ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/303563207

Intelligent and adaptive temperature control for large-scale buildings and homes

Conference Paper · April 2016

DOI: 10.1109/ICNSC.2016.7478971

CITATIONS

0

READS

14

2 authors, including:

Yuan Wang

Arizona State University

3 PUBLICATIONS **4** CITATIONS

SEE PROFILE

All content following this page was uploaded by Yuan Wang on 28 August 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

Proceedings of 2016 IEEE 13th International Conference on Networking, Sensing, and Control
Mexico City, Mexico, April 28-30, 2016

Intelligent and Adaptive Temperature Control for Large-Scale Buildings and Homes

Yuan Wang
Arizona State University
Tempe, AZ, USA
Email: Yuan.Wang.4@asu.edu

Partha Dasgupta
Arizona State University
Tempe, AZ, USA
Email: partha@asu.edu

***Abstract*—Temperature control in smart buildings and homes can be automated by having computer controlled air-conditioning systems along with temperature sensors that are distributed in the controlled area. However, programming actuators in large-scale buildings and homes can be time consuming and expensive. We present an approach that algorithmically sets up the control system that can generate optimal actuator settings for large-scale environments.**

This paper clearly describes how the temperature control problem is modeled using convex quadratic programming. The impact of every air conditioner(AC) on each sensor at a particular time is learnt using linear regression model. The resulting system controls air-conditioning equipments to ensure the maintenance of user comforts and low cost of energy consumptions. Our method works as generic control algorithms and are not preprogrammed for a particular place. The system can be deployed in large scale environments. It can accept multiple target setpoints at a time, which improves the flexibility and efficiency for temperature control. The feasibility, adaptivity and scalability features of the system have been validated through various actual and simulated experiments.

I. Introduction

About 70% of the electricity load is consumed by commercial and residential buildings in the US. Studies show by the year 2025, building electricity energy costs will be over 430 billion dollars. With the rapid increase of energy costs in the building sector, carbon dioxide(CO_2) emissions are growing faster than before. The sector contributes around 39% CO_2 emissions in the US per year, more than any other sector such as transportation, which results in threat of climate change.

Heating, ventilation and air conditioning(HVAC) system is the largest energy consuming sector in buildings and homes. In the year 2009, about 48% energy costs are contributed by space heating and cooling. Most of traditional HVAC systems in buildings and homes are controlled by thermostats that are manually configured either by centralized control technicians or individual users, which requires a lot of human efforts. For maintaining user comfort levels in work places, these systems usually keep AC settings all the day, even during off hours when the places are not occupied, which causes a lot of unnecessary energy consumptions. Some of thermostats in office areas do not provide friendly interfaces and flexible functionalities for users to control. In a centralized air-conditioning control area, users can not customize the temperature settings for a particular place, which causes a lot of inconvenience.

Saving energy and satisfying user comforts are two main aims for HVAC control systems design. For saving energy, amount of total energy costs should be minimized during the control. For maintaining user comforts, inside room temperature(sensed by temperature motes) should be uniform and stay in a satisfied range. There always exist tradeoffs between these two goals. One of the popular products in current market for smart thermostat technology is the Nest Learning Thermostat [1]. It learns user preferences such as temperature levels during a day for about a week and then it can automatically generate AC control plans based on user input data. The drawback of this approach is it relies too much on user selected data. If users randomly provide some incorrect data to the thermostat or they disable it for a particular time, the effectiveness of the thermostat may drop down. [2] proposed an approach called *smart thermostat* that can automatically turn on/off air conditioning systems by sensing occupancy and sleep patterns. It can also generate an energy efficient plan for the preheating stage by looking at system configurations and analyzing historical occupancy patterns. One limitation of this work is it relies on a lot of information from the equipment itself and this paper only evaluates a single type. The scalability and adaptivity of the system need to be improved.

Although there are a lot of great improvements for smart building technologies nowadays, current control plans for HVAC systems still have some drawbacks and limitations. First, in most of the commercial buildings, one thermostat is used for controlling multiple vents at same time. People in different working areas can not customize their preferences when they have conflicts on temperature settings. Second, current HVAC control approaches are not good for large scale environments where there are many standalone ACs in an area. Each AC is independently controlled by a single switch. A subset of ACs need to be turned on according to users' preference settings. The control plan should optimize user comfort levels without sacrificing too much energy. Last, in the current HVAC control system, temperature sensors are usually built in thermostats that are put on the wall. They should be placed at sitting areas in order to better capture surrounding temperature for users.

In this paper, we design an air-conditioning control system for tackling the above challenges. Our system can be used for controlling large commercial and residential buildings where there exist multiple ACs, control switches and temperature sensors. The goal of the system is to optimize the user comfort levels and minimize energy consumptions. There are two main stages in this system: predictive stage and adjustive stage. In the predictive stage, mathematical models are used to formalize the problem and optimization algorithms are used to get a predicted solution. In the adjustive stage, feedback system are designed to reduce potential errors. Our system is good for providing fine-grained settings during precooling/preheating stage. Our approach is adaptive, which means it can be easily applied in any type of building environments.

II. Related Work

Thermostats technologies have been widely used in HVAC systems for automatically controlling HVAC equipments in buildings and homes. The basic logic behind is HVAC equipments are turned on when its controlled area is occupied and turned off when occupants leave the area. The thermostats are preprogrammed and temperature setpoints are predefined according to the local environments such as static occupancy patterns. It relies on too much static information therefore it is hard to adapt to environmental changes. An alternative method called reactive thermostat is proposed for tackling the problem. It uses various sensors such as motion sensors or door sensors to detect user activities in real time so that the control system can adjust system settings when pattern changes. However, some studies found that this method didn't improve the efficiency as people expected since there usually exists long delay for the system reaction due to the hardware limitations which even save less energy than the previous programmable thermostat approach.

[2] proposed an approach called *smart thermostat* that can automatically turn on/off air conditioning systems by sensing occupancy and sleep patterns in real time. It can also generate an energy efficient plan for the preheating stage by looking at system configurations and analyzing historical occupancy patterns. This approach dynamically controls HVAC systems based on occupancy status in a place, which can effectively adapt to environmental changes. One limitation of this work is it relies on a lot of information from the equipment itself and this paper only evaluates a single type. The scalability and adaptivity part of the system need to be improved.

Nest Learning Thermostat [1] is a popular user-centric tool for HVAC system control, which is very easy to use compare to the traditional programmable thermostat. It automatically learns a user's preferences and behaviors based on some pre-collected data from the user. The drawback of this approach is it relies too much information on the user-input data. If users randomly provide some incorrect data to the thermostat or they disable it for a particular time, the effectiveness of the thermostat may drop down. Another drawback of this approach is it is usually set up in a single AC environment such as residential homes where only one target temperature value at a time is accepted. When two people have conflicts on target temperature values in a place or the thermostat is set up in a large scale environment where multiple ACs exist, the thermostat can not give effective solutions.

A fuzzy inference system for adaptively doing heating control is proposed in [3]. The main idea is it takes power profile of the previous day, adjusts the profile based on current conditions and then applies the latest profile to the current day. It uses Artificial Neural Network model to predict the future comfort levels and a fuzzy rule is designed for the setting adjustment step. The drawback of the system is it mainly focuses on maintaining user comforts during the control. The energy saving part is only considered when a place is not occupied. The fuzzy rule should be improved so that some fine-grained settings can be provided instead of using the words "large", "medium" and "small".

Our approach is proposed for tackling the above problems. It can be deployed in a large scale environment without any static or customized configurations. If needed, users can set their own preferences on temperature setpoints in a workplace. The system can efficiently and effectively control HVAC equipments even when there exist multiple target values on sensors. It provides fine-grained pre-cooling plans based on occupancy schedules. When environment changes, the system can automatically update calibration data and adjust the settings accordingly.

WSN technologies have been applied into various areas such as [4] [5][6]. It consists of portable wireless sensor motes such as Crossbow's TelosB or MICAz to monitor the values of physical conditions, such as temperature, light, humidity, and so on. WSN data collections use several specific protocols such as Collection Tree Protocol [7].

III. Formal Model

This section describes how we use mathematical model to formulate the problem. For simplicity, we build the model for cooling strategy of air-conditioning control. Heating strategy can be modeled in a similar way.

In this model we have a set of on/off switches that control arbitrary n ACs (one per switch) and a set of m temperature sensors. The sensors are connected to the control system via a wireless sensor network and the switches are activated via actuators connected to the system. Physical locations and correlations between them are initial unknown to the control system.

The ultimate task is to compute the positions of switches over time, i.e. durations for each switch to be kept on. We assume an AC will keep running until its control switch is turned off. Let $t = (t_1, ..., t_n)^\intercal$ denote the assignment for AC switches where $t_i \geq 0$ and it denotes switch i is kept on for t_i time units.

The goal is to optimize the energy $E(t)$ and the comfort $C(t)$. $E(t)$ is the total energy consumptions in the control period. We assume the power of each AC is constant during the control period. Let $w = (w_1, ..., w_n)^\intercal$ denote the electric power for each AC. Then $E(t) = w^\intercal \times t$. For $C(t)$, it is satisfied when all sensor values approach their target values and stay in a satisfied range. Let $e = \langle e_1, ..., e_m \rangle$ denote all sensor readings. Each sensor has a target value to be reached, for example, sensor j's target value is tar_i. Let $tar = \langle tar_1, ..., tar_m \rangle$ denote all target values. They can be set either by users or default standard values. Hence the problem can be stated as:

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & \langle w^\intercal \times t, \|e - tar\|_2^2 \rangle \\ \text{subject to} \quad & min \leq e_j \leq max, \ j = 1, ..., m \\ & 0 \leq t_i \leq t_{max}, \ i = 1, ..., n \end{aligned} \tag{1}$$

By applying the ϵ-constraint method [8] designed for solving multi-objective optimization problems, the new objective function can be defined as:

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & \|e - tar\|_2^2 \\ \text{subject to} \quad & min \leq e_j \leq max, \ j = 1, ..., m \\ & w^\intercal \times t \leq \epsilon \\ & 0 \leq t_i \leq t_{max}, \ i = 1, ..., n \end{aligned} \tag{2}$$

We assume that the impact of ACs on sensors(number of sensor values decrease) over a short time period is additive, i.e., the impact of two ACs over a short time period on one sensor(total number of sensor values decrease during the period) is the sum of the individual impact for each AC. Since the time period is considered to be small for solving the above equation, the relationship between time and temperature can be learnt using linear regression model, which will be discussed in the next section.

IV. Control Approaches

There are two main stages for solving the air-conditioning problem. The first stage is called predictive stage that happens before the room is occupied, which is used for predicting an approximate setting for air-conditioning control. The second stage is called adjustive stage that happens after room is occupied. It is used for eliminating the potential offsets from the predictive stage and providing fine-grained air-conditioning control for maintaining user comfort levels. In order to achieve max electricity savings, we decide to turn off all ACs when the controlled area is not occupied, and pre-cool the area in advance before users enter the space. By default we assume we have a list of occupancy schedules. The schedule can be updated by users in real time, for example, a user can send a notification at any time telling the system when he expects to get back.

A. Predicting AC Impacts on Sensors Over time

In the cooling stage, an AC continuously provides cooling air to reduce the inside temperature levels until it is turned off. Since the relationship between time and temperature in each cycle is investigated over a short time period, up to an upperbound t_{max}, a linear regression model can be identified with a high coefficient of determination [9]. It is defined in the form of:

$$T = k * t + b$$

where t stands for interval time and T stands for temperature level. In order to learn k and b, some experimental data points should be pre-collected. These coefficients should be updated when environment changes such as building patterns or weather conditions. There are multiple ACs and multiple sensors in the system. For each sensor, every AC has an impact factor on it, i.e. a pair of (k, b). Since sensor readings are additive, to a particular sensor j, its reading e_j after an operation of n ACs on a vector t time is:

$$e_j = s_j - m_j^\intercal * t \tag{3}$$

where s_j stands for the initial sensor j's value and

$$m_j = \begin{pmatrix} k_{1j} \\ k_{2j} \\ \vdots \\ k_{nj} \end{pmatrix} \quad t = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{pmatrix}$$

s_j can be collected from the calibration step and m_j can be calculated based on linear regression model.

B. Calibrating and Calculating Impact Factor

In the last section, we discussed how we use a linear regression model to learn the relationship between time and temperature. Note that this model is effective when time vector is small. Assume all ACs are off at the beginning. Calibration steps involve the following:

- Turn on one AC at a time for a period of time t, record the temperature changes on every sensor. Selection of t should be reasonable, so that a range from initial temperature to the target temperature can be covered for each sensor, if applicable.
- Analyze the collected data using linear regression model. Perform linear regression analysis for different period of time, if necessary. Thus, a impact factor k_{ij} might have different values for different periods of time.
- Repeat step 1 n times until all ACs are counted. For each sensor j, a vector m_j is created. If more than one k_{ij} exist, then for each period of time, a m_j is recorded.

In each AC operation cycle, either the predictive stage or the adjustive stage, only one m_j can be used for sensor j. The selection of m_j is decided by the initial sensor value on j. Each AC is assigned an upperbound value for its operational time during every cycle, formulated as $0 \leq t_i \leq t_{max}$. Value of t_{max} is associated with value of k_{ij}. For each AC i, from time 0 to time t_{max}, its impact factor on every sensor j k_{ij} should remain the same.

Ideally, impact factors and room initial temperatures should be updated frequently to ensure the accuracy of the computation. However, in some cases, these values remain the same in a period of time. Thus in order to increase the efficiency, these values don't have to be re-calibrated or updated if the change doesn't exceed a threshold. The threshold value can be set based on local environments.

C. Solving the Equations to Compute Approximate AC Settings

The result of equation 2 is a time vector t that indicates the working time interval for every AC in the room. The upperbound value t_{max} for each t_i in vector t can be different. The largest value among all values of t_{max} is assigned to q. If users return at time p, then the system will start calculating t from time $p-q$ in order to cool the room at time p. After time p, the system will go to the adjustive stage until the room is unoccupied again.

Applying equation 3 to equation 2, we get:

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & \sum_{j=1}^{m}(m_j^{\intercal} * t - x_j)^2 \\ \text{subject to} \quad & s_j - max \le m_j^{\intercal} * t \le s_j - min,\ j = 1, ..., m \\ & w^{\intercal} \times t \le \epsilon \\ & 0 \le t_i \le t_{max},\ i = 1, ..., n \end{aligned} \tag{4}$$

where $x_j = s_j - tar_j$. The objective function of equation 4 is a quadratic function that can be re-written in the following form:

$$t^{\intercal} * (\sum_{j=1}^{m} m_j * m_j^{\intercal}) * t - 2 * (\sum_{j=1}^{m} x_j * m_j^{\intercal}) * t + \sum_{j=1}^{m} x_j^2 \tag{5}$$

The constraints of equation 4 can be re-written as $A * t \le b$ where

$$A_{m,n} = \begin{pmatrix} k_{11} & k_{21} & \cdots & k_{n1} \\ k_{12} & k_{22} & \cdots & k_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ k_{1m} & k_{2m} & \cdots & k_{nm} \\ -k_{11} & -k_{21} & \cdots & -k_{n1} \\ -k_{12} & -k_{22} & \cdots & -k_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ -k_{1m} & -k_{2m} & \cdots & -k_{nm} \\ w_1 & w_2 & \cdots & w_n \end{pmatrix}$$

and

$$b = \begin{pmatrix} s_1 - min \\ s_2 - min \\ \vdots \\ s_m - min \\ max - s_1 \\ max - s_2 \\ \vdots \\ max - s_m \\ \epsilon \end{pmatrix}$$

The above transformation satisfies the format of quadratic programming model [10] where $H = 2 * \sum_{j=1}^{m} m_j * m_j^{\intercal}$, $f^{\intercal} = -2 * \sum_{j=1}^{m} x_j * m_j^{\intercal}$, lb is a vector of zeros, ub is a vector of all t_{max} values. Number of elements in lb and ub is both n. Thus our problem belongs to quadratic programming problem, which can be solved by some existing algorithms such as interior point method. In addition, the objective function appearing in equation 4 is a standard form of least-squares [11] that is convex, therefore our problem is a convex quadratic programming problem. The result set will be a set of global minimum.

The weight factor ϵ (which appeared in equation 4) is used to balance the tradeoff between saving energy and maintaining user comforts. The assignment of ϵ will significantly affect the final result, therefore, it should stay in a reasonable range. In order to rationally set ϵ, we first study the approximate minimum value of ϵ, i.e., $\epsilon \ge \epsilon_{min}$, by solving the following equation:

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & w^{\intercal} * t \\ \text{subject to} \quad & A' * t \le b' \\ & 0 \le t_i \le t_{max},\ i = 1, ..., n \end{aligned} \tag{6}$$

where A' and b' can be obtained from A and b by removing their last rows($w^{\intercal}$ and ϵ) respectively. This equation is simpler than the previous one since we remove an optimized variable(approaching all sensor values to the targets) from the original equation. In other words, compared to the original problem, in this equation, we focus on minimizing the energy usage only, with all sensor readings requiring to stay within an accepted range. This equation satisfies the form of linear programming model, which can be solved by existing algorithms such as interior point or simplex algorithm. The value of ϵ can be set as ϵ_{min} + a reasonable threshold that is assigned according to the real environments.

D. Adjusting AC Settings for Maintaining User Comforts and Adapting Environmental Changes

There might exist some errors in the predictive stage. In order to eliminate them, we add an adjustive stage in the system that can adjust the settings accordingly so that the user comfort levels can be maintained. Compared to the predictive stage, the adjustive stage has some similarities and some differences. The similarity in both stages rely on the proposed computational model to generate AC settings. Differences are:

- Unlike the predictive stage, the adjustive stage can be done repeatedly until the controlled area is unoccupied. In other words, predictive stage is used for precooling the area before the room is occupied. Adjustive stage needs to be activated whenever users are in the area and comfort levels are not satisfied.
- During the first adjustive cycle, impact factor m_j is selected based on initial value on sensor j, as predictive

cycle does. Afterwards, in every adjustive cycle, the vector m_j should be checked using the partial real data points collected from sensors. If there is any change, the vector should be updated accordingly. This update reflects self-calibrating and self-learning features of the system, which increases the accuracy of the computation.

When environment changes such as location changes or season changes, impact factor m_j need to be re-updated. Data for learning the factor need to be re-collected from the environments, but the general approaches remain the same. For the AC control at a particular day when calibration data for previous (similar) days have been recorded, impact vectors should be first chosen from similar categories and updated accordingly in the real time, similar to the update step in the adjustive stage. For example, we wish to predict the AC settings for today at 5pm. We first use yesterday's m_j in the category of 5pm. After getting the results from the predictive stage, we turn on the ACs based on the predictive settings. If in the first 2 minutes, we detect the impact factor varies a lot compared with yesterday's one, we will update the m_j accordingly based on today's data and recompute the settings. Thus a new dataset of m_j for today's 5pm is obtained. It will be inserted into database for future computations. If allowed, the interval time for recalibration can be set longer for better accuracy.

V. Experimental Work and Simulation Results

A. Verification of Linear Regression Model

We performed two calibration experiments in the test room with one AC(540 watt) and one sensor in a same day at 3pm and 4pm respectively. For each experiment, the AC is operated continuously for about 40 minutes and temperature changes are recorded. The calibration follows the steps described in section IV-B. Figure 1 shows the AC impacts on the sensor at different time points, and their corresponding fitted lines generated by MATLAB. From the figures, following things are observed:

- Both two figure can be divided into 2 parts. The first part covers the temperature points from initial point to 24°C. The second part covers the temperature points from 23.5°C to 19°C. Both parts are fitted by linear regression models with different impact factors. The regressed line for the first part has a larger scope than the second one's.
- Figure 1(b) has similar slopes compared with Figure 1(a), for both lines.
- Two figures have similar initial sensor values. Temperature's range is approximately from 19°C to 33°C.

From the above findings, we verify the accuracy of linear regression model for learning AC's impacts on sensors during selected time intervals. The experiment also shows the calibration data can be reused for a period of time to increase efficiency. In this experiment, during the 40-minute interval, the AC has two impact factors on sensors. Every AC operation cycle should select appropriate impact value for computation.

B. Simulation Experiments and Results

In section IV, the computational model is proved to be convex, which means the result set is guaranteed be global minimum. In this section, we use MATLAB to simulate a multi-AC environment and compute the optimal solution. The optimization toolbox is called *Quadratic Programming*. Simulated set-up is given based on some real experimental data, described as follows:

Assume there are 3 ACs and 2 sensors in a room. Sensors are placed at user-sitting areas to capture inside temperature levels. Initial values(s_1 and s_2) are both 30. Target values tar_1 and tar_2 are both 26. Therefore, x_1 and x_2 are both 4. min is 25 and max is 28. $w = (0.45, 0.46, 0.47)^\intercal, m_1 = (0.05, 0.06, 0.07)^\intercal, m_2 = (0.05, 0.06, 0.07)^\intercal$. Based on these settings, we can get $H, f^\intercal, A_{m,n}, b$ defined in section IV-C. These factors are needed in the Quadratic Programming toolbox.

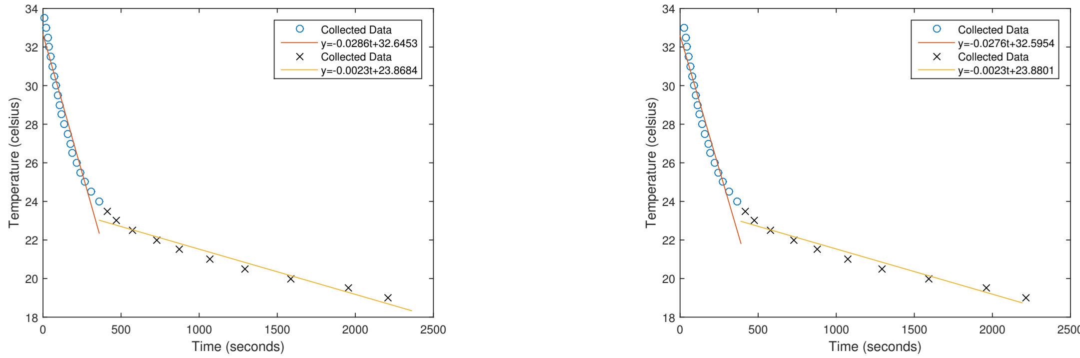
In equation 4, weight factor ϵ affects the final result. It represents the tradeoff between energy savings and user satisfactions. In this simulation experiment, we set different values to ϵ and hope to learn the effect of ϵ on the final result set. In order to better assign the values, the approximate lowest ϵ is learnt by solving the equation 6 using interior-point method. For the particular example stated above, lowest ϵ is 14.

From the graph, we can see that when ϵ is becoming larger, minimum value of the objective function is becoming smaller(sensor values are more converged to the targets) while the energy consumption is becoming larger. This matches theoretical analysis as well: when ϵ is larger(more flexibilities on energy consumption), user comforts can be better optimized(more candidates are counted), therefore sensor readings are more converged to the target values.

We also scale our simulations and do simulated experiments on a larger number of ACs and sensors. We set $n = 100$ and $m = 100$. Min, target and max values of sensors do not change. Assume all ACs have same operational powers(0.46kw). Impact vectors are randomly given on a 0.005-base increase from 0.05, for example, if $k_{11} = 0.05$, then $k_{21} = 0.055$ and so on. To a particular AC i, $k_{ip} = k_{iq}$. We run the above set-up in MATLAB. It quickly gives us the result of $MinValue = 0$(global minimum) and $Energy = 10.72$. It verifies the scalability of our approach.

VI. Conclusion and Future Work

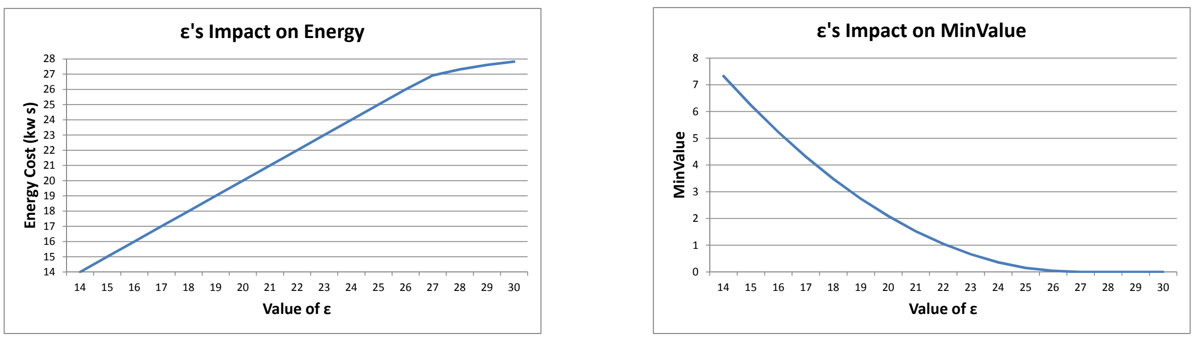
To enable automated temperature control in a multi-AC environment, under varying conditions of occupancy, weather, seasons and other influences it is essential to have a robust air-conditioning control system that is effective and adaptive. Such system must be deployable in a simple, cost effective way without the need for customizations and reprogramming as conditions change. It also needs to create a comfortable environments with energy savings as a goal. This paper presents such a complete core system that can pre-cool the room before users come back and maintain the comfort levels with a relative small energy cost. It also clearly shows how



(a) 3pm

(b) 4pm

Fig. 1. AC Impacts on Sensors over Time



(a) Energy

(b) MinValue

Fig. 2. Impact of ϵ on Energy and MinValue

the control problem can be written in convex quadratic programming form, which is guaranteed to get global minimum values. Impact factors of AC on sensors are learnt using linear regression model. Results have been validated in both experimental and simulated scenarios, which shows the feasibility and effectiveness of our system.

Our current air-conditioning control system only considers temperature value as the main target for maintaining user comfort levels. Other factors such as humidity, air flow, etc might affect the user comforts as well. These factors should be added into the system for computation in future. Appropriate models need to be selected for solving the new problem. In this paper, we assume we have a list of occupancy schedules, which is used for the pre-cooling stage. This information needs to be further studied in future to increase the accuracy of our system control. With the use of the system, more data will be collected in the database. These data can be used to better predict the impact factors. The results will be applied into the current system control plan. When dataset is made larger, the system will become more robust and precise.

References

[1] Nest thermostat. [Online]. Available: https://nest.com/thermostat/life-with-nest-thermostat/

[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*. ACM, 2010, pp. 211–224.

[3] A. Guillemin and N. Morel, "An innovative lighting controller integrated in a self-adaptive building control system," *Energy and Buildings*, vol. 33, no. 5, pp. 477–487, 2001.

[4] Y. Wang and P. Dasgupta, "Designing an adaptive lighting control system for smart buildings and homes," in *Networking, Sensing and Control (ICNSC), 2015 IEEE 12th International Conference on*. IEEE, 2015, pp. 450–455.

[5] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 88–97.

[6] Y. Wang and P. Dasgupta, "Designing adaptive lighting control algorithms for smart buildings and homes," in *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*. IEEE, 2014, pp. 279–284.

[7] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 1–14.

[8] G. Mavrotas, "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems," *Applied Mathematics and Computation*, vol. 213, no. 2, pp. 455–465, 2009.

[9] Wikipedia. Coefficient of determination. [Online]. Available: http://en.wikipedia.org/wiki/Coefficient_of_determination

[10] MathWorks. Quadratic programming. [Online]. Available: http://www.mathworks.com/help/optim/ug/quadprog.html

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.