Real-Time Human Activity-Based Energy Management System Using Model Predictive Control

Chiyang Zhong, Jingfan Sun, Jiahao Xie, Santiago Grijalva, and A. P. Sakis Meliopoulos
Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA, USA
Email: {c.zhong, jingfan, orbitx}@gatech.edu, sgrijalva@ece.gatech.edu, sakis.m@gatech.edu

***Abstract*—The challenges of energy efficiency and comfort management of intelligent homes and buildings are usually tackled with methods relying on historical data and a large number sensors. In this paper, we propose a real-time human activity-based energy management system (HAEMS), which tracks and processes human movement, and achieves device control based on real-time data with a significantly reduced number of sensors. A human activity detection algorithm and a model predictive control scheme are developed and implemented to optimally manage energy. A multi-objective optimization problem is formulated to minimize electricity cost and control temperature for thermal comfort. The HAEMS is deployed in a scaled-down laboratory setup and the performance is evaluated in an embedded system and hardware environment. Experimental results show that this system is able to optimize both electricity cost and thermal comfort.**

***Index Terms*—Energy Management System, Human Activity Detection, Model Predictive Control.**

I. Introduction

According to a report from the Department of Energy, residential buildings account for about 38% of the electricity consumption in the US in 2016 [1], representing a major opportunity for energy efficiency and energy management improvements. The need to better utilize electrical energy and at the same time maintain customer satisfaction and comfort calls for novel designs of energy management systems.

Intelligent buildings focusing on energy efficiency and comfort management based on user activity have been discussed in previous literature. Binary detection of residential building occupancy using hidden Markov models has been proposed [2], where a method is developed that achieves energy saving by determining the probability of the residents being at home. In order to reduce electricity usage, reference [3] proposes an algorithm to control a programmable home thermostat using occupancy data collected by sensors deployed in rooms. A soft-computing method using neural networks for illumination control is developed in [4]. In order to track individual energy consumption, authors in [5] use human-equipped proximity sensors that interact with other sensors deployed near home appliances. The data is used to report the energy consumption profile of each appliance for the users. In [6], a sliding window approach, based on streaming data from a large number of sensors placed in a limited space, is developed to perform activity recognition for residential building control. References [7] and [8] have proposed a system that uses state estimation methods and a high fidelity electro-thermal model of a house to monitor the actual temperatures of the various rooms of a house. The approach provides the real time model of the house as well as the operating conditions of the house which are used to define an optimization problem. Zero inconvenience constraints are added to the optimization problem. Its solution provides the optimal control of house resources to optimize the cost of electricity without inconveniencing the house occupants.

Although a lot of work has been done in this area, previous approaches mostly rely on pre-collected data [2]–[4] or use too many sensors, most of which are temperature or proximity sensors [5], [6]. In this paper, we propose a real-time human activity-based energy management system (HAEMS) with a significantly reduced number of sensors. In this paper human activity is measured by human movement. By utilizing human movement data in real time based on a motion sensing input device, the system is able to respond more accurately to changes in activity driving energy control. In addition, the installation of the system can be streamlined, since users may already have the sensor at home. In order to demonstrate the proposed technology, we use Microsoft Kinect as the motion sensing input device. The Kinect is coupled with a control logic that manages the temperature for users' comfort and simultaneously help them save money on electricity bills. This technology can be applied directly to improve gaming experience for gamers, by adjusting the room temperature in a predictive manner when they use Kinect to play games.

In Section II of this paper, we present an overview of the HAEMS and a scaled-down hardware testbed to evaluate the performance of the proposed system. Section III provides an introduction to Random Forest Regression (RFR) and applies it to process human activity intensity data. In Section IV, we discuss the model predictive control (MPC) scheme, which utilizes the information in real time to control the scaled-down system. The experimental hardware setup and system parameters are presented in Section V. In Section VI, the performance of the system and experimental results are discussed. Section VII presents conclusions and future work.

II. System Overview

Let us consider a residential building with an air conditioner (AC). The AC is used to cool down the building for comfort of the residents. Let us assume that the outdoor ambient temperature is higher than the indoor temperature. There is heat exchange between indoor and outdoor environments through the walls, which increases the indoor temperature.

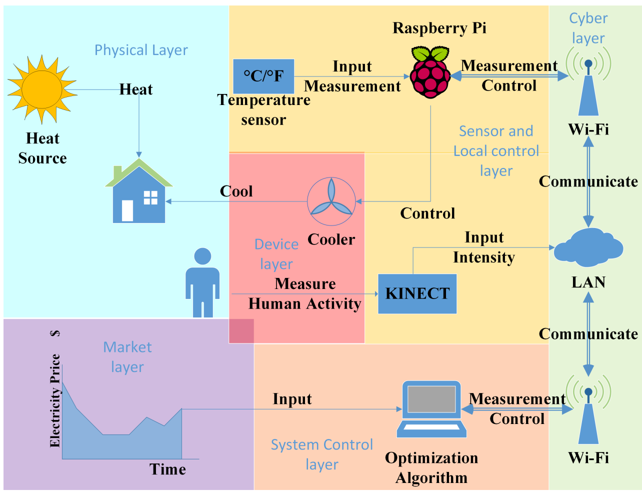

Fig. 1. HAEMS design overview.

In this paper, we use a small system containing a heat source and a cooler to simulate the cooling of a home environment, as illustrated in Fig. 1. According to the distributed control architecture for smart grid [9], the components in HAEMS are categorized into five layers as shown in Table I. Note that there are only two sensors in this implementation, which makes the system cheap and easy to install.

TABLE I. HAEMS layers overview.

Layers	Components
Market Layer	Electricity Price Function, Cost Optimization
System Control Layer	Model Predictive Control
Cyber Layer	Wi-Fi, LAN
Local Control Layer	Kinect, Temperature Sensor, Raspberry Pi
Device Layer	Cooler

We use the heating effect of a resistor connected to a power supply to simulate the increase in indoor temperature. The resistor is attached to a hydro CPU cooler, as a first approximation of an AC system. This scaled-down physical system is driven by the following closed loop control. The input of the HAEMS is the temperature set-point that follows human activity intensity detected by a Kinect. The feedback is the controlled temperature that is monitored by a temperature sensor. Based on these data and a pre-acquired electricity price function, a model predictive control (MPC) scheme is designed to minimize the cost of energy consumption of the cooler over a specified time range. The output of this algorithm is a control signal that adjusts the speed of the cooler fans. A detailed system description of the HAEMS is given in the next sections. In summary, we need to perform the following steps to achieve real-time energy management.

1) Detect human activity with a Kinect.
2) Identify the model to be controlled and predict the necessary control actions of the system using the MPC by optimizing electricity cost and thermal comfort.
3) Implement the optimal control actions.

III. Human Activity Detection

Different from the traditional home energy management system (HEMS), HAEMS uses human activity intensity as a key factor to adjust the control of the system. Due to high household availability and relatively low price, we use a Kinect sensor to capture the human skeleton information with Random Forest Regression (RFR) algorithm.

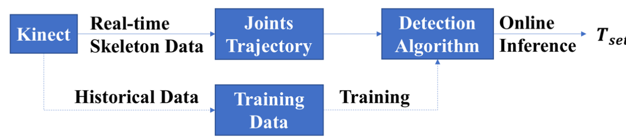

Fig. 2. Human activity detection module overview.

A general overview of the detection module is provided in Fig. 2. In the training phase, movements of human joints are captured and labeled with corresponding levels of intensity. These data are fed into the detection algorithm to tune the model parameters. Once the training is done, the human activity detection module is ready for on-line inference. Real-time skeleton data are acquired in a speed of 30 fps (frames per second) and transformed into human joint trajectories, which are used as inputs of the pre-trained detection algorithm. As a result, the temperature set-point T_{set} is generated for temperature control.

A. Joint Trajectory

Human raw skeleton data are captured frame by frame from the Kinect. Each frame contains 3-dimensional information of 25 joints of a body, including head, neck, shoulders, elbows, wrists, hands, spine (shoulder, mid, and base), hips, knees, ankles, and feet.

In this study, 11 joints are selected for further analysis, including head, shoulders, elbows, wrists, hips, and ankles. The movement of these joints can be used to extract the human activity intensity, which is a positive real number indicating how much the person has moved in the past time interval. The more the target person moves per unit time, the higher the intensity. Examples of time-series trajectories from 4 joints are provided in Fig. 3.

With 11 joints involved, the incoming raw data are stored in a $33 \times n$ matrix M, where 33 represents 3-dimensional information from 11 joints and n is the number of frames. In order to get rid of noises, a sliding window Gaussian filter is implemented to smooth the trajectories. The window length l is set to 5 in this case. The convolution kernel c of this filter

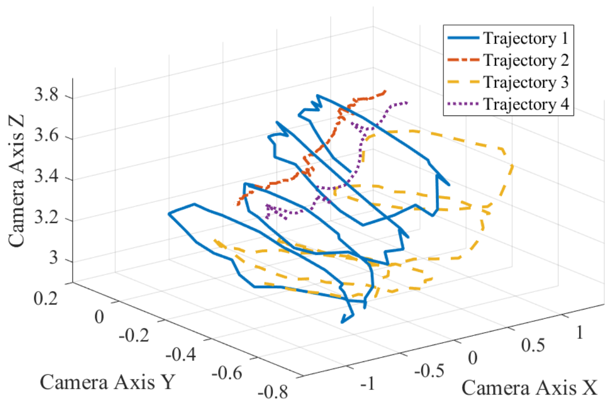

Fig. 3. Time-series 3-dimensional trajectory of 4 joints.

is chosen to be $0.0625 \cdot [1, 4, 6, 4, 1]$. The kernel is applied to every row of M by calculating the dot products. The smoothed trajectory data are saved in a new matrix $\widetilde{M}$, which is fed to the detection algorithm every 0.5 seconds. Each element of $\widetilde{M}$ is obtained in equation (1), where i and j are the row and column indices, respectively.

$$\widetilde{M}_{i,j} = \sum_{k=1}^{5} c_k \times M_{i,j-3+k} \quad (1)$$

B. Detection Algorithm

The RFR algorithm is adopted here to detect human activity intensity given the joint trajectory matrix $\widetilde{M}$. RFR is an ensemble learning method that works by forming a number of decision trees and generating the output from the mean prediction of each individual tree. The detailed algorithm of RFR is provided in Algorithm 1. For model training, 100 samples of $\widetilde{M}$ are created. For each $\widetilde{M}$, a relative movement matrix is calculated. Then, n_{tree} bootstrap subsets are randomly sampled with replacement. These subsets are used to grow regression trees. Starting from a single node, the algorithm searches over all binary splits of all variables for the one that will reduce standard deviation (SD) as much as possible. If the largest reduction (SDR_{max}) in SD is less than some threshold δ, the growth stops. Otherwise, the node is split into two new nodes where the growth procedure repeats recursively.

During the on-line inference, a coming $\widetilde{M}$ is fed into the pre-trained n_{tree} regression trees. The temperature set-point T_{set} is computed by taking average of the predictions $z_k \in [0, 1]$ and doing linear transformation, which is given by

$$T_{set} = T_{base} - T_{range} \times \frac{1}{n_{tree}} \sum_{k=1}^{n_{tree}} z_k \quad (2)$$

where T_{base} and T_{range} are the base temperature and set-point range, respectively. It can be noticed from (2) that less intensity yields a higher T_{set}.

Currently, the detection algorithm can only recognize the activity of one person. We plan to extend the functionality to process more people in the future.

ALGORITHM 1: Random Forest Regression

Input: Joint trajectory matrices $\widetilde{M}$
Output: Temperature set-point T_{set}
Training:
$n_{tree} = 10$; $\delta = 0.1$;
Calculate relative movement matrix R by
    $R_{i,j} = \frac{1}{n-1} \sum_{k=1}^{n-1} |\widetilde{M}_{i,j+1} - \widetilde{M}_{i,j}|$
Randomly draw n_{tree} bootstrap samples from original data;
for *each bootstrap iteration* **do**
    // Grow a regression tree
    for *each node in tree* **do**
        while $SDR_{max} > \delta$ **do**
            for *each binary split s* **do**
                $SDR = SD(original) - \sum_{s \in X} P(s) \cdot SD(s)$
            end
            Search for the split with maximum SDR_{max};
        end
    end
end
Inference:
Take average from predictions of n_{tree} trees;
Transfer human activity intensity to temperature set-point T_{set};

IV. Model Predictive Control

Model predictive control (MPC) is utilized to control the temperature. This is a closed-loop control based on iterative, finite-horizon optimization of a model. At each time step k, the states are measured and a multi-objective mixed integer linear programming (MILP) problem takes place for a short-time horizon in the future $[k, k+n]$, where n is the number of time steps in the horizon. Then, only the first step of the control strategy is implemented. After that, the horizon shifts forward by one time step and the process is repeated.

With the MPC, we want to minimize electricity cost of the cooler over a certain time period. Meanwhile, the temperature controlled should be kept in or driven to a range for the comfort of customers. We use the MPC for more accurate control of the temperature, as it recomputes the optimal control signals after periodically resetting the initial conditions of the optimization problem to the true states of the system.

A. System Identification

The fan speed of the hydro CPU cooler is controlled by a pulse-width modulation (PWM) signal. The larger the duty cycle of the PWM signal, the faster the fans spin, and thus the quicker the temperature drops. In order to run the MPC, we have to acquire the model of the physical system and its parameters, so system identification is needed. In this paper, we assume that the system is linear. Denote t as time, D as duty cycle, T as temperature, and P as power consumption of the cooler. Equations $\partial T/\partial t = \beta_0$ and $\partial/\partial t = \gamma D + \beta_1$ demonstrate the temperature variation when the cooler is OFF and ON, respectively. Let $P = 0$ when the cooler is OFF and $P = \alpha D + \varphi$ when the cooler is ON.

We did experiments on the physical system to determine parameters α, β_0, β_1, γ, and φ by curve fitting the recorded data.

B. Electricity Cost Minimization

The first objective is to minimize the energy consumption cost of the cooler. The objective is given by

$$\min \quad \sum_{k=1}^{n} c_k(\alpha D_k + \varphi u_k)\Delta t \tag{3}$$

where c_k is the electricity price at time step k and Δt is the time interval between two consecutive time steps. Binary variable u is introduced to denote the OFF and ON status of the cooler with values 0 and 1, respectively.

With the model parameters pre-acquired from Section IV-A, the relationship between temperature and duty cycle can be written as

$$T_{k+1} = T_k + \gamma D_k \Delta t + (\beta_1 - \beta_0)u_k \Delta t + \beta_0 \Delta t \tag{4}$$

where $k = 1, \ldots, n$. At the first step of each optimization horizon, the measurements provide the following initial values, where subscript $init$ represents the initialization.

$$D_1 = D_{init}, \quad T_1 = T_{init}, \quad u_1 = u_{init} \tag{5}$$

It is assumed that the change of duty cycle between consecutive steps is limited by ϵ and $D \in [D_{min}, D_{max}]$ when the cooler is ON. Hence, the following constraints are required.

$$-\epsilon \leq D_{k+1} - D_k \leq \epsilon \qquad \forall k = 1, \ldots, n-1 \tag{6a}$$

$$D_{min}u_k \leq D_k \leq D_{max}u_k \qquad \forall k = 2, \ldots, n \tag{6b}$$

C. Temperature Control

The second objective of the optimization problem is to keep the temperature within a desired range $[T_{set} - \delta, T_{set} + \delta]$ or pull the temperature towards the range if it is outside, where T_{set} is the temperature set-point generated from the human activity intensity detected.

We consider two possibilities, when T is within the desired range and when T is not. Specify binary variable x_k, $\forall k = 2, \ldots, n+1$ so that

$$x_k = \begin{cases} 0 & T_{set} - \delta \leq T_k \leq T_{set} + \delta \\ 1 & otherwise \end{cases} \tag{7}$$

Since $x_k = 1$ introduces a disjunctive constraint, $y_k \in \{0, 1\}$ is used to give the following statement.

$$y_k = \begin{cases} 0 & T_k > T_{set} + \delta \\ 1 & T_k < T_{set} - \delta \end{cases} \tag{8}$$

Note that (8) only matters when $x_k = 1$. By using the big M method, (7) and (8) impose the following constraints on temperature for $k = 2, \ldots, n+1$.

$$T_{set} - \delta - M_1 x_k \leq T_k \leq T_{set} + \delta + M_1 x_k \tag{9a}$$

$$T_k \geq T_{set} + \delta - M_1(1 - x_k) - M_2 y_k \tag{9b}$$

$$T_k \leq T_{set} - \delta + M_1(1 - x_k) + M_2(1 - y_k) \tag{9c}$$

M_1 and M_2 are large values chosen carefully so that extra constraints are not imposed. Now consider $x_k = 1$, meaning T_k is outside of the specified range and the goal is to pull the temperature towards it. This yields a minimax problem given by

$$\min \quad \max\{T_k - T_{set} - \delta, T_{set} - \delta - T_k\} \tag{10}$$

which for $k = 2, \ldots, n+1$ is equivalent to the minimization problem given in (11). Again, the big M method is utilized and M_3 is a large number selected carefully.

$$\min \quad h_k \tag{11a}$$

$$\text{s.t.} \quad T_{set} - \delta - T_k - M_3(1 - x_k) \leq h_k \tag{11b}$$

$$T_k - T_{set} - \delta - M_3(1 - x_k) \leq h_k \tag{11c}$$

$$h_k \geq 0 \tag{11d}$$

D. Optimization Problem

A multi-objective optimization problem is formulated in this section. As described in Section IV-B and Section IV-C, the objectives of the optimization problem for the MPC include cost minimization and temperature control. We assign a heavy weight w on variable h so that it does not affect the control strategy generated by minimizing the energy consumption cost. Hence, together with the objective given in (3), the overall optimization problem is an MILP problem that can be formulated as

$$\min \quad \sum_{k=1}^{n} c_k(\alpha D_k + \varphi u_k)\Delta t + \sum_{k=2}^{n+1}(w h_k) \tag{12a}$$

$$\text{s.t.} \quad u_k, x_k, y_k \in \{0, 1\} \tag{12b}$$

$$(4) - (6), (9), (11b) - (11d)$$

This optimization problem is implemented in the algorithm to run at every time step. The model states in the first step ($k = 1$) of each horizon are the initial values measured, while those in the second step ($k = 2$) are implemented as commands. The algorithm is implemented in MATLAB on a PC.

For future improvements, the model can be assumed quadratic in system identification and quadratic programming may be considered in the control process to increase the accuracy of temperature estimation.

V. Experimental Setup

In this section, the hardware testbed shown in Fig. 4 is discussed in detail. The scaled-down system is implemented in this paper because it captures the main features of a home environment. The lower part of Fig. 4 shows the main parts of the system. The average heating of a room can be simplified to a heat source, while the cooler works similarly to a home AC system.

A. Heat Source and Cooling System

In the experimental setting, the heat source is simulated by resistors, which are energized using a DC power supply. The power consumption of the resistors is kept constant during the experiment. The cooler is a hydro CPU cooler, which consists of a pump and two fans. In order to reduce the influence of air temperature, the heat source and the cooler are both attached to a heat sink. The temperature of this heat sink can be viewed as the room temperature in a home environment.

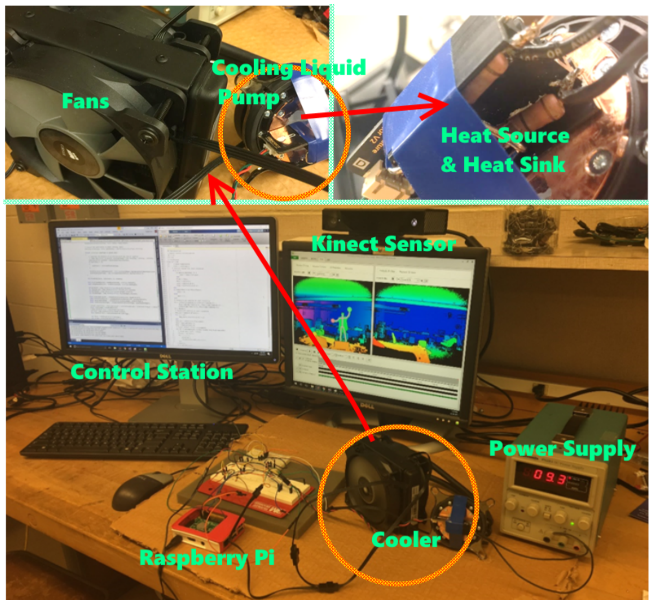

Fig. 4. Experimental hardware testbed.

B. Local Control

In this experiment, DS18B20 [10] measures the temperature of the heat sink in real time. This digital temperature sensor measures temperature from $-55°$C to $+125°$C, covering the temperature range in the test. Raspberry Pi plays a key role in the local control layer. First, it collects data from the temperature sensor via 1-Wire® Interface, and pre-processes the data to remove bad data and noise. Then it generates PWM to control the cooler fans. Raspberry Pi also exchanges data with PC via Wi-Fi and TCP/IP. The application layer protocol of this connection is remote procedure call (RPC). RPC allows computer programs to call procedures to execute in Raspberry Pi. The operating system running on Raspberry Pi is Raspbian, a Debian-based Linux distribution. The human motion detection is fulfilled by Kinect, which is connected with PC.

C. Optimization Parameters

The system is tested over 100 prediction horizons, with each horizon containing 7 time steps (5 sec/step). T_{set} is obtained from (2) in real time within a range between 40°C to 45°C and kept constant in every 5 steps. This temperature range is chosen based on the features of the heat source. The electricity price function is randomly generated. A bound of $\pm 1°$C relative to T_{set} is used. Other important parameters used in the experiment are provided in TABLE II.

TABLE II. Experimental system parameters.

Parameters	Values	Parameters	Values
α	2.781 W	γ	-0.04929°C/s
φ	1.02 W	ϵ	0.3
β_0	0.010148°C/s	D_{min}	0.2
β_1	-0.03702°C/s	D_{max}	0.9

VI. Results and Discussion

Fig. 5 shows the performance of the proposed system. The estimated and measured temperature waveforms behave similarly and follow the set-point T_{set} generated by the human activity detection algorithm. As shown in Fig. 5(a), the objective described in Section IV-C has been achieved. The temperature is kept within the desired range, and drawn towards the range when it is outside. Fig. 5(b) displays the duty cycle command from the optimization. Due to the optimization constraint (6b), the variation of duty cycle in adjacent steps is limited to 0.3 to prevent the fan speed from changing abruptly. When the temperature enters the acceptable range, the duty cycle remains zero. Additionally, the optimization uses the electricity price to regulate the control signal to make sure the total cost is low, which is consistent with the goal of electricity cost minimization defined in Section IV-B. Fig. 5(c) shows the normalized electricity price considered in this project. The base value of price is 5.65 ¢/kWh. In the high price period (time steps 40-60), the temperature is kept near the upper bound, which helps minimize the cost.

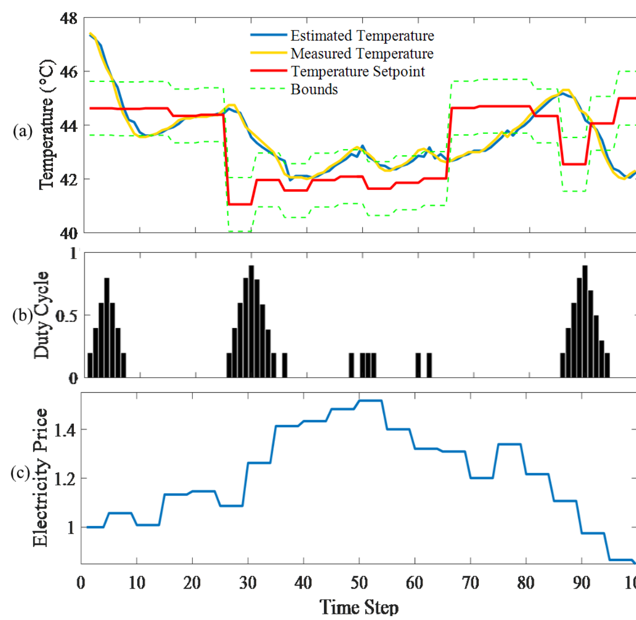

Fig. 5. Experimental results.

Unlike our approach of both optimizing electricity cost and thermal comfort (Case 1), many existing papers consider only controlling temperature for comfort without taking costs into account [11], [12]. Hence, simulations using only a comfort objective function (Case 2) were conducted as well. In this case, the optimization objective in (12a) becomes

$$\min \sum_{k=2}^{n+1} (w h_k) \tag{13}$$

The results are compared with our approach proposed in (12). In the two simulation cases, everything is the same except for the optimization objective. In each iteration, the initial values are extracted from the second step of the solutions of the previous horizon. The comparison between the two cases are given in Fig. 6. From Fig. 6(a), observations suggest that Case 2 tends to give a lower temperature compared to Case 1. This is consistent with the bar graph in Fig. 6(b) as Case 2 sees larger duty cycles. Hence, it costs more over the 100 time steps. With the base price being 5.65 ¢/kWh and the price curve being Fig. 5(c), the total electricity costs for Case 1 and Case 2 are 0.000247 ¢ and 0.000281 ¢, respectively. The difference is very small because we are using a scaled-down system and the total operating duration is 500 seconds. Note that the percentage reduction in total cost from Case 2 to Case 1 is about 12.1%. If we consider an actual household AC running for 24 hours, the reduction in cost is quite significant when optimizing cost and comfort is compared to optimizing comfort only. It can also be noticed from Fig. 6(b) that the number of ON/OFF fan switchings is larger in Case 1. This is reasonable as the cooler operates around the temperature upper bound. The fan is required to switch between ON and OFF more frequently according to the designed controller. This would help reducing the maximum temperature seen on the corresponding distribution transformer [13]. Hence, our proposed method is preferable compared to comfort optimization. We expect to test our method in a more realistic environment and compare the results with other existing methods in future work.

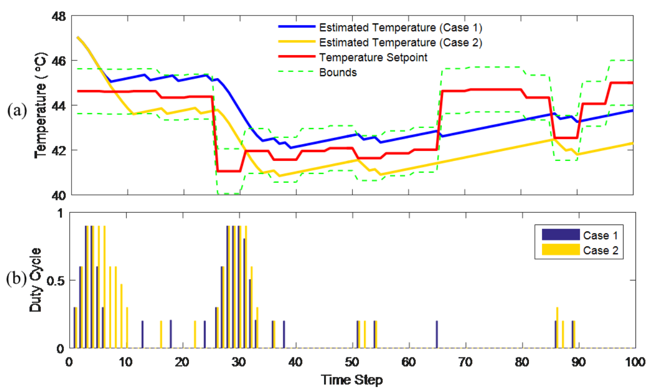

Fig. 6. Case comparison.

VII. Conclusions

In this paper, we propose a real-time human activity-based energy management system (HAEMS) with a significantly reduced number of sensors. A human activity detection algorithm and a model predictive control scheme are developed and implemented. The system is deployed in a scaled-down laboratory setup and the performance is evaluated in a hardware environment.

The experimental results obtained in Section VI meet the expectation of human activity detection and achieve the objectives set by the optimization problem. Human activity intensity is precisely reflected on the temperature set-point input to the optimization problem. Both temperature control and energy consumption cost minimization are realized in real time. Moreover, compared to the method that solely optimizes comfort, the proposed method not only achieves thermal comfort, but also minimizes the total electricity cost.

The successful implementation of the HAEMS on this scaled-down setup verifies the feasibility and effectiveness of the proposed technology. In addition, our system has a low deployment cost as it only requires the installation of a single Kinect, which is already available in many households.

The team is planning two improvements on the proposed technology. First, the human activity detection will be extended to more than one person. We expect to implement the HAEMS in larger public areas such as gyms and office buildings with more people involved, to achieve temperature control and energy management intelligently. In this case, more sensors may be required. Second, the concept could be applied on other devices that measure customers' health condition or temperature directly, such as Fitbit and Apple Watch. If those devices are used, the temperature could be adjusted with more precision to meet individual customer's needs according to the data collected.

References

[1] U.S. Energy Information Administration,"Electric Power Monthly with Data for March 2017," May 2017. [Online]. Available: https://www.eia.gov/electricity/monthly/current_year/may2017.pdf.

[2] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse, "The smart thermostat: using occupancy sensors to save energy in homes," *in Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, Zurich, Switzerland, 2010.

[3] G. Gao and K. Whitehouse, "The self-programming thermostat: optimizing setback schedules based on home occupancy patterns," *in Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, Berkeley, California, 2009.

[4] M. C. Mozer, "The neural network house: An environment hat adapts to its inhabitants," *in Proceedings of AAAI Spring Symposium Intelligent Environments*, 1998.

[5] Y. Kim, Z. Charbiwala, A. Singhania, T. Schmid, and M. B. Srivastava, "Spotlight: Personal natural resource consumption profiler," *in Proceedings of the 5th Workshop on Hot Topics in Embedded Networked Sensors*, 2008.

[6] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, vol. 10, Part B, pp. 138-154, 2014.

[7] Thamer S. Alquthami, "A Smart House Energy Management System", Ph.D. dissertation, Georgia Tech, Atlanta, Georgia, 2015.

[8] T. Alquthami and A. P. S. Meliopoulos, "Smart House Management and Control without Customer Inconvenience," *IEEE Transactions on Smart Grid*, to be published.

[9] S. Grijalva, M. Costley, and N. Ainsworth, "Prosumer-based control architecture for the future electricity grid," *in Proceedings of IEEE International Conference on Control Applications (CCA)*, Denver, CO, 2011.

[10] Maxim Integrated Products, Inc, "Programmable Resolution 1-Wire Digital Thermometer," DS18B20 datasheet, Mar. 2007.

[11] D. Li, D. Zhao, Y. Zhu, and Z. Xia, "Thermal comfort control based on MEC algorithm for HVAC systems," *in Proceedings of International Joint Conference on Neural Networks (IJCNN)*, Killarney, 2015.

[12] S. Li, S. Ren, and X. Wang, "HVAC Room Temperature Prediction Control Based on Neural Network Model," *in Proceedings of the Fifth International Conference on Measuring Technology and Mechatronics Automation*, Hong Kong, 2013, pp. 606-609.

[13] C. Zhong and J. L. Mathieu, "Relation between overheating of distribution transformers and switching frequency of electric loads used for demand response," *in Proceedings of North American Power Symposium (NAPS)*, Charlotte, NC, 2015.