Reducing the Peak Power through Real-Time Scheduling Techniques in Cyber-Physical Energy Systems

Tullio Facchinetti
Computer Engineering and Systems Science
University of Pavia
Pavia, Italy
Email: tullio.facchinetti@unipv.it

Enrico Bini
RETIS Laboratory
Scuola Superiore Sant'Anna
Pisa, Italy
Email: e.bini@sssup.it

Marko Bertogna
RETIS Laboratory
Scuola Superiore Sant'Anna
Pisa, Italy
Email: marko@sssup.it

***Abstract*—This paper presents a method for applying real-time scheduling techniques to balance the power usage of electric loads in cyber-physical energy systems. The aim of the proposed approach is to achieve predictability of the activation of electric loads to guarantee an upper bound on the peak electric power consumption.**

The contribution of this paper encompasses several aspects. The relevance of balancing electric loads is discussed, motivating the use of real-time scheduling techniques to achieve predictability on electric-load management. We introduce the innovation of modeling the physical system as a set of periodically activated loads, that can be effectively managed by adequately adapting traditional real-time system models and scheduling algorithms, to guarantee an upper bound on the peak power consumption. For this purpose, we present a problem formulation based on linear programming, while a low-complexity heuristic is proposed to limit the complexity of the optimization process. Simulation results are presented to assess the performance of proposed methods.

***Keywords*-Cyber-Physical Energy Systems; Load Balancing; Real-Time; Modeling; Peak Load**

I. Introduction

Cyber-physical systems (CPSs) represent an emerging technology that aims to integrate embedded processing devices to monitor and control physical processes. Cyber-physical systems are intended to address critical applications operating in dynamic and uncertain environments, made by a high number of devices and characterized by complex relationships among components. Several factors can affect system operations, such as hardware and software failures, and partial knowledge of the system operating state. Example applications include: automotive, avionics and medical systems; critical infrastructure management, such as electric power and water resources; traffic control and safety; advanced robotics for manufacturing or telemedicine (see [1] for details on some specific systems).

While the current application design is based on the use of traditional embedded systems, which emphasizes the computational issues performed by embedded processing units, cyber-physical systems are more focused on the tight integration between physical and computational systems. This paper concentrates the attention on those systems dedicated to energy management, i.e., Cyber-Physical Energy Systems (CPESs) [2]. In such systems, the "physical" process is made by a network of electric devices that are controlled by a complex set of interconnected embedded systems.

The current technology trend is moving towards the automatic, distributed and coordinated control of electric devices. Some examples can be found in home and factory automation systems [3], large networks of electric cars [4], and automated energy supply and distribution for town and city districts organized in *smart grids* [5]. On the other hand, the diffusion of compact, flexible and low-cost embedded systems is making more practical and attractive the implementation of CPESs, since monitoring and control actions can be accurately applied on devices composing the considered physical system. Such embedded systems can be connected to build large distributed networks, thus being able to coordinate the actions on large systems.

The limited sources and the growing request of electric energy, together with the impact of power generation, transportation and usage on the environment and the eco-system, motivates the research on techniques to optimize the energy utilization in cyber-physical energy systems.

An "electrical system" can be defined as a set of electrical devices, or *loads*. Within the scope of this paper, a load is characterized by its power consumption, i.e., the maximum amount of power consumed when the load is active. Loads can be turned on/off depending on different conditions: the purpose of the device, the relationship with other devices, environmental conditions, and *time*.

Electrical systems may be composed of tens to thousands loads, where each device (or groups of them, which could be seen as a single logical device) must be driven in a timely fashion.

The balancing of energy utilization is fundamental for the efficient behavior of an electrical system [6], [7]. For this purpose, specific technical and economical approaches are used to control the distribution of power usage over time. One of the most widely adopted methods is the "peak-load pricing", which assigns higher prices to larger peak-load demands [8]. The most recent analysis of this pricing policy originates from economic research in the 60's and 70's. Peak-load pricing is often used by electricity suppliers and telephone utilities to enforce a limitation on the peak service demand, avoiding expensive over-dimensioned distribution infrastructures associated with high peak demands. In case of energy distribution, a supplier can only provide a finite amount of energy. This is due, among other factors, to the limited number of available sources for energy generation, or the bounded flow of energy that can be supported by the distribution infrastructure. Therefore, the price of the service is raised to discourage the resource usage under peak load conditions.

Peak load conditions, i.e., the simultaneous request of large electrical powers by many users, may cause severe issues such as the disruption of power provision, leading to technical and economic problems for both suppliers and users. Moreover, during peak load conditions, the cost of energy production may unpredictably increase in a short time frame due to the impossibility of generating enough energy to satisfy the request of customers. Therefore, energy providers — that must observe contractual obligations with their customers to supply electricity at pre-defined fixed prices — may experience a relevant financial burden. On the other hand, an adequate management of peak load conditions is desirable for energy utilities [9], so that an appropriate load management aiming at achieving predictable load conditions may lead to potential contractual benefits to the user. As a consequence, both energy providers and consumers are likely to be interested in load balancing and predictable energy consumption.

Given the aforementioned technical and economic issues, an efficient management of peak-load conditions has the following advantages:

1) the least efficient, i.e., the most expensive, power plants can be turned off if the peak power demand is guaranteed to remain under a given threshold;
2) the electric distribution infrastructure can be tailored for lower peak loads, with less technical issues and reduced costs;
3) the curve of power usage can be smoother and flatter, allowing the final users to have better pricing conditions on the free energy market, where pricing strategies are often driven by the peak-load pricing policy [8].

This work aims to apply real-time scheduling techniques to the management of loads in cyber-physical energy systems. The goal is to balance the total consumed power and the peak power consumption. The main expected advantage of this approach is to leverage the strong mathematical background of the real-time scheduling discipline for modeling and analyzing a physical system. On one hand, real-time scheduling algorithms can be used to predictably activate/deactivate electrical devices to guarantee the desired system features, in terms of timing constraints and energy consumption. On the other hand, the typical large size of cyber-physical energy systems will take advantage of efficient scheduling algorithms and analysis techniques to determine the system feasibility and properties.

It is worth noting that this paper does not deal with the architecture or the engineering of a cyber-physical energy system. The proposed approach must be intended as a viable modeling technique for the physical energy system, allowing the development of predictable and robust control strategies based on real-time scheduling methodologies. To the best of our knowledge, this is the first work addressing the application of real-time scheduling techniques to cyber-physical energy systems in order to balance the consumed power and to achieve a bound on the peak load. Therefore, the first part of the paper is dedicated to the description of the approach and of the potential contributions that real-time methodologies could bring to properly modeled CPESs.

A. *Paper organization*

The paper is organized as follows: a detailed explanation of the analogy between real-time computing systems and electrical systems is given in Section II, motivating the proposed approach with examples and possible scenarios. Section III introduces the system model, under which the theoretical results of Section IV are derived. Sections V and VI present, respectively, an optimal and an approximated method to reduce the peak load. The effectiveness of such methods is assessed in Section VII by means of simulations. Finally, Section VIII states our conclusions and outlines several possible directions for future enhancements of this work.

II. Electric loads as real-time tasks

This paper introduces the application of real-time analysis techniques to schedule the activation of electric devices in electrical networks. For this purpose, an analogy is established between real-time computing systems and cyber-physical energy systems.

Real-time scheduling allows managing the execution of tasks on processors under timing constraints. In more general terms, real-time scheduling can be seen as the discipline of allocating resources over time to a set of time-consuming tasks, so that given timing constraints are satisfied. However, in this more general formulation, resources may not necessarily be processors or computing devices. In fact, real-time scheduling techniques are also applied to communication systems, where real-time algorithms are used to schedule sets of messages over a communication channel [10]. In this case, an analogy holds between computing tasks and messages, as well as between processors and communication channels. The meaning of "available bandwidth" changes depending on the particular context, referring to the channel capacity in communication systems, and to processor's computing time in computing systems. Finally, timing constraints are enforced on the execution times in one case, and (typically) on message's end-to-end latency in the other. In other words, a real-time task must be guaranteed to terminate its execution before its deadline, while a message must be delivered to the receiver within the given time limit. This analogy allows extending to communication networks many results that have been originally developed for real-time computing systems, and vice versa. An example is given by the "real-time calculus" [11], a real-time extension of the network calculus. The above considerations lead to the opportunity of profitably applying real-time scheduling techniques, with suitable adaptations and extensions, to systems presenting similar analogies.

Electric devices are modeled as periodically activated tasks, with a bound on the total time that a load can remain active — thus consuming power — in each period. This bound recalls the Worst Case Execution Time (WCET) of a real-time task in computing systems. As for computing tasks, all the time properties of electric loads (periods, deadlines and activation time) must be selected according to their application requirements. Section II-A provides some examples of timing constraints related to specific electrical loads. Based on the system model, a priority-based scheduling algorithm can be applied to selectively activate/deactivate each device. The goal is to meet the timing constraints of each load, while guaranteeing an upper bound on the total instantaneous power consumed by the concurrent activation of electric components.

In the real-time systems literature, there is active research on power-aware scheduling strategies to save energy while achieving timing constraints. Such scheduling policies aim at reducing the power consumption using special features of modern electronic hardware, such as Dynamic Voltage Scaling (DVS) [12], [13]. As in those works, the model proposed in this paper associates a maximum consumed power to each electric device. However, some distinctions can be identified. First of all, we do not aim at directly reducing the overall energy required by the system. The objective is, instead, to determine a bound on the peak power consumption, and to predictably enforce this bound by scheduling electric devices activations in a timely manner.

Some recent works are addressing the real-time issues related with some special cases of cyber-physical energy systems. In [14], the authors propose a technique to improve the efficiency of batteries charge/discharge, for electric vehicles. However, this work is limited to batteries, while our approach is oriented at establishing a general framework for managing energy systems in a real-time manner. In [15], the authors aim at finding optimal schedules for microCHP (Combined Heat and Power) systems. The approach is based on global optimization through an Integer Linear Programming formulation of the problem. However, the application of the proposed method is strictly limited to offline optimization, while our technique can be applied online. Moreover, we introduce the novelty of modeling electric loads using real-time parameters, to allow the use of real-time techniques for scheduling the activation of loads. Finally, in [16] the authors describe a cyber-physical energy system as a set of components modeled as dynamical systems. While the modeling of electrical devices is more advanced than the one proposed in this paper (refined modeling is subject of future research in our framework), the goal is not related with achieving peak load constraints and, again, no real-time issues are considered.

A. Load modeling

This section provides informal examples of electric devices and applications that are suitable for being integrated in a real-time management system. Their relevant characteristics are described, outlining a possible modeling of their timing properties.

Household appliances: Typical household devices like ovens, washing machines, dryers, dishwashers, have each a peculiar duty cycle. The tighter the timing requirements — i.e., the closer the deadline to the maximum activation time — the more constraints are imposed on the scheduling algorithm, reducing the chances of finding a lower peak load. Anyway, a certain slack is usually available in the working cycles of household appliances, and programmable devices are already used to control the activation of electric loads depending on the energy prices in the stock market.As an example, these devices are used to control washing machines in domestic environments, where postponing by a few hours the time at which the laundry is ready does not cause any problem.

Lighting.: Consider the corridor lights of a building, that may need to be turned on in the evening, for example at 8:30pm, and turned off in the morning at 7:00am. In this case, no service interruption can be tolerated. During the active period, the total power consumption is the sum of power consumed by each lighting device, while in the rest of the time, the power consumption is negligible.

In this simple case, the load has a period of 24h, an active time of 10:30h, and a relative deadline equal to the active time. In this way, the load must be continuously scheduled at the beginning of the period, without allowing any "preemption" while the lights are switched on, as is expected from a lighting system. Clearly, this requirement has a negative impact on the level of concurrency of load activations. Electrical loads of this kind (i.e., with no activation slack) will lead to an increase in the number of concurrently active loads. Since no slack is available in the activation cycle, there is no way of reducing the impact of these loads on the resulting peak load. However, it is still possible to control the schedule of less interactive loads, so that they be activated when there is a smaller energy requirement.

III. System model

We consider a system composed of a set $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ of n independent *electric loads* that request to be turned on and off (or activated/deactivated), depending on their specific timing requirements. A load is said to be *active* when it is turned on, *inactive* otherwise.

The j-th request for activating the load λ_i happens at time $r_{i,j}$. The i-th load λ_i is modeled by the tuple (T_i, C_i, P_i), where

- T_i is the minimum separation between two consecutive requests of activation $r_{i,j}$, $r_{i,j+1}$ (as in the sporadic model for real-time computing tasks [17]). Hence,

$$\forall \lambda_i, \forall j \quad r_{i,j+1} \geq r_{i,j} + T_i \tag{1}$$

- C_i is the longest time the load λ_i can be active between two consecutive requests;
- P_i is the nominal power consumed by the load λ_i during its active time.

We define the *utilization* of λ_i as $U_i = C_i/T_i$. The *total utilization* of Λ is $U = \sum_{i=1}^{n} U_i$.

The load activity is controlled by a *load scheduler* that decides when each load is activated/deactivated. Formally, the scheduler assigns to each load λ_i a schedule that is modeled by the function $s_i(t)$

$$s_i(t) = \begin{cases} 1 & \lambda_i \text{ is active at } t \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

The *schedule of loads* is then given by $\mathcal{S} = \{s_1, \ldots, s_n\}$.

A schedule $\mathcal{S}$ is said to be *valid* if it assigns to each load λ_i an amount of activity time equal or larger than C_i between two consecutive requests. Formally,

$$\forall \lambda_i, \forall j \quad \int_{r_{i,j}}^{r_{i,j+1}} s_i(t)\, \mathrm{d}t \geq C_i \tag{3}$$

Notice that the equality will suffice in Equation 3 if traditional scheduling algorithms (such as EDF or RM) are used to generate the schedule.

For a given schedule $\mathcal{S}$, the actual power consumed by the load λ_i at time t is

$$p_i(t) = P_i\, s_i(t). \tag{4}$$

The overall actual power consumption $p(t)$ at time t is

$$p(t) = \sum_{i=1}^{n} p_i(t). \tag{5}$$

Finally, we define the *peak load* P of a set of loads Λ as the maximum instantaneous power consumption over the system lifetime

$$P = \max_{t \geq 0} p(t). \tag{6}$$

Given these hypothesis, we can formulate our problem as follows

$$\begin{aligned} &\text{minimize } P \\ &\text{subject to } \mathcal{S} \text{ being a valid schedule} \end{aligned} \tag{7}$$

Unfortunately, solving the problem in this wide formulation is very hard. In the next sections, we will show how to exploit well known real-time scheduling algorithms to find a suitable solution for this problem.

IV. RT scheduling algorithm for electric loads

We propose to use classic real-time scheduling algorithms, such as Rate Monotonic (RM) or Earliest Deadline First (EDF) [18], to schedule the loads in Λ. Specifically, each load can be considered as a task with computation time C_i and period (equal to the deadline) T_i. For example, when $U \leq 1$, the EDF scheduling algorithm can build a schedule $\mathcal{S}$ with the minimum possible peak power, that is $P = \max_i P_i$.

However, if $U > 1$ some loads must be contemporarily activated, leading to a possibly larger peak power consumption P. Hence, we suggest to partition the Λ load set into m disjoint sets Λ_j, $j = 1 \ldots, m$, that we call *scheduling groups*. Scheduling groups are determined such that their total utilization, defined as

$$U(\Lambda_j) = \sum_{\lambda_i \in \Lambda_j} U_i,$$

is smaller than or equal to 1. This property enables EDF to find a valid schedule within each scheduling group. The maximum peak in this case happens when the loads with the highest powers are contemporarily activated in all the scheduling groups. Notice that Equation (5), which is evaluated over all loads $\lambda_i, 1 \leq i \leq n$, could also be evaluated over all scheduling groups $\Lambda_i, 1 \leq i \leq m$, since in each scheduling group only one load is active at any given time t. An upper bound P^* on the peak load can be found considering the contemporary activation on all groups of the load with the largest power. Therefore,

$$P^* = \sum_{\Lambda_j} \max_{\lambda_i \in \Lambda_j} p_i. \tag{8}$$

It is worth noting that P^* represents an upper bound, but it is not tight, i.e., it could be overly pessimistic.

In this section, we provide some theoretical results related to the considered system model, and propose strategies to produce a valid schedule of a given set of loads, with the goal of reducing and bounding the peak load. We will present two scheduling algorithms with different complexities. One algorithm produces a smaller peak load, although it requires a large computational effort; the second one is simpler, although it could result in a larger peak load.

Before presenting the algorithms, we first state the following theoretical result on the minimum achievable peak load.

Theorem 1: For any load set Λ, no valid schedule can produce a peak load lower than

$$P^{\min} = \sum_{\lambda_i \in \Lambda} P_i U_i. \tag{9}$$

Proof: Assume, by contradiction, a load allocation for Λ grants a peak load $P < P^{\min}$. Let H be the least common multiple of all load periods $T_1, \ldots, T_n$. The overall energy consumed by Λ over H when all loads are synchronously activated at time $t = 0$, and then periodically activated as soon as possible, is

$$\sum_{i=1}^{n} \frac{H}{T_i} C_i P_i = H \sum_{i=1}^{n} U_i P_i.$$

Since the peak load is assumed to be equal to P, the overall energy consumed by Λ in H can not be greater than PH. Therefore,

$$H \sum_{i=1}^{n} U_i P_i \leq PH,$$

and,

$$\sum_{i=1}^{n} U_i P_i \leq P.$$

Using Equation (9), we get

$$P^{\min} \leq P.$$

leading to a contradiction. ■

V. Linear programming formulation

The problem of partitioning the set of loads as introduced in Section III, can be formalized as a *level packing* problem [19]. In level packing, a strip must accommodate a set of rectangles such that the total height is minimized. The peculiarity of level packing is that rectangles are partitioned in horizontal levels of decreasing height from the bottom to the top. In each level, items are packed from left to right by decreasing height, similarly to the arrangement of books within a bookshelf (see Figure 1).

Since the height of a level is equal to the leftmost rectangle, such a rectangle is said to *initialize* the level. The advantage of level packing is that a two-dimensional problem is transformed into a pair of one-dimensional problems, namely the packing of levels, and the packing of rectangles into levels.

In this paper, the level packing problem is solved using a Binary Integer Linear Programming (BILP) technique after a proper modeling of the problem, which brings to the introduction of suitable optimization variables.

Each load is modeled as a rectangle whose height corresponds to the power consumption p_i and width is determined by its utilization u_i. Without loss of generality, all loads are assumed to be sorted by decreasing power, namely $p_i \geq p_j \;\Leftrightarrow\; i \leq j$. In the worst case, there are n possible levels, one for each rectangle as the starting item. A set of n variables $y_i \in \{0, 1\}$ defines level initialization. There is one such variable for each load, being $y_i = 1$ if item i initializes level i, $y_i = 0$ otherwise. A level is labelled by the index of the item initializing it. The variables $x_{i,j}$ with $i \in \{1, \ldots, n-1\}$ and $j > i$ define the packing of item j when it does not initialize a level. The value $x_{i,j} = 1$ is set if item j is packed in level i, $x_{i,j} = 0$ otherwise.

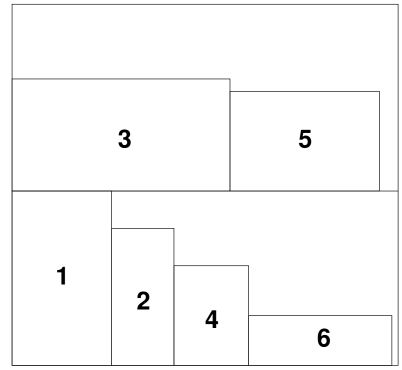

Figure 1. Example of level packing.

For example, in the case depicted in Figure 1, it holds $y_1 = y_3 = 1$, because only items 1 and 3 initialize a level, while $y_i = 0$ is set for all remaining items. The allocation of other rectangles to their respective levels is encoded in $x_{1,2} = x_{1,4} = x_{1,6} = 1$ and $x_{3,5} = 1$, with all other values being $x_{i,j} = 0$.

First of all, since each load can either initialize one level or it can be one of the rectangles following the initializer, the following constraint must hold:

$$y_j + \sum_{i=1}^{j-1} x_{i,j} = 1 \qquad \forall j = 1, \ldots, n \tag{10}$$

Notice that, thanks to the ordering of the rectangles by decreasing height, item j can be allocated as one of the non-initializing items only in the levels from 1 to $j-1$.

A second constraint arises from the maximum width of the resource. The value W is defined to be equal to the utilization upper bound that guarantees the schedulability of a load set. For example, if Earliest Deadline First (EDF) with implicit deadlines is used, then we set $W = 1$. Since the horizontal dimension is interpreted as utilization, then each level can not exceed the width W of the rectangle. Therefore, it holds

$$\sum_{j=i+1}^{n} u_j \, x_{i,j} \leq (W - u_i) \, y_i \qquad \forall i = 1, \ldots, n-1 \tag{11}$$

To enforce the consistency of the contraint given by Equation 11, notice that when level i does not exist ($y_i = 0$), then all $x_{i,j}$ are forced to 0 as well. The constraint specified by Equation 11 enforces the utilization based schedulability test. Therefore, it makes the proposed solution suitable for scheduling algorithms where feasibility can be evaluated by an utilization-based test. However, in [20], the authors propose the description of the EDF scheduling algorithm, where deadlines are less than periods, using a set of linear inequalities that could be used within the BILP framework. Therefore, the approach proposed in this paper can be easily extended to such system model.

The goal of the optimization approach based on BILP is to minimize the sum of the peak powers on each group, that is

$$\text{minimize} \quad \sum_{i=1}^{n} p_i \, y_i \tag{12}$$

The evaluation of the number of variables and constraints provides an estimate the problem complexity. In the proposed scheme, the number of y_i variables is n, because all rectangles may initialize one level. The $x_{i,j}$ variables are $\frac{n(n-1)}{2}$. Hence, the total number of variables is $\frac{n(n+1)}{2}$. Moreover, by counting the number of inequalities in Equations (10) and (11), we find that the number of constraints is $2n-1$.

VI. Load balancing heuristic

This section introduce a heuristic algorithm to address the problem of generating scheduling groups. Algorithm 1 shows the pseudo-code of the proposed method. The key point of the algorithm consists in sorting the global set of loads Λ in a descending order with respect to powers, such as $\lambda_i < \lambda_j \Leftrightarrow p_i > p_j$. The algorithm is essentially a first-fit bin-packing algorithm applied to the ordered set of loads. The λ_i load is inserted into the first scheduling group when the schedulability of the group is feasible. Otherwise, a new scheduling group is created and the current load is inserted into the newly created group.

The proposed technique recalls the RM-FFDU (Rate Monotonic First-Fit Decreasing Utilization) partitioning scheme for scheduling fixed priority real-time tasks on a multi-processor system [21], where bin-packing techniques are used to allocate tasks on processors. However, the mentioned previous work does not address the optimization of the total power consumption. Moreover, the key distinction is that in our method the ordering is made with respect to the value of load's consumed power, and utilization is not considered for this purpose.

Since no specific scheduling algorithm is assumed within each scheduling group, the feasibility test to be performed in Algorithm 1 is not specified, being dependent on the adopted scheduling policy. The complexity of the proposed method is therefore $O(\alpha \cdot n^2)$, where α represents the complexity of the feasibility test adopted. As an example, when using EDF with the associated utilization-based feasibility test, the complexity is $O(n^3)$.

Algorithm 1 The pseudo-code of the load balancing heuristic.

```
sort Λ in decreasing order of power
Λ_1 ... Λ_m are the scheduling groups
m = 1 is the initial number of scheduling groups
for all λ_i ∈ Λ do
    for j = 1 to m do
        if λ_i is schedulable in Λ_j then
            add λ_i to Λ_j
            goto end-loop
        end if
    end for
    create a new scheduling group Λ_{m+1}
    add λ_i to Λ_{m+1}
    m = m + 1
    :end-loop
end for
```

VII. Experimental assessment

This section reports some results obtained by generating random electric loads while changing some of the most relevant parameters. The goal is to investigate, under different circumstances, the reduction of the peak load achieved both by solving the optimization problem and using the heuristic approach.

The peak load achievable using the proposed schemes is compared with the worst possible case where all the loads are active at the same time:

$$P^{\max} = \sum_{i=1}^{n} p_i.$$

The parameters that have been taken into account in the experiments are: the total number of loads n, the total utilization of the set of loads U, and the range for the power assigned to the loads. Given those parameters, the value of each load is randomly generated using the algorithm UUniFast presented in [22].

Figure 2 shows the efficiency of different approaches with respect to $P^{\max}$, as a function of the ratio between the total utilization U and the number of loads n. The efficiency η is calculated as

$$\eta = \frac{P^{\max} - P^{\text{meth}}}{P^{\max}} \cdot 100$$

where P^{meth} represents the peak load achieved by the given method: lower bound, LP and heuristic refer, respectively, to the peak load obtained from Theorem 1, the method of Section V and the approximated approach of Algorithm 1. The value of the peak load used to calculate the efficiency is an aggregated value obtained by averaging the outcome of thousands of simulation runs. The number of loads assumes values in the range $[2, 30]$, while the total utilization ranges

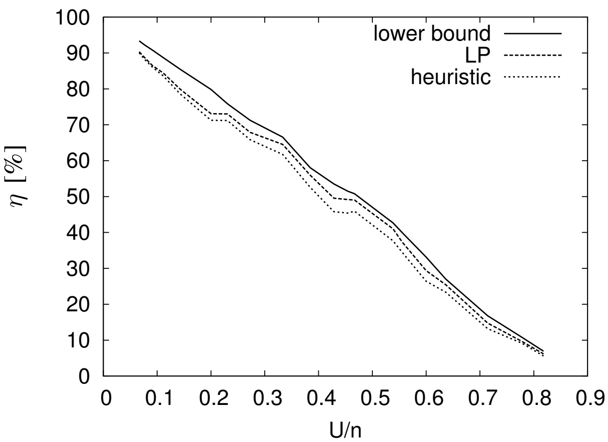

Figure 2. Efficiency of different approaches as a function of the average utilization.

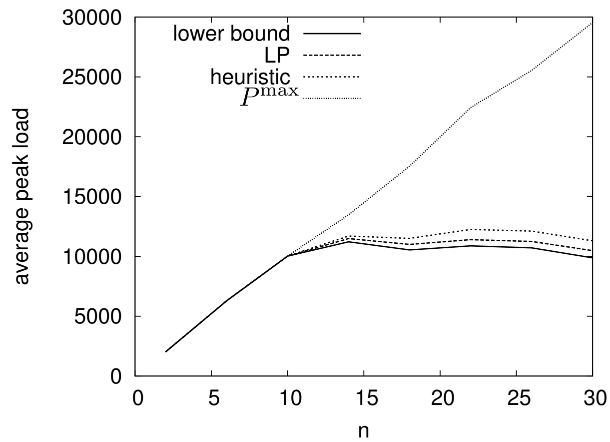

Figure 3. Average peak load obtained by the different techniques as a function of the number of loads n, with $U = 10$.

in the interval $[2, 18]$. The nominal power of each load is randomly selected in the range $[20, 2000]$, which is a reasonable range for typical household appliances.

The results of Figure 2 show that for lower values of the U/n ratio, i.e., having a high number of loads and a small total utilization, the proposed methods allow reducing the peak load up to more than 90% with respect to $P^{\max}$. Therefore, the explicit control on load activations brings to a remarkable improvement in comparison to the absence of control actions. When the U/n ratio tends to 1, the benefits of using a scheduling approach disappear. This is due to the fact that, when U tends to n, the load generation algorithm presented in [22] generates an increasing number of loads having $U_i = 1$ in order to obtain the desired total utilization. In this situation, the loads cannot be efficiently aggregated into scheduling groups, so that each created scheduling group contains just a few loads (only one load in the worst case). Therefore, the number of scheduling groups tends to n and the peak load achievable by all methods converges to the maximum possible peak load $P^{\max}$, leading to $\eta \to 0$. Notice that when $U \geq n$, it holds $\eta = 0$.

Figure 3 shows the average peak load obtained by the different techniques as a function of the number of loads n when the total utilization is constant ($U = 10$). It can be noticed that, when $U \leq 10$, the peak load achieved by the optimized methods can not be better than $P^{\max}$ for the same reason above: every load λ_i is generated with $U_i = 1$, and thus there is no opportunity to apply the scheduling of loads since each scheduling group contains exactly one load. When $U > 10$, the optimized methods guarantee an improvement that increases with n, accordingly with the results presented in Figure 2. Moreover, Figure 3 shows that the peak load achieved by the heuristic method is very close to the peak load guaranteed by the Linear Programming formulation which, in turn, is rather close to the lower bound

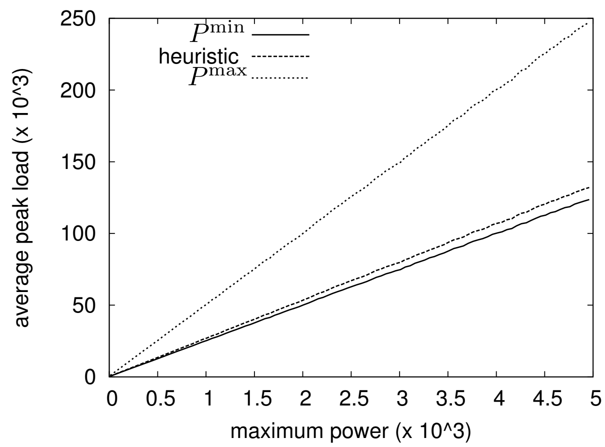

Figure 4. Average peak load obtained by the heuristic method as a function of the maximum possible power for each load; we considered $n = 100$, $U = 50$ and the minimum possible power equal to 10.

P^{min} imposed by Theorem 1. This characteristic behavior has been steadily detected throughout all experiments.

Finally, Figure 4 shows the average peak load obtained by the heuristic method as a function of the maximum possible power for each load. In this experiment we considered $n = 100$, $U = 50$ and the minimum possible power equal to 10. Similarly to the previous results, a noticeable decrease of the peak load is achieved by the heuristic with respect to $P^{\max}$. This improvement is independent from the range in which the power is selected for each load. Moreover, the solution found by the heuristic is relatively close to the lower bound $P^{\min}$.

VIII. Conclusions and future works

This paper presented a methodology for modeling the physical system of a cyber-physical energy system as periodic activities that can be scheduled by adapting traditional real-time scheduling algorithms. The goal of the proposed approach is to limit the peak of power consumption, which is a desirable feature for both the user and the energy provider.

To the best of our knowledge, this is the first attempt of using real-time scheduling techniques to organize the activation of electric loads in a cyber-physical energy system. In this paper, a number of simplifying assumptions have been made, such as considering periodic activations only, time-invariant load states, etc. Several improvements and refinements to the proposed model are thus possible: accounting for event-driven (aperiodic) load activations; an optimal scheduling strategy that would consider the interaction among loads of different groups (i.e., a global scheduling of loads); more accurate modeling of specific devices, considering different working modes, i.e. with different power consumption (full power, power saving mode, standby, etc.), time-varying load states, or accounting for the cost of "context switches", since switching on and off an electric motor has a cost that, in the long term, may shorten its life cycle. All those topics will be subject of future research.

References

[1] J. A. Stankovic, I. Lee, A. Mok, and R. Rajkumar, "Opportunities and obligations for physical computing systems," *Computer*, vol. 38, no. 11, pp. 23–31, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1541875

[2] S. T. Morris, A. Srivastava, B. Reaves, S. A. K. Pavurapu, R. Vaughn, W. McGrew, and Y. Dandass, "Engineering future cyber-physical energy systems: Challenges, research needs, and roadmap," in *Proceedings of the IEEE North American Power Symposium*, October 2009.

[3] "HomePlug homepage," 2009, http://www.homeplug.org/.

[4] "Betterplace project homepage," 2009, http://www.betterplace.com/.

[5] R. Brown, "Impact of smart grid on distribution system design," in *Proceedings of the IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, July 2008, pp. 1–4.

[6] A. I. Cohen, "An optimization method for load management scheduling," *IEEE Transactions on Power Systems*, vol. 3, no. 2, May 1988.

[7] H. Hermanns and H. Wiechmann, "Future design challenges for electric energy supply," in *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, September 2009.

[8] M. A. Crew, C. S. Fernando, and P. R. Kleindorfer, "The theory of peak-load pricing: A survey," *Journal of Regulatory Economics*, vol. 8, no. 3, pp. 215–248, November 1995. [Online]. Available: http://www.springerlink.com/content/q0l615457743lv71/

[9] D. Du Bois, "Time of use electricity billing: How puget sound energy reduced peak power demands (case study)," *Energy Priorities Magazine*, February 2006.

[10] P. Pedreiras, P. Gai, L. Almeida, and G. Buttazzo, "FTT-Ethernet: A flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 3, August 2005.

[11] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *International Symposium on Circuits and Systems (ISCAS)*, vol. 4, Geneva, Switzerland, March 2000, pp. 101–104.

[12] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proceedings of the* 18^{th} *ACM Symposium on Operating System Principles*, Banff, Canada, Oct. 2001, pp. 89–102.

[13] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez, "Power-aware scheduling for periodic real-time tasks," *IEEE Transactions on Computers*, vol. 53, no. 5, pp. 584–600, May 2004.

[14] H. Kim and K. Shin, "Scheduling of battery charge, discharge, and rest," in *Proceedings of the 30th IEEE Real-Time Systems Symposium (RTSS)*, December 2009.

[15] M. Bosman, V. Bakker, A. Molderink, J. Hurink, and G. Smit, "The microCHP scheduling problem," in *Proceedings of the Second Global Conference on Power Control and Optimization (PCO)*. Springer Verlag, June 2009, p. 8.

[16] M. D. Ilic, L. Xie, U. A. Khan, and J. M. F. Moura, "Modeling future cyber-physical energy systems," in *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, July 2008, pp. 1–9.

[17] S. K. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in *Proceedings of the* 11^{th} *IEEE Real-Time Systems Symposium*, Lake Buena Vista (FL), U.S.A., Dec. 1990, pp. 182–190.

[18] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46–61, January 1973.

[19] A. Lodi, S. Martello, and D. Vigo, "Models and bounds for two-dimensional level packing problems," *Journal of Combinatorial Optimization*, vol. 8, no. 3, pp. 363–379, 2004.

[20] S. Baruah and E. Bini, "Partitioned scheduling of sporadic task systems: an ilp-based approach," in *Conference on Design and Architectures for Signal and Image Processing*, Bruxelles, Belgium, Nov. 2008.

[21] Y. Oh and S. Son, "Allocating fixed-priority periodic tasks on multiprocessor systems," *Real-Time Systems*, vol. 9, no. 3, pp. 207–239, 1995.

[22] E. Bini and G. C. Buttazzo, "Biasing effects in schedulability measures," in *Proceedings of the* 16^{th} *Euromicro Conference on Real-Time Systems*, Catania, Italy, Jun. 2004, pp. 196–203.