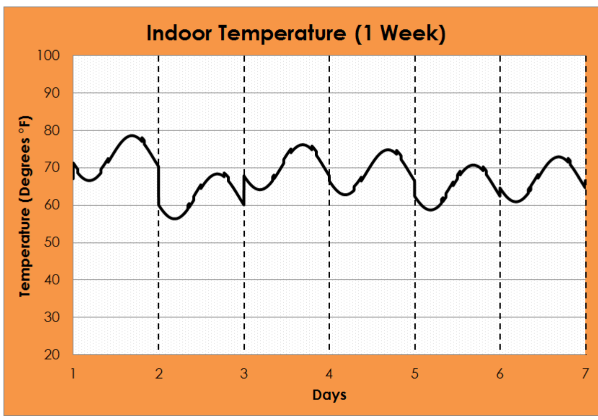

Figure 17: Indoor temperature over one week.

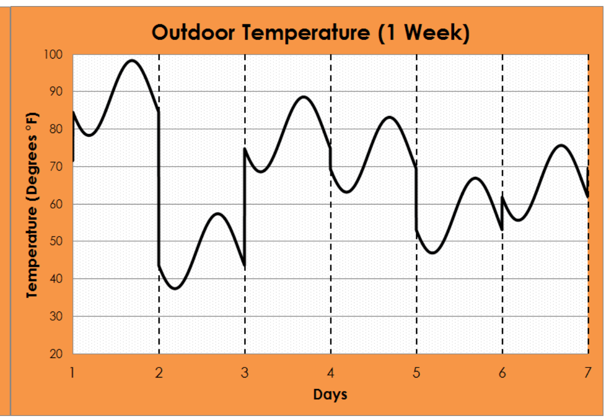

Figure 18: Outdoor temperature over one week.

Again, the outdoor temperature and indoor temperature can be seen below, but over 1 day. As stated above, the temperature models have some variability designed into them to mimic reality, and so the models in Figures 19 and 20 are representative of 1 possible day. The temperature spikes due to the stove being turned on in the indoor temperature can be better seen here.

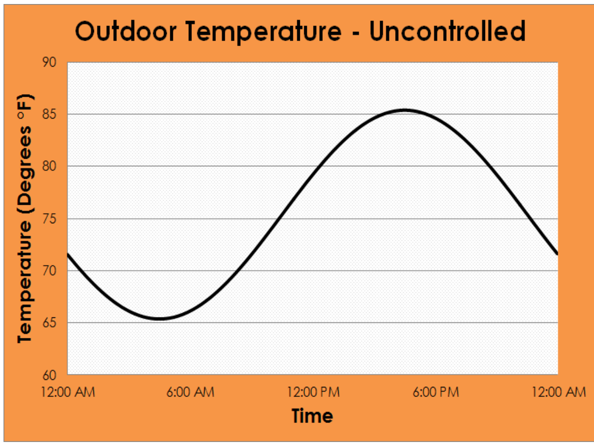

Figure 19: Outdoor temperature over one day.

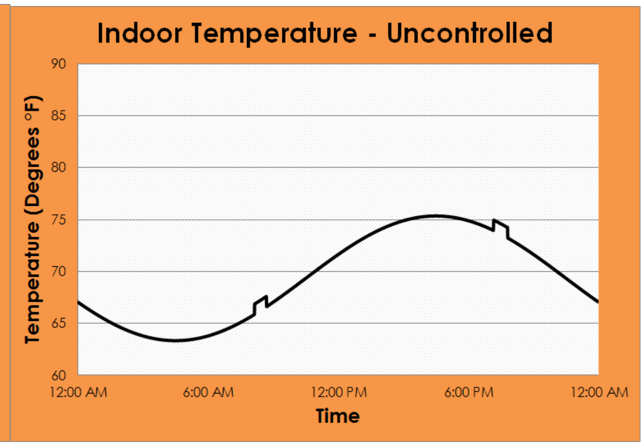

Figure 20: Indoor temperature over one day.

Figures 21 and 22 represent CO2 and Motion models that mimic reality. Their purpose in the system was to act as a check that people were actually home, to reduce the risk of appliances (especially HVAC which is triggered by changes in temperature that can happen regardless of people being home or not) which is a common power drain that is not necessary. This CO2 and motion check is very important because the goal of the system is to make power consumption more efficient. The graphs in Figures 21 and 22 represent what CO2 or Motion sensors would be able to pick up, in that on an average day most people are less likely to be home in the middle of the day and more likely to be home in the morning and at night. These signals both need to be above a certain value for the system's controller to recognize that people are home and then allowing for different appliances to be deployed.

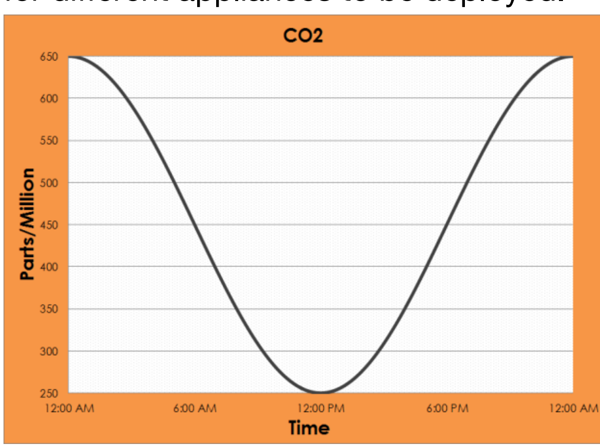

Figure 21: CO2 graph over one day.

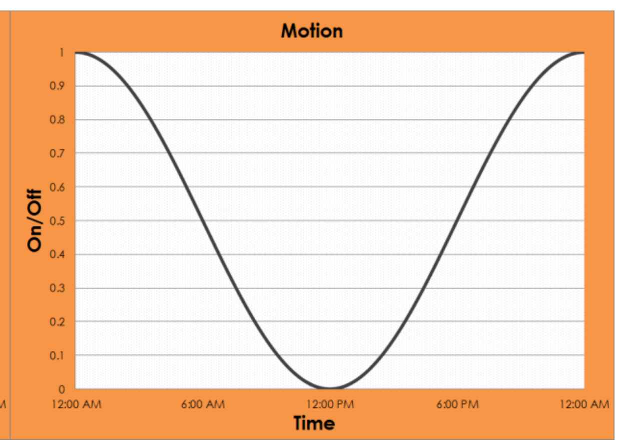

Figure 22: Motion sensor graph over one day.

5.3 Results - Appliance and HVAC

The two graphs in Figures 23 and 24 show the difference with a flexible budget and a strict budget for the dishwasher. Both have the same start time of 5PM. The flexible runs right away, showing the two-hour window that the dishwasher cycle runs. With a strict budget the start time is delayed until 8PM. Although cycle run time is static, the delay pushes the energy use out of peak times. With a start time of 8PM, only half the cycle is in partial-peak and the rest is off.

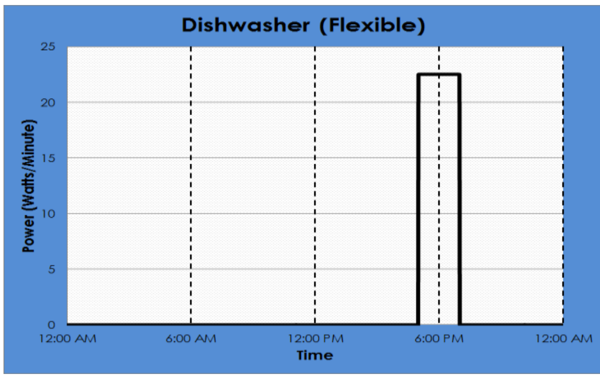

Figure 23: Dishwasher cycle, flexible budget, 5PM start.

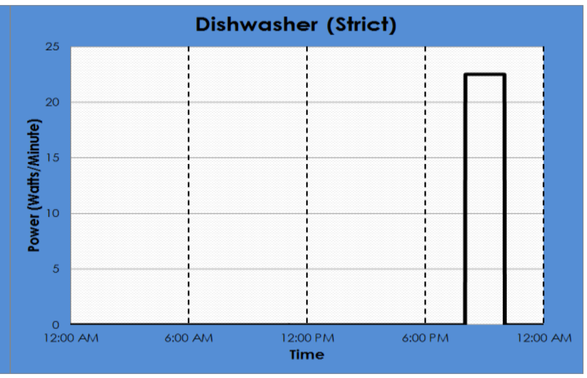

Figure 24: Dishwasher cycle, strict budget, 5PM start.

The graphs below show the differences of the heater with a strict and a flexible budget. In terms of heat use it is around an hour and a half reduction. This will, essentially, give a wider window around your set point that is acceptable. It is a sacrifice in user comfort, but with the smart thermostat, it is not going to be managed. It is a similar situation for the AC on times, which can be seen in Figures 25 and 26. This time the reduction in use was around two hours, which would be offset by the natural cooling that happens at night.

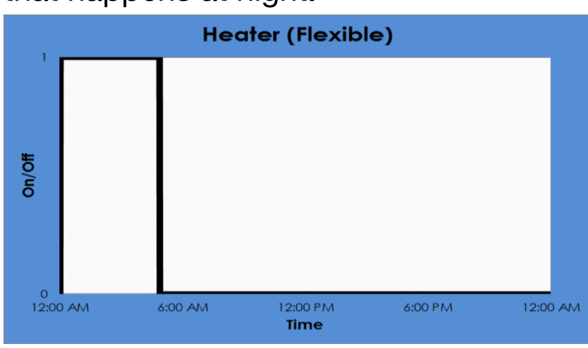

Figure 25: Heater on time flexible budget, one day.

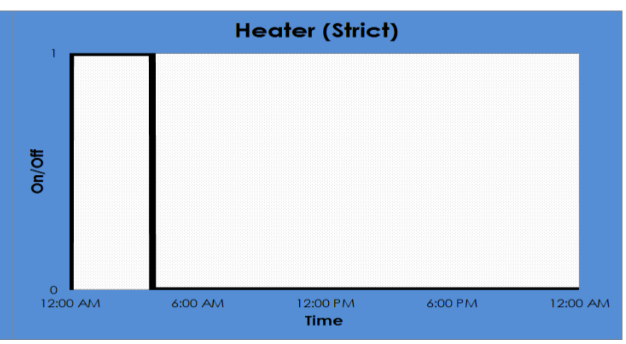

Figure 26: Heater on time strict budget, one day.

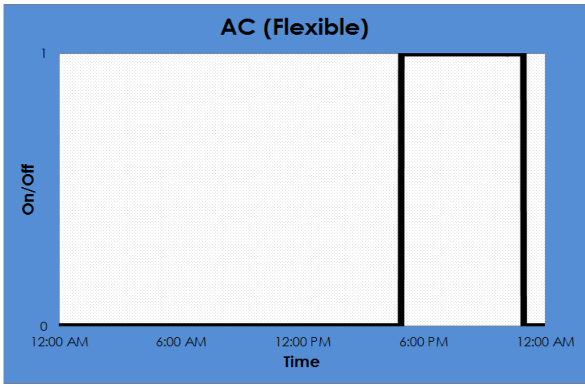

Figure 27: AC on time flexible budget, one day.

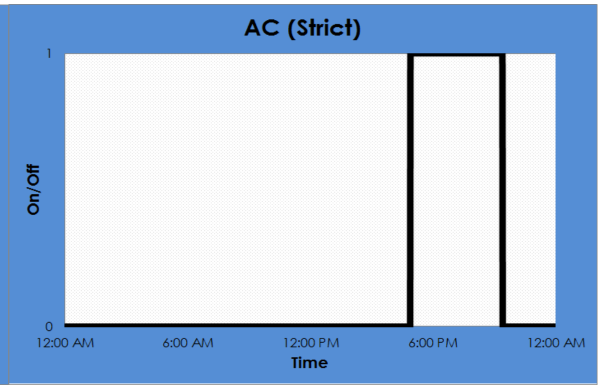

Figure 28: AC on time strict budget, one day.

The side-by-side comparison of Figures 29 and 30 show the effect that EMC² has on the indoor temperature. Figure 29 shows a fluctuation in temperature over a day from 75.2 to 63.3 degrees Fahrenheit. This range represents the indoor temperature without EMC² in place. Given a user selected set point of 69 degrees Fahrenheit, with a desired range of 5 degrees in either direction, we see that the indoor temperature exceeds the user selected comfort level on both the hot and cold ends of the range. Figure 30 shows how the fluctuation in temperature is reduced when EMC² is in place. The curve is truncated exactly within the desired range set by the user, from 74 to 64 degrees Fahrenheit. EMC² recognizes when the indoor temperature is either too hot or too cold, and adjusts accordingly by activating either the heater or A/C. Figure 30 specifically shows how the indoor temperature is affected when the budget is on the strictest setting. EMC² allows the home to fluctuate to the max ranges of the set point to save the most money by reducing HVAC on-time. On the most flexible budget setting, EMC² would disregard budget and maximize comfort by keeping a steady temperature exactly at the set point of 69 degrees Fahrenheit.

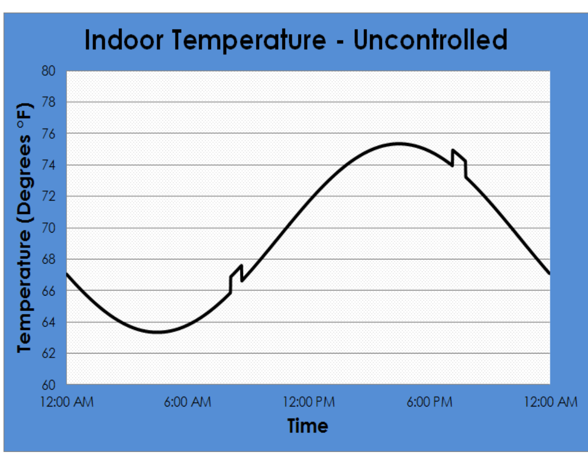

Figure 29: Indoor Temperature one day, uncontrolled.

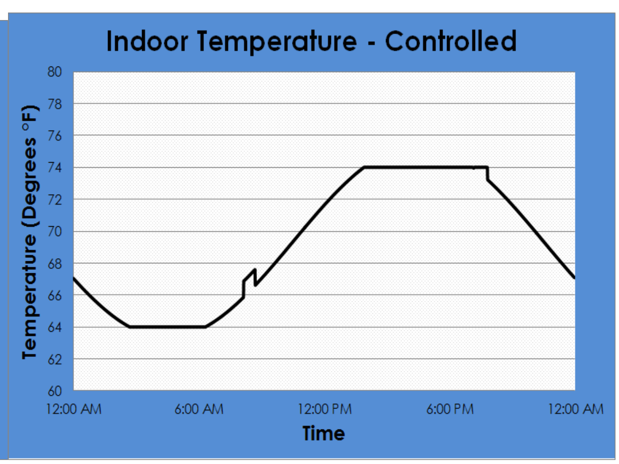

Figure 30: Indoor Temperature one day, controlled +/- 5 deg.

5.4 Results - Fuzzy Logic Controller Outputs

Below are the output curves from the EMC^2 fuzzy logic controller, shown in Figures 31 and 32. The controller takes in total power use, budget, time of day, and user override to compute an output as a value that ranges from 0 to 1. The output is based on how the inputs correlate to the fuzzy logic rules. When the output value is 0.5 or greater, this corresponds to an input signal to various functions to determine when to turn on. When the output value is less than 0.5, the system recognizes that is in not a proper time to use more power than necessary. This system allows EMC^2 to delay appliances, like the dishwasher, until such a time when it saves the most money.

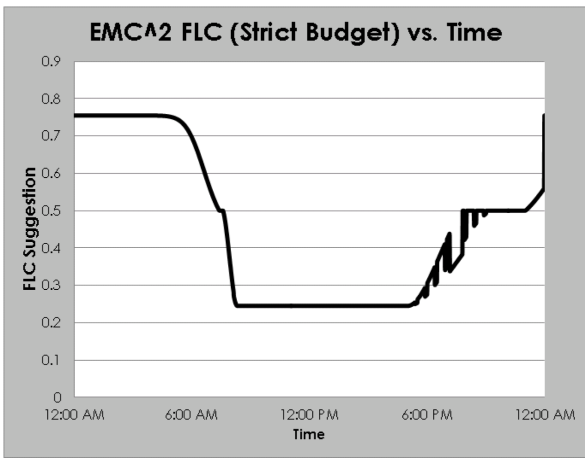

Figure 31: EMC2 Fuzzy Logic Output Strict Budget.

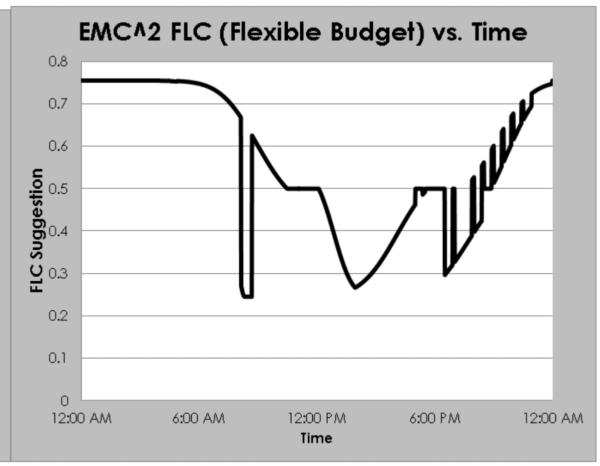

Figure 32: EMC2 Fuzzy Logic Output Flexible Budget.

Figures 33 and 34 show the output curves from the Smart Thermostat fuzzy logic controller. The controller takes in outdoor temperature, indoor temperature, CO2 levels, motion sensor readings, and budget to compute an output as a value that ranges from -1 to 1. The output is based on how the inputs correlate to the fuzzy logic rules. When the output value is 0.5 or greater, this corresponds to an input signal to turn on the heater. When the output value is -0.5 or less, this corresponds to an input signal to turn on the A/C. Lastly, when the output value is between 0.5 to -0.5, the system recognizes that is in not a proper time to use more power than necessary. Unlike the EMC^2 fuzzy logic controller output, the actual value of the Smart Thermostat fuzzy logic controller output is very important when signaling to turn on either the heater or A/C. That value corresponds to the percentage of the duty cycle that the HVAC system needs to be on to maintain the user-selected comfort range. A value that is close to the max output values of 1 and -1 will cause the HVAC to be on for a period of time very near the duration length of 30 minutes. On the other hand, a value closer to 0.5 and -0.5 will cause the HVAC system to turn on for a period of time near 15 minutes. The HVAC will never turn on for a period of time less than 15 minutes to minimize the risk of damaging the HVAC system through repeated activation/deactivation. A more flexible budget will allow the Smart Thermostat fuzzy logic output to give values closer to 1 and -1 more often.

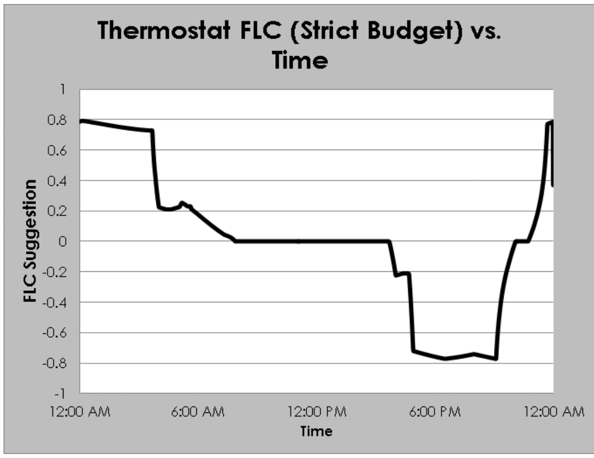

Figure 33: Strict Fuzzy Logic output for thermostat.

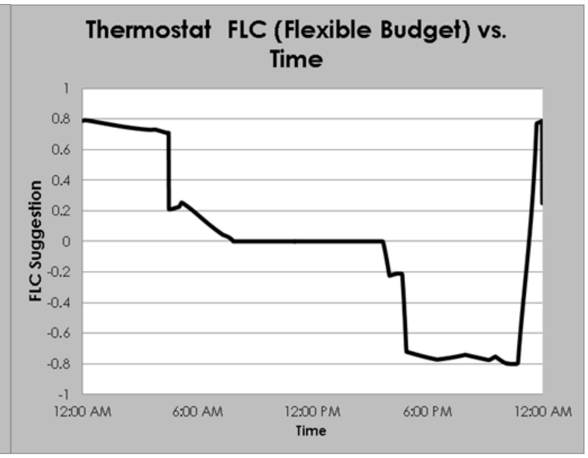

Figure 34: Flexible Fuzzy Logic output for Thermostat.

5.5 Simulated Savings

Figures 35 and 36 are the most representative of the overall purpose of EMC^2 with regard to the amount of potential savings. Figure 35 shows how power is normally consumed over the course of a day. The amount of power being used is denoted by the area under the red curve. The green lines represent the cost associated with a certain power use, which depends on the time of day as a function of the off-peak, partial-peak, and on-peak rates. The key issue with high energy bills under normal circumstances is the amount of power that gets used during on-peak times. Figure 36 shows how EMC^2 redistributes power use from on-peak to partial-peak and off-peak times. This is done to minimize electricity costs.

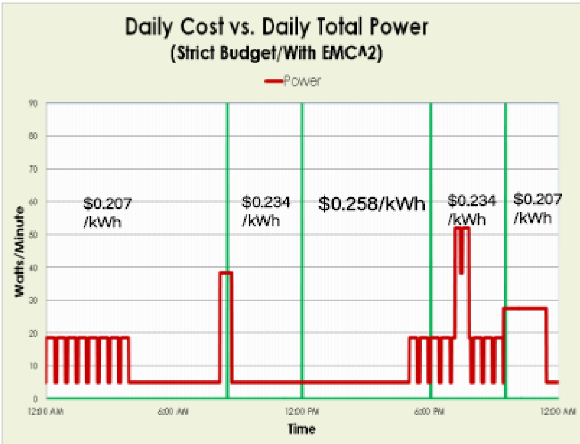

Figure 35: Daily cost without EMC2.

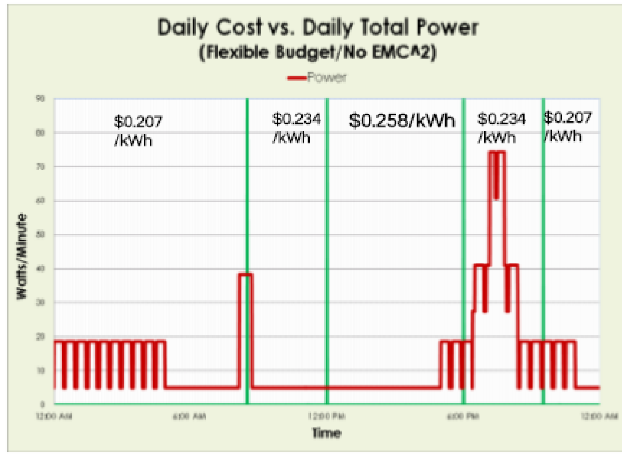

Figure 36: Daily cost with EMC2.

Figure 37 shows how much money can be saved with EMC^2. The black curve shows the cost of electricity of a home operating under normal conditions while the green curve shows the same parameters with EMC^2 in operation. After a week, a home, modeled for EMC^2, consuming 15 kWh per day will accumulate an energy bill of $31.90. In comparison, that same home will reduce its energy bill to $29.10 with EMC^2 in place. This $2.80 reduction in weekly energy costs grows to an annual 8.77% savings of $145.60. However, it is important to note that these costs savings are only calculated with the 3 appliances modeled for EMC^2. On average, an American home will consume 30 kWh per day, resulting in $291.20 in savings per year.

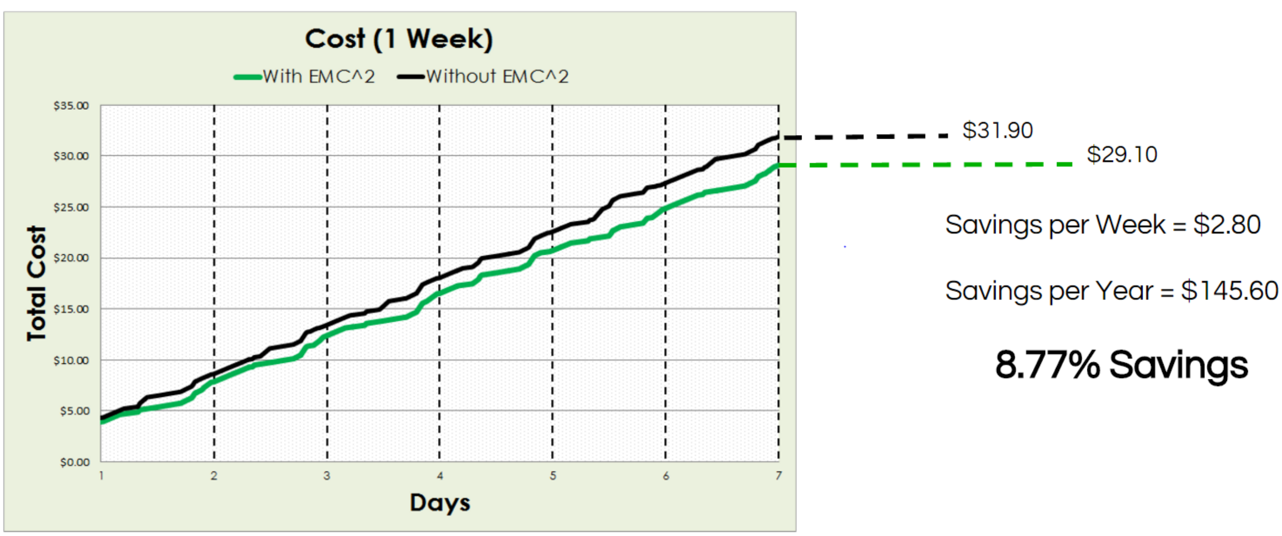

Figure 37: Cost savings with the system (green) and without (black).

5.6 Theoretical Product Cost

A “starter kit” EMC^2 system will theoretically cost around \$279, which is the total price of the parts needed for a single marketable product. At a cost of \$279 and an annual savings of \$291.20, a typical user can expect to reach return on investment in just under a year. The theoretical price is for this “starter kit” includes all of the necessary sensors, 2 Raspberry Pi microprocessors, and 5 smart plugs for major appliances. Further costs for a more complex system would be mainly associated to additional smart plugs, which can be purchased for \$23.00 each. Figure 38 shows a breakdown of the cost for all of the specific components.

Component	**# of Units**	**Unit Cost**
Raspberry Pi 2	2	\$39.25(x2)
Satechi IQ Smart Plug	5	\$23.00 (x5)
DS18B20 Thermometer(set of 10)	1	\$15.99
Bluetooth USB	2	\$5.34 (x2)
JQX-15F Magnetic Relay	1	\$5.79
S108T02 Solid State Relay	1	\$3.72
HC-SR501 Motion Sensor (set of 5)	1	\$7.83
MG-811 CO_2 Sensor	1	\$41.49
		TOTAL: \$279.00

Figure 38: Bill of Materials.

6.0 Evaluation of Design

Having completed this project, we now evaluate the final product against the initial goals we set for ourselves. When doing this project, there were constraints that dictated what we had to do and problems that required us to think of creative solutions. The following is a discussion of our creative solutions and concerning aspects.

6.1 Creative Solutions

When thinking of our project, we wanted to make a home energy management system that sustains user comfort through predictive model control. This would have been done through a low cost implementation and a system that can monitor power load to electrical appliances using certain communication protocols. This in and of itself was more difficult than what we initially expected as it requires us to not only create physical hardware that allows us to measure power but also a smart hub that will take that information and communicate with appliances and the HVAC system in the house. To make this project doable in the time allotted, we changed our requirements so that it would be a simulation based project that does not require predictive model control but rather models.

The main problem we faced during this project was modelling real world systems with mathematical functions. Because we are working with a simulation, extensive research had to be done in order to model appliances in Simulink as correctly and as accurately to their real world counterparts. For example, we found that the stove maintained a certain power throughout its activation thus allowing us to safely assume that a rectangular function would be represent its power usage. Adding to power modeling, we also had to do extensive research of how appliances affect the indoor temperature. We found that the stove and dishwasher release enough heat to heat up a small room by a degree every five minutes. This became an output from the EMC^2 fuzzy logic that outputs a degree depending on appliance activation.

Another problem we had was interfacing with the previous smart thermostat project. Because our project builds on the smart thermostat, we had to find creative solutions to help us interface with their project. We ultimately created more rules to their fuzzy logic as well as functions that helped the fuzzy logic perform to what the smart thermostat did. For example, the smart thermostat took in sensor data in real time and gave an output in real time. Our functions allowed us to bypass the real time and translate it into simulation time which will allow us to simulate multiple days in quick succession. Adding to this, models had to be made with indoor temperature, outdoor temperature, carbon dioxide, and motion. Using the data from the smart thermostat design, we found models that best represents each sensor. Overall, we found that each sensor is best represented with a sinusoid shifted and scaled to what seems logical and what is similar to the data of the previous smart thermostat.

All of these changes had to be made in order to make a robust simulation that tests multiple cases for multiple days. The major benefit we had with this is that we were able to simulate for multiple days without having to wait for multiple days. This allowed us to receive data quickly and allowed us to analyze the data quickly.

6.2 Concerning Aspects

A variety of challenges, obstacles, and concerns were tackled while working on this project. These concerns are mostly simulation based and should aid groups willing to work on this project.

The first concern we faced was the HVAC output. While working on this project, we initially found issues in which the AC would turn on during cold periods and the heater would turn on during hot periods. Of course, this logically did not make sense so debugging had to be done. Now, we have fixed this issue and the AC and heater should turn on appropriately. However, issues could still arise if there are certain cases in the fuzzy logic that our inputs do not address which can affect HVAC output.

Another concern we faced was with our functions that allowed us to run multiple days. Because we are using function blocks that feedback into themselves, memory blocks had to be used to first initialize those inputs. We initialized these inputs to be zero which can be an issue for those who want to add data to existing data after a simulation is stopped. Setting these initial conditions does not carry over past simulation data and thus could be an issue to those working with this project.

7.0 Future Work

If a group were to take on this project in the future, their main objective would be to build all of the physical components of EMC^2. Their main goals would be to install the EMC^2 control strategy on one of the Raspberry Pi boards, install the Smart Thermostat control strategy on the second Raspberry Pi board, successfully have both boards communicate with each other via Bluetooth, and finally have the whole system communicate with Bluetooth smart plugs. For further advancement, the system also would need to be able to seamlessly integrate varying numbers of smart plugs on demand. Additionally, EMC^2 needs to include demand response functionality to allow for constantly changing electricity rates and times.

8.0 Ethical Analysis

The most important technology in today's world is technology that impacts people. Almost all new technology is in one way or another trying to make people's lives better. Whether that be in the form of healthcare monitoring, electric cars, managing home energy efficiency, etc., there are multiple ways technology is benefiting people's lives. However, the more dynamic technology gets, the more in tune it has to be with ethical codes. Ethics is a topic that is now integrated with the very core of technology, and should be accounted for at every stage of a product's development and deployment. Ethical questions tend to complicate the advancement of the project. Often times, larger and more complicated projects require more ethical considerations. However, ethics should always be kept at the forefront of the design process because a world operating on unethical technology can be dangerous. As progress continues with our Senior Design project, we cannot afford to ignore the ethics. In analyzing our ethics, we think that looking at it in terms of three main components is the most effective. The first is the internal ethics that the team faces interpersonally and organizationally. The second component is social ethics and how exactly our project will impact the community. Finally, we must consider the ethical boundaries that come with developing a new idea and differentiating it from ideas already out there. An analysis of the ethics of our project will greatly improve its overall quality, and assist the team in having the most successful Senior Design possible.

Our design team consists of four electrical engineers with an electrical engineer advisor. Before a major decision was made, we consulted with each other to see if the decision was reasonable. Smaller decisions (i.e. grammatical changes in the proposal) were done through word of mouth and mutual agreement. Ultimately, our advisor was the person who made the final decision, in conjunction with unanimity amongst our group members. Our advisor has discussed with us in great length how she worked in industry for a long time and has the understanding to tackle issues between team members and organizations. Organizational concerns were tackled by first discussing amongst ourselves, and then going to our advisor and when further questions were asked, we would go to the ethics department of Santa Clara University.

Team decisions were made in a democratic fashion. Our advisor stressed that we do a senior design project that all members of the group enjoy doing, and have equal part in. When one person in the group found something amiss, we edited our project to address his concerns. After, we consulted with our advisor to see if a decision we made seemed logical and plausible. As mentioned earlier, our advisor made the final decision due to her expertise in this matter.

We based our ethical decisions on the IEEE code of Ethics. When parts of our project were seen to break any part of the code of ethics, we edited our project so that it followed all of the appropriate rules.

For potential users, the ethical duty of our product is to provide users with a safe and secure system that lessens the amount of electricity used in a home. Unfortunately, this gives a lot of power to the user. If our project were to physically do as it is made to do, the user would have power over the entire home and potentially other homes since our project would connect to the power grid. To remedy this, a future group working on the physical aspect of our project would have to implement a system that restricts the device to only target the power system of a user's home. This would require further research and advising to implement but would have to be done to boost the welfare of other homeowners.

Our project can have unintended ethical consequences under the wrong hands. If the central hub connecting all appliances were to be hacked, all appliances connected to the hub would be under the control of the hacker. This can hurt and in the worst cases kill the intended user of the product. Due to the gravity of this issue, our project will need to be secure and safe. We will have to create a strong security system that would only allow access for the intended user.

To ensure integrity in our research and design project, we documented all our research material and project designs. We created a shared folder in which we placed all our research material as well as design specifications. This is intended to show our research process and how we achieved our final design. For truthfulness and accuracy in testing, we ran many simulations to ensure that the objectives we claimed stand true.

To ensure our data is trustworthy, we consulted with experts in the field. Thankfully, our advisor is well connected and helped us in any way she can to verify our results, whether it was through her or through a colleague of hers. Because we built on last year's senior design project, we utilized most of their data. To ensure the data is accurate, we contacted members of the previous group and got clarification on their data. However, for open source data, we ran many simulations and tests to ensure the data is correct. Ultimately, our goal is to patent our project. Because of this, we are hesitant to share our research results. If the situation comes to it, we might disclose some information but not so much as to have our design copied.

The ethical obligations we have as a group will carry on to other teams wanting to continue our work. Because we never created a working physical prototype, the subsequent group will have to work on the physical prototype. The ethical obligation will still continue, but on a larger scale due to the physical risks that are involved. Our project risks the dangers of power usage and high voltage systems. If things go awry, members can be faced with electrocution and potentially a shutdown of the electrical grid.

9.0 Sustainability

Sustainability is an area of our Senior Design Project that we are putting a good amount of thought into and auditing constantly. As technology gets more and more dynamic and present in different parts of our lives, we must take into account its environmental impact.

Our senior design does not need any physical resources due to the underlying fact that our senior design is solely a simulation and a proof of concept. However, the physical build would require a variety of materials. Those materials include electrical components, various PCBS, communication modules, and other materials used in a modern day thermostat, plug, as well as any other appliance. When picking our parts, we would need the essential components for hardware design to run the software, controlling the home power usage. These components include items such as wires, fuses, and PCBs. One line of thinking is that maybe we could make our product more sustainable, by switching out the outer plastic casing of PCBs with biodegradable materials. It even allows for a cheaper way to dispose the outer casing if need be and would have little impact on the environment. However, the whole point of this energy management system is to be a permanent addition to the home. For example, the "Smart Plug" can be used for multiple appliances rather than just one. The "Central Hub" can be moved around the house as long as it has connection with other appliances within the home. And so in looking at this further and doing some more research we found that biodegradable materials are more expensive and would benefit products. Because of this we would probably go with the traditional electrical components. Ultimately, our product would lead to e-waste when disposed. Disposing these materials can lead to health risks, poor sanitation, and poisoning of the soil much like other appliances found around a typical home.

For both the hub and the Smart Plug, electronics are the main components and one big energy resource that is used up when manufacturing electronics is copper. Copper mining is an output energy resource depletion method of the PCBs that would be in our project. Silicon is also something that would be needed for the manufacturing of our potential product, so there are the energy resources that are depleted in the methods used to obtain silicon that we need to take into consideration. To be sold to consumers, packaging would also be required to protect the contents of the product. Product packaging is commonly some form of plastic, which is derived from organic products. The underlying goal of our project is energy savings, so our system is designed to be as efficient as possible. For operation, our system would only require a minimal amount of electricity to power the electronic components. In terms of energy use, our product would be approximately equivalent to plugging in another Wi-Fi router into the home.

If the project is to be taken to the next level, there will not be much in the way of needs for maintenance. The main thing would be software updates. The sensors may have a specific lifespan, but these are things that depend on the materials chosen. Repair would probably mean replacement, as it often does with electronics. The products operational lifespan would be in the 5-10 years range. Of course this depends on construction, but with modern day electronics it is entirely possible that it could last longer. There would be little need to replace it often as most changes would be through updates or the additions of new types of plugs. With proper planning, it could be expanded to anything that is plugged in and take into account for heat output more granularly. This will increase the lifespan of the device as well as improving its ability to control and save energy. Disposal of the device would not be something that depends on the construction, but it can be assumed with electronics that you would not be able to just toss it in the garbage. Instead it would need to be recycled akin to how computers and televisions are treated. Some parts will be able to be saved, but not everything, unfortunately. Hopefully recycling methods can improve enough to the point where the entire device is able to be reused.

10.0 Gantt Chart and Schedule

Below, in Figure 39, you can see the Gantt chart. The work is broken down on the left. Most were done in teams of two as the team had the manpower available. Many sections involved the whole team working on a specific part. This was to ensure the team was always kept up to date on all both models and Simulink, as well as functioning as a check we were meeting requirements.

Task Name	Start Date	Due Date	Assigned To
Research	09/28/15	12/11/15	All
Create Specifications	10/05/15	01/03/16	All
Create Simulink Models	01/03/16	03/03/16	Ren and Matthew
Controller Initial Design	02/01/16	03/03/16	Devon and Francis
Testing	03/03/16	05/06/16	All
Adjust Simulink Models	03/03/16	05/06/16	Ren and Matthew
Redesign Controller and Finalize	03/03/16	05/06/16	Devon and Francis
Presentation Prep	05/06/16	05/11/16	All
Final Changes to Project	05/11/16	05/31/16	All
Thesis	05/11/16	05/31/16	All

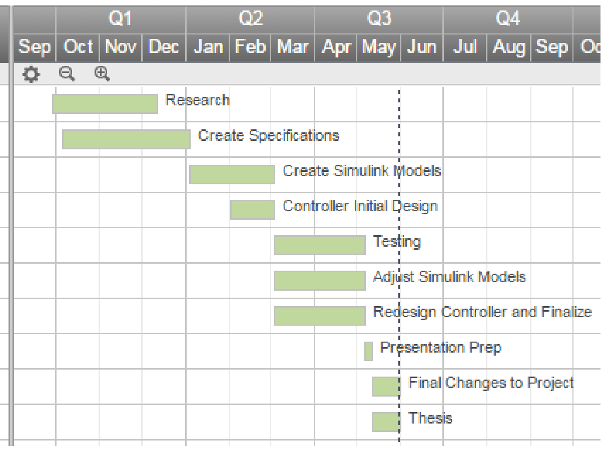

Figure 39: Year long Gantt Chart and corresponding Schedule.

11.0 Bibliography

Anderson, Don L. *Theory of the Earth.* Boston: Blackwell Scientific Publications, 1989. Print.

LaMarche, Janelle, Katherine Cheney, Sheila Christian, and Kurt Roth. “Home Energy Management Products & Trends.” Fraunhofer Center for Sustainable Energy Systems., Cambridge, CA. 2011.

Park, Herie, Marie Ruellan, Nadia Martak, Rachid Bennacer, and Eric Monmasson. “Generic Thermal Model of Electric Appliances Integrated in Low Energy Building.” University of Cergy-Pontoise., Cergy-Pontoise, France. 2012.

Saha, Avijit, Murat Kuzlu, and Manisa Pipattanasomporn. “Demonstration of a Home Energy Management System with Smart Thermostat Control.” Advanced Research Institute., Arlington, VA. 2013.

Veleva, Sanja, Danco Davcev, and Marija Kacarska. "Wireless smart platform for Home Energy Management System," Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on, Manchester, 2011, pp. 1-8.

Yashiro, Takeshi, Shinsuke Kobayashi, Noboru Koshizuka, and Ken Sakamura. “A Software HEMS Framework for Consumer Electronics.” YRP Ubiquitous Networking Laboratory., Tokyo, Japan. 2013.

Yoshihisa, Tomoki, Naotaka Fujita, and Masahiko Tsukamoto. "HEMS toolkit: A toolkit for constructing a home energy management system," 2011 IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2011, pp. 822-823.

12.0 Appendices

Appendix A

A.1 Stove Model Code

```
Function [stovepowerect,breaktime1,breaklength1,luntime1,lunlength1,dintime1,dinlength1] =
stovesim(t,count,breaktime,breaklength,luntime,lunlength,dintime,dinlength)

breaktime1=breaktime;
breaklength1=breaklength;
luntime1=luntime;
lunlength1=lunlength;
dintime1=dintime;
dinlength1=dinlength;


%randomness generation
hour=60;
breakrand=abs(floor(normrnd(8*hour,15,1,1)));
breaklengthrand=abs(floor(normrnd(25,5,1,1)));

lunrand=abs(floor(normrnd(12*hour,15,1,1)));
lunlengthrand=abs(floor(normrnd(35,10,1,1)));

dinrand=abs(floor(normrnd(19*hour,30,1,1)));
dinlengthrand=abs(floor(normrnd(60,15,1,1)));


if t==0
   breaktime1=breakrand;
   breaklength1=breaklengthrand;

   luntime1=lunrand;
   lunlength1=lunlengthrand;

   dintime1=dinrand;
   dinlength1=dinlengthrand;

end



breakfaststart=breaktime1;
breakfastend=breaktime1+breaklength1;
lunchstart=luntime1;
lunchend=luntime1+lunlength1;
dinnerstart=dintime1;
```

```
dinnerend=dintime1+dinlength1;

%power info
stovepowerinitial=5/9*60;
stovepowerect=0;



    if (t>breakfaststart && t<breakfastend)
        stovepowerect=stovepowerinitial;
    end

    if (t==breakfastend)
        stovepowerect=0;
    end

if count==5 || count==6

    if (t>lunchstart && t<lunchend)
        stovepowerect=stovepowerinitial;
    end

    if (t==lunchend)
        stovepowerect=0;
    end

end

    if (t>dinnerstart && t<dinnerend)
        stovepowerect=stovepowerinitial;
    end

    if (t==dinnerend)
        stovepowerect=0;
    end

end
```

A.2 Dishwasher Model Code

```
function [dishwasherPower,flag3,user1,starttime1,rand1] =
dishpowerBlock412(t,flcout,flag1,user,starttime,rand)

%Need to add memory blocks to ensure that all variables will be defined.
%They reset every increment of t, so the if setup only works once then
%leaves s


%memory for user, flag1, flag2, starttime
```

```
rand1=rand;
flag3=flag1;
user1=user;
starttime1=starttime;
hour=60;
noon= 12*hour;
peakStart = 8*hour;
peakEnd = 21*hour;
twoHours=2*hour;
dishwashduration=twoHours;
power=0.375*60;
true=1;
false=0;

%setup randomness

r=abs(floor(normrnd(14*hour,8*hour,1,1)));

if t==0
   rand1=r
end

if(t==rand1)
   user1=1;
end

swtch=user1+flcout;

if(swtch>=1.5)
   starttime1 = t;
   flag3=true;
   swtch = 0;
   user1=0;
end

cycle = starttime1 + dishwashduration;
cycle1=mod(cycle,1439);

if(flag3==true && t < cycle1)
   dishwasherPower = power;
else
   dishwasherPower=0;
end


if(flag3 && t==cycle1)
   flag3=false;
end

end
```

A.3 Uncontrolled Thermostat Input Code

```
function [outtemp,intemp,CO2pep,motpep,rand1] = masterfun(t,count,therm,rand)

%% Co2 sensor
CO2pep = 0;

if count>=0 && count<=4
   CO2pep=550 + 175*sin(.0043633231*t+pi/2);
elseif count==5
   CO2pep=550 + 175*sin(.0043633231*2*t+pi/2);
elseif count==6
   CO2pep=725;
end

%% Motion Sensor
motpep=0;

if count>=0 && count<=4
   motpep=.5 + .5*sin(.0043633231*t+5*pi/8);
elseif count==5
   motpep=.5 + .5*sin(.0043633231*2*t+5*pi/8);
elseif count==6
   motpep=1;
end


%% Temperatures
%random function
rand1=rand;

if t==0
   rand1=normrnd(0,5,1,1);
end

%outtemp
outtemp = rand1+ 70 + 20*sin(.0043633231*t+pi+pi/8);

%intemp
if therm >= 0.5
   outdeg = 1;
else
   outdeg = 0;
end

randin=rand1/4;

intemp = randin + outdeg + 12*sin(0.0043633231*t+pi+pi/8) + 68;
```

```
end
```

A.4 HVAC Control Code

```
function[Heater, AC, Start_Time1, Thresh1] =  SmartTherm(t,u, Start_Time, Thresh)

HVAC_Duration = 30;
count = mod(t,HVAC_Duration);

Start_Time1 = Start_Time;
Thresh1 = Thresh;

Heater = 0;
AC = 0;

if count == 0
  Start_Time = t;
  Start_Time1 = Start_Time;
  Thresh = u;
  Thresh1 = Thresh;
end

cycle = Start_Time1 + HVAC_Duration;

if Thresh1 >= 0.5 && t < cycle %turn heat on
   T_on = Thresh1*HVAC_Duration + Start_Time1;

   if t <= T_on
      Heater = 1;
   end

   if t > T_on
      Heater = 0;
   end
end

if Thresh1 <= -0.5 && t < cycle %turn AC on
   T_on = -Thresh1*HVAC_Duration + Start_Time1;

   if t <= T_on
      AC = 1;
   end

   if t > T_on
      AC = 0;
   end
end
```

```
if Thresh1 < 0.5 && Thresh1 >-0.5 && t < cycle %turn nothing on
   Heater = 0;
   AC = 0;
end
```

A.5 HVAC Power Calculation Code

```
function powerout  = ACpower(Heateron,ACon)

ACpowertotalwatt=815;
ACpower=ACpowertotalwatt/60;

Heaterpowertotalwatt=815;
Heaterpower=Heaterpowertotalwatt/60;

if ACon >= 0.5
   ACpowerout = ACpower;
else
   ACpowerout = 0;
end


if Heateron>=0.5
   Heaterpowerout = Heaterpower;
else
   Heaterpowerout = 0;
end

powerout=ACpowerout + Heaterpowerout;

end
```

A.6 Cost Calculator Code

```
function cost1 = savingsCalc(t,totPower,cost)

offpeak=0.207; %per KWh
partialpeak=0.234;
onpeak=0.258;
cost1=0;
totPower1=totPower/1000; %KW


if (t>=0 && t<=510)

   cost1=cost+totPower1*offpeak;

elseif (t>=511 && t<=720)
```

```
    cost1=cost+totPower1*partialpeak;
elseif (t>=721 && t<=1080)
    cost1=cost+totPower1*onpeak;
elseif (t>=1081 && t<=1290)
    cost1=cost+totPower1*partialpeak;
elseif (t>=1291 && t<=1440)
    cost1=cost+totPower1*offpeak;
end

end
```

A.7 Day Counting Code

```
function [time,count1] = daycount(t)

%ensure time between 0 and 1339 for 1440 minute day
time=mod(t,1439);

%count days between 0 and 6 for 7 day week
s=t/1440;
ex=floor(s);
count1=mod(ex,6);

end
```

Appendix B

B.1 EMC2 Rule Code

```
[System]
Name='Control'
Type='mamdani'
Version=2.0
NumInputs=4
NumOutputs=3
NumRules=60
AndMethod='min'
```

```
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='PowerUsage'
Range=[0 70]
NumMFs=3
MF1='Low':'trimf',[-28 0 20]
MF2='Med':'trimf',[15 25 35]
MF3='High':'smf',[32 35]

[Input2]
Name='BudgetGiven'
Range=[1 10]
NumMFs=2
MF1='Strict':'zmf',[3 6]
MF2='Flexible':'trimf',[4 10 11.25]

[Input3]
Name='TimeOfDay'
Range=[0 1440]
NumMFs=5
MF1='PartialPeak1':'gauss2mf',[60 490 60 680]
MF2='Peak':'gauss2mf',[120 720 120 1080]
MF3='PartialPeak2':'gauss2mf',[120 1150 120 1290]
MF4='Offpeak2':'smf',[1020 1410]
MF5='OffPeak1':'zmf',[432 500]

[Input4]
Name='Override'
Range=[0 1]
NumMFs=2
MF1='OFF':'zmf',[0.5032 0.5032]
MF2='ON':'smf',[0.5067 0.5106]
```

```
[Output1]
Name='OutputCommand'
Range=[0 1]
NumMFs=2
MF1='OFF':'zmf',[0.498 0.499]
MF2='ON':'smf',[0.5 0.501]

[Output2]
Name='Suggestion'
Range=[0 1]
NumMFs=3
MF1='Dont':'trimf',[-0.4 0 0.4]
MF2='Can':'trimf',[0.1 0.5 0.9]
MF3='Should':'trimf',[0.6 1 1.4]

[Output3]
Name='Thermostat'
Range=[0 1]
NumMFs=2
MF1='Add0':'zmf',[0.498 0.499]
MF2='Add1':'smf',[0.5 0.501]

[Rules]
1 1 5 1, 2 3 1 (1) : 1
1 1 1 1, 1 1 1 (1) : 1
1 1 2 1, 1 1 1 (1) : 1
1 1 3 1, 1 1 1 (1) : 1
1 1 4 1, 2 3 1 (1) : 1
1 2 5 1, 2 3 1 (1) : 1
1 2 1 1, 2 2 1 (1) : 1
1 2 2 1, 1 1 1 (1) : 1
1 2 3 1, 2 2 1 (1) : 1
1 2 4 1, 2 3 1 (1) : 1
2 1 5 1, 2 3 1 (1) : 1
2 1 1 1, 1 2 1 (1) : 1
2 1 2 1, 1 1 1 (1) : 1
2 1 3 1, 1 2 1 (1) : 1
```

2 1 4 1, 2 3 1 (1) : 1
2 2 5 1, 2 3 1 (1) : 1
2 2 1 1, 2 2 1 (1) : 1
2 2 2 1, 1 1 1 (1) : 1
2 2 3 1, 2 2 1 (1) : 1
2 2 4 1, 2 3 1 (1) : 1
3 1 5 1, 2 3 2 (1) : 1
3 1 4 1, 2 3 2 (1) : 1
3 1 1 1, 1 2 2 (1) : 1
3 1 2 1, 1 1 2 (1) : 1
3 1 3 1, 1 2 2 (1) : 1
3 2 5 1, 2 3 2 (1) : 1
3 2 4 1, 2 3 2 (1) : 1
3 2 1 1, 1 2 2 (1) : 1
3 2 2 1, 1 1 2 (1) : 1
3 2 3 1, 1 2 2 (1) : 1
1 1 5 2, 2 3 1 (1) : 1
1 1 1 2, 2 2 1 (1) : 1
1 1 2 2, 2 1 1 (1) : 1
1 1 3 2, 2 2 1 (1) : 1
1 1 4 2, 2 3 1 (1) : 1
1 2 5 2, 2 3 1 (1) : 1
1 2 1 2, 2 2 1 (1) : 1
1 2 2 2, 2 1 1 (1) : 1
1 2 3 2, 2 2 1 (1) : 1
1 2 4 2, 2 3 1 (1) : 1
2 1 5 2, 2 3 1 (1) : 1
2 1 1 2, 2 2 1 (1) : 1
2 1 2 2, 2 1 1 (1) : 1
2 1 3 2, 2 2 1 (1) : 1
2 1 4 2, 2 3 1 (1) : 1
2 2 5 2, 2 3 1 (1) : 1
2 2 1 2, 2 2 1 (1) : 1
2 2 2 2, 2 1 1 (1) : 1
2 2 3 2, 2 2 1 (1) : 1
2 2 4 2, 2 3 1 (1) : 1

```
3 1 5 2, 2 3 2 (1) : 1
3 1 4 2, 2 3 2 (1) : 1
3 1 1 2, 2 2 2 (1) : 1
3 1 2 2, 2 1 2 (1) : 1
3 1 3 2, 2 2 2 (1) : 1
3 2 5 2, 2 3 2 (1) : 1
3 2 4 2, 2 3 2 (1) : 1
3 2 1 2, 2 2 2 (1) : 1
3 2 2 2, 2 1 2 (1) : 1
3 2 3 2, 2 2 2 (1) : 1
```

B.2 Smart Thermostat Rule Code

```
[System]
Name='FuzzyControlThermWithBudget'
Type='mamdani'
Version=2.0
NumInputs=5
NumOutputs=1
NumRules=108
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='OutdoorTemp'
Range=[-20 120]
NumMFs=3
MF1='cold':'trimf',[-200 20 60]
MF2='average':'trimf',[50 70 90]
MF3='hot':'trimf',[80 100 500]

[Input2]
Name='IndoorTemp'
Range=[50 85]
```

```
NumMFs=3
MF1='cold':'trimf',[-500 62 68]
MF2='medium':'trimf',[65 69 73]
MF3='hot':'trimf',[70 75 500]

[Input3]
Name='CO2'
Range=[300 700]
NumMFs=3
MF1='Zero':'trimf',[-100 300 500]
MF2='One':'trimf',[450 550 650]
MF3='Many':'trimf',[550 650 1000]

[Input4]
Name='MotionSensor'
Range=[0 1]
NumMFs=2
MF1='Off':'trimf',[0 0 0.5]
MF2='On':'trimf',[0.5 1 1]

[Input5]
Name='BudgetGiven'
Range=[1 10]
NumMFs=2
MF1='Strict':'zmf',[3 6]
MF2='Flexible':'trimf',[4 10 11.25]

[Output1]
Name='HVACsignal'
Range=[-1 1]
NumMFs=3
MF1='AC':'trimf',[-1.2 -1 -0.4]
MF2='off':'trimf',[-0.6 0 0.6]
MF3='Heat':'trimf',[0.4 1 1.2]

[Rules]
1 1 1 1 2, 2 (1) : 1
```

1 1 1 2 2, 2 (1) : 1
1 1 2 1 2, 3 (1) : 1
1 1 2 2 2, 3 (1) : 1
1 1 3 1 2, 3 (1) : 1
1 1 3 2 2, 3 (1) : 1
1 2 1 1 2, 2 (1) : 1
1 2 1 2 2, 2 (1) : 1
1 2 2 1 2, 3 (1) : 1
1 2 2 2 2, 3 (1) : 1
1 2 3 1 2, 2 (1) : 1
1 2 3 2 2, 3 (1) : 1
1 3 1 1 2, 2 (1) : 1
1 3 1 2 2, 2 (1) : 1
1 3 2 2 2, 1 (1) : 1
1 3 2 1 2, 1 (1) : 1
1 3 3 1 2, 1 (1) : 1
1 3 3 2 2, 1 (1) : 1
2 1 1 1 2, 2 (1) : 1
2 1 1 2 2, 2 (1) : 1
2 1 2 1 2, 2 (1) : 1
2 1 2 2 2, 3 (1) : 1
2 1 3 1 2, 2 (1) : 1
2 1 3 2 2, 3 (1) : 1
2 2 1 1 2, 2 (1) : 1
2 2 1 2 2, 2 (1) : 1
2 2 2 1 2, 2 (1) : 1
2 2 2 2 2, 2 (1) : 1
2 2 3 1 2, 1 (1) : 1
2 2 3 2 2, 1 (1) : 1
2 3 1 1 2, 2 (1) : 1
2 3 1 2 2, 2 (1) : 1
2 3 2 1 2, 2 (1) : 1
2 3 2 2 2, 1 (1) : 1
2 3 3 1 2, 2 (1) : 1
2 3 3 2 2, 1 (1) : 1
3 1 1 1 2, 2 (1) : 1

3 1 1 2 2, 2 (1) : 1
3 1 2 1 2, 3 (1) : 1
3 1 2 2 2, 3 (1) : 1
3 1 3 1 2, 3 (1) : 1
3 1 3 2 2, 3 (1) : 1
3 2 1 1 2, 2 (1) : 1
3 2 1 2 2, 2 (1) : 1
3 2 2 1 2, 1 (1) : 1
3 2 2 2 2, 1 (1) : 1
3 2 3 1 2, 2 (1) : 1
3 2 3 2 2, 1 (1) : 1
3 3 1 1 2, 2 (1) : 1
3 3 2 1 2, 1 (1) : 1
3 3 2 2 2, 1 (1) : 1
3 3 3 1 2, 1 (1) : 1
3 3 3 2 2, 1 (1) : 1
3 3 1 2 2, 2 (1) : 1
1 1 1 1 1, 2 (1) : 1
1 1 1 2 1, 2 (1) : 1
1 1 2 1 1, 3 (1) : 1
1 1 2 2 1, 3 (1) : 1
1 1 3 1 1, 3 (1) : 1
1 1 3 2 1, 3 (1) : 1
1 2 1 1 1, 2 (1) : 1
1 2 1 2 1, 2 (1) : 1
1 2 2 1 1, 2 (1) : 1
1 2 2 2 1, 3 (1) : 1
1 2 3 1 1, 3 (1) : 1
1 2 3 2 1, 3 (1) : 1
1 3 1 1 1, 2 (1) : 1
1 3 1 2 1, 2 (1) : 1
1 3 2 2 1, 2 (1) : 1
1 3 2 1 1, 1 (1) : 1
1 3 3 1 1, 1 (1) : 1
1 3 3 2 1, 1 (1) : 1
2 1 1 1 1, 2 (1) : 1

2 1 1 2 1, 2 (1) : 1
2 1 2 1 1, 2 (1) : 1
2 1 2 2 1, 2 (1) : 1
2 1 3 1 1, 3 (1) : 1
2 1 3 2 1, 3 (1) : 1
2 2 1 1 1, 2 (1) : 1
2 2 1 2 1, 2 (1) : 1
2 2 2 1 1, 2 (1) : 1
2 2 2 2 1, 2 (1) : 1
2 2 3 1 1, 2 (1) : 1
2 2 3 2 1, 2 (1) : 1
2 3 1 1 1, 2 (1) : 1
2 3 1 2 1, 2 (1) : 1
2 3 2 1 1, 2 (1) : 1
2 3 2 2 1, 2 (1) : 1
2 3 3 1 1, 1 (1) : 1
2 3 3 2 1, 1 (1) : 1
3 1 1 1 1, 2 (1) : 1
3 1 1 2 1, 2 (1) : 1
3 1 2 1 1, 2 (1) : 1
3 1 2 2 1, 3 (1) : 1
3 1 3 1 1, 3 (1) : 1
3 1 3 2 1, 3 (1) : 1
3 2 1 1 1, 2 (1) : 1
3 2 1 2 1, 2 (1) : 1
3 2 2 1 1, 2 (1) : 1
3 2 2 2 1, 1 (1) : 1
3 2 3 1 1, 1 (1) : 1
3 2 3 2 1, 1 (1) : 1
3 3 1 1 1, 2 (1) : 1
3 3 1 2 1, 2 (1) : 1
3 3 2 1 1, 1 (1) : 1
3 3 2 2 1, 1 (1) : 1
3 3 3 1 1, 1 (1) : 1
3 3 3 2 1, 1 (1) : 1