Discrete-Event Systems-Based Power Admission Control of Thermal Appliances in Smart Buildings

Waselul H. Sadid, Saad A. Abobakr, and Guchuan Zhu, *Senior Member, IEEE*

***Abstract*—This paper addresses the admission control of thermal appliances in the context of smart buildings. The scheduling of thermal devices operation is formulated in the framework of discrete-event systems, which allows for the modeling and design of admission control to be carried out in a systematic manner and ensuring the existence of the feasible scheduling prior to exploring control solutions. Two algorithms are developed for the purpose of peak demand reduction. While the firs algorithm validates the schedulability for the control of thermal appliances, the second algorithm may achieve a more efficien use of available capacity by exploring the concept of max–min fairness. Simulation studies are carried out in MATLAB/Simulink platform and the results show a noticeable improvement on peak power reduction.**

***Index Terms*—Admission control, smart buildings, discrete-event systems, scheduling, max–min fairness.**

I. INTRODUCTION

RESIDENTIAL and commercial sectors are the largest consumer of electricity worldwide. Nowadays, the residential and commercial buildings consume about 60% of the global electricity according to the united Nations Environment Programme (UNEP) [1]. It is therefore economically and environmentally important to develop solutions for power consumption reduction in buildings in an efficien and effective manner, which will greatly contribute to the initiative of the Smart Grid.

Different studies indicate that reducing the *peak* power consumption is one of the most effective ways for energy management in buildings in the context of the Smart Grid [2]–[6]. Heating, ventilation and air conditioning (HVAC) systems can play an important role in reducing the energy demand because they represent about 50% of the total energy utilization in buildings [7]–[9]. In addition of other thermal appliances, the control of this category of devices may have a significan and definit ve impact on peak power demand reduction. This is because of two main reasons. First, thermal appliances are the largest energy consumers in building sectors. In Canada, for example, almost 82% of the total energy use is consumed by these devices [10], including space heating (63%), space cooling (2%), and water heating (17%). Second and more importantly, the operation of thermal appliances exhibits an elastic property due to the slow thermodynamics and fl xible performance requirements. This allows designing and implementing different schemes to operate these devices to reduce the power consumption while providing an adequate comfort level for the occupants. Furthermore, peak power can cost as much as 200-400 times that of the nominal rate [11]. Hence, peak demand reduction has a very significan impact on real-life problems.

There exists a rich literature regarding energy consumption management in smart buildings. One of the most popular methods is to formulate the control of appliances operation as an optimal scheduling problem. A model developed in [12] is based on mixed integer linear programming (MILP) for household users, allowing the consumers to modify the scheduling of appliances operation by taking into account the comfort level and the total energy cost. In [4], an autonomous scheduling is evolved by accounting real-time information exchange between micro-grid and consumers. A greedy approach has been developed in [13] relying on the varying price to ensure an optimal start time of each appliance. Another greedy algorithm has been proposed in [14] to reduce the peak demand considering two cases, where load demands are either random or known in advance. A nonlinear optimization model is proposed in [15] to schedule home appliances to reduce electricity costs by shifting the consumption. Different scheduling algorithms were proposed in [16] to optimally manage the renewable energy and the energy from the grid to reduce the peak power consumption for deferable loads.

Some other approaches, such as game theory and model predictive control (MPC), have also been proposed and implemented for peak power demand reduction. The game theory is an interesting tool in designing control strategies for energy management in smart buildings. A tradeoff between these objectives has been proposed in [17] in the context of game theory, which optimizes the energy consumption by shifting the low priority appliances to off-pick period. A global optimal performance has been achieved in [6] based on the energy consumption of the consumers. A cooperative game has been set up in [18] among the users by sharing their load profiles which reduces the peak to average ratio. MPC is also

Manuscript received July 14, 2015; revised August 2, 2015, December 4, 2015, and January 21, 2016; accepted February 22, 2016. This work was supported in part by the Mitacs-Accelerate Graduate Research Internship Program, in part by Fusion Energy Inc., and in part by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-312116-2013. The work of S. A. Abobakr was supported by the Ministry of Higher Education of Libya and the College of Electronic Technology, Baniwaleed, Libya. Paper no. TSG-00825-2015.

The authors are with the Department of Electrical Engineering, Polytechnique Montreal, Université de Montréal, Montreal, QC H3T 1J4, Canada (e-mail: md-waselul-haque.sadid@polymtl.ca; saad.abobakr@polymtl.ca; guchuan.zhu@polymtl.ca).

Color versions of one or more of the f gures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifie 10.1109/TSG.2016.2535198

a popular control strategy, helping regulate the energy consumption and peak load reduction. An MPC-based dynamic power control scheme has been presented in [5] to reduce the peak demand while keeping the desired range of thermal comfort. A distributed MPC algorithm developed in [19] was applied on single and multi-zone thermal buildings, which reduces the power consumption significantl . An MPC controller is introduced in [20] for scheduling thermal and non-thermal appliances in a residential building, which shows a notable energy cost saving by considering time-varying pricing provided by the Smart Grid.

Note that the building is a complex system that may combine and integrate a big number of subsystems with significantl different dynamics, operating in different manners (time-triggered or event-triggered). Consequently, it may not be feasible to fin a single solution to manage the power consumption in the context of Demand-Response to achieve the global optimality. The layered structure developed in [2] can be considered as a solution that allows splitting the problem to a set of subproblems. With this architecture, a building energy management system can be divided into different modules in three main layers: admission controller (AC), load balancer (LB), and demand response manager (DRM). The AC interacts with appliances and manages the acceptance of the requests based on their priority and the available capacity. The DRM acts as an interface to the grid, collects data from both the grid and the building, and takes decision for demand regulation based on energy price. The LB coordinates the operation of AC and DRM by producing a long term scheduling to distribute the load over a time horizon and handles the power capacity constraints.

It should be noticed that extensive work on Demand-Response in the Smart Grid has been reported in the recent literature. It is recognized that the theory of optimization is a powerful tool to model and solve a great variety of problems in this fiel (see, e.g., [21] and the references therein). It is a common framework used in the development of diverse incentive-based and price-based Demand-Response programs in which different optimization techniques, such as convex optimization, integer programming, dynamic programming, stochastic programming, robust optimization, Markov decision process, game theory, and MPC control, have employed to achieve such objectives as utility maximization and cost reduction in the operation of the Smart Grid. From the consumer side, mathematical programming is also a powerful tool for the management of loads of different natures, the integration of renewable resources and storage, and the operation of different systems in the context of the Smart Grid. Indeed, most of the existing solutions are applicable or can be adapted to implement different components in the aforementioned architecture. In particular, the introduction of the AC layer allows hiding the details regarding the configuratio (e.g., the number of appliances in a system and their properties) and the operation of physical systems, and hence it will greatly facilitate the integration of different solutions for complex applications.

The present work focuses on the AC of thermal appliances in the context of the aforementioned layered architecture [2]. It is motivated by the fact that the AC interacts with the energy management system and the physical building, which is essential for the implementation of the concept and the solutions of smart buildings. The aim is to reduce peak power demand by coordinating the operation of appliances while respecting the power capacity constraints imposed by the higher layers. In this paper, we formulate the problem of scheduling a number of thermal appliances in the framework of discrete-event systems, with which the operation of each appliance can be expressed by a set of *states* and *events*, representing the status and the actions of the corresponding appliance and the communication within the AC layer. In DES, a system can be represented by a finite-stat machine (FSM) as a regular language over a finit set of events [22]. Admission control for appliances operation amounts then to deciding to accept or reject a request based on the priority and the capacity constraint.

Comparing with the existing work, the application of the framework of DES allows for the design of complex control systems to be carried out in a systematic manner. The main behaviors of a control scheme, such as the schedulability with respect to the given constraints, can be deduced from the basic system properties, in particular the controllability and the observability. In this way, we can benefi from the theory and the tools developed since decades for DES design and analysis to solve diverse problems with ever growing complexities arising from the emerging fiel of the Smart Grid. Moreover, it is worth noting that the operation of many power systems, such as economic signaling, demand-response, load management, and decision making, exhibits a discrete event nature. This type of problems can eventually be handled by utilizing continuous-time models. For example, the Demand Response Resource Type II (DRR Type II) at Midcontinent ISO market is modeled similar to a generator with continuous-time dynamics (see, e.g., [23]). Nevertheless, the framework of DES is a viable alternative for many modeling and design problems related to the operation of the Smart Grid.

The main contributions of this work are the following:

1) we introduced a formal formulation of power admission control in the framework of DES;
2) we established criteria for schedulability assessment based on DES theory;
3) we developed two algorithms for appliance scheduling in the framework of DES, which allow for peak power consumption reduction in smart buildings.

The remaining of the paper is organized as follows. Section II introduces the basic notions and terminologies of DES. Section III presents the model of appliances using finite-stat automaton. The DES-based admission control is addressed in Section IV. Two algorithms are proposed in this section for power distribution among the appliances. Simulation studies are carried out in Section V to validate the developed control schemes, followed by some conclusions and recommendation on future work in Section VI.

II. Background on DES

The DES is a framework for the modeling of systems whose dynamics can be described by transitions among a set of finit

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

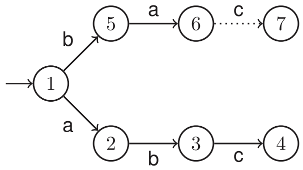

Fig. 1. An FSM M_L.

states, which can be used in the analysis and design of the sequences of (high-level) commands in many control systems. The aim is to fin a controller that forces the system to behave according to the imposed constraints. One of the most popular control architectures for DES is the supervisory control, which will restrict the system behavior to a set of design specifica tions [22], [24], [25]. The state space of a DES is a *discrete* set, and the transition mechanism is *event-driven*. Therefore, the system states can be changed only at discrete points of time corresponding to the occurrence of events.

The behavior of a DES is usually described by a regular language. A language, denoted by L, is said to be regular which can be represented by a finit automaton or FSM. An FSM, denoted by M_L, for L is a f ve-tuple: $M_L = (Q, \Sigma, \delta, q_0, Q_{\mathsf{m}})$, where Q is a finit set of states, Σ is a finit alphabet, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and Q_{m} is the set of marked states. The required behavior or specification denoted by K, must be a subset of the system behavior requiring control, i.e., $K \subseteq L$. In the supervisory control of DES, the controller keeps the system in K by issuing control directives to prevent the system from performing behavior in $L \setminus K$ ($L \setminus K$ denotes the set of sequences of L that are not in K). The controller issues a control decision (e.g., enable or disable an event) to fin a sublanguage of L, and the control objective is reached when correct pattern of control decisions is issued to keep the system in K.

Figure 1 shows an example of M_L in which the language generated from M_L is $L = \{\varepsilon, \mathsf{a}, \mathsf{b}, \mathsf{ab}, \mathsf{ba}, \mathsf{abc}, \mathsf{bac}\}$. If the specificatio contains only the solid line transitions, then $K = \{\varepsilon, \mathsf{a}, \mathsf{b}, \mathsf{ab}, \mathsf{ba}, \mathsf{abc}\}$.

The *prefi closure* of a language L is define as $\overline{L} := \{s \in \Sigma^* \mid (\exists s' \in \Sigma^*)$ such that $ss' \in L\}$. When L is prefix-closed $L = \overline{L}$. Unless otherwise stated, we work exclusively with prefix-close languages.

The closed behavior of a system, denoted by L, contains all possible event sequences the system may generate. The marked behavior of the system is L_{m}, which is a subset of the closed behavior, representing completed tasks (behaviors), and is define as $L_{\mathsf{m}} := \{s \in L \mid \delta(q_0, s) = q' \wedge q' \in Q_{\mathsf{m}}\}$. A system is nonblocking if $\overline{L}_{\mathsf{m}} = L$. A language K is said to be L_{m}-closed if $K = \overline{K} \cap L_{\mathsf{m}}$.

The synchronous product of two FSMs $M_i = (Q_i, \Sigma_i, \delta_i, q_{0i}, Q_{\mathsf{m}i})$, for $i = \{1, 2\}$, is denoted by $M_1 \| M_2$. It is define as $M_1 \| M_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_1 \| \delta_2, (q_{01}, q_{02}), Q_{\mathsf{m}1} \times Q_{\mathsf{m}2})$ [26].

By assumption, the set of events Σ is partitioned into disjoint sets of controllable and uncontrollable events, denoted by Σ_c and Σ_{uc}, respectively. Only controllable events can be prevented from occurring (i.e., may be disabled), as uncontrollable events are supposed to be permanently enabled. If in a supervisory control problem, only a subset of events can be observed by the controller, then the events set Σ can be partitioned also into the disjoint sets of observable and unobservable events, denoted by Σ_o and Σ_{uo}, respectively.

The controllable events of the system can be dynamically enabled or disabled by a controller according to the specifi cation, so that a particular subset of the controllable events is enabled to form a control pattern. The set of all control patterns can be define as:

$$\Gamma = \{\gamma \in P(\Sigma) | \gamma \supseteq \Sigma_{uc}\}, \tag{1}$$

where γ is a control pattern consisting only a subset of controllable events that are enabled by the controller. $P(\Sigma)$ represents the power set of Σ. Note that the uncontrollable events are also a part of the control pattern since they cannot be disabled by any controller.

A supervisory control for a system can be define as a mapping from the language of the system to the set of control patterns as $S : L \to \Gamma$. A language K is controllable if and only if [22]

$$\overline{K}\Sigma_{uc} \cap L \subseteq \overline{K}. \tag{2}$$

The controllability criterion means that an uncontrollable event $\sigma \in \Sigma_{uc}$ can not be prevented from occurring in L. Hence, if such an event σ occurs after a sequence $s \in \overline{K}$, then σ must remain through the sequence $s\sigma \in \overline{K}$. For example, in Fig. 1, K is controllable if the event c is controllable; otherwise, K is uncontrollable.

For event based control, a controller's view of the system behavior can be modeled by the natural projection, $\pi : \Sigma^* \to \Sigma_o^*$, define as

$$\pi(\sigma) = \begin{cases} \varepsilon, & \text{if } \sigma \in \Sigma \setminus \Sigma_o; \\ \sigma, & \text{if } \sigma \in \Sigma_o, \end{cases} \tag{3}$$

where ε denotes the empty sequence. This operator removes the events σ from a sequence in Σ^* that are not found in Σ_o. The above definitio can be extended to sequences as follows: $\pi(\varepsilon) = \varepsilon$, and $\forall s \in \Sigma^*, \sigma \in \Sigma.\ \pi(s\sigma) = \pi(s)\pi(\sigma)$. The inverse projection of π for $s' \in \Sigma_o^*$ is the mapping from Σ_o^* to $P(\Sigma^*)$:

$$\pi^{-1}(s') = \left\{u \in \Sigma^* \mid \pi(u) = s'\right\}. \tag{4}$$

A language K is said to be observable with respect to L, π, and Σ_c if for all $s \in \overline{K}$ and all $\sigma \in \Sigma_c$ it holds [27]

$$(s\sigma \notin \overline{K}) \wedge (s\sigma \in L) \Rightarrow \pi^{-1}[\pi(s)]\sigma \cap \overline{K} = \emptyset. \tag{5}$$

In other words, the projection π provides the necessary information to the controller to decide whether an event to be enabled or disabled to attain the system specification

If the controller receives the same information through different sequences $s, s' \in L$, it will take the same action based on its partial observation. Hence, the decision is made in observationally equivalent manner as below:

$$\left(\forall s, s' \in L\right)\pi(s) = \pi\left(s'\right) \Rightarrow \Gamma(\pi(s)) = \Gamma\left(\pi\left(s'\right)\right). \tag{6}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

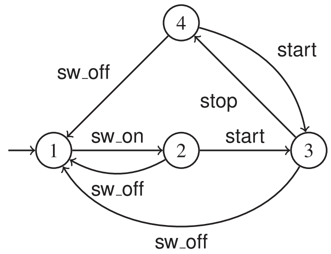

Fig. 2. A finite-stat automaton of an appliance.

A controller can be synthesized when the specificatio $K \subseteq L$ is controllable and observable. This is the case where the controller can take correct control decision based only on its observation of a sequence.

III. DES Model and Control of Smart Appliances

A. Modeling Appliance Operation in DES

In the considered problem, the AC manages the access to power and controls the appliances operation. Requests from each appliance will be accepted or rejected by the AC depending on the priority and the capacity available in each invocation. Appliance load management can be represented by an FSM as shown in Fig. 2. In other words, appliance operation can be characterized by a set of states and events. An appliance changes its state by executing an event. Hence, it can be easily formulated in DES, which describes the behavior of the load requiring control and the specificatio as regular languages (over a set of discrete events).

In the considered problem, each appliance is characterized by a set of variables: status, preemption, heuristic value, and requested power. The status of an appliance can be define w.r.t. the states of the FSM shown in Fig. 2 as below:

State 1 (Off): it is not enabled.

State 2 (Ready): it is ready to start.

State 3 (Run): it runs and consumes power.

State 4 (Idle): it stops and consumes no power.

Since the interruption and the priority of each appliance should be considered while scheduling the operation, preemption and heuristic strategies will be used for these tasks. Preemption specifie whether the task is interrupted to give priority to another task, while heuristic value define the priority of the appliance. The computation of heuristic values is based on performance considerations and other factors, such as the deadline and the laxity of requests [2], [16].

The requested power is the power that required for an appliance to run. However, the total power consumption has to respect the maximum demand of the building during power distribution over the appliances.

B. Schedulability Analysis of Admission Control

In the supervisory control to accept or reject a request of an appliance, a controller decides which events are enabled through a sequence based on its observation. Essentially, the schedulability of appliances operation depends on two basic properties of DES: controllability and observability. Therefore, in control synthesis, a view $\mathcal{C}_i$ is firs designed for each appliance i from controller perspective. If the appliances operations are independent of each other, then a centralized controller's view $\mathcal{C}$ can be formulated by taking the synchronous product of the individual views. However, when two or more appliances are dependent, the view of the corresponding appliances are firs generated together and then, we take the synchronous product with the rest of the appliances to design $\mathcal{C}$. The schedulability of such a scheme will be assessed based on the properties of the considered system and the controller. Figure 3 illustrates the process for accepting or rejecting the request of an appliance.

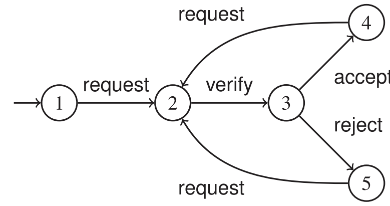

Fig. 3. Accepting or rejecting the request of an appliance.

Let $L_{\mathcal{C}}$ be the language generated from $\mathcal{C}$ and $K_{\mathcal{C}}$ be the specification Then the control law is a map

$$\Gamma : \pi(L_{\mathcal{C}}) \rightarrow P(\Sigma),$$

such that $\forall s \in \Sigma^*$, $\Sigma_{L_{\mathcal{C}}}(s) \cap \Sigma_{uc} \subseteq \overline{K}_{\mathcal{C}}$, where $\Sigma_{\mathcal{L}}(s)$ define the set of events occur after the sequence $s : \Sigma_{\mathcal{L}}(s) = \{\sigma \in \Sigma | s\sigma \in \overline{\mathcal{L}}\}$, $\forall \mathcal{L} \subseteq \Sigma^*$.

The control decisions will be made in observationally equivalent manner as specifie in (6).

Definitio 1: Denote by $\Gamma/L_{\mathcal{C}}$ the controlled system under the supervision of Γ. The closed behavior of $\Gamma/L_{\mathcal{C}}$ is define as a language $\mathcal{L}(\Gamma/L_{\mathcal{C}}) \subseteq L_{\mathcal{C}}$, such that

(i) $\varepsilon \in \mathcal{L}(\Gamma/L_{\mathcal{C}})$, and

(ii) $\forall s \in \mathcal{L}(\Gamma/L_{\mathcal{C}})$ and $\forall \sigma \in \Gamma(s)$, $s\sigma \in L_{\mathcal{C}} \Rightarrow s\sigma \in \mathcal{L}(\Gamma/L_{\mathcal{C}})$.

The marked behavior of $\Gamma/L_{\mathcal{C}}$ is $L_{\mathsf{m}}(\Gamma/L_{\mathcal{C}}) = \mathcal{L}(\Gamma/L_{\mathcal{C}}) \cap L_{\mathsf{m}}$.

Denote by $\Gamma/M_{L_{\mathcal{C}}}$ the system $M_{L_{\mathcal{C}}}$ under the supervision of Γ and let $\mathcal{L}(\Gamma/M_{L_{\mathcal{C}}})$ be the language generated by $\Gamma/M_{L_{\mathcal{C}}}$. Then, the necessary and sufficien conditions for the existence of a controller satisfying $K_{\mathcal{C}}$ are given by the following theorem [22].

Theorem 1: There exists a controller Γ for the system $M_{L_{\mathcal{C}}}$ such that $\Gamma/M_{L_{\mathcal{C}}}$ is nonblocking and the closed behavior of $\Gamma/M_{L_{\mathcal{C}}}$ is restricted to K (i.e., $\mathcal{L}(\Gamma/M_{L_{\mathcal{C}}}) \subseteq K_{\mathcal{C}}$) if and only if

(i) $K_{\mathcal{C}}$ is controllable w.r.t. $L_{\mathcal{C}}$ and Σ_{uc},

(ii) $K_{\mathcal{C}}$ is observable w.r.t. $L_{\mathcal{C}}$, π and Σ_c, and

(iii) $K_{\mathcal{C}}$ is L_{m}-closed.

IV. DES-Based Scheduling Algorithm

In the considered AC problem, a controller is designed to schedule the appliances operation depending on the available capacity. Note that we consider the worst case in algorithm design, because all the appliances may issue requests at the same time.

Theorem 1 indicates that a feasible scheduling is achievable if and only if the specificatio $K_{\mathcal{C}}$ is controllable and

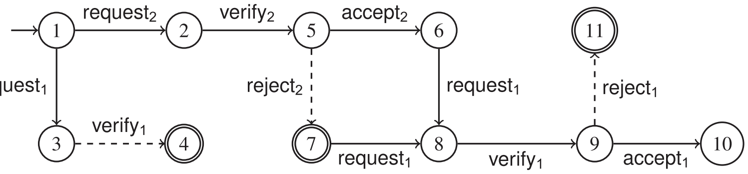

Fig. 4. A portion of centralized controller for 2 appliances.

observable. For the controller's observation shown in Fig. 3, only one request with the highest priority can be verifie in one state in the system behavior $L_{\mathcal{C}}$. But the other requests cannot be disabled, because the event request generated by an appliance is not under the control. In this case, the subsequent event verify is disabled by the controller. In addition, if a request is accepted, then the event accept related to that request is enabled and the event reject is disabled by the controller. Otherwise, accept event is disabled and reject event is enabled. After verifying all the requests in $L_{\mathcal{C}}$, the corresponding accept or reject events are enabled. In this way, the sequence representing the optimal scheduling of the appliances is explored from a set of sequences in DES, which shows alternate ways to do the scheduling. As request is the only uncontrollable event, which is never disabled by the controller, $K_{\mathcal{C}}$ is controllable. In addition, when an appliance is non-preemptive, the corresponding accept event becomes uncontrollable. In that case, if there is no available capacity to enable that event, then $K_{\mathcal{C}}$ becomes uncontrollable. Consequently, there is no feasible schedule. At the same time, the controller takes the same decision for the sequences observationally equivalent. Hence, $K_{\mathcal{C}}$ is observable. In addition, the language L_{m} define the sequence that represent the scheduling of appliances. Then $K_{\mathcal{C}}$ is L_{m}-closed.

A. Scheduling Algorithm in DES

The firs algorithm to implement is adapted from [2] and [28]. As detailed in Algorithm 1, when the requests from different appliances are generated, the priority of each appliance is define according to its heuristic value. When an event request is generated, in the subsequent states, its acceptance will be checked through the event verify to determine the next state to transit to (accept or reject). After executing a request, the one with the highest priority will be accounted in the next state.

In addition, the requests come from the non-preemptive appliances are verifie first because such appliances cannot be interrupted. The remaining appliances are verifie sequentially according to their priorities. Indeed, the preemptivity, which is a static behavior of an appliance, and the priority are the only information required by the controller to decide which is the next request to be verified

Since a DES control scheme can be generated as a regular language, it is possible to modify the language according to the given specification The controlled behavior to be constructed is ensured by Algorithm 1 for the considered AC. We consider the following variables in the algorithm:

- $\mathcal{N}$: set of requests
- n: number of requests
- i, j: request $\in \mathcal{N}$
- p_i: requested power of the appliance associated to request i
- *cons*: total power consumption of the accepted requests
- C: capacity limit

To illustrate the operation of the AC scheme given in Algorithm 1, we consider the following example.

Example 1: A controller's view $\mathcal{C}_i$ is modeled as shown in Fig. 3 for each appliance $i \in \mathcal{N}$, where the appliances operations are independent on each other. Then $\mathcal{C}$ is formulated by taking the synchronous product $\mathcal{C}_i, \forall i \in \mathcal{N}$ as $\mathcal{C} = \mathcal{C}_1 || \dots || \mathcal{C}_N$. Suppose that $L_{\mathcal{C}}$ is the language generated by the collection of all transitions, whereas $K_{\mathcal{C}}$ is the language generated by solid line transitions.

For simplicity, we consider the case of 2 appliances. The synchronous product gives 25 states and 60 transitions in this case. Suppose that Appliance 2 has a higher priority. Then Appliance 2 is verifie firs to decide whether it is accepted. Then corresponding accept_2 or reject_2 event enabled. Then, the request generated by Appliance 1 ($\mathsf{request}_1$) is considered and the same steps are repeated. A portion of the controlled system is shown in Fig. 4 in which the marked states are denoted by double circle (avoid to reach in this example) and the disabled transitions are denoted by dotted line. In this example, both requests are accepted as the execution result. In this way, the scheduling generated by Algorithm 1 satisfie $K_{\mathcal{C}}$.

Note that if the request generated from Appliance 1 depends on the acceptance of Appliance 2, then $\mathcal{C}$ becomes simpler. In that case, we have to design the FSM in such a way that the event $\mathsf{request}_1$ only appears after the event accept_2. Consequently, the sequences with the appearance of $\mathsf{request}_1$ before accept_2 will be removed from $\mathcal{C}$ while performing the synchronous product. □

The following theorem provides the schedulability assessment for the developed AC scheme.

Theorem 2: If the appliances are scheduled following Algorithm 1, then $\sum_{i \in \mathcal{N}} p_i \leq C$.

Proof: Scheduling of the appliances in DES provides the controlled behavior which is obtained from Algorithm 1. That means it satisfie the specificatio $K_{\mathcal{C}}$, which implies that $K_{\mathcal{C}}$ is controllable. Then assume that $\mathcal{L}(\Gamma/M_{L_{\mathcal{C}}}) \subseteq K_{\mathcal{C}}$, but $\sum_{i \in \mathcal{N}} p_i > C$. That means there exists a request $j \in \mathcal{N}$, which is accepted by the controller Γ, exceeds the capacity C. Then $\exists j \in \mathcal{N}$ such that $\sum_{i \in \mathcal{N} \setminus \{j\}} p_i \leq C$. This implies that $\sum_{i \in \mathcal{N} \setminus \{j\}} p_i + \sum_{j \neq i} p_j > C$. Hence, $\mathsf{request}_j$ cannot be accepted by Algorithm 1. Therefore, there exists a sequence $s \in L_{\mathcal{C}}$ such that $s\mathsf{accept}_j \notin K_{\mathcal{C}}$ and $s\mathsf{reject}_j \in K_{\mathcal{C}}$. In that case, if $s\mathsf{accept}_j \in \mathcal{L}(\Gamma/M_{L_{\mathcal{C}}})$, then $\mathcal{L}(\Gamma/M_{L_{\mathcal{C}}}) \supset K_{\mathcal{C}}$, which leads to a contradiction. ■

Algorithm 1 Algorithm for Admission Controller

Input: Sort the requests according to the heuristic value in descending order
$cons = 0$
$i = 1$
repeat
  if request i is non-preemptive **then**
    enable accept_i
    $cons = cons + p_i$; remove request i from $\mathcal{N}$
  end if
  $i = i + 1$
until $i <= n$
update $\mathcal{N}$ and n
$i = 1$
repeat
  enable verify_i, disable verify_j for $j \in \mathcal{N} \setminus \{i\}$
  if $cons + p_i <= C$ and appliance i is running **then**
    enable accept_i and disable reject_i
    $cons = cons + p_i$
  else
    enable reject_i and disable accept_i
  end if
  $i = i + 1$
until $i <= n$

Algorithm 2 Max-Min-Scheduling Algorithm

Input: Sort the requests according to the heuristic value in descending order
$cons = 0$, $p_i = 0$ for all $i \in \mathcal{N}$
$i = 1$
repeat
  if request i is non-preemptive **then**
    enable accept_i
    $cons = cons + p_i$; remove request i from $\mathcal{N}$
  end if
  $i = i + 1$
until $i <= n$
update $\mathcal{N}$ and n
$t = \min\{p_{\mathrm{lb}_i}\}$
$i = 1$
repeat
  enable verify_i, disable verify_j for $j \in \mathcal{N} \setminus \{i\}$
  if $cons + t <= C$ and appliance i is running **then**
    enable accept_i
    $p_i = p_i + t$, $cons = cons + t$
    $p_{\mathrm{lb}_i} = p_{\mathrm{lb}_i} - t$ and $p_{\mathrm{ub}_i} = p_{\mathrm{ub}_i} - t$
  end if
  $i = i + 1$
until $i <= n$
Repeat the previous steps considering $t = \min\{p_{\mathrm{ub}_i}\}$

Note that the scheduling algorithm is applicable to the considered system regardless of the dependency among the appliances. In fact, the dependency will be handled by the controller's view $\mathcal{C}$ through the design of appropriate FSMs. In each state of the FSM, the role of the algorithm is to decide which is the next state to be executed, so that the whole system can evolve following the correct sequence. Moreover, $\mathcal{C}$ becomes less complex in the presence of dependencies.

B. Max-Min-Scheduling Algorithm

Algorithm 1 only accepts or rejects a request depending on the available capacity. However, it may happen that more than one request have the same priority while the total demand exceeds the capacity limit. It implies that not all of the requests can be accepted. Obviously, disabling an amount of appliances may result in an inefficien use of the available power and lead to the performance degradation. To fin solutions to distribute the available power among the appliances in an optimal manner, we suppose that the acceptable power for each appliance i belongs to a so-called fallibility range $p_{\mathrm{lb},i} \leq p_i \leq p_{\mathrm{ub},i}$, where $p_{\mathrm{lb},i}$ and $p_{\mathrm{ub},i}$ denote, respectively, the lower and the upper bounds of the requested power. We then consider an optimal power distribution in the sense of max-min fairness, which is a widely used scheme for resource allocations (see, e.g., [29]). The use of max-min fair schemes can allocate as much power as possible to all the appliances while keeping the distribution balanced. This means that we can run the maximum number of appliances simultaneously.

The max-min fair power distribution among a set of N appliances indexed by $\mathcal{N} = \{1, \dots, N\}$ can be formulated as a linear programming problem by setting $t \leq p_i$, for all $i \in \mathcal{N}$, where p_i are the decision variables, expressed as:

$$\begin{aligned} \max \quad & t; && \text{(7a)} \\ \text{s.t.} \quad & t - p_i \leq 0, \quad \forall i \in \mathcal{N}; && \text{(7b)} \\ & p_{\mathrm{lb},i} \leq p_i \leq p_{\mathrm{ub},i}, \quad \forall i \in \mathcal{N}; && \text{(7c)} \\ & \sum_{i \in \mathcal{N}} p_i \leq C. && \text{(7d)} \end{aligned}$$

Using the above formulation, we update Algorithm 1 to a max-min-scheduling (MMS) scheme, given in Algorithm 2, that allows for a more efficien use of the available capacity. In addition to the variables of Algorithm 1, few more parameters are considered as below:

- p_{lb_i}: minimum power required for request i
- p_{ub_i}: maximum power required for request i

The MMS algorithm initially distributes the minimum power request selected from the feasibility set to the appliances. Then it maximizes the power consumption of each appliance according to its feasibility range and the available capacity. Note that the linear programming problem involved in MMS can be solved efficientl by using the progressive fillin algorithm [29]–[31], which is the solution adopted in Algorithm 2.

Schedulability assessment of the AC scheme produced by Algorithm 2 is given below.

Corollary 1: If the appliances are scheduled following Algorithm 2, then $\sum_{i \in \mathcal{N}} p_i \leq C$.

Proof: When appliances are scheduled by employing Algorithm 2, the corresponding event sequence provides the controlled behavior of the system. That means $K_{\mathcal{C}}$ is controllable. The rest of the proof follows Theorem 2. ∎

Finally, the worst case computation complexity of both algorithms is $\mathcal{O}(n^2)$, where n is the number of requests to proceed.

V. Simulation Studies

A. Simulation Setup

The simulation studies are carried out with Matlab/Simulink. Two different types of thermal appliances, namely heater and refrigerator, are considered in the simulation. The dynamical model of the heating system and the refrigerators are taken from [5] and [32]. Specificall , the dynamics of the heating system are given by:

$$\frac{\mathrm{d}T_i}{\mathrm{d}t} = \frac{1}{C_i R_i^a}(T_a - T_i) + \frac{1}{C_i}\sum_{j=1, j\neq i}^{N} \frac{1}{R_{ji}}\left(T_j - T_i\right) + \frac{1}{C_i}\Phi_i, \quad (8)$$

where N is the number of rooms, T_i is the interior temperature of room i, $i \in \{1, \ldots, N\}$, T_a is the ambient temperature, R_i^a is the thermal resistance between room i and the ambient, R_{ji} is the thermal resistance between Room i and Room j, C_i is the heat capacity and Φ_i is the power input to the heater of room i. The dynamical model of the refrigerator takes a similar, but simpler form:

$$\frac{\mathrm{d}T_r}{\mathrm{d}t} = \frac{1}{C_r R_r^a}(T_a - T_r) - \frac{A_c}{C_r}\Phi_c, \quad (9)$$

where T_r is the refrigerator chamber temperature, T_a is the ambient temperature, C_r is a thermal mass representing the refrigeration chamber, the insulation is modeled as a thermal resistance, R_r^a, A_c is the overall coefficien performance, and Φ_c is the refrigerator power input. The parameters of thermodynamic models used in the simulaton are given in the Appendix.

The considered setup in experiment contains four rooms. Room 1 and 2 include one heater and one refrigerator each, and Room 3 and 4 contain only one heater each. A classical proportional-integral (PI) controller is used to control the heating system with the maximal temperature set to 24°C. An ON/OFF scheme is used to control the refrigerators that are turned on when the chamber temperature is 3°C and shut down when temperature reaches 2°C. In addition, the room temperature is considered as the ambient temperature for the refrigerator placed in the corresponding room, while the ambient temperature of each room is set to 10°C. Moreover, unlike the heaters, the refrigerators are non-preemptive loads. The centralized controller's view ($\mathcal{C}$) for all the appliances is generated using Matlab Toolbox DECK [33], which results in a system with 4096 states and 18432 transitions. Note that if there exists any dependency between the appliances, some transitions will be excluded sequentially. Consequently, the corresponding $\mathcal{C}$ becomes smaller with less number of states and transitions.

In the experiments, the heuristic value is define as a factor between 0 and 1. As the heaters are controlled by PI controller, which amounts to regulating the temperature to a predefine reference $T_{\max}$, the heuristic value for heaters is computed as:

$$v_h = \max\left\{\frac{2}{\pi}\arctan(T_{\max} - T), 0\right\}. \quad (10)$$

On the other hand, the refrigerators are operated in an ON/OFF manner depending on the lower and upper temperature bounds. Therefore, the heuristic value for refrigerators takes 0 or 1 when the temperature is in or out of the define zone.

B. Simulation Results and Analysis

We validate firs Algorithm 1. In the simulation, the time span is normalized to 200 time units. It is supposed that the initial requested power is 1600 W for Heater 1 and 2, and 1200 W for Heater 3 and 4. The requested power for the refrigerators is 500 W. Hence, the total power required to accept all the requests can be as high as 6600 W. However, we consider 3000 W for the firs 60 time units as the total capacity limit, which is less than the 1/2 of the total initial power required in the worst case. Then the capacity limit is reduced to 1000 W at 60 time steps and to 500 W at 120 time steps. The three power capacity levels are about, respectively, 45.5%, 15%, and 7.5% of the worst case peak power demand. This configuratio is reasonable in the sense that the start-up or pre-heating should be performed in off-peak time periods, so that the total consumption may become much less during the on-peak period. Note that 500 W is the capacity limit below which there does not exist feasible scheduling. Note also that the capacity limit is a trade-off between the energy consumption and the thermal comfort which has to be generated by the LB and DRM layers in the context of the considered architecture.

The acceptance status of the heaters ($H1, \ldots, H4$) and the refrigerators ($FR1, FR2$) is shown in Fig. 5. The operational state of the appliances is indicated by 'ON' and 'OFF'. The total power consumption of all the appliances, as well as the power constraints, are depicted in Fig. 6. Figure 7 shows individual room temperatures ($R1, \ldots, R4$) and the refrigerator temperatures ($FR1, FR2$). The temperature of all the rooms is maintained in a range between 22.6° and 24°C until 120 time steps. With a power capacity limit of 500W, the overall performance is still acceptable where the room temperature is maintained in a range between 22.6° and 23.6°C most of the time. However, there is a slight performance degradation at some points with a room temperature as low as 20.8°C, because all the available capacity has been used by a refrigerator. Furthermore, the temperature of the refrigerators is always maintained in the range between 2° to 3°C, as imposed in experiments. Therefore, this control scheme with an adequate tuning of heuristic values can keep the room and refrigerator temperatures in the desired range when the power capacity is set to 1000 W, which is only 15% of the worst case peak power demand.

Then we run the simulation to validate Algorithm 2 for the same setup but with a capacity limit of 2400 W for (0, 60) time steps, which is about 36% of the worst case peak power demand, and then 1000 W and 500 W for the remaining time as in the previous experiment. Note that by using this algorithm, all the appliances are accepted at the startup step with less capacity. This means that Algorithm 2 is more effective in terms of peak demand reduction. Figures 8 and 9 illustrate the acceptance status and the total consumption corresponding to Algorithm 2. The results show that compared to Algorithm 1,

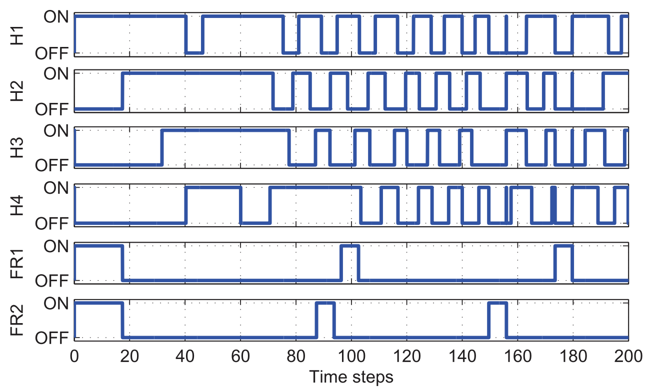

Fig. 5. Acceptance of appliances using Algorithm 1.

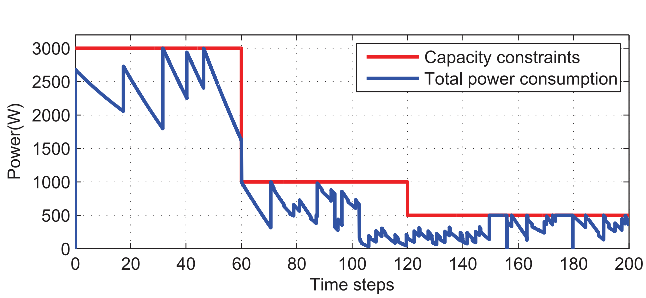

Fig. 6. Total power consumption of appliances using Algorithm 1.

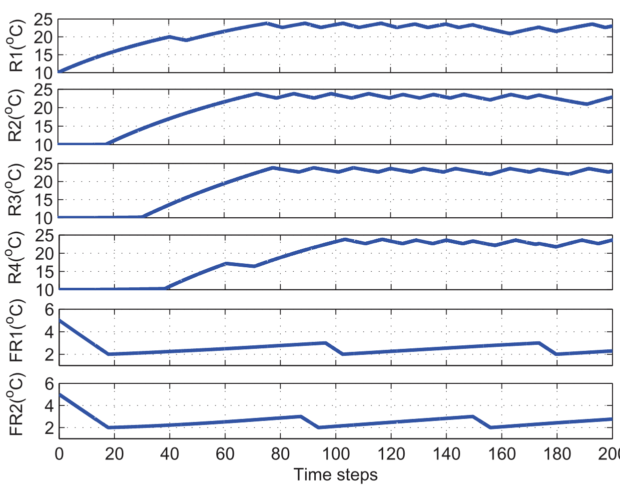

Fig. 7. Room and refrigerator temperatures using Algorithm 1.

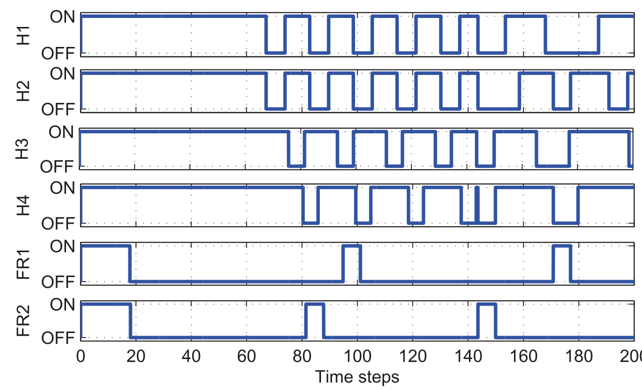

Fig. 8. Acceptance of appliances by Algorithm 2.

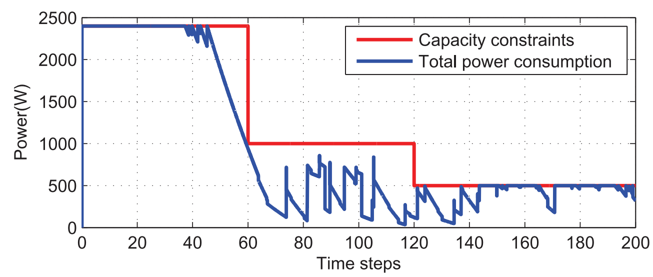

Fig. 9. Total power consumption of appliances using Algorithm 2.

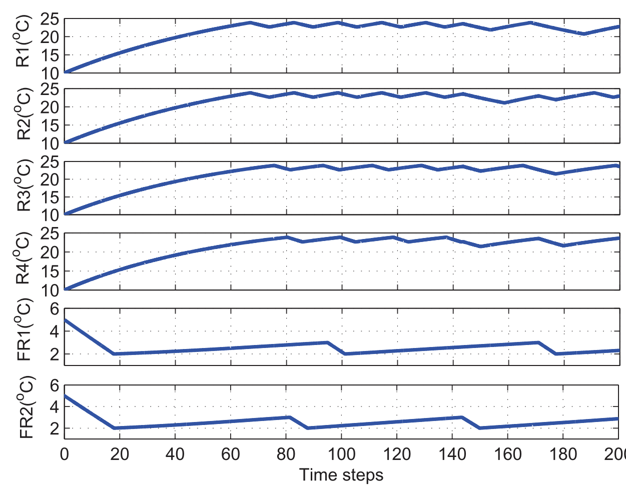

Fig. 10. Room and refrigerator temperatures using Algorithm 2.

the appliances switch less frequently between ON and OFF and the total power profil is much flatte . Therefore, this scheme is less harmful to the appliance life and power system stability. The temperature of individual rooms and refrigerators is shown in Fig. 10. It can be seen that the room temperatures reach to the desired range in a more uniform manner. Moreover the temperature in all the rooms is maintained between 22.6° and 24°C most of the time, except a slight degradation for a room with a temperature as low as 20.6°C when a refrigerator is turned on while the power capacity limit is set to 500 W. It shows that this scheme is very efficien for peak power demand reduction.

Finally, the simulation of 200 time steps with a sampling rate of 0.01 time unit takes 24.5s for Algorithm 1 and 28.4s for Algorithm 2 in a Pentium 4 PC.

VI. Conclusion

This paper presents a novel formulation for the admission control of thermal appliances in the framework of DES. This approach allows assessing the existence of the controller for scheduling appliances operation with respect to the power capacity constraints in smart buildings. Two algorithms are developed to implement the proposed scheme. The simulation results show that both algorithms are effective in peak demand reduction, while Algorithm 2, based on max-min fairness, is more efficien requiring less power capacity than Algorithm 1.

TABLE I
CONFIGURATION OF THERMAL PARAMETERS FOR HEATERS

Room	1	2	3	4
R_j^a	69.079	88.652	128.205	105.412
C_j	0.94	0.94	0.78	0.78

TABLE II
PARAMETERS OF THERMAL RESISTANCES FOR HEATERS

R_{12}^r, R_{21}^r	R_{13}^r, R_{31}^r	R_{14}^r, R_{41}^r	R_{23}^r, R_{32}^r	R_{24}^r, R_{42}^r
709.2	1063.8	1063.8	1063.8	1063.8

TABLE III
PARAMETERS OF REFRIGERATORS

Fridge	R_r^a	C_r	$T_r(0)$	A_c	Φ_c
1	1.4749	893.74	5	0.34017	500
2	1.4749	804.37	5	0.2846	500

Note that eventually, other type of fairness can also be considered in control design. Moreover, decentralized discrete-event control is more suitable for large scale systems. We may also consider the application of MPC-based techniques in the framework of DES in order to guarantee the performance of appliances operation.

APPENDIX
MODEL PARAMETERS

The parameters of the thermodynamic models are listed in Tables I, II and III, which have been normalized from the real values to run the simulation for 200 time units.

REFERENCES

[1] United Nations Environment Programme. (2015). *Why Buildings*. [Online]. Available: http://www.unep.org/sbci/AboutSBCI/Background.asp.

[2] G. T. Costanzo, G. Zhu, M. F. Anjos, and G. Savard, "A system architecture for autonomous demand side load management in smart buildings," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 2157–2165, Dec. 2012.

[3] M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "An algorithm for intelligent home energy management and demand response analysis," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 2166–2173, Dec. 2012.

[4] C. O. Adika and L. Wang, "Autonomous appliance scheduling for household energy management," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 673–682, Mar. 2014.

[5] J. Yao, G. T. Costanzo, G. Zhu, and B. Wen, "Power admission control with predictive thermal management in smart buildings," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2642–2650, Apr. 2015.

[6] J. L. Mathieu, P. N. Price, S. Kiliccote, and M. A. Piette, "Quantifying changes in building electricity use, with application to demand response," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 507–518, Sep. 2011.

[7] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy Build.*, vol. 40, no. 3, pp. 394–398, 2008.

[8] A. Afram and F. Janabi-Sharifi "Theory and applications of HVAC control systems—A review of model predictive control (MPC)," *Build. Environ.*, vol. 72, pp. 343–355, Feb. 2014.

[9] F. R. Pazheri *et al.*, "Smart grid can make Saudi Arabia megawatt exporter," in *Proc. Power Energy Eng. Conf. (APPEEC)*, Wuhan, China, 2011, pp. 1–4.

[10] Natural Resources Canada. (2011). *Energy Efficienc Trends in Canada, 1990 to 2008*. [Online]. Available: http://oee.nrcan.gc.ca/publications/statistics/trends10/chapter3.cfm?attr=0.

[11] T. Nghiem, M. Behl, R. Mangharam, and G. Pappas, "Green scheduling of control systems for peak demand reduction," in *Proc. 50th IEEE Conf. Control Decis.*, Orlando, FL, USA, 2011, pp. 5131–5136.

[12] A. Agnetis, G. de Pascale, P. Detti, and A. Vicino, "Load scheduling for household energy consumption optimization," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2364–2373, Dec. 2013.

[13] P. Chavali, P. Yang, and A. Nehorai, "A distributed algorithm of appliance scheduling for home energy management system," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 282–290, Jan. 2014.

[14] G. O'Brien and R. Rajagopal, "Scheduling non-preemptive deferrable loads," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 835–845, Mar. 2016.

[15] D. Setlhaoloa, X. Xia, and J. Zhang, "Optimal scheduling of household appliances for demand response," *Elect. Power Syst. Res.*, vol. 116, pp. 24–28, Nov. 2014.

[16] A. Subramanian *et al.*, "Real-time scheduling of deferrable electric loads," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, 2012, pp. 3643–3650.

[17] N. Yaagoubi and H. T. Mouftah, "User-aware game theoretic approach for demand management," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 716–725, Mar. 2015.

[18] S. Caron and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the smart grid," in *Proc. IEEE Smart Grids Commun.*, Gaithersburg, MD, USA, Oct. 2010, pp. 391–396.

[19] P.-D. Moroşan, R. Bourdais, D. Dumur, and J. Buisson, "Building temperature regulation using a distributed model predictive control," *Energy Build.*, vol. 42, no. 9, pp. 1445–1452, 2010.

[20] C. Chen, J. Wang, Y. Heo, and S. Kishore, "MPC-based appliance scheduling for residential building energy management controller," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1401–1410, Sep. 2013.

[21] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 570–582, Jun. 2015.

[22] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[23] F. Rahimi and A. Ipakchi, "Demand response as a market resource under the smart grid paradigm," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 82–88, Jun. 2010.

[24] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Autom. Control*, vol. 37, no. 11, pp. 1692–1708, Nov. 1992.

[25] R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems," *SIAM J. Control Optim.*, vol. 36, no. 2, pp. 488–541, 1998.

[26] W. M. Wonham, "Notes on control of discrete-event systems," Systems Control Group, Dept. Elect. Comput. Eng., Univ. Toronto, Tech. Rep. ECE 1636F, Jul. 2007.

[27] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inf. Sci.*, vol. 44, no. 3, pp. 173–198, 1988.

[28] G. T. Costanzo, "Demand side management in the smart grid," M.Sc. thesis, Dept. Elect. Eng., École Polytechnique de Montréal, Montreal, QC, Canada, 2011.

[29] B. Radunović and J.-Y. Le Boudec, "A unifie framework for max-min and min-max fairness with applications," *IEEE/ACM Trans. Network.*, vol. 15, no. 5, pp. 1073–1083, Oct. 2007.

[30] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data Networks*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[31] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Amsterdam, The Netherlands: Elsevier, 2004.

[32] G. T. Costanzo, F. Sossan, M. Marinelli, P. Bacher, and H. Madsen, "Grey-box modeling for system identificatio of household refrigerators: A step toward smart appliances," in *Proc. Int. Youth Conf. Energy*, Siófok, Hungary, 2013, pp. 1–5.

[33] S. H. Zad, *Discrete Event Control Kit*, Concordia Univ., Montreal, QC, Canada, 2013.

Waselul H. Sadid received the B.Sc. degree in electrical and electronic engineering from the Rajshahi University of Engineering and Technology, Rajshahi, Bangladesh, in 2000; the M.Sc. degree in computer science and engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2005; and the Ph.D. degree in electrical and computer engineering from Concordia University, Montreal, QC, Canada, in 2014.

He is currently an Industrial Postdoctoral Intern awarded by the Mitacs-Accelerate Graduate Research Internship Program at Fusion Energy Inc., Laval, QC, Canada, and École Polytechnique de Montréal, Montreal. His research interests include decentralized control of discrete-event systems, applications of game theory, robust control, and demand side energy management in smart buildings.

Saad A. Abobakr was born in Baniwaleed, Libya, in 1978. He received the B.Sc. degree in control engineering from the College of Electronic Technology, Baniwaleed, Libya, in 2001, and the M.Sc. degree in instrumentation and control engineering from the Academy of Graduate Studies, Tripoli, Libya, in 2008. He is currently pursuing the Ph.D. degree at École Polytechnique de Montréal, Montreal, QC, Canada. His research is oriented toward the application of model predictive control to demand response management in smart buildings, demand side energy management, building energy systems modeling and simulation, system identification and adaptive control.

Guchuan Zhu (M'07–SM'12) received the M.S. degree in electrical engineering from the Beijing Institute of Aeronautics and Astronautics, Beijing, China, in 1982; the Ph.D. degree in mathematics and control from École des Mines de Paris, Paris, France, in 1992; and the graduate Diploma degree in computer science from Concordia University, Montreal, QC, Canada, in 1999.

He is currently an Associate Professor with École Polytechnique de Montréal. His research interests include control of distributed parameter systems, nonlinear and robust control, and optimization with their applications to microsystems, aerospace systems, communication networks, and smart grid.