Title
Comparison of Control Strategies for Energy Efficient Building HVAC Systems

Authors
Maasoumy, Mehdi
Sangiovanni-Vincentelli, Alberto

Publication Date
2014-04-13

Peer reviewed

Comparison of Control Strategies for Energy Efficient Building HVAC Systems

Mehdi Maasoumy
Department of Mechanical Engineering
University of California
Berkeley CA, 94720, USA
mehdi@me.berkeley.edu

Alberto Sangiovanni Vincentelli
Department of Electrical Engineering and Computer Science
University of California
Berkeley CA, 94720, USA
alberto@eecs.berkeley.edu

Keywords: Model predictive control, energy-efficient buildings, embedded platform.

Abstract
A framework for the design and simulation of a building envelope and an HVAC system is used to compare advanced control algorithms in terms of energy efficiency, thermal comfort, and computational complexity. Building models are first captured in Modelica [1] to leverage Modelica's rich building component library and then imported into Simulink [15] to exploit Simulink's strong control design environment. Four controllers with different computational complexity are considered and compared: a proportional (P) controller with time varying temperature bounds, a tracking linear quadratic regulator (LQR) controller with time varying tuning parameters, a tracking disturbance-aware linear quadratic regulator (d-LQR) controller with time varying tuning parameters which incorporates predictive disturbance information and a model predictive controller (MPC). We assess the performance of these controllers using two defined criteria, i.e. *energy* and *discomfort* measured with appropriate metrics. We show that the d-LQR and MPC, when compared to the P controller, manage to reduce energy by 41.2% and 46% respectively, and discomfort from 3.8 to 0. While d-LQR and MPC have similar performance with respect to energy and discomfort, simulation time in the case of d-LQR is significantly less than the one of MPC.

1. INTRODUCTION
We present an approach to optimize the energy efficiency of HVAC systems by designing smart controllers with full integration of several factors such as system dynamics, weather predictions, and occupancy schedules, with attention paid to the needed computational power of the embedded platform.

We leverage a number of previous results: A physics-based building model is proposed in [7, 9–12, 16]. A hierarchical control architecture is utilized and simulation results are compared with the ones obtained with a flat architecture. [18] lays out a simple model predictive control (MPC) formulation for building temperature control and presents the advantages and disadvantages of model predictive control when applied to building air temperature regulation. [3] uses a reduced-order model of the airflow in buildings and derive an optimal control law in closed form for rejecting a known disturbance while minimizing a quadratic cost.

In this paper, we build upon [13, 19], where we presented a building automation and control system, and the co-design of control algorithm and embedded platform with focus on sensing system accuracy. In this paper, we extend the co-design framework to include computational complexity of the control algorithm. Hence, we analyze the computational complexity of four control algorithms along with their performance in terms of energy and comfort metrics. We compare the performance of the following controllers: a proportional controller, a linear quadratic regulator (LQR), a disturbance-aware linear quadratic regulator (d-LQR), all suitable for platforms with limited computational power, and MPC, a computationally demanding control algorithm, which requires online computation of control policy through solving optimization problems [11, 12].

Our flow is as follows:

1) Modeling: We first model the building HVAC system and envelope in Modelica [1][1]. Building models are usually highly nonlinear. To automatically derive the mathematical model of the building for control purposes, we use Modelica's features to obtain the linearized systems about the operating point of the system. This model is then imported into Simulink that is particularly suited for simulation and control design.

2) Control Design: We design the control strategy using MATLAB and implement it using MATLAB/Simulink. The plant model (originally in Modelica) and the control algorithm (in MATLAB) are co-simulated using the MATLAB (Simulink) simulation environment. We adapted four controllers to the HVAC problem: a simple P controller, an LQR, a modified d-LQR, and an MPC. In particular, we modified the tracking LQR presented in [10] by using time-varying tuning parameters (matrices) reflecting different temperature bounds at

[1] The building library developed by LBNL [17] can also be used in our framework for a more detailed modeling of the system.

different times of the day. We also modified the LQR controller, called d-LQR, by using the a-priori knowledge of the disturbance data. The performance of the four controllers is compared and contrasted.

The paper is organized as follows. Section 2. introduces a mathematical model for building. Section 3. describes the four different control algorithms that have been used. It presents the derivation of the closed-form solution to the tracking LQR and d-LQR problems and lays out the formulation for MPC. Finally, Section 4. shows results obtained from simulations and discusses the performance and computational characteristics of the four controllers. Conclusions are drawn in Section 5..

2. MATHEMATICAL MODELING

In this paper, we use the model that was proposed in [10, 14] in which the building is considered as a network. There are two types of nodes in the network: walls and rooms. There are in total n nodes, m of which represent rooms and the remaining $n-m$ nodes represent walls. Temperature dynamics of the i-th wall is governed by the following equation:

$$C_{w_i}\frac{dT_{w_i}}{dt} = \sum_{j\in\mathcal{N}_{w_i}} \frac{T_j - T_{w_i}}{R'_{ij}} + r_i\alpha_i A_i q''_{rad_i} \tag{1}$$

Where T_{w_i}, C_{w_i}, α_i and A_i are the temperature, heat capacity, absorption coefficient and area of wall i, respectively. R'_{ij} is the total resistance between wall i and node j. q''_{rad_i} is the radiative heat flux density on wall i. $\mathcal{N}_{w_i}$ is the set of all of neighboring nodes to node w_i and, r_i is equal to 0 for internal walls, and to 1 for peripheral walls.

The temperature of the i-th room is governed by the following equation:

$$C_{r_i}\frac{dT_{r_i}}{dt} = \sum_{j\in\mathcal{N}_{r_i}} \frac{T_j - T_{r_i}}{R'_{ij}} + \dot{m}_{r_i} c_p (T_{s_i} - T_{r_i}) + w_i \tau_{w_i} A_{w_i} q''_{rad_i} + \dot{q}_{int} \tag{2}$$

Where T_{r_i}, C_{r_i} and $\dot{m}_{r_i}$ are the temperature, heat capacity and air mass flow into room i, respectively. c_a is the specific heat capacity of air, A_{w_i} is the total area of window on walls surrounding room i, τ_{w_i} is the transmissivity of glass of window i, q''_{rad_i} is the radiative heat flux density radiated to node i and $\dot{q}_{int_1}$ is the internal heat generation in thermal zone i. $\mathcal{N}_{r_i}$ is the set of all of the neighboring nodes to room i and, w_i is equal to 0 if none of the walls surrounding room i has window, and is equal to 1 if at least one of them has.

A detailed model of building envelope and HVAC system is captured in Modelica with the air mass flow into each thermal zone as inputs, and the temperature of each thermal zone and temperature of walls as outputs. This plant is imported into Simulink using the Dymola[2]-Simulink interface.

[2]Dymola is a commercial simulation environment for Modelica

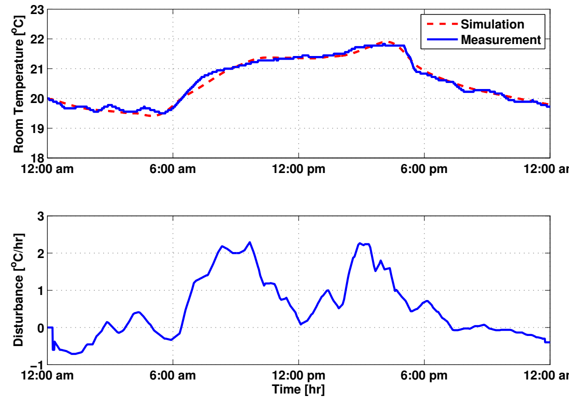

Figure 1. Validation of the proposed model against historical data. We concurrently estimate the model parameters and the unmodeled dynamics using parameterization of the unmodeled dynamics based on the measured data. Details can be found in [12].

The entire plant is imported into Simulink by using first the C-generation capability of Dymola and then encapsulating the code so generated into a DymolaBlock. The DymolaBlock is a wrapper around an S-function MEX block.

We capture the model obtained from (1), and (2) for a three-room model in Modelica. The model has been validated against historical data. The validation result is shown in Fig. 1.

3. CONTROLLER DESIGN

We implemented *a P controller, a tracking LQR, a tracking LQR with disturbance knowledge (d-LQR), and a model predictive controller (MPC).*

In general, a *P* controller is the least computationally complex, as the control input in this case is proportional to a temperature offset. LQR is more computationally complex than *P*, as it involves matrix multiplication and inversion. However, a closed form solution can be derived for the optimal control policy. *P* easily handles a large number of states. LQR with disturbance knowledge (d-LQR) makes use of the available predictive knowledge to enhance the control input to reject disturbances in a smarter fashion than LQR. The computational complexity of this controller is similar to that of LQR. MPC is the most computationally complex algorithm but it yields a solution that always meets the state and input constraints optimally as defined by the cost function.

We use system dynamics that is linearized about the equilibrium point of the system. Note that since the range of thermal zone temperature is small (usually 20 - 22 ° C). The four controllers adapted for the building problem are described as follows:

3.1. P Control

The proportional controller used in this paper is given by the following equations:

$$u_k = \begin{cases} K_p[\overline{T}_k - T(k)] & \text{if } T(k) > \overline{T}_k \\ 0 & \text{if } \underline{T}_k < T(K) < \overline{T}_k \\ K_p[\underline{T}_k - T(k)] & \text{if } T(k) < \underline{T}_k \end{cases} \tag{3}$$

Where K_p is the proportional gain of the controller which should be chosen for best performance. Note that we only consider this controller as a basis against which to compare the performance of other controllers.

3.2. Tracking LQR

We implemented a tracking LQR controller with time-varying tuning matrices on the plant model. Consider the plant model given by

$$x_{k+1} = Ax_k + Bu_k \tag{4}$$
$$y_k = Cx_k \tag{5}$$

Where the state vector $x_k \in \mathbb{R}^n$ at time $t = k$ contains the temperatures of all the nodes in the thermal network of the building: $x_k = [T_{w_1}(k) \ \cdots \ T_{w_p}(k) \ T_{r_1}(k) \ \cdots \ T_{r_m}(k)]$. Where p is the number of walls, m is the number of rooms, and $p+m = n$. The vector $u_k \in \mathbb{R}^m$ is the input at time k, $u_k = [\dot{m}_1(k) \quad \cdots \quad \dot{m}_m(k)]$. Vector y^d_k is the desired output trajectory (i.e. desired temperature for each room), specified for all $k = 1,2,\cdots,N$ and $U_k := [u_k \quad u_{k+1} \quad \cdots \quad u_{N-1}]$. The LQR tracking problem is formulated as follows:

$$\begin{aligned} \min_{U_0} \ & \frac{1}{2}(y^d_N - y_N)^T \mathbf{Q_N}(y^d_N - y_N) \\ & + \frac{1}{2}\sum_{k=0}^{N-1}\Big((y^d_k - y_k)^T \mathbf{Q_k}(y^d_k - y_k) + u_k^T \mathbf{R_k} u_k\Big) \end{aligned} \tag{6}$$

The solution to the LQR problem is the optimal control input of a linear system according to a quadratic cost function of the states and the inputs, hence the name LQR. The offset from the desired trajectory and the inputs are penalized with weight matrices called $\mathbf{Q_k}$ and $\mathbf{R_k}$ at each time k, respectively.

$$\mathbf{Q_k} = \mathbf{diag}\ [q^1_k,\ q^2_k,\ \ldots,\ q^m_k] \tag{7}$$
$$\mathbf{R_k} = \mathbf{diag}\ [r^1_k,\ r^2_k,\ \ldots,\ r^m_k] \tag{8}$$

Where the superscript i refers to the room i in the building and the subscripts k implies the k^{th} time step.

Assumption 1: We assume that the matrices $\mathbf{Q_k}$ are positive semidefinite and symmetric and the matrices $\mathbf{R_k}$ are positive definite and symmetric. This translates to $q^i_j \geq 0 \ \ \forall i, j$ and $r^i_j > 0 \ \ \forall i, j$. Note that the symmetry assumption is fulfilled by the diagonal structure of these matrices.

The controller can be tuned by varying the weight matrices according to the occupancy schedules. Using *Bellman's principle of optimality*, a recursive relation can be obtained. The resulting optimization problem can be solved by *dynamic programming* backwards in time to determine the optimal control law. In [14] the optimal control law was shown to be as follows:

$$u^o_k = F_k - K_k x_k \tag{9}$$
$$K_k = [\mathbf{R_k} + B^T P_{k+1} B]^{-1} B^T P_{k+1} A \tag{10}$$
$$F_k = -[\mathbf{R_k} + B^T P_{k+1} B]^{-1} B^T b_{k+1} \tag{11}$$

Where P_k and b_k can be calculated backwards in time using

$$\begin{aligned} P_{k-1} =& A^T P_k A - A^T P_k B[\mathbf{R_{k-1}} + B^T P_k B]^{-1} B^T P_k A \\ & + C^T \mathbf{Q_{k-1}} C \end{aligned} \tag{12}$$

$$\begin{aligned} b_{k-1} =& -A^T P_k B[\mathbf{R_{k-1}} + B^T P_k B]^{-1} B^T b_k - C^T \mathbf{Q_{k-1}} y^d_{k-1} \\ & + A^T b_k \end{aligned} \tag{13}$$

with the terminal conditions being $P_N = C^T \mathbf{Q_N} C$ and $b_N = -C^T \mathbf{Q_N} y^d_N$. Note that K_k can be regarded as the feedback gain and F_k as the feed-forward gain [4].

3.3. Tracking d-LQR

The difference of LQR and d-LQR is that d-LQR integrates the predictive disturbance knowledge to enhance the performance of the LQR. The classic LQR problem solution can be found in the literature [5]. The solution to the non-homogeneous discrete time d-LQR was developed recently [3] using Lagrange multipliers for the state dynamics as constraints and then solving the problem using the Karush-Kuhn-Tucker (KKT) optimality conditions. Here we derive the same solution using a different method, namely *dynamic programming*. Assume the plant model given by (14) where d_k is the disturbance to the system at time $t = k$.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + E\mathbf{d_k} \\ y_k &= Cx_k \end{aligned} \tag{14}$$

We consider the same cost function (6). We use dynamic programming to solve for the solution of this problem. The optimal control is given by

$$u^o_k = F_k - K_k x_k \tag{15}$$
$$K_k = [\mathbf{R_k} + B^T P_{k+1} B]^{-1} B^T P_{k+1} A \tag{16}$$
$$F_k = -[\mathbf{R_k} + B^T P_{k+1} B]^{-1} B^T (b_{k+1} + P_{k+1} E\mathbf{d_k}) \tag{17}$$

Where P_k and b_k can be calculated backwards in time using

$$\begin{aligned} P_{k-1} =& A^T P_k A - A^T P_k B[\mathbf{R_{k-1}} + B^T P_k B]^{-1} B^T P_k A \\ & + C^T \mathbf{Q_{k-1}} C \end{aligned} \tag{18}$$

$$\begin{aligned} b_{k-1} =& -A^T P_k B[\mathbf{R_{k-1}} + B^T P_k B]^{-1} B^T (b_k + P_k E\mathbf{d_{k-1}}) \\ & - C^T \mathbf{Q_{k-1}} y^d_{k-1} + A^T (b_k + P_k E\mathbf{d_{k-1}}) \end{aligned} \tag{19}$$

with the terminal conditions being $P_N = C^T \mathbf{Q_N} C$ and $b_N = -C^T \mathbf{Q_N} y^d_N$. Note the appearance of disturbances in the update equation for b_{k-1}. The solution to the problem developed here agrees with the solution developed by [3] using a different method.

Note that the boundedness of the Riccati equation solution of the homogeneous LQR problem, and the asymptotic stability of the resulting closed-loop system is guaranteed [6].

3.4. Model Predictive Control

A model predictive control problem is formulated with the objective of minimizing a cost function which is a linear combination of the total cooling and heating power consumption and the peak of air flow and temperature-bound violation at each time subject to system dynamics and constraints. The predictive controller solves at each time step t the following problem

MPC Algorithm:

$$\min_{U_t,\bar{\varepsilon},\underline{\varepsilon}} \quad \{||\mathbf{U_t}||_1 + c_1||\mathbf{U_t}||_\infty + c_2(||\bar{\varepsilon}_\mathbf{t}||_1 + ||\underline{\varepsilon}_\mathbf{t}||_1)\}$$

subject to:

$$x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k|t} + Ed_{t+k|t} \tag{20a}$$

$$y_{t+k|t} = Cx_{t+k|t} \tag{20b}$$

$$\underline{T}_{t+k|t} - \underline{\varepsilon}_{t+k|t} \le y_{t+k|t} \le \overline{T}_{t+k|t} + \bar{\varepsilon}_{t+k|t} \tag{20c}$$

$$u_{t+k|t} \le \bar{u} \tag{20d}$$

$$\underline{\varepsilon}_{t+k|t}, \;\; \bar{\varepsilon}_{t+k|t} \ge 0 \tag{20e}$$

where $\mathbf{U_t} = [u_{t|t}, u_{t+1|t}, \cdots, u_{t+N-1|t}]$ is the vector of control inputs, $\underline{\varepsilon}_\mathbf{t} = [\underline{\varepsilon}_{t+1|t}, \cdots, \underline{\varepsilon}_{t+N|t}]$ is the temperature violations from the lower bound, $\bar{\varepsilon}_\mathbf{t}$ the temperature violation from the upper bound, $y_{t+k|t}$ is the thermal zone temperatures, $d_{t+k|t}$ is the load prediction, and $\underline{T}_{.|t}$ and $\overline{T}_{.|t}$ are the lower and upper bounds on the zone temperature, respectively. $\bar{u}$ is the upper bound on the input air flow. c_2 is the penalty on the comfort constraint violations, and c_1 is the penalty on peak power consumption. Note that constraints (20a), and (20d) should hold for $k = 0, 1, ..., N-1$ and constraints (20b), (20c), and (20e) should hold for $k = 1, 2, ..., N$.

At each time step only the first entry of $\mathbf{U_t}$ is implemented on the plant. At the next time step the prediction horizon N is shifted leading to a new optimization problem. This process is repeated over and over until the total time span of interest is covered. The prediction horizon is $N = 24$. We used YALMIP [8] to formulate the MPC problem in MATLAB, and used IBM CPLEX [2] to solve the resulting optimization problem.

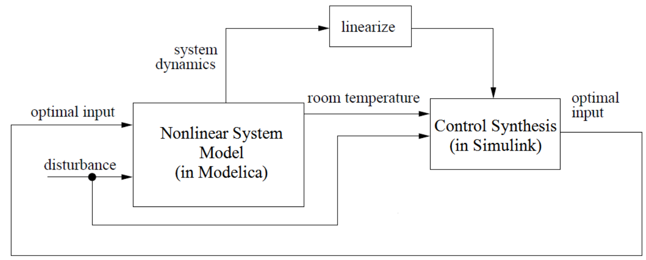

Figure 2. Schematic of the closed loop system including the nonlinear system model in *Modelica* and the control implementation in *Simulink*.

4. SIMULATION RESULTS

4.1. Simulating Heterogeneous Models

To verify the performance of the controllers, we simulated the combination of controller and plant (Fig. 2). We argued that modeling the plant in Modelica has several advantages while the design and implementation of the controller is best done in Simulink. There are two strategies to simulate the composed system:

- *Hosted Simulation* where either the Modelica model is imported into the Simulink input language and simulation occurs in Simulink or the Modelica model is imported into the Simulink environment and simulation occurs in Modelica.

- *Co-simulation.* where the two components are evaluated during simulation each in its own simulation environment (for Modelica, the simulation environment we chose is Dymola). In this case, a "master" is needed that orchestrates the actions of the two simulators while simulation progresses. Commercial co-simulation platforms have been developed to act as the "master", for example, EXITE, Silver and TISC [1]. EXITE, for instance, provides interfaces to Dymola, Simulink, ASCET, Rhapsody, ARTISAN Studio and C/C++. During co-simulation all models stay in their simulation environment and EXITE implements the communication among them via dedicated communication blocks.

Simulink has a feature (the S-Function C-Mex mechanism) that can be used to deal with models described in Dymola if the models are expressed in terms of input and output signals (Modelica has also the capability of capturing models in equation form where inputs and outputs are not explicitly identified, a feature that is not available in Simulink). The models can be imported into Simulink as an S-Function C-Mex file.

There are two mechanisms that can be used to perform simulation in the Dymola environment:

- *In-line Integration*. The simulation engine is Simulink. In this option, the user has to select a particular integra-

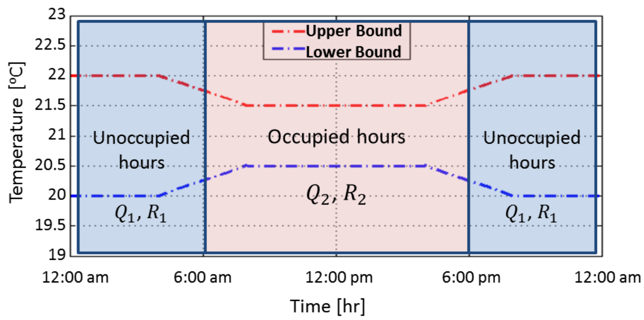

Figure 3. Temperature bounds for occupied and unoccupied hours.

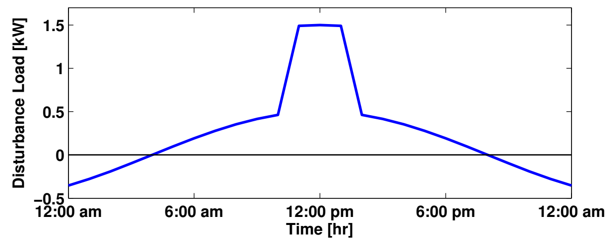

Figure 4. Aggregate effect of disturbance from outside weather and internal loads.

tion method that is used by Dymola (e.g., *explicit/implicit Euler, trapezoidal method, explicit/implicit Runge Kutta*) to generate "C"-code that is then managed by the Simulink simulation engine.

- *In-line integration method not used*. The generated "C" code includes variable declarations and a call to the Dymola environment to evaluate the model. In this case, Simulink acts as the master.

For some integration methods such as *Explicit Euler*, the output diverges (this is to be expected because of the limited absolute stability region of this method), while no divergence is observed in the case of *No in-line integration* since the integration methods are dictated by the two tools that use robust numerical integration methods. Hence, we use "No in-line integration" for the simulations presented in this paper.

4.2. Comparing Controllers

We compare now the four different controllers described in Section (3.). The comfort zone is defined to be the space between the lower and upper temperature bounds as shown in Fig. (3).

For the disturbance model we assume that the disturbance load from the outside weather to the building is a sinusoidal load which is negative at night and positive during the day. We also assume an additive load due to a high density of occupants (e.g. a meeting) in the considered room from 11 am to 1 pm. The cumulative effect of both disturbances is shown in Fig. (4)

To be consistent and fair to all controllers we used a sampling time of one hour for all cases. In order to be fair for evaluation of the energy savings, we kept running simulations until we experienced a difference of less than 0.2 °C between the room temperature at the start time of one day (12am) and the end time of the same day (12am next day).

4.2.1. P Control

P control logic is given by (3).

4.2.2. Tracking LQR

In this case we have considered the desired temperature, y_k^d to be a constant temperature of 21^oC. We consider time varying tuning parameters $\mathbf{Q_k}$ and $\mathbf{R_k}$ to reflect the occupancy schedule knowledge in the LQR control derivation.

We exploit the following strategy for tuning the weight matrices which reflects the temperature constraints at each time $t = k$:

$$\mathbf{Q_k} = \begin{cases} \mathbf{Q_1} & \text{if } 12am \leq k \leq 4am \\ \mathbf{Q_1} + \frac{(\mathbf{Q_2}-\mathbf{Q_1})*[k-4]}{4} & \text{if } 4am \leq k \leq 8am \\ \mathbf{Q_2} & \text{if } 8am \leq k \leq 4pm \\ \mathbf{Q_2} + \frac{(\mathbf{Q_1}-\mathbf{Q_2})*[k-4]}{4} & \text{if } 4pm \leq k \leq 8pm \\ \mathbf{Q_1} & \text{if } 8pm \leq k \leq 12am \end{cases} \tag{21}$$

Where $\mathbf{Q_1}$ and $\mathbf{Q_2}$ are the weight matrices corresponding to *unoccupied* and *occupied* hours, respectively, as shown in Fig. 3. The same strategy can be defined for $\mathbf{R_k}$ as well. In this case the disturbance knowledge is not used in the control derivation.

4.2.3. Tracking d-LQR

In this case we also assume a constant temperature of 21^oC as the desired temperature and pick the tuning parameters of $\mathbf{Q_k}$ and $\mathbf{R_k}$ based on the same strategy as (21). The difference of this case with case 1 is that the disturbance knowledge is used in the control derivation for a better tracking and disturbance rejection.

4.2.4. MPC

We have considered an MPC with soft constraints on the room temperature. However, to have a fair comparison among different controllers, we choose a large value for c_2 to force the temperature to stay within the upper and lower bounds (i.e comfort zone).

The results are shown in Fig. (5). A quantitative comparison of different controllers is depicted in Fig. (6). The controller parameters and the simulation time for each is provided in Table (1).

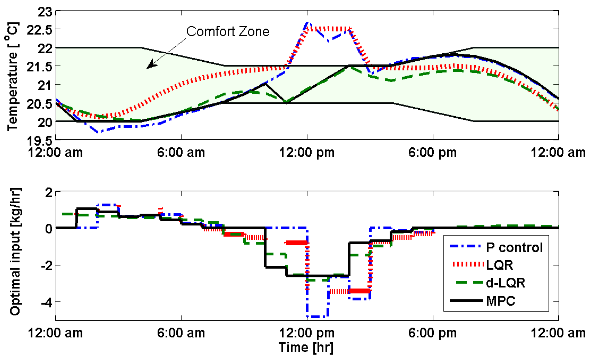

Figure 5. Temperature and optimal input for four different controllers.

4.3. Control Performance Comparison

To compare the overall performance of the proposed controllers we define two metrics to measure the energy consumption and comfort level provided by each controller. The energy metric is defined as:

$$I_e = \int_{t=0}^{24} [P_c(t) + P_h(t) + P_f(t)]\, dt \qquad (22)$$

Where cooling power P_c, heating power P_h and fan power P_f are defined as

$$P_c(t) = \dot{m}_c(t) c_p [T_{out}(t) - T_c(t)] \qquad (23)$$
$$P_h(t) = \dot{m}_h(t) c_p [T_h(t) - T_{out}(t)] \qquad (24)$$
$$P_f(t) = \alpha \dot{m}^3(t) \qquad (25)$$

The discomfort metric is defined as the sum of all the temperature violations during the course of a day.

$$I_d = \int_{t=0}^{24} \left[\min\left(\left|T(t) - \overline{T}(t)\right|, \left|T(t) - \underline{T}(t)\right| \right) . \mathbf{1}_{\mathcal{B}(t)^c}(T(t)) \right] dt$$

Where $\mathcal{B}(t) = [\underline{T}(t), \overline{T}(t)]$ is the allowable temperature boundary at time t and $\mathbf{1}$ is the indicator function.

Remark 1. As shown in the lower plot of Fig (5) the control input for the d-LQR is more similar to the MPC control input rather than to the LQR. On the other hand, the control input of the LQR is comparable to the one from P control.

Remark 2. The d-LQR and MPC use the disturbance knowledge to *pre-cool* the space several hours before the disturbance load hits as opposed to the LQR and P control which only react to the disturbance load instantaneously. The utilization of the disturbance knowledge results in a lower peak air flow demand and full satisfaction of temperature bounds for these two controllers versus the high peak air flow and constraint violation for the LQR and P controllers.

Remark 3. As shown in Table (1) the time required to simulate the MPC operations is three orders of magnitude higher than the one required for the simulation of LQR. This is because the LQR problems have closed-form solutions while the MPC solves optimization problems on the fly. The high computational effort and delay of MPC can be problematic as also indicated in [18], for implementing on embedded platforms with limited computational power. The *d-LQR* controller introduced in this paper can be regarded as a less computationally intensive alternative to MPC for large systems with a high number of states and inputs.

Table 1. Simulation time and Parameters for different controllers.

Controller:	P Ctrl	LQR	d-LQR	MPC
Simulation time $[s]$	1.31	0.13	0.11	115.1
Parameters	$K_p = 4$	$q_1 = 0.01$ $q_2 = 100$ $r_1 = 10$ $r_2 = 0.02$	$q_1 = 0.24$ $q_2 = 0.54$ $r_1 = 1$ $r_2 = 0.09$	$c_1 = 5$ $c_2 = 500$

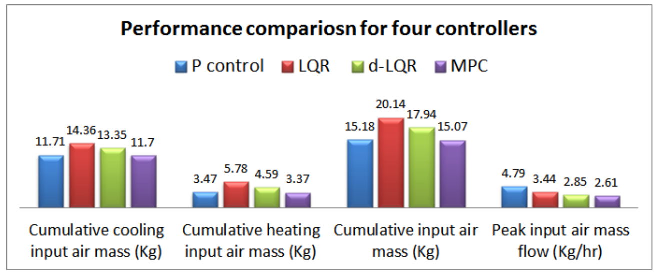

Figure 6. Quantitative comparison of different controllers.

The comparison result is shown in Fig. 7. From this comparison, we note that the LQR is not sensitive to parameter changes, the P controller is the worst controller among the proposed controllers and the performance of d-LQR and MPC are very close.

It was shown that the developed d-LQR controller acts more like an MPC rather than an LQR controller in terms of rejecting disturbances, and results in a smarter controller which uses disturbance knowledge to decrease both the total air mass and the peak air mass flow into the thermal zone.

5. CONCLUSION AND FUTURE WORK

We presented and compared four controllers for the control of an HVAC system in smart buildings. We presented and validated a heterogeneous model comprising a building envelope and an HVAC system model in DymolaBlock.It was shown that the d-LQR controller is able to reject the distur-

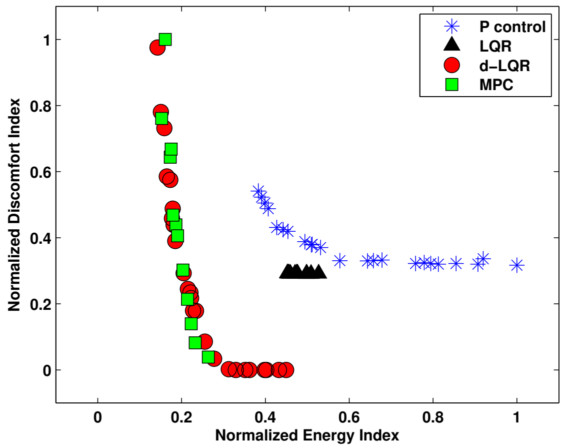

Figure 7. Normalized discomfort level versus normalized energy consumption for different controllers. It is shown that P control exhibits the worst performance; LQR, in the presence of disturbance is not so responsive to parameter changes; and d-LQR and MPC have similar performances. Normalized values along each axis are obtained by dividing the absolute values for each axis, by the maximum experienced value along that axis.

bance by using the knowledge of the disturbance and to keep the temperature within a set of given bounds at all times as opposed to the LQR derived with no a-priori knowledge of the disturbance characteristics. As a result the energy index of the d-LQR was reduced by 41.2% and the discomfort metrics from 3.8 to 0 when compared to the P control. A model predictive controller was also designed and implemented. This controller was shown to reduce the energy index by 46% and the discomfort index from 3.8 to 0 when compared to the P controller. d-LQR and MPC manage to keep the temperature within the temperature bounds at all times as opposed to P control and LQR which fail in doing so.

In future work, we will extend the co-design approach presented in [13] of control algorithm and embedded platform considering the computational complexity of the control algorithms and and the computational capabilities of the embedded platforms.

6. ACKNOWLEDGEMENT

Mehdi Maasoumy is funded by the Republic of Singapore's National Research Foundation through a grant to the Berkeley Education Alliance for Research in Singapore (BEARS) for the Singapore-Berkeley Building Efficiency and Sustainability in the Tropics (SinBerBEST) Program. BEARS has been established by the University of California, Berkeley as a center for intellectual excellence in research and education in Singapore. Alberto Sangiovanni Vincentelli is supported in part by the TerraSwarm Research Center,one of six centers administered by the STARnet phase of the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

REFERENCES

[1] Modelica website: https://www.modelica.org, February 2012.

[2] IBM ILOG CPLEX Optimizer, September 2013.

[3] S. Ahuja, A. Surana, and E. Cliff. Reduced-order models for control of stratified flows in buildings. In *American Control Conference (ACC), 2011*, pages 2083–2088. IEEE, 2011.

[4] B.D.O. Anderson and J.B. Moore. *Linear optimal control*. Prentice-Hall Englewood Cliffs, NJ, 1971.

[5] D.P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 1996.

[6] D.P. Bertsekas. Dynamic programming and optimal control 3rd edition, volume i. 2011.

[7] D. Gyalistras and M. Gwerder. Use of Weather and Occupancy Forecasts for Optimal Building Climate Control (OptiControl): Two years progress report. *Terrestrial Systems Ecology ETH Zurich, Switzerland and Building Technologies Division, Siemens Switzerland Ltd., Zug, Switzerland*, 2010.

[8] J. Lofberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[9] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. In *American Control Conference (ACC), 2010*, pages 5106–5111. IEEE, 2010.

[10] M. Maasoumy, A. Pinto, and A. Sangiovanni-Vincenteli. Model-based hierarchical optimal control design for HVAC systems. In *Dynamic System Control Conference (DSCC), 2011*. ASME, 2011.

[11] M. Maasoumy and A. Sangiovanni-Vincentelli. Optimal control of HVAC systems in the presence of imperfect predictions. In *Dynamic System Control Conference (DSCC)*. ASME, 2012.

[12] M. Maasoumy and A. Sangiovanni-Vincentelli. Total and peak energy consumption minimization of hvac systems using model predictive control. *IEEE Design and Test of Computers*, Jul/Aug 2012.

[13] Mehdi Maasoumy, Qi Zhu, Cheng Li, Forrest Meggers, and Alberto Sangiovanni-Vincentelli. Co-design of control algorithm and embedded platform for HVAC systems. In *The 4th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS'13) (ICCPS 2013)*, Philadelphia, USA, April 2013.

[14] Mehdi Maasoumy Haghighi. Modeling and optimal control algorithm design for hvac systems in energy efficient buildings. Master's thesis, EECS Department, University of California, Berkeley, Feb 2011.

[15] N. Mendes, G.H.C. Oliveira, and H.X. de Araújo. Building thermal performance analysis by using matlab/simulink. In *Seventh International IBPSA Conference, Rio de Janeiro, Brazil*, 2001.

[16] F. Oldewurtel, A. Parisio, C.N. Jones, M. Morari, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and K. Wirth. Energy Efficient Building Climate Control using Stochastic Model Predictive Control and Weather Predictions. In *American Control Conference (ACC)*, pages 5100–5105. IEEE, 2010.

[17] M. Wetter, W. Zuo, and T.S. Nouidui. Modeling of heat transfer in rooms in the modelica buildings library.

[18] A. Dally Y. Ma, A. Kelman and F. Borrelli. Model predictive control of thermal energy storage in building cooling systems. *IEEE Control System Magzine*, pages 1–65, 2011.

[19] Yang Yang, Qi Zhu, M. Maasoumy, and A. Sangiovanni-Vincentelli. Development of building automation and control systems. *Design Test of Computers, IEEE*, 29(4):45 –55, aug. 2012.