Purdue University
Purdue e-Pubs

International High Performance Buildings Conference | School of Mechanical Engineering

2016

Simulation Based Design and Testing of a Supervisory Controller for Reducing Peak Demand in Buildings

James Nutaro
Oak Ridge National Laboratory, United States of America, nutarojj@ornl.gov

Ozgur Ozmen
Oak Ridge National Laboratory, United States of America, ozmeno@ornl.gov

Jibonananda Sanyal
Oak Ridge National Laboratory, United States of America, jibonananda.sanyal@gmail.com

David Fugate
Oak Ridge National Laboratory, United States of America, fugatedl@ornl.gov

Teja Kuruganti
Oak Ridge National Laboratory, United States of America, kurugantipv@ornl.gov

Follow this and additional works at: http://docs.lib.purdue.edu/ihpbc

Nutaro, James; Ozmen, Ozgur; Sanyal, Jibonananda; Fugate, David; and Kuruganti, Teja, "Simulation Based Design and Testing of a Supervisory Controller for Reducing Peak Demand in Buildings" (2016). *International High Performance Buildings Conference.* Paper 182.
http://docs.lib.purdue.edu/ihpbc/182

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
Complete proceedings may be acquired in print and on CD-ROM directly from the Ray W. Herrick Laboratories at https://engineering.purdue.edu/Herrick/Events/orderlit.html

Simulation Based Design and Testing of a Supervisory Controller for Reducing Peak Demand in Buildings

James NUTARO[1]*, Ozgur OZMEN[2], Jibonanada SANYAL[3], David FUGATE[4], Teja KURUGANTI[5]

[1]Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
nutarojj@ornl.gov
[*]Corresponding author

[2]Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
ozmeno@ornl.gov

[3]Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
sanyalj@ornl.gov

[4]Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
fugatedl@ornl.gov

[5]Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
kurugantipv@ornl.gov

ABSTRACT

We describe a supervisory control strategy for limiting peak power demand by small and medium commercial buildings while still meeting the business needs of the occupants. The objective of the supervisory control is to operate no more than N loads at all times while satisfying equipment limits on acceptable operation, and to exceed N only when these constraints cannot otherwise be satisfied. This is accomplished with a scheduling algorithm that prioritizes the operation of equipment according to data provided with each request to operate. We demonstrate this algorithm using a simulation model of a gymnasium with four roof top air conditioning units. Notably, this simulation is performing using the control software as it will be installed in the building by having the simulation software expose an interface identical to that used for interacting with the air conditioning equipment. The performance of the proposed algorithm relative to uncoordinated operation based on a typical on/off thermostat is also explored via simulation. These simulations suggest that the scheduling algorithm can reduce peak demand under a modest load and that it degrades gracefully under heavy loads while maintaining the temperature of the gymnasium within the desired range.

1. INTRODUCTION

Reducing the peak power demand in a building can reduce electricity expenses for the building owner and contribute to the efficiency and reliability of the electrical power grid. For the building owner, reducing peak power demand can reduce expenses by eliminating peak power charges from electricity bills. For the power system operator, reducing peak power demand leads to a more predictable load profile. Reducing peak power demand favors scheduling the operation of cost-efficient but inflexible power generating resources such as coal and nuclear plants.

This can reduce the need for more flexible, but less efficient and more expensive, generating resources used to meet unanticipated, short-term demand.

This article describes a supervisory control strategy for limiting peak power demand by small and medium commercial building while still meeting the business needs of the occupants. This control strategy has two features that make it relevant to new and existing buildings. First, it is designed to operate with building equipment, such as air condition and refrigeration systems, as they are presently installed in most small and medium commercial buildings. Because of this, the supervisory control could be realized as a software-only retrofit to existing building management systems. Second, the proposed control acts as a supervisory management layer over existing control systems, rather than replacing them outright. The primary idea of this approach is that the controls for individual building equipment request energy resources and the supervisory control examines these requests and decides which to allow while satisfying a limit on peak power demand.

Our approach to developing the control software departs from the common approach of separate modeling and software development tasks. Instead, we build and test the control algorithm from the start in a form that is suitable for deployment by using a simulated environment in all stages of development. Our approach is similar to the model continuity techniques described by Hu and Zeigler (2004), which were introduced to minimize the effort and costs associated with shifting from algorithm development in simulations to the writing of software that will actually implement the control.

Our simulation environment is structured like those that have been used in the aerospace industry to develop high reliability software (see Brown et. al. (2005) and Shankar et. al. (2014)). The modeled environment is built to expose the interfaces that will appear in the fielded hardware or operating system. In our case, the model is built to replace the operating system clock services (i.e., sleep, gettimeofday, etc.) and a Modbus protocol library for talking to HVAC equipment. An important advantage of this approach is that it can enable faster than real time execution of the software, thereby permitting more comprehensive testing than could be achieved in a hardware laboratory. This aspect of our development method will become increasingly important as building control software grows in complexity with a corresponding increase in the likelihood of software defects and the economic impact of software faults in fielded control systems.

2. CONTROL ALGORITHM

The proposed control strategy borrows key concepts from prior theoretical work on scheduling electrical loads to limit power consumption while meeting criteria for the satisfactory performance of the loads (see, in particular, Vedova and Facchinetti (2013) and Nghiem et. al. (2011)). Our novel contributions are (i) the introduction of a priority scheme that ensures graceful degradation of the control process during unusually hot or cold weather when the desired peak power demand cannot be satisfied while maintaining an acceptable temperature; and (ii) the ability to effectively manage the HVAC load without requiring a predictive model of the building.

The essential ingredients of ours and prior control strategies are assumptions concerning the behavior of the electrical loads and the information that is available regarding their operation. In our approach, we make the following assumptions:

1. Each electrical load has its own control strategy that determines when the load should be idle and when it should be active according to its application. Information concerning this strategy is not available to the supervisory control. In particular, the supervisory control does not know when the loads will desire to operate and what the desired duration of that operation will be.
2. The electrical load supplies a priority with its request to operate. A priority of zero indicates that the load does not need to operate, and there is a maximum priority M that indicates the load must begin operating immediately. Other priority values indicate the urgency for operation within these extremes, with higher priority indicated a greater need to operate.
3. There is some number N of electrical loads that can operate simultaneously while still reducing peak demand charges.

The supervisor maintains a list of equipment sorted in decreasing order of priority, where priority values may be any integer between zero and M inclusive. Upon receiving a change in priority from any piece of equipment, the order of the list is updated and the scheduling algorithm is run. The scheduling algorithm has three steps. First, it scans the list of equipment from front to back, requesting that each equipment item shutdown if it satisfies one of two conditions:

(1) if the equipment item has a priority equal to zero; or
(2) the equipment item has a priority less than M and its position in the list is greater than N.

These requests can be disregarded by the equipment if necessary to avoid violating some operating constraint; for example, an HVAC unit may ignore this request to avoid rapid cycling of its compressor. Therefore, the second step is to scan the list again and count the number R of equipment items that are currently running. In the third step, equipment is activated according to its priority and position in the list. The list is scanned again from front to back and equipment is instructed to operate if it satisfies one of two conditions:

(1) if $R < N$ and the item occupies a position in the list less than $N - R$ or;
(2) the item has a priority equal to M.

Under extreme conditions, this control strategy will favor satisfactory operation of the equipment over reducing peak demand. This occurs when more then N units must operate because too many have reached the maximum priority. Otherwise, the system will respect the N unit limit and keep peak demand beneath N times the maximum power consumption of any equipment item.

To demonstrate that there exists a set of conditions under which the control can satisfy its limits, let us make the following assumptions. Each item of equipment can switch between on and off with a frequency of at most $1/T$. Furthermore, there is h such that the priority $p(t)$ of the equipment evolves according to

$$p(t+h) = \begin{cases} p(t)+1 & \text{if the equipment is off} \\ p(t)-1 & \text{if the equipment is on and } p(t) > 0 \\ 0 & \text{otherwise} \end{cases}$$

and the initial priority is $p(0) = 0$. We can normalize time by the smallest allowed switching interval and rewrite this expression as

$$p(t^* + h/T) = \begin{cases} p(t^*)+1 & \text{if the equipment is off} \\ p(t^*)-1 & \text{if the equipment is on and } p(t^*) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $t^* = t/T$.

To satisfy the constraint that the priorities remain less than or equal to M, it is necessary for the sum of the priorities over all equipment items to be bounded for all t^*. Let there be E equipment items in total. Clearly, $\sum p(0) = 0$ and $\sum p(h/T) = E$. Because $N < E$, it will be the case at each subsequent step in time that $E - N$ equipment items are off and increase their priority by one and N equipment items are on and decrease their priority or have it remain at zero. Therefore, at each step in time we add $E - N$ to the sum of priorities and subtract N, for a total change in the sum of $E - 2N$. The total priority at time step kh/T will be

$$\sum p(kh/T) = kE - 2(k-1)N$$

We prove this by induction on k. For $k = 1$ we already know that $\sum p(h/T) = E$ which agrees with the above expression. Suppose it is true for kh/T and consider $(k+1)h/T$. Adding $E - 2N$ to $\sum p(kh/T)$ gives

$$kE - 2(k-1)N + E - 2N = (k+1)E - 2kN + 2N - 2N = (k+1)E - 2kN = \sum p((k+1)h/T)$$

as expected.

To bound the sum of the priorities, it is necessary for there to be some k such that for all subsequent steps

$$2(k-1)N = kE$$

Rearranging this expression for N we have

$$N = \frac{kE}{2(k-1)} = \frac{E}{2-1/k}$$

and as $k \to \infty$ we get $N = E/2$. Hence, we can expect the control to bound the priority of each equipment item when $N \geq E/2$. In this case, the sum of the priorities is

$$\sum p(kh/T) = kE - 2(k-1)E/2 = kE - \frac{2kE}{2} + \frac{2E}{2} = E$$

and so if $M \geq E$ and $N \geq E/2$ the control scheme can satisfy its constraints.

This derivation demonstrates that there are plausible conditions under which the proposed control limits peak demand. We will show in the next section that this derivation can also serve as a practical guide for selecting N when installing the control in a building.

3. SIMULATION BASED DEVELOPMENT AND TESTING

The new control algorithm was implemented in software that reuses large parts of a prior, deployed control system (see Nutaro et. al. (2014)). Briefly, this existing system is installed in the basketball gymnasium of the Family Life Center at the Central Baptist Church in Fountain City, Tennessee (see Figure 1). This gymnasium has four 10 ton HVAC roof top package units each having two-stage gas heat and two-stage conventional refrigerant cooling. The prior, deployed control software is hosted on a Linux PC with connections via Modbus to four Tempstat 6 thermostats from Temco Controls. The parts of this software that we have reused to implement the new control enables direct switching of the thermostat relay contacts, reading the temperature sensors, and reading the temperature set point. This portion of the software uses the libmodbus library (see libmodbus.org) to communicate with the thermostats.

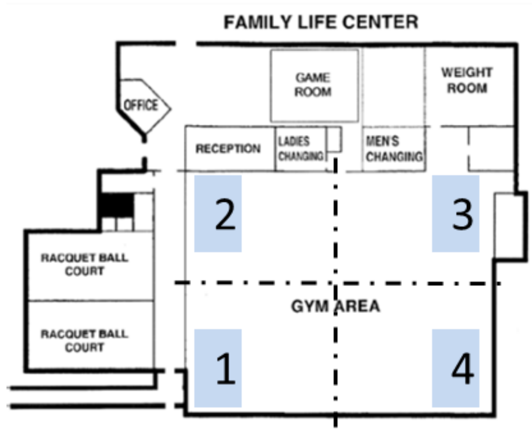

Figure 1: Floor plan for the gymnasium. Dotted lines in the gym show the HVAC zone boundaries.

To facilitate testing prior to deployment, we implemented a simulated building that is accessed by the new control software via simulated Modbus library calls and operating systems calls. The simulated Modbus library and system calls are implemented in a software library that is linked with the control software for testing. For deployment, the control software would be linked with the actual Modbus library and operating system functions.

In testing, a call to one of the simulated Modbus or operating systems functions causes the building model to be advanced in time by the amount of wall clock time that has passed since the prior, simulated call. Then the effect of the call is determined by referencing the appropriate element of the building model. For instance, if the call was to send a Modbus message that closes a relay, the corresponding HVAC unit is activated in the model; if the call was to read a temperature sensor, then the corresponding model variable value will be returned. Calls to sleep are unique in that they advance the simulation clock as described above plus an amount of time equal to the requested sleep period. This approach is illustrated in Figure 2 for a code segment that periodically samples the temperature sensor.

Testing in this way allows the simulated software to operate more quickly that real time, greatly facilitating testing by compressing days of simulated software operation into several minutes of real time.

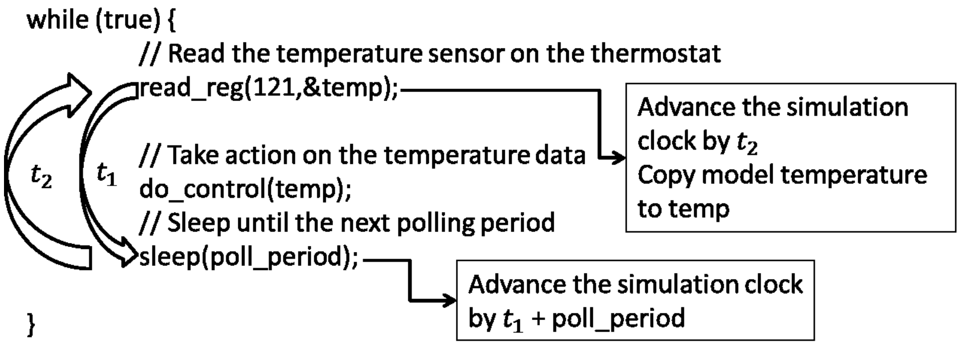

Figure 2: Redirecting calls to read_reg and sleep to interact with a simulated building

To demonstrate the operation of the proposed control and its software implementation, we constructed a model of a four zone building that is based on our prior experience with the Family Life Center (see Nutaro et. al. (2014)), and we consider only summer months when electrical power is used to cool the building. The model HVAC equipment has two cooling stages, which gives its three modes of operation – off, stage 1 on, and both stages on. An HVAC unit will not change its mode of operation more often than once every ten minutes. The HVAC equipment has 10 priority levels, and the priority p of the air conditioner is based on its thermostat temperature T [C] and temperatures set point S [C] as follows:

$$p = \begin{cases} \text{ceil}\left(\frac{T-S}{0.1}\right) & \text{if } 0 < T - S < 1 \\ 0 & T \leq S \\ 10 & T - S \geq 1 \end{cases}$$

The first stage is activated if $T \leq S + 1$ and the second stage is activated if $T > S + 1$.

The model equations that describe temperature change are listed below. The R_{ik} terms model heat flow between the ith and kth zones or the outside air; the T_k terms are the zone air temperatures; T_a is the outside air temperature with daily low T_{low} and daily high T_{high}; the Q_{sk} is solar heating, occupants, and other sundry sources with maximum daily heating of Q_k in zone k; the h_k is cooling power provided by the HVAC unit in zone k; u_k is the operating condition of the HVAC unit which can be 0 (off), 1 (stage 1 on), or 2 (stages 1 and 2 on); and C_k is the zone's thermal capacitance. The time t is in seconds.

$$T_1 = \frac{1}{C_1}\left(\frac{T_2 - T_1}{R_{12}} + \frac{T_4 - T_1}{R_{14}} + \frac{T_a - T_1}{R_{a1}} + Q_{s1} + u_1 h_1\right)$$

$$T_2 = \frac{1}{C_2}\left(\frac{T_1 - T_2}{R_{12}} + \frac{T_2 - T_3}{R_{23}} + \frac{T_a - T_2}{R_{a2}} + Q_{s2} + u_2 h_2\right)$$

$$T_3 = \frac{1}{C_3}\left(\frac{T_2 - T_3}{R_{23}} + \frac{T_4 - T_3}{R_{34}} + \frac{T_a - T_3}{R_{a3}} + Q_{s3} + u_3 h_3\right)$$

$$T_4 = \frac{1}{C_4}\left(\frac{T_1 - T_4}{R_{14}} + \frac{T_3 - T_4}{R_{34}} + \frac{T_a - T_4}{R_{a4}} + Q_{s4} + u_4 h_4\right)$$

$$T_a = T_{low} + (T_{high} - T_{low})\frac{1 + \cos\left(2\pi \frac{t}{86400}\right)}{2}$$

$$Q_{sk} = Q_k \frac{1 + \cos\left(2\pi \frac{t}{86400}\right)}{2}$$

We simulate the operation of this building with several choices of parameters selected at random from the uniform distributions listed in Table 1; only the set point is fixed across all runs at 21.111 C (70 F). Though the model is based roughly on the Family Life Center such that it yields similar high and low temperatures for a summer day and roughly approximates the rates of temperature change seen in the gymnasium, no attempt has been made to validate the model against experimental data. Nonetheless, the randomized parameter sets represent a plausible range of values for this building and serve to demonstrate the robustness of the control across a range of similar buildings and circumstances.

Table 1: Ranges for parameter values

Parameter	Value range	Units
C_k	252,628.5 ± 10%	Joules / Degrees Centigrade
R_{ik}	1000 ± 10%	Watts / Degrees Centigrade
R_{ai}	10 ± 10%	Watts / Degrees Centigrade
Q_k	1,710 ± 10%	Watts
h_k	17,500 ± 10%	Watts
S	21.111 (70)	C (F)

To evaluate the performance of the control algorithm, we conducted one thousand simulations with this model using a randomly selected set of parameters in each simulation. This was repeated with $N = 1, \ldots, 4$. The case with $N = 4$ is equivalent to uncoordinated control of the loads, as would occur in a building with common thermostat equipment. The simulations each spanned 10 days and we recorded the maximum number of simultaneously operating units (peak load) and the maximum temperature above the thermostat set point (temperature error) that was observed over those ten days.

The distribution of peak load and temperature error for each choice of N is shown in Table 2. In these results it is clear that the control reduces the expected peak load from 3 to approximately 2 with only a small change in the maximum and average errors. In all cases these metrics are within 2/10 of a degree Centigrade. Referring to the prior theoretical consideration of the minimum value for N that can be reliably achieved, the simulations show that $N = 2$ is maintained across 91% of the scenarios considered. On the other hand, this simulation data also suggests an optimistic strategy in which $N = 1$ is selected and the natural degradation of the control is relied upon to maintain an acceptable temperature error. This strategy produces an expected peak load only slightly worse than the $N = 2$ case, while occasionally beating that case by running just a single unit.

Table 2: Simulation results

N	Distribution of peak load across runs				Expected peak load	Temperature error [C]		
	1	2	3	4		Min.	Avg.	Max.
1	5.7%	79%	14.5%	0.8%	2.104	0.4069	0.7601	1.343
2	0%	91.4%	7.9%	0.7%	2.093	0.0590	0.6784	1.332
3	0%	1.8%	97.1%	1.1%	2.993	0.0587	0.6114	1.321
4	0%	2.5%	82.3%	15.2%	3.127	0.0587	0.6276	1.344

4. CONCLUSIONS

The simulation results show that theoretical limits on performance calculated for an idealized model are satisfied in the majority of variations of a model derived from a real building. This is encouraging, but not conclusive. We intend to install the control software in the Family Life Center and observe its performance through the summer of 2016. This experiment will be greatly facilitated by having developed and tested the control software in a simulated environment that looks, from the point of view of the control application, indistinguishable from the real building. Indeed, the simulation results reported in the previous section have also provided us with the equivalent of 4,000 days of operational testing for the primary control logic, and this should substantially reduce the amount of experimental time lost to software errors. Though we have demonstrated the proposed control in the context of limiting peak demand caused by HVAC equipment, it can be readily extended to coordinate refrigeration, ventilation, lighting and other energy intensive systems by defining suitable priority schemes. These types of simple extensions are possible chiefly because of the limited information that is required by the control, and this feature most strongly distinguishes the proposed system from prior work on load scheduling in small and medium commercial buildings.

REFERENCES

Vedova, M. L. D., Facchinetti, T. (2013). Real-time scheduling for peak load reduction in a large set of HVAC loads. In *Proceedings of the third international conference on smart grids, green communications, and IT energy-aware technologies* (161-166). Lisbon, Portugal: IARIA

Nghiem, T., Behl, M, Pappas, G.J., & Mangharam, R. (2011). Green scheduling: scheduling of control systems for peak power reduction. In *Proceedings of the 2011 international green computing conference and workshop* (1-8). Orlando, Florida, USA: IEEE

Nutaro, J., Kuruganti, T., Fugate, D., & Starke, M. (2014). An inexpensive retrofit technology for reducing peak power demand in small and medium commercial buildings. In *Proceedings of the 3rd international high performance buildings conference at Purdue* (1-7). West Lafayette, Indiana, USA: Purdue University.

Hu, X., Zeigler, B.P. (2004). Model Continuity to Support Software Development for Distributed Robotic Systems: A Team Formation Example. *Journal of Intelligent and Robotic Systems*, 39(1), 71-87.

Brown, J., Lam, D., Chang, L., Burk, T., & Wette, M. (2005). The role of the Flight Software Development System simulator throughout the Cassini mission. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit* (1-13). San Francisco, California: AIAA

Shankar, S., Desai, K., Dutta, S., Chetwani, R., Ravindra, M., & Bharadwaj, K.M. (2014). Mission critical software test philosophy: a SILS based approach in Indian Mars Orbiter mission. In *Proceedings of the 2014 International Conference on Contemporary Computing and Informatics* (414-419). Mysore, India: IEEE

ACKNOWLEDGEMENT

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).