1-1-2015

Comparison Of Lazy Controller And Constant Bandwidth Server For Temperature Control

Zhen Sun
Wayne State University,

COMPARISON OF LAZY CONTROLLER AND CONSTANT BANDWIDTH SERVER FOR TEMPERATURE CONTROL

by

ZHEN SUN

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2015

MAJOR: COMPUTER SCIENCE

Approved by:

Advisor Date

DEDICATION

To my parents, Yi Sun and Guilan Liu.

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to my advisor, Dr.Fisher who supported me throughout this thesis. I am thankful for his aspiring guidance, invaluably constructive criticism and friendly advice during the thesis work. I am sincerely grateful to him for sharing his truthful and illuminating views on a number of issues related to this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In building systems, like heating, cooling systems are all related to temperature control. The temperature should be kept within a desired range to maintain system work. Inefficient temperature control may causes temperature abnormal and make people in the building feel uncomfortable.

There are several methods to balance the temperature. For example, MPC (Model predictive control. The main idea of MPC is to use the model of the buildings to predict the future evolutions of the system,but the scalability for large systems might be limited because of its high computational requirement for optimization [7].

Recently, green scheduling control approach was proposed which included lazy controller. The lazy controller can achieve temperature control well to keep all the temperatures within desired ranges. Consider a heating system of multiple zones and zones are heated by heaters. Assume this heating system is applied in several rooms. Each zone represents a room and it is heated by a heater. The room temperature must be between the lower and upper threshold; that is,the room temperature is safe. Green scheduling control introduced a simple scheduling controller called lazy controller. In lazy scheduling,the heater only has two operation modes : ON and OFF [6]. Obviously mode ON means the heater is turned on. However, in the mode OFF, that means when it consumes no energy and provides no heat input a heater is turned off. The heater will be switched between ON and OFF modes once the temperature reaches the upper or lower thresholds. If the temperature within the thresholds, heater will keep it's current mode. Besides ensuring all the tasks are safe, a schedule policy must satisfy a resource constraint that at most k heaters can be in mode ON simultaneously at any time [8].

1.1 Thesis and Contribution

However,it seems that we do not need to switch the heater ON or OFF only when the temperature hits the upper or lower threshold because the constraint is kept task within the upper and lower thresholds. Therefore, we would like to using CBS (constant bandwidth server) in the heating system to see whether this approach is more efficient than Green scheduling control [9] [1]. CBS is one of the most used algorithms for implementing resource reservation on a real-time system. Each task can be assigned a server (one task per server).The server is characterized by period and computation time, which is the budget. Besides, each server has deadline which initial value is 0 and will be updated when the budget exhausted. The ratio budget/period yields bandwidth, which is the fraction of CPU to be reserved by the scheduler for each subsequent period. The CBS is scheduled in such a way that it will never demand more than its reserved bandwidth,which means the heater does not need to turn off only when it hits to the upper threshold. Every time the heater turn on for budget unit,once the budget exhausted then it will be replenished and postpone the server deadline.

In my thesis, I will analyze and compare green scheduling controller and CBS scheduling system in some aspects. By implementing small-scale experiment, I can analyze the efficiency of green scheduling controller and CBS controller under different circumstances.

1.2 Organizations

The rest of the paper is structured in the following manner: Chapter 2 will introduce two models include the Worst Case Execution Time (WCET). We also state and prove two theorems related to the budget's condition. Chapter 3 will formally introduce the constant bandwidth server, giving some key notations and task schedules in our case. Besides, it will give the schedubillity analysis regaring the CBS rules. Chapter 4 will evaluate the green scheduling method and CBS method concurrently and make comparisons between these two approaches. Chapter 5 will state the conclusions and some future works.

1.3 Related Work

This section surveys previous work in structured text processing. Energy efficiency plays an important role in real-time system, usually there are two widely-used technique are Dynamic Voltage Scaling and Dynamic Power Management [4]. The first one is used to reduce CPU power consumption by reducing CPU voltage and frequency, while second one is used to reduce power consumption by transitioning a device from the active state to sleep state. Both Green scheduling controller and CBS controller are related to DPM by transitioning from one mode to another mode.

Recently the popular approaches to energy efficient control for commercial buildings are Green scheduling and MPC(model predictive control). I will introduce Green scheduling controller later.

In general terms, the MPC is a flexible and well-developed advanced control technique. The idea of MPC is to design a predictive model and use this model to predict the future evolution of the system. For example, weather prediction, occupancy prediction from MPC could help to control the building climate [10]. The main purpose of MPC is to minimize building's energy consumption, but the scalability of this method for large systems might be limited duty to its high computational requirements. Green scheduling control may be better than MPC in this aspect.

Generally speaking, there are several ways to achieve reduction of energy consumption except green scheduling control and model predictive control. My thesis will put emphasis on green scheduling control and CBS system. Analyzing the existing green scheduling control system, apply and improve CBS algorithm to the same model with green scheduling to see whether the CBS algorithm can be efficiently used for temperature control or not.

CHAPTER 2

TASK MODELS

Consider a heating system with multiple zones, and each zone represents a room. Let us denote that there are i zones, where $1 \leq i \leq n$, and heat input rate is $Q_i \geq 0(KW)$. The room temperature in Zone i which denoted by $x(°C)$ is comfortable when it stays between the lower threshold l_i and an upper threshold h_i. T_a is the ambient air temperature(°C). The conservation law of energy gives us the following equation for zone i:

$$C_i \frac{\mathrm{d}x_i}{\mathrm{d}t} = K_i(T_a - x_i) + Q_i \tag{2.1}$$

where $C_i(KJ/K)$ is the thermal capacity of the zone and $K_i(KW/K)$ is the thermal conductance [8].

2.1 Worst Case Execution Time Model

In this section, we define a Worst Case Execution Time model. The worst case execution time of a task is the maximum length of time the task could take to execute. In this model, the reason why it is called worst case execution is that heater's execution time of every period is same. The model has a state which means room temperature in this paper. l_i and h_i represent the lower and upper bounds. Assume applying the budget B_i to the model(we will explain this budget B_i later), turn on the heater for continuous B_i time, then turn off until to T_s. A model M is a tuple$(x(t_i), l_i, h_i, B_i, m, T_i)$, in which:

- $x(t_i)$ is room temperature;
- l_i and h_i, where $l_i < h_i$, are the lower and upper thresholds of the x_i;
- B_i is the budget;
- m is a set of operation mode;

- T_i is the period.

According to Truong's Green scheduling system [8], the task model has two operation modes which are ON and OFF, that is m=ON or OFF. ON means turn on the heater, OFF stands for turn off the heater and meantime heat input rate $Q_i = 0$. In the heating system, when the mode is ON, then the state (room temperature) will increase. However, when the mode is OFF, the state will decrease. The system growth temperature is given by

$$x(t) = \frac{b^+}{a^+} + (x(t_1) - \frac{b^+}{a^+})e^{-a^+(t-t_1)} \tag{2.2}$$

and the decay of x's equation is

$$x(t) = \frac{b^-}{a^-} + (x(t_1) - \frac{b^-}{a^-})e^{-a^-(t-t_1)} \tag{2.3}$$

$x(t_1)$ is the initial temperature.where a^+,b^+,a^-,and b^- are parameters. The constraint of threshold is as follows:

$$\frac{b^-}{a^-} < l < h < \frac{b^+}{a^+} \tag{2.4}$$

Constant bandwidth server(CBS) is a scheduling algorithm based on reserving a fraction of the processor bandwidth to serve a periodic jobs. A CBS server is defined by (B_i,T_i), where T_i is the period of the server and B_i is the maximum budget in each period. According to the update rules, the budget and deadline will be updated for every job arrival. In this model, we just assume the budget B_i is represent time demand. Here we introduce a worst case. In every period, the time of mode ON is q which is WCET(Worst Case Execution Time), and time of OFF mode is $T_i - B_i$. [9] [2]

Lemma 1. *Given one room and one server which denoted by period T_i and budget B_i, it includes two heating intervals and two cooling intervals. If moving the later heating*

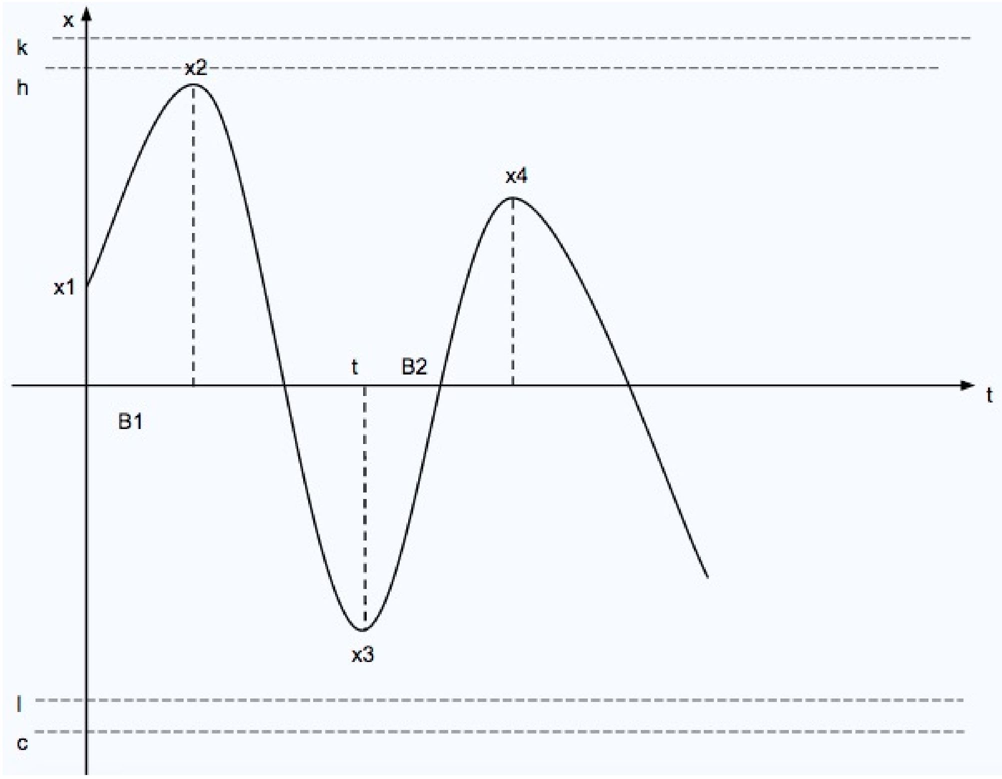

Figure 2.1: Room temperature model

interval forward, then the new peak temperature will higher than the either of previous peak temperature.

Proof. Assume B_1 and B_2 are separate time for the heating intervals and $x(t_1)$ is the initial state, see Figure 2.1. In Figure 2.1, the horizontal ordinate represents time and vertical ordinate means room temperature. $x(t_1)$ is initial temperature,$x(t_2)$ and $x(t_4)$ are highest peak temperature, however $x(t_3)$ is lowest peak temperature. Turn on the heater at very beginning, So according to the equation 2.2 and 2.3, let $k = \frac{b^+}{a^+}$and $c = \frac{b^-}{a^-}$we can see that first peak state $x(t_2) = k{+}(x(t_1) - k)e^{-a^+B_1}$. the second peak state $x(t_4) = k{+}(c{-}k)e^{-a^+B_2}{+}(k{-}c)e^{-a^-t-a^+B_2}{+}(x(t_1){-}k)e^{-a^+(B_1+B_2)-a^-t}$. After i moving the later heating interval forward, the new peak state will be $x = k + (x(t_1) - k)e^{-a^+(B_1+B_2)}$ we can see from the equations since the $x(t_1) - k < 0$, and $e^{-a^+(B_1+B_2)} < 1$, besides, $e^{-a^+(B_1+B_2)} < e^{-a^+B_1}$, thus x higher than $x(t_2)$. As for $x(t_4)$, same way,$x - x(t_4) =$

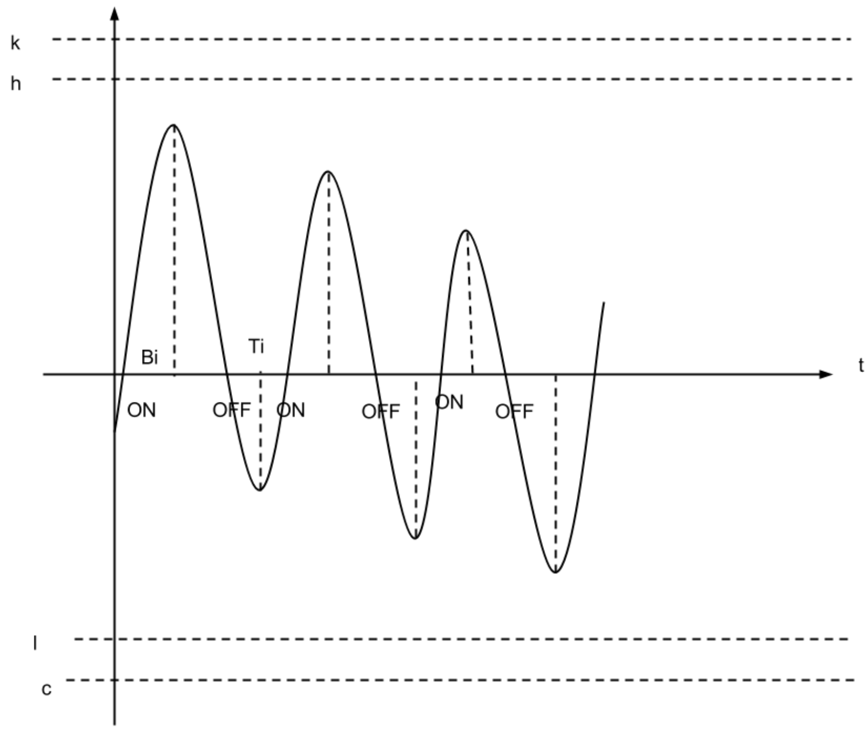

Figure 2.2: WCET model: Room in mode ON at beginning, every period heating time is B, then cooling that is in mode OFF for $T - B$ time unit.

$(x(t_1) - k)(e^{-a^+(B_1+B_2)} - e^{-a^+(B_1+B_2)-a^-t}) + (k-c)(e^{-a^+B_2} - e^{-a^-t-a^+B_2}) > 0$, therefore, $x > x(t_4)$. □

Theorem 1. *Under lemma 1, assume the heater of room in mode ON at beginning, the minimum budget B_i should satisfy the following condition :*

$$\frac{(k-c)e^{-a^-}(T_i - B_i) \times (1 - e^{-a^+B_i})}{1 - e^{-a^-(T_i-B_i)}e^-a^+B_i} \geq l - c \tag{2.5}$$

Proof. Assume $x(t_1)$ is the initial state, so according to 2.2:

$$x(t_2) = k + (x(t_1) - k)e^{(-a^+B)} \tag{2.6}$$

Then $x(t_3)$ equals

$$x(t_3) = c + (x(t_2) - c)e^{-a^-(T-B)} \tag{2.7}$$

we can infer that

$$x(t_3) = c + (k + (x(t_1) - k)e^{-a^+B} - c)e^{-a^-(T-B)} \tag{2.8}$$

$$x(t_4) = k + (x(t_3) - k)e^{(-a^+B)} \tag{2.9}$$

plug in $x(t_3)$ in $x(t_4)$, then $x(t_4) = k+(k-c)e^{-a^-(T-B)}e^{a^+B}+(x(t_1)-k)e^{-a^+B}e^{-a^-(T-B)}+(c-k)e^{-a^+B}$

Overall, we can conclude that: every peak temperature $x(t_{2n-1}) = k+(c-k)e^{-a^+B} \times \frac{1-A^{n-1}}{1-A} + (k-c)e^{-a^+B}e^{-a^-(T_i-B)} \times \frac{1-A^{n-1}}{1-A} + (x(t_1)-k)e^{-(n-1)a^-(T_i-B)}e^{-na^+B}$. where $A = e^{-a^-(T_i-B)}e^{-a^+B}$ when n approaching to infinity, the upper thresholds of state x is h. According to Lemma 1 therefore xt_{2n-1} need to less and equal to h, thus we can get the minimum value of the budget B. □

Therefore the budget q should satisfy the following condition: $\frac{(k-c)e^{-a^-}(T_i-B_i)\times(1-e^{-a^+B_i})}{1-e^{-a^-(T_i-B_i)}e^{-a^+B_i}} \geq l-c$

Theorem 2. *Under Lemma 1, assume the heater of room in mode OFF at beginning, then budget B_i should satisfy the following condition that:*

$$\frac{(c-k)e^{-a^+B}(1-e^{-a^-}(T_i-B_i))}{1-e^{-a^+B_i}e^{-a^-(T_i-B_i)}} \leq h-k \tag{2.10}$$

Let room cooling for $T_i - B$ time at beginning then turn on the heater for B, the scenario will be different, see in figure 2.2. Assume $x(t_1)$ is the initial state, then according to 2.3:

$$x(t_2) = c + (x(t_1) - c)e^{-a^-)}(T_i - B) \tag{2.11}$$

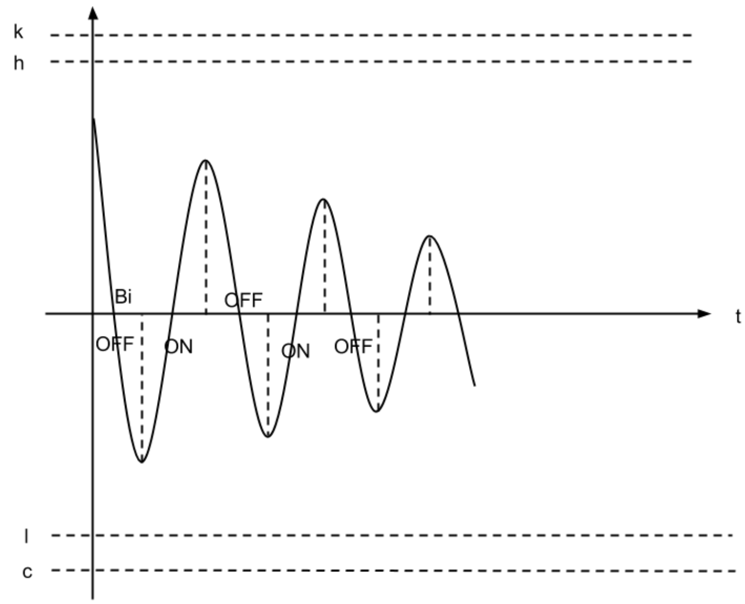

Figure 2.3: WCET Model:Room in mode OFF at beginning, every period cooling time is B_i, then heating time is $T_i - B$ time unit

According to 2.2, $x(t_3)$ equals

$$x(t_3) = k + (x(t_2) - k)e^{-a^+}B \tag{2.12}$$

So we can infer that $x(t_{2n-1}) = c + (k-c)e^{-a^-}(T_i - B) \times \frac{1-A^{n-1}}{1-A} + (c-k)e^{-a^+B}e^{-a^-(T_s-q)} \times \frac{1-A^{n-1}}{1-A} + x(t_1) - k)e^{-(n)a^-(T_i-B)}e^{-(n-1)a^+B}$

where $A = e^{-a^-(T_s-B)}e^{-a^+B}$ The lower threshold of $x(t_{2n})$is l, after n approaching to the ∞, the $x_(t_{2n})$ should greater the lower threshold. Therefore the budget B should satisfy the following condition: $\frac{(c-k)e^{-a^+B}(1-e^{-a^-}(T_i-B_i))}{1-e^{-a^+B_i}e^{-a^-(T_i-B_i)}} \leq h - k$

Above states the two WCET models. In WCET model, during every period, there is no schedule, the heating time and cooling time is totally same. If the model is WCET model, then the budget must satisfied conditions in equation 2.5 and 2.10 in case of temperatures doesn't within the desired boundaries.

CHAPTER 3
HEATER SCHEDULE

The role of schedule is provide a better way to control the temperature in order to make temperature more stable. Instead of turning heater ON and OFF at boundary all the time, i am trying to find one way to control the temperature automatically regardless of caring about upper and lower threshold. Here, we apply the Constant Bandwidth Server(CBS) in the schedule, introducing the server into our schedule [5] [3].

The CBS is a budget aware extension of EDF scheduler. This algorithm used to assign a dynamic scheduling deadline d_i to the task.The main goal of CBS is to ensure temporal isolation of tasks meaning that a task's execution in terms of meeting deadlines must not be influenced by other tasks as if they were run on multiple independent processors. Here the temporal isolation denoted the worst case finishing time for a task does not depend on the other tasks running in the system. when job arrives, put the job in queue, then CBS become active. After current job finish, next pending job is in queue.

Just considering about the situation of heating system, for example, the temperature in rooms is the state variable. Assume there are two rooms, the temperature in the rooms should be within upper and lower bound which is a safe task. The room temperature must always within the upper and lower interval; otherwise it is unsafe. The server is used to monitor the the system workload on every processor.

Comparing CBS schedule method with Green scheduling method, we can see which one is better for temperature control. [8]

3.1 Key Notations

Table 3.1 summarizes the key notations used in the rest of paper.

3.2 Basic Rules

The CBS is equipped with a set of rules which are as follows:

Table 3.1: key notations

x	state variable
B_i	the maximum budget of the server
C_i	the current budget
d_i	the server deadline
l	lower threshold
h	upper threshold
s_i	constant bandwidth servers
T_i	server period
U_i	server bandwidth

1. Assign the initial d_i and c_i to 0 once the task created.

2. When the job arrival, that means we need to turn on the heater, assign the

$$\begin{cases} d_i = r_i + T_i \\ C_i = B_i \end{cases}$$

3. Based on the EDF scheduling(Earliest Deadline First), server s_i with the earlier deadline has priority to execute first.

4. If $(r_i + \frac{c_i}{U_i} < d_i)$, then we update the deadline and recycle the current budget and deadline since there exist remaining budget when the state already reach to the upper threshold.

$$\begin{cases} d_i = d_i + C_i + T_i \\ C_i = C_i \end{cases} \tag{3.1}$$

5. if budget exhausted, then postpone the deadline and replenish the budget.

$$\begin{cases} d_i = d_i + T_i \\ C_i = B_i \end{cases} \tag{3.2}$$

Above is the basic CBS, in our case, we can see that the maximum budget means the maximum execution time for heater in state ON. Each room is associated with a dynamic deadline d_i, According the deadline, we used EDF scheduling algorithm, assign the higher priority to room which with the earliest deadline. If there are at most k heaters can in mode ON, that means we have k server can be used. At beginning release time, choose k rooms with earliest deadline to turn on the heater and room with lower threshold initial temperature must turn on the heater first. If the some room's budget exhausted, turn off the heater and then update the deadline according to the equation 3.2, next compare with other deadline of idle rooms and still choose the earliest deadline to execute. But if before ran out of the budget the temperature hits the upper threshold, then the deadline should be updated according equation 3.1, the next steps are the same with budget exhausted situation.

3.3 Schedulability Analysis

In the CBS, WCET B_i denoted budget and T_i denoted bandwidth server period, the ratio $U_i = B_i/T_i$ is denoted as the server bandwidth which is fraction of CPU time reserved to task. In our case, that is the maximum execution time for the heater cannot exceed B_i, the deadline of the server need to be updated for every period T_i. CBS assign a dynamic scheduling deadline d_i to each task, and both d_i and current budget C_i are initialized to 0 when the task is created.

In our case, the EDF(Earliest Deadline First) is used(based on the scheduling deadlines assigned by the CBS). That is assign the dynamic deadline to the task first, then make the schedule according to the EDF. Then,

Lemma 2. *The CBS just act as a pure EDF if the served hard task* τ *with parameters*(C_s, T_s) *and* $C_i \leq B_s$ *and* $T_i = T_s$*. where* T_s *is the task's period and* C_s *is task's execution time [1].*

Proof. According to the definition of CBS, each hard task is assigned deadline d_i=r_i+T_i, and maximum budget B_s less thanC_i, each task finishes before the budget is exhausted, so the deadline does not change and is totally same as the one used by EDF. □

CHAPTER 4
SIMULATION RESULTS

I implement the lazy scheduling controller that in Truong's paper in MATLAB, and compare the proposed approach CBS in two examples of small scales. I consider 3 rooms and 4 rooms whose parameters were around mean thermal capacity $C = 5000KJ/K$ and mean thermal conductance $K = 0.35KW/K$.

4.1 Lazy scheduling controller

The principle of lazy controller is very simple. when the state hits to its lower threshold, then it must be switched ON immediately. The reason why it is called lazy is that switching decisions are made only when the task's state reaches either its lower or upper threshold. As long as state in the interior of boundaries, the task stays in their current mode. If tasks at their upper threshold then the heater should be turned OFF and it should be turned ON when tasks at their lower threshold. The schedulibility condition of lazy scheduling controller is if and only if $d > k$ and at most k tasks start at their lower threshold, where k is numbers of heaters can be in mode ON simultaneously at any time, and d is the sum of d_i, $d_i = \frac{{a_i}^{-} l_i - {b_i}^{-}}{{a_i}^{-} l_i - {b_i}^{-} - {a_i}^{+} l_i - {b_i}^{+}}$ [8]. The value of d in my two implementations are all around 1.9 and less than 2 which is heaters quantity that can be in mode ON simultanesouly , so they are all satisfied feasibility condition.

4.1.1 Three rooms simulation of lazy controller

Assume these three tasks as three rooms. In the lazy scheduling controller, these 3 rooms were heated by three identical heaters with 3 heat input rates, $Q = 6, 9, 12(KW)$. Parameters a^{+}, a^{-} for room 1 are 0.000067,0.000067, room2 are 0.000068 and 0.000068, room3 are 0.00007 and 0.000069. Room temperature must be kept between $20°C$ and $22°C$. According to the feasibility analysis of tasks [8], there are at most two tasks can be ON simultaneously and start at their lower thresholds. So Let Room1 and Room2

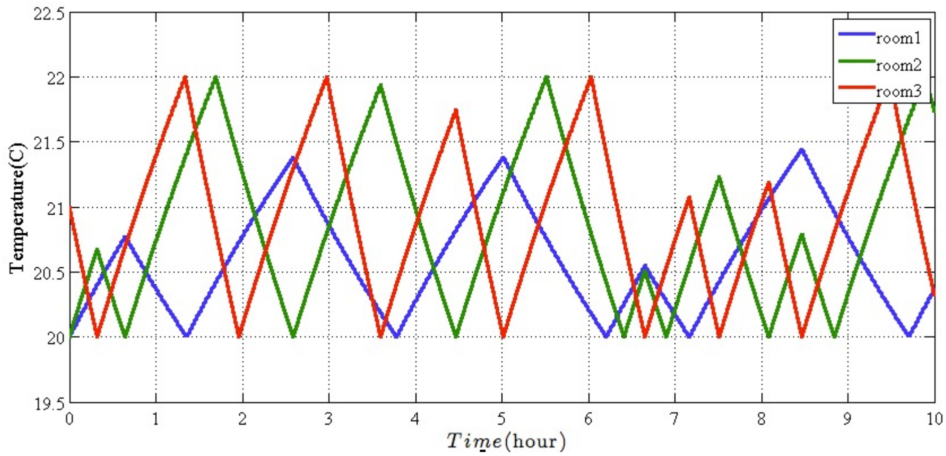

Figure 4.1: Room temperatures for Lazy scheduling controller

start at 20°C, Room3 start at 21°C. The simulation time was 10 hours. In Figure 4.1 are simulated room temperatures that always stay within the desired range, And Figure 4.2 shows the number of heaters simultaneously that at mode ON. We can see from Figure 4.2 that during most time there is 1 heater or 2 heaters be turned ON, only within few time that no heater working is allowable.

4.1.2 Four rooms simulation of lazy controller

Now we can add one more room to see whether the lazy controller can support four room temperatures. The heat input rate$Q$ for the fourth room is 12KW, a^+, a^- are 0.00007 and 0.00005. The initial temperature for this room is 21°C. Still, there are at most two heaters can be turned On simultaneously and d's value is satisfied with feasibility condition. Temperature result is in Figure 4.3, we can see from Figure 4.3 that temperature at any time is within the upper and lower threshold, therefore the controller is working for this 4 rooms situation. Figure 4.4 shows the number of heaters in mode ON simultaneously, it's always two heaters all the time.

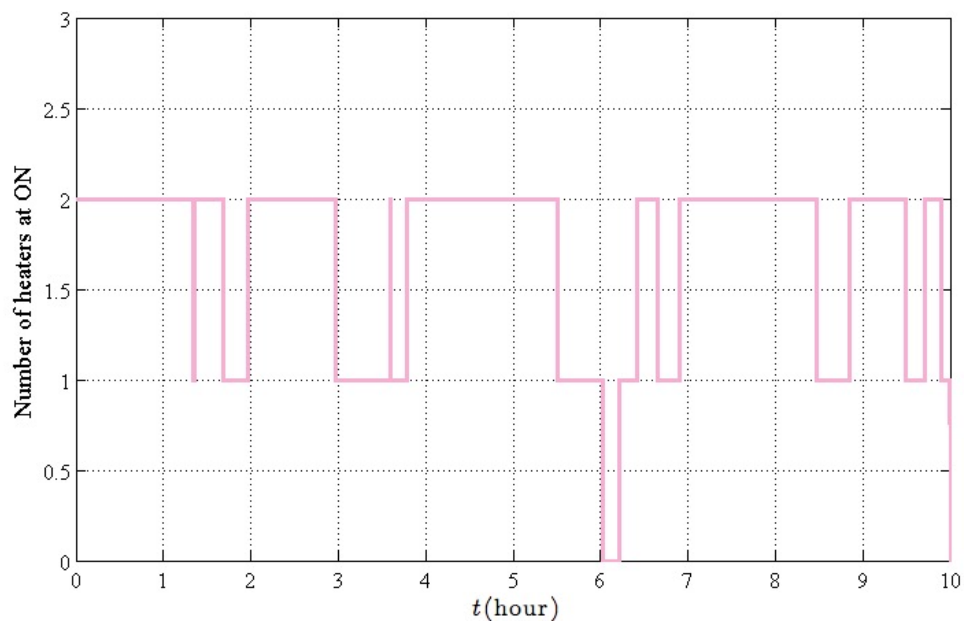

Figure 4.2: Number of heaters at ON mode for Lazy scheduling controller

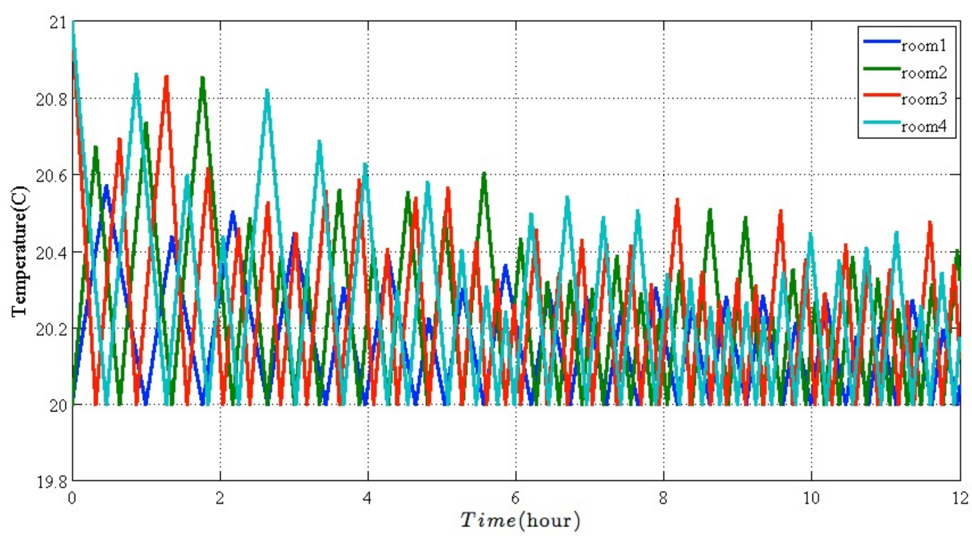

Figure 4.3: Four room temperatures for Lazy controller

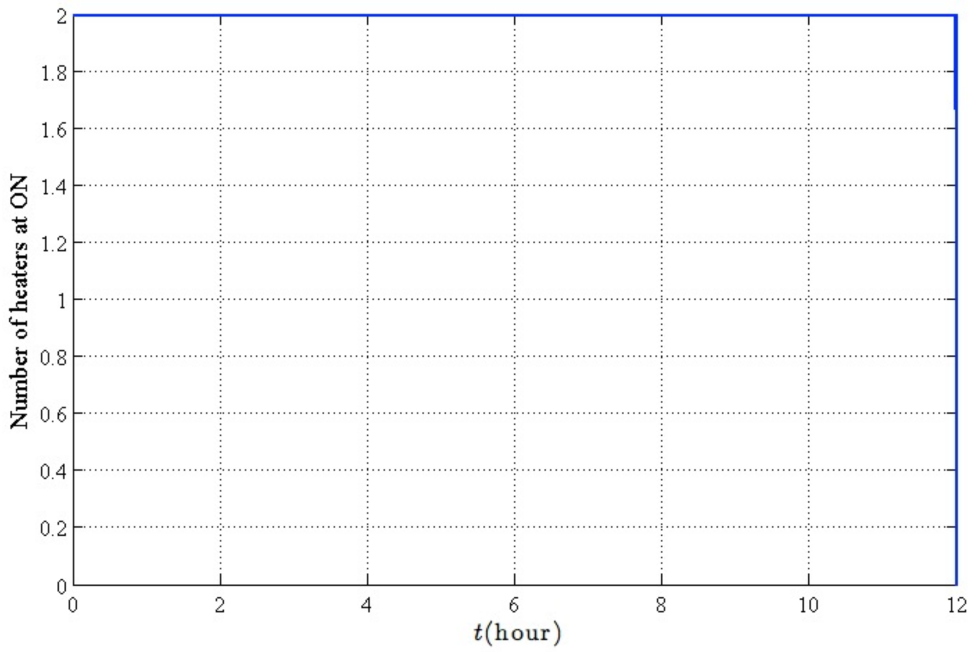

Figure 4.4: Number of heaters at ON simultaneously for Lazy controller

4.2 Constant Bandwidth Server

4.2.1 Three rooms simulation of CBS

In the simulation of CBS, i still consider 3 rooms which are exactly same with the rooms in simulation of Lazy scheduling controller. The steps are as follows: choosing first two rooms with earliest deadlines and turn on the heater of these two rooms, then just execute for budget time. Then if the budget exhausted or hits the upper threshold before ran out of the budget, update the deadline of current room and turn off the heater. Every time whenever exists idle server, compare the remaining inactive rooms and choose rooms with earliest deadline to turn on the heater. All the parameters are same, heat input rate $Q = 3, 6, 9, (KW)$. Room temperature must be kept between $20°C$ and $22°C$. The budget of these three rooms are 250, 288, 316(second), and period T are 300, 348, 407(second). In Figure 4.2 are simulated room temperatures, all the temperatures except room3 are within the desired range, and all the temperatures are around $22°C$. This is because we only have three rooms but two heaters at ON mode all the time, the growth

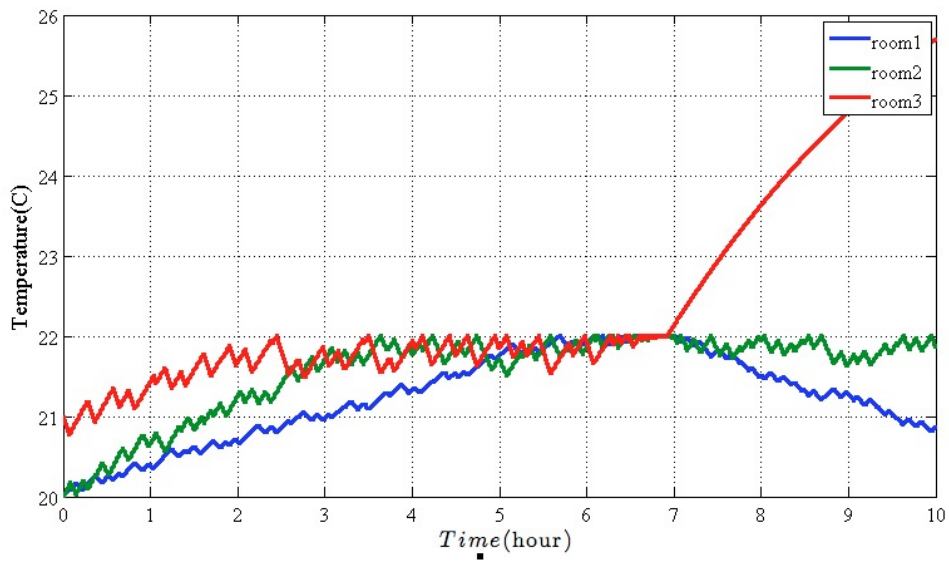

Figure 4.5: Room temperatures for CBS scheduling

of temperature is too fast. Number of heaters at mode ON simultaneously are showed in figure 4.3. So i change the parameters a^+ and a^- which affect the growth and decrease velocity of room temperature. The new a^+ and a^- of these three rooms are 0.000060, 0.000067; 0.000063, 0.000065 and 0.000060. 0.000069. The simulation result is in Figure 4.7. we can clearly see that all the temperatures are within desired ranges.

4.2.2 Four rooms simulation of CBS

Same with implementation of lazy controller, after implementing three rooms, now i add one more room, keep at most two heaters in mode ON. The parameters like a^+, a^-,b^+ and b^- are same with the forth room of lazy controller. The initial temperature $x(t_1)$ of this room is $21°C$, the budget and period are 384,502(second) and input heat rate Q is $12KW$. Temperature result is in Figure 4.7. According to Figure 4.7, it is clearly shows that temperatures of room3 and room4 falling down to the lower threshold $20°C$ which doesn't satisfy the condition that all the temperatures must within boundaries. We can see that the growth velocity is too slow and decay velocity is too fast, one more room cannot support this situation. Then i change the parameters of these four rooms

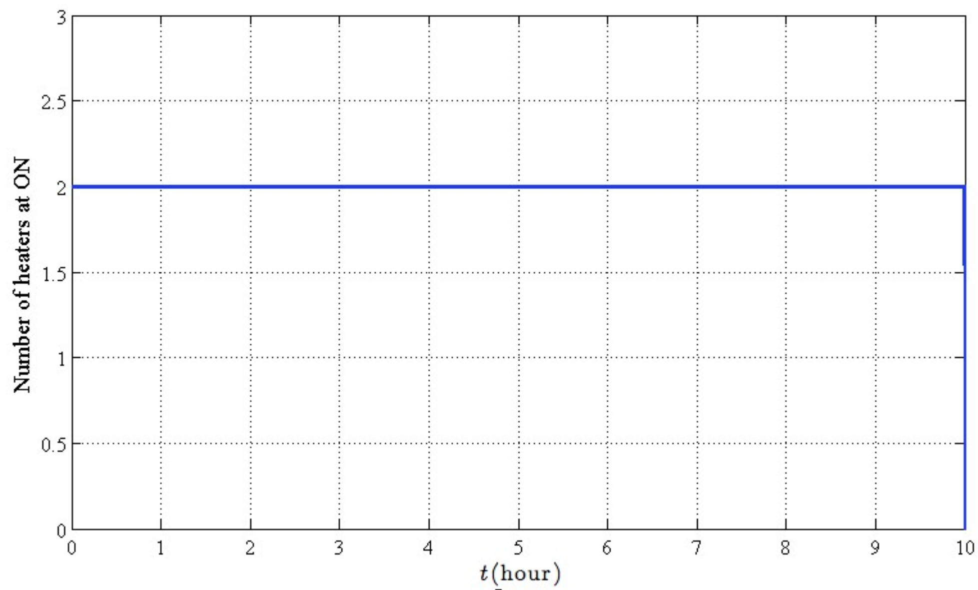

Figure 4.6: No.of heaters at mode ON simultaneously

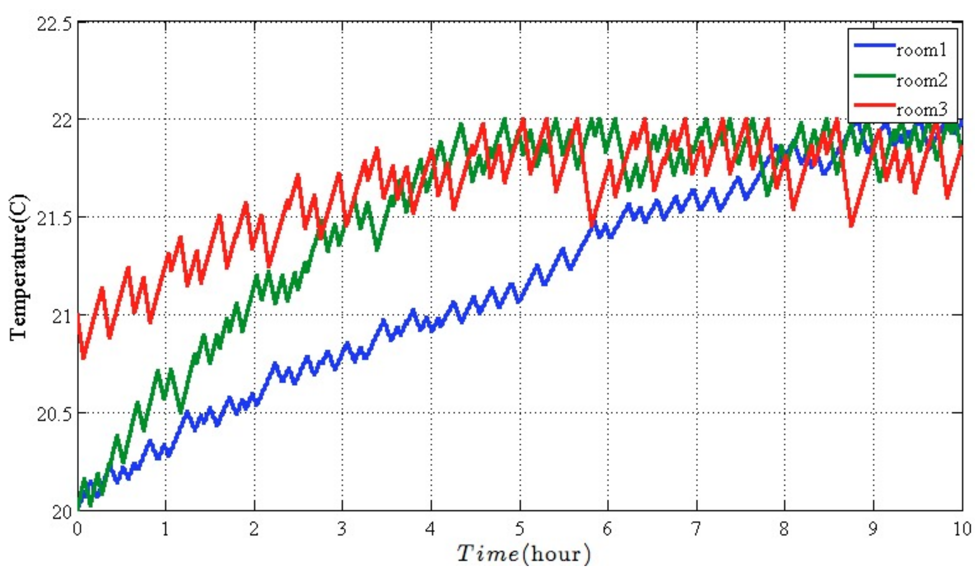

Figure 4.7: Room temperatures for CBS scheduling after changing parameters

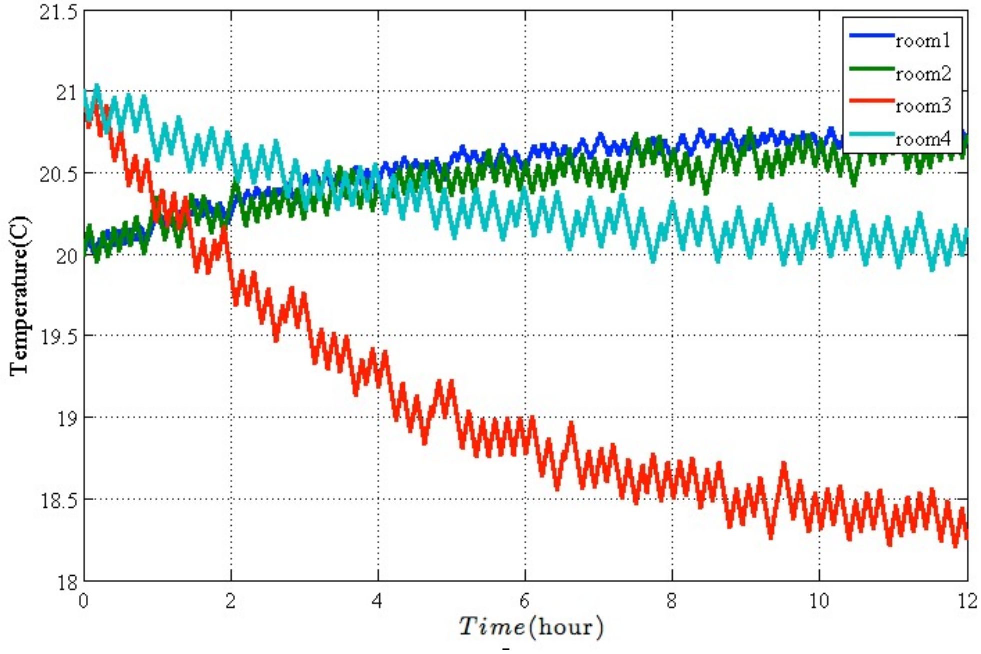

Figure 4.8: Room temperature for four rooms

a^+ and a^-, increase the value of a^+ a little bit to accelerate growth. New a^+ of these four rooms are 0.000075, 0.000080, 0.000095, and 0.000080. The simulation time is still 10 hours, now i get the new result of room temperatures in Figure 4.8. It is clear from Figure 4.8 that after changing the parameters all temperatures are within upper and lower threshold, The heaters quantity at any time for both three rooms and four rooms situation is 2 in CBS since there are at most two heaters can be in mode ON(this i explained in Chapter 3), see in Figure 4.9.

According to Figure 4.3 and Figure 4.8, we can see that using CBS to control temperature, the temperatures all more stable. Unlike lazy controller, temperature hits the upper and lower threshold all the time since the algorithm is keeping heater ON until it hits to the upper boundary or turning heater OFF and make temperature decreasing until it hits to the lower threshold, therefore, the changes of temperature is kind of severe. But as for constant bandwidth server, because the budget is fixed, so the room does not need to be heated to the higher threshold or decreasing until hits to the lower threshold.

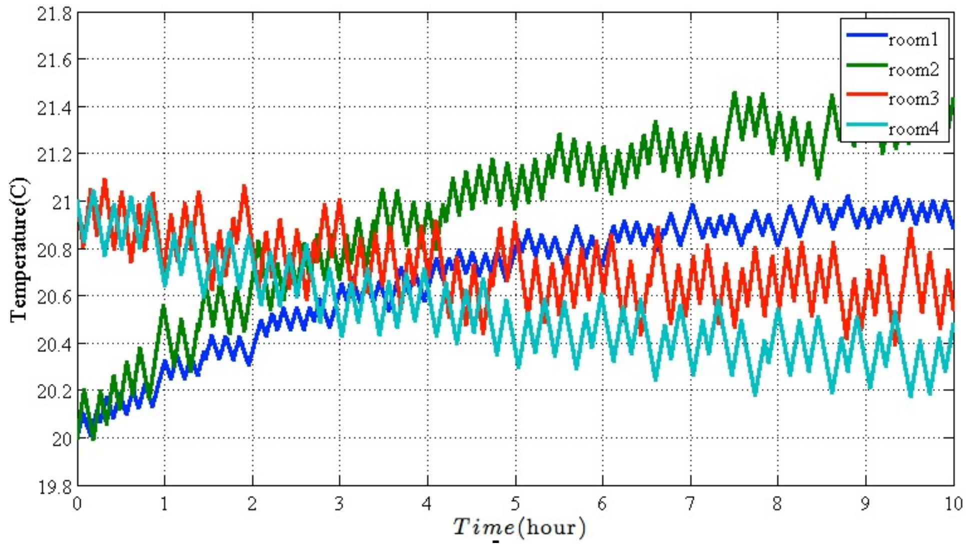

Figure 4.9: Room temperature for four rooms after changing parameters

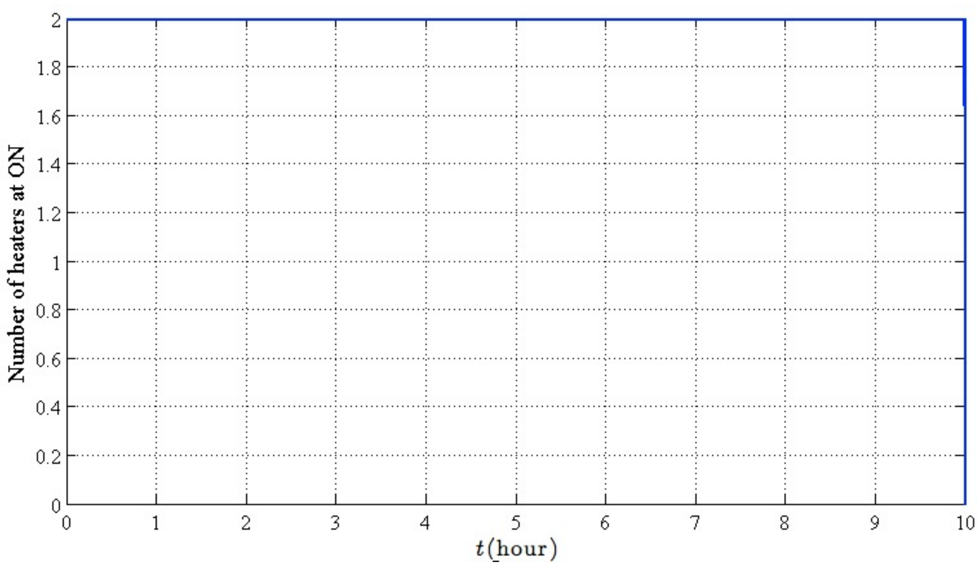

Figure 4.10: Room temperature for four rooms

Under the appropriate parameters, CBS can achieve temperature control successfully and the temperatures are more stable.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this paper, we make comparisons between existing lazy controller and constant bandwidth server for temperature control of specific task models, making assumptions related to worst case execution time task models, implementing both lazy controller and constant bandwidth server of small scales. The proposed method is applying constant bandwidth server to task model which Truong proposed in green scheduling system.

Turing's paper proposed lazy controller, the reason why it is called "lazy" is that this controller doesn't make any schedule, just letting temperature keep temperature increasing or decreasing until it hits upper or lower threshold then turn on or or turn off the heater. By applying constant bandwidth server is different since CBS can provide a schedule to achieve temperature control. Constant bandwidth server is a real-time algorithm, it guarantees fraction of processor time(which is bandwidth, in this paper is budget) assigned to the sever. Our simulation results shows CBS works fine for temperature control and the temperatures under constant bandwidth server scheduling are more stable than lazy controller.

5.2 Future Work

In future work, we will provide simulation results for large scale since this paper only provide small scale's simulation results and analysis. We would like to see constant bandwidth server scheduling can be feasible for large scale even for building systems, whether the CBS could make temperature more stable for large scale or not. We also intend to explore exact feasibility test for constant bandwidth server temperature control, like what's the relationship among rooms number, initial temperature, budget, period and parameters that can apply for all kinds of building systems.

Furthermore, We can investigate more dynamic systems since in this paper the model is fixed to see whether lazy controller or constant bandwidth server scheduling has more flexibility to incorporate complicated systems.

REFERENCES

[1] L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *Real-Time Systems Symposium, 1998. Proceedings., The 19th IEEE*, pages 4–13, Dec 1998.

[2] A. Biondi, A. Melani, and M. Bertogna. Hard constant bandwidth server: Comprehensive formulation and critical scenarios. In *Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on*, pages 29–37, June 2014.

[3] G. Buttazzo and E. Bini. Optimal dimensioning of a constant bandwidth server. In *Real-Time Systems Symposium, 2006. RTSS '06. 27th IEEE International*, pages 169–177, Dec 2006.

[4] Fanxin Kong, Yiqun Wang, Qingxu Deng, and Wang Yi. Minimizing multi-resource energy for real-time systems with discrete operation modes. In *Real-Time Systems (ECRTS), 2010 22nd Euromicro Conference on*, pages 113–122, July 2010.

[5] P. Kumar, Jian-Jia Chen, L. Thiele, A. Schranzhofer, and G.C. Buttazzo. Real-time analysis of servers for general job arrivals. In *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2011 IEEE 17th International Conference on*, volume 1, pages 251–258, Aug 2011.

[6] Bin Li and A.G. Alleyne. Optimal on-off control of an air conditioning and refrigeration system. In *American Control Conference (ACC), 2010*, pages 5892–5897, June 2010.

[7] Yudong Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. *Control Systems Technology, IEEE Transactions on*, 20(3):796–803, May 2012.

[8] T.X. Nghiem, M. Behl, R. Mangharam, and G.J. Pappas. Green scheduling of control systems for peak demand reduction. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 5131–5136, Dec 2011.

[9] T. Nolte and Kwei-Jay Lin. Distributed real-time system design using cbs-based end-to-end scheduling. In *Parallel and Distributed Systems, 2002. Proceedings. Ninth International Conference on*, pages 355–360, Dec 2002.

[10] F. Oldewurtel, A. Ulbig, A. Parisio, G. Andersson, and M. Morari. Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 1927–1932, Dec 2010.

ABSTRACT

COMPARISON OF LAZY CONTROLLER AND CONSTANT BANDWIDTH SERVER FOR TEMPERATURE CONTROL

by

ZHEN SUN

May 2015

Advisor: Dr.Nathan Fisher

Major: Computer Science

Degree: Master of Science

Temperature control plays an important role in building control systems; there are numerous methods for controlling temperature. Recently, a popular controller is the lazy controller which proposed by Truong et al. However, by applying lazy control, the temperature is not stable since this controller simply lets temperature increase or decrease until it reaches the upper or lower temperature thresholds. We seek a heater-control schedule that can make room temperatures more stable. The Constant Bandwidth Server (CBS) was developed to handle soft real-time tasks characterized by the execution time and period. By employing the concepts of CBS, we can derive a CBS budget and deadline for each room; for every budget release time, the temperature can increase/decrease for only a bounded time with the goal of stabilizing each room's temperature. The performance of proposed CBS method is compared with lazy controller through several simulation experiments.

AUTOBIOGRAPHICAL STATEMENT

ZHEN SUN

Zhen Sun, the child of Yi Sun and Guilan Liu, was born in China, Asia. She graduated from Tie Yi High School in June 2009. Then she entered XiDian University majoring Electronic Engineering. After 3 years, she successfully got the bachelor of Science degree. Then she went to Unites States to pursue higher studies in August 2012.