Automated Demand Response Refrigerator Project

Jessica Tran[1], Jordan Gilles[1], Ryan Mann[2], and Vishnu Murthy[2]
[1] Civil and Environmental Engineering, University of California, Berkeley
[2] Energy Engineering, University of California, Berkeley

Abstract – This project aimed to create a smart refrigerator for both perishables and non-perishables that is capable of using prior usage data and the user's choice of objectives to optimize its energy usage and performance. Using prior usage data, the refrigerator can predict times at which it is most likely to be opened and its contents consumed, and (when in non-perishables mode) adjust its temperature setpoint accordingly to ensure that it reaches the desired temperature at that time. In addition, an online dashboard allows the user to view sensor data and control the refrigerator mode remotely. In either of the two modes, the refrigerator's compressor schedule is optimized using a thermal-model-based mixed integer linear program to minimize marginal carbon emissions (using the WattTime API) and electricity cost (using a time-of-use rate schedule).

I. Introduction

A. Motivation & Background

CALIFORNIA, as well as many other parts of the world, is rapidly increasing the percentage of renewables on its grid. However, there are significant challenges associated with accommodating a high penetration of variable and intermittent energy generators.

The current approach relies primarily on using natural gas peaker plants; however, this strategy is ultimately incompatible with long-term climate policy goals, and could result in increased electricity prices.

Another method of reducing variability involves promoting technological and geographic diversity among renewable generators. While effective, this tactic is not sufficient to guarantee that power can be provided reliably.

Deployment of energy storage devices has been proposed as an alternative strategy. Although storage is promising, cost and scale remain as hurdles. Fortunately, there exists another approach that is functionally equivalent to deploying massive amounts of storage infrastructure, yet only requires the use of relatively inexpensive electronics and software: automated demand response.

Demand response involves strategically shifting demand for electric power away from times when it stresses the grid and towards times when it can be more easily accommodated. Once primarily performed by large industrial and commercial consumers on hot summer days after a phone call from the local utility, DR now has the potential for adoption on a more distributed scale with the introduction of Internet-connected home and commercial appliances. Of these, the most promising points of use are thermostatically controlled loads (TCLs) such as HVAC systems, refrigerators/freezers, and hot water heaters.

This project focuses specifically on refrigerators and freezers, and on combining sensor data, user control, and real-time marginal emissions data from the WattTime API into a model-based optimal control framework that can provide services to the grid while maintaining food safety. One key goal is to ensure that the required hardware is as low-cost, modular, and easy-to-install as possible, with the hope of future deployment at large scale in the UC Berkeley residence halls.

B. Relevant Literature

The California Independent System Operator offers a useful overview [1] of the challenges associated with having a higher percentage of renewables on the grid.

Mathieu, Dyson, and Callaway's 2012 ACEEE paper [2] details how devices such as refrigerators, AC units, and water heaters can function as a form of energy storage, and provide much-needed services to renewables-heavy electric power systems at low cost.

Unlike many papers involving the use of refrigerators for demand response, which make use of typical thermal resistance and capacitance coefficients in their model, Taneja, Culler, and Dutta's 2010 IEEE paper [3] uses a sensor-based approach to develop a thermal model of their refrigerator.

Another similar sensor-based thermal model is proposed in Mann, Ju, Rosa, and Barido 2015 [4], including a simple model of temperature changes in response to a door-opening event.

DeWitt and Roeschke 2015 [5] includes a large amount of relevant and helpful information about refrigerator parameter and state estimation using sensor data, user behavior prediction, and integration of WattTime and retail rate structure information to perform emissions and cost optimization.

C. Focus of this Study

The objective of this project is to reduce the environmental and economic cost of operating a dorm room refrigerator. To achieve this goal, the refrigerator will incorporate all aspects of a cyber-physical system: gathering sensor data from the environment, processing the data to optimize system behavior, visualizing the data through a website dashboard, and ultimately actuating the refrigerator compressor.

The end result is an energy consumption schedule that minimizes electricity cost and carbon emissions while ensuring that perishables do not expire, and non-perishables are cold when consumed.

II. Technical Description

A. Hardware

The hardware for this cyber-physical system is listed below. The communications connectivity for these devices is explained in the following section.

Bill of Materials:

- Small refrigerator
- Arduino Uno
- Arduino Data Logger Shield
- WiFi-enabled Raspberry Pi
- Edimax Nano USB WiFi Adapter
- Two temperature sensors (ambient and internal)
- Magnetic door position switch
- Solid-state relay to control compressor

B. Connectivity

1) Cyber-Physical System Architecture

To make these modes of operation possible and thereby minimize cost and emissions, the cyber-physical system includes several points of data acquisition and processing:

- **Arduino**
  - Collects data every 60 seconds from the three refrigerator sensors: a temperature sensor inside the fridge, an ambient temperature sensor outside the fridge, and a magnetic switch that records door position. The sensor data is logged to an SD card on the shield, and is sent to a Raspberry Pi over the serial port.
  - Receives data about the state of the compressor from the Raspberry Pi over the serial port, and actuates the compressor accordingly.
- **Raspberry Pi**
  - Receives sensor data from the Arduino through the serial port. Parses the data into four data streams: refrigerator temperature, ambient temperature, door state, and measurement timestamp.
  - Using an Edimax Nano USB WiFi adaptor, the Raspberry Pi connects to the Internet and sends each data stream to the server, hosted at netfridge-jgilles.c9users.io.
  - Pulls data about the compressor state from the server and sends it to the Arduino over the serial port.
- **Server**
  - Written in Python and hosted on Cloud9.
  - Accepts sensor data from Raspberry Pi and compressor actuation command from optimization algorithm. Records data to data streams in a SQLite database.
  - Posts data from SQLite to the web page.
- **Web Page**
  - Written in HTML, CSS, and JavaScript. Receives data from the server.
  - Shows temperature, door state, compressor state, and refrigerator mode as exportable tables and live-updating graphs on a dashboard interface.
  - Allows user input to select the fridge mode. This data is then sent to the server as a POST command.
- **Optimization Algorithm**
  - Written in Python, hosted locally on an Internet-connected computer.
  - Makes GET request to pull most recent refrigerator temperature, ambient temperature, compressor state, and refrigerator mode datapoints from server.
  - Pulls emissions and ambient temperature data (current and forecasted) from the WattTime and WeatherUnderground APIs respectively.
  - Behavioral door-opening patterns, thermal model parameters, and PG&E rate structures are hard-coded in to the script.
  - Using a mixed-integer linear program, creates optimal compressor schedule for 24-hour time horizon that minimizes electricity cost and emissions subject to a model-based temperature forecast, temperature constraints corresponding to the refrigerator mode, and the requirement that the compressor cycle frequency be no less than 5 minutes.
  - Send first compressor decision variable value to server. This corresponds to the optimal compressor state for the next 5 minutes.
  - Algorithm updates with new sensor data and re-runs optimization every 5 minutes.

The communication network between all of these devices can be seen in Figure 1 below.

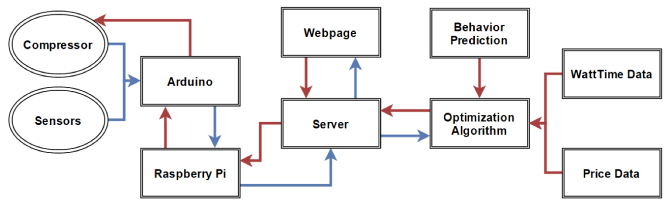


Fig. 1. Cyber-Physical System Architecture.

2) Refrigerator Modes

The refrigerator has three modes, each one representing a common usage of a dorm fridge, which the user can select through a web page. These modes impact the temperature constraints within the optimization algorithm, allowing it more or less flexibility in reducing the cost and carbon emissions of the fridge's performance:

- **Perishable Mode:** In this mode, the temperature of the fridge must always be kept within a tight hysteretic band (34°F - 40 °F) to ensure that the contents do not go bad. Despite the small temperature range, it is still possible to modify the compressor's behavior to minimize carbon emissions and cost.
- **Drink/Non-Perishable Mode:** In this mode, the fridge has a larger temperature range (34°F - 59 °F) during times when the refrigerator is unlikely to be opened, and ensures that contents will be cold (34°F - 40 °F) when its contents are most likely to be consumed. Prior door-opening patterns were recorded and analyzed to create a predictive model that dictates the allowable fridge temperature range at any given time of the day. Then, the

optimization algorithm is used to actuate the compressor only when necessary, and in a way that minimizes carbon emissions and cost.

- **Empty Mode:** In this mode, the compressor remains off. In a residence-hall setting, this feature alone could result in significant energy savings, and the computer-based control app provides convenience for the user.

A decision flowchart for these three modes within the context of the cyber-physical system architecture can be seen in Figure 2 below.

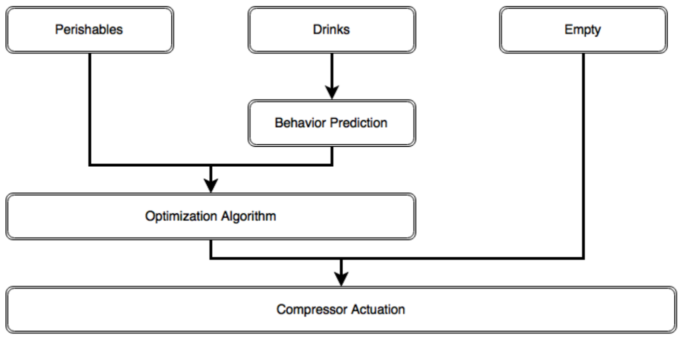


Fig. 2. Refrigerator Modes Decision Flowchart.

C. Data Analysis

Analysis of two weeks of door-opening and temperature data collected from an in-use dormitory fridge was used to create a statistical model of user consumption behavior, and to estimate parameter values for the thermal model used in the optimization algorithm.

1) Behavior Prediction

When in non-perishables mode, the fridge uses behavior prediction to ensure that the fridge is cold only when the user is most likely to take drinks out of the fridge. Using the sample data, an algorithm predicts when the fridge is likely to be opened.

First, the timestamps of all refrigerator-opening events are recorded to an array $openTime$. For simplicity, a current time $currTime$ is set to be a few hours after the most recent opening of the fridge. The age of each $openTime$ in days was recorded by comparing $currTime$ with $openTime$, and was recorded to an array $openAge$.

Each $openTime$ was analyzed to determine when in a 15-minute discretized 24-hour period the door was opened. This value was then added to a score array using the following equation:

$$weightedScore(i) = score(i) + \alpha^{age(j)} \tag{1}$$

i is an integer between 0 and 95, representing a 15-minute time period within the 24-hour day.

j is the index of each value in the age array.

For this analysis, the decay factor α is set to 0.9. This factor places a large emphasis on recent door-opening events and a smaller emphasis on more distant events. For example, opening the door of the fridge one month ago is counted as only 0.042 towards the score.

The score was then distributed over a 2-hour period using the following equation:

$$\begin{aligned}&distScore(i) = 0.25 \cdot score(i-3)+ \\ &0.5 \cdot score(i-2) + 0.75 \cdot score(i-1) + score(i)+ \\ &0.75 \cdot score(i+1) + 0.5 \cdot score(i+2) + 0.25 \cdot score(i+3)\end{aligned} \tag{2}$$

i is the index of each time period in $score$.

The maximum score possible for any time period, $maxScore$, corresponding to the fridge being opened at all prior timesteps, is calculated as follows for α <1.

$$maxScore = \int_{-\infty}^{0} \alpha^x dx = -\frac{1}{ln(\alpha)} \tag{3}$$

The distributed score is normalized by dividing by $maxScore$ such that:

$$normScore = \frac{distScore}{maxScore} \tag{4}$$

A threshold for an acceptable level of score for which the fridge should be cold is calculated below and corresponds to the fridge being opened about 3 times over the past week.

$$threshold = \frac{\alpha^7 + \alpha^5 + \alpha^3}{maxScore} \tag{5}$$

The distributed score is then compared to the $threshold$. If the distributed score $distScore$ is larger than the $threshold$, the fridge will be constrained to tighter temperature constraints. Otherwise, it will be constrained to the wider temperature bounds.

The $distScore$ is plotted in the histogram Figure 3 below, and the $threshold$ is shown as a horizontal dotted line.

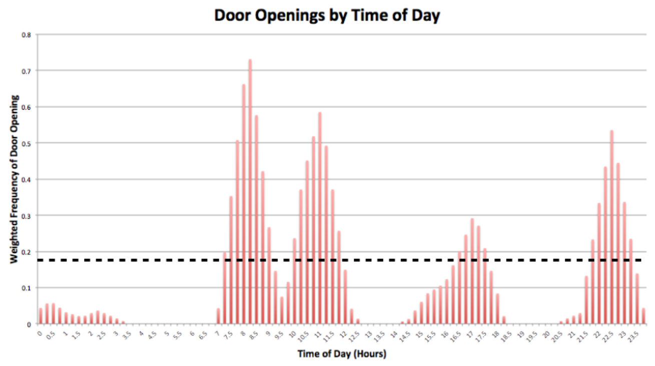


Fig. 3. Door Openings Histogram.

As an important note, although this analysis was done for two weeks of collected data, the analysis is set up such that it can be updated in real time with each fridge opening. This would allow the system to learn over time when the fridge is likely to be used and, therefore, should be cold.

2) *Thermal Model Parameter Estimation*

Data was collected over a two-week period using an Arduino in a Berkeley dorm fridge. The fridge was the same model as was used for this project, and was utilized by the room's three occupants. Just as in the final system, fridge temperature, outside temperature, and door state was collected from the fridge. After the two-week period, the data was analyzed to determine when the compressor was on, but also when the user was most likely to use the fridge.

A simple algorithm was used to determine when the fridge was on; if the temperature in the fridge was decreasing, it was assumed that the compressor was on; otherwise, it was assumed that the compressor was off. This analysis revealed that the compressor simply turned on at regular intervals regardless of the current temperature inside the fridge.

The fridge temperature, outside temperature, and compressor state was sent into an algorithm to determine parameters that could be used to predict future temperatures of the fridge. A linear equation of the following form was used:

$$T_f(k+1) = \theta_1 T_f(k) + \theta_2 T_a(k) + \theta_3 s(k) \tag{6}$$

$T_f(k)$ is the temperature of the refrigerator at timestep k, $T_a(k)$ is the ambient temperature of the refrigerator at timestep k, and $s(k)$ is the on/off state of the compressor. θ_1, θ_2, and θ_3 are parameters corresponding to the thermal properties of the refrigerator.

The first week of fridge data was processed by an online linear regression model to determine the thermal model parameters. The data from the second week was not analyzed so that the results of the parameter estimate could be tested.

For the test, the parameters from the regression analysis were used to predict the temperature of the fridge over the second week, given the initial refrigerator temperature, the temperature in the room, and the compressor state. The results are plotted below in Figure 4:

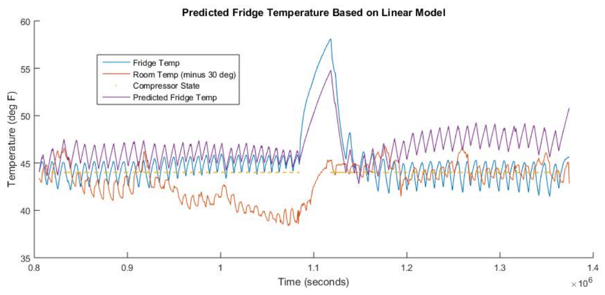


Fig. 4. Modeled vs. Measured Refrigerator Temperatures.

As the plot shows, the linear model closely matched the measured temperature values. Given only the starting temperature of the fridge, over a one week period, the predicted temperature only varied from the actual temperature by a few degrees.

3) *Optimization Algorithm*

The results of the behavior prediction and thermal parameter estimation, along with PG&E rate data, were hard-coded into the optimization algorithm. Current refrigerator mode and temperature were pulled from the server, and marginal emissions and ambient temperature predictions were pulled from the WattTime and Weather Underground APIs.

The optimization algorithm used is a slight variation on the mixed-integer linear program developed in DeWitt and Roeschke 2015 [5]:

$$\min_{s(k),T_f(k)} \sum_{k=0}^{N-1} (\lambda c(k) + (1-\lambda)e(k)) P s(k) \tag{7}$$

Subject to:

$$T_f(k+1) = \theta_1 T_f(k) + \theta_2 T_a(k) + \theta_3 s(k) \tag{8}$$

$$T_{f,min,on} \le T_f(i) \le T_{f,max,on} \tag{9}$$

$$T_{f,min,off} \le T_f(j) \le T_{f,max,off} \tag{10}$$

$$T_f(0) = T_{f,o} \tag{11}$$

$$\begin{aligned} 0 \le\; & s(k-5) + s(k-4) + s(k-3) \\ & + s(k-2) - 4s(k-1) + 5s(k) \le 5 \end{aligned} \tag{12}$$

$$s(k) = [0,1] \tag{13}$$

$$\begin{aligned} & i \in k \mathrel{\widehat{=}} [7:15-9:00] \cup [10:00-11:45] \cup \\ & [16:30-17:30] \cup [21:45-23:15] \end{aligned} \tag{14}$$

$$\begin{aligned} & j \in k \mathrel{\widehat{=}} [0:00-7:15] \cup [9:00-10:00] \cup \\ & [11:45-16:30] \cup [17:30-21:45] \cup [23:15-24:00] \end{aligned} \tag{15}$$

λ is a multi-objective optimization constant that determines the relative weights of the electricity and carbon cost functions. For this project, a value of 0.6 was used, giving carbon intensity a slightly higher influence over the end result. This stems from the lack of significant variability in retail electricity rates, and any potential mismatch between predetermined electricity rates and real-time grid conditions.

$c(k)$ and $e(k)$ are the carbon and electricity cost functions respectively. PG&E's E-20 summer time-of-use rates for large commercial customers was used, as this appears to be the rate paid by the University of California for electricity. P is the energy consumption of the compressor, assumed to be 0.1 kW when in use.

The refrigerator temperature T_f changes in time according to a linear model using the estimated parameter values. The refrigerator temperature must stay within the set constraints for each timestep. The initial modeled temperature is equal to the most recent sensor value pulled from the server. The compressor is not allowed to cycle more frequently than once per 5 minutes.

i represents times when the refrigerator is likely to be opened. The refrigerator temperature is constrained to the range [34°F - 40 °F], for both the perishable and non-perishable refrigerator modes. j represents times when the refrigerator is not likely to be opened. The refrigerator temperature is constrained to the range [34°F - 59 °F] when the refrigerator is in non-perishable mode.

D. *Visualization*

The first step of the visualization is logging on to a personal fridge website, seen in Figure 5. If deployed at scale, one login and password would be assigned to each room, and would need to be stored in a separate database.

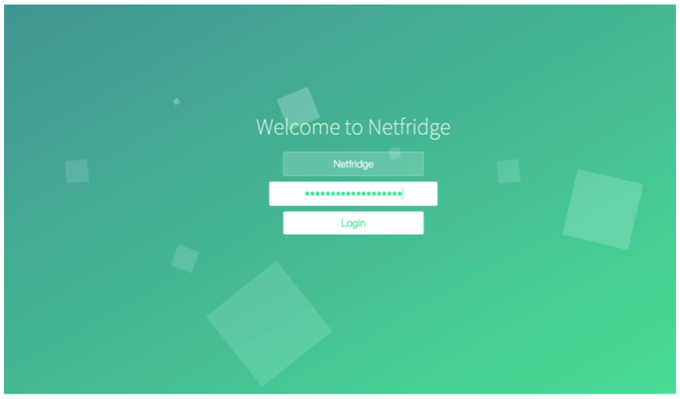


Fig. 5. Web Dashboard Login Page.

Once the user has logged on, the website redirects to a dashboard interface, which provides the user with an overview of the site, along with a sidebar containing further details about the user's refrigerator. Perhaps the most significant of these is a real-time data stream found on the Plots page. Figures 6 shows a sample plot of interior temperature data.

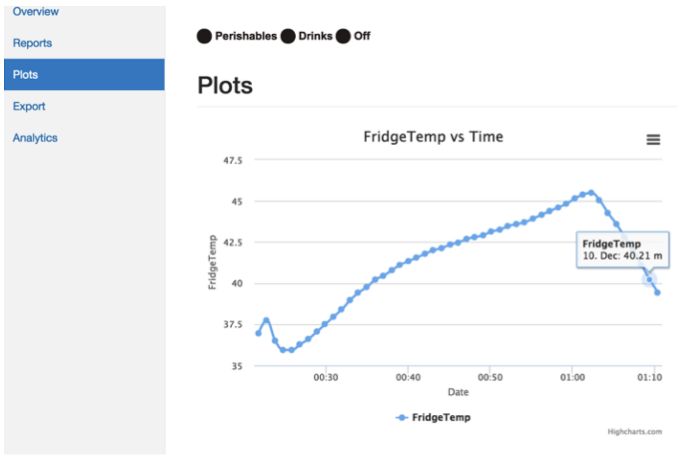
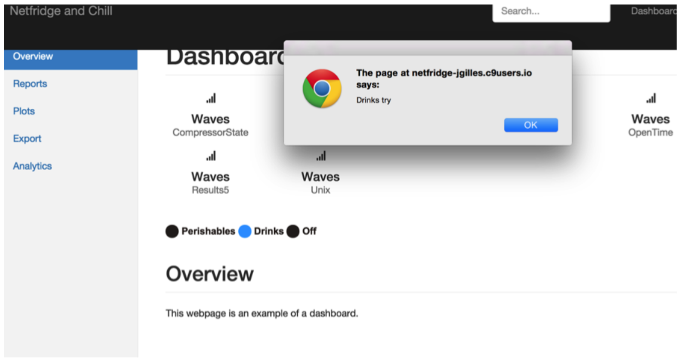


Fig. 6. Refrigerator Temperature Plot.

The dashboard features radio buttons corresponding to the three refrigerator modes: Perishables, Drinks, and Off. As previously described, these three different options were chosen to most accurately reflect the variety of contents that could be placed in a dorm fridge. When one of those modes is clicked by the user, a notification will pop up on the present browser confirming the selection, as seen in Figure 7.



The page at netfridge-jgilles.c9users.io says:

Drinks try

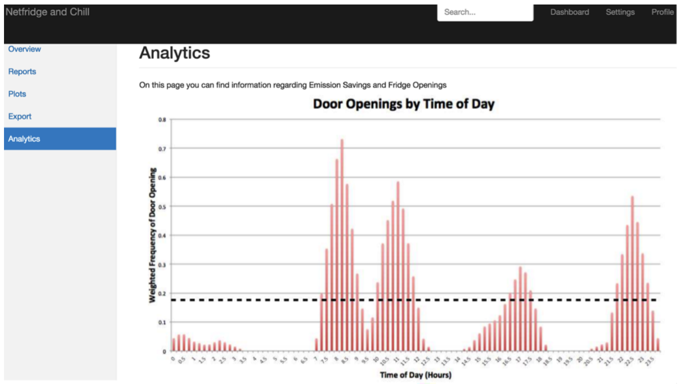
Overview

This webpage is an example of a dashboard.

Fig. 7. Refrigerator Mode Selection Notification.

This data is then sent to the server. This is done through a POST request run by jQuery and Ajax. The selection can then be read by the optimization algorithm.

Additionally, the Analytics page (Figure 8) is designed to show a histogram describing historic usage patterns for each individual fridge. This data is an important parameter used in the optimization algorithm.



Analytics

On this page you can find information regarding Emission Savings and Fridge Openings

Fig. 8. Analytics Page.

All parts of the visualization were created using HTML, CSS, Javascript, and jQuery. Aside from the images shown, the website also includes an Export sub-menu which displays a table containing the most recent data points.

III. Discussion

The original motivation for this cyber-physical system project came from the authors' personal experience with dormitory refrigerators in their freshman and sophomore years. Although dorm fridges do prevent perishables from expiring, their unsophisticated, timer-based compressor scheduling controls mean that they consume far more energy than their role requires. By contrast, the NetFridge developed in this project uses data from WattTime API, PG&E utility rates, prior fridge usage data, real-time fridge sensors, and user preferences to optimize how to best operate the fridge.

Although dorm fridges are set to store perishable food 24/7, this rarely matches the actual fridge contents, as students living in the dorms generally only have perishable food when there are leftovers from the dining hall. The rest of the time, the fridge is storing drinks, or nothing at all.

NetFridge's combination of behavior prediction, user mode selection, and compressor optimization allow for significant energy reduction, while preserving core food-safety functionality.

Although significant progress was made this semester, there are several opportunities for further improvement and study that could be implemented by future CE 186 students:

- Install the NetFridge for an extended period of time in actual dorm conditions, and compare energy savings, carbon emissions and demand reduction to a control group of non-optimally-actuated refrigerators.
- Incorporate real-time locational marginal (wholesale) pricing as an alternate cost function, and to estimate the grid infrastructure benefits of smart fridge.
- Improve ambient indoor temperature forecasting by using previous days building temperature data, ambient temperature sensor readings, and the Weather Underground API's forecast.
- Use Kalman Filter state-estimation algorithm to re-compute the refrigerator model parameters every time the refrigerator is opened. Sensor data suggests that the thermal capacitance changes significantly when food or drinks are added or removed from the refrigerator.
- Update optimization bounds in real time with stream of past behavior data.
- Host optimization on the server, instead of running it locally on a computer.
- Replace Arduino and Raspberry Pi with lower-cost hardware to improve scalability.
- Create an enclosure for all sensors and electronics that protects them from damage while allowing for easy installation.
- Allow users to set future temperature setpoints, overriding or augmenting the optimization.

IV. Summary

The automated-demand-response refrigerator incorporates all features of a cyber-physical system as follows:

1) **Infrastructure:** The fridge interacts with California's electric grid through the minimization of peak demand and carbon emissions.
2) **Sensing:** The cyber-fridge senses the conditions of the fridge through two temperature sensors, a door position switch, and the compressor state. In addition, the fridge uses past behavior to ensure that non-perishables are cold when consumed.
3) **Actuation and Decision Support:** The main point of actuation in the system is the compressor on the fridge, which turns off or on depending on refrigerator state and grid conditions. There is also user decision support in the data displayed to the user through the web page, informing the user's choice of fridge mode.
4) **Connectivity:** The cyber-fridge is connected to the user through the web app, is informed about the state of the grid through WattTime data, and impacts the grid by providing demand response services.
5) **Data Analysis:** The majority of the data analysis happens in the Python processing stage. Here, all of the data streams are integrated and analyzed to determine whether the compressor should be on or off.

An estimate provided in Dewitt and Roeshke 2015 [5] suggests that NetFridge may be able to provide up to 68% electricity savings per dorm room, assuming that is is run in non-perishables mode for 6 hours per day on average. Each fridge uses approximately 200 kWh annually; if each of the 1,700 dormitories at UC Berkeley were outfitted with a NetFridge, total energy savings could amount to approximately 271,660 kWh. Using an average electricity cost of $0.10/kWh, total annual savings for the campus aggregate to $27,166. Given that this technology is scalable, implementation across other campuses is easy to imagine, and could contribute to further economic and environmental benefits.

V. Acknowledgements

The authors would like to thank Professor Scott Moura and Eric Burger for providing the hardware used in this project, as well as their assistance with server setup and many other aspects of the software development process. We also would like to thank Matthew Roeshke and Zoltan DeWitt for allowing us to make use of their optimization code, and for helping us install the solver required to run it. In addition, thanks to Anna Schneider from WattTime for her assistance in accessing the WattTime API. Finally, we would like to thank Diego Ponce de Leon Barido, Javier Rosa, and Sandro Iacovella, whose FlexBox refrigerator sensing and control research project inspired our own.

References

[1] California Independent System Operator, 'What the duck curve tells us about managing a green grid', 2013.
[2] J. Mathieu, M. Dyson, and D. Callaway, 'Using Residential Electric Loads for Fast Demand Response: The Potential Resource and Revenues, the Costs, and Policy Recommendations', American Council for an Energy-Efficient Economy, 2012.
[3] J. Taneja, D. Culler, and P. Dutta, 'Towards Cooperative Grids: Sensor/Actuator Networks for Renewables Integration', Institute of Electrical and Electronics Engineers, 2010.
[4] R. Mann, J. Ju, J. Rosa, and D. Barido, 'Sensor-based validation of thermostatically controlled load refrigerator models', Energy Modeling, Analysis, and Control Group, 2015.
[5] Z. DeWitt and M. Roeschke, 'Optimal Refrigeration Control for Soda Vending Machines', Energy, Controls, & Applications Lab, 2015.