REAL-TIME SCHEDULING FOR INDUSTRIAL LOAD MANAGEMENT

Marco L. Della Vedova and Tullio Facchinetti

Robotics Lab., Dpt. of Industrial and Information Engineering
University of Pavia, via Ferrata 1, 27100 Pavia, Italy
{marco.dellavedova, tullio.facchinetti}@unipv.it

ABSTRACT

A new approach based on real-time scheduling has been recently proposed for the management of electric loads composing a power system with the goal to reduce the peak load of power consumption. The novel idea is to model the power system in terms of timing parameters traditionally adopted in the field of real-time computing systems. In this way, the activation/deactivation of devices can be managed using classical scheduling algorithms developed for executing processing tasks in real-time systems. To prove the effectiveness of the proposed methodology, this paper shows its application to an industrial plant. Electric loads are modeled using timing parameters and a real-time scheduler is used to coordinate their activation. The coordinated management achieves the peak load reduction while meeting given constraints on the industrial process under control. The paper presents the modeling approach of the industrial process. The behavior of the control method is assessed by a simulated example.

***Index Terms*—** Modeling, Smart Grid, Real time systems, Power system control, Scheduling, Load Shedding.

1. INTRODUCTION

The *Smart Grid* is conceived as an automation infrastructure for the coordinated management of power systems [1]. The key characteristic of the Smart Grid is the integration of a digital communication infrastructure to support new features of power generation, distribution and usage. One important issue, with a potential large positive economic impact on power generation, distribution and usage, is the predictable control of power loads to achieve load balancing and peak load reduction and smoothing.

The goal of this paper is to prove the suitability of an innovative method based on real-time scheduling to manage a set of electric loads to limit the peak load of power usage. The modeling and control approach adopted in this paper is based on the definition of Real-Time Physical System (RTPS) proposed in [2]. The background idea is to model electric loads as a set of periodically activated tasks. A load is characterized by an activation period and activation time. The value of timing parameters are selected to cope with the physical process associated with the load itself. In a simple example, a refrigerator may need to be activated for 5 minutes every 35 minutes to keep the temperature within a predefined range. After the modeling, classical scheduling algorithms traditionally used in the field of real-time computing systems can be used to coordinate the activation of loads. The scheduler generates a predictable pattern of load activation, and achieves the chance to dynamically adapt the schedule to variable working conditions. Clearly, the selection of proper values of timing parameters needs to keep into account system dynamics and the influence of external variables, as well as the characteristics of the adopted scheduling algorithm.

RTPSs represent a special case of hybrid dynamic system. A RTPS is characterized by a discrete and a continuous behavior. The discrete behavior is related to the real-time schedule, while the continuous one refers to the physical values whose variation is affected by the schedule. In other words, the schedule defines when a load shall be turned on or off. A physical variable behaves accordingly to the on/off state of the load. Recalling the example of the refrigerator, the physical variable associated with the load is the internal temperature. The temperature will increase when the device is turned off, while it will decrease otherwise. RTPSs can easily integrate different types of loads, physical processes and constraints, once they have been modeled in terms of timing parameters. For this reason, RTPS represent a general framework for modeling power systems suitable for the regulation of peak power demand. The goal of this paper is to show the potential of RTPS to model and control a physical industrial process with the goal of limiting the peak load of electric power usage.

The system considered in this paper is made by a set of electric devices composing an industrial plant, where the peak load of electric power consumption must be limited for economic purposes. The physical industrial process is inspired to the one described in [3]. In that paper, the authors study the problem of controlling an industrial process using an optimization method to schedule the activation of electrical devices composing the plant. The schedule is generated offline to reduce the peak load of power usage in order to obtain economic benefits due to electricity pricing policies offered by the energy provider. The solution proposed in [3] is to solve a constrained optimization problem using integer linear programming. The paper provides a detailed description of the physical process to control. Several typical parameters of a production plant are considered, such as production capacity and sequential constraints. The detailed description made in [3] is leveraged in this paper to build the necessary system model to elaborate the real-time control approach.

2. REAL-TIME MODELING OF A POWER SYSTEM

The innovative approach adopted in this paper is to model electric loads of a power system as a set of controllable real-time tasks described by timing parameters. The modeling effort allows to apply existing real-time scheduling algorithms to manage the concurrent activation of different loads. In practice, an analogy is drawn between real-time computing systems and power systems. Such analogy allows to use existing modeling, scheduling and analysis methodologies studied in real-time systems for the management of electric loads. Electric loads are modeled as real-time processing tasks executed on one or more computing processors. A load consumes a known amount of electric power while active, and no power when inactive. The goal of a real-time scheduling algorithm is to assign a limited number of processors to a set of tasks under timing constraints. In the context of power systems, the use of a real-time scheduling algorithm will limit the number of simultaneously activated loads, thus limiting the peak load.

This paper adopts the periodic task model [4] to represent electric loads as periodically activated tasks. A period T and an activation time C are associated with each load. This means that a load can be turned on for at most C every T time units. In this way, a load remains active for a fraction $U = C/T$ of the time. The value of periods and activation times are selected to cope with constraints imposed by the underlying physical process. Based on this model, a real-time scheduling algorithm can generate a predictable pattern of device activations/deactivations, reducing the peak load of consumed power by limiting the simultaneous activation of electric devices.

3. RELATED WORKS

The application of real-time scheduling techniques to optimize the peak load have been introduced in [5]. The paper proposes an optimization method to solve a bin-packing problem related with the simultaneous activation of several loads, without considering physical variables associated to each load. In [2], loads associated with physical variables having exponential behaviors in the time domain are considered. Loads are independent each other. Some interesting theoretical features have been proved in [6]. In such paper, a closed-loop control strategy is based on real-time scheduling, while timing parameters are described by statistical distribution to consider errors and modeling inaccuracies. A statistical assessment of real-time scheduling techniques applied to power systems is provided in [7] by comparing with absence of explicit load control. In [8] real-time techniques are used to model and schedule the opening of valves in air compressed systems to reduce the total energy required for air compression. A similar approach is discussed in [9], where the "green scheduling" problem is defined as the peak load reduction problem faced by load scheduling. Such work proposes an alternative solution to the problem with respect to [2] and [5].

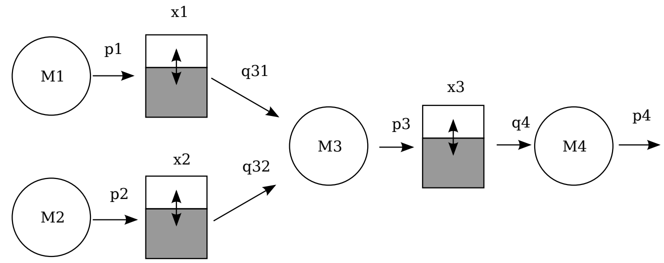

Fig. 1. Example of an industrial plant made by 4 loads (circles) and 3 buffers. Arrows indicate the flow of produced/consumed goods.

On the other hand, several approaches are presented in the literature on power systems to limit the peak load. Techniques based on fuzzy logic [10], self-organizing agents [11], and game theory [12] have been proposed, as well as other methods based on artificial intelligence as expert systems [13]. However, no mention to formal properties of proposed approaches is made in those papers. Moreover, they do not address the explicit modeling of physical constraints.

In this paper, the discussion is inspired by [3], which discusses the peak load optimization in an industrial process under different kinds of constraints. There are other works describing peak load reduction approaches in large distributed installations of power-consuming devices by means of remote direct load control policies [14, 15]. However, those works do not take into account the underlying physical process, but only the consumed power of loads.

4. SYSTEM MODEL

The system is composed by a set $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ of n machines. Each machine performs a specific operation in the production plant. In the scope of this paper, a machine is an electric load that can be turned on and off by the process controller. A load is said to be *active* when it is turned on, *inactive* otherwise.

The operation of a machine is modeled such that the machine uses some goods taken from one or more input buffers, and it produces goods that are stored in an output buffer. Each active machine λ_j produces (respectively, consumes) goods at rate $p_{j,i}$, storing such goods in (respectively, bringing goods from) the i-th buffer. The value $p_{j,i} > 0$ indicates an output flow of goods, while $p_{j,i} < 0$ indicates an input flow. The machine does not produce and consume any goods when inactive. The i-th buffer is characterized by its current level $x_i(t)$ and a maximum storage capacity $x_i^{\max}$. Figure 1 shows an example of modeled system, where machines are represented by circles and buffers by rectangles. The graphical representation also shows the input/output relationships among loads and buffers.

The system dynamics can be described in the continuous time domain by the following LTI (Linear-Time Invariant) system:

$$\begin{aligned} \dot{x}(t) &= Bs(t) \\ x(0) &= x_0 \end{aligned} \tag{1}$$

where:

- $t \in \mathbb{R}^+$ is the continuous time;
- $x(t) = [x_1(t) \ldots x_n(t)]' \in \mathbb{R}^n$ is the column vector of the n state variables, which are the buffer levels;
- $x_0 \in \mathbb{R}^n$ is the initial state of the system;
- $s(t) \in \mathbb{B}^m$ is a vector, whose elements $s_j(t)$ are the activity state of machine λ_j: $s_j(t) = 0$ if the λ_j machine is OFF at time t, $s_j(t) = 1$ if it is ON.
- $s : \mathbb{R}^+ \rightarrow \mathbb{B}^m$ is the switching signal or schedule;
- $B \in \mathbb{R}^{n\times m}$ is a matrix, whose element $b_{i,j}$ represents the production/consumption rate of the λ_j machine on the i-th buffer. The algebraic sign of the $b_{i,j}$ element discriminates between production and consumption: a production corresponds to a positive value, while a negative value indicates a consumption. If $b_{i,j} = 0$ then the i-th buffer is not affected by the actions of the λ_j machine. For example, the matrix B for the system represented in Figure 1 takes the following values:

$$B = \begin{bmatrix} p_1 & 0 & -p_{13} & 0 \\ 0 & p_2 & -p_{23} & 0 \\ 0 & 0 & p_3 & -p_4 \end{bmatrix}$$

Alternatively, in discrete time the system takes the following form:

$$x(t + 1) = x(t) + Bs(t) \tag{2}$$

The load λ_j consumes a given amount of electric power P_j when active, no power otherwise. Formally:

$$p_j(t) = \begin{cases} P_j & \text{if } s_j(t) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

4.1. Constraints and requirements

In [3], a set of constraints are introduced to enforce the correct behavior of the industrial process.

The *Production Constraints* is required to guarantee a given minimum output of the final product during the system lifetime. The Production Constraint can be expressed as follows:

$$\int_0^H q_i s_i(t)\ \mathrm{d}t \geq Q_i \quad \forall \lambda_i \in \Lambda^{\text{out}} \tag{4}$$

where $\Lambda^{\text{out}} \subset \Lambda$ is the subset of all machines whose production is not formally used to fill a buffer but it represents the final product of the industrial process.

The *Storage Constraint* encapsulates the requirement, for each buffer level, to not exceed the maximum capacity of the buffer itself.

$$x(t) \in \mathcal{X} \quad \forall t \geq 0 \tag{5}$$

where $\mathcal{X} = [x^{\min},\ x^{\max}] \subset \mathbb{R}^n$ is a convex set, which represents the constraint.

Process flow constraint: for satisfactory operation of process machines, a certain minimum amount of material needs to be maintaned in the buffers. This reserve allows the machine served by a buffer to always have an input material available.

$$x_i^{\min} > 0 \quad \forall i \tag{6}$$

5. MODELING USING REAL-TIME PARAMETERS

To exploit a real-time scheduling algorithm to manage the concurrent activation of loads, the physical process requires to be modeled in terms of timing parameters. In this paper the periodic task model is adopted to describe the physical process [4]. In the periodic task model, a task (corresponding to a load in this paper) is described by the tuple $\lambda_j = (C_j, D_j, T_j)$. The C_j parameter is said Worst-Case Computation Time (WCET), while T_j is the activation *period* and D_j is the relative deadline. The meaning of such parameters is that a new activation of λ_j is triggered at every time $t = kT_j$, for $k = 0, 1, \ldots$. Moreover, load λ_j is allowed to stay active for at most C_j time units within the interval $[kT_j, kT_j + D_j]$, for every k. Therefore, a load generates an infinite sequence of instances (or *jobs*) where the $(k + 1)$-th instance is ready for the activation T_j time units after the k-th instance. An important figure in periodic systems is the task utilization U_j, defined as

$$U_j = C_j/T_j \tag{7}$$

which is the maximum percentage of time that a task is active. The total system utilization is defined as

$$U = \sum_{j=1}^{n} U_j \tag{8}$$

The total utilization U is very important since it is used to derive relevant system properties, as it will be described later.

It is worth to note that in traditional real-time systems the value of C_j is the maximum amount of time that a task can run in each period, meaning that it may happen that a task will run for less time than C_j. A characteristic of a periodic real-time system is that it suffices to evaluate the properties of a schedule until the time $H = LCM(T_1, \ldots, T_n)$, i.e., until the Least Common Multiple of all periods. The H quantity is often referred as *hyper-period* in the traditional real-time scheduling theory.

5.1. Load scheduling

Given above definitions, the class of *valid schedules* can be defined as follows:

$$\mathcal{S}_{U,T} = \{s : \mathbb{R}^+ \to \mathbb{B} \text{ s.t. } \int_{kT}^{kT+D} s(t)\ \mathrm{d}t = UT\ \forall k \in \mathbb{N} \geq 0\} \tag{9}$$

A real-time scheduling algorithm will generate a valid schedule for each load, allowing to satisfy the following requirement:

$$s_i \in \mathcal{S}_{U_i,T_i} \quad \forall i \tag{10}$$

An example of schedule generated for 3 tasks is depicted in Figure 2. The total utilization is 0.89. The adopted scheduling algorithm is Earliest Deadline First (EDF) [4]. In each time slot, the load which is assigned the highest priority is scheduled for the activation. At any time t, the scheduling policy of EDF assigns the highest priority to the load having the closest absolute deadline d_j to the time t, where d_j is set at the beginning of the k-th period as $d_j = kT_j + D_j$. EDF is known to be optimal in the class of dynamic priority scheduling algorithms for uniprocessors [16]. This means that EDF can meet the timing constraints of every task/load when proper conditions are satisfied. For example, when $D_j = T_j$ for all loads, EDF guarantees that only one load will be active at any given time instant if and only if the condition $U \leq 1$ holds. The condition or procedure that allows to determine whether a scheduling algorithm is able to meet timing constraints of all tasks is said *schedulability test*. When $D_j \neq T_j$ for some j, the schedulability test of EDF becomes slightly more complicated, but still being fully affordable thanks to the low computational complexity [17]. On the other hand, when the number of loads increases, it may be impossible to guarantee that *only one load* will be active at any given time. In this case, however, scheduling policies developed in the multiprocessor real-time scheduling domain can be used, as done in [5].

When the schedulability test is not satisfied, in traditional real-time computing systems this means that the set of tasks can not be executed on the considered processing platform without violating the timing constraints of one or more tasks. For instance, the number of processors may be insufficient to suitably execute the given task set. In the considered domain of power systems, there is no equivalent concept of "processors". Therefore, in principle, there are no constraints on the maximum number of loads that can be activated at the same time. In practice, the purpose of the scheduling policy is to limit as much as possible the number of active loads at any given time, in order to reduce the peak load of power consumption. Clearly, the peak load reduction must cope with the physical process requirements. In other words, a load can not be switched off if it is essential for the proper system behavior.

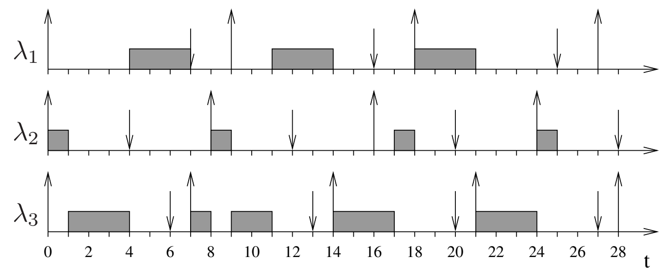

Fig. 2. Example of schedule generated for 3 loads using the EDF scheduling algorithm. Load parameters are $\lambda_1 = (9, 7, 3)$, $\lambda_2 = (8, 4, 1)$, $\lambda_3 = (7, 6, 3)$. Utilizations are 0.33, 0.12 and 0.42 respectively. Up arrows indicate activation times at each period, while down arrows indicate the deadline of respective job.

6. THEORETICAL RESULTS

In this section a number of interesting theoretical results will be derives regarding the considered system model and the adopted control approach based on real-time scheduling.

6.1. Periodicity of the schedule

We define $\overline{U} = (U_1, \ldots, U_n)'$ as the column vector composed by the utilizations of the n machines.

Theorem 1. *If the vector $\overline{U}$ is in the null-space of the matrix B:*

$$B\overline{U} = 0 \tag{11}$$

then the system (1) *is stable and at the hyper-period H it holds:*

$$x(H) = x(0) = x_0 \tag{12}$$

Proof. The solution of the ordinary differential equation describing the system's dynamics , obtained by integrating (1), is:

$$x(t) = B\int_0^t s(\tau)\mathrm{d}\tau + x_0 \tag{13}$$

Since H is by definition a multiple of every T_i, from (9) can be derived:

$$\int_0^H s(t)\mathrm{d}t = \overline{U}H \tag{14}$$

Combining (13) and (14), it follows: $x(H) = B\overline{U}H + x_0$. When the condition in the theorem's hypothesis (11) holds, $x(H) = x_0$ follows. □

Theorem 1 states that, for some specific values of the utilization of all loads, the level of each buffer $x_i(t)$ at time $t = H$ will be equal to the initial level $x_i(0)$ of that buffer. This results has an important impact on the analysis of the system. Assuming that all utilization are set as given by Theorem 1, the analysis of the system for an arbitrary lifetime extension can be restricted to the analysis of the system behavior in one hyper-period. In other words, despite the industrial

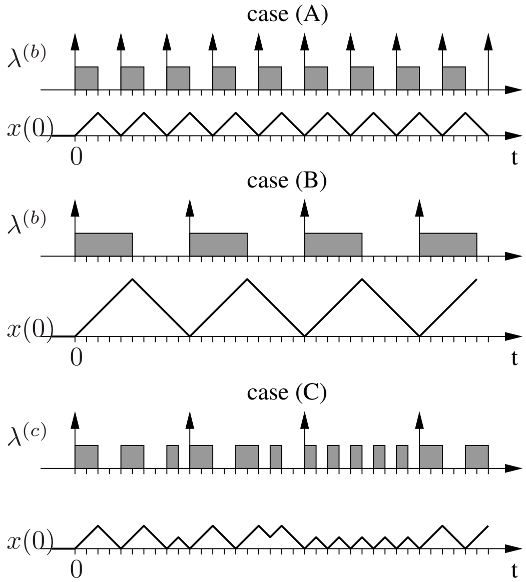

Fig. 3. Example of the effect of period selection for a load having a given utilization. The considered load λ has the same utilization $U = 0.5$ in all cases. However, the different chosen values for the period in the two cases (A) and (B) determine a noticeable difference in the worst case variation of the buffer level with respect to $x(0)$. Case (C) has the same period of case (B), but due to activity interruptions (preemptions), the maximum variation is less than in the worst-case.

process may work for an arbitrary time interval $t^{\max} \gg H$, all relevant system properties can be derived by studying the system in the time interval $[0, H]$.

6.2. Impact of period values on the variation of $x(t)$

Consider a load λ supplying good to a buffer (subscripts are suppressed for clarity). The machine has its own filling ratio p, and has an assigned utilization U. The value of U can be assigned with the result provided by Theorem 1. It is worth to note that, for any value U, there is an infinite set of pair (C, T) that can be assigned to obtain $U = C/T$. Therefore, the smaller the period T, the lower the maximum variation of $x(t)$ with respect to $x(0)$ in the worst-case. The behavior is shown in Figure 3. Note that we are considering the worst-case of the maximum variation of $x(t)$. In fact, in case (C) of Figure 3, the maximum variation is less than the worst case despite $\lambda^{(b)}$ and $\lambda^{(c)}$ have the same utilization and period. This is due to the fact that $\lambda^{(c)}$ is interrupted several times (preempted) during each period, which limits the maximum variation of $x(t)$.

The reminder of section is dedicated to put into relationship the selection of timing parameters (C and T) the maximum variation of $x(t)$ in the worst case. This is very important in order to determine whether a possible selection of timing parameters may violate the constraint on the maximum level of a buffer (Storage Constraint) or on the minimum level of the buffer (Process Flow Constraint).

Consider two machines λ_{in} (in = input) and λ_{out} (out = output). The two machines respectively supply and consume the goods of the same buffer.

The smaller the periods T_{in} and T_{out}, the lower the maximum variation of $x(t)$ in the worst case, i.e., when s_{in} and s_{out} are generated by the scheduler such that they produce the maximum variation of $x(t)$ with respect to $x(0)$.

$$\max_{s_i, s_o \in \mathcal{S}} \max_{t \geq 0} x(t) = f\left(T_i, T_o, x_0\right) \tag{15}$$

In the following, a bound to the maximum variation of a buffer level $x(t)$ will be derived.

Let's denote with M the least common multiple between T_{in} and T_{out}, i.e., $M = \text{LCM}(T_{in}, T_{out})$. Suppose, without lack of generality, that $T_{in} < T_{out}$. Moreover, the first activation of λ_{out} will happen at the same time as λ_{in}. We denote with Δ_{in} the increment imposed to $x(t)$ determined by λ_{in} within every period T_{in} in absence of any decrement. Similarly, Δ_{out} denotes the decrement imposed to $x(t)$ determined by λ_{out} within every period T_{out} in absence of any increment.

We will denote with $\Delta^{\max}$ the maximum positive variation with respect to x_0, i.e., $\max_{0 \leq t \leq M} x(t) = x_0 + \Delta^{\max}$. Similarly, we will denote with $\Delta^{\min}$ the maximum negative variation with respect to x_0, i.e., $\min_{0 \leq t \leq M} x(t) = x_0 - \Delta^{\min}$. The following results allow to reduce the complexity to calculate $\Delta^{\min}$ and $\Delta^{\max}$.

For every $t = kT_{out}$, being $k = 1, 2, \ldots, \frac{M}{T_{out}}$, the following shall be evaluated:

$$\Delta^{\max} = \max_{t} \left\lceil \frac{t}{T_{in}} \right\rceil - (k-1)\Delta_{out} \tag{16}$$

For $t = kT_{out}$, being $k = 0, 1, \ldots, (\frac{M}{T_{out}} - 1)$, the following is to be evaluated:

$$\Delta^{\min} = \min_{t} \left\lfloor \frac{t}{T_{out}} \right\rfloor - (k+1)\Delta_{out} \tag{17}$$

Finally, by imposing the constraints it holds:

$$\begin{cases} x^{\max} \geq x_0 + \Delta^{\max} \\ x^{\min} \leq x_0 + \Delta^{\min} \end{cases} \tag{18}$$

Therefore, suitable values for T_{in} and T_{out} can be obtained by solving (18).

7. SIMULATION EXAMPLE

In this section is provided an example of industrial load management as proof of concept for the scheduling methodology presented in this paper. Figure 4 shows the schedule of the four machines composing the industrial plant depicted in Figure 1. The activity of the machines, together with the levels of buffers and the overall power consumption, is reported for a time horizon of 12 hours, which is twice the hyper-period H for this system.

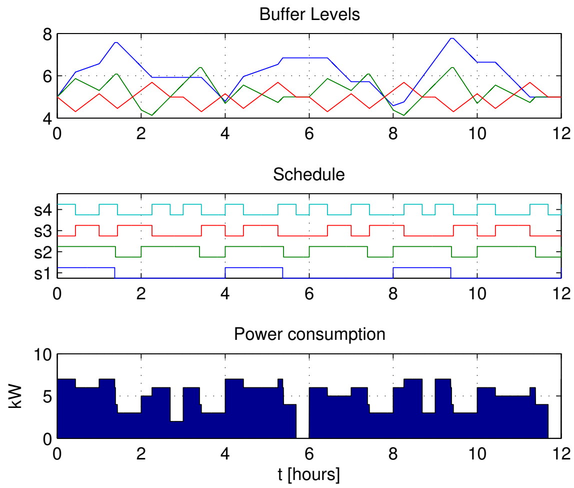

Fig. 4. Example of the scheduling of the four loads composing the industrial process of Figure 1.

The following values have been used in the simulation: nominal power of the machines $P = [1, 2, 3, 4]$ kW; production rates $p_{1,1} = 2.7$, $p_{2,2} = 2$, $p_{3,3} = 1.5$ and $p_{4,4} = 1.6$ units (e.g. tons) of goods per hour; consumption rates $p_{1,3} = -2$, $p_{2,3} = -3$, $p_{4,3} = -1.6$ units of goods per hour; buffers initial level $x_0 = [5\ 5\ 5]$ units of goods; minimum and maximum levels for each buffer are 2 and 10 units of goods.

Considering Theorem 1, the desired system behavior is obtained with machine utilizations equal respectively to 34%, 69%, 46% and 43%. According to the grouping policy in [5], machines λ_3 and λ_4 forms a scheduling group and hence are never active at the same time. The periods have been chosen according to (18) and their values are respectevely 4, 3, 2 and 1 hours. In this particular case, an un-coordinated load management brings to a peak power consumption of 10 kW, while the proposed method cuts this value down to 6 kW, introducing a significant improvement of 40%.

8. COMMENTS ON THE PROPOSED APPROACH

In past section the modeling and control approach based on real-time scheduling has been described to limit the peak load of electric power usage in an industrial process. As stated, the research has started by considering the work in [3]. This section will discuss the main differences, considering pros and cons, between the approach applied in this paper and the method used in [3]. In particular, the main benefits of the proposed method, namely flexibility and scalability, will be justified.

In [3], the scheduling problem was solved *offline* using a linear programming solver, after the problem has been modeled in terms of a minimization problem constrained by linear inequalities. Some parameters were calculated by the solver. In particular, the outcome was made by a set of time intervals having variable durations when loads were selected to be switched on or off. The main advantage of this approach is that, after an elegant and relatively simple system modeling, a general linear programming solver can be used to derive the *optimal* solution to the problem. However, the main disadvantage is that the complexity of the optimization problem becomes prohibitive even for relatively small systems. As stated by authors, their optimization problem was made by 99 constraints, 120 decision variables, and evaluated over a 24-h time horizon. The authors made no mention regarding the computational effort required to solve the problem. As a result, the proposed optimal method does not appear to suit the case of a flexible scenario where the schedule needs to dynamically change due to unpredictable events as faults or errors. In addition, there is no chance to coordinate different power systems in an integrated manner, since this would require the extension of the modeling effort to include subsystems not yet modeled. Aforementioned observations bring to the conclusion that the optimization-based approach arguably presents scalability issues when the system size increases or gets more complex.

In this paper, the time is divided in time slots having fixed size. The scheduler decides *at runtime* which loads shall be active in each slot. Under realistic assumptions, scheduling algorithms and analysis techniques having linear or polynomial complexity are available. The drawback of most commonly adopted simplifying assumptions is that they bring to sub-optimal solutions. However, as shown in [5], the performance loss is limited. Moreover, it is largely compensated by advantages such as design flexibility, scalability, robustness and integration possibilities. Note that nowadays there exists several optimal approaches. However the typical complexity of such approaches is non-polynomial.

The low complexity of available algorithms provides advantages in terms design flexibility, since it makes it easier and quicker the system design. For example, it allows the practical application of sensitivity analysis methodologies to tune system parameters. Low complexity also enables the management system to scale from few components to hundreds or thousands of controllable loads. This good scalability properties are fundamental in the application domain of the Smart Grid, where large scale control systems are likely to be deployed and coordinated.

The run-time adaptation of the schedule is a key supporting feature to manage dynamic changes to system requirements and working conditions. Variations can be due to faulty or error situations, that may happen at unpredictable time instants. Properly management of such variations can only be achieved at run-time. Therefore, the scheduling algorithm must be able to adapt the schedule with limeted overhead, which is the case of considered real-time scheduling algorithms.

Finally, one of the most remarkable advantage is related with the possibility to integrate different sub-systems in the same framework. This possibility allows to obtain further benefits by the coordinated actions of loads belonging to different sub-systems. The integration would happen transparently once each load and its relations with other loads are described in terms of timing parameters. In this case, the scheduler would trasparently manage and coordinate all loads to reduce unnecessary simultaneous activations. In this sense, the proposed approach represents a general framework for the management of power/energy systems made by a huge number of components. Beside the industrial process described in this paper, other examples of etherogeneous systems that can be modeled using the proposed approach are Heating Ventilating and Air Conditioning systems (HVACs) [6] and air compressed systems [8]. Such systems are rather common in industries, buildings and apartments. Moreover, they are critical systems from the energy efficiency viewpoint. All those systems can be integrated in one common coordination framework with clear advantages on the overall efficiency.

9. CONCLUSION

This paper presented a modeling and control approach for an industrial process based on real-time scheduling. The physical system is modeled as a set of periodic activities that can be scheduled by adapting traditional real-time scheduling algorithms. The method achieves to limit the peak load, while guaranteeing the desired behavior of the physical process, confirming the possibility to use real-time scheduling techniques to organize the activation of electric loads in a power system.

10. REFERENCES

[1] B. Bach, D. Wilhelmer, and P. Palensky, "Smart buildings, smart cities and governing innovation in the new millennium," in *Proc. of the 8th IEEE Int. Conf. on Industrial Informatics (INDIN)*, July 2010, pp. 8–14.

[2] M. L. Della Vedova, M. Ruggeri, and T. Facchinetti, "On real-time physical systems," in *Proceedings of the 18th International Conference on Real-Time and Network Systems (RTNS)*, November 2010, pp. 41–49.

[3] S. Ashok and R. Banerjee, "An optimization mode for industrial load management," *Power Systems, IEEE Transactions on*, vol. 16, no. 4, pp. 879–884, nov 2001.

[4] 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.

[5] T. Facchinetti, E. Bini, and M. Bertogna, "Reducing the peak power through real-time scheduling techniques in cyber-physical energy," in *Proc. of 1st Int. Workshop on Energy Aware Design and Analysis of Cyber Physical Systems (WEA-CPS)*, April 2010, pp. 18–25.

[6] T. Facchinetti and M. Della Vedova, "Real-time modeling for direct load control in cyber-physical power systems," *IEEE Transactions on Industrial Informatics, special issue on Information Technology in Automation*, vol. 7, no. 4, pp. 689–698, November 2011.

[7] M. L. Della Vedova, E. Di Palma, and T. Facchinetti, "Electric loads as real-time tasks: an application of real-time physical systems," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, July 2011, pp. 1117–1123.

[8] G. Quartarone, N. Anglani, and T. Facchinetti, "Improving energy management of electrically driven air compressors through real-time scheduling techniques," in *Proceedings of the 37th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, November 2011, pp. 2697–2702.

[9] T. Nghiem, M. Behl, G. Pappas, and R. Mangharam, "Green scheduling: Scheduling of control systems for peak power reduction," in *Int. Green Computing Conference and Workshops (IGCC)*, July 2011, pp. 1–8.

[10] K. Elgazzar, H. Li, and L. Chang, "A centralized fuzzy controller for aggregated control of domestic water heaters," in *Electrical and Computer Engineering (CCECE)*, may 2009, pp. 1141–1146.

[11] M. Deindl, C. Block, R. Vahidov, and D. Neumann, "Load shifting agents for automated demand side management in micro energy grids," in *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*, oct. 2008, pp. 487 –488.

[12] A. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *Smart Grid, IEEE Trans. on*, vol. 1, no. 3, pp. 320–331, dec. 2010.

[13] J.-L. Chen and Y.-Y. Hsu, "An expert system for load allocation in distribution expansion planning," *Power Delivery, IEEE Transactions on*, vol. 4, no. 3, pp. 1910–1918, Jul 1989.

[14] D.-C. Wei and N. Chen, "Air conditioner direct load control by multi-pass dynamic programming," *Power Systems, IEEE Transactions on*, vol. 10, no. 1, pp. 307 –313, feb 1995.

[15] C. Kurucz, D. Brandt, and S. Sim, "A linear programming model for reducing system peak through customer load control programs," *Power Systems, IEEE Trans. on*, vol. 11, no. 4, pp. 1817 –1824, nov 1996.

[16] M. L. Dertouzos, "Control robotics: The procedural control of physical processes," *Information Processing*, vol. 74, 1974.

[17] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 2nd ed. Springer Verlag, 2004.