2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Green Scheduling of Control Systems for Peak Demand Reduction

Truong X. Nghiem, Madhur Behl, Rahul Mangharam and George J. Pappas

Abstract— **Building systems such as heating, air quality control and refrigeration operate independently of each other and frequently result in temporally correlated energy demand surges. As peak power prices are 200-400 times that of the nominal rate, this uncoordinated activity is both expensive and operationally inefficient. We present an approach to fine-grained coordination of energy demand by scheduling the control systems within a constrained peak while ensuring custom climate environments are facilitated. The peak constraint is minimized for energy efficiency, while we provide feasibility conditions for the constraint to be realizable by a scheduling policy for the control systems. The physical systems are then coordinated by the scheduling controller so as both the peak constraint and the climate/safety constraint are satisfied. We also introduce a simple scheduling approach called *lazy scheduling*. The proposed control and scheduling strategy is implemented in simulation examples from small to large scales, which show that it can achieve significant peak demand reduction while being efficient and scalable.**

I. INTRODUCTION

Building systems such as heating, ventilating, air conditioning and refrigeration (HVAC&R) systems, chiller systems, and lighting systems operate independently of each other and frequently trigger concurrently, resulting in temporally correlated power demand surges. Most commercial buildings are subject to peak demand pricing which can be over 200 times that of the nominal power rate [1] and this uncoordinated behavior results in both expensive and inefficient energy consumption.

While there exist several different approaches to balance power consumption by load shifting and load shedding, they operate on coarse grained time scales and do not help in de-correlating energy sinks. The focus of this paper is on a new approach for fine-grained scheduling of control systems within an aggregate peak power envelop while ensuring the custom climate conditions are maintained within the desired ranges. We achieve this by combining: (a) *minimization* of the feasible peak power constraint of the systems; and (b) *coordination* of the individual systems within a global schedule to satisfy the said constraint.

While traditional real-time scheduling algorithms [2] may be applied to such resource sharing problems, they impose stringent constraints on the task model. Generally, real-time scheduling is restricted to tasks whose worst case execution times are fixed and known a priori. While this simplifies the runtime complexity, for control systems it does not effectively capture the system's behavior whose operation is dependent on the plant dynamics and environmental conditions. A contribution of this paper is the formulation of a framework for scheduling control tasks with an aggregate resource envelope. Although [3] used a similar model, the authors assumed a periodic task activation model so that traditional real-time scheduling algorithms could be applied. This made the system less flexible to changes in system dynamics because task scheduling was not based on state feedback. We provide schedulability conditions and a state-feedback scheduling scheme for a set of control systems. The proposed model is scalable and effective for the large class of systems with affine dynamics.

Recently, a popular approach to energy efficient control for commercial buildings and data centers is model predictive control (MPC) ([4], [5], [6], [7]). MPC is a flexible and well-developed advanced control technique with broad applications in complex systems ([8], [9], [10]). In [4] the authors investigated MPC for thermal energy storage in building cooling systems. Stochastic MPC was used to minimize building's energy consumption in [5], while peak demand reduction by MPC with real-time electricity pricing was considered in [6]. However, the scalability of this approach for large systems might be limited due to its high computational requirement for optimization. Though our approach also utilizes optimization, it is more scalable while still effective in reducing peak demand. This is achieved by decomposing the system into a *feasible peak minimization* and a *scheduling controller*, as presented in the rest of this paper. In [11] optimal On-Off control was considered for air conditioning and refrigeration, in which temperature bounds were statically optimized for a single system. Our approach dynamically coordinates a large number of systems to achieve energy efficiency.

The rest of the paper is structured in the following manner. Section II presents the problem considered in this paper and our approach to solve it. Section III formulates the task model that abstracts dynamic systems, whose schedulability analysis is provided in Section IV. We present in detail the proposed approach in Sections V and VI, and two simulation examples of different scales in Section VII. Section VIII concludes the paper with a road map of our future work.

II. PROBLEM FORMULATION

This paper presents a control approach for reducing peak power demand of the heating system of multiple zones. Consider $n > 1$ zones. Each zone is heated by a heater, whose power can be controlled to vary its heat input rate to the zone. However, its power cannot be changed continuously

This research was supported by the Greater Philadelphia Innovation Cluster (GPIC) for Energy Efficient Buildings sponsored by the U.S. Department of Energy. The authors are with the Department of Electrical and Systems Engineering, the University of Pennsylvania, Philadelphia, PA 19104, USA. E-mail: {nghiem, mbehl, rahulm, pappasg}@seas.upenn.edu

but on a discrete scale, e.g., 50%, 75%, and 100% of full power. A heater can also be turned off when it consumes no energy and provides no heat input. The aggregate power demand is the sum of the powers of individual heaters. Similarly, the aggregate energy consumption is calculated by summing all individual energy consumptions. In a demand-based tariff for commercial energy customers, the utility bill is calculated by $\mathrm{Bill} = p_u \times \mathrm{Tot} + p_d \times \mathrm{Peak}$ where Peak is the peak aggregate demand, Tot the total energy consumption, p_u the usage price, and p_d the demand price. Here, p_d is much higher than p_u, e.g., by approximately 240 times in Pennsylvania, USA [1]. The high penalty for peaks in power consumption means that reducing the peak demand not only saves energy but also makes a system cost effective.

A. System modeling

Let us denote the heat input rate of the heater in zone i, $1 \leq i \leq n$, by $Q_i \geq 0$ (kW), which can only receive values from a finite set $\mathcal{Q}_i$ determined by the allowable power levels of the heater. The thermal environment of zone i is comfortable for its occupants if its air temperature, denoted by x_i (°C), stays between a lower threshold l_i and an upper threshold $h_i > l_i$. Let T_a be the ambient air temperature (°C). For simplicity, we assume that there are no thermal interactions between zones, there are no disturbances, and the ambient air temperature is the same for all zones. The law of conservation of energy gives us the following simplified heat balance equation for zone i:

$$C_i \frac{\mathrm{d}x_i}{\mathrm{d}t} = K_i \left(T_a - x_i\right) + Q_i \tag{1}$$

where C_i is the thermal capacity of the zone (kJ/K) and K_i the thermal conductance between the zone and the ambient air (kW/K). It follows that:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -\frac{K_i}{C_i} x_i + \left(\frac{K_i}{C_i} T_a + \frac{Q_i}{C_i}\right) = -a_i x_i + b_i \tag{2}$$

in which

$$a_i = \frac{K_i}{C_i}, \quad b_i = \frac{K_i}{C_i} T_a + \frac{Q_i}{C_i}. \tag{3}$$

Let $P_i(Q_i)$ be the fixed power demand of heater i corresponding to heat input rate Q_i, where $P_i(Q_i) \geq Q_i$ due to the energy efficiency of the heater. It is assumed that we do not have closed-form expressions of the mappings P_i. When heater i is turned off, $Q_i = 0$ and $P_i(Q_i) = 0$.

In a billing cycle $\mathcal{B} > 0$, usually $1\,$month, the total energy consumption is $\mathrm{Tot} = \int_0^{\mathcal{B}} \sum_{i=1}^{n} P_i(Q_i(t))\,\mathrm{d}t$ and the peak demand is $\mathrm{Peak} = \max_{0 \leq t \leq \mathcal{B}} \sum_{i=1}^{n} P_i(Q_i(t))$. The goal of an energy efficient control policy for these zones is to reduce either Tot or Peak or Bill while ensuring that zone temperatures are always within corresponding thresholds. In this paper, we consider the control problem for peak demand reduction, as stated below.

Problem 1: *Compute control input $Q_i(t) \in \mathcal{Q}_i$ for each heater i, $1 \leq i \leq n$, at time t, $0 \leq t \leq \mathcal{B}$, so as to minimize the peak demand* Peak *while maintaining thermal comfort in each zone, i.e., $x_i(t) \in [l_i, h_i]$ at all time.*

In practice, each heater is commonly controlled by a thermostat, usually with a two-position rule (i.e., ON/OFF rule). The simplest strategy to reduce peak demand is to set each heater to its lowest power. However, the thermostats are uncoordinated, which often leads to high peak power demand. For instance, it is possible that at one time, all heaters are turned on, causing a spike in power demand.

B. Scheduling control for peak demand reduction

Problem 1 can be formulated as an optimization. First, the system models are discretized with a period $\Delta_T > 0$ to obtain difference equations $x_i(t+1) = A_i x_i(t) + B_i Q_i(t) + D_i T_a(t)$ for each $i = 1, \ldots, n$. The initial air temperatures of the zones are $x_{0,i} \in [l_i, h_i]$. Then the optimization program for minimizing peak demand during a finite horizon $N > 0$, where $N\Delta_T = \mathcal{B}$, is formulated as:

$$\begin{aligned} \text{minimize} \quad & \max_{0 \leq t \leq N-1} \textstyle\sum_{i=1}^{n} P_i(Q_i(t)) \\ \text{subject to} \quad & x_i(t+1) = A_i x_i(t) + B_i Q_i(t) + D_i T_a(t) \\ & x_i(0) = x_{0,i} \\ & x_i(t) \in [l_i, h_i]\,, \; Q_i(t) \in \mathcal{Q}_i \end{aligned} \tag{4}$$

in which the constraints are satisfied for all $0 \leq t \leq N-1$ and all $1 \leq i \leq n$. The variables are the control inputs $Q_i(0), \ldots, Q_i(N-1)$ for each i (that is $n \times N$ variables). Since N is usually very large[1], a popular approach to this problem is *model predictive control* (MPC) with horizon $H \ll N$ ([8], [4], [5], [6]). One drawback of MPC is its computational requirement for solving optimization at each time step. Since (4) is a combinatorial optimization (recall that $\mathcal{Q}_i$ are finite sets), this requirement is generally very high, especially for large systems. This paper presents an alternative approach which is efficient and more scalable.

In our approach, each zone and its heater are abstracted as a control task and the n zones as a set of n tasks. As recommended by [12], the power demand is constrained: $\sum_{i=1}^{n} P_i(Q_i(t)) \leq P_{max}$ where P_{max} is the *peak permissible demand*. P_{max} is an optimization variable and is often calculated offline. However, in our approach, *P_{max} is imposed indirectly by reducing Q_i and restricting the number of heaters that can be on simultaneously to k, for $1 \leq k \leq n$, while maintaining the feasibility of this restriction for the tasks*. By that, P_{max} is the sum of the k largest powers of the heaters. Then *the tasks are coordinated, i.e., the heaters are switched on and off, under that constraint while ensuring thermal comfort in all zones*. Specifically, Problem 1 is decomposed into two sub-problems (Fig. 1):

- **Feasible peak minimization:** for an ambient air temperature T_a, we compute Q_i and k to minimize P_{max} while maintaining schedulability of the tasks:

  $$\begin{aligned} \text{minimize} \quad & P_{max}\left(\{P_i(Q_i)\}, k, T_a\right) \\ \text{subject to} \quad & 1 \leq k \leq n,\; Q_i \in \mathcal{Q}_i \,\forall i = 1 \ldots n \\ & \text{tasks are schedulable,} \end{aligned}$$

  where k and Q_i, $i = 1, \ldots, n$, are variables. Since this minimization depends only on T_a, which usually varies

[1] For example, if billing cycle $\mathcal{B} = 30\,$days and $\Delta_T = 15\,$min then $N = 2880$.

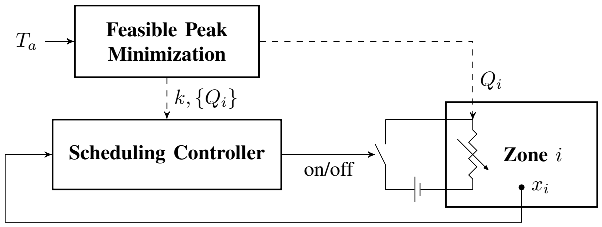

Fig. 1. Control and scheduling structure.

slowly, it only needs to be re-computed at a slow rate (e.g., every hour) during which Q_i and k are fixed.

- **Scheduling control:** we switch on and off the heaters so that the constraint set by the *feasible peak minimization* is satisfied and thermal requirements are maintained, i.e., each zone temperature stays within its given thresholds at all time. The *scheduling controller* is a state-feedback controller (Fig. 1).

The first concern regarding this approach is how much P_{max} can be reduced while ensuring that the tasks are schedulable. This question will be addressed in the feasibility analysis in Section IV, after the formulation of task model in the next section. Section V will discuss the *feasible peak minimization* and Section VI will present the *scheduling controller*.

III. Task Model

In this section, we define a task model to abstract the dynamic systems being coordinated. Each system has a state variable (e.g., room temperature) whose dynamics is governed by the current mode of the system. A task T is a tuple (x, l, h, X_0, M, dyn) in which:

- $x \in \mathbb{R}$ is the continuous state variable;
- $l \in \mathbb{R}$ and $h \in \mathbb{R}$, where $l < h$, are the lower and upper thresholds of the state x, respectively;
- $X_0 \subseteq [l, h]$ is the set of initial states;
- M is a finite set of operation modes;
- $dyn : M \times \mathbb{R} \to \mathbb{R}$ specifies the dynamics in each mode.

A schedule for task T is a function $m : \mathbb{R}_{\geq 0} \to M$ which specifies the operation mode $m(t) \in M$ of T at any time $t \geq 0$. Given a schedule m, the task's state trajectory is a function $x : \mathbb{R}_{\geq 0} \to \mathbb{R}$ satisfying the differential equation $\dot{x}(t) = dyn(m(t), x(t))$ for every $t \geq 0$ and with $x(0) \in X_0$.

In this paper, we consider tasks with two operation modes: $M = \{\text{ON}, \text{OFF}\}$. In addition, the dynamics of a task in each mode is given by an affine differential equation as follows:

$$\dot{x}(t) = \begin{cases} -a^+ x(t) + b^+ & \text{if } m(t) = \text{ON} \\ -a^- x(t) + b^- & \text{if } m(t) = \text{OFF} \end{cases} \tag{5}$$

where $a^+ > 0$, $a^- > 0$, b^+ and b^- are parameters. We assume that when T is ON, its state x grows, and when it is OFF, x decays. In particular, if T is ON during an interval $[t_1, t_2]$, the growth of x is given by

$$x(t) = \tfrac{b^+}{a^+} + \left(x(t_1) - \tfrac{b^+}{a^+}\right) e^{-a^+(t-t_1)}, \quad t_1 \leq t \leq t_2$$

where $x(t_1) < \frac{b^+}{a^+}$ and $x(t)$ asymptotically converges to $\frac{b^+}{a^+}$ as $t \to \infty$ (Fig. 2). Similarly, if T is OFF during an interval $[t_1, t_2]$, the decay of x is given by

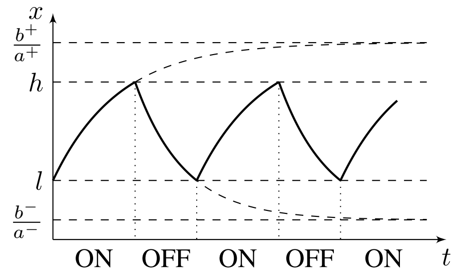

Fig. 2. Task dynamics.

$$x(t) = \tfrac{b^-}{a^-} + \left(x(t_1) - \tfrac{b^-}{a^-}\right) e^{-a^-(t-t_1)}, \quad t_1 \leq t \leq t_2$$

where $x(t_1) > \frac{b^-}{a^-}$ and $x(t)$ asymptotically converges to $\frac{b^-}{a^-}$ as $t \to \infty$ (Fig. 2). Thus x is bounded between $\frac{b^-}{a^-}$ and $\frac{b^+}{a^+}$, and a^-, a^+, b^-, b^+ must satisfy $\frac{b^-}{a^-} < \frac{b^+}{a^+}$. The state thresholds are such that $[l, h] \subset (\frac{b^-}{a^-}, \frac{b^+}{a^+})$, that is

$$\tfrac{b^-}{a^-} < l < h < \tfrac{b^+}{a^+}. \tag{6}$$

The dynamics of T is illustrated in Fig. 2.

A task is *safe* if its state always stays within the desired interval $[l, h]$; otherwise, it is *unsafe*. Let $\mathcal{T}$ be a set of $n > 1$ tasks: $\mathcal{T} = \{T_i\}_{i=1,\ldots,n}$. We index the variables and parameters of T_i by a subscript i, for example x_i and h_i. A *scheduling policy* for $\mathcal{T}$ is an algorithm that specifies the mode of every task T_i in $\mathcal{T}$ at any time during their execution. A schedule or scheduling policy for $\mathcal{T}$ is *safe* if it can keep all the tasks in $\mathcal{T}$ safe. Besides ensuring that all tasks are safe, a scheduling policy must satisfy a *system-wide resource constraint that at most k tasks can be in mode ON simultaneously at any time*, where k is given and $1 \leq k \leq n$. $\mathcal{T}$ is *schedulable* by a policy π if π is a safe scheduling policy for $\mathcal{T}$ and the resource constraint is met. If $\mathcal{T}$ is schedulable by some scheduling policy, it is *feasible*; otherwise, if $\mathcal{T}$ is not schedulable by any policy, it is *infeasible*.

Remark 1: Although the results in this paper are presented for systems resembling heated zones or buildings, they can be readily extended to systems which decay in ON mode and grow in OFF mode (e.g., cooling systems) by a change of variables. The applications are certainly not limited to heating and cooling systems but any system which can be abstracted by the task model defined in this section.

IV. Feasibility Analysis of Tasks

This section presents theorems for the feasibility and infeasibility of a set $\mathcal{T}$ of tasks as defined in the previous section. Their proofs can be found in [13].

The first theorem states a sufficient infeasibility condition.

Theorem 1 ([13]): For each task $T_i \in \mathcal{T}$, define

$$d_i = \frac{a_i^- l_i - b_i^-}{(a_i^- l_i - b_i^-) - (a_i^+ l_i - b_i^+)}. \tag{7}$$

If $d = \sum_{i=1}^{n} d_i > k$ then $\mathcal{T}$ is infeasible.

The next theorem provides a feasibility condition for the same task set $\mathcal{T}$ in Theorem 1.

Theorem 2 ([13]): If $d < k$ and at most k tasks start at their lower thresholds then $\mathcal{T}$ is feasible.

A. Interpretation of d

Consider the value d_i for task T_i as defined in (7). It is straightforward to show that $0 < d_i < 1$ and d_i is strictly increasing as l_i increases. We have

$$d_i \left|-a_i^+ l_i + b_i^+\right| = (1 - d_i) \left|-a_i^- l_i + b_i^-\right|$$

in which $|-a_i^+ l_i + b_i^+|$ and $|-a_i^- l_i + b_i^-|$ are respectively the growing and decaying rates of x_i at l_i. As x_i increases, the growing rate decreases while the decaying rate increases. It follows that, intuitively, d_i is the minimum fraction of time that T_i must be ON in order to keep x_i above l_i. For example, if $d_i = 0.6$ then on average, T_i must be ON for at least 60% of the time to stay safe. In other words, d_i can be thought of as the *minimum utilization* of task T_i. This minimum utilization is important because we must ensure that at most k tasks can be ON simultaneously, whereas there is no such constraint on the number of OFF tasks.

With this utilization-based interpretation of d_i, Theorems 1 and 2 become more intuitive. If $d > k$, the minimum total utilization of the tasks exceeds the resource capacity, thus they are infeasible. On the other hand, if $d < k$ then it is possible to schedule them. The case when $d = k$ is tricky. The minimum total utilization does not exceed the capacity; however, the actual total utilization is always larger than d due to the tasks' dynamics. Therefore the tasks are infeasible. The following corollary is straightforward.

Corollary 3: $\mathcal{T}$ is feasible if and only if $d < k$ and at most k tasks start at their lower thresholds.

V. Feasible Peak Minimization

In this section, we go back to Problem 1 and present the *feasible peak minimization* sub-problem in our approach.

Recall that this minimization is to compute Q_i, for $i = 1 \dots n$, and k so as to minimize the peak permissible demand P_{max}, subject to the constraint that with these values of Q_i and k, the tasks are feasible (Section II-B). As presented in Section IV, the feasibility of $\mathcal{T}$ is determined by the value d, which is the sum of the values d_i defined in (7). Each d_i depends on l_i, a_i^-, a_i^+, b_i^-, and b_i^+. It can be seen from (3) that while l_i, a_i^- and a_i^+ are fixed, b_i^- and b_i^+ depend on T_a and Q_i. Thus we can write $d\left(T_a, \{Q_i\}\right)$ as a function of T_a and all Q_i for $i = 1, \dots, n$. The feasible peak minimization is formulated as:

$$\begin{aligned} &\text{minimize} && J = \mathrm{Peak}\left(\{P_i(Q_i)\}, k\right) && \text{(8a)} \\ &\text{subject to} && Q_i \in \mathcal{Q}_i, \quad i = 1, \dots, n && \text{(8b)} \\ &&& k \in \{1, \dots, n\} && \text{(8c)} \\ &&& d\left(T_a, \{Q_i\}\right) < k && \text{(8d)} \end{aligned}$$

in which:

- The variables are k and Q_i for $i = 1, \dots, n$.
- $\mathrm{Peak}\left(\{P_i(Q_i)\}, k\right)$ computes the peak permissible demand by summing the k largest values in the set $\{P_i(Q_i)\}$ of tasks' powers.
- Constraint (8d) specifies the feasibility condition.

A. Complexity of feasible peak minimization

Recall that $\mathcal{Q}_i$ is the set of input heat levels of heater i excluding its OFF mode, and is finite. Thus (8) is a combinatorial optimization and in the worst case would require us to search over all combinations of Q_i, $i = 1, \dots, n$. For each combination, k is assigned to the smallest integer greater than d (constraint (8d)). The number of possibilities is $\prod_{i=1}^n |\mathcal{Q}_i|$. As a comparison, the MPC in (4) is also a combinatorial optimization with the number of possibilities being $\prod_{i=1}^n \left(1 + |\mathcal{Q}_i|\right)^H$, where H is the MPC horizon and 1 is added for the OFF mode. Since H is often greater than 1, the complexity of the MPC optimization is often exponentially larger than that of (8).

Although the feasible peak minimization is complex, its performance can be significantly improved by applying several boosting techniques which reduce the size of the search space. For instance, for the same value of k and a given combination $(Q_1, \dots, Q_n)$, a new combination can reduce the objective function Peak only if at least one of the Q_i is decreased. By only examining combinations that have the potential to decrease Peak and skipping the rest, the search space can be reduced in size. Similar techniques can be used for the MPC optimization (4).

Moreover, since the only changing parameter in (8) is T_a, the feasible peak minimization can be carried out off-line to obtain a table of solutions $(k, \{Q_i\})$ for each range of values of T_a. In run-time we only need to look up the solution in that table for the current value of T_a. This feature of the proposed approach allows it to be scalable and makes its implementation simple and light computationally.

VI. Scheduling Controller

The role of the *scheduling controller* is to carry out fine grained scheduling of the tasks based on state feedback from the zones (Fig. 1). It must ensure two constraints:

- *Resource constraint:* at most k tasks can be ON simultaneously; and
- *Safety constraint:* tasks always remain *safe*, i.e., $x_i(t) \in [l_i, h_i]$ at all time for every $i = 1, \dots, n$.

A. Scheduling control problem

Once the feasible peak minimization has computed a solution, the values of Q_i, for $i = 1, \dots, n$, and k are fixed until its next invocation. Meanwhile, each task T_i can only be turned ON or OFF. Let $m_i = 1$ when T_i is ON and $m_i = 0$ when T_i is OFF. The dynamics of x_i is given by (5). Let $x = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ denote the vector of states and $m = [m_1, \dots, m_n]^T \in \{0, 1\}^n$ the vector of modes of all tasks. The resource constraint is equivalent to the constraint $m \in \mathcal{M}$ where $\mathcal{M} = \{m \mid \|m\|_1 \leq k\}$. For matrices $X, Y \in \mathbb{R}^{m \times n}$, their Hadamard product $X \circ Y$ is defined as $[X \circ Y]_{ij} = X_{ij} Y_{ij}$. Let $\mathrm{diag}(v)$ denote the diagonal matrix constructed from the entries in vector v. Then the dynamics of the entire system is governed by

$$\dot{x}(t) = A_0 x(t) + B_0 + (A_1 x(t) + B_1) \circ m(t) \qquad (9)$$

in which the control inputs are $m(t) \in \mathcal{M}$ and

$$A_0 = -\mathrm{diag}\left(\left[a_1^-, \ldots, a_n^-\right]\right), \quad B_0 = \left[b_1^-, \ldots, b_n^-\right]^T$$
$$A_1 = -\mathrm{diag}\left(\left[a_1^+, \ldots, a_n^+\right]\right) - A_0,\ B_1 = \left[b_1^+, \ldots, b_n^+\right]^T - B_0$$

The safety constraint requires that $x(t)$ must stay within the invariant set $\mathcal{I} = [l_1, h_1] \times \cdots \times [l_n, h_n]$ at all time. Therefore the *scheduling control problem* can be stated as follows:

Problem 2: *(Scheduling control problem) Compute the control inputs $m(t)$ for system* (9) *so that at all time t:*

1) $m(t) \in \mathcal{M}$*; and*
2) $x(t) \in \mathcal{I}$.

Because the solution returned by the feasible peak minimization satisfies the feasibility condition in Theorem 2, such a scheduling controller always exists.

Essentially, a scheduling controller together with the tasks is a *hybrid system* S ([14]) with continuous state x, discrete modes $\mathcal{M}$, affine dynamics (9), and mode transitions determined by the controller. This system must be *safe* with respect to the set $\mathcal{I}$. The transition from mode $m \in \mathcal{M}$ to mode $m' \in \mathcal{M}$, $m' \neq m$, is associated with a guard $g_{m \to m'}(x)$ on the system's state. Whenever S is in mode m and $g_{m \to m'}(x)$ is satisfied, S switches to mode m'. Thus, a scheduling controller is equivalent to a set $\mathcal{G}$ of guards $g_{m \to m'}$ for all pairs of modes (m, m') such that S is safe.

B. Lazy scheduling controller

The number of discrete modes in $\mathcal{M}$ is $|\mathcal{M}| = \sum_{j=0}^{k} \binom{n}{j}$. When n and k are large, $|\mathcal{M}|$ becomes very large and a scheduling controller for the tasks can be large and complex. We propose a simple scheduling control policy called *lazy scheduling*. It is termed as *lazy* because switching decisions are only made when a task reaches either of its thresholds. Thus mode transitions only occur at the boundaries of the invariant set $\mathcal{I}$, and as long as x is in the interior of $\mathcal{I}$, the tasks stay in their current modes.

The lazy scheduling controller is simple. A task is *critical* if its state is at its lower threshold, thus it must be switched ON immediately. Let function $\mathbb{1} : 2^{\mathcal{T}} \to \{0, 1\}^n$ map a subset $\mathcal{O} \subseteq \mathcal{T}$ of tasks to mode vector $m = \mathbb{1}(\mathcal{O})$ such that $m_i = 1$ if and only if $T_i \in \mathcal{O}$. At time t, let m be the current mode and m' the next mode, i.e., m' is the decision of the controller. Then the lazy scheduling algorithm is as follows:

1) If $l_i < x_i(t) < h_i$ for all i then $m' = m$; otherwise
2) Let $\mathcal{C} = \{T_i \mid m_i = 0 \wedge x_i(t) = l_i\}$ be the set of critical tasks and $\mathcal{D} = \{T_i \mid m_i = 1 \wedge x_i(t) < h_i\}$ the set of tasks that can remain ON. Note that $\mathcal{C} \cap \mathcal{D} = \emptyset$. There are three cases:
   a) If $|\mathcal{C}| \geq k$ then $m' = \mathbb{1}(\mathcal{C})$; otherwise
   b) If $|\mathcal{C}| + |\mathcal{D}| > k$ then select a subset $\mathcal{D}' \subseteq \mathcal{D}$ of $(k - |\mathcal{C}|)$ tasks and $m' = \mathbb{1}(\mathcal{C} \cup \mathcal{D}')$; otherwise
   c) $m' = \mathbb{1}(\mathcal{C} \cup \mathcal{D})$.

If x is in the interior of $\mathcal{I}$ then Step 1 keeps the current mode. At the boundaries of $\mathcal{I}$, Step 2 turns OFF tasks at their upper thresholds, turns ON critical tasks, and keeps the other tasks' current modes so that at most k tasks can be ON in m', unless there are more than k critical tasks (Step 2a). By construction, the lazy scheduling controller is safe. It also tries to satisfy the resource constraint, but will violate it in the highly improbable event that more than k tasks become critical at *exactly* the same time. This can be avoided if a look-ahead rule is employed, in which the controller predicts whether such a situation may happen and switches ON certain tasks ahead of time.

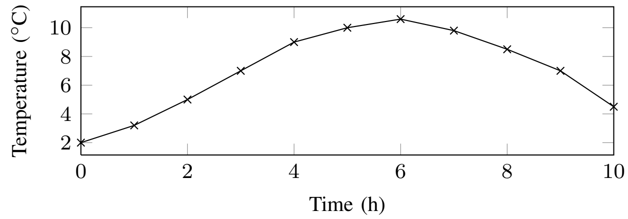

Fig. 3. Ambient air temperature profile.

VII. Simulation Results

We implemented the *feasible peak minimization* and the *lazy scheduling controller* in MATLAB, and compared the proposed approach to uncoordinated On-Off control and MPC in two examples of different scales.

A. Small-scale example

We considered 5 zones whose parameters were randomly generated around mean thermal capacity $\bar{C} = 5000\mathrm{kJ/K}$ and mean thermal conductance $\bar{K} = 0.35\,\mathrm{kW/K}$. They were heated by identical heaters with 5 heat input rates, $\mathcal{Q} = \{3, 6, 9, 12, 15\}$ (kW), and corresponding power $P(\mathcal{Q}) = \{3.26, 6.29, 9.24, 12.14, 15\}$ (kW). Zone temperatures must be kept between $l = 20\,°\mathrm{C}$ and $h = 22\,°\mathrm{C}$. We simulated them with uncoordinated On-Off control, MPC with 15-minute time steps and 3-step horizon, and our proposed green scheduling control. The simulation time was 10 hours. The ambient temperature profile is given in Fig. 3. In Fig. 4 are simulated zone temperatures, which always stayed inside the desired range. Fig. 5 plots the aggregate power demands of all controllers for comparison. Table I reports the peak demands and total energy consumptions.

B. Large-scale example

The previous example was extended to a larger scale. There were two groups of zones. The first consisted of 10 zones similar to those in the small-scale example. The second consisted of 10 significantly larger zones: their parameters were randomly generated around mean thermal capacity $\bar{C} = 30\,000\,\mathrm{kJ/K}$ and mean thermal conductance $\bar{K} = 1.4\,\mathrm{kW/K}$. The heaters for these zones were also larger, with heat input rates $\mathcal{Q}_{\mathrm{large}} = \{15, 30, 45, 60, 75\}$ (kW) and corresponding power $P(\mathcal{Q}_{\mathrm{large}}) = \{16.31, 31.44, 46.18, 60.68, 75\}$ (kW). Because the MPC optimization became much larger in this example (cf. Section V-A), we implemented for each group of zones an MPC controller with 15-minute time steps and 1-step horizon. Note that increasing the horizon to 2 would have made the MPC optimization intractable because of the large search space. Similarly, for the green scheduling

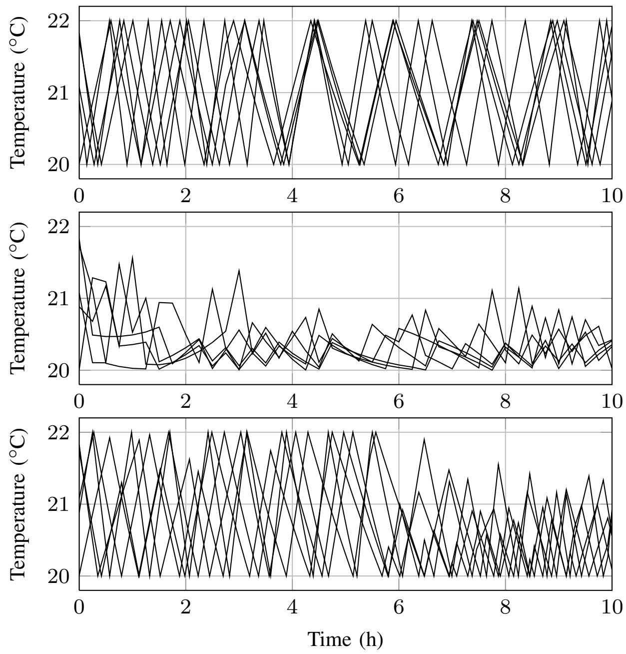

Fig. 4. Zone temperatures for small-scale example: uncoordinated On-Off control (top), MPC (middle), green scheduling control (bottom).

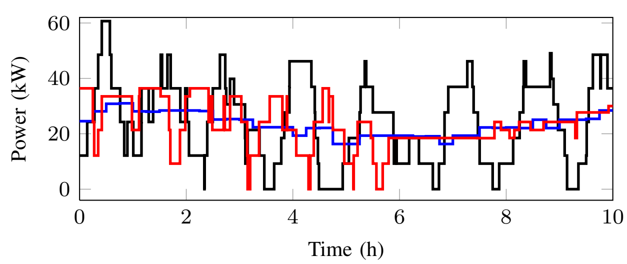

Fig. 5. Power demands for small-scale example: uncoordinated On-Off control (black), MPC (blue), green scheduling control (red).

controller, we solved the feasible peak minimization for each group of zones separately. Again, all controllers managed to keep zone temperatures in the desired range. Their peak demands and energy consumptions are reported in Table I.

C. Performance

Compared to the On-Off controller, the other two controllers significantly reduced the peak demand by about 40% or more (Fig. 6). Energy consumptions were similar, although the On-Off controller always consumed a little more. The MPC controller was slightly better than the green scheduling controller in terms of peak demand.

In the large-scale example, the feasible peak minimization took approximately 1.3 seconds on average. Thus, the proposed approach is scalable for large systems, even with online optimization. For comparison, each MPC optimization

TABLE I

PEAK DEMANDS P (kW) AND ENERGY CONSUMPTIONS E (kW h).

	Small-scale example			Large-scale example		
	On-Off	MPC	Green	On-Off	MPC	Green
P	60.68	31.24	36.41	587.87	331.90	347.49
E	243.97	233.27	234.14	2401.07	2286.5	2280.77

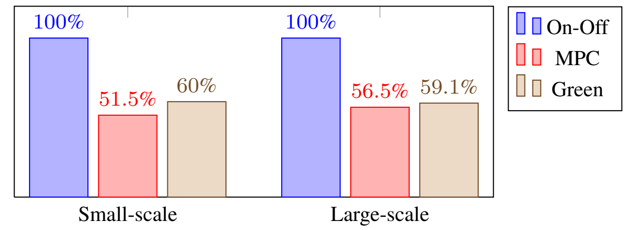

Fig. 6. Peak demand percentage (On-Off control as baseline).

took an average of approximately 174 seconds to complete.

VIII. Conclusion

A formulation for the scheduling problem of multiple control systems was presented. We proposed a new approach to energy efficient control of systems by combining optimization and scheduling. The optimization, which can be carried out efficiently and off-line, sets a feasible peak demand while the scheduling controller coordinates the systems within that constraint. Through simulations, this approach was shown to be effective in reducing peak demand and scalable for large systems. Its implementation is also simple and light computationally. This work is an initial step in the direction of resource and energy efficient coordination of control systems. In the future, we aim to extend the results to incorporate more complex system dynamics and disturbances. We are also investigating dynamic pricing models, operational efficiency and task-specific cost functions for system-wide optimization.

References

[1] TRFund Energy Study, "Understanding PECO's general service tariff."

[2] J. W. S. Liu, *Real-time Systems*. Prentice Hall, 2000.

[3] M. D. Vedova, M. Ruggeri, and T. Facchinetti, "On real-time physical systems," in *Proc. RTNS'10*, 2010, pp. 41–49.

[4] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," in *Proc. ACC'10*, 2010, pp. 5106–5111.

[5] 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 *Proc. ACC'10*, Jun. 2010, pp. 5100 –5105.

[6] 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 *Proc. IEEE CDC'10*, 2010, pp. 1927–1932.

[7] L. Parolini, E. Garone, B. Sinopoli, and B. H. Krogh, "A hierarchical approach to energy management in data centers," in *Proc. IEEE CDC'10*, 2010.

[8] E. Camacho and C. Bordons, *Model predictive control*. Springer, 2004.

[9] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[10] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[11] B. Li and A. G. Alleyne, "Optimal on-off control of an air conditioning and refrigeration system," in *Proc. ACC'10*, 2010, pp. 5892–5897.

[12] ASHRAE, *HVAC Applications*, ser. ASHRAE Handbook. Atlanta, GA: ASHRAE, 2007, ch. 41, p. 41.36.

[13] T. X. Nghiem, M. Behl, R. Mangharam, and G. J. Pappas, "On the schedulability of affine tasks," University of Pennsylvania, Tech. Rep., 2011.

[14] R. Alur, C. Courcoubetis, T. Henzinger, and P. Ho, "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid Systems*. Springer, 1993, pp. 209–229.