Cost Optimal Operation of Thermal Energy Storage System with Real-Time Prices

Toru Kashima, Member, IEEE and Stephen P. Boyd, Fellow, IEEE

***Abstract*— In this paper we propose a method to optimize operation of a thermal energy storage (TES) system for heating, ventilation and air conditioning (HVAC) in terms of electricity cost. We pose this optimization problem as a mixed integer linear programming (MILP) problem where future thermal demand and electricity prices are predicted. The proposed method uses a branch and bound algorithm to solve the problem, using linear relaxation of the integer variables that represent future on-off states of the equipment. We conduct simulations based on real building data, which show that significant cost reduction can be obtained.**

I. INTRODUCTION

Cutting peak electricity demand has been a serious issue in Japan especially after the Great East Japan Earthquake in 2011 since the country's electricity supply capacity significantly declined mainly due to shut down of almost all of the nuclear plants [3]. Considering building energy consumption, using thermal energy storage (TES) is one of the most effective ways to cut peak demand. TES is a concept of storing energy in the form of heat or cold for future use [2]. If electricity price is higher through peak time, consumers have an incentive to use a TES system to shift their demand.

To use a TES system cost-effectively, for example, the operation for the next day is defined taking account of thermal demand for the TES system and electricity prices for the next day [4]. The problem can be treated as a mixed integer linear programming (MILP) problem if true thermal demand and true electricity prices are given. The true thermal demand can't be obtained beforehand; therefore thermal demand prediction is necessary and the prediction error can lead to deterioration of the result [4]. The same can be said for time varying real-time prices. Real-time energy pricing is not yet introduced on a large scale in Japan at this point (in June 2013). But in the near future this structure will become more common as smart grid is introduced [4].

In this paper we propose a method to operate a TES system cost-effectively under the condition that thermal demand prediction has an error and the real-time pricing rate structure is introduced. The proposed method uses a branch and bound algorithm with a custom linear programming (LP) solver which is generated by CVXGEN [1] to solve the MILP problem consecutively. While the branch and bound algorithm is executed, some of integer variables, which represent on-off statuses of equipment in the future, remain linearly relaxed.

We developed a MILP solver in C using the proposed method and conducted simulations based on real building data. The results were compared to the conventional operations.

II. PROBLEM

A. Thermal Energy Storage System

A TES system we consider in this paper consists of thermal energy storages and heat sources such as chillers. Energy resources such as electricity or natural gas are bought from suppliers at certain prices. For simplicity, we assume that there is only one kind of energy resource, electricity. The electrical energy is transformed into thermal energy by the heat sources. The thermal energy has to meet the demand from the downstream air-conditioning system. Thermal energy storage systems can store thermal energy for a while. In other words the storages can delay the timing of thermal energy usage from electricity energy usage. Fig. 1 shows the energy flow of a TES system.

In Fig. 1, n_s is the number of storages (with its own chiller) and n_d is the number of support chillers which are used when the stored energy is insufficient.

B. MILP Formulation

The characteristics of the chillers and the support chillers are represented as

$$\begin{array}{c} l_u^{[i]} x_{[t]}^{[i]} \le u_{[t]}^{[i]} \le h_u^{[i]} x_{[t]}^{[i]} \\ l_v^{[j]} y_{[t]}^{[j]} \le v_{[t]}^{[j]} \le h_v^{[j]} y_{[t]}^{[j]} \end{array} \quad (1)$$

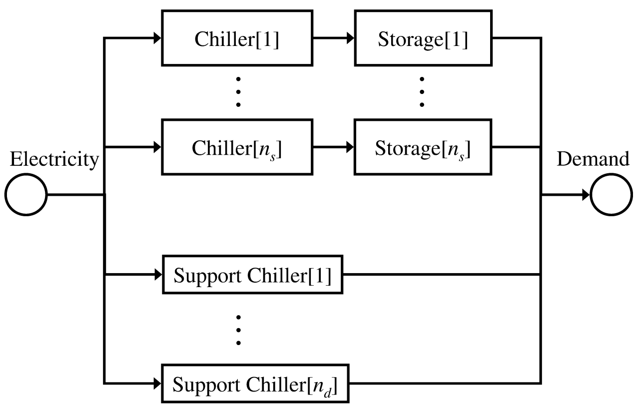

Fig. 1. Energy flow of thermal energy storage system

Manuscript received June 15, 2013.
Toru Kashima is with Azbil Corporation, Kanagawa, Japan. t.kashima.74@azbil.com
Stephen P. Boyd is with Information Systems Laboratory, Department of Electrical Engineering, Stanford University, CA, USA. boyd@stanford.edu

where i means the i-th chiller and j means the j-th support chiller; $u_{[t]}^{[i]}$ and $v_{[t]}^{[j]}$ are thermal energy generated by the i-th chiller and j-th support chiller at time t, respectively; $l_u^{[i]}$, $l_v^{[j]}$, $h_u^{[i]}$ and $h_v^{[j]}$ are the lower and upper bounds on the generated thermal energy. $x_{[t]}^{[i]}$ and $y_{[t]}^{[j]}$ are integer (Boolean) variables which represent on (1) or off (0) of the i-th chiller and the j-th support chiller at time t respectively.

The characteristics of the storages are represented as

$$\begin{array}{c} z_{[t+1]}^{[i]} = (1 - \epsilon^{[i]})(z_{[t]}^{[i]} + u_{[t]}^{[i]} - w_{[t]}^{[i]}) \\ l_z^{[i]} \le z_{[t]}^{[i]} \le h_z^{[i]} \\ 0 \le w_{[t]}^{[i]} \end{array} \quad (2)$$

where $z_{[t]}^{[i]}$ is stored energy in the i-th storage at time t; $u_{[t]}^{[i]}$ is input energy at time t for the storage, which is output energy of the chiller connected to the storage at the same time; $w_{[t]}^{[i]}$ is output energy of the storage at time t; $\epsilon^{[i]}$ is the heat loss coefficient of the i-th storage; $l_z^{[i]}$ and $h_z^{[i]}$ are the lower and upper bounds of the stored energy in the i-th storage.

The demand for the TES system must be equal to the output of the system. This relation is represented as

$$d_{[t]} = \sum_{i}^{n_s} w_{[t]}^{[i]} + \sum_{i}^{n_d} v_{[t]}^{[j]} \quad (3)$$

where $d_{[t]}$ is the demand at time t.

The cost function f is expressed as

$$f = \sum_{t=1}^{T} \left(p[t] \left(\sum_{i}^{n_s} \frac{u_{[t]}^{[i]}}{c_x^{[i]}} + \sum_{j}^{n_d} \frac{v_{[t]}^{[j]}}{c_y^{[j]}} \right) \right) \quad (4)$$

where $p[t]$ is the electricity price at time t. $c_x^{[i]}$ and $c_y^{[j]}$ are the coefficients of performance (COP) of the i-th chiller and the j-th support chiller respectively. The objective of the problem is to minimize f.

From (1)-(4), the problem is defined as a standard MILP form (5):

$$\begin{array}{ll} \text{minimize} & f(\tilde{x}) \\ \text{subject to} & A\tilde{x} = b \\ & G\tilde{x} \le h \end{array} \quad (5)$$

where $\tilde{x} = (x, y, z, u, v, w,)$; $x \in \{0,1\}^{n_s T}$, $y \in \{0,1\}^{n_d T}$, $z \in \mathbf{R}^{n_s T}$, $u \in \mathbf{R}^{n_s T}$, $v \in \mathbf{R}^{n_d T}$ and $w \in \mathbf{R}^{n_s T}$ are the variable vectors and the elements of those are $x_{[t]}^{[i]}$, $y_{[t]}^{[j]}$, $z_{[t]}^{[i]}$, $u_{[t]}^{[i]}$, $v_{[t]}^{[j]}$ and $w_{[t]}^{[i]}$ ($i = 1, 2, \ldots, n_s$; $j = 1, 2, \ldots, n_d$; $t = 1, 2, \ldots, T$), respectively; A, b, G and h are constant matrices and vectors and the elements of those are easily defined from (1)-(4).

The problem (5) is an operation scheduling problem that is to decide how to operate the heat sources and the storages at time $t = 1, 2, ..., T$.

However, the demand $d_{[t]}$ can't be obtained beforehand in practice; therefore it is necessary to build a demand predictor. The electricity price $p_{[t]}$ is also unknown if the real-time pricing rate structure is introduced. To handle this we use a standard model predictive control (MPC; also called receding horizon control, RHC) method, using predictions of future demand and prices; see, e.g., [6].

III. METHOD

A. Branch and Bound Algorithm

We use a branch and bound algorithm [7] to solve the problem (5). The algorithm finds the global minimum of a function $f : \mathbf{R}^n \longrightarrow \mathbf{R}$ over an n-dimensional set, or search space, $\mathcal{Q}_{\text{init}}$. This section shows the general algorithm.

For a space $\mathcal{Q} \subseteq \mathcal{Q}_{\text{init}}$ we define

$$\Phi(\mathcal{Q}) = \inf_{x \in \mathcal{Q}} f(x). \quad (6)$$

We also define Φ_{lb} and Φ_{ub} which compute lower and upper bounds on $\Phi_{\min}$, respectively:

$$\Phi_{\text{lb}}(\mathcal{Q}) \le \Phi_{\min}(\mathcal{Q}) \le \Phi_{\text{ub}}(\mathcal{Q}). \quad (7)$$

The branch and bound algorithm has two procedures: one is called "branching" and the other "bounding." The branching procedure splits a search space into two smaller spaces. The bounding procedure computes lower and upper bounds.

The two procedures are iterated one after the other until the difference between the minimum of the lower bounds and the minimum of the upper bounds becomes lower than the tolerance that is set in advance.

If $\Phi_{\text{lb}}(\mathcal{Q}_I) \ge \Phi_{\text{ub}}(\mathcal{Q}_{II})$ where $\mathcal{Q}_I \subseteq \mathcal{Q}_{\text{init}}$, $\mathcal{Q}_{II} \subseteq \mathcal{Q}_{\text{init}}$ and $\mathcal{Q}_I \cap \mathcal{Q}_{II} = \emptyset$, then the search space $\mathcal{Q}_I$ can be eliminated from consideration obviously. This procedure is called "pruning". Pruning is not necessary to be done while the algorithm proceeds, but it can reduce computer storage requirements.

The algorithm (without pruning) is summarized below.

$$\begin{array}{l} k = 0; \\ \mathcal{L}_0 = \{\mathcal{Q}_{\text{init}}\}; \\ L_0 = \Phi_{\text{lb}}(\mathcal{Q}_{\text{init}}); \\ U_0 = \Phi_{\text{ub}}(\mathcal{Q}_{\text{init}}); \\ \text{while } U_k - L_k > \epsilon, \{ \\ \quad \text{pick } \mathcal{Q} \in \mathcal{L}_k \text{ for which } \Phi_{\text{lb}}(\mathcal{Q}) = L_k; \\ \quad \text{split } \mathcal{Q} \text{ into } \mathcal{Q}_I \text{ and } \mathcal{Q}_{II}; \\ \quad \text{form } \mathcal{L}_{k+1} \text{ from } \mathcal{L}_k \text{ by removing } \mathcal{Q} \\ \quad\quad \text{and adding } \mathcal{Q}_I \text{ and } \mathcal{Q}_{II}; \\ \quad L_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{lb}}(\mathcal{Q}); \\ \quad U_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{ub}}(\mathcal{Q}); \\ \quad k := k + 1; \\ \} \end{array} \quad (8)$$

where k is the iteration index; $\mathcal{L}_k$ denotes the list of search spaces; L_k and U_k denote the lower and upper bounds for $\Phi_{\min}(\mathcal{Q}_{\text{init}})$ at the end of k iterations; ϵ is the tolerance.

B. Linear Relaxation

To apply the branch and bound algorithm above to the problem (5), linear relaxation given below is used:

$$\begin{array}{ll} \text{minimize} & f(\tilde{x}) \\ \text{subject to} & A\tilde{x} = b \\ & G\tilde{x} \le h \\ & 0 \le x_{[t]}^{[i]} \le 1 \\ & 0 \le y_{[t]}^{[j]} \le 1 \end{array} \tag{9}$$

where $x_{[t]}^{[i]}$ and $y_{[t]}^{[j]}$ are not Boolean variables. The problem (9) is a linear programming (LP) problem; this can be solved by an LP solver.

The optimum value of (9) is a lower bound L_0 of the original problem (5). An upper bound U_0 is obtained by rounding each $x_{[t]}^{[i]}$ and $y_{[t]}^{[j]}$ of the solution of (9) to 0 or 1. If the rounded solution is infeasible, the upper bound is $+\infty$.

The branching procedure is to pick one element from x or y to fix its value to 0 and 1. Our method picks the element which has the closest value to 0.5 regarding the solution of the problem (9). If the picked element is $x_{[1]}^{[1]}$, for example, the inequality constraints of $x_{[1]}^{[1]}$ is changed to $0 \le x_{[1]}^{[1]} \le 0$ to generate one new problem. On the other hand, $1 \le x_{[1]}^{[1]} \le 1$ is used instead to generate the other new problem. These new problems are also LP; branching and bounding can be done as well. These procedures are iterated to obtain the solution of the problem (5).

The reason why we change the inequality constraints instead of adding new equality constraints is that we use a custom LP solver generated by CVXGEN [1]. CVXGEN exploits the sparsity of a problem to generate a fast QP solver which uses primal-dual interior point methods with Mehrotra predictor corrector. (QP obviously includes LP.) The custom LP solver solves the problem very rapidly but the structure of the problem must remain the same. If the branch and bound algorithm splits the problem changing inequality constraints as we mentioned above, the structure does not change.

C. Linear Relaxation in the Future

Here we define the problem instead of (5) to cope with erroneous demand prediction and real-time pricing.

$$\begin{array}{ll} \text{minimize} & f(\tilde{x}) \\ \text{subject to} & A\tilde{x} = b \\ & G\tilde{x} \le h \\ & \begin{cases} x_{[t]}^{[i]} \in \{0,1\} \ (t \le t_s) \\ 0 \le x_{[t]}^{[i]} \le 1 \ (t_s < t) \end{cases} \\ & \begin{cases} y_{[t]}^{[j]} \in \{0,1\} \ (t \le t_s) \\ 0 \le y_{[t]}^{[j]} \le 1 \ (t_s < t) \end{cases} \end{array} \tag{10}$$

where t_s is a threshold; the branch and bound algorithm we use picks $x_{[t]}^{[i]}$ or $y_{[t]}^{[j]}$ only if $t \le t_s$ to split the problem. This means that it is not necessary to decide on-off statuses of the equipment in the future further than t_s. Using the solution of (10) instead of (5) the result can be more cost-effective since the solution is less affected by the erroneous future prediction.

Another good point of (10) is that the calculation time becomes shorter. If the electricity price changes in real-time, to respond to the change as soon as possible is vital to achieve a cost-effective operation. The parameter t_s is adjusted considering the cost efficiency of the operation and the calculation time.

IV. SIMULATIONS

We developed a MILP solver in C using the method mentioned above to conduct simulations based on real data from an office building in Japan. The data were sampled every hour in July, August and September 2010.

We developed a demand predictor also, using an ordinary linear regression model. We used 2008 and 2009 data to make the model. The model uses past outside enthalpy, past residual and future workday/holiday flag as the input variables. The model can predict the demand for 1-24 hours in advance.

The actual demand $d_{[t]}$ was obtained from the building's energy management system. Table I shows the values of the parameters used in the simulations.

We used a laptop PC described below to conduct the simulations: Lenovo ThinkPad T420 with Intel Core i7 (2.70GHz), Microsoft Windows 7 Professional 64 bit and Microsoft Visual C++ 2010 as a compiler.

A. Fixed Prices with Demand Prediction

First we conducted simulations where the electricity prices were fixed and known. Table II shows the prices based on a real contract between the building's owner and an electricity supplier.

TABLE I
PARAMETERS FOR SIMULATIONS

Symbols	Values
n_s, n_d	2
T	24 (hour)
$l_u^{[1]}$, $l_u^{[2]}$	0.13 (GJ)
$h_u^{[1]}$, $h_u^{[2]}$	1.3 (GJ)
$l_v^{[1]}$, $l_v^{[2]}$	0.098 (GJ)
$h_v^{[1]}$, $h_v^{[2]}$	0.98 (GJ)
$l_z^{[1]}$, $l_z^{[2]}$	0.0 (GJ)
$h_z^{[1]}$, $h_z^{[2]}$	8.6 (GJ)
$c_x^{[1]}$, $c_x^{[2]}$, $c_y^{[1]}$, $c_y^{[2]}$	3.0
$\epsilon^{[1]}$, $\epsilon^{[2]}$,	0.01

TABLE II
ELECTRICITY PRICES

Time	Prices (JPY/kWh)
Peak (13:00-17:00)	12.7
Day (8:00-13:00, 17:00-22:00)	10.5
Night (22:00-8:00)	9.3

The owner had an incentive to shift electricity demand from peak time to day time or night time considering the prices.

We executed demand prediction and optimization for 24 hours in the future every hour for the three months. We changed t_s from 1 to 24 and calculated the total electricity costs. The average calculation time was also measured.

B. Real-Time Pricing

Secondly we conducted another simulation under the real-time pricing rate structure. We made simulated prices adding noise with normal distribution, $\sigma \in \mathcal{N}(0, 1^2)$ JPY, to the prices shown in Table II. The simulated prices in the future are unknown when the optimization was executed. The original prices are the unbiased estimators; therefore we used the prices as $p_{[t]}$. The parameter t_s was set to 1 because the future prices are unknown. The total cost was compared to the cost of the conventional operation which would be executed in practice.

We used the same demand predictor as in the previous simulations.

V. RESULTS

A. Fixed Prices with Demand Prediction

The demand predictor's error is shown in Fig. 2. The root mean squared error (RMSE) of the prediction becomes larger in the further future. This suggests that a precise operation scheduling for the whole period is useless when the demand predictor is used.

Fig. 3 shows the total electricity costs as t_s is changed from 1 to 24. As expected from Fig. 2, the total cost is small when t_s is small.

Fig. 4 is a semi-log plot of the average calculation time for each t_s. The calculation time increases exponentially as t_s becomes larger. The shortest calculation time was 0.009 sec when $t_s = 1$. Even the longest calculation time 2.896 sec when $t_s = 24$ was short enough since the simulations were based on one hour data. However, if the scale of the problem becomes larger and the price changes more frequently, the proposed method will be more useful.

And then, we compared the cost when $t_s = 1$ to the conventional operation in Fig. 5.

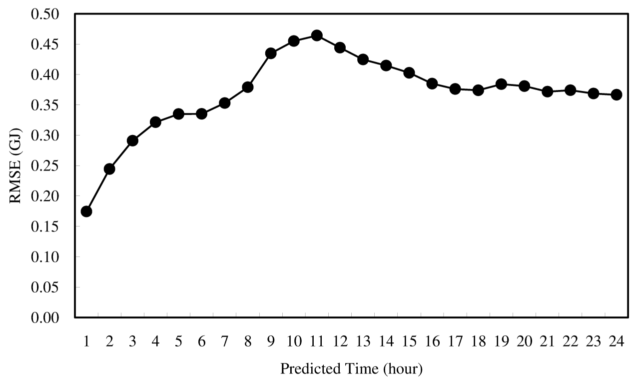

Fig. 2. Demand prediction error

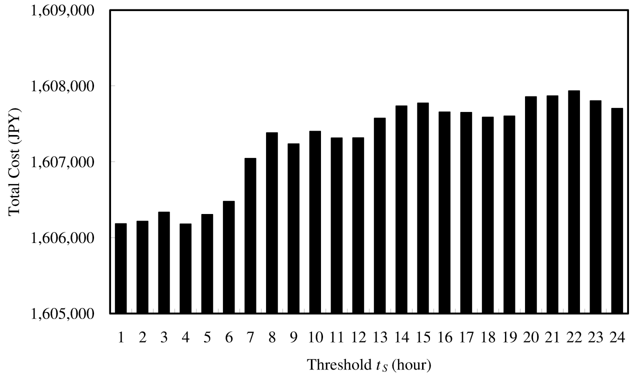

Fig. 3. Total electricity costs as the threshold t_s increases

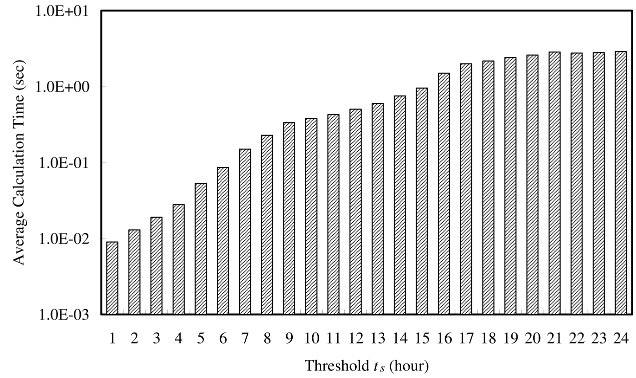

Fig. 4. Average calculation time as the threshold t_s increases

Fig. 5 shows that our method can reduce the total cost by 4.3%.

B. Real-Time Pricing

Fig. 6 shows the total costs of the simulation with real-time pricing when $t_s = 1$ and the conventional operation.

The result shows that our method can reduce the total cost by 7.6% if the real-time pricing rate structure is introduced. The reduction was more significant in comparison to the previous results when the electricity prices were fixed.

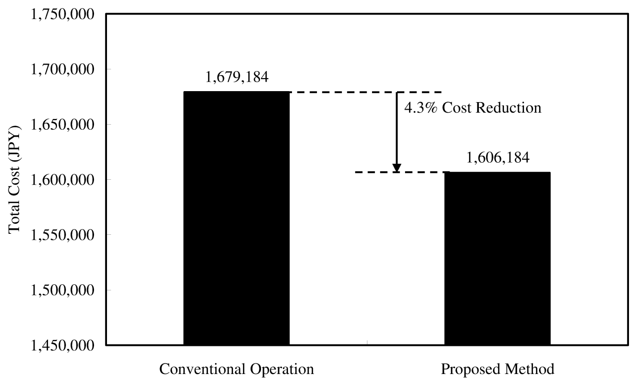

Fig. 5. Total electricity cost comparison

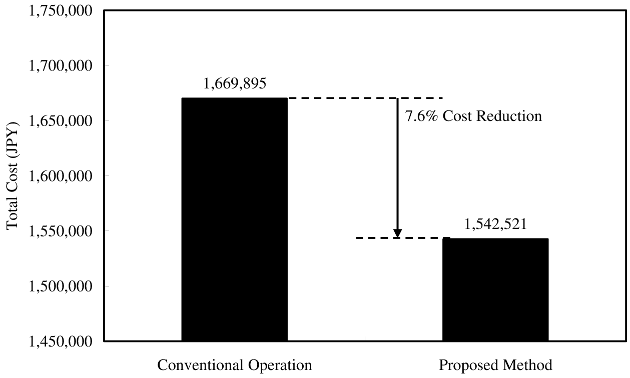

Fig. 6. Total electricity cost comparison under real-time pricing rate structure

VI. CONCLUSIONS

To optimize the operation of a TES system, the proposed method takes advantage of linear relaxation of the integer variables which represents on-off statuses of the equipment in the future. The method is useful when the demand prediction is erroneous and the real-time pricing rate structure is introduced.

The simulations based on the real building data with the real electricity prices showed that the method could reduce electricity cost by 4.3%. It was also shown that if the real-time pricing rate structure is introduced the method can reduce electricity cost by 7.6%.

The simulations also showed that the calculation time of the proposed method was short enough even if the real-time pricing rate structure is introduced in practice.

REFERENCES

[1] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, 13:1-27, Springer, 2012.

[2] Z. Zhang, H. Li, W. D. Turner, and S. Deng, "Optimal Operation of a Chilled-Water Storage System with a Real-Time Pricing Rate Structure," *Trans. of American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE)*, Vol. 117, Part 1, 2011.

[3] T. Imanishi, J. Nishiguchi, T. Konda, R. Dazai, and C. Kaseda, "Building Energy Savings via SaaS/ASP utilizing Data Modeling," *Proc. of The First International Conf. on Universal Village*, Beijing, 2013.

[4] J. Nishiguchi, A. Kurosaki, C. Kaseda, S. Kitayama, M. Arakawa, H. Nakayama, and Y. Yun, "Robust Optimal Operation for Building HVAC Systems with Uncertainty in Demand Prediction," *Proc. of the 57th Annual Conf. of the Institute of Systems, Control and Information Eng. (ISCIE)*, Kobe, May 15-17, 2013.

[5] M. Yamamoto, Y. Nakamura, K. Kuriyama, and T. Matsuyama, "Optimum Control of Heating and Cooling Plant with Nonlinear Operation Characteristics," *Trans. of the Society of Heating, Air-Conditioning and Sanitary Engineers of Japan*, No. 122, May 2007.

[6] J. Mattingley, Y. Wang, and S. Boyd "Receding Horizon Control: Automatic Generation of High-Speed Solvers," *IEEE Control Systems Magazine*, 31(3):52-65, June 2011.

[7] V. Balakrishnan, S. Boyd, and S. Balemi, "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," *Int. J. of Robust and Nonlinear Control*, 1(4):295-317, October-December 1991.

[8] H. Asano, M. Takahashi and N. Yamaguchi, "Market Potential and Development of Automated Demand Response System," *IEEE Power and Energy Society General Meeting*, San Diego, CA, 2011.

[9] Z. Li, Z. Huo and H. Yin, "Optimization and Analysis of Operation Strategies for Combined Cooling, Heating and Power System," *Power and Energy Engineering Conference (APPEEC)*, 2011 Asia-Pacific, March 2011.

[10] L. Xie, Y. Gu, A. Eskandari, and M. Ehsani, "Fast MPC-Based Coordination of Wind Power and Battery Energy Storage Systems," *J. Energy Eng.*, 138(2), 43-53, 2012.