Università degli Studi di Pavia

Scuola di Dottorato in
Ingegneria Elettronica, Informatica ed Elettrica

XXV Ciclo

Real-Time Physical Systems and Electric Load Scheduling

Tesi di dottorato di:
Marco L. Della Vedova

Relatore:
Dott. Tullio Facchinetti

A.A. 2011/2012

Marco L. Della Vedova
Department of Electrical, Computer and Biomedical Engineering
University of Pavia
via Ferrata, 1
27100 Pavia, Italy
email: marco.dellavedova@unipv.it

Printed in Pavia, Italy
April 30, 2013

"We know the truth, not only by the reason, but also by the heart."
(B. Pascal, *Pensées*, 1669)

Abstract

The real-time scheduling theory has been developed in the field of computer science to manage concurrent computing tasks on a processing platform under timing constraints. This thesis pioneers the implications of applying real-time scheduling to manage the activation of electric loads in a power system, with the goal to minimize the peak load of power consumption. For this purpose, the concept of Real-Time Physical System (RTPS) is proposed as a modeling and control approach. The novelty of RTPSs is to jointly study a dynamical system together with a real-time scheduler: the evolution over time of system state variables (e.g., room temperature, air pressure, basin levels, etc.) is driven by a switching signals generated by a real-time scheduling algorithm. Differently from the traditional approach of real-time control systems, where the control signal is computed by a real-time controller; in a RTPS the schedule *is* the control signal.
According to this intuition, electric loads are modeled in terms of real-time parameters and timing constraints. Suitable timing parameters are those that guarantee the respect of user requirements, which determine the quality of service on the physical process actuated by the electric loads. This representation enables the use of a real-time scheduling algorithm to manage the timely allocation of power loads in order to avoid unnecessary simultaneous activations, thus reducing the peak load. The main advantage of the proposed technique is to leverage the strong mathematical background of real-time systems in order to achieve predictability and timing correctness on the physical process controlled by loads. The proposed approach is characterized by a low computational cost and thus it is suitable to dynamically and adaptively manage large power systems. It is indeed adequate to coordinate the demand-side management operations

in electric power systems of different sizes, from small smart buildings to large power grids.

The thesis deals with different types of dynamical system models, suitable to represent different types of loads and appliances, ranging from household appliances (e.g., heating and air conditioning systems, refrigerators, washing machines, ovens) to industrial devices, such as air compressed systems and pumps. For each model, specific rules are derived to find the timing parameters to represent the behavior of power loads. Timing parameters are used to analyze the feasibility of the system, i.e., to verify that timing and physical constraints are guaranteed. This approach allows to build a general framework based on the RTPS approach, where a set of heterogeneous devices and appliances can be concurrently managed.

The performance of the proposed methods is assessed through extended simulations based on realistic parameters. For example, the application of the coordination approach to a large set of air conditioning systems shows outstanding performance, i.e., the ability to reduce the peak load by the 35% in normal working conditions, and up to 60% with respect to worst case situations, without affecting thermal comfort for occupants. Therefore, the RTPS approach proves to be a promising solution to realize an effective Demand-Side Management framework.

Contents

List of acronyms

CPES Cyber-Physical Energy System. 25, 26

CPS Cyber-Physical System. 24, 25, 70

CPU Central Processing Unit. 53, 54, 56

DLC Direct Load Control. 31, 48

DR Demand Response. 31, 32

DSM Demand-Side Management. 31, 35, 47, 48, 187

EDF Earliest Deadline First. 49, 59–63, 65, 67, 72, 76, 110, 148, 149, 152, 154–156, 170, 172, 174, 179, 183

ELM Electric Load Management. 20, 21, 26, 27, 33, 34, 49, 144, 169

FFDH First-Fit Decreasing Height. 155, 156, 158, 174, 179, 183

HVAC Heating, Ventilation and Air Conditioning. 21, 41, 45, 47–49, 104, 162, 163

RM Rate Monotonic. 59, 60, 65, 66, 76, 149

RTPS Real-Time Physical System. 21, 50, 70–72, 74, 78, 79, 81, 141, 144, 148, 149, 169–176, 179–184, 187, 188, 192

SLR Systematic Literature Review. 34, 35, 39, 40, 46

WCET Worst-Case Execution Time. 73, 74

1 Introduction

As the population, the economy and the wealth of emerging countries are growing, and the pressure on the existing natural resources is increasing. Modern economies need to overcome current development models uniquely based on the concept of "growth", and possibly switch to other models that explicitly account for the efficiency in the usage of natural resources. Energy drives the consumption of many limited natural resources. The efficient use of energy has become a major issue in most of countries (US, EU, together with China). A fundamental driver of the development, in particular during an economic crisis, is the investment on an efficient resources usage. Nowadays, energy represents one of the most demanded (hence critical) resources.

Energy, and in particular electric energy, production still relies on fossil fuels like oil, natural gas, and coal, and in general on non-renewable sources[1]. In the global scale, energy demand grows by the 1.6% each year

[1] In Italy fossil sources of energy contribute for the 73% of the overall production (Terna, 2008), in the EU-27 for the 61% (Eurostat, 2007) and in the US for the 68% (Energy Information Administration, 2008).

(World Energy Outlook, 2008) and conversely fossil source of energy are depleting and their use is the cause of many environmental problem, such as pollution, greenhouse gases, carbon emission, climate change. Since non-renewable means non-sustainable, the energetic problem is strategic and so it is seriously tackled by governments worldwide. The E.U. launched in 2009 the "20-20-20" program[2] that commits member countries achieving by the 2020 the following goals: (i) reduce the by the 20% the carbon emission, (ii) produce at least the 20% of the primary energy by renewable sources, (iii) reduce energy consumption by the 20% thanks to improvements on energy efficiency. The U.S. government presented in 2008 the "New Energy for America" plan[3] aiming to invest $150 billion over the next ten years to encourage private efforts to develop clean energy sources and technology.

Contributions The energetic problem must be addressed from both sides: energy production and energy consumption. The focus of this thesis is related to the energy consumption side and in particular on energy efficiency. We aim to presented a novel method for electric load scheduling that makes energy systems more efficient. The contribution of this thesis will be both theoretical and application-oriented. While the theoretical contribution will be focused on modeling and analyzing the considered energy systems, the application-oriented research will validate the derived theoretical results.

Apply real-time techniques to the resource allocation The modeling effort will have a solid foundation in the domain of real-time scheduling and power minimization on multiple resources (such as multiprocessors). Real-time scheduling can be seen as the discipline of allocating resources

[2] See http://ec.europa.eu/europe2020.
[3] See http://apps1.eere.energy.gov/news/news_detail.cfm/news_id=12194.

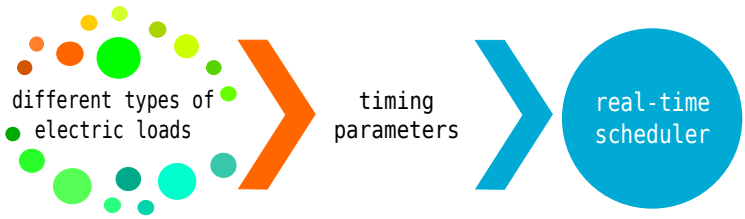


Figure 1.1: Description of the framework for electric load management using real-time scheduling techniques: heterogeneous electrical loads are modeled using timing parameters whose values are fed to the real-time scheduler, which in turn triggers the actual activation of loads.

over time to a set of resource-consuming tasks, so that given constraints (often modeled as deadlines) are satisfied. The key idea is to extend and adapt existing scheduling techniques in the real-time computing domain to multi-resource energy systems. This approach will allow to use state of the art modeling techniques, scheduling algorithms and optimization methods developed in the field of real-time computing systems to manage devices and sub-systems in energy systems. This methodology will be achieved by properly modeling systems in terms of the consumed/produced resources and timing/physical constraints. Once loads have been modeled in terms of suitable timing parameters, they can be managed by the scheduler, as depicted in Figure 1.1. This approach has the huge potential to automate the management of energy systems with energy efficiency as a primary goal. Among other benefits, the vast amount of theoretical results available for real-time scheduling systems will be exploited. The strong mathematical background of real-time systems will simplify the development of innovative formal models and powerful analysis techniques for energy systems.

The developed methodologies and models must be able to span onto different system granularities. The multi-resource scheduling problem presents

at least the following levels of investigation:

1. **Device level** (for example, a single appliance). Each device may have constraints that have to be satisfied ("a refrigerator must keep the temperature within a given interval") and it is possible to drive each single component to achieve a desired goal (a lower peak consumption).

2. **Network level**. Several devices can be grouped on the basis of specific spatial partitioning ("all the appliances in the same house") and controlled in a coordinated manner. For example, a set of appliances can be scheduled to balance the overall peak power demand. In an industrial scenario, a set of plants can be coordinated to achieve the manufacturing of a desired number of products, while optimizing the consumed energy.

3. **District/city level**. It may be possible to schedule on a larger space/time scale the power demanding subjects to balance between daylight activities, that usually involve human operators, and nightly activities, that can typically address industrial tasks, or residential and industrial areas.

These characteristics of power demand and supply suggest that the same hierarchical scheduling methodologies that have been developed for the real-time scheduling can be applied in this domain as well. Following this methodology, several resource demands at the same abstraction level are aggregated to constitute a demand at the upper level. This allows the decomposition of a high level goal into many lower level goals.

Controlling the system After the involved quantities are modeled, it is necessary to establish a policy that governs the systems under control. In

control systems, for example, this happens when the periods of controllers are selected such that deviation between the sampled state and the state evolution is minimized. The optimal policy design is the phase where parameters are selected by a formal technique (e.g. optimization methods) and not by the designers.

The management of resource consumption can follow two different control strategies: proactive or reactive. Proactive control can be implemented when a deeper knowledge of the involved physical process is available. In this case the resource allocation is established in a given sequence that aims at reducing the cost function (e.g., the peak consumption). In this case, the detailed knowledge of resource requirements allows the adoption of open loop controllers. However, in many circumstances the modeling of resource consumption can not be a priori determined. Therefore, a proactive control strategy is not possible. This happens, for example, in very complex systems or when some unpredictable actions can change the resource requirement (for example, frequently opening a refrigerator). In such cases a closed loop controller is more appropriate. Reactive control is based on a feedback loop between the measured physical value and the expected setpoint. In the considered systems, this will establish a relationship between the actual consumed resource and allocated resources. Both approaches will be investigated, as well as their combination. In fact, the overall resource requirement of combined systems (such as a house) is the composition of both well known and uncertain events. Hence, a methodology that integrates both approaches is essential for a realistic environment.

A simple example is provided to show the optimal policy design in proactive control. One of the most important goals to achieve when scheduling electric loads is to reduce the peak requirement. This explains why the electricity meter may have the possibility to automatically cut off the power link

when the peak bound is exceeded. All components of the electric grid must be dimensioned to bear the peak request. Hence any effort in the reduction of the peak requirement can be translated into a direct cost reduction.

On the other hand, many devices have physical constraints to be guaranteed. Suppose that in a house two refrigerators have to keep their internal temperature between 1°C and 4°C (Celsius degrees). If they both work according to a classic threshold mechanism, then the temperature and their energy consumption evolve as indicated in the figure below.

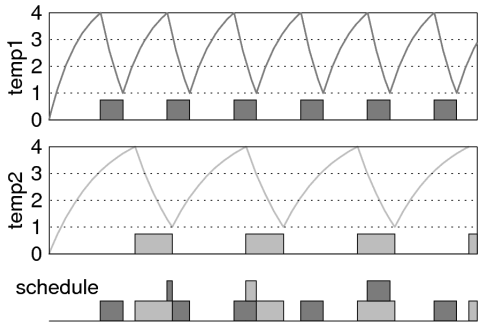


Figure 1.2: Power consumption and internal temperature of two non-coordinated refrigerators

For simplicity in this experiment the two refrigerators are modeled by a first order system, and their power consumption is either absent (0) or full (1). Nonetheless, the dynamics of the power consumption is similar if a more detailed modeling is assumed. It can be noticed in the figure that sometimes it happens that the two devices are active at the same time, achieving a peak consumption that is twice the single consumption. By a proper scheduling of the switching it is possible to meet the temperature constraint without doubling the peak consumption, as shown below.

In the switching scheme represented in the figure it can be noticed that

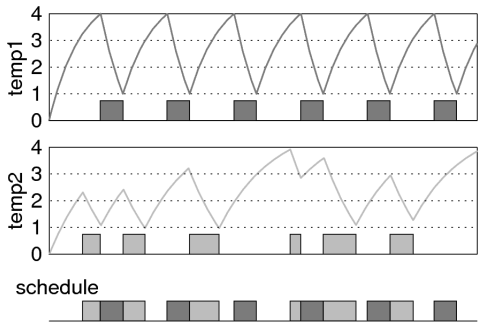


Figure 1.3: Power consumption and internal temperature of two coordinated refrigerators

it is possible to switch on the second device only in the intervals where the first device is off, without violating the temperature constraint. Clearly this extremely simple example aims at illustrating only a trivial case when smart scheduling policies can reduce the peak consumption.

In more general scenarios, the consumption is a continuous variable, and the actions that can be undertaken may be continuous as well (the motors can be driven with different speeds). Moreover, the power consuming devices may have precedence constraints, for example when a product must undergo to a sequence of stages. All these cases makes the scheduling problem more complex, and suitable algorithms and analysis must be investigated to guarantee the application constraints.

Even in more realistic scenario, the picture does not change significantly. In the example below (Figure 1.4) the energy consumption of an apartment has been monitored for 5 hours (upper-left graph). The contribution of each load to the power consumption has also been determined (lower-left graph). In the lower-right graph, a peak load reduction around 25% is achieved by properly scheduling some controllable loads, refrigerators compressors in

this case. Notice that other loads are not controllable. For example, the TV can not be freely switched on and off to comply an automatic schedule without affecting the device usability.

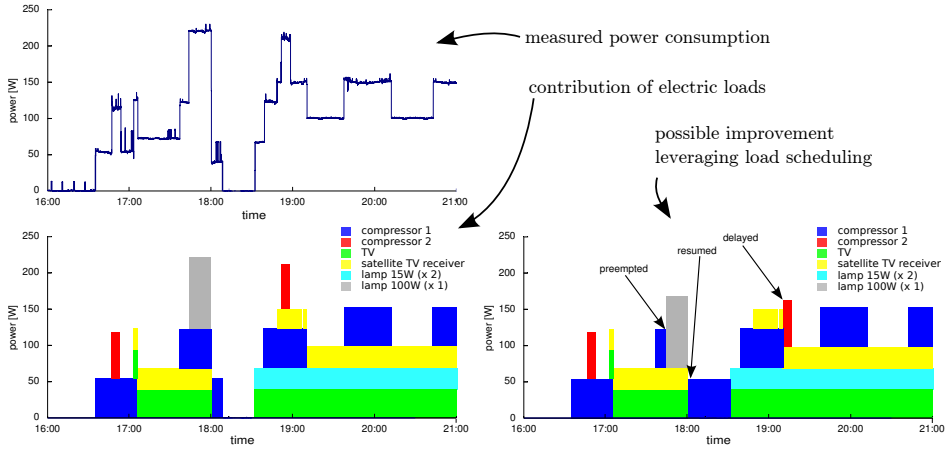


Figure 1.4: Measurements of consumed power in an apartment. Left graphs depict the normal power consumption of some electric loads, both measured power and contributions of specific loads. Right chart shows that an adequate scheduling of load activations provides allows to achieve a peak load reduction of about the 25% (adapted from E. Di Palma master thesis, 2009).

1.1 Thesis organization

This chapter gave an introduction to the present thesis. The reminder of the document is organized as follows.

Chapter 2 is a literature review about the state of the art in the field of Electric Load Management (ELM). Due to the multi-disciplinary nature of the field, it has been necessary to introduce a taxonomy between keywords in order to identify synonyms and relationships between them and eventu-

ally to perform a Systematic Literature Review. In addition, a selection of publications has been commented further.

Chapter 3 is a brief introduction to Real-Time systems. All the notions, which our proposed methodology is based on, are presented thereby.

The main theoretical contribution of this thesis is Chapter 4, where Real-Time Physical System (RTPS) models are described in details. We first show the general RTPS model and then the feasibility analysis for each specific dynamical system model. This chapter aims to answer the question about finding appropriate timing parameters for modeling electric loads in order to meet some kind of user requirements. Once electric loads have been modeled in such way, then they can be included in the RTPS control scheme.

Chapter 5 treats the application of RTPS to Electric Load Management. The goal is to reduce the peak power demand of a set of electric loads without affecting the quality of service.

Results for selected case studies are discussed in Chapter 6. The proposed RTPS scheme proves to have excellent performance on reducing the peak load. For example, for a set of a hundred Heating, Ventilation and Air Conditioning (HVAC) devices, RTPS scheme recorded an improvement of 31% in average conditions, and up to 71% in worst-case conditions, with respect to traditional thermostatic control.

At last, Chapter 7 states concluding remarks.

2

Electric Load Management: a literature review

Contents

2.1 Introduction

Balancing the power demand curve is big challenge for the power delivery system. The problem is that electric power production has to constantly match the consumption because nowadays an efficient system to store energy does not exist. Therefore, dealing with a highly variable power demand is a big issue for the utilities. Figure 2.1 shows the overall power demand in Italy as a nation in 2012. The curve is anything but flat: the ratio between the peak (i.e., about 50GW) and the minimum power (i.e., about 30GW) required during an average day is 1.6, meaning that the peak corresponds to the 60% more than the minimum power.

Peak load conditions are bad, for all players of the power delivery system. Energy providers have the problem that the less efficient (and the most polluting) power plants are the most expensive to stop. For example, interrupt the working operation of a gas turbine is much more complicated than stop a hydroelectric generation unit. As a consequence, power plants that use renewable sources are not used at their full capacity, since it is more convenient to stop them instead of the other plants. Moreover, all the power distribution infrastructure must be dimensioned for the peak. End-users pay all this disadvantage, for example with the critical-peak tariffs.

So, it is desirable to have a flatter power demand curve both at a national level and a building level. New technologies enable new approaches to this issue: the emerging Smart Grid at the national level and embedded systems for smaller systems.

The research trend on embedded systems is focusing on the integration of computing resources within the physical system under monitoring and control. This approach represents the foundation of Cyber-Physical System

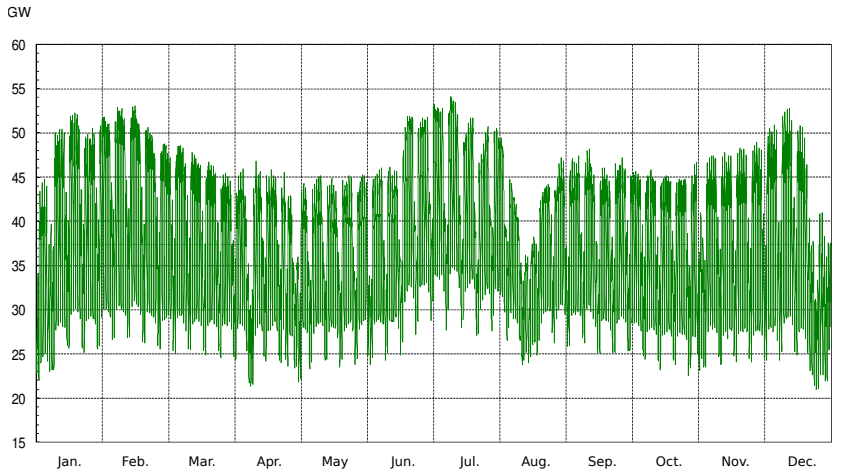


Figure 2.1: Italian power demand curve during 2012. Source: Terna monthly report, Dec. 2012, available on http://www.terna.it.

(CPS) (Lee and Seshia [42]). The most relevant characteristic of a CPS is the tight integration between the physical process under control and the controlling digital computing system. Key issues in CPS are sensing and actuation, the modeling of the physical system, real-time computing, and networking. Example applications for CPS are in the field of manufacturing control, energy systems, automotive and avionics systems, traffic control, medical systems, cooperative robotics and smart buildings (see (Stankovic et al. [66]) for detailed examples).

Home and industrial automation systems more and more require to address the issue of energy efficiency. Cyber-Physical Energy System (CPES) are a dedicated case of CPS dealing with energy systems (Morris et al. [52]). The research on CPES mainly focuses on power systems. In those systems, embedded computing is integrated within the energy system to gather information about the most relevant parameters, such as voltage,

current, phases, consumed energy and power. Environmental parameters, as temperature, humidity and pressure, are also important for system characterization. Acquired data are then combined and processed to generate suitable control commands to achieve the desired application goal. Typical goals include efficient power and energy usage.

The Smart Grid is an example of a highly automated power system. Smart grids focus on the interaction between electricity suppliers and users, where a two-way flow of energy between (distributed) providers and users is supported by a pervasive and interconnected information infrastructure.

Monitoring and control tasks can be applied on components of a CPES by a set of networked embedded devices to perform the so-called **Electric Load Management (ELM)**. Figure 2.2 provides an example of networked electric components at building and neighborhood level. The figure represents consuming, generation (solar cells) and storage (electric cars) units. Devices are coordinated through their interconnection to a communication network. Several buildings can be managed leveraging the same communication network.

Electric Load Management has gained a lot of interest in the last years. The multidisciplinary nature of ELM caused the spread of many keywords. Figure 2.3 shows the frequency of some keywords in the scientific literature by a so-called *word cloud*[1] visualization, in which the font size is proportional to the number of papers that contain the keyword in the title or in the abstract until 2012. Figure 2.4 shows the number of results in Google search. It is interesting to see that the popularity of some keywords in web pages number is inverse in respect of the scientific literature. In particular, a list in decreasing order of popularity for web searches is: demand

[1] See Wikipedia, The Free Encyclopedia – `http://en.wikipedia.org/w/index.php?title=Tag_cloud&oldid=539553855` for further details.

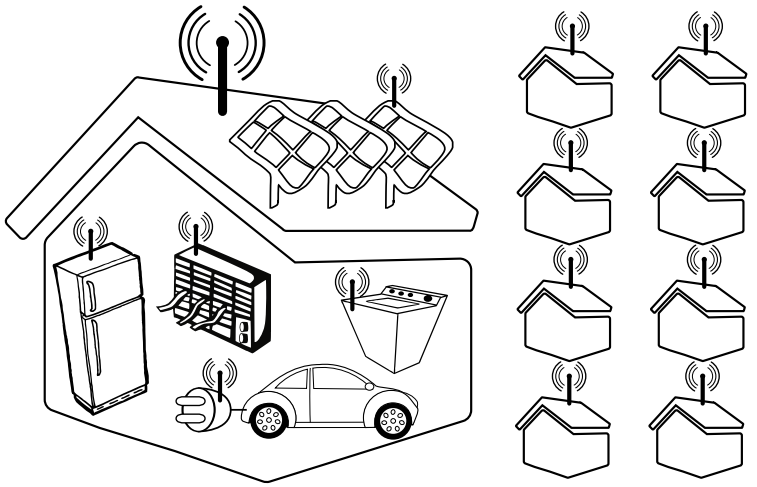


Figure 2.2: Cyber-Physical Energy System composed by networked electric devices at building and neighborhood level. The interconnection is achieved by a wireless network.

response, demand side management, electric load management; while in the scientific literature the order is exactly the opposite: electric load management, demand side management, demand response. This fact indicates a distance between researcher and the people interest, at least in terms of language.

2.2 Taxonomy

As we said, due to its multidisciplinary nature, a plenty of keywords refer to the Electric Load Management world. This section aims to keep track of most of these keywords by identifying synonyms and relations between them. The following convention is used for listing keyword synonyms:

- words between brackets are alternatives when separated with com-

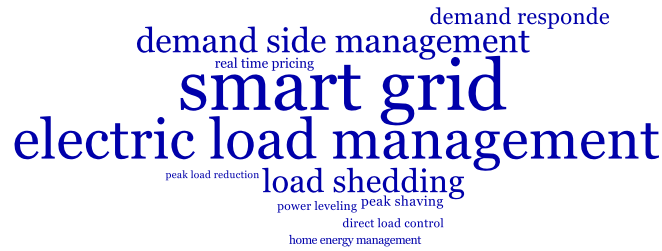


Figure 2.3: Word cloud visualization for some keywords related to electric load management. The font size is proportional to the number of scientific papers that contain it in the Scopus database.

mas;

- words between round brackets are optional;
- words between square brackets are necessary.

For example, regarding the concept of *peak load reduction*, in the scientific literature *peak load* and *peak demand* can be found as synonyms of peak. Moreover, *shaving* and *clipping* are often used instead of reduction. Therefore, according to our convention, synonyms for *peak load reduction* are listed as:

peak (load, demand) [reduction, shaving, clipping].

The identification of keywords and synonyms is very important for seeking research papers, especially in a multidisciplinary field like this one, where several communities tackle the problems from different points of view and tend to publish their work within conferences and journals specific of the original community.

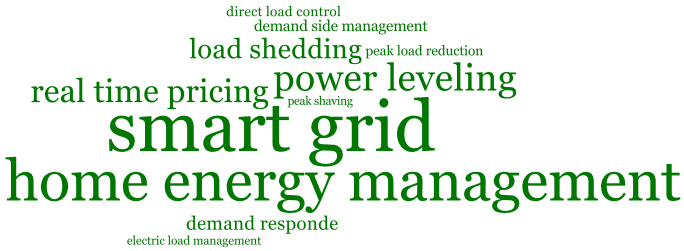


Figure 2.4: Word cloud visualization for some keywords related to electric load management. The font size is proportional to the number of Google search results.

It follows a list of keywords grouped by domain (technologies, techniques, objectives, methods). For each keyword are indicated:

- a brief description,
- related terms, and
- the synonyms, displayed according to the aforesaid convention.

2.2.1 Technologies

Smart Grid A Smart Grid is "the integration of power, communications, and information technologies for an improved electric power infrastructure serving loads while providing for an ongoing evolution of end-use applications" (IEEE [34]). Basically, the technology of the smart grid is the big picture that include all the other techniques and methods.

Smart Building A smart building is a building equipped with an integrated control system that gives to it some kind of automation. If some of this automation is related to energy and power usage, than the smart building is an active part of the smart grid.
Related terms: intelligent energy systems.

Smart Home It is a smart building and the building is a residential building.
Synonyms: smart house.
Related terms: Home Management System, [Home, Residential, Household] energy management, home automation, domotics.

Smart Load It is an electric load (e.g. household appliance) that is connected and can communicate with other loads in its domain. Smart loads are the building blocks of a smart home or a smart building. Ultimately, it is part of a smart grid.
Synonyms: intelligent (electric) load.

Smart meter "A smart meter is usually an electrical meter that records consumption of electric energy in intervals of an hour or less and communicates that information at least daily back to the utility for monitoring and billing purposes. Smart meters enable two-way communication between the meter and the central system. Unlike home energy monitors, smart meters can gather data for remote reporting. Such an Advanced Metering Infrastructure (AMI) differs from traditional Automatic Meter Reading (AMR) in that it enables two-way communications with the meter"[2].
Related terms: advanced metering infrastructure.

[2]Source: Wikipedia, The Free Encyclopedia – http://en.wikipedia.org/w/index.php?title=Smart_meter&oldid=545204777

2.2.2 Techniques

Demand-Side Management (DSM) DSM in general refers to any activity adopted by an electric utility that ultimately changes the utility's system load curve. It "includes everything that is done on the demand side of an energy system, ranging from exchanging old incandescent light bulbs to compact fluorescent lights up to installing a sophisticated dynamic load management system" (Palensky and Dietrich [59]).

Demand Response (DR) A DR program is "a tariff or program established to motivate changes in electric use by end-use customers in response to changes in the price of electricity over time, or to give incentive payments designed to induce lower electricity use at times of high market prices or when grid reliability is jeopardized" (US Dept. of Energy [69]).

Incentive-Based DR:

- **Direct Load Control (DLC)**: utility or grid operator gets access to customer loads for managing activations/deactivations.
- Interruptible/curtailable rates: customers get special contract with limited sheds.
- Emergency demand response programs: voluntary response to emergency signals.
- Capacity market programs: customers guarantee to pitch in when the grid is in need.
- Demand bidding programs: customers can bid for curtailing at attractive prices

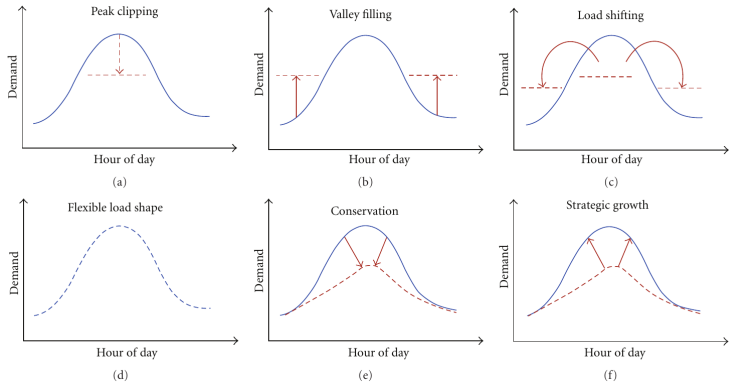


Figure 2.5: DSM load shape objectives. Source: (Kailas et al. [35]).

Time-Based Rates DR

- Time-of-use (TOU) rates: a static price schedule is applied.
- **Critical peak pricing**: a less predetermined variant of TOU.
- Real-time pricing (RTP): wholesale market prices are forwarded to end customers.

2.2.3 Objectives

Energy Efficiency It is the most general objective considered here, and so the most widespread term. Increasing energy efficiency is not just about reducing the consumed energy, it is also about consuming, producing and distributing power in a more convenient (money-wise, environment-wise and/or comfort-wise) way. Electric load management can have also other objectives, such as comfort, security, and safety.

Related terms: energy optimization.

Peak load reduction The peak load is the maximum amount of power demanded by a set of electric loads. Reducing the peak load his an important goal for all the actors in the power grid. The main reason is that with the current technology it is not possible to store energy efficiently and so power generation must match power demand at every time instant. As a consequence of this fact, together with the fact that the less efficient power plants are the less convenient to stop, it turns out that flatter the power demand curve is, more efficient can be power generation and the whole system.
Synonyms: peak (load, demand) [reduction, shaving, clipping]
Related terms: power leveling, load shifting, peak to average power ratio

2.2.4 Methods

Electric Load Management (ELM) This term has a quite general scope. A smart building has an ELM system that coordinates and operates electric loads. The problem with these keywords is that has many synonyms: so, instead of management it is possible to find terms like scheduling or control in the literature. The absence of a widespread term to indicate the same thing is due to the multidisciplinary nature of this field. Different communities use different terms.
Synonyms: (Residential) (Electric) (Deferrable) Load [Scheduling, Management, Control, Shifting, Shedding, Balancing], energy consumption scheduling, power scheduling

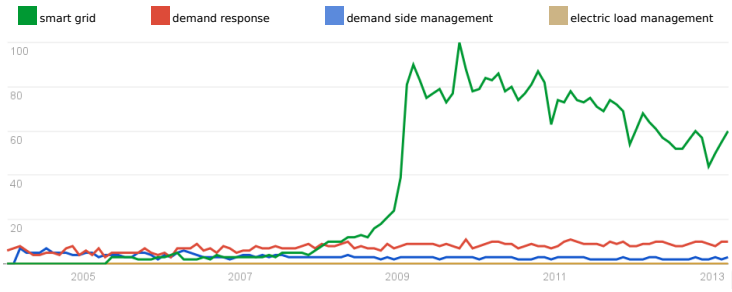


Figure 2.6: Google Trends. Interest over time of some keywords related to electric load management measured by the number of queries to Google Search website since 2004. The scale of the vertical axis is normalized to the peak value. Source: http://www.google.com/trends

2.3 Systematic literature review

The primary objective of this section is to give an overview of the research about peak load reduction in Electric Load Management through a direct and systematic analysis of a sample of mainstream articles in the 1981–2012 period. The study aims to investigate what constitutes this field by examining topics, themes, and research strategies in the literature. This establishes a benchmark for tracking the status of field development and evolution, while focusing attention on areas required more research. The present Systematic Literature Review (SLR) follows the principles proposed by (Kitchenham et al. [38]).

2.3.1 Source definition

The first activity of a SLR is to define the source. Among other on line databases, we chose to use the Sciverse Scopus database (http://info.scopus.com) because it indexes work published in those journals, maga-

zines, conferences, and workshops which are of recognized quality by the research community. Scopus claims to be "the world's largest abstract and citation database of peer-reviewed literature with smart tools that track, analyze and visualize research"[3].

2.3.2 Studies selection

The following step of the SLR is to select the studies. Study selection within on line databases translates to formulate the query string. The task is not easy because of the great variety of synonyms in keywords related to this field. For example, by using just "demand side management" as query string, all those papers where authors refer to "demand response" but not explicitly to DSM will be excluded from the results although they are about the same topic. The inclusion of these latter papers is important however, for the sake of completeness of the literature review. The definition of a taxonomy presented in Section 2.2 appears very useful in this phase. In fact, thanks to the synonyms definitions and to the research tools of Scopus, it is possible to include all the important works by using the "OR" statement.

The perfect query should include all and only the relevant papers for the subject. The use of synonyms is an attempt to avoid excluding relevant papers. However, an abuse of "OR" statements causes the inclusion in the results of not relevant work. This happens when one or more keywords are used in other research fields, too. For example, "peak load" refers to types of loads other than electric, such as weight loads in material science or traffic peak hours in traffic control systems.

Taking into account all the aforesaid considerations and the taxonomy presented in Section 2.2, we used the following query string on Scopus:

[3]Cit. `http://info.scopus.com` (visited on Jan. 2013).

```
TITLE-ABS-KEY("demand side management" OR "demand response"
OR "electric load management" OR "home energy management" OR
"direct load control") AND TITLE-ABS-KEY("peak load" OR "peak
clipping" OR "peak shaving" OR "load shifting" OR "power
leveling" OR "power levelling" OR "load profile" OR "peak
power" OR "peak demand") AND LANGUAGE(english) AND PUBYEAR <
2013
```

The query is divided into three parts. The simplest is the last part, i.e. `AND LANGUAGE(english) AND PUBYEAR < 2013`, by which we exclude papers in language other than English or published in the current year (2013) from results. Including also paper from the current year brings to incompleteness with respect to the analyzed period. With this choice we can claim to have considered all the papers published until 2012, or, more precisely, all the papers until 2012 indexed by Scopus by the date when we have performed the search (i.e., March 22, 2013). With the first and the second part of the query, we seek papers having at least one of the listed keywords explicitly written either in the title, or in the abstract, or in the keyword section. The first part of the query, i.e. `TITLE-ABS-KEY("demand side management" OR "demand response" OR "electric load management" OR "home energy management" OR "direct load control")`, is related to techniques and technologies of interest and so keywords are: demand side management, demand response, electric load management, home energy management, direct load control. The second part of the query, i.e. `TITLE-ABS-KEY("peak load" OR "peak clipping" OR "peak shaving" OR "load shifting" OR "power leveling" OR "power levelling" OR "load profile" OR "peak power" OR "peak demand")`, is related instead to objectives and keywords are: peak load, peak clipping, peak shaving,

load shifting, power leveling, power levelling (with two *l*), load profile, peak power, peak demand. The query as a whole returns the intersection between results of the first part and results of the second part. In this way, we aim to have query which is as much as possible:

- **complete**, i.e. it includes (ideally) *all* relevant papers, thanks to the use of synonyms and to the exclusion of the current year, and
- **precise**, i.e. it includes (ideally) *only* the relevant papers, thanks to the intersection between keywords related to methods and those related to objectives.

2.3.3 Preliminary results

The result for the research performed on March 22, 2013 is a set of 831 papers. Figure 2.7 shows the number of publications per year. A remarkable increasing of the number of papers published per year happened in the very last years. Comparing this fact with the trends of Figure 2.6, which plots the number of researches in Google, it is evident a similarity. These facts can be interpreted as a general rise of interest, both in scientific community and not, in energy-related and smart grid issues. One of the main reason is, in our opinion, the contemporary beginning of big government investments in energy efficiency, such as the EU 20-20-20 program and the "new energy for America" plan, both already mentioned in Chapter 1.

Figure 2.8 shows Countries from which publications come from. More precisely it is the nation of the authors' institution. Not surprisingly, most of the papers are from the U.S.A., even though the interest on this topic is world-wide: as many as 60 Countries are represented.

Figure 2.9 shows the number of publications per research area, as defined within Scopus.

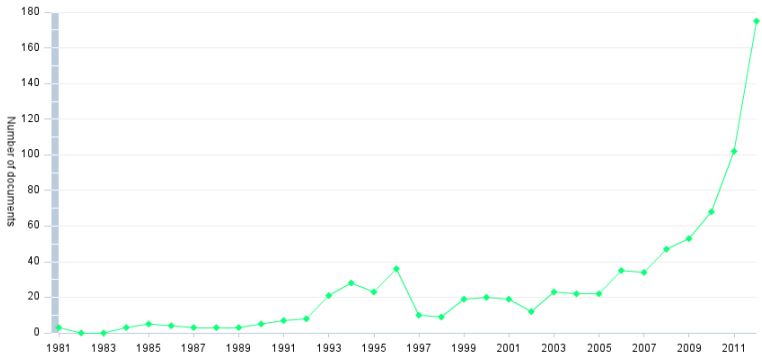


Figure 2.7: Number of publications per year. Source: http://www.scopus.com

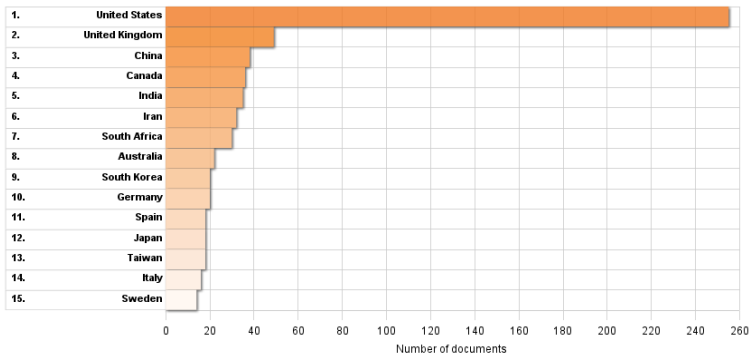
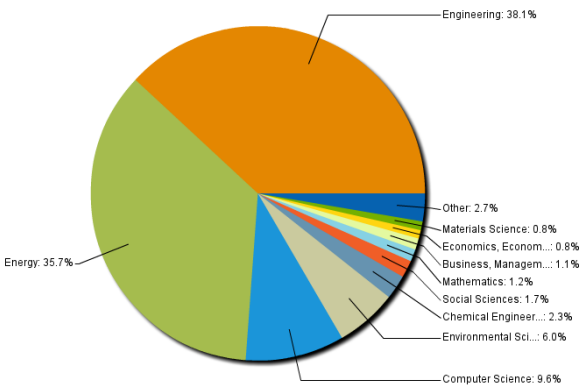


Figure 2.8: Number of publications per country. Source: http://www.scopus.com



Subject Area	Documents
Engineering	454
Energy	426
Computer Science	114
Environmental Science	71
Chemical Engineering	28
Social Sciences	20
Mathematics	14
Business, Management and Accounting	13
Economics, Econometrics and Finance	10
Materials Science	10
Earth and Planetary Sciences	9
Physics and Astronomy	8
Chemistry	6
Agricultural and Biological Sciences	5
Decision Sciences	2

Figure 2.9: Number of publications per research area. Source: http://www.scopus.com

2.3.4 Studies further selection

The next step of the SLR is to filter the studies in order to have a smaller number of papers to be further analyzed. We used the following two criteria:

1. first, we limit results to journal papers, reducing the total number of woks to 335. The reason is that, generally speaking, the quality and the maturity of journal papers is higher than conference papers. This is not an universal rule, but it applies in most of the cases. Moreover, when a conference paper has a good success, authors usually consider to forward an extended version to a journal. In Scopus, this filtering can be easily done by adding to the query string the following:

   ```
   AND (LIMIT-TO(SRCTYPE, "j"))
   ```

2. second, we took only the first 200 most cited papers. The obvious reason is to consider the most influencing, and therefore important, works. The last paper in this order has been cited 2 times, so all papers with at least 3 citations are included in our list.

2.3.5 Classification

The following step of the SLR is the information extraction. So, we classified all the 200 papers by manually labeling them with a set of tags. There are two ways to classify elements. The first is to create first a classification scheme and then place each paper in one class, or at most to a set of self-excluding classes. The other method is using tags. Tags, unlike categories, does not need to have a fixed classification scheme. The idea is to assign to each paper some tags from a tag set, allowing many-to-many relations between papers and tags. The general advantage to tag-classification with respect to category-classification is to be free of fixed classification schemes.

It follows the list of tags used for the classification together with a very brief description, grouped by topic.

- General purposes
  - *survey*: the paper is a survey
  - *modeling*: the paper presents electric load models
  - *control*: the paper discusses control systems or algorithms
  - *renewables*: renewable sources of energy (solar panels, wind turbines, etc.) are explicitly considered in the paper
  - *storage*: the paper explicitly refers to systems for storing energy
  - *experiment*: the paper presents real test case scenario, not just simulations
  - *economics*: an economic model is discussed in the paper
  - *social*: the paper talks about social impacts of the proposal
  - *not relevant*: the paper topic is not relevant in this SLR; if the query string were perfect then only relevant papers would appear in the results

- Domain
  - *industrial*: the discussed method applies to an industrial domain
  - *residential*: the discussed method applies to a residential domain
  - *tertiary*: the discussed method applies to a tertiary domain (offices, schools, etc.)
- Load type (no description is needed in this case)
  - *HVAC*
  - *water heaters*
  - *electric vehicles*
  - *pumps*
  - *lighting*
  - *industrial load*
  - *refrigerators*
  - *heterogeneous*
- Stakeholders
  - *end-user*: the proposed method is suitable for end users
  - *utility*: the proposed method is suitable for utilities
- Models
  - *dynamics*: the paper discusses dynamical systems
  - *black-box*: loads are modeled as something that just consumes power; the physical process actuated by loads is not modeled

  - *statistical*: the paper introduces statistical analysis for load characterization
  - *stochastic*: stochastic/probabilistic models are used
  - *agent-based*: the system is modeled as a multi-agent system

- Control methods (keywords refers to the method, which the discussed approach is based on; no description is needed)
  - *fuzzy logic*
  - *optimization*
  - *expert systems*
  - *real-time*
  - *direct load control*

- Tariffs (see Section 2.2.2 for descriptions)
  - *time-of-use*
  - *critical-peak*
  - *real-time-pricing*

2.3.6 Results

The first result concerns the precision of the research query, measured as the percentage of papers tagged as not relevant. They are 32, which corresponds to the 16% of the total. Therefore, we can claim that the proposed query has a precision of the 84%, which appears to be a good result, considering the discussed difficulties about synonyms and shared keywords with other disciplines.

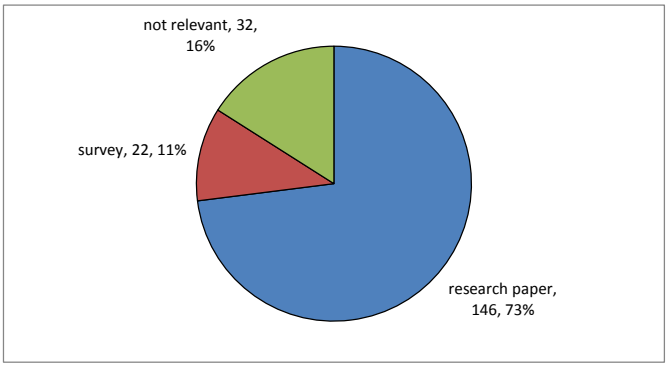


Figure 2.10: Paper typologies percentages. Not relevant papers are the 16%, which is a good result in terms of query precision.

Regarding papers typology, surveys are 22 (11%) and research papers 146 (73%). Figure 2.10 summarizes the aforesaid.

Figure 2.11 shows the number of papers labeled by a tag within the generals. Modeling, control and economic issues related to electric load management are topics for more than a quarter of the papers. Very few papers explicitly addresses social aspects.

Figure 2.12 shows a comparison between modeling papers and control papers in terms of model typology. The chart shows a mismatch between modeling and control: the majority of the papers that actually propose models concerns statistical or dynamic models, while most of the papers that propose control scheme treated black-box models. This latter observation suggests that most control schemes proposed in the literature do not take into account the physical model of the process related to the electric load, but they care just the power consumption. Conversely, our approach takes explicitly into account both power demand and the physical system.

Figure 2.13 shows proposed methods typology. The majority of the

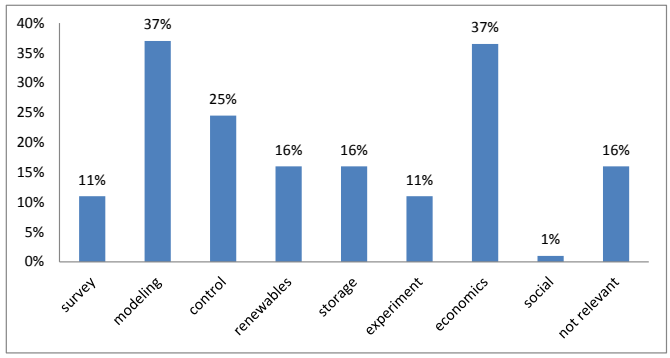


Figure 2.11: Percentage of papers by general tags with respect to the total number of papers (i.e., 200).

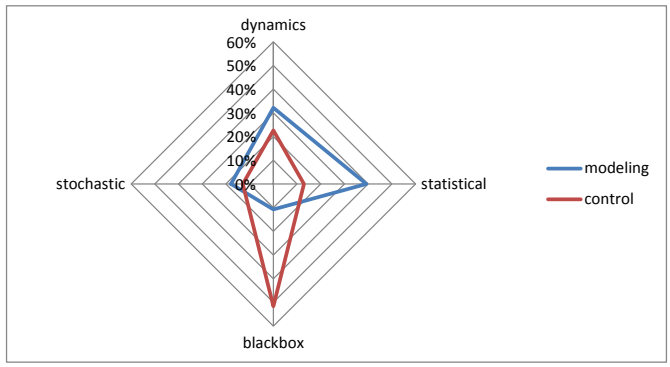


Figure 2.12: Radar chart that compares modeling-related papers to control-related papers for the type of model that is treated. It is plotted the relative percentage.

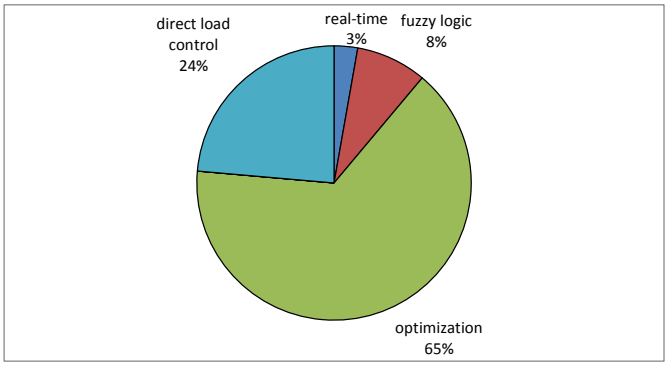


Figure 2.13: Methods relative percentage.

papers are about optimization. Main advantages of optimization is the capability to include into the model many types of constraints and the availability of well-known resolution methods. The drawback is the computational complexity that affect scalability. Our method, although does not reach the optimal solution, has a low computational cost that allows it, for example, to be implemented in low-power embedded systems.

Figure 2.14 lists the load types considered in the papers. There is a little prevalence of HVAC systems, which is justified by the fact that this type of loads consumes more than a third of the total electricity in the residential buildings[4] and about a half in commercial buildings[5] (these data refer to the U.S., but the situation is comparable in other developed Countries).

[4]Source: U.S. Energy Information Administration http://www.eia.gov/energyexplained/index.cfm?page=electricity_use (visited in March, 2013).

[5]Source: U.S. Energy Information Administration http://www.eia.gov/energyexplained/index.cfm?page=us_energy_commercial (visited in March, 2013).

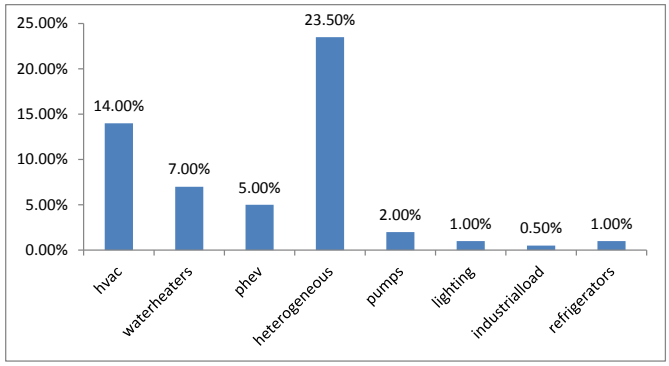


Figure 2.14: Papers by load types.

2.4 Comments on selected publications

The third most cited paper of the SLR list is titled "Smart Integration" (Vojdani [70]), from IEEE Power and Energy Magazine. The inspirational abstract states motivations, technologies and goals of the integration process in the Smart Grid: "Electric utilities in the United States and globally are heavily investing to upgrade their antiquated delivery, pricing, and service networks including investments in the following areas: - smart grid, which generally includes improvements upward of the meters all the way to the transmission network and beyond - smart metering, sometimes called advanced metering infrastructure (AMI), which usually includes control and monitoring of devices and appliances inside customer premises - smart pricing including real-time pricing (RTP) or, more broadly, time-variable pricing, sometimes including differentiated pricing - smart devices and in-home energy management systems such as programmable controllable thermostats (PCTs) capable of making intelligent decisions based on smart prices - peak load curtailment, demand-side management (DSM), and de-

mand response (DR) - distributed generation, which allows customers to be net buyers or sellers of electricity at different times and with different tariffs, for example, plug-in hybrid electric vehicles (PHEVs), which can be charged under differentiated prices during off-peak hours. The main drivers of change include: - insufficient central generation capacity planned to meet the growing demand coupled with the increasing costs of traditional supply-side options - rising price of primary fuels including oil, natural gas, and coal - increased concerns about global climate change associated with conventional means of power generation - demand for higher power quality in the digital age." (Vojdani [70])

The literature on power systems addressing DSM approaches is wide. Some works focuses on the modeling aspects, without proposing a control method based on those models (Molina-Garcia et al. [51]). Optimization methods are often adopted to minimize the peak load. Sometimes the focus is only on the consumed power, while the underlying physical process is neglected (Kurucz et al. [40], Wei and Chen [71]). In other cases, despite the physical process is taken into account (e.g., sets of HVAC loads), the adopted solutions are based on off-line algorithms that can not cope with the dynamic nature of a large system (Lee et al. [43], Ramanathan and Vittal [61]). Several proposals are based on artificial intelligence methods, such as fuzzy logic (Chu et al. [14], Elgazzar et al. [27]), self-organizing agents (Deindl et al. [17]) or expert systems (Chen and Hsu [12]). However, the properties of these methods (and predictability in particular) are not formally proved.

(Ericson [28]) discuss benefits of direct load control residential water heaters with reference to Norway. The paper analyses data from an experiment where residential water heaters were automatically disconnected during peak periods. A regression model is used to predict the average

residential consumption and loads are switched off when it is convenient according to an econometric model. Results shows reduction both in peak load and energy consumption. With respect to our approach, this method is dedicated to one type of electric load.

(Dietrich et al. [26]) concerns demand response program with reference to Spain. The paper analyses the operation of an electric system with high wind penetration, which is modeled by means of a unit commitment problem. "Demand shifting is modeled in two different ways. Firstly, the system operator controls the shift of demand; secondly, each consumer decides its reaction to prices depending on its elasticity. The model is applied to the isolated power system of Gran Canaria." As DSM scheme, two ways to model demand shifting measures are presented. In the first one, the decision to shift demand is taken using a pure cost criterion. In the second one, elasticities and demand functions are introduced to model demand reactions. Unlike our approach, end-user devices are not explicitly modeled: the demand-side is dealt as a load that can be shifted. This is a typical utilities-oriented approach: the focus is not on scheduling electric devices, it is instead on the production-side of the energy.

(Newsham et al. [54]) is analysis of the peak load reductions due to a residential direct load control program for air-conditioners in southern Ontario (CA) in 2008. Although this paper does not propose any DLC scheme, it presents an interesting comparison of four evaluation methods for DLC. Results shows that average load reductions were 0.2–0.9 kW per household, or 10–35%, which is comparable with results expected by the application of our approach.

(Chu and Jong [13]) proposes a novel DLC method for HVAC systems that takes into account thermal comfort of consumers and payback load effects. The method is based on fuzzy logic and in particular on a least

enthalpy estimation (LEE)-based thermal comfort controller. This paper deals with complex HVAC systems, which are different with the loads considered in our approach. Focus is more on energy optimization of complex HVAC systems, which is different from our approach that has the focus on electric load scheduling. Another interesting work with the same focus is (Ma et al. [49]), where the optimization is made with a model predictive control approach.

(Subramanian et al. [67]) has an approach very similar to our proposal. It applies classical real-time scheduling algorithms, namely Earliest Deadline First (EDF) and Least Laxity First (LLF), to the management of electric loads in presence of power generated by renewable sources. Results shows a comparison between the aforesaid methods and a Model Predictive Control (MPC) scheme. Authors attest that coordinated scheduling via any of these 3 policies decreases the required reserve energy to meet load requirements while only EDF and MPC reduce the reserve capacity requirement.

We look with a particular interest to the work of two research groups that recently published many good papers about Electric Load Management and their approach is closely related to our work.

The first group is from University of Pennsylvania (USA) and key people are T. Nghiem and G. Pappas. They proposes the so-called green scheduling approach, which addresses peak demand reduction problem by scheduling multiple interacting control systems within a constrained peak demand envelope while ensuring that safety and operational conditions are facilitated. The peak demand envelope was formulated as a constraint on the number of binary control inputs that can be activated simultaneously. Main works are: (Li et al. [44], Nghiem et al. [55, 56], Nghiem [57]).

The second group, from Virginia Tech (USA), is lead by S. Shao, M. Pi-

pattanasomporn and S. Rahman (Kuzlu et al. [41], Pipattanasomporn et al. [60], Shao et al. [62, 63, 64]). They studied and proposed many Home Energy Management algorithms and electric load models. Unlike our RTPS approach, the majority of their algorithms is based on load priorities, preset by the user. Priority-based strategies are also a common approach to face the load management problem. The weak point of the approach is the manual assignment of priorities. Our approach is also inherently based on the assignment of priorities. In fact, the scheduler dynamically determines and activates the load having the highest priority. However, the real-time scheduling provides *an automatic method* for the priority assignment on the basis of timing parameters. Moreover, timing parameters are determined on the basis of physical characteristics and constraints of the underlying controlled process. This is a distinguishing feature of real-time scheduling algorithms.

3 Real-Time Scheduling

Contents

This chapter presents some traditional results of the Real-Time scheduling discipline. Contributions of this thesis, which will be presented in the next chapters, "stands on the shoulders" of these giants.

Large parts of this chapter are adapted from (Buttazzo [7]).

3.1 Basic concepts

Real-time systems are computing systems that must react within precise time constraints to events in the environment. As a consequence, the correct behavior of these systems depends not only on the value of the computation but also on the time at which the results are produced (Stankovic and Ramamritham [65]).

Although the term *real time* is frequently used in many application fields, it is subject to different interpretations, not always correct. Often, people say that a control system operates in real time if it is able to *quickly* react to external events. According to this interpretation, a system is considered to be real-time if it is fast. The term *fast*, however, has a relative meaning and does not capture the main properties that characterize these type of systems (Buttazzo [7]).

In fact, whereas the objective of fast computing is to minimize the average response time of a given set of tasks, the objective of real-time computing is to meet the individual timing requirements of each task. However short the average response time can be, without a scientific methodology we will never be able to guarantee the individual timing requirements of each task in all possible circumstances. When several computational activities have different timing constraints, average performance has little significance for the correct behavior of the system. To better understand this issue, it is worth thinking about this little story: "*There was a man who drowned crossing a stream with an average depth of six inches*" (Buttazzo [7]).

At the process level, the main difference between a real-time and a not-real-time task is that a real-time task is characterized by a **deadline**, which is the maximum time within which it must complete its execution. Depending on the consequences that may occur because of a missed dead-

line, real-time tasks are usually distinguished in two classes, *hard* and *soft*:

- A real-time task is said to be **hard** if missing its deadline causes a system failure.
- A real-time task is said to be **soft** if missing its deadline decreases the performance of the system but does not jeopardize its correct behavior.

3.1.1 Definitions

The most important software entity treated by any operating system is the **process**, or **task**. A task is a computation that is executed by the processor in a sequential fashion. When a single processor has to execute a set of concurrent tasks – that is, tasks that can overlap in time – the Central Processing Unit (CPU) has to be assigned to the various tasks according to a predefined criterion, called **scheduling policy**. The set of rules that, at any time, determines the order in which tasks are executed is called a **scheduling algorithm** (Buttazzo [7]).

Thus, a task that could potentially execute on the CPU can be either in execution, if it has been selected by the scheduling algorithm, or waiting for the CPU, if another task is executing. A task waiting for the processor is called a **ready** task, whereas the task in execution is called a **running** task. All ready tasks waiting for the processor are kept in a queue, called ready queue. Operating systems that handle different types of tasks may have more than one ready queue.

In many systems that allow dynamic task activation, the running task can be interrupted at any point, so that a more important task that arrives in the system can immediately gain the processor and does not need to wait in the ready queue. In this case, running task is interrupted and inserted

in the ready queue, while the CPU is assigned to the most important ready task which just arrived. The operation of suspending the running task and inserting it into the ready queue is called **preemption**.

Given a set of tasks, $J = \{J_1, \ldots, J_n\}$, a **schedule** is an assignment of tasks to the processor, so that each task is executed until completion. More formally, a schedule can be defined as a function $\sigma : \mathbb{R}^+ \to \mathbb{N}$ such that $\forall t \in \mathbb{R}^+$, $\exists t_1, t_2$ such that $t \in [t_1, t_2)$ and $\forall t' \in [t_1, t_2)$ $\sigma(t) = \sigma(t')$. In other words, σ is an integer step function ad $\sigma(t) = k$, with $k > 0$, means that task J_k is executed at time t, while $\sigma(t) = 0$ means that the CPU is idle.

In general, a real-time task J_i can be characterized by the following timing parameters, which are depicted in Figure 3.1:

- **Request time** r_i: is the time at which a task becomes ready for execution; it is also referred as *arrival time* or *release time*;
- **Computation time** C_i: is the time necessary to the processor for executing the task without interruption;
- **Absolute deadline** d_i: is the time before which a task should be completed to avoid failure (if hard), or performance degradation (if soft);
- **Relative deadline** D_i: is the difference between the absolute deadline and the request time: $D_i = d_i - r_i$
- **Start time** s_i: is the time at which a task starts its execution;
- **Finishing time** f_i: is the time at which a task finishes its execution;
- **Response time** R_i: is the difference between the finishing time and the request time;

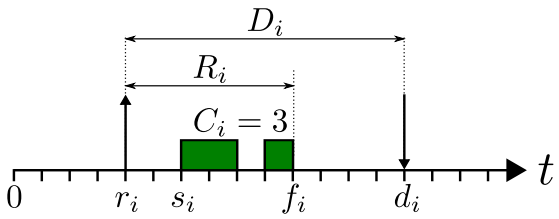


Figure 3.1: Illustration of the real-time parameters.

- **Laxity** X_i: is the maximum time a task can be delayed on its activation to complete within its deadline $X_i = D_i - C_i$.

Another timing characteristic that can be specified on a real-time task concerns the regularity of its activation. In particular, tasks can be defined as **periodic** or **aperiodic**. Periodic tasks consist of an infinite sequence of identical activities, called **instances** or **jobs**, that are regularly activated at a constant rate.

The activation time of the first periodic instance is called **phase**. If ϕ_i is the phase of the i-th periodic task, the activation time of the k-th instance is given by $\phi_i + (k-1)T_i$, where T_i is called **period** of the task. In many practical cases, a periodic task can be completely characterized by its period T_i, its computation time C_i and its relative deadline D_i (often set equal to the period). Moreover, the parameters C_i, D_i and T_i are considered to be constant for each instance. Aperiodic tasks also consist of an infinite sequence of identical jobs; however, their activation is not regular. An aperiodic task where consecutive jobs are separated by a minimum inter-arrival time is called a **sporadic** task.

If n tasks are periodic and are simultaneously activated at time $t = 0$,

then the schedule repeats itself every **hyper-period** H, where:

$$H = \text{LCM}(\{T_1, T_2, \ldots, T_n\}). \tag{3.1}$$

In (3.1) LCM(Γ) returns the least common multiple in the finite set of positive real numbers Γ.

Given the above definitions, an important figure for the periodic task model is the **processor utilization factor** U^{tot}, which is the fraction of processor time spent in the execution of the task set. Since

$$U_i \doteq \frac{C_i}{T_i} \tag{3.2}$$

is the fraction of processor time spent in the executing the i-th task, the utilization factor for n tasks is given by

$$U^{\text{tot}} = \sum_{i=1}^{n} \frac{C_i}{T_i}. \tag{3.3}$$

The processor utilization factor provides a measure of the computational load on the CPU due to the periodic task set. Although the CPU utilization can be improved by increasing tasks' computation times or by decreasing their periods, there exists a maximum value of U^{tot} below which the task set $\mathcal{R}$ is schedulable and above which it is not. Such a limit depends on the task set and on the algorithm used to schedule the tasks. Let $U^{\text{ub}}(\mathcal{R}, \mathcal{A})$ be the upper bound of the processor utilization factor for a task set $\mathcal{R}$ under a given algorithm $\mathcal{A}$.

When $U^{\text{tot}} = U^{\text{ub}}(\mathcal{R}, \mathcal{A})$, the set $\mathcal{R}$ is said to *fully utilize* the processor. In this situation, $\mathcal{R}$ is schedulable by $\mathcal{A}$, but an increase in the computation time in any of the tasks will make the set infeasible. For a given

algorithm $\mathcal{A}$, the *least upper bound* $U^{\text{lub}}(\mathcal{A})$ of the processor utilization factor is the minimum of the utilization factors over all task sets that fully utilize processor:

$$U^{\text{lub}}(\mathcal{A}) = \min_{\mathcal{R}} U^{\text{ub}}(\mathcal{R}, \mathcal{A}). \tag{3.4}$$

U^{lub} defines an important characteristic of a scheduling algorithm because it allows to easily verify the schedulability of a task set. In fact, any task set whose processor utilization factor is below this bound is schedulable by the algorithm. On the other hand, utilization above this bound can be either schedulable or not, depending on relation among periods of the tasks.

If the utilization factor of a task set is greater than one, the task set cannot be scheduled by any algorithm.

3.2 Scheduling algorithms

In general, to define a scheduling problem we need to specify three sets: a set of n tasks $J = \{J_1, \ldots, J_n\}$, a set of m processors $P = \{P_1, \ldots, P_m\}$, and a set of r types of resources $R = \{R_1, \ldots, R_r\}$. Moreover, precedence relations among tasks can be specified through a direct acyclic graph, and timing constraints can be associated with each task. In this context, scheduling means assign processors from P and resources from R to task from J in order to complete all tasks under the imposed constraints. This problem, in its general form, has been shown to be NP-complete and hence computationally intractable (Garey and Johnson [32]).

3.2.1 Classification

Among the great variety of algorithms proposed for scheduling real-time tasks, we can identify the following main classes:

- **Preemptive**: with preemptive algorithms, the running task can be interrupted at any time to assign the processor to another active task, according to a predefined scheduling policy;
- **Non-preemptive**: with non-preemptive algorithms, a tasks, once started, is executed by the processor until completion. In this case, all scheduling decisions are taken as a task terminate its execution.
- **Static**: static algorithms are those in which scheduling decisions are based on fixed parameters, assigned to tasks before their activation.
- **Dynamic**: Dynamic algorithms are those in which scheduling decisions are based on dynamic parameters that may change during system evolution
- **Off line**: We say that a scheduling algorithm is used off line if it is executed on the entire task set before actual task activation. The schedule generated in this way is stored in a table and later executed by a dispatcher.
- **On line**: We say that a scheduling algorithm is used on line if scheduling decisions are taken at runtime every time a new task enters the system or when a running task terminates.
- **Optimal**: an algorithm is said to be optimal if it minimizes some given cost function defined over the task set. When no cost function is defined and the only concern is to achieve a feasible schedule, then

an algorithm is said to be optimal if it always finds a feasible schedule whenever there exists one.

- **Heuristic**: An algorithm is said to be heuristic if it searches for a feasible schedule using an objective function. Heuristic algorithm do not guarantee to find the optimal schedule, even if there exists one.

3.2.2 Rate monotonic scheduling

The Rate Monotonic (RM) scheduling is a simple rule that assigns priorities to tasks according to their request rates. Specifically, tasks with higher request rates (that is, with shorter periods) will have higher priorities. Since periods are constant, RM is a fixed-priority assignment: priorities are assigned to tasks before execution and do not change over time. Moreover, RM is intrinsically preemptive: the currently executing task is preempted by a newly arrived task whit shorter period.

(Liu and Layland [45]) showed that RM is optimal among all fixed-priority assignments, in the sense that no other fixed-priority algorithm can schedule a task set that cannot be scheduled by RM. The least upper bound for RM is

$$U^{\text{lub}}(\text{RM}) = n(2^{1/n} - 1) \tag{3.5}$$

where n is the number of tasks.

Figure 3.2 shows an example of a RM schedule of three tasks.

3.2.3 Earliest Deadline First

The Earliest Deadline First (EDF) algorithm is a dynamic scheduling rule that selects tasks according to their absolute deadlines. Specifically, tasks

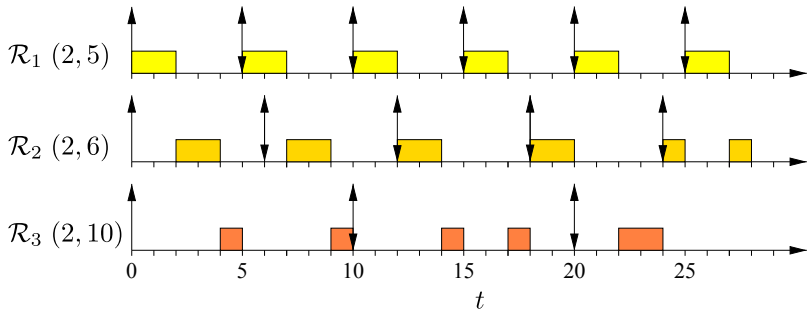


Figure 3.2: Example of RM schedule. Tasks have deadline equals to period and are indicated by $\mathcal{R}_i$ (C_i, T_i)

with earlier deadlines will be executed at higher priorities. Since the absolute deadline of a periodic task depends on the current kth instance as

$$d_{i,k} = \phi_i + (k-1)T_i + D_i, \tag{3.6}$$

EDF is a dynamic priority assignment (at a task level), although the priority of each job is fixed. Moreover, it is intrinsically preemptive: the currently executing task is preempted whenever another periodic instance with earlier deadline becomes active.

Notice that EDF does not make any specific assumption on the periodicity of the tasks; hence it can be used for scheduling periodic as well as aperiodic tasks. In (Dertouzos [24]) it is proved that **EDF is optimal** in the sense of feasibility. This means that if there exists a feasible schedule for a task set, then EDF is able to find it.

For task set with deadlines equal to periods, the least upper bound for EDF is

$$U^{\text{lub}}(\text{EDF}) = 1. \tag{3.7}$$

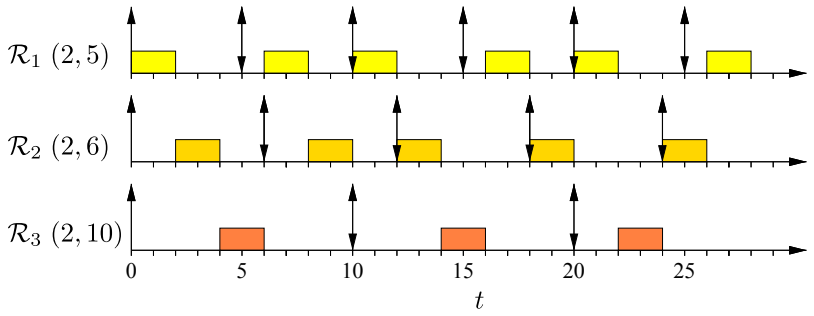


Figure 3.3: Example of EDF schedule. Tasks have deadline equals to period and are indicated by $\mathcal{R}_i$ (C_i, T_i)

Therefore, a periodic task set with deadlines equal to periods is schedulable with EDF if and only if

$$\sum_{i=1}^{n} \frac{C_i}{T_i} \leq 1. \tag{3.8}$$

Figure 3.3 shows an example of a EDF schedule of three tasks.

For any sporadic task and any real number $t \geq 0$, the **demand bound function** $\mathrm{DBF}(\mathcal{R}_i, t)$ is the largest cumulative execution requirement of all jobs that can be generated by $\mathcal{R}_i$ to have both their arrival times and their deadlines within a contiguous interval of length t. It has been shown (Baruah et al. [4]) that the cumulative execution requirement of jobs over an interval $[t_0,\ t_0 + t)$ is maximized if one job arrives at the start of the interval — i.e., at time-instant t_0 — and subsequent jobs arrive as rapidly as permitted. (3.9) below follows directly:

$$\mathrm{DBF}(\mathcal{R}_i, t) \doteq \max\left(0, \left(\left\lfloor \frac{t - D_i}{T_i} \right\rfloor + 1\right) \times C_i\right). \tag{3.9}$$

For a sporadic task set with relative deadlines less than or equal to periods, it can be scheduled by EDF if and only if

$$\forall L \in \mathcal{D} \quad \sum_{i=1}^{n} \mathrm{DBF}(\mathcal{R}_i, L) \leq L \tag{3.10}$$

where

$$\mathcal{D} = \{d_k : d_k \leq \min(L^*, H)\}$$

and

$$L^* = \frac{\sum_{i=1}^{n}(T_i - D_i)U_i}{1 - U^{\mathrm{tot}}}$$

It can be easily shown that for deadlines equals to periods, (3.8) and (3.10) are equivalent. In summary, the schedulability test for a generic sporadic task set using EDF is (3.10); when deadlines are equals to periods, the test reduces to (3.8).

3.2.4 Limited Preemption EDF

The optimality of the Earliest Deadline First scheduler for uniprocessor systems is one of the main reasons behind the popularity of this algorithm among real-time systems. The ability of fully utilizing the computational power of a processing unit however requires the possibility of preempting a task before its completion. In many systems, limiting preemptions is desirable. Arbitrary preemptions, in fact, can introduce a significant run-time overhead and may cause high fluctuations in task execution times, so degrading system predictability. The limited-preemption EDF algorithm permits preemption where necessary for maintaining feasibility, but attempts to avoid unnecessary preemptions during runtime. This is done by determining, for each task in the system, the longest amount of time for

which the task may execute non-preemptively without compromising the feasibility of the system.

Definition 1 (Non-preemption chunck)
The non-preemption parameter q_i is the longest amount of time for which the i-th task may execute non-preemptively without compromising the feasibility of the system.

In (Baruah [2]) it has been shown how to compute $q_i,\ i = 1, \ldots, n$ efficiently. Algorithm 1 reports the method.

The implementation of a limited-preemption EDF scheduler requires a mechanism to enforce the maximum time for which an executing task could delay the preemption requests of higher priority jobs. This can be done by setting a timer to the corresponding value at the time a preemption is requested. Note that the timer is set only at the first time a higher priority job tries to preempt the executing task. Additional preemption requests from other jobs can be ignored until the timer expires. When this happens, an interrupt is triggered whose only effect is to call the scheduling function. The scheduler will then simply select for execution the job with the earliest deadline, according to the EDF rules.

The operations involved are therefore just the following (Bertogna and Baruah [5]):

- At the beginning, the earliest deadline task is selected for execution.
- The first time a higher priority job arrives, the time-to-deadline of the executing task is computed as the difference between absolute deadline and current time. The value corresponding to the computed time-to-deadline is used to program a timer.

Algorithm 1 Algorithm for computing the non-preemption parameters q_i – maximum non-preemptive chunks – for a collection of sporadic tasks (Baruah [2]).

Input: Sporadic task system $\mathcal{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$; $\mathcal{R}_i = (C_i, D_i, T_i)$ for all $i,\ 1 \leq i \leq n$. Let $d_1, d_2, \ldots$ denote the time-instants, in increasing order (i.e., $d_k < d_{k+1}$ for all k), at which some task in $\mathcal{R}$ has a deadline if each task $\mathcal{R}_i$ generates its first job at time-instant 0 and subsequent jobs exactly T_i time-units apart.

Output: Non-preemption parameters q_i and system feasibility check.

$\text{SLACK}(d_1) := d_1 - \sum_i \text{DBF}(\mathcal{R}_i, d_1)$
for all k **do**
  if $d_k > T^*(\mathcal{R})$ **then**
    the system is feasible
  end if
  $\text{SLACK}(d_k) := \min\left(\text{SLACK}(d_{k-1},\ d_k - \sum_i \text{DBF}(\mathcal{R}_i, d_k)\right)$
  if $\text{SLACK}(d_k) < 0$ **then**
    the system is infeasible
  end if
  if $d_k = D_i$ for some task $i \in (1, \ldots, n)$ **then**
    $q_i := \text{SLACK}(d_k)$
  end if
end for

- Further preemption requests arriving before the timer expires are ignored.

- When the timer expires, or a task completes its execution, stop the timer and return to the first point.

(Buttazzo et al. [9]) is a recent survey about limited preemption applications in real-time systems.

3.3 Multiprocessors scheduling

This section is adapted from (Carpenter et al. [11]).

Traditionally, there have been two approaches for scheduling periodic task systems on multiprocessors: *partitioning* and *global* scheduling.

In global scheduling, all eligible tasks are stored in a single priority-ordered queue; the global scheduler selects for execution the highest priority tasks from this queue. Unfortunately, using this approach with optimal uniprocessor scheduling algorithms, such as the RM and EDF algorithms, may result in arbitrarily low processor utilization in multiprocessor systems (Dhall and Liu [25]).

In partitioning, each task is assigned to a single processor, on which each of its jobs will execute, and processors are scheduled independently. The main advantage of partitioning approaches is that they reduce a multiprocessor scheduling problem to a set of uniprocessor ones. Unfortunately, partitioning has two negative consequences. First, finding an optimal assignment of tasks to processors is a bin-packing problem, which is NP-hard in the strong sense. Thus, tasks are usually partitioned using non-optimal heuristics. Second, as shown later, task systems exist that are schedulable

if and only if tasks are not partitioned. Still, partitioning approaches are widely used by system designers.

In differentiating among multiprocessor scheduling algorithms according to the degree of migration allowed, the following three categories can be distinguished:

1. **No migration (partitioned)** – In partitioned scheduling algorithms, the set of tasks is partitioned into as many disjoint subsets as there are processors available, and each such subset is associated with a unique processor. All jobs generated by the tasks in a subset must execute only upon the corresponding processor.

2. **Restricted migration** – In this category of scheduling algorithms, each job must execute entirely upon a single processor. However, different jobs of the same task may execute upon different processors. Thus, the runtime context of each job needs to be maintained upon only one processor; however, the task-level context may be migrated.

3. **Full migration** – No restrictions are placed upon interprocessor migration.

In differentiating among scheduling algorithms according to the complexity of the priority scheme, three other categories are considered:

1. **Static priorities** – A unique priority is associated with each task, and all jobs generated by a task have the priority associated with that task. Thus, if task T_1 has higher priority than task T_2, then whenever both have active jobs, T_1's job will have priority over T_2's job. An example of a scheduling algorithm in this class is the RM algorithm.

2. **Job-level dynamic priorities** – For every pair of jobs J_i and J_j, if J_i has higher priority than J_j at some instant in time, then J_i always

	static	**job-level dynamic**	**unrestricted dynamic**
full migration	$\frac{M^2}{3M-2} \leq U^{\text{lub}} \leq \frac{M+1}{2}$	$\frac{M^2}{2M-1} \leq U^{\text{lub}} \leq \frac{M+1}{2}$	$U^{\text{lub}} = M$
restricted migration	$U^{\text{lub}} \leq \frac{M+1}{2}$	$M - \alpha(M-1) \leq U^{\text{lub}} \leq \frac{M+1}{2}$	$M - \alpha(M-1) \leq U^{\text{lub}} \leq \frac{M+1}{2}$
partitioned	$(\sqrt{2}-1)M \leq U^{\text{lub}} \leq \frac{M+1}{1+2^{1/(M+1)}}$	$U^{\text{lub}} = \frac{M+1}{2}$	$U^{\text{lub}} = \frac{M+1}{2}$

Table 3.1: Known bounds on worst-case achievable utilization (denoted U^{lub}) for different classes of multi-processors scheduling algorithms. The number of processors is denoted by M.

has higher priority than J_j. An example of a scheduling algorithm that is in this class, but not the previous class, is EDF.

3. **Unrestricted dynamic priorities** – No restrictions are placed on the priorities that may be assigned to jobs, and the relative priority of two jobs may change at any time. An example scheduling algorithm that is in this class, but not the previous two classes, is the Least Laxity First (LLF) algorithm (Dertouzos and Mok [23]).

Table 3.1 summarizes the known bounds on worst-case achievable utilization for algorithms belonging to aforesaid categories.

4

Real-Time Physical System models

Contents

A Real-Time Physical System (RTPS) is a time-dependent controlled switched hybrid system (Daafouz et al. [16]), in which the switching signal is generated by a scheduler that operates according to a real-time scheduling algorithm. The name aims to capture the nature of RTPS as an extension of traditional Real-Time systems that copes with some physical quantities of interest under control.

The name is inspired by the emerging field of Cyber-Physical System (CPS) (Lee and Seshia [42]). In CPS, embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. In RTPS the "physical" component of the system is modeled with a switched dynamical system, while the "cyber" component is the scheduling technique, which leverages the Real-Time systems theory.

4.1 General model

The most general model of a RTPS is composed by a set of m ON/OFF actuators that act on a system characterized by n physical quantities of interest.

4.1.1 Dynamical system

The dynamics of the system is described by a time-dependent controlled switched hybrid system (Daafouz et al. [16]), defined by the following equations:

$$\Phi : \begin{cases} \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = f_{s(t)}\left(x(t)\right) + d(t) \\ x(0) = x_0 \end{cases} \tag{4.1}$$

Terms in (4.1) are:

- $t \in \mathbb{R}^+$ is the continuous time span;
- $x(t) = [x_1(t) \ \ldots \ x_n(t)] \in \mathbb{R}^n$ is the vector of n state variables;
- $x_0 = \in \mathbb{R}^n$ is the state variable initial value;
- $s(t) \in \mathbb{B}^m \equiv \{0,1\}^m \equiv \{\mathrm{OFF}, \mathrm{ON}\}^m$ is the *operation mode* of the system;
- $s_i(t) \in \mathbb{B}$ with $i = 1 \ldots m$ is the operation mode (or *activation status*) of the i-th actuator;
- $s : \mathbb{R}^+ \to \mathbb{B}^m = [s_1 \ \ldots \ s_m]$ is called *switching signal* or *schedule*;
- $f_{s(t)}$ is a set of 2^m vector fields representing the dynamics of the system;

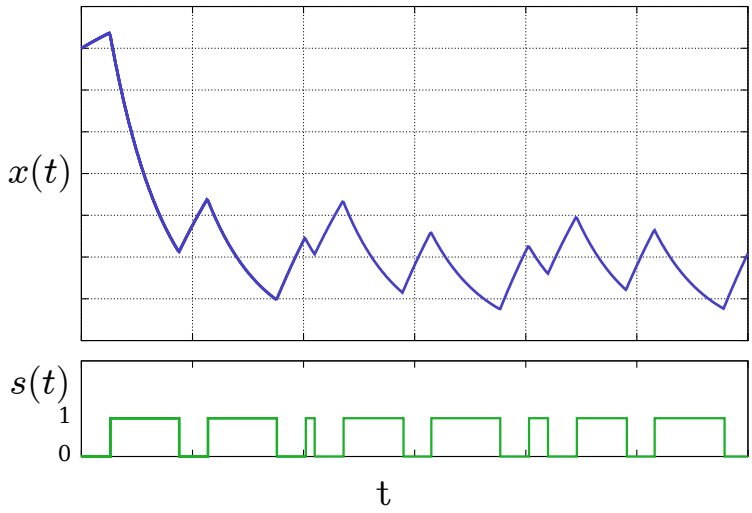


Figure 4.1: Evolution over time example of a RTPS composed by one load. The state variable decreases while the switching signal is equal to 1 and it increases when the switching signal is equal to 0.

- $d : \mathbb{R}^+ \to \mathbb{R}^n$ is a disturbance.

The notation Φ refers to the whole dynamical system defined by x, f_s, d and x_0. Figure 4.1 shows an example of the evolution over time of a RTPS.

4.1.2 Switching signal

The *cyber* component of a Real-Time Physical System concerns the scheduling of actuators activity. The distinguishing point of a RTPS is that the switching signal is generated by a real-time scheduling algorithm, such as the Earliest Deadline First (EDF) (see Section 3.2.3). Actuators form the so-called "load set", which is therefore composed by m loads. According to the periodic task model for real-time computing tasks, to each load is

assigned a tuple of timing parameters:

$$\mathcal{R}_i \doteq (T_i, D_i, C_i, P_i) \qquad i = 1, \ldots, m \tag{4.2}$$

where, similarly to what introduced in Chapter 3:

- $T_i \in \mathbb{R}^+$ is the so-called **period**, which is the minimum time frame between two consecutive request times. A **request time** $r_{i,k}$ is defined as the k-th request for activating the load; it holds $r_{i,k+1} - r_{i,k} = T_{i,k} > T_i,\ k \in \mathbb{N}$. In the periodic task model, periods are constant for all system lifetime, i.e. $T_{i,k} = T_i,\ \forall k \in \mathbb{N}$. Note that we use the word "period" to refer either to the time interval $[r_{i,k},\ r_{i,k+1}]$ or to T_i, which is a real number. The context of the sentence should identify clearly the meaning.

- $D_i \in \mathbb{R}^+ : D_i \leq T_i$ is the **relative deadline**; it defines the time frame $[r_{i,k}, r_{i,k} + D_i]$, for $k \in \mathbb{N}$, in which a load must perform its activity within each period;

- $C_i \in \mathbb{R}^+ : C_i \leq D_i$ represents the maximum **activation time** duration within each period; the actual activation time can vary from period to period and it is indicated with $C_{i,k}$;

- $P_i \in \mathbb{R}$ is a constant weighting parameter associated to the i-th load.

In traditional real-time systems, an important role is played by the concept of Worst-Case Execution Time (WCET). The WCET is the longest possible execution time of a real-time computing task. The WCET is used to perform the schedulability analysis in the worst case, while during the system behavior the actual duration of a task can be less than WCET. In

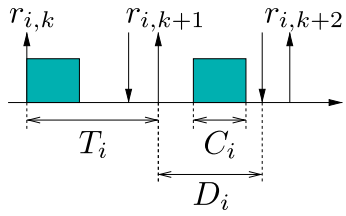


Figure 4.2: Visualization of the timing parameters of the switching signal.

RTPS, C_i plays the role of the WCET and $C_{i,k}$ is the actual activation time, which can vary from one to another job of the task.

According to the cyber component modeling based on the sporadic task model, (4.2) contains only the strictly required parameters, plus a weighting parameter. Nevertheless, a load can be described with different parameters defined by the system designer. Depending on the selected model, an adequate real-time scheduling algorithm will be used to manage the access to the limited resource by loads.

Definition 2 (RT scheduling algorithm)
A real-time scheduling algorithm $\mathcal{A}$ *is an application that maps a set of real-time parameters* $\mathcal{R}$ *to a schedule* s*, given a time unit* $\tau \in \mathbb{R}^+$ *and the value of the state variable* x*.*

$$\mathcal{A} : \{\mathcal{R}, \tau, x\} \rightarrow s \tag{4.3}$$

The real-time scheduling algorithm is executed by the **scheduler**. The time quantum τ is the smallest time interval between two scheduling action. If the scheduler model is in continuous time and the scheduling actions can happen anytime, we indicate $\tau = 0^+$ by convention meaning that τ is

infinitesimally close to zero. The algorithm is said to be **closed-loop** if it considers the actual value of the state variables x to produce the schedule; it is said **open-loop** otherwise. The value $C_{i,k}$ is determined by the online control policy at the $r_{i,k}$ request time.

The scheduler, under some conditions later discussed, generates valid switching signals. The concept of validity for switching signal is stated by the following definitions.

Definition 3 (Local validity)
Given a set of timing parameters $\mathcal{R}_i$ and a time interval between two request times $r_{i,p}$ and $r_{i,q}$ (with $q > p \geq 0$), the class $\mathcal{S}^{\mathcal{R}_i}_{[r_{i,p},r_{i,q}]}$ of all valid signals on the time interval $[r_{i,p}, r_{i,q}]$ is defined as follows:

$$\mathcal{S}^{\mathcal{R}_i}_{[r_{i,p},r_{i,q}]} := \left\{ s_i : \mathbb{R}^+ \to \mathbb{B} \quad \middle| \quad (4.5a) \text{ and } (4.5b) \right\} \tag{4.4}$$

$$\int_{r_{i,k}}^{r_{i,k}+D_i} s_i(t)\,\mathrm{d}t = C_{i,k} \quad \forall k : p \leq k < q \tag{4.5a}$$

$$s_i(t) = 0 \quad \forall t \in (r_{i,k} + D_i,\, r_{i,k+1}) \;\forall k : p \leq k < q \tag{4.5b}$$

A switching signal is said to be valid within the time interval $[r_{i,p}, r_{i,q}]$ *if it belongs to the class of validity, i.e., $s_i \in \mathcal{S}^{\mathcal{R}_i}_{[r_{i,p},r_{i,q}]}$.*

Based on Definition 3, a switching signal is valid within the time interval if it always assigns to each load an amount of activity time equal to $C_{i,k}$. Condition (4.5a) states that all the activity time within a period must be before the deadline. Condition (4.5b) states that the load must be OFF

during the time interval between a deadline and the consecutive request time.

Definition 4 (Global validity)
A switching signal s_i is said to be globally valid*, or just* valid*, if and only if it belongs to the class of validity within all time intervals. The class of globally valid signals is indicated with $\mathcal{S}_\infty^{\mathcal{R}_i}$. Formally:*

$$s_i \in \mathcal{S}_\infty^{\mathcal{R}_i} \qquad iif \qquad s_i \in \mathcal{S}_{[r_{i,0}, r_{i,q}]}^{\mathcal{R}_i},\ \forall q > 0 \tag{4.6}$$

Definitions 3 and 4 are crucial for all the analysis in the reminder of the chapter. Note that those definitions regards just switching signals and not the state variable.

The next definition refers to the system as a whole and it regards all together the individual switching signals that compose the schedule.

Definition 5 (Valid schedule)
A schedule s is said to be valid when all the switching signals s_i are globally valid.

$$s \in \mathcal{S}_\infty^{\mathcal{R}} \qquad iif \qquad s_i \in \mathcal{S}_\infty^{\mathcal{R}_i},\ \forall i \in [1,\, m] \tag{4.7}$$

A valid schedule can be generated, for instance, by a real-time scheduling algorithm as EDF or RM when the real-time parameters $\mathcal{R}$ have certain properties. For all scheduling algorithms there exists a specific schedulability test $T_{\mathcal{A}}(\mathcal{R})$ that check those properties. In general, a schedulability test is defined as follows.

Definition 6 (Schedulability test)
A schedulability test $\mathcal{T}_{\mathcal{A}}$ *for the scheduling algorithm* $\mathcal{A}$ *is defined as the following function of* $\mathcal{R}$*:*

$$\mathcal{T}_{\mathcal{A}}(\mathcal{R}) = \begin{cases} 1 & \textit{if a valid schedule exists} \\ 0 & \textit{otherwise} \end{cases} \tag{4.8}$$

To derive results in following sections, two common figures used in real-time systems are recalled here: the load utilization $U_i \doteq C_i/T_i$ (3.2) and the total utilization $U^{\text{tot}} \doteq \sum_{i=1}^{m} U_i$ (3.3). While $U_i \leq 1$ is the fraction of time in which the i-th load is active, U^{tot} represents the total fraction of activity time of the whole load set. The total utilization U^{tot} is particularly useful. In fact, it is used to perform the schedulability test of many scheduling algorithms.

4.1.3 User requirements

User requirements Ψ are a set of constraints on the physical quantities of interest. Formally:

$$x(t) \in \Psi(t) \tag{4.9}$$

where Ψ is a time-variant convex subset of the x domain, i.e. $\mathbb{R}^n$. User requirements capture the desired behavior of the system. For example, in many applications the physical quantities of interest must be bounded within a given range.

User requirements can be either "hard" or "soft". Similarly to what happen in the context of real-time computing systems, hard user requirements are those that can never be violated. Soft user requirements, instead, can be violated sometimes without compromising the correct behavior of

the system. More violations in soft user requirements happen, poorer is the quality of service.

4.1.4 Feasibility problem

According to previous definitions, a RTPS includes a dynamical system, user requirements, a set of real-time parameter and a scheduling algorithm. Formally

$$\text{RTPS} \doteq (\Phi,\ \Psi,\ \mathcal{R},\ \tau,\ \mathcal{A}) \tag{4.10}$$

While the dynamical system Φ and user requirements Ψ are bounded to the underlying physical process, the real-time parameters $\mathcal{R}$, the time quantum τ, and the scheduling algorithm $\mathcal{A}$ can be selected by the system designer. The selection should be made in order to obtain a feasible RTPS.

Definition 7 (Feasibility)
A RTPS $(\Phi, \Psi, \mathcal{R}, \tau, \mathcal{A})$ *is said to be* feasible *if and only if user requirements are satisfied by every valid schedule.*

$$\textit{feasible}\ \text{RTPS} \quad \textbf{iif} \quad s \in \mathcal{S}_{\infty}^{\mathcal{R}} \implies x(t) \in \Psi(t),\ \forall t \tag{4.11}$$

Definition 8 (Feasibility problem)
The feasibility problem *concerns the identification of suitable timing parameters* $\mathcal{R}^*$*, such that* $(\Phi, \Psi, \mathcal{R}^*, \tau, \mathcal{A})$ *is feasible.*

In other words, the RTPS feasibility problem concerns the identification of the class of valid switching signals such that user requirements are

guaranteed. This problem translates to the identification of suitable timing parameters $\mathcal{R}^*$ to drive the evolution of physical variables in compliance with user requirements.

We study the feasibility problem with a **worst-case analysis**. The methodology is based on the observation that the scheduler generates a valid switching signal among all the possible valid signals. Therefore, the analysis is performed considering the worst case signal, i.e., the signal that brings to the worst possible situation in terms of user requirements violation. This allows to assess the behavior of all other "less critical" valid switching signals. In this way the analysis is independent from the scheduling algorithm $\mathcal{A}$. In fact, the unique hypothesis on $\mathcal{A}$ is that it generates a valid schedule. So, derived results are valid whatever the scheduling algorithm $\mathcal{A}$ is.

Note that the worst case condition in terms of predictability for the switching signal model is when deadlines are equal to periods. In fact, if deadlines are less than periods than the timespan between a deadline and the following request time is always a OFF-time. Recall that we study the properties of the RTPS for any valid switching signal. In this context, the timing parameters model in which deadlines are equal to periods is the most general because the set of any valid switching signal of this model is a superset of the set of any valid switching signal of other models. Therefore, results derived for deadlines equal to periods in this prospective are valid for deadlines less than periods as well.

Once suitable timing parameters have been identified, it is possible to schedule actuators activation as if they were computing tasks in a real-time computing system.

4.2 Switched integrators

This system has been studied in (Della Vedova and Facchinetti [19], Facchinetti and Della Vedova [29]).

4.2.1 System model

The system is composed by decoupled subsystems, which are independent from each others. Each ON/OFF actuator operates on one physical subsystem, i.e. $m = n$. The dynamics of the i-th subsystem is described by Φ_i^{SI}, in which the superscript SI stands for Switched Integrators:

$$\Phi_i^{\mathsf{SI}} : \begin{cases} \dfrac{\mathrm{d}x_i(t)}{\mathrm{d}t} = f_i^{s_i(t)} = \begin{cases} -\alpha_i^{\mathrm{on}} & \text{if } s_i(t) = 1 \\ +\alpha_i^{\mathrm{off}} & \text{if } s_i(t) = 0 \end{cases} \\ \\ x_i(0) = \bar{x}_i \end{cases} \tag{4.12}$$

The quantities involved in (4.12) are:

- $t \in \mathbb{R}^+$ is the continuous time span;
- $x_i(t) \in \mathbb{R}$ is the state variable of the subsystem and represents the physical quantity of interest;
- $\bar{x}_i$ is the initial value of the state variable;
- $s_i(t) \in \mathbb{B} \equiv \{0, 1\}$ is the *operation mode* of the subsystem. It represents the activation status of the i-th load: $s_i(t) = 0$ if the load is not active at time t, and conversely, $s_i(t) = 1$ when the load is active;
- $s_i : \mathbb{R}^+ \to \mathbb{B}$ is called *switching signal* or *schedule* for the load i-th;

- $f_i^{s_i(t)} \in \mathbb{R}$ is the integrator dynamics' parameter of the subsystem i-th at time t; its value can be either $-\alpha_i^{\text{on}} \in \mathbb{R}$ or $\alpha_i^{\text{off}} \in \mathbb{R}$ depending on the value of $s_i(t)$.

User requirements

The physical quantity of interest of each subsystem requires to be bounded within a given working range Ψ_i^{mm}:

$$x_i(t) \in \Psi_i^{\mathsf{mm}} \equiv \left[x_i^{\min}, x_i^{\max}\right] \tag{4.13}$$

The superscript mm stands for min-max. An example of this kind of requirements is the internal temperature of a refrigerator, which needs to be maintained within the desired range.

4.2.2 Feasibility analysis

The analysis consider for the switching signal deadlines equal to periods, which is the "worst case" condition in terms of predictability of the signal. In fact, if deadlines are less than periods than the timespan between a deadline and the following request time is always a OFF-time. Recall that we study the properties of the RTPS for any valid switching signal. In this context, the timing parameters model in which deadlines are equal to periods is the most general because the set of any valid switching signal of this model is a superset of the set of any valid switching signal of other models. Therefore, results derived in this section are valid for deadlines less than periods as well.

Periodicity of state variable values

The first result regards the evolution of the state variable when the system is driven by a valid switching signal. In particular, Observation 1 indicates the state variable value in correspondence of request times.

Observation 1 *For a dynamical system* Φ_i^{SI} (4.12) *driven by a valid switching signal* (4.4)*, it holds:*

$$x_i(r_{i,k+1}) = x_i(r_{i,k}) - \alpha_i^{\text{on}} C_{i,k} + \alpha_i^{\text{off}} (T_{i,k} - C_{i,k}) \tag{4.14}$$

Proof. By integrating (4.12), it holds:

$$x_i(t) = x_i(t_0) - \alpha_i^{\text{on}} \int_{t_0}^{t} s_i(t)\,\mathrm{d}t + \alpha_i^{\text{off}} \int_{t_0}^{t} \neg s_i(t)\,\mathrm{d}t \tag{4.15}$$

Taking into account the property of integration of Boolean functions

$$(b-a) = \int_a^b s_i(t)\,\mathrm{d}t + \int_a^b \neg s_i(t)\,\mathrm{d}t \tag{4.16}$$

and by imposing $t = r_{i,k+1}$ and $t_0 = r_{i,k}$ in (4.15), it follows:

$$\begin{aligned} x_i(r_{i,k+1}) =& x_i(r_{i,k}) - \alpha_i^{\text{on}} \int_{r_{i,k}}^{r_{i,k+1}} s(t)\,\mathrm{d}t + \\ &+ \alpha_i^{\text{off}} \left((r_{i,k+1} - r_{i,k}) - \int_{r_{i,k}}^{r_{i,k+1}} s(t)\,\mathrm{d}t \right) \end{aligned} \tag{4.17}$$

Finally, taking into account the Definition 3 of local validity and the definition of timing parameters (4.2), the (4.14) follows by replacing the two definitions into (4.17).

Taking the previous observation into account, it is possible to establish a relation between the load utilization U_i (3.2) and the dynamics of the related physical process.

Theorem 1 (Stability for Switched Integrators RTPS)
Given a dynamical system Φ_i^{SI} (4.12) *and periodic timing parameter* $\mathcal{R}_i = (C_i, T_i)$*, if the switching signal is valid* (4.6) *then it holds* $\forall T_i$*:*

$$x_i(kT_i) = \bar{x}_i, \iff C_{i,k} = U_i^* T_i, \ \forall k \in \mathbb{N} \tag{4.18}$$

where

$$U_i^* = \frac{\alpha_i^{\text{off}}}{\alpha_i^{\text{on}} + \alpha_i^{\text{off}}} \tag{4.19}$$

Proof. Considering the the definition of load utilization U_i (3.2), the property of periodic task model $r_{i,k} = kT_i$, and the initial value of the state variable $x(0) = x_i(r_{i,0}) = \bar{x}_i$, Equation (4.14) can be rewritten as:

$$x_i(kT_i) = \bar{x}_i + \alpha_i^{\text{off}} k(T_i - U_i^* T_i) - \alpha_i^{\text{on}} k U_i^* T_i \tag{4.20}$$

$$= \bar{x}_i + kT_i(\alpha_i^{\text{off}} - (\alpha_i^{\text{off}} + \alpha_i^{\text{on}})U_i^*) \tag{4.21}$$

Equation (4.18) can be easily verified by replacing the value of U_i^* into (4.20).

Theorem 1 states that, for every load, the state variable assumes the same value $\bar{x}_i$ at every request time $r_{i,k}$ if and only if its utilization is set

as in (4.19). It also shows that, to achieve this result, the load utilization U_i^* depends only on α_i^{off} and α_i^{on}. Two key consequences derive from Observation 1 and Theorem 1. First, since a state variable assumes the same value at every request time, the analysis of global properties (i.e., for every time t) of the state variable can be performed by restricting the analysis to one period. Second, since the remaining results derived in this section are based on Theorem 1, they hold for deadlines less than periods. In fact, deadlines are never mentioned and results do not depend from the actual points in time when a load is activated within a period. Therefore, deadlines can be shortened to improve the system responsiveness without affecting the achievement of user requirements. Clearly, the shortening can be performed as far as timing constraints can be met by the scheduling algorithm. This is a relevant result since, in the analysis of traditional real-time computing systems, substantial complications arise when deadlines are allowed to be less than periods.

Bounding the maximum state variable variation

Since we are interested to bound the state variable variation within a specified range, we introduce the definition of largest variation with respect to $\bar{x}_i$ within a timespan equal to a period.

Observation 2 *The largest possible ascending and descending variation of the state variable $x_i(t)$ with respect to $\bar{x}_i$ on one period are, respectively,*

$$\delta_i^{\text{inc}} = \alpha_i^{\text{off}}(T_i - C_i) \tag{4.22}$$

and

$$\delta_i^{\text{dec}} = \alpha_i^{\text{on}} C_i \tag{4.23}$$

Those cases happen when the consecutive ascending/decreasing time is maximized within a period.

Proof. Since the value of $x_i(t)$ equals $\bar{x}_i$ at each request time (Theorem 1), we limit the analysis to the time interval $[0, T_i]$.

Let us define $\hat{t}$ as the time instant after which the state variable $x_i(t)$ can only decrease, i.e.,

$$\forall t : \hat{t} < t \leq T_i \rightarrow x_i(t) < x_i(\hat{t}).$$

Therefore, the maximum value of $x_i(t)$ must correspond to a time instant t^* such as $0 < t^* \leq \hat{t}$. The value of $x_i(\hat{t})$ can be calculated by integrating (4.12), obtaining

$$x_i(\hat{t}) = \bar{x}_i + \alpha_i^{\text{off}} \int_0^{\hat{t}} \neg s_i(t)\,\mathrm{d}t - \alpha_i^{\text{on}} \int_0^{\hat{t}} s_i(t)\,\mathrm{d}t \tag{4.24}$$

The first integral corresponds to the amount of time that $x_i(t)$ increases in the range $[0, \hat{t}]$, which is equal to $T_i - C_i$. In fact, by definition of $\hat{t}$, the range $[0, \hat{t}]$ contains all the amount of time that the state variable has a negative derivative. Therefore, C_i is the amount of time that the state variable has a positive derivative in the whole range $[0, T_i]$, and $T_i - C_i$ is the amount of time that the state variable derivative is negative. Thus, it holds

$$x_i(\hat{t}) = \bar{x}_i + \alpha_i^{\text{off}}(T_i - C_i) - \alpha_i^{\text{on}} \int_0^{\hat{t}} s_i(t)\,\mathrm{d}t \tag{4.25}$$

The proof concludes by noticing that (4.25) is composed by two constants and a negative term (the value of the integral). Therefore, the maximum

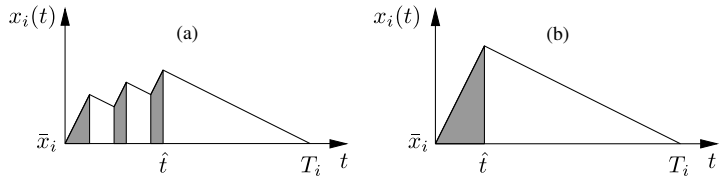


Figure 4.3: Examples of behavior of the state variable variation within a period T_i ($D_i = T_i$); right figure, in particular, represents the case when the state variable increasing variation is maximized.

value of $x_i(\hat{t})$ holds when the negative term is equal to zero. In other words, it holds

$$\forall t : 0 < t \leq \hat{t} \rightarrow s_i(t) = 0$$

involving

$$\hat{t} = T_i - C_i = t^* \tag{4.26}$$

Observation 2 states that largest variations of $x_i(t)$ with respect to $\bar{x}_i$ take place when the whole amount of activation and deactivation times are located either at the beginning or at the end of the time range $[r_{i,k}, r_{i,k}+T_i]$. For example, the largest increasing variation happens when the state variable behaves as in Figure 4.3 (b), while in Figure 4.3 (a) the variation is less than the largest possible one, since the consecutive activation/deactivation time is not maximized. In other words, the load activation suffers from preemptions[1].

[1]In traditional real-time computing systems, the term *preemption* indicates the action of temporarily interrupting a running processing activity to execute an higher priority task; the interrupted task is resumed later.

Observation 3 *Within the same hypotheses of Theorem 1, if* $U_i = U_i^*$ *then:*

$$\delta_i^{\text{inc}} = \delta_i^{\text{dec}} \doteq \Delta_i \tag{4.27}$$

Proof. Considering Theorem 1 and Observation 2, it follows:

$$\begin{aligned}
\delta_i^{\text{inc}} &= \alpha_i^{\text{off}}(T_i - C_i) && \text{(from (4.22))} \\
&= \alpha_i^{\text{off}}(T_i - U_i T_i) && \text{(since } C_i = U_i T_i\text{)} \\
&= \frac{T_i \alpha_i^{\text{on}} \alpha_i^{\text{off}}}{\alpha_i^{\text{on}} + \alpha_i^{\text{off}}} && \text{(replacing } U_i^* \text{ and rearranging)}
\end{aligned}$$

and

$$\begin{aligned}
\delta_i^{\text{dec}} &= \alpha_i^{\text{on}} C_i && \text{(from (4.23))} \\
&= \alpha_i^{\text{on}} U_i T_i && \text{(since } C_i = U_i T_i\text{)} \\
&= \frac{T_i \alpha_i^{\text{on}} \alpha_i^{\text{off}}}{\alpha_i^{\text{on}} + \alpha_i^{\text{off}}} && \text{(replacing } U_i^*\text{)}
\end{aligned}$$

The range of feasible periods

The next relevant result is recalled by Theorem 2. It allows to calculate the upper bound on the period T_i such that, if used together with the load utilization U_i^* (4.19), it guarantees that load the state variable $x_i(t)$ is maintained within the required range $[x_i^{\min}, x_i^{\max}]$.

Theorem 2 (Feasible Switched Integrators RTPS)
If timing parameters $\mathcal{R}_i^* = (C_i^*, T_i^*)$ *are set such that*

$$C_{i,k} = C_i^* = U_i^* T_i^*, \ \forall k \tag{4.28a}$$

$$T_{i,k} = T_i^*, \ \forall k \tag{4.28b}$$

and

$$T_i^* < T_i^{\text{lub}} = \min \left\{ \frac{x_i^{\max} - \bar{x}_i}{\alpha_i^{\text{off}}(1 - U_i^*)}, \frac{\bar{x}_i - x_i^{\min}}{\alpha_i^{\text{on}} U_i^*} \right\} \tag{4.29}$$

then the RTPS subsystem $\left(\Phi_i^{\mathsf{SI}}, \Psi_i^{\mathsf{mm}}, \mathcal{R}_i^*, 0^+, \mathcal{A}\right)$ *is feasible.*

Proof. Considering Observations 2 and 3, in order to keep the state variable into user requirements bounds, it must be imposed:

$$\begin{cases} x_i^{\max} \geq \bar{x}_i + \delta_i^{\text{inc}} \\ x_i^{\min} \leq \bar{x}_i - \delta_i^{\text{dec}} \end{cases} \tag{4.30}$$

By replacing the expressions of δ_i^{inc} and δ_i^{dec} together with the value of U_i^* indicated in Theorem 1 into the previous constraint, it holds:

$$\begin{cases} T_i \leq \dfrac{x_i^{\max} - \overline{x}}{\alpha_i^{\text{off}}(1 - U_i^*)} \\ T_i \leq \dfrac{\overline{x} - x_i^{\min}}{\alpha_i^{\text{on}} U_i^*} \end{cases} \tag{4.31}$$

Since both inequalities in (4.31) must hold, T_i is upper bounded by the minor of the two quantities, and (4.29) follows.

Theorem 2 allows to determine suitable values of timing parameters, i.e. U_i^* from (4.19) and T_i^* from (4.29), to achieve the requirements on physical variables, specified by Ψ_i^{mm} (4.13).

The T_i^{lub} value represents an upper bound on the range where the load period could be selected. Therefore, if needed, a shorter period could also be preferred. Such shorter value still achieves the user requirements. It is worth to discuss the implications of such a possible choice. Shorter periods correspond to shorter distances in time between two consecutive request times. Therefore, shorter periods determine a sequence of activation/deactivation in which the switching events are likely to be closer each other. This observation holds in general when a load is considered alone, i.e., it is not affected by the activations of other loads having higher priorities (i.e., preempted). In fact, in presence of more than one load, preemptions may generate a similar effect, although in this case such behavior does not emerge from the timing characteristics of a given load but arises from the interaction among load activations. The effect is to narrow down the state variable variation range around $\bar{x}_i$. Although in general this behavior may be considered as a desirable feature, a side effect needs to be taken into account, which is related with the characteristics of the physical process under control. Some types of electric loads, such as motors, do not well tolerate sequences of activation/deactivation which are too close each other, since this may have a negative impact on the actuator lifetime. Therefore, a larger state variable variation range (once the state variable is guaranteed to remain within the allowed range) can achieve a longer system lifetime.

4.3 Switched integrators with modeling errors

This type of systems has been studied in (Della Vedova and Facchinetti [19]).

The results introduced in Section 4.2 essentially consist of an open-loop control strategy whose parameters are tuned to meet the desired system constraints and requirements on the state variable. In particular, the utilization is set according to Theorem 1. However, this approach may lead to the violation of user requirements when inaccuracies are present and not properly taken into account. Inaccuracies are determined by several factors: mismatch between the physical system and the adopted model, rounding in calculations, noise or interference on the physical system. For example, when the value of α_i^{off} and/or α_i^{on} is subject to variations due to external factors (with respect to the adopted model) the results of Theorem 1 may no longer hold. In other words, the value of a state variable in correspondence to the k-th request time $r_{i,k}$ may differ from the one in $r_{i,k+1}$. To cope with the effect of uncertainties, the relationship between timing parameters and physical parameters requires a deeper analysis.

4.3.1 System model

This section extends the model presented in the previous section by introducing errors that model the uncertainties on

1. slopes α_i^{off} and α_i^{on} and
2. time quantization.

In the perspective of coping with "hard" user requirements, uncertainties will be modeled in terms of worst case conditions. For this purpose, every

parameter will be modeled with an unknown real value, which is assumed to be bounded within a given interval.

The uncertainties on state variable slopes are modeled by the parameters introduced in Definition 9.

Definition 9 (Modeling error)
The difference between the real (unknown) values of the state variable slopes, i.e. α_i^{on} *and* α_i^{off}*, and the modeled values, i.e.* $\tilde{\alpha}_i^{\text{on}}, \tilde{\alpha}_i^{\text{off}} \in \mathbb{R}$*, is bounded such that:*

$$\|\alpha_i^{\text{on}} - \tilde{\alpha}_i^{\text{on}}\| \leq \delta_i^{\text{on}} \tag{4.32a}$$

$$\left\|\alpha_i^{\text{off}} - \tilde{\alpha}_i^{\text{off}}\right\| \leq \delta_i^{\text{off}} \tag{4.32b}$$

In Definition 9, the terms $\delta_i^{\text{on}}, \delta_i^{\text{off}} \in \mathbb{R}^+$ indicate the known maximum gaps between real and modeled values.

A second source of modeling approximation is related with the quantization of real-time parameters with respect to a given time-base. Since the controller is based on a digital clock, actual scheduling actions (i.e. load activations/deactivations) can only occur at integer multiples of a time quantum $\tau \in \mathbb{R}^+$. The time quantum can be either imposed by the system, e.g. by the digital clock of the computer performing the scheduling algorithm, or it can be considered as a design parameter. In this second case, it allows to bound the minimum amount of time between two consecutive switching actions of loads. In both cases, a possible source of approximation is due to the quantization on the values of activation time C_i and period T_i with respect to the granularity introduced by τ. Errors determined by quantization arise since the values of real-time parameters

derived in Theorem 2 may not necessarily be integer multiples of the time-base τ. In presence of errors, it may happen that some requirements on the state variable variation could be violated. Quantization errors are defined by Definition 10.

Definition 10 (Quantization errors)
The differences between desired timing parameters, i.e. $\tilde{C}_i, \tilde{T}_i \in \mathbb{R}^+$, *and quantized timing parameters actually used by the controller, i.e.* $C_i, T_i \in \{0, \tau, 2\tau, 3\tau, \dots\}$, *are so-called quantization errors* δ_i^C *and* δ_i^T. *Quantization errors are less than the time quantum* τ.

$$\delta_i^C = \left\| C_i - \tilde{C}_i \right\| < \tau \tag{4.33a}$$

$$\delta_i^T = \left\| T_i - \tilde{T}_i \right\| < \tau \tag{4.33b}$$

Supposing for example that the desired timing values are rounded to the closest lower multiple of τ by the controller, it is straightforward to determine the values of quantization errors by observing that

$$\delta_i^C = \tilde{C}_i \bmod \tau$$
$$\delta_i^T = \tilde{T}_i \bmod \tau$$

It is worth to note that the effect of δ_i^T can be easily eliminated by selecting $T_i = c\,\tau$, for some $c \in \mathbb{N}$, provided that $T_i \leq T_i^{\text{lub}}$ as required by Theorem 2.

In summary, all the results presented in the previous Section 4.2 are still valid. However, in practice it is not possible to set U_i^* properly as indicated in the Theorem 1 because: α_i^{on} and α_i^{off} are unknown, C_i and T_i are subject to quantization error. A natural choice may be to set U_i^* using

the approximated values $\tilde{\alpha}_i^{\text{on}}$ and $\tilde{\alpha}_i^{\text{off}}$. As a consequence, the state variable may drift from the desired value $\bar{x}_i$ because in general $x_i(kT_i) \neq \bar{x}_i$ since U_i^* is calculated using "wrong" values. Anyhow, a bound on the maximum variation of the state variable due to illustrated approximations is provided by the following:

Observation 4 *At the k-th request time, the drift of the state variable value from its initial value is bounded to:*

$$\|x(r_{i,k}) - \bar{x}_i\| \leq k\epsilon_i \tag{4.34}$$

where

$$\begin{aligned} \epsilon_i = \tilde{C}_i(\delta_i^{\text{on}} + \delta_i^{\text{off}}) + \tilde{T}_i\delta_i^{\text{off}} + \tilde{\alpha}_i^{\text{on}}\delta_i^C + \\ + \tilde{\alpha}_i^{\text{off}}(\delta_i^C + \delta_i^T) + \delta_i^{\text{on}}\delta_i^C + \delta_i^{\text{off}}\delta_i^C + \delta_i^{\text{off}}\delta_i^T \end{aligned} \tag{4.35}$$

Proof. The proof is based on the observation that, once errors have been modeled as in Definitions 9 and 10, the terms in (4.14) provided by Observation 1 are all affected by errors. The goal is to find the value of ϵ_i, which is an upper bound on the absolute value of the error on $x(kT_i)$ by (4.34). The value of ϵ_i can be derived from (4.14) by considering the additive and multiplicative properties of uncertain values (see (Taylor [68]) for details).

The state variable value results to be increased or decreased by a maximum of ϵ_i at each subsequent request time, as illustrated in Figure 4.4. Therefore, the system can no longer be suitably controlled by using the value of timing parameters as calculated in absence of errors. This issue can not be avoided unless a proper feedback on the state variable is introduced to dynamically adapt real-time parameters according to measured

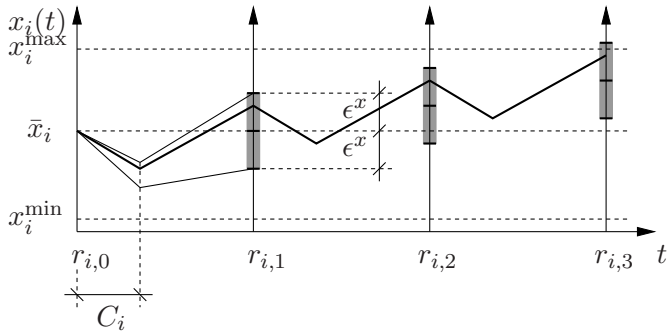


Figure 4.4: Effect of modeling errors. The state variable constantly drifts from the desired value $\bar{x}_i$, causing the violation of user requirements.

values. In other words, it is mandatory to suitably *measure* the actual value of the state variable to compensate for the effect of uncertainties.

4.3.2 Using closed-loop adaptive scheduling to cope with uncertainties

To properly control the system in presence of errors the effect of such errors must be canceled. For this purpose, a closed-loop approach is proposed to adapt the value of timing parameters $T_{i,k}$ and $C_{i,k}$ at every request time $r_{i,k}$. The adapted value will be valid for the next time frame $[r_{i,k}, r_{i,k+1})$. The idea is to measure the value of the state variable in correspondence to a request time. The measured value at time t is denoted with $\hat{x}_i(t)$. The detected gap between measured value $\hat{x}_i(r_{i,k})$ and expected value $\bar{x}_i$ is used to calculate the actual values of $C_{i,k}$ and $T_{i,k}$. Such values are set to guarantee that the state variable will fall into a bounded range in correspondence to the next request time $r_{i,k+1}$, and user requirements are met in the time frame $[r_{i,k}, r_{i,k+1})$.

Since the feedback technique is based on the measurement of the state

variable in correspondence with request times, the measurement error is firstly defined to account for the uncertainty on the measurement.

Definition 11 (Bounded measurement error)
The measurement error on the state variable $x_i(t)$ is bounded by a known constant δ_i^x, defined as

$$\|\hat{x}_i(t) - x_i(t)\| \leq \delta_i^x \tag{4.36}$$

where $\hat{x}_i(t)$ represents the measured value, while $x_i(t)$ is its unknown real value.

Considering the model for the measurement error, a theoretical result is provided to allow the compensation of errors arising from sources modeled by Definitions 9 to 11.

Theorem 3 (Robust Stability of Switched Integrator RTPS)
Given the dynamical system Φ_i^{SI} (4.12), the definitions of parameter uncertainties (4.32)-(4.33) and the error model on sensor measurements (4.36), if $\tilde{C}_{i,k}$ and $\tilde{T}_{i,k}$ are set in order to balance the following equation

$$\bar{x}_i - \hat{x}_i(r_{i,k}) = \tilde{T}_{i,k}\tilde{\alpha}_i^{\text{off}} - \tilde{C}_{i,k}\left(\tilde{\alpha}_i^{\text{on}} + \tilde{\alpha}_i^{\text{off}}\right) \tag{4.37}$$

then

$$\|x_i(r_{i,k}) - \bar{x}_i\| \leq \epsilon_{i,k} + \delta_i^x, \quad \forall k \in \mathbb{N} \tag{4.38}$$

Proof. The goal is to determine the values of $T_{i,k}$ and $C_{i,k}$ so that $x_i(r_{i,k+1}) = x_i(r_{i,k}) = \bar{x}_i$ in (4.14). However, the terms in (4.14) are affected by errors with known bounds, as stated in Definitions 9 to 11,

where known terms are $\hat{x}_i$, $\tilde{\alpha}_i^{\text{off}}$, $\tilde{\alpha}_i^{\text{on}}$. Therefore, it can not be achieved to obtain exactly $x_i(r_{i,k+1}) = \bar{x}_i$. However, it is guaranteed that $x_i(r_{i,k+1})$ will fall in an interval containing $\bar{x}_i$, as stated in (4.38). This latter is obtained simply from (4.14) by inserting the expression of the errors. Finally, since $x_i(r_{i,0}) = \bar{x}_i$ by the system model definition (4.12), (4.38) holds for every $k \in \mathbb{N}$.

In (4.38), the term $\epsilon_{i,k}$ has the same expression of ϵ_i (i.e. (4.35)) where the terms $\tilde{C}_{i,k}$ and $\tilde{T}_{i,k}$ replace $\tilde{C}_i$ and $\tilde{T}_i$, respectively. Considering that possible values for period and activation time are bounded, it is possible to bound $\epsilon_{i,k}$ as follows:

$$0 \leq \epsilon_{i,k} \leq \epsilon_i^*, \quad \forall k \in \mathbb{N}. \tag{4.39}$$

Theorem 3 provides a convenient method to achieve the desired system behavior in presence of errors. In fact, the state variable is bounded in the range $\bar{x}_i \pm (\epsilon_i + \delta_i^x)$ at each request time $r_{i,k} : \forall k \in \mathbb{N}$. See Figure 4.4 for an illustration where $\tilde{C}_{i,k}$ is changed while $\tilde{T}_{i,k}$ is kept constant.

In order to meet the user requirements Ψ_i^{mm} (4.13), Theorem 2 must be extended to take into account modeling and measurements errors and the variability of $\tilde{T}_{i,k}$ and $\tilde{C}_{i,k}$.

Theorem 4 (Feasibility with modeling errors)
If (adaptive) timing parameters $\mathcal{R}_i^*$ are set such that $\forall k$ the values of $\tilde{T}_{i,k}$ and $\tilde{C}_{i,k}$ respect (4.37) and

$$\begin{cases} x_i^{\min} \leq \hat{x}_i(r_{i,k}) - \delta_i^x - (\tilde{\alpha}_i^{\text{on}} + \delta_i^{\text{on}})C_{i,k} \\ x_i^{\max} \geq \hat{x}_i(r_{i,k}) + \delta_i^x + (\tilde{\alpha}_i^{\text{off}} + \delta_i^{\text{off}})(T_{i,k} - C_{i,k}) \end{cases} \tag{4.40}$$

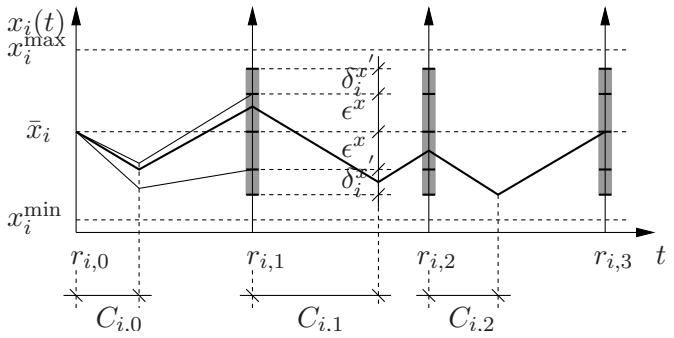


Figure 4.5: Effect of modeling errors compensation. The state variable at every request time $r_{i,k}$ is maintained within a bounded range centered in $\bar{x}_i$ by properly setting the value of $C_{i,k}$.

then the RTPS subsystem $\left(\Phi_i^{\mathsf{SI}}, \Psi_i^{\mathsf{mm}}, \mathcal{R}_i^*, \tau, \mathcal{A}\right)$ *is feasible.*

Proof. Taking into account Observation 2, the maximum (minimum) possible value of the state variable x_i, depending on the scheduling signal s_i, occurs when the activation time is concentrated at the end (beginning) of the time frame defined by one period. Thus it must be guaranteed that $\forall k$:

$$\begin{cases} x_i^{\min} \leq x_i(r_{i,k}) - \alpha_i^{\text{on}} C_{i,k} \\ x_i^{\max} \geq x_i(r_{i,k}) + \alpha_i^{\text{off}} (T_{i,k} - C_{i,k}) \end{cases} \tag{4.41}$$

Considering the worst case scenario for errors that affect the terms in (4.41), (4.40) follows directly and the thesis of the Theorem holds since all conditions are valid $\forall k$.

From (4.36), (4.38) and (4.39), it follows that the gap between measurement and expected value of the state variable at the k-th request time is

bounded:

$$\|\hat{x}_i(r_{i,k}) - \bar{x}_i\| \leq \epsilon_i^* + 2\delta_i^x, \quad \forall k \in \mathbb{N} \tag{4.42}$$

Thanks to (4.42) and considering the constraints derived in (4.40), upper bounds T_i^{lub} and C_i^{lub} can be calculated, respectively on period and activation time, so that user requirements are satisfied.

The opportunity to select appropriate periods and activation times at every request time suggests some interesting considerations related to the guarantee of timing constraints. In fact, the utilization $U_{i,k} = C_{i,k}/T_{i,k}$ ranges in an interval from 0% (i.e. when $C_{i,k} = 0$ and $T_{i,k} > 0$, which are possible values according to the constraints) to 100% (for some $C_{i,k} = T_{i,k} > 0$). From (4.37), it is possible to derive the expression of $U_{i,k}$ in the $[r_{i,k}, r_{i,k+1}]$ time frame as follows:

$$U_{i,k} = \frac{C_{i,k}\tilde{\alpha}_i^{\text{off}}}{C_{i,k}(\tilde{\alpha}_i^{\text{off}} + \tilde{\alpha}_i^{\text{on}}) + \bar{x}_i - \hat{x}_i(r_{i,k})} \tag{4.43}$$

or, equivalently:

$$U_{i,k} = \frac{T_{i,k}\tilde{\alpha}_i^{\text{off}} + \hat{x}_i(r_{i,k}) - \bar{x}_i}{T_{i,k}(\tilde{\alpha}_i^{\text{off}} + \tilde{\alpha}_i^{\text{on}})} \tag{4.44}$$

To simplify the presentation of subsequent results, two equivalent expressions are provided, respectively, as a function of $C_{i,k}$ (4.43) and $T_{i,k}$ (4.44). In order to apply existing utilization-based schedulability tests, it is possible to set a constant activation time $C_{i,k} = C_i \leq C_i^{\text{lub}}, \forall k$, and set periods $T_{i,k}$ according to the result of Theorem 3. In this case, taking into account equations (4.42) and (4.43), the highest possible utilization can be expressed as follows.

$$U_i \doteq \sup_{k \in \mathbb{N}} U_{i,k} = \frac{C_i\tilde{\alpha}_i^{\text{off}}}{C_i(\tilde{\alpha}_i^{\text{off}} + \tilde{\alpha}_i^{\text{on}}) - \epsilon_i^* - 2\delta_i^x} \tag{4.45}$$

The utilization derived in (4.45) can be used to test the schedulability of the load set.

4.3.3 Comments on results

To summarize, to compensate for the effects of uncertainties the actual values of $T_{i,k}$ and $C_{i,k}$ are adapted depending to the gap between the measured value of the state variable $\hat{x}_i$ and its expected value $\bar{x}_i$. While there is the possibility to simultaneously change both period and activation time, a simpler solution is to keep constant one of the two parameters and change the other one.

The option to maintain a constant period T_i and to change the activation time $C_{i,k}$ makes the resulting system identical to the well known task model with strict periods. In this case, the longest possible activation time plays the role of the Worst Case Execution Time (WCET) in the traditional task model. It is straightforward to show that the longest possible activation time occurs when the measured value of the state variable is equal to the highest possible value. The corresponding utilization can be calculated using (4.44), and such value can be used for the schedulability test.

The other option is to maintain a constant activation time C_i and to change the period $T_{i,k}$ at each request time. This option makes the resulting system working as the traditional sporadic task model, where T_i represents the minimum time frame between two consecutive request times. In this case, the shortest possible period will occur when the measured value of the state variable is equal to the lowest possible value, and the utilization is expressed by (4.45). The option to dynamically change the period within

an interval given by:

$$T_{i,k} \in \left[\frac{C_i}{U_i},\ \frac{C_i(\tilde{\alpha}_i^{\text{off}} + \tilde{\alpha}_i^{\text{on}}) + \epsilon_i^* + 2\delta_i^x}{\tilde{\alpha}_i^{\text{off}}}\right] \tag{4.46}$$

suggests the possibility to apply techniques like the elastic scheduling (Buttazzo et al. [8]). An interesting result could be to formally put into relation the value of the elastic coefficient to physical system parameters. Depending on the system characteristics and requirements, the designer can choose the most convenient option.

4.3.4 Simulation examples

A first example shows the effectiveness of the proposed closed-loop technique in compensating the effect of uncertainties. In the example, one generic electric device acts on a subsystem. System parameters, affected by uncertainties, are: $\alpha^{\text{on}} = 2.0 \pm 5\%$, $\alpha^{\text{off}} = 1.0 \pm 2\%$, $\bar{x}_i = 0$, $\delta_i^x = 0.1$. User requirements define a working range of $\Psi^{\mathsf{mm}} = [-1.0, 1.0]$ for the state variable, while the time granularity is $\tau = 0.07$. Considering to control the device activation with the open-loop technique recalled in Section 4.2, the value of the timing parameters derived from (4.19) and (4.29) are: $U^* = 0.33$, $T^{\text{lub}} = 1.50$, $T^* = 1.47$. Considering instead to control the device with the closed-loop technique, and in particular by fixing a period T and varying the activation time C_k, the timing parameters, derived from (4.35)-(4.40), are: $C^{\text{lub}} = 0.49$, $T = 1.33$, taking into account that $\delta^{\text{on}} = 0.1$.

Figure 4.6 shows the behavior of both open-loop and closed-loop techniques. The simulation has been done using $\alpha^{\text{on}} = 1.9$ and $\alpha^{\text{off}} = 1.0$, i.e., respectively the worst and best cases for the tolerance. The figure illus-

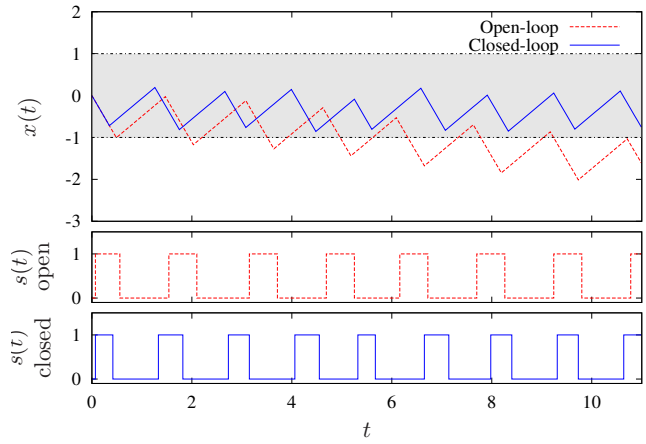


Figure 4.6: Example of scheduling of one load in presence of uncertainties on physical parameters using the open-loop and closed-loop approaches. The closed-loop scheduling successfully keeps the state variable within the desired bounds.

trates how the physical variable steadily drifts away in absence of feedback, while it is kept within the working range by the closed-loop method.

4.4 Switched integrators with stochastic disturbance

This type of systems has been studied in (Facchinetti and Della Vedova [30]).

4.4.1 System model

The system dynamics recalls the one presented in Section 4.2 on page 80, with an additional disturbance signal. The system is composed by decoupled subsystems, which are independent from each others. Each ON/OFF actuator operates on one physical subsystem, i.e. $m = n$. The dynamics of the i-th subsystem is described by (4.12).

The variation of a state variable is described by the continuous switched hybrid dynamical system Φ_i^{SID} (superscript SID stands for Switched Integrators with Disturbance), which is defined as follows:

$$\Phi_i^{\mathsf{SID}} : \begin{cases} \dot{x}_i(t) = f_{s_i(t)} + d_i(t), \quad x_i(0) = x_{i,0} \\ d_i \sim \mathrm{WGN}(0, \sigma_{d_i}^2) \end{cases} \tag{4.47}$$

System Φ_i^{SID} is equal to Φ_i^{SI} (4.12) except that it includes the stochastic additive disturbance d_i. The quantities involved in (4.47) are:

- $t \in \mathbb{R}^+$ is the continuous time span;
- $s_i(t) \in \mathbb{B} \equiv \{0, 1\}$ is the *operation mode* of the i-th load; each load can be active (i.e. $s_i(t) = 1$) or inactive (i.e. $s_i(t) = 0$);
- $s : \mathbb{R}^+ \to \mathbb{B}^n = [s_1 \ \ldots \ s_n]$ is called *switching signal* or *schedule*;

- $f_{s_i(t)}$ is a constant value representing the system dynamics. Its value depends on the switching signal s_i as follows:

$$f_{s_i(t)} = \begin{cases} -\alpha_i^{\text{on}} & \text{if } s_i(t) = 1 \\ +\alpha_i^{\text{off}} & \text{if } s_i(t) = 0 \end{cases} \tag{4.48}$$

  where $\alpha_i^{\text{on}} > 0$, $\alpha_i^{\text{off}} > 0$.

- d_i is a disturbance modeled as a white Gaussian noise with zero expectation and known variance $\sigma_{d_i}^2$.

The system Φ_i^{SID} can be seen as a Wiener process (Karatzas and Shreve [36]). The term Φ^{SID} refers to the whole dynamical system defined by x, f_s and x_0. Without loss of generality, the noise expectation is assumed to be null. The case of a non-null noise expectation can be brought back to the modeled case by suitably modifying the values of $f_{s_i(t)}$.

Each state variable x_i linearly increases with a slope defined by α_i^{off} when the i-th actuator is inactive (i.e., $s_i(t) = 0$), while it linearly decreases with slope defined by $-\alpha_i^{\text{on}}$ when the actuator is active. Notice that the decision to associate a decreasing state variable with an active load and vice-versa does not affect the generality of the problem statement and its solution. Despite its simplicity, the considered dynamical system can suitably approximate more complex systems, especially when working ranges associated with user requirements are sufficiently narrow with respect to time constants, as done in (Callaway and Hiskens [10], Lu and Shahidehpour [48], Maly and Kwan [50]).

User requirements

For this system model, user requirements in the following form are considered:

$$x_i(t) \in \Psi_i^{\mathsf{mmt}}(t) \equiv \left[x_i^{\min}(t),\, x_i^{\max}(t)\right] \tag{4.49a}$$

$$x_i^{\min}(t) = \bar{x}(t) - \Delta_i, \quad x_i^{\max}(t) = \bar{x}(t) + \Delta_i \tag{4.49b}$$

In these terms, $\bar{x}(t)$ can be seen as a set point and Δ_i is a tolerance around it. The region defined by user requirements will also be referred as *working range*. User requirements capture the desired behavior of the system. For example, when the physical quantity of interest is a temperature, as for HVAC systems, that quantity is usually required to be bounded within a given range.

Set point and working range are function of t. Their value are allowed to change over the time. This fact leads to consider transient behaviors. A transient takes places when a state variable must reach a new set point, after the working range has changed. Figure 4.10 shows an example of this case. Another situation when a transient may take place is at the system startup, when the state variable initial value is outside the working range. Malfunctioning situations, when the lack of proper control causes a violation of user requirements, can be brought back to a startup event. In this case, there will be a time instant when the faulty condition is solved and regular operations start again. From the state variable viewpoint, such time instant can be considered as a startup instant. On the other hand, we assume that a set point will remain unchanged at least until the completion of a transient. The evaluation of transient behaviors is done in Section 4.5.

4.4.2 Feasibility analysis

Recall that for the analysis switching signal is assumed to have a general unknown behavior, while belonging to the class of signals corresponding to a valid schedule defined by the properties stated in the "Switching signal" section on page 72, and in particular Definition 4. Therefore, the system analysis holds for any switching signal in that class, i.e., it is independent from the specific scheduling pattern.

First, the state variable behavior is studied in correspondence of request times, since switching signal properties are defined between two consecutive request times.

Observation 5 *For a dynamical system* Φ_i^{SID} (4.47) *driven by switching signal valid within the time interval* $[r_{i,k},\, r_{i,k+1}]$ (4.4)*, it holds:*

$$x_i(r_{i,k+1}) = x_i(r_{i,k}) + \alpha_i^{\text{off}}(T_{i,k} - C_{i,k}) - \alpha_i^{\text{on}} C_{i,k} + e_{i,k} \tag{4.50a}$$

$$e_{i,k} \sim N\left(0, T_{i,k}\sigma_{d_i}^2\right) \tag{4.50b}$$

where $e_{i,k}$ *is a random Gaussian variable modeling the effect of the disturbance.*

Proof. The proof is analogue to the one of Observation 1, taking into account the basic properties of the white Gaussian noise.

From basic properties of the Weiner process, the effect of the disturbance, i.e. the uncertainty on the value of $x_i(t)$, increases linearly with time. In fact it holds $\text{Var}[x_i(t)] = t\sigma_{d_i}^2$. Therefore, in an open loop control strategy, when the values of period and activation time are set at the first

request time and kept unchanged, i.e. $T_{i,k} = T_i$ and $C_{i,k} = C_i \ \forall k$, for a time t large enough the state variable will be almost certainly far from the desired value.

Closed-loop adaptive scheduling

In order to cope with the disturbance highlighted in Observation 5, it is necessary to introduce a closed-loop control strategy. Similarly to what it has been shown in Section 4.3, the proposed strategy is to suitably set the value of $C_{i,k}$ at each request time $r_{i,k}$ taking into account the measurement $\hat{x}_i(t)$ of the state variable $x_i(t)$. Note that it is easy to show that the dual strategy, in which C_i is constant and $T_{i,k}$ is set at each request time, would lead to the same result.

The model of the measurement error is defined as follow:

Definition 12 (Gaussian measurement error)
The measurement error on the state variable $x_i(t)$ is affected by an uncertainty defined as

$$\hat{x}_i(t) = x_i(t) + m_i, \quad m_i \sim N(0, \sigma^2_{m_i}) \tag{4.51}$$

where m_i is a random Gaussian variable with zero expectation and a variance $\sigma^2_{m_i}$.

Considering the measurement error model, a theoretical result is provided to allow the compensation of disturbance.

Theorem 5 (Adaptive activation time)
Given the system model (4.47), the measurement error model (4.51), and periodic timing parameters $\mathcal{R}_i = (C_i, T_i)$, if the switching signal is valid

and $\forall k$

$$C_{i,k} = \frac{T_i \alpha_i^{\text{off}} - \bar{x}_i + \hat{x}_i(r_{i,k})}{\alpha_i^{\text{on}} + \alpha_i^{\text{off}}} \tag{4.52}$$

then $\forall k > 0$ *the state variable has the following properties:*

$$\mathrm{E}[x_i(r_{i,k})] = \bar{x}_i \tag{4.53a}$$

$$\mathrm{Var}[x_i(r_{i,k})] = \sigma_{m_i}^2 + T_i \sigma_{d_i}^2 \tag{4.53b}$$

Proof. Both equation (4.53a) and (4.53b) can be derived by replacing the expression of $C_{i,k}$ provided by (4.52) within (4.50a). The result is

$$x_i(r_{i,k+1}) = \bar{x}_i + x_i(r_{i,k}) - \hat{x}_i(r_{i,k}) + e_{i,k}.$$

By the definition of measurement error (4.51), the expected value of this latter is the (4.53a) since $\mathrm{E}[\hat{x}(r_{i,k})] = x_i(r_{i,k})$. Equation (4.53b) holds since the covariance between $\hat{x}_i$ and $e_{i,k}$ is null.

Theorem 5 states that the uncertainty (i.e. the variance of x_i in (4.53b)) on the state variable do not increase indefinitely with the time. In particular, at each request time the variance is constant. The result of Theorem 5 provides the actual control rule to compensate the uncertainty, which requires to calculate the value of $C_{i,k}$ at every request time $r_{i,k}$. Such value will be used as activation time during the next period.

Theorem 5 indicates the rule to set $C_{i,k}$ at a given $r_{i,k}$ request time in order to obtain an expected value of the state variable at $r_{i,k+1}$ equal to the desired value $\bar{x}_i$, taking into account the measurement $\hat{x}_i(r_{i,k})$. Since the term $\hat{x}(r_{i,k})$ in (4.52) is a random Gaussian variable, the term $C_{i,k}$

is a random Gaussian variable as well. From basic properties of random variables combination, the following properties of $C_{i,k}$ can be derived:

$$C_{i,k} \sim N(\mu_{C_i}, \sigma^2_{C_i}) \tag{4.54a}$$

$$\mu_{C_i} = \frac{T_i \alpha_i^{\text{off}}}{\alpha_i^{\text{on}} + \alpha_i^{\text{off}}} \tag{4.54b}$$

$$\sigma^2_{C_i} = \frac{\sigma^2_{m_i} + T_i \sigma^2_{d_i}}{(\alpha_i^{\text{on}} + \alpha_i^{\text{off}})^2} \tag{4.54c}$$

Term $C_{i,k}$ in (4.54a) can potentially take any real value, since $\sigma^2_{C_i} \geq 0$,. However, the control policy must enforce that $C_{i,k} \in [0, C_i]$ for schedulability reasons, since C_i is used as worst case value in the schedulability test. The need to limit the value of $C_{i,k}$ causes the expected value of the state variable at the next request time to differ from the desired $\bar{x}_i$ value.

Considering C_i as a design parameter, it must be chosen within the interval $C_i \in [\mu_{C_i}, T_i]$. In fact, the C_i must be less or equal than T_i for schedulability reasons, otherwise no real-time algorithm would be able to generate a valid schedule. On the other hand, $C_i \geq \mu_{C_i}$ is required to allow the controller to select a $C_{i,k} \geq \mu_{C_i}$. Otherwise, it would not be possible to counterbalance a potential drift which would lead to an unbounded increase of the state variable. For choosing C_i, a possible suitable option that limits the probability of having $C_{i,k} > C_i$ to almost 2% is

$$C_i = \min\left(\mu_{C_i} + 2\sigma^2_{C_i},\ T_i\right). \tag{4.55}$$

Figure 4.7 shows this aspect.

The selection of C_i represents a trade-off between larger values, suitable to limit the probability that $C_{i,k} > C_i$, and shorter values, which

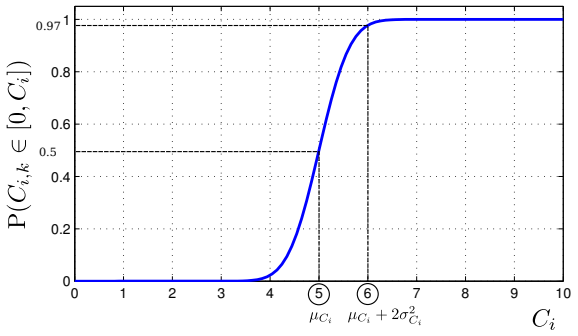


Figure 4.7: The probability that any value of the random variable $C_{i,k}$ is within the range $[0, C_i]$, as a function of the timing parameter C_i. The plot is obtained with the following parameters: $\alpha_i^{\mathrm{on}} = \alpha_i^{\mathrm{off}} = \sigma^2_{m_i} = 1$, $\sigma^2_{d_i} = 0.1$, $T_i = 10$.

are desirable to reduce the total utilization, thus having the possibility to concurrently schedule more loads.

Theorem 2 on page 87 proves that, in absence of disturbance, if the period is set such as $T_i \leq T_i^{\mathrm{lub}}$ where

$$T_i^{\mathrm{lub}} = \Delta_i \frac{(\alpha_i^{\mathrm{on}} + \alpha_i^{\mathrm{off}})}{\alpha_i^{\mathrm{on}} \alpha_i^{\mathrm{off}}} \tag{4.56}$$

then the user requirements are satisfied for every valid switching signal. This value is calculated considering the worst-case switching signal, which occurs when the load activation is either all crammed at the beginning or at the end of the time frame between two consecutive request times.

The disturbance model discussed in this section, by the nature of the considered white Gaussian noise, does not allow a guarantee of user requirements satisfaction. While it can be easily shown that a larger T_i leads

to a higher probability to exceed the user requirements, a formal study of such probability as a function of T_i is more complicated, and requires a characterization in terms of probabilities of the switching signal. For this reason, a statistical evaluation is done in Section 4.4.3, to assess the impact of uncertainties and parameter selection on user requirements satisfaction.

4.4.3 Statistical evaluation

This section reports a statistical evaluation of the impact of disturbance and selection of timing parameters on the achievement of user requirements.

Parameters have been randomly selected in uniform distributions within the following ranges: $n \in [3, 100]$, $\alpha_i^{\text{on}} \in [0.1, 10]$, $\alpha_i^{\text{off}} \in [0.02, 2.5]$. Other parameters are $\bar{x}_i = 0$, $\Delta_i = 1$, $\sigma_{m_i} = 0$. Each graph is the result of 500 schedules generate by the random parameters. The EDF scheduling algorithm has been used for all simulations.

Figure 4.8 shows the impact of disturbance d_i (see (4.47)) on the user requirements satisfaction. Each plot corresponds to one load in one of the schedules. It shows the percentage of time in which x_i is outside the feasible region during the schedule. Each schedule has been calculated for $\bar{t} = 1000$ time units with a simulation tick of 0.001 time unit. The variance of the disturbance is selected from a uniform distribution in the range $\sigma_{d_i}^2 \in [0, 10]$. The period is set as $T_i = T_i^{\text{lub}}$. Plots have been divided in two groups: loads having relatively larger periods ($T_i > 2.8$) and shorter ones ($T_i \leq 2.8$), where 2.8 is the average among all assigned periods. It can be observed that the average amount of time of user requirements violation increases linearly with $\sigma_{d_i}^2$. Loads having large periods ($T_i > 2.8$ in this plot) are likely to violate the user requirements more often.

Figure 4.9 shows how the choice of T_i affects the user requirements

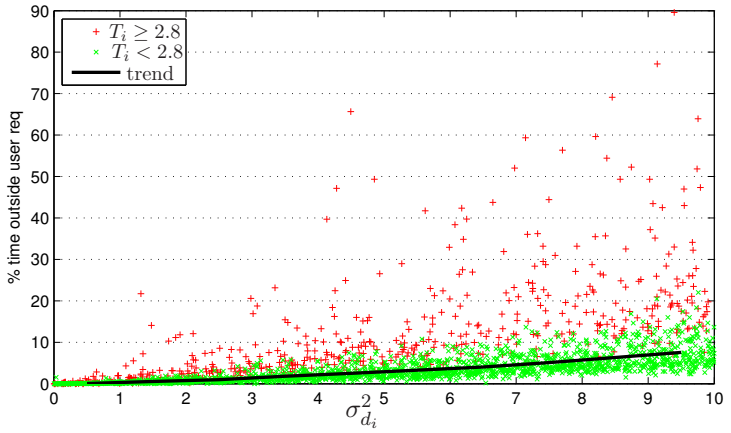


Figure 4.8: Percentage of the time in which x_i is outside the user requirements as function of the disturbance variance. The period is chosen such that $T_i = T_i^{\text{lub}}$.

satisfaction. The error term ϵ_i is defined as follows:

$$\epsilon_i \doteq \int_0^{\bar{t}} (\text{st}(x_i(t) - x_i^{\max}))^2 + (\text{st}(x_i^{\min} - x_i(t)))^2 \text{d}t \tag{4.57}$$

where $\text{st}(x) = x$ if $x > 0$ and $\text{st}(x) = 0$ if $x \leq 0$. The variance $\sigma^2_{d_i}$ is set to 1 for every load. It can be observed that the quadratic error increases as a function of the ratio T/T^{lub}, as $T > T^{\text{lub}}$. When $T < T^*$, the error is negligible.

In both figures, the black solid line represents the median of all samples contained within an abscissa sub-interval having width equal to 1/10 of the whole interval on the x axis.

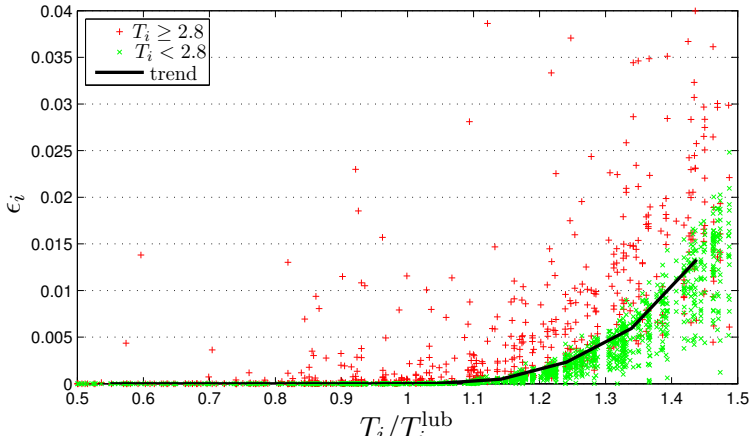


Figure 4.9: Error ϵ_i (see (4.57)) varying the period T_i as ratio with T_i^{lub}.

4.4.4 Comments on results

The fundamental observation is that, in case of constant dynamics, the result of Theorem 5 holds independently from the time intervals when a load is activated within a period. In other words, the behavior of a state variable in correspondence of request times is independent from the schedule within each period, provided that the load is activated for exactly $C_{i,k}$ time unit. This means that scheduling-related aspects such as preemptions, jitter, or response time, do not affect the system behavior in terms of user requirements satisfaction. For the same reason, this behavior does not change when deadlines are less than periods. Therefore, deadlines can be adjusted to cope with application timing constraints or to achieve the feasibility without affecting the state variable behavior. This latter must be considered a non-obvious result since, in traditional real-time computing systems, substantial complications arise when deadlines are less than

periods.

Another observation regards the impact of the period selection. Shorter periods determine a sequence of activations/deactivations with switching events that are likely to be closer each other. This observation holds when a load is considered alone, i.e., it is not affected by the activations of other loads having higher priorities (i.e., preempted). In fact, in presence of more than one load, preemptions may generate a similar effect (frequent switching on/off), although in this case such behavior does not emerge from the timing characteristics of a given load but arises from the interaction among loads. In both cases, the effect is to limit the maximum actual variation range of the state variable around the value of $\bar{x}_i$. Although in general this is a desirable behavior, it may present a side effect related with the characteristics of the physical process under control. Some types of loads, such as electric motors, do not well tolerate sequences of activation/deactivation which are too close each other, since this may have a negative impact on their lifetime[2]. On the other hand, larger periods lead to larger state variables variations. However, once a state variable is guaranteed to remain within the desired range, larger periods can achieve a longer system lifetime.

[2]Every time a motor is warmed-up, it absorbs much more current than during steady operations. This increases the device temperature, leading to a quicker wearing.

4.5 Notes on transients for switched integrators

In this context, a transient is a timespan during which the system moves from a working range to another. This situation can happen, e.g., during an initialization phase when the control system starts driving the physical system. In this case the system can be in an initial state, which is outside the desired working range: i.e., $x_0 \notin \Psi(0)$. Another case when a transient takes place is when the user requirements change, i.e. $\Psi(t_1) \neq \Psi(t_2)$ for $t_1 < t_2$. In both situations above the transient ends once the control scheme has driven the state variable back into the working range.

Section 4.2 shows that, in absence of uncertainties on the state variable, when timing parameters C_i and T_i are properly set, the state variable assumes the same value at every request time. Therefore, given the value of T_i, transients can be managed by changing the value of the activation time $C_{i,k}$ during each period, likewise the proposed method for compensating disturbances (see Figure 4.5). In this case, the maximum activation time C_i must set larger than the value that guarantees the invariability of the state variable between request times. So, a necessary condition to make the system able to manage transients is having a closed-loop system.

Let us assume that the state variable has value X_i at some request time $r_{i,k}$. Supposing the transient takes place when $X_i > x_i^{\max}$, it can be managed by activating the load for its full activation time in each period (i.e., $C_{i,k} = C_i$) until the set-point is reached. Notice that, if the transient starts below the working range, i.e., $X_i < x_i^{\min}$, the same result can be obtained by preventing the activation of the load (i.e., $C_{i,k} = 0$) until the set-point is reached. The time required to reach the set-point during a transient depends on timing parameters and state variable slopes, and can be calculated with simple geometrical considerations. Assuming $X_i > x_i^{\max}$,

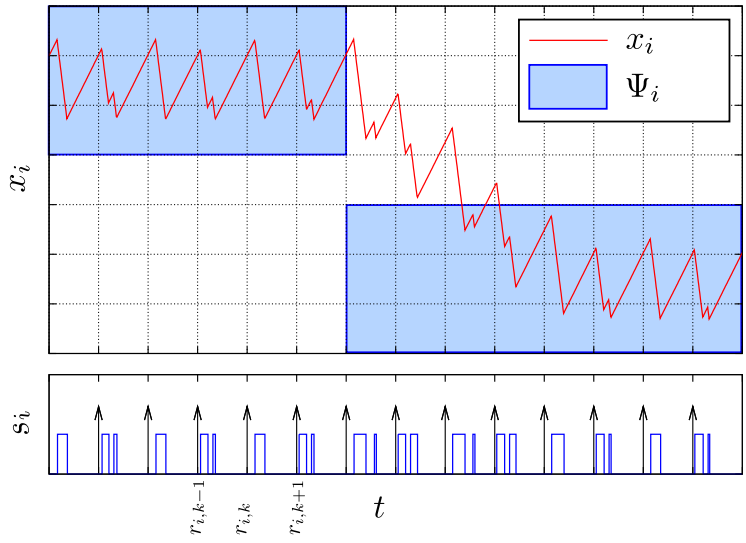


Figure 4.10: Example of Real-Time Physical System with switched integrators dynamics. The $x(t)$ physical value decreases linearly when the resource is scheduled for execution, and it increases linearly otherwise. The figure shows a possible behavior in correspondence of a transient.

at every subsequent request time the state variable decreases of $\delta = (\alpha_i^{\text{off}} + \alpha_i^{\text{on}})C_i - \alpha_i^{\text{off}} T_i$. A number of periods equal to $N = \lceil (X_i - \bar{x}_i)/\delta \rceil$ will be required to reach the set-point $\bar{x}_i$. Thus the time required for the transient will be $t^{\text{trans}} \leq NT_i$.

Figure 4.10 shows an example of transient.

4.6 Switched affine system

This systems have been studied in (Della Vedova and Facchinetti [20], Della Vedova et al. [21]).

4.6.1 System model

The dynamic behavior of a physical process is described by a single-state affine switched hybrid system in continuous time. In the state-space representation, the dynamics of the generic i-th subsystem is:

$$\Phi_i^{\mathsf{SA}} : \begin{cases} \dfrac{\mathrm{d}x_i(t)}{\mathrm{d}t} = A_i^{s_i(t)} x_i(t) + F_i^{s_i(t)} \\ x_i(0) = x_{i,0} \end{cases} \tag{4.58}$$

where:

- $t \in \mathbb{R}^+$ is the continuous time span;
- $x_i(t) \in \mathbb{R}$ is the state variable of the i-th subsystem and represents the physical quantity of interest;
- $x_{i,0}$ is the initial value of the state variable;
- $s_i(t) \in \mathbb{B} \equiv \{0, 1\} \equiv \{\text{off}, \text{on}\}$ is the *operation mode* of the subsystem; $s_i(t)$ represents the activation status of the i-th load, being $s_i(t) = 0$ if the load is not active at time t, and conversely, $s_i(t) = 1$ when the load is active;
- $s_i : \mathbb{R}^+ \to \mathbb{B}$ is the *switching signal* of the i-th load;
- $A_i^{s_i(t)} \in \mathbb{R}$ and $F_i^{s_i(t)} \in \mathbb{R}$ are the terms of the dynamics, associated to the operation mode $s_i(t)$.

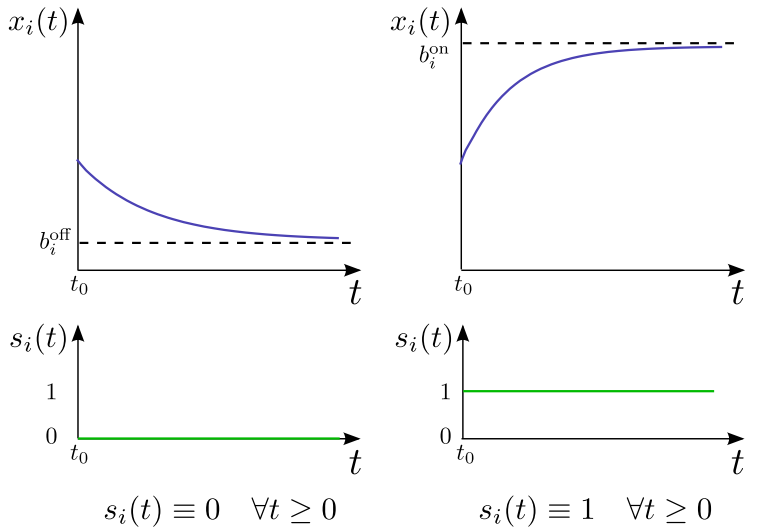


Figure 4.11: Example of asymptotic evolution over time of system (4.58) when the switching signal is a constant function.

The model describes a system where the evolution over the time of a state variable depends on the activation status of the corresponding load. Only stable subsystems are considered in this paper, i.e. $A_i^{\text{on}} < 0$ and $A_i^{\text{off}} < 0$. For those systems, there exists an asymptotic value in the evolution of $x_i(t)$ when the subsystem is driven by a switching signal s_i having constant value. This asymptotic value is $b_i^{\text{on}} \doteq -F_i^{\text{on}}/A_i^{\text{on}}$ when s_i is steadily on, and $b_i^{\text{off}} \doteq -F_i^{\text{off}}/A_i^{\text{off}}$ when s_i is steadily off. Without loss of generality, it can be assumed that $b_i^{\text{on}} > b_i^{\text{off}}$.

Figure 4.11 shows an example of the evolution over time of the system in which asymptotes are highlighted.

User requirements

User requirements are a set of constraints on the physical quantities of interest. They capture the desired behavior of the physical process. User requirements considered here are stated such as the physical quantity of interest of each subsystem requires to be bounded within an invariant working range Ψ_i^{mm} (recall Equation (4.13) on page 81): As a necessary condition for having achievable user requirements, it must hold $\Psi_i^{\mathsf{mm}} \cap [b_i^{\text{off}}, b_i^{\text{on}}] \neq \emptyset$. This kind of requirements is rather common in practical applications. For instance, they may encapsulate the requirements on the internal temperature of a refrigerator, which needs to be maintained within the desired range.

4.6.2 Feasibility analysis

This section derives interesting properties regarding the relationship between real-time and physical parameters. Since physical subsystems are independent from each others, the analysis will be carried out for one load only. For the sake of readability, the index "i" has been omitted from the equations in this section.

Definition 13 (Limit-case switching signals)
Given a time interval between two consecutive request times $[r_k, r_{k+1}]$ *and a set of timing parameters* $\mathcal{R}$*, the switching signals* $s^{\Downarrow} : [r_k, r_{k+1}] \to \mathbb{B}$ *and* $s^{\Uparrow} : [r_k, r_{k+1}] \to \mathbb{B}$ *are defined as follows:*

$$s^{\Downarrow}(t) = \begin{cases} 1 & \forall t \in [r_k, r_k + C) \\ 0 & \forall t \in [r_k + C, r_{k+1}) \end{cases} \tag{4.59}$$

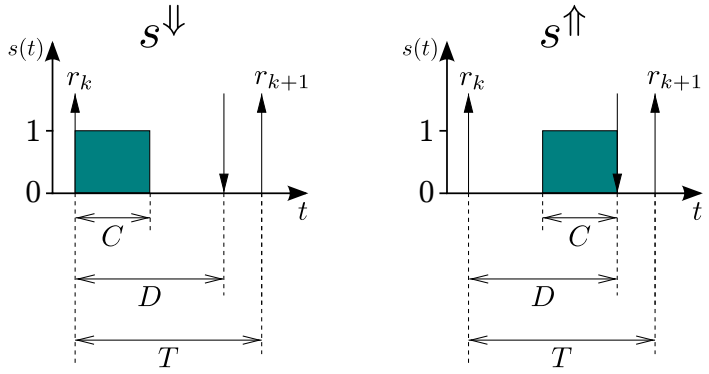


Figure 4.12: Graphical representation of switching signals $s^{\Uparrow}$ and $s^{\Downarrow}$, introduced by Definition 13

$$s^{\Uparrow}(t) = \begin{cases} 0 & \forall t \in [r_k,\, r_k + D - C) \cup [r_k + D,\, r_{k+1}) \\ 1 & \forall t \in [r_k + D - C,\, r_k + D) \end{cases} \tag{4.60}$$

Figure 4.12 depicts the previous defined switching signals. Given such definitions, it is worth noting the following

Observation 6 *Signals defined in Definition 13 are valid within the given time interval:*

$$s^{\Downarrow},\, s^{\Uparrow} \in \mathcal{S}^{\mathcal{R}}_{[r_k, r_{k+1}]} \tag{4.61}$$

Proof. To prove the previous claim, simply note that validity conditions (4.5) are satisfied by $s^{\Uparrow}$ and $s^{\Downarrow}$.

Signals defined in Definition 13 have an important property: they are the "worst case" signals in terms of state variable evolution within a period.

The first result regards the evolution of the state variable when the subsystem is driven by a valid switching signal. In particular, Lemma 1 indicates the interval of values that the state variable can take in correspondence of the next request times.

Lemma 1 (Range for $x(r_{k+1})$) *Consider a system (4.58), a set of timing parameter $\mathcal{R}$ for the switching signal, and a time interval between two consecutive request times $\mathcal{T} = [r_k, r_{k+1}]$. For any finite value of $x(r_k)$, the following property holds:*

$$s \in \mathcal{S}^{\mathcal{R}}_{[r_k, r_{k+1}]} \quad \Longrightarrow \quad x(r_{k+1}) \in \left[x^{\min}_{k+1}, x^{\max}_{k+1}\right] \tag{4.62}$$

where:

$$\begin{aligned} x^{\min}_{k+1} = &\left(x(r_k) + b^{\text{on}}\right) e^{A^{\text{off}} T + (A^{\text{on}} - A^{\text{off}})C} + \\ &+ \left(b^{\text{off}} - b^{\text{on}}\right) e^{A^{\text{off}}(T-C)} - b^{\text{off}} \end{aligned} \tag{4.63a}$$

$$\begin{aligned} x^{\max}_{k+1} = &\left(x(r_k) + b^{\text{off}}\right) e^{A^{\text{off}} T + (A^{\text{on}} - A^{\text{off}})C} + \\ &+ \left(b^{\text{on}} - b^{\text{off}}\right) e^{A^{\text{off}} C} - b^{\text{on}} \end{aligned} \tag{4.63b}$$

Proof. We limit our proof to the first case, since the second one can be proven with similar arguments.

We prove this property for a simple switching signal, and we will extend by induction the considerations to generic schedules. Let's consider to divide the $[t_1, t_5]$ range in 4 sub-ranges $\mathcal{T}_1 = [t_1, t_2]$, $\mathcal{T}_2 = [t_2, t_3]$, $\mathcal{T}_3 = [t_3, t_4]$ and $\mathcal{T}_4 = [t_4, t_5]$, being $t_i < t_j$ for all $i < j$. Suppose the schedule

switches between active and inactive state at each range boundary. An example of state variable behavior generated by this case is depicted by the solid line in Figure 4.13. According to (4.58) the following equations hold:

$$x_1 = x(t_1) = x(r_k) \tag{4.64a}$$
$$x_2 = x(t_2) = b^{\text{off}} - (b^{\text{off}} - x_1)e^{-A^{\text{off}}(t_2 - t_1)} \tag{4.64b}$$
$$x_3 = x(t_3) = b^{\text{on}} - (b^{\text{on}} - x_2)e^{-A^{\text{on}}(t_3 - t_2)} \tag{4.64c}$$
$$x_4 = x(t_4) = b^{\text{off}} - (b^{\text{off}} - x_3)e^{-A^{\text{off}}(t_4 - t_3)} \tag{4.64d}$$
$$x_5 = x(t_5) = b^{\text{on}} - (b^{\text{on}} - x_4)e^{-A^{\text{on}}(t_5 - t_4)} \tag{4.64e}$$

Now we obtain a second schedule by moving a time slice Δ, being $0 < \Delta \leq (t_4 - t_3)$ of active time from sub-range $[t_3, t_4]$ to $[t_1, t_2]$. Notice that the validity in $[t_1, t_5]$ still applies. We obtain a new set of sub-ranges which are $\mathcal{T}_1' = [t_1, t_2 + \Delta]$, $\mathcal{T}_2' = [t_2 + \Delta, t_3 + \Delta]$, $\mathcal{T}_3' = [t_3 + \Delta, t_4]$ and $\mathcal{T}_4' = [t_4, t_5]$, and the corresponding values at sub-ranges boundaries:

$$x_1' = x'(t_1) = x(r_k) \tag{4.65a}$$
$$x_2' = x'(t_2 + \Delta) = b^{\text{off}} - (b^{\text{off}} - x_1)e^{-A^{\text{off}}(t_2 - t_1)}e^{-A^{\text{off}}\Delta} \tag{4.65b}$$
$$x_3' = x'(t_3 + \Delta) = b^{\text{on}} - (b^{\text{on}} - x_2)e^{-A^{\text{on}}(t_3 - t_2)} \tag{4.65c}$$
$$x_4' = x'(t_4) = b^{\text{off}} - (b^{\text{off}} - x_3)e^{-A^{\text{off}}(t_4 - t_3)}e^{A^{\text{off}}\Delta} \tag{4.65d}$$
$$x_5' = x'(t_5) = b^{\text{on}} - (b^{\text{on}} - x_4)e^{-A^{\text{on}}(t_5 - t_4)} \tag{4.65e}$$

The state variable behavior generated in this case is depicted by the dotted line in Figure 4.13.

We show now that the valid schedule having the active time located at the beginning of the time frame (i.e., a schedule in the form of $s^{\Downarrow}(t)$) will generate the maximum possible value of x_5'. For this purpose, we prove that $x_5' > x_5$ for any value of Δ. Therefore, since we are considering generic sub-ranges, we will have proven that every time a slice is taken from the interval $[t_3, t_4]$ and it is moved to $[t_1, t_2]$, the value of x_5' can only increase. Thus a schedule in the form of $s^{\Downarrow}(t)$ generates the highest possible value for $x(t_5)$.

From (4.65e) and (4.64e) it holds

$$x_5' - x_5 = (x_4' - x_4)e^{-A^{\text{on}}(t_5 - t_4)} \tag{4.66}$$

Since $e^{-A^{\text{on}}(t_5-t_4)} > 0$, then it holds $x_5' > x_5$ if and only if $(x_4' - x_4) > 0$. From (4.65d) and (4.64d) we obtain

$$x_4' - x_4 = e^{-A^{\text{off}}(t_4-t_3)}((b^{\text{off}} - x_3) - (b^{\text{off}} - x_3')e^{A^{\text{off}}\Delta}) \tag{4.67}$$

Equations (4.64c) and (4.65c) can be put within (4.66), then Equations (4.64b) and (4.65b) can be inserted in the newly obtained equation. Then, after simple calculations, the following equation can be written

$$x_4' - x_4 = e^{-A^{\text{off}}(t_4-t_3)}(b^{\text{on}} - b^{\text{off}})(1 - e^{-A^{\text{on}}(t_3-t_2)})(e^{A^{\text{off}}\Delta} - 1)$$

in which each term is strictly greater than zero (since $\Delta > 0$), thus proving that (4.66) is always positive.

The proof follows by noting that above procedure and considerations can be generalized to any valid schedule in which there is an arbitrary number of switches.

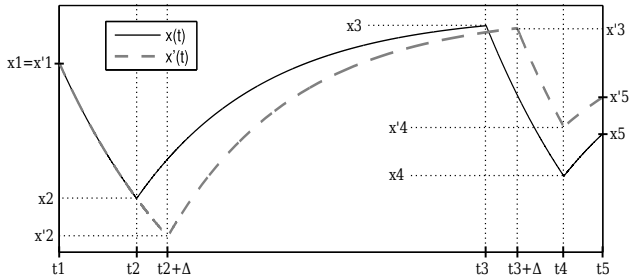


Figure 4.13: The state variable behavior and related parameters when a slice Δ is moved in the schedule from the interval $[t_3, t_4]$ to $[t_1, t_2]$.

Lemma 1 proves that the extreme values of the interval are obtained for specific behaviors of the switching signal. In particular, considering the timespan $[r_k, r_{k+1}]$, a load activity that is all concentrated at the beginning of the timespan corresponds to $x_{k+1}^{\min}$, while an activity concentrated at the end of the timespan corresponds to $x_{k+1}^{\max}$. Figure 4.14 illustrates this result. The value of $x(r_{k+1})$ is bounded within a range whose limits are function of physical and timing parameters (4.63). The exact value of $x(r_{k+1})$ depends on the actual values assumed by the switching signal within the time-span $[r_k, r_{k+1}]$.

Theorem 6 (Attraction range for Switched Affine RTPS)
Given a system (4.58) and a set of timing parameters $\mathcal{R}$, consider the succession $\{x(r_k)\}$. If the switching signal is valid then the interval $\tilde{\mathcal{I}} \doteq \left[\tilde{x}^{\inf}, \tilde{x}^{\sup}\right]$ is an attraction range for the succession, hence:

$$x(r_k) \in \left[\tilde{x}^{\inf}, \tilde{x}^{\sup}\right] \Longrightarrow x(r_{k+1}) \in \left[\tilde{x}^{\inf}, \tilde{x}^{\sup}\right], \quad \forall k,\ \forall s \in \mathcal{S}_{\infty}^{\mathcal{R}} \tag{4.68}$$

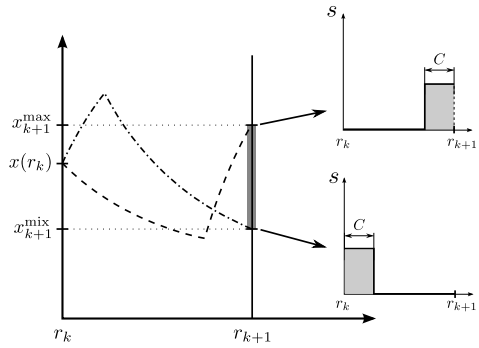


Figure 4.14: Limit-cases for the evolution of the state variable between two consecutive request times. The value of $x(r_{k+1})$ is bounded between $x_{i,k+1}^{\min}$ and $x_{i,k+1}^{\max}$ (see Observation 1).

$$x(r_k) \in \left[\tilde{x}^{\inf}, \tilde{x}^{\sup}\right], \quad \forall k \geq k^*,\ \forall x_0 \in \mathbb{R},\ \forall s \in \mathcal{S}_\infty^{\mathcal{R}} \tag{4.69}$$

where:

$$\tilde{x}^{\inf} = \frac{b^{\text{on}} + \left(b^{\text{off}} - b^{\text{on}}\right) e^{-A^{\text{on}}UT} - b^{\text{off}} e^{-(A^{\text{on}}U + A^{\text{off}}(1-U))T}}{1 - e^{-(A^{\text{on}}U + A^{\text{off}}(1-U))T}} \tag{4.70a}$$

$$\tilde{x}^{\sup} = \frac{b^{\text{off}} + \left(b^{\text{on}} - b^{\text{off}}\right) e^{-A^{\text{off}}(1-U)T} - b^{\text{on}} e^{-(A^{\text{on}}U + A^{\text{off}}(1-U))T}}{1 - e^{-(A^{\text{on}}U + A^{\text{off}}(1-U))T}} \tag{4.70b}$$

Proof. From Observation 1, the succession $\{x(r_k)\}$ decreases regardless of the switching signal while $x(r_k) > x_{k+1}^{\max}$. We are interested in finding the value of $\tilde{x}^{\sup}$, such that it delimits the region in which the succession $\{x(r_k)\}$ can only decrease. This value can be obtained by finding the fixed point of the succession $\{x(r_k)\}$ considering $x(r_{k+1}) = x_{k+1}^{\max}$, $\forall k$, leading

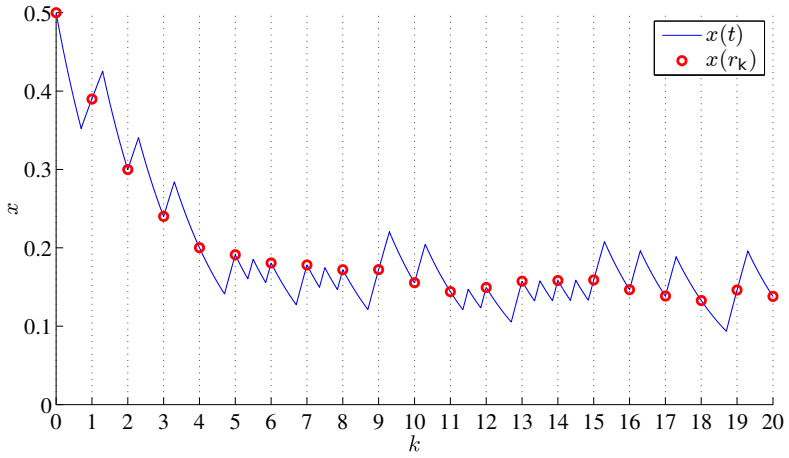


Figure 4.15: Example of state variable evolution and corresponding succession $\{x(r_k)\}$. In this example $(T, D, C) = (1, 1, 0.3)$.

to (4.70b). With similar arguments the lower bound of the attraction range (4.70a) can be derived.

The state variable will move towards the attraction range $\tilde{\mathcal{I}}$ regardless the position of the initial value x_0. As a consequence of Theorem 6, when the state variable initial value is within the attraction range, i.e. $x_0 \in \tilde{\mathcal{I}}$, $x(t)$ remains within $\tilde{\mathcal{I}}$ at every request time (i.e., $k^* = 0$). On the other hand, in case the state variable is initially placed outside the attraction range, it will evolve towards $\tilde{\mathcal{I}}$ and, once reached, will remain within the range.

However, the goal is to bound the state variable behavior not only in correspondence with request times, but for every time instant. Results provided by Theorem 6 simplify the analysis, since they allow to study the

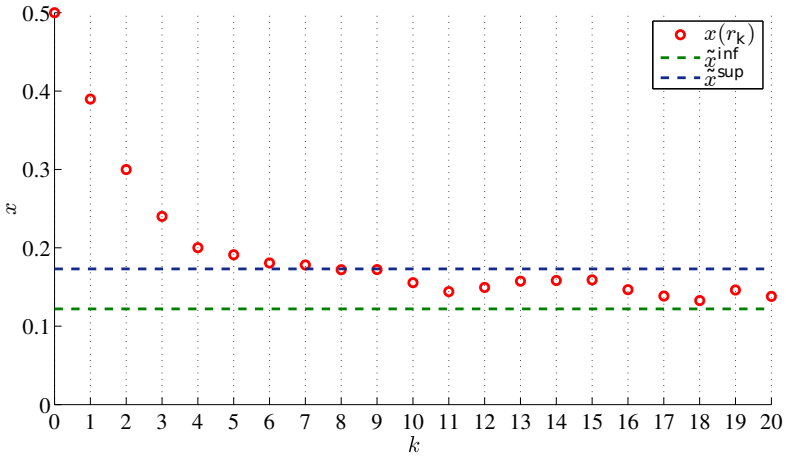


Figure 4.16: Example of state variable evolution and corresponding succession $\{x(r_k)\}$. For $k \geq 8$ the succession $\{x(r_k)\}$ falls within the attraction range. In this example $(T, D, C) = (1, 1, 0.3)$.

worst cases behavior of the state variable values between two consecutive request times, instead of having to consider the system lifetime for every t. In fact, once identified the worst case conditions of the state variable at request times, it is possible to evaluate the worst case conditions within a generic period. Therefore, the worst case will hold for every t.

Theorem 7 (Working range for Switched Affine RTPS)
Within the same hypotheses of Theorem 6, it holds:

$$x(r_k) \in \tilde{\mathcal{I}} \Longrightarrow x(t) \in \mathcal{I}, \quad \forall t \geq r_k,\ \forall s \in \mathcal{S}_\infty^{\mathcal{R}} \tag{4.71}$$

where $\mathcal{I} \doteq \left[x^{\text{inf}}, x^{\text{sup}}\right] \supseteq \tilde{\mathcal{I}}$ *and*

$$x^{\text{inf}} = b^{\text{on}} - \left(b^{\text{on}} - \tilde{x}^{\text{inf}}\right) e^{-A^{\text{on}}UT} \tag{4.72a}$$

$$x^{\sup} = b^{\text{off}} - \left(b^{\text{off}} - \tilde{x}^{\sup}\right) e^{-A^{\text{off}}(1-U)T} \tag{4.72b}$$

Proof. The upper bound of the working range, namely $x^{\sup}$, can be found considering the worst case scenario during the timespan from r_k to r_{k+1}. This worst case happens when the state variable value at r_k is maximum, which is $x(r_k) = \tilde{x}^{\sup}$ from Theorem 6, and the switching signal has the on part (i.e. the part that increases x) all concentrated at the beginning of the timespan. In this case, at $t = r_k + C$ the state variable reaches the maximum and its value is (4.72b). A similar argument leads to (4.72a) as well.

Theorem 7 provides the bounds for the state variable variation range after a large enough time, i.e. $t > k^* T_i$. Again, if the state variable initial value is within $\tilde{\mathcal{I}}$, then its variation is bounded by $\mathcal{I}$ for every t. It is worth to note that bounds are function of physical and real-time parameters only, and represent the relationship between such parameters.

It is interesting to note that the state variable has an asymptotic behavior for $T \to 0$, as outlined by the next observation.

Observation 7 (Limit value) *For a dynamical system* (4.58) *driven by a valid switching signal* (4.6)*, it holds:*

$$\lim_{t\to\infty,\, T\to 0} x(t) = \bar{x} = \frac{F^{\text{on}}U + F^{\text{off}}(1-U)}{A^{\text{on}}U + A^{\text{off}}(1-U)} \tag{4.73}$$

In general, the interest is to find real-time parameters U and T that allow to satisfy the user requirements, i.e. $x(t) \in \Psi$. For this purpose, the

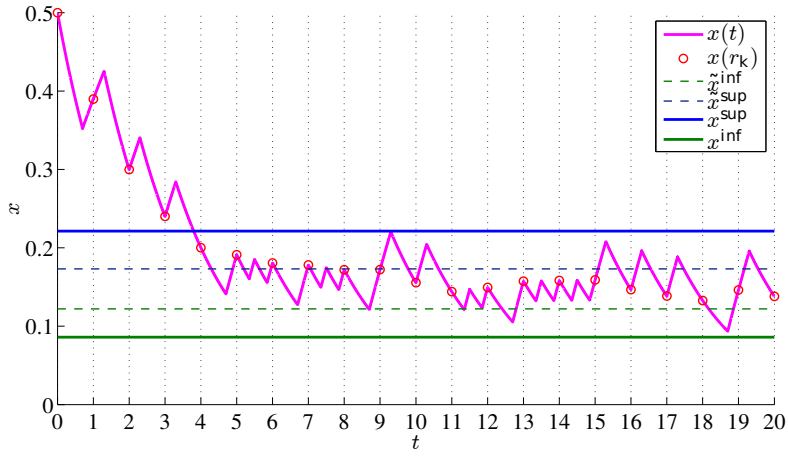


Figure 4.17: Example of state variable evolution with highlighted succession $\{x(r_k)\}$, attraction range bounds (Theorem 6), working range bounds (Theorem 7). In this example $(T, D, C) = (1, 1, 0.3)$.

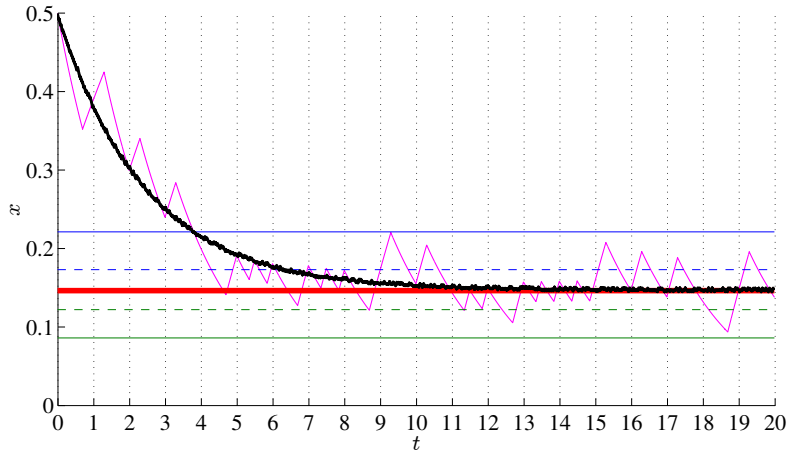


Figure 4.18: Illustration of Observation 7 with the example in Figure 4.17. The black line corresponds to the state variable evolution with a switching signal characterized by a very small value for the period, i.e. $(T, U) = (10^{-5}, 0.3)$. The red line is the value calculated with (4.73).

following definition is introduced.

Definition 14 (Feasibility region)
The feasibility region Ω *is a region in the* $U-T$ *space composed by all and only pairs* (U,T) *such that every valid switching signal, characterized by such timing parameters, drives the subsystem satisfying user requirements.*

Given a pair (U,T), it is possible to check whether it belongs to the feasibility region Ω by checking that:

$$\Psi^{\mathsf{mm}} \subset \mathcal{I}. \tag{4.74}$$

In other words the following inequalities are satisfied: $x^{\inf} \geq x^{\min}$ and $x^{\sup} \leq x^{\max}$. Recall $x^{\min}$ and $x^{\max}$ are given by the user requirements, and $x^{\inf}$ and $x^{\sup}$ are function of the physical and the real-time parameters, as indicated in (4.72).

In general, there exists a set of pairs $(U,T) \in \Omega$. Unfortunately, it is not possible to find the values of U and T from (4.72) in closed form. Therefore, pairs $(U,T) \in \Omega$ must be found using numerical techniques. On the other hand, the result from Observation 7 can be used to find the bounds of Ω on the U axis. In fact, it is possible to find a range $[U^{\min}, U^{\max}]$ in which a pair $(U,T) \in \Omega$ can be found. In other words, conditions

$$U^{\min} \leq U \leq U^{\max} \tag{4.75}$$

are necessary conditions for the system feasibility. Range bounds can be determined by imposing $\bar{x} = x^{\max}$ and $\bar{x} = x^{\min}$, respectively, in (4.73),

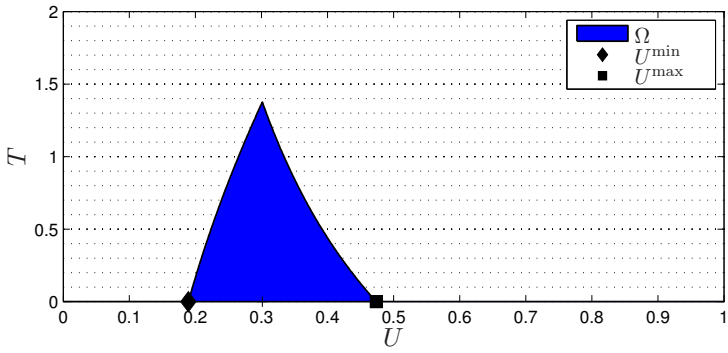


Figure 4.19: Example of feasibility region (Definition 14) in the $U - T$ space. In a system with a valid switching signal characterized by timing parameters members of the feasibility region, user requirements satisfaction is guaranteed.

leading to the following equations:

$$U^{\max} = \frac{A^{\text{off}}(b^{\text{off}} - x^{\max})}{(A^{\text{on}} - A^{\text{off}})x^{\max} - (F^{\text{on}} - F^{\text{off}})} \tag{4.76a}$$

$$U^{\min} = \frac{A^{\text{off}}(b^{\text{off}} - x^{\min})}{(A^{\text{on}} - A^{\text{off}})x^{\min} - (F^{\text{on}} - F^{\text{off}})} \tag{4.76b}$$

Once the value of U has been chosen, it is possible to select a proper value of T complying with (4.74). An example of feasibility region is depicted in Figure 4.19. The choice of a lower value for U helps to obtain more loads to be scheduled together. However, lower U brings to lower T, thus generating a higher switching frequency. This is often not tolerable in practical applications such as processes driven by electric motors.

4.7 Industrial process

This type of systems has been studied in (Della Vedova and Facchinetti [18]) and differs from the others presented in this chapter even if it is still a refinement of the general model defined in Section 4.1. The main difference is that while other models are decoupled, meaning that one switching signal acts on one state variable, here subsystems are coupled. In fact the behavior of state variables is given by two or more switching signals. Moreover, one switching signal can have influence over two or more state variables.

The following model can by applied to industrial processes, where goods produced as output by a machine can be used as input by another machine.

4.7.1 System model

The system is composed by a set $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$ of m machines and n containers, which store goods. Each machine performs a specific operation in the production plant. In the scope of this model, 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 containers, and it produces goods that are stored in one or more output containers. Production/consumption rates are system parameters specified in the sequel. The machine does not produce and consume any goods when inactive.

Containers are characterized by current level and maximum storage capacity: $x_i(t)$ and $x_i^{\max}$ for the i-th container. Figure 4.20 shows an example of modeled system, where machines are represented by circles and containers by rectangles. The graphical representation also shows the input/output relationships among loads and containers.

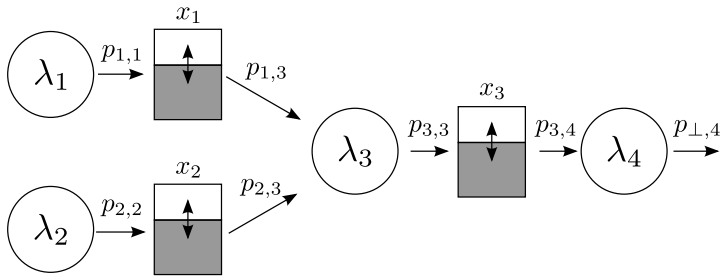


Figure 4.20: Example of an industrial plant made by 4 machines (circles) and 3 containers. Arrows indicate the flow of produced/consumed goods.

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

$$\begin{cases} \dot{x}(t) = Bs(t) \\ x(0) = x_0 \end{cases} \tag{4.77}$$

where:

- $t \in \mathbb{R}^+$ is the continuous time;
- $x(t) = [x_1(t) \dots x_n(t)]' \in \mathbb{R}^n$ is the column vector of the n state variables, which are the containers levels;
- $x_0 \in \mathbb{R}^n$ is the initial state of the system, i.e. the initial levels for the containers;
- $s(t) \in \mathbb{B}^m$ 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.
- $s : \mathbb{R}^+ \to \mathbb{B}^m$ 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 container. 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 container is not affected by the actions of the λ_j machine. For example, the matrix B for the system represented in Figure 4.20 takes the following values:

$$B = \begin{bmatrix} p_{1,1} & 0 & -p_{1,3} & 0 \\ 0 & p_{2,2} & -p_{2,3} & 0 \\ 0 & 0 & p_{3,3} & -p_{3,4} \end{bmatrix}$$

4.7.2 Constraints and requirements

In (Ashok and Banerjee [1]), 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:

$$\int_0^{t_f} p_{\perp,i}\, s_i(t)\, \mathrm{d}t \;\geq Q_i, \quad \forall i : \lambda_i \in \Lambda^{\text{out}} \tag{4.78}$$

where t_f is the end of the desired production time span (e.g. a day, a week, etc.), $\Lambda^{\text{out}} \subset \Lambda$ is the subset of all machines whose production is not formally used to fill a container but it represents the final product of the industrial process and Q_i is the final product desired quantity.

The **Storage Constraint** encapsulates the requirement, for each con-

tainer level, to not exceed the maximum capacity of the container itself.

$$x(t) \in \mathcal{X} \quad \forall t \geq 0 \tag{4.79}$$

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 maintained in the containers. This reserve allows the machine served by a container to always have an input material available.

$$x_i^{\min} > 0 \quad \forall i \tag{4.80}$$

4.7.3 Feasibility analysis

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.

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

Theorem 8 (Periodicity for Industrial Process RTPS)
Given the system (4.77) driven by a valid schedule and a set of timing parameters $\mathcal{R}^*$ that includes $\bar{U}^*$, if the vector $\bar{U}^*$ is in the null-space of the matrix B then the state of the system at the hyper-period is equal to to initial state. Formally:

$$\forall s \in \mathcal{S}_\infty^{\mathcal{R}^*} : \qquad B\bar{U}^* = 0 \implies x(H) = x_0 \tag{4.81}$$

Proof. The solution of the ordinary differential equation describing the system's dynamics , obtained by integrating (4.77), is:

$$x(t) = B \int_0^t s(\tau) \mathrm{d}\tau + x_0 \tag{4.82}$$

Since H is by definition a multiple of every T_i, from (4.5a) can be derived:

$$\int_0^H s(t) \mathrm{d}t = \bar{U} H \tag{4.83}$$

Combining (4.82) and (4.83), it follows: $x(H) = B\bar{U}H + x_0$. When the condition in the theorem's hypothesis (4.81) holds, $x(H) = x_0$ follows.

Theorem 8 states that, for some specific values of the utilization of all loads, the level of each container $x_i(t)$ at time $t = H$ will be equal to the initial level $x_i(0)$ of that container. This results has an important impact on the analysis of the system. Assuming that all utilization are set as given by Theorem 8, 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 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 $[0, H]$.

It is worth noting that the *Production Constraint* (4.78) can be translated into the following inequality on U:

$$U_i \geq \frac{Q_i}{t_f p_{\perp,i}}, \quad \forall i : \lambda_i \in \Lambda^{\text{out}} \tag{4.84}$$

Consider a load λ supplying good to a container (subscripts are sup-

pressed 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 8. 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 4.21. Note that we are considering the worst-case of the maximum variation of $x(t)$. In fact, in case (C) of Figure 4.21, 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 container (Storage Constraint) or on the minimum level of the container (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 container.

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 $x(0)$.

$$\max_{s_i, s_o \in \mathcal{S}} \max_{t \geq 0} x(t) = f\left(T_i, T_o, x_0\right) \tag{4.85}$$

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

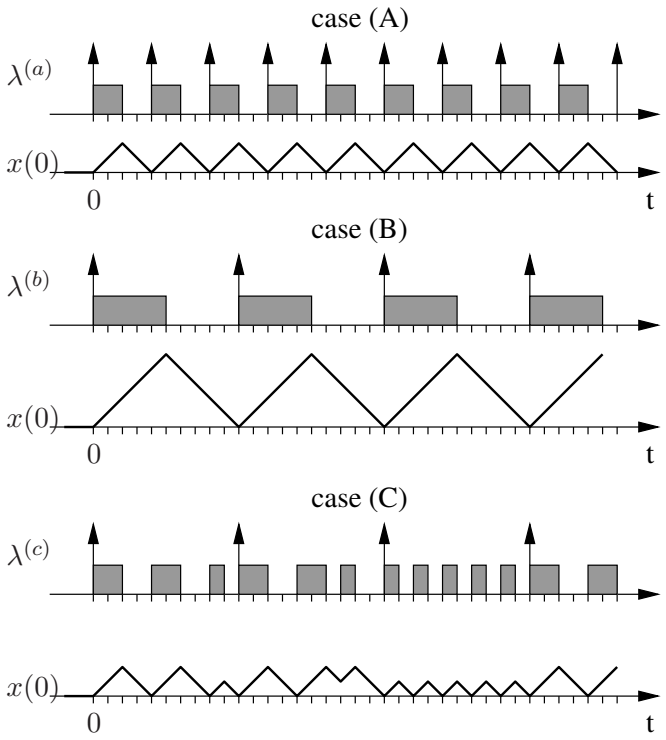


Figure 4.21: Example of the effect of period selection for a load having a given utilization. The considered load λ has the same utilization $U = 0.5$ in all cases. However, the different chosen values for the period in the two cases (A) and (B) determine a noticeable difference in the worst case variation of the container level with respect to $x(0)$. Case (C) has the same period of case (B), but due to activity interruptions (preemption), the maximum variation is less than in the worst-case.

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 decrease. Similarly, Δ_{out} denotes the decrease 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., $\max_{0 \le t \le M} x(t) = x_0 + \Delta^{\max}$. Similarly, we will denote with $\Delta^{\min}$ the maximum negative variation with respect to x_0, i.e., $\min_{0 \le t \le M} x(t) = x_0 - \Delta^{\min}$. 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:

$$\Delta^{\max} = \max_{t} \left\lceil \frac{t}{T_{in}} \right\rceil - (k-1)\Delta_{out} \tag{4.86}$$

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

$$\Delta^{\min} = \min_{t} \left\lfloor \frac{t}{T_{out}} \right\rfloor - (k+1)\Delta_{out} \tag{4.87}$$

Finally, by imposing the constraints it holds:

$$\begin{cases} x^{\max} \ge x_0 + \Delta^{\max} \\ x^{\min} \le x_0 + \Delta^{\min} \end{cases} \tag{4.88}$$

Therefore, suitable values for T_{in} and T_{out} can be obtained by solving (4.88).

Figure 4.22 depicts the two schedules that cause the maximum positive

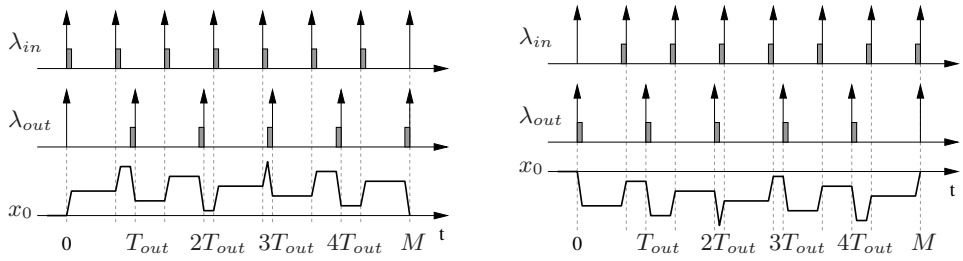


Figure 4.22: Time diagram representing the maximum positive (on the left) and negative (on the right) variations of the state variable due to respective schedules.

and negative variations of the state variable. The maximum positive variation happens when the activation time of the supplier machine is always at the beginning of its period and the activation time of the consumer machine is always at the end of its period. For the maximum negative variation, vice-versa applies.

4.7.4 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 4.23 shows the schedule of the four machines composing the industrial plant depicted in Figure 4.20. The activity of the machines, together with the levels of containers and the overall power consumption, is reported for a time horizon of 24 hours, which is twice the hyper-period H for this system.

The following values have been used in the simulation: 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; containers initial level $x_0 = [5\ 5\ 5]$ units of goods; minimum and

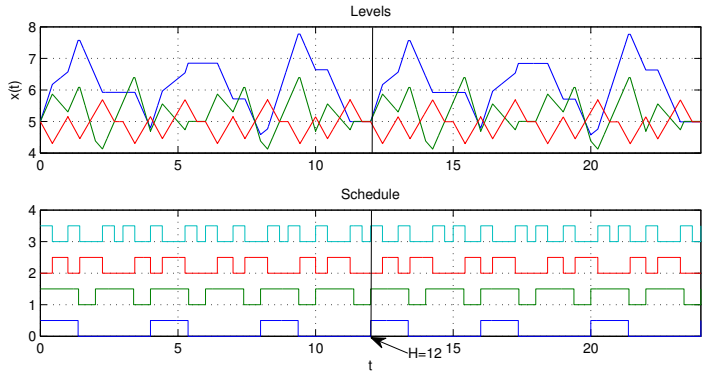


Figure 4.23: Example of the scheduling of the four loads composing the industrial process of Figure 4.20. Hyper-period is $H = 12$.

maximum levels for each container are 2 and 10 units of goods.

Considering Theorem 8, the desired system behavior is obtained with machine utilizations equal respectively to 34%, 69%, 46% and 43%. The periods have been chosen according to (4.88) and their values are respectively 4, 3, 2 and 1 hours.

4.8 Summary

This chapter has presented the Real-Time Physical System model, first in general terms and then in details for many specific models. For each model, the feasibility analysis has been carried out. The feasibility analysis is used to find appropriate timing parameters, by which a load can be included into the RTPS framework. Suitable timing parameters are those that guarantee the respect of user requirements on the state variable, which determine the quality of service on the physical process actuated by the switching signal.

In Section 4.1 has been introduced the study **methodology**, by presenting all the important properties for the switching signal (in particular the notion of **validity**, Definitions 3 to 5), and the definition of the so-called **feasibility problem** (see Definitions 7 and 8).

The first model presented were **switched integrators**, in Section 4.2. Switched integrators systems captures physical processes where the quantity of interest evolves linearly in time. It has been proved that, by properly setting the switching signal utilization U, it is possible to have a stable system (Theorem 1); moreover, by properly setting also the switching signal period T, user requirements are always satisfied by any valid open-loop schedule characterized by those parameters (Theorem 2).

Section 4.3 discussed the effect of bounded **modeling errors** for the previous system. It turned out that an open-loop schedule cannot guarantee the respect of user requirements. So, a closed-loop scheme has been introduced: the adaptive scheduling algorithm that changes load activation times $C_{i,k}$ and/or periods $T_{i,k}$ at every request time according to Theorem 3 is able to compensate errors and to drive the state variable evolution correctly.

In Section 4.4 **stochastic disturbance** has been taken into account.

In this case, Theorem 5 has demonstrated that by properly setting $C_{i,k}$ is possible to minimize the probability to exceed user requirements, but not to certainly avoid the undesired situation. In fact, soft user requirements must be considered because of the unbounded nature of the considered disturbance. Again a closed-loop adaptive schedule can compensate disturbances and drive the state variable evolution as expected.

Section 4.5 has illustrated how to manage **transient** conditions.

Section 4.6 has presented **switched affine systems**, in which the state variable evolution of each system mode is an exponential decay. Theorem 7 proved that, by properly setting the load utilization U_i and its period T_i, it is possible to limit the state variable within a desired working range. In this case, there exists the so-called *feasibility region* in the U_i–T_i plane for which any valid switching signal drives the system correctly.

In Section 4.7 we modeled **industrial processes**, in which goods produced as output by a machine are used as input by another machine. Differently from previous models, two or more switching signals can have influence on one state variable. It has been shown that also in this case it is possible to find feasible parameters for load utilizations (Theorem 8) and periods (Equation (4.88)).

5

Real-Time Physical Systems applied to Electric Load Management

Contents

This chapter presents an innovative approach to Electric Load Management which is based on modeling and scheduling techniques studied for Real-Time Physical Systems. The balance of power usage aims to avoid dangerous peak load conditions, happening when too many loads are simultaneously active. Overload conditions can lead to technical and economic issues on the power distribution infrastructure, including blackouts. The proposed technique allows to guarantee that the peak power demand is upper bounded by a known threshold. The coordinated activation of loads realizes a *peak shaving*, producing a smoother and flatter curve of power usage over time.

5.1 Peak load minimization

The application of RTPSs proposed in this paper is to limit the peak load of power consumption generated by a set of electric loads, while meeting requirements on physical values. Each electric device can be either active or not. The activity of loads is controlled by the scheduler that generates the s_i schedule for the i-th load. The i-th device consumes either a $P_i \in \mathbb{R}^+$ amount of electric power when active, no power otherwise[1]. Hence, the power demand over time of an electric load is modeled with the function $p_i : \mathbb{R}^+ \to \mathbb{R}^+$, defined as follows:

$$p_i(t) \doteq P_i s_i(t) \tag{5.1}$$

Clearly, the overall instantaneous electric power absorbed by the entire system at time t is the sum of the power consumed by every loads, as stated

[1] Transient phases between active and inactive states are not captured by the proposed model

in (5.2).

$$w(t) \doteq \sum_{i=1}^{m} p_i(t). \tag{5.2}$$

The maximum value taken by the function w during the system lifetime is the **peak load**:

$$W \doteq \max_{t \geq 0} w(t) \tag{5.3}$$

The goal is to reduce and possibly minimize the peak load while meeting all the requirements of the system. Taking into account the dynamical system and the user requirements introduced in (4.1) and (4.9) respectively, the peak load minimization problem can be formally defined as follows.

Definition 15 (Peak load minimization problem)
The peak load minimization problem consists in finding the optimal schedule $s^* : \mathbb{R}^+ \to \mathbb{B}^n$, which minimizes the peak load and satisfies user requirements.

$$s^* = \arg\min_{s} \max_{t \geq 0} \sum_{i=1}^{n} P_i s_i(t) \tag{5.4a}$$

$$\text{subj. to} \quad \dot{x}(t) = f_{s(t)}\left(x(t)\right) + d(t) \tag{5.4b}$$

$$x(t) \in \Psi(t), \quad \forall t > 0 \tag{5.4c}$$

The optimization problem (5.4) is a hard problem in the sense of computational complexity. In fact, the problem of finding an optimal allocation minimizing the peak load is NP-hard in the strong sense, since the easier problem in which all loads have the same power consumed is NP-hard in the strong sense, for analogy with the multiprocessor case.

Despite the complexity of the general problem, it is easy to find an upper-bound and a lower-bound for the peak load value W.

Observation 8 (Peak load upper-bound) *No schedule, whether it is valid or not, can lead to a peak load greater than*

$$W^{\text{ub}} = \sum_{i=1}^{m} P_i \tag{5.5}$$

Proof.

$$\begin{aligned}
W &\leq \max_{s} \left(\max_{t \geq 0} \sum_{i=1}^{m} P_i s_i(t) \right) && \text{(by definitions)} \\
&\leq \sum_{i=1}^{m} \max_{s_i} \max_{t \geq 0} P_i s_i(t) && \text{(max of sum} \leq \text{sum of max)} \\
&\leq \sum_{i=1}^{m} P_i \left(\max_{s_i} \max_{t \geq 0} s_i(t) \right) && (P_i \text{ are constants}) \\
&\leq \sum_{i=1}^{m} P_i && (s_i \text{ maps to } (0,1))
\end{aligned}$$

The previous observation has an immediate interpretation: in fact, it is obvious that the maximum possible peak load happens when all the electric device are active at the same time.

Conversely, it is not obvious to find the minimum possible peak load, which would be the partial solution of the peak load minimization problem. However, assuming that the schedule does not affect the total energy required by the system, it is worth noting that:

Observation 9 (Peak load lower-bound) *No valid schedule can lead to a peak load lower than*

$$W^{\text{lb}} = \sum_{i=1}^{m} P_i U_i \tag{5.6}$$

Proof. Assume, by contradiction, a load allocation grants a peak load $W < W^{\text{lb}}$. Let H be the least common multiple of all load periods $T_1, \ldots, T_m$. The overall energy consumed by the system 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 W, the overall energy consumed by the system in H can not be greater than WH. Therefore,

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

and

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

Using (5.6), we get

$$W^{\text{lb}} \leq W.$$

leading to a contradiction.

Observations 8 and 9 have introduced two important figure that can be used as benchmarks for evaluating the effectiveness of scheduling methods.

It is worth noting the following properties for the peak load bounds. Considering an uncoordinated schedule, the actual peak load can be equal to the upper bound W^{ub}. In fact, the situation for which the system power demand equals the peak load upper bound, i.e. $w(t) = W^{\text{ub}}$, is possible and it corresponds to have all electric loads active at the same time. As it will be shown later on, it is an improbable case, but possible. Conversely, the lower peak load bound W^{lb} is often impossible to be reached. The simplest example is a system composed by one electric load that consumes a power $P_1 = 600$ Watt for 15min every 60min, so load utilization is $U_1 = 15/60 = 0.25$. Therefore, according to (5.6), $W^{\text{lb}} = U_1 P_1 = 150$ Watt, but clearly the minimum peak load for the system is the nominal power of the single load: 600 Watt. The value of W^{lb} corresponds to the minimal power required to do the same work, in terms of energy, of the given electric load characterized by (T_1, U_1).

In other words, the situation $w(t) = W^{\text{ub}}$ is possible (but unlikely) for every system, while $w(t) = W^{\text{lb}}$ is not even possible for the great majority of the systems.

5.1.1 Comments

A RTPS scheduler generates the optimal schedule s^* when a uni-processor scheduling algorithm, such as EDF, is able to schedule the load set. In this case, the algorithm achieves that only one load is active at any given time, and the peak load is equal to the power consumed by the most power-consuming load. The schedulability test can be used to determine whether there exists a feasible schedule, provided that user requirements are also met.

On the other hand, if simultaneous activations can not be avoided, i.e.

when a uni-processor scheduling algorithm is not able to schedule the load set, then the minimization of the peak load becomes more complex. In this case, multi-processors scheduling algorithms can be exploited and a RTPS scheduler will generate a schedule that drives the system to a peak load close to the optimal. Therefore, the RTPS method can be seen as an efficient heuristic for the peak load minimization problem. The efficiency of the heuristic is due to the lower computational complexity of classic real-time scheduling algorithm compared to standard optimization methods.

5.2 Group Scheduling

This part has been discussed in (Della Vedova and Facchinetti [20], Facchinetti et al. [31]). We propose the use of classic real-time scheduling algorithms to manage the set of electric loads, such as Rate Monotonic (RM) or Earliest Deadline First (EDF). Those scheduling algorithms require the specification of T_i and C_i for every load to build a schedule. In Chapter 4 the feasibility analysis of many system models showed how to find appropriate timing parameters T_i and C_i that guarantee to meet user requirements.

Well known real-time scheduling algorithms such as RM and EDF can generate a schedule where only one single load is active at any given time. However, this is possible only when the total utilization U^{tot} is less or equal of an upper bound $U^{\text{lub}}(\mathcal{A})$ whose value depends on the considered algorithm $\mathcal{A}$. For example, $U^{\text{lub}}(\text{EDF}) = 1$. Therefore, when $U^{\text{tot}} \leq 1$, preemptive EDF can build a schedule without activating more than one load at any given time. As a consequence, the peak load $W = \max_i P_i$ is minimized.

On the other hand, if $U^{\text{tot}} > U^{\text{lub}}$ then the simultaneous activation of

two or more loads can not be avoided. The proposed solution is to partition the whole load set into k disjoint sets Λ_j, $j = 1, \ldots, k$, called *scheduling groups*. Scheduling groups are determined such that their total utilization, defined as

$$U_{\Lambda_j} = \sum_{i \in \Lambda_j} U_i, \tag{5.7}$$

is smaller than or equal to $U^{\text{lub}}(\mathcal{A})$. This property enables an uni-processor scheduling algorithm $\mathcal{A}$ to find a valid schedule independently for each scheduling group.

Since there is no relationship between the schedule generated within any pair of scheduling group, the maximum overall peak load will happen when the loads with the highest power are simultaneously activated in every scheduling group. Therefore, an upper bound $W^{\max}$ on the peak load can be found considering the simultaneous activation within every group of the load with the highest power P_i, i.e.:

$$W^{\max} = \sum_{\Lambda_j} \max_{i \in \Lambda_j} P_i. \tag{5.8}$$

5.2.1 Level packing

The problem of partitioning the set of loads can be formalized as a *level packing* problem (Lodi et al. [47]). Level packing is a special case of the generic two-dimensional packing problem. In level packing, one or more strips are filled to accommodate a set of rectangles such that the total height is minimized. The peculiarity of level packing is that rectangles are partitioned in horizontal strips or 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 book-

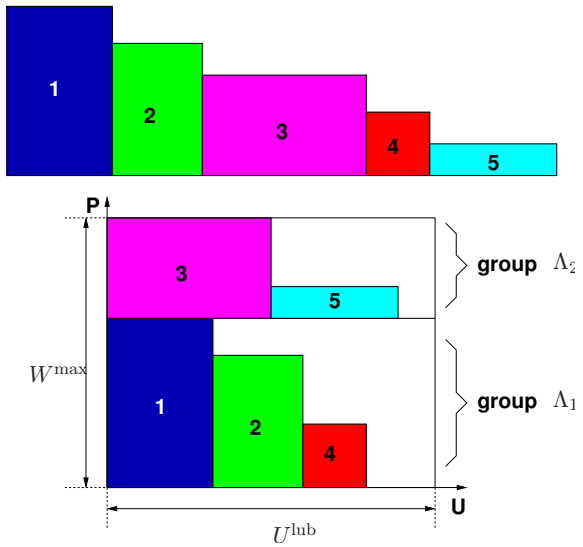


Figure 5.1: Example of level packing using the FFDH algorithm. Five items are firstly ordered by non-increasing height and then packed into two levels. Note that item 3 generates a new level Λ_2 since it does not fit at the right of item 2 within level Λ_1. The load utilizations of the 5 loads are respectively $(0.28, 0.26, 0.49, 0.22, 0.30)$. Their consumed power are $(5, 4, 3, 2, 1)$.

shelf (see Figure 5.1). 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.

The complexity of the level packing problem is NP-hard; in fact, it can be easily reduced to a classical one-dimensional packing problem having NP-hard complexity.

The application of the level packing to the scheduling of electric loads requires the proper modeling of the loads. Therefore, each load λ_i is rep-

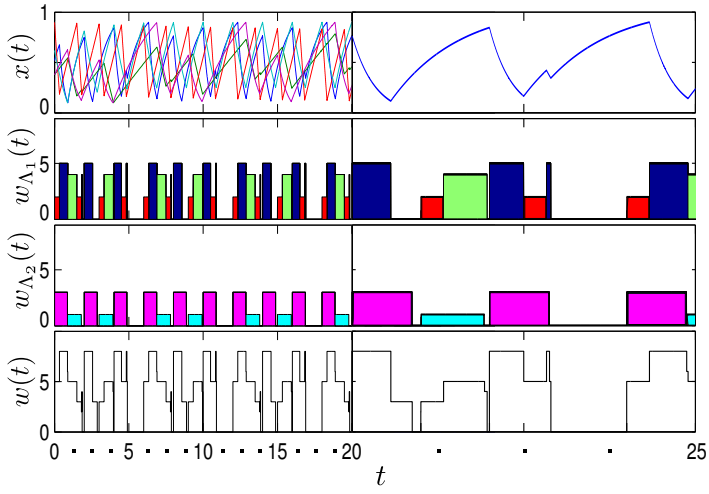


Figure 5.2: Example of group scheduling of loads of Figure 5.1. The time-scale is enlarged from $t = 20$ to $t = 25$ to better depict the system evolution. In the top chart shows the evolutions of state variables normalized in their working ranges. The second and third top charts show the consumed power by two independently scheduled groups. Active loads are indicated by their color. The bottom chart shows the overall power demand.

resented as a rectangle having height equal to the power consumption P_i and width equal to its utilization U_i, being $U_i \leq 1$. The packing happens in a two-dimensional space where the utilization appears on the x axis, while the consumed power is on the y axis. The width of the packing space corresponds to the least upper bound on the utilization of the considered scheduling algorithm (e.g., $U^{\text{lub}} = 1$ for EDF). The goal to limit the total height of packed rectangles clearly corresponds to the goal of limiting the peak load of power consumption of the whole power system. On the other hand, fitting the items on the x axis in each level corresponds to

group a set of loads whose total utilization is less than or equal to U^{lub}, thus composing a set of loads that is successfully schedulable by the considered real-time scheduling algorithm. Once all loads have been grouped into scheduling groups using the level packing, each group of loads is scheduled independently from other groups. The schedulability is guaranteed since the utilization U_{Λ_j} of the group Λ_j is less than or equal to the upper bound U^{lub} that achieves the schedulability of the load set. Figure 5.2 shows an example of schedule of loads whose relevant parameters are listed in Figure 5.1.

We will present two grouping 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.

5.2.2 Linear programming formulation

In this section, 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 m possible levels, one for each rectangle as the starting item. A set of m 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 labeled 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.

For example, in the case depicted in Figure 5.1, 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:

$$y_j + \sum_{i=1}^{j-1} x_{i,j} = 1 \qquad \forall j = 1, \ldots, m \tag{5.9}$$

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 U^{lub} is defined to be equal to the utilization upper bound that guarantees the schedulability of a load set. For example, if EDF with implicit deadlines is used, then we set $U^{\text{lub}} = 1$. Since the horizontal dimension is interpreted as utilization, then each level can not exceed the width U^{lub} of the rectangle. Therefore, it holds

$$\sum_{j=i+1}^{m} U_j \, x_{i,j} \leq (U^{\text{lub}} - U_i) \, y_i \qquad \forall i = 1, \ldots, m-1 \tag{5.10}$$

To enforce the consistency of the constraint given by (5.10), 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 (5.10) 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 (Baruah and Bini [3]), 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

$$\text{minimize} \quad \sum_{i=1}^{m} P_i \, y_i \tag{5.11}$$

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 m, because all rectangles may initialize one level. The $x_{i,j}$ variables are $\frac{m(m-1)}{2}$. Hence, the total number of variables is $\frac{m(m+1)}{2}$. Moreover, by counting the number of inequalities in (5.9) and (5.10), we find that the number of constraints is $2m - 1$.

5.2.3 FFDH Heuristic

Approximation methods have been proposed to face the complexity issue (Lodi et al. [46]). The approximation methods are built by ordering the rectangles by non-increasing heights. Rectangles are grouped to fill the strips using different strategies. In each level, items are packed from left to right by non-increasing height, similarly to the arrangement of books within a bookshelf (see Figure 5.1). The First-Fit Decreasing Height (FFDH)

scheme is conceived such that it inserts the next item X (in non-increasing height ordering) on the first level where X fits. If no level can accommodate X, a new level is created. After the packing action, the height of a level is equal to the height of the leftmost item.

Algorithm 2 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. The algorithm is essentially a first-fit bin-packing algorithm applied to the ordered set of loads. The i-th 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.

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

Moreover, the approximation ratio has been formally derived. In particular, it is proven in (Coffman et al. [15]) that, for any instance I of the problem, if the heights are normalized to one, then

$$\mathrm{FFDH}(I) \leq (17/10) \cdot \mathrm{OPT}(I) + 1 \tag{5.12}$$

where $\mathrm{FFDH}(I)$ is the height obtained by FFDH, and $\mathrm{OPT}(I)$ is the height produced by the optimal algorithm. The asymptotic bound is proved to be tight, meaning that the multiplicative constant is the smallest possible.

The proposed technique recalls the *Rate Monotonic First-Fit Decreasing Utilization* (RM-FFDU) partitioning scheme for scheduling fixed priority

Algorithm 2 The pseudo-code of First-Fit Decreasing Height (FFDH) heuristic.

sort the load set Λ in decreasing order of power
$g = 1$ is the initial number of scheduling groups
create a new scheduling group Λ_1
for all $i \in \{1, \ldots, m\}$ **do**
  initialize the boolean flag `assigned := FALSE` that will track whether the load will be assigned to a scheduling group or not
  for $j = 1$ to g **do**
    if load i is schedulable in Λ_j **then**
      add i to Λ_j
      `assigned := TRUE`
      break
    end if
  end for
  if `assigned == FALSE` **then**
    create a new scheduling group Λ_{g+1}
    add i to Λ_{g+1}
    $g = g + 1$
  end if
end for

real-time tasks on a multi-processor system (Oh and Son [58]), where bin-packing techniques are used to allocate tasks to processors. However, (Oh and Son [58]) does not address the optimization of the total power consumption. Moreover, the key distinction is that in this paper the ordering is made with respect to the value of load's consumed power, and utilization is not considered for this purpose.

5.2.4 Performance evaluation

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 FFDH heuristic.

The peak load achievable using the proposed schemes is compared with the upper bound (5.5).

The parameters that have been taken into account in the experiments are: the total number of loads m, the total utilization of the set of loads U^{tot}, 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 (Bini and Buttazzo [6]).

Figure 5.3 shows the efficiency of different approaches with respect to $P^{\max}$, as a function of the ratio between the total utilization U^{tot} and the number of loads m. The efficiency η is calculated as

$$\eta = \frac{W^{\text{ub}} - W^{\max}_{\text{meth}}}{W^{\text{ub}}} \cdot 100$$

where $W^{\max}_{\text{meth}}$ represents the peak load achieved by the given method: lower bound, LP and heuristic refer, respectively, to the peak load obtained from

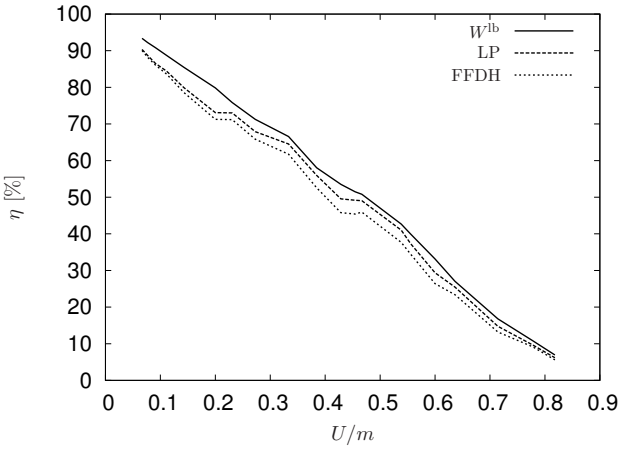


Figure 5.3: Efficiency of different approaches as a function of the average utilization.

Observation 9, the method of Section 5.2.2 and the approximated approach of Algorithm 2. 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 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 5.3 show that for lower values of the U/m 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 W^{ub}. Therefore, the explicit control on load activations brings to a remarkable improvement in comparison to the absence of control actions. When the U/m ratio tends to 1, the benefits of using a scheduling approach disappear. This is due to the fact that, when U tends to m, the load

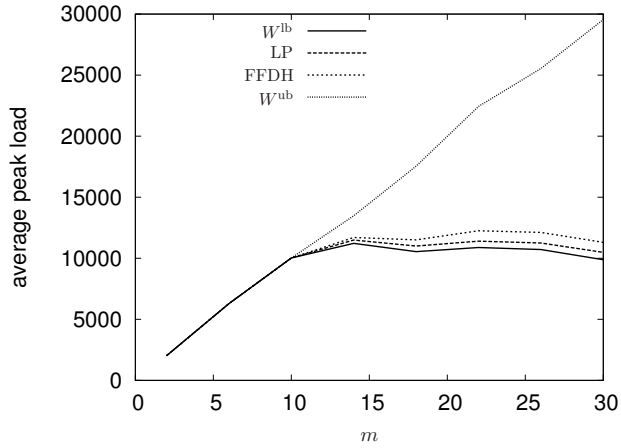


Figure 5.4: Average peak load obtained by the different techniques as a function of the number of loads m, with $U^{tot} = 10$.

generation algorithm presented in (Bini and Buttazzo [6]) 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 m and the peak load achievable by all methods converges to the maximum possible peak load w^{ub}, leading to $\eta \to 0$. Notice that when $U \geq m$, it holds $\eta = 0$.

Figure 5.4 shows the average peak load obtained by the different techniques as a function of the number of loads m when the total utilization is constant ($U^{tot} = 10$). It can be noticed that, when $U^{tot} \leq 10$, the peak load achieved by the optimized methods can not be better than W^{ub} for the same reason above: every load is generated with $U_i = 1$, and thus there is no opportunity to apply the scheduling of loads since each scheduling

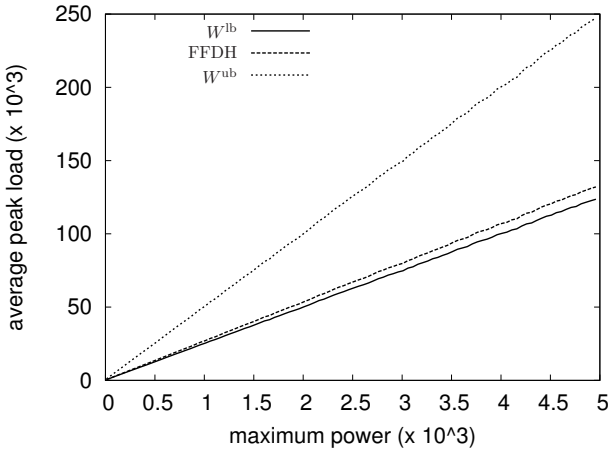


Figure 5.5: Average peak load obtained by the heuristic method as a function of the maximum possible power for each load; we considered $m = 100$, $U^{tot} = 50$ and the minimum possible power equal to 10.

group contains exactly one load. When $U^{tot} > 10$, the optimized methods guarantee an improvement that increases with m, accordingly with the results presented in Figure 5.3. Moreover, Figure 5.4 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 W^{lb} imposed by Observation 9. This characteristic behavior has been steadily detected throughout all experiments.

Finally, Figure 5.5 shows the average peak load obtained by the FFDH heuristic method as a function of the maximum possible power for each load. In this experiment we considered $m = 100$, $U^{tot} = 50$ 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 W^{ub}. 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 W^{lb}.

5.3 Residential loads modeling examples

This section provides examples of how real-time parameters can be used to model the behavior of electric loads, as well as applications that are suitable for being managed by the proposed techniques. The goal is to show the possibility to seamlessly manage sets of heterogeneous electric loads in a transparent manner. In fact, the following types of loads (as well as many others) can be coordinated altogether to limit the peak load once they are modeled in terms of timing parameters and constraints. These examples are partially taken from (Della Vedova et al. [22]).

5.3.1 HVAC systems and refrigerators

A typical goal of a Heating, Ventilation and Air Conditioning (HVAC) system is to keep the room temperature within the desired range. Therefore, heating or cooling is provided depending on the actual room temperature, which is affected by the temperature of the external environment. For a given external temperature the activation pattern of a HVAC can be suitably approximated by a periodic activity (Koutitas and Tassiulas [39]).

A simple but accurate model for a HVAC system is proposed in (Ramanathan and Vittal [61]), and it is recalled in the following. The adopted model describes a first order dynamic system, which has been proven to capture the behavior of HVAC loads accurately.

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = \frac{X_o(t) - X(t) - X_g s(t)}{\tau} \tag{5.13}$$

In (5.13), $X(t)$ is the internal room air temperature, $X_o(t)$ is the outside air temperature, X_g is the temperature gain of the air-conditioner, and τ is the actual time constant of the room expressed in minutes. The above model characterizes the behavior of a single HVAC. The controller must guarantee that the inside air temperature lays within a comfort range:

$$X(t) \in \left[X^{\min}, X^{\max}\right], \quad \forall t \tag{5.14}$$

In Section 4.6 it is shown by a worst-case analysis how to derive the required timing parameters, namely the period T and the utilization U, from a first order model of a load like the one expressed by (5.13)-(5.14). The external temperature is represented by a constant function $X_o(t) = X_o$. Basically, there exists a set of pair (U, T) for which the temperature is always kept within the comfort range for any possible activation pattern generated by the real-time scheduler using a given value for the (U, T) parameters. In particular, it is shown that U must be chosen within the range $\left[U^{\min}, U^{\max}\right]$, where:

$$U^{\min} = \max\left\{0, \frac{X_g + X^{\min} - X_o}{X_g}\right\} \tag{5.15a}$$

$$U^{\max} = \min\left\{\frac{X_g + X^{\max} - X_o}{X_g}, 1\right\} \tag{5.15b}$$

It is easy to show that a valid choice is $U = \frac{U^{\min}+U^{\max}}{2}$. Once the utilization U has be determined, the period T must be chosen such that both inequalities (5.16) hold,where $\overline{U} = 1 - U$.

$$X_o - X^{\min} > \frac{2X_o e^{UT/\tau} - X_g e^{2UT/\tau} + X_g e^{T(1+U)/\tau}}{1 - e^{T/\tau}} \tag{5.16a}$$

$$X_g - X_o + X^{\max} > \frac{2X_o e^{\overline{U}T/\tau} - X_g e^{2\overline{U}T/\tau} + X_g e^{T(2-\overline{U})/\tau}}{1 - e^{T/\tau}} \tag{5.16b}$$

Since a closed form of inequalities (5.16) is not available, valid values of T can be found using numerical methods. Following this procedure, it is possible to assign timing parameters to the electric load such that the temperature is always kept within the comfort range.

In (Shao et al. [63]) this type of loads is called *controllable load*, since it can be shed to achieve the peak load reduction. Preferably, in this specific case a feedback control scheme should be integrated to regulate the temperature upon external temperature variations (see Section 4.4). Similar considerations apply to refrigerators.

5.3.2 Lighting

The lightning system of a building is another type of load that can be modeled in terms of timing parameters. The lights of a building (e.g., corridor lights) may require to be turned on in the evening, for example at 8:30pm, and turned off in the morning, e.g., at 7:00am. Such timing parameters represent the request time and the absolute deadline respectively. In this case, no interruption (i.e., preemption) is allowed on the service of the lighting system. As a consequence of above considerations, the load has a period of 24h, an activity time of 10:30h, and a relative deadline equal to the activity time. In this way, the scheduler will always activate the load at the beginning of the period. Moreover, no temporary interruptions will be inserted while the lights are switched on. On the other hand, these requirements has a negative impact on the level of concurrency of load activations. In other words, there is no flexibility in the activation of the load. In (Shao et al. [63]) this type of loads is called *critical load*.

The presence of such type of loads, i.e., with limited or no scheduling flexibility, leads to an increase in the number of concurrently active loads, and thus of the peak load. Nevertheless, the interesting aspect is that it is possible to transparently integrate these loads in the coordination framework, so that the operation of controllable loads can be suitably coordinated.

5.3.3 Household appliances

Typical household devices like washing machines, dryers, dishwashers, have each a peculiar duty cycle and timing characteristics. These electric loads are characterized by an activation time C and a deadline D. The activation time is the amount of time needed by the device to complete the working cycle. It can vary depending on the actual cycle and it is assumed to be known by the scheduler at the request time. In simple terms, the request time can be seen as the instant when the ON button is pressed by the user. The deadline is the time instant in which the user wants to have the cycle completed. The deadline is also assumed to be known by the scheduler at the request time and it is set by the user. For example, the user may set the time instant (the deadline) at which the laundry has to be ready. 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. However, a certain degree of flexibility is usually available in the working cycles of household appliances, and programmable devices are already available to control the activation of electric loads depending on the energy prices in the stock market (MVV Energie Group [53]). As an example, these devices are used to control dishwashers in domestic environments, where postponing a

few hours the time at which the washing process is started does not arise any issue to the user. The same approach could be adopted for washing machines and dryers.

Regarding the preemption in the activation of a task/load, many works in literature of real-time systems address this issue (Buttazzo [7]). Basically, it is a common practice to define time intervals in which a load can not be preempted. Alternatively, it is possible to define a maximum number of preemptions during each load activation, or to define a minimum time interval after a preemption in which the load can not be preempted again. Depending on the load, one or more of the aforesaid techniques can be applied.

For example, a common washing machine has a working cycle that can be divided into stages. Figure 5.6 shows the sequence of working stages of a common washing machine, together with the power demand corresponding to each stage. Such power demand has been derived from measurements on a commercial appliance. Referring to Figure 5.6, stage 1 is the soaking phase. Its execution can be preempted almost anytime (it is Fully Preemptable – FP), since it is a passive stage in which the laundry is soaking without any particular action by the machine. During this stage there is the water heating, which is the most power demanding operation of the whole working cycle. The next stage (stage 2) is the washing phase. During this operation the laundry is actually cleaned by the rotation of the basket with hot water. The washing stage can be interrupted for a short while (Limited Preemptable – LP), because longer stops may lead to the resettlement of the dirt. The stage 3 is the rinse, which can be interrupted with no problems. The last stage is the spin-dryer phase. The operation is non-preemtable (NP) in order to maintain the spin. Similar considerations can be applied to the working cycle of other appliances.

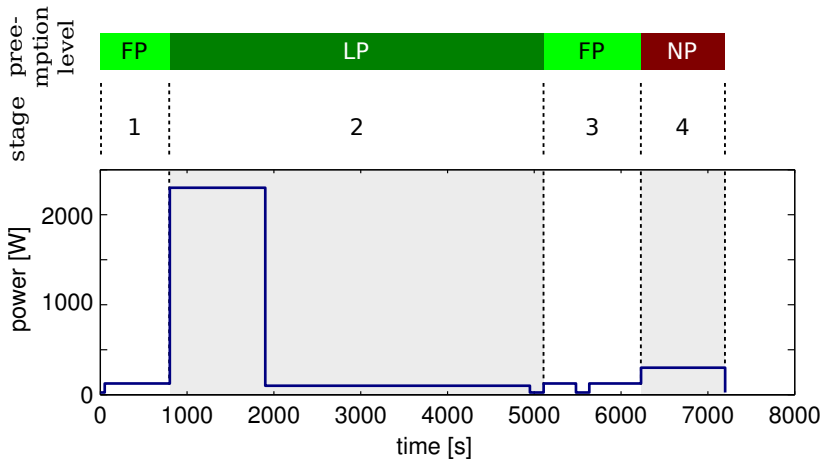


Figure 5.6: Power demand of a typical washing machine working cycle. Stages: 1 - soak, 2 - wash, 3 - rinse, 4 - spin. Preemption opportunity: FP - fully preemptable, LP - limited preemptable, NP - non-preemptable.

5.3.4 Electric cars

Current hybrid and fully electric cars are equipped with batteries that must be periodically recharged. As the penetration of electric vehicles increases, the problem of balancing the power demand for their charging becomes more prominent (He et al. [33]). In (Kempton and Tomić [37]), the authors discuss the relevance of integrating vehicle batteries in large scale electric networks. Recharge times may vary depending on different factors, such as manufacturing technology, battery capacity, and sinking current. Nevertheless, the whole recharging process takes typically a few hours. Moreover, the charging process can be suitably modeled as a periodic activity. A reasonable example of periodic behavior of electric vehicles recharging is made by platoons of business vehicles. In the encompassed scenario, one or more

cars that may stop working at 7:30pm and should be fully recharged for the next working day, between 5:30am and 7:30am depending on the vehicle. Each car consumes a given amount of power while recharging, and no power otherwise. In this scenario, each electric load associated to the battery recharging process has a period of 24h, a relative deadline in the range $[10 \dots 12]$h, and a maximum active time that is related to the duration of the battery recharge (for example, a few hours).

6

Case studies results

Contents

This chapter presents some case studies that evaluate the performance of the proposed Electric Load Management method, which is based on Real-Time Physical Systems, through simulations characterized by realistic parameters.

6.1 Effect of model errors

This section presents the application of the proposed modeling technique to a simple system composed by 3 electric devices. Physical systems asso-

i	α_i^{off}	α_i^{on}	P_i	$\overline{x}_i$	$x_i^{\min}$	$x_i^{\max}$
1	1.0	3.0	1	-3	-4	-2
2	1.5	6.0	2	-3	-4	-2
3	2.0	8.0	3	-12	-14	-10

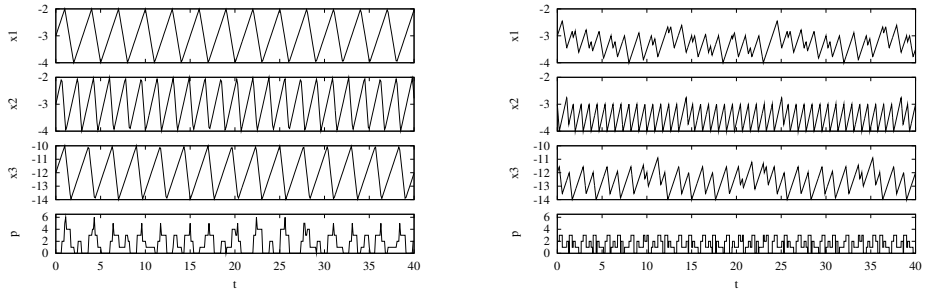
Table 6.1: Values of physical parameters and user requirements used for the simulation

ciated with devices are modeled as a decoupled switched integrators RTPS with constant range user requirements (see Section 4.2) having parameters depicted in Table 6.1. This kind of systems can represent a fridge where the state variable x_i is the internal temperature (assumed to be uniform, and expressed in Celsius degree) that must be kept within a desired working range. Temperature increase and decrease are approximated with functions linear in time. This linear approximation can be considered accurate when the working temperature is far enough from both the ambient and the of cooling fluid temperature. Values α_i^{off} and α_i^{on} represent the temperature variation speed and are expressed in Celsius degree per hour (°C/h). The nominal powers P_i of electric devices can be considered a-dimensional and they have been chosen such way (i.e., $P_i = i$) to be able to detect which load is active just looking at the power demand $w(t)$ chart. Each temperature x_i must remain within the bounds $[x_i^{\min}, x_i^{\max}]$ specified by user requirements.

For the sake of simplicity, we limit this example to a set of loads having $U^{\text{tot}} < 1$. This allows to clearly show how the absence of a proper management of load activations brings to the highest possible peak load in the worst case, while our approach improves (i.e., it decreases) the peak load. It is worth to note that, being $U^{\text{tot}} \leq 1$, a real-time scheduling algorithm such as EDF is able to schedule the load set without any concurrent activation of loads.

i	U_i	T_i^*	C_i
1	0.33	1.50	0.50
2	0.25	0.88	0.22
3	0.25	1.33	0.33

Table 6.2: Values of real-time parameters calculated for the simulation



(a) (b)

Figure 6.1: Comparison between systems driven by hysteresis controller (left) and RTPS-scheduler (right). Loads parameters are specified in Table 6.1. The peak shaving effect is notable in the lower chart.

Figure 6.1(a) shows the behavior of a simple on/off control technique applied to the considered load set. Each load is independently controlled by an hysteresis controller so that the load is turned on when the state variable reaches the upper bound of the working range, and it is kept active until the lower bound is reached. This control strategy, which is simple but rather common in practice, easily allows to individually maintain the state variable within the required working range. However, since the activation of loads is not coordinated, it is possible that more than one load is active at the same time. As a consequence, the peak load increases. In fact, several times in the depicted time range, the three considered loads are activated simultaneously, thus determining a peak load $W = \sum P_i = 6$.

The same three loads have been managed with the RTPS-technique. Real-time parameters, reported in Table 6.2, are calculated using the results presented in Section 4.2. Figure 6.1(b) shows the behavior of state variables and the instantaneous consumed power $w(t)$. The figure shows that state variables are confined within the desired working ranges, while the load scheduling allows to limit the peak load to $W = \max_i P_i$. In fact, in this example the load set is scheduled using the EDF scheduling algorithm. Therefore, since the total utilization is $U^{\text{tot}} = 0.83 \leq 1$, EDF guarantees that only one load is active at any given time.

Comparing both charts in Figure 6.1, it becomes evident the peak shaving effect of the proposed approach.

Next, we illustrate the issues arising in presence of errors, and provides an example of application of the adaptive technique proposed in "Switched integrators with modeling errors" Section 4.3. Load parameters are reported in Table 6.1. The total utilization is around 83%, thus EDF is able to generate a schedule without any simultaneous load activation. In this case, however, the value of each α_i^{on} is affected by a constant bias equal

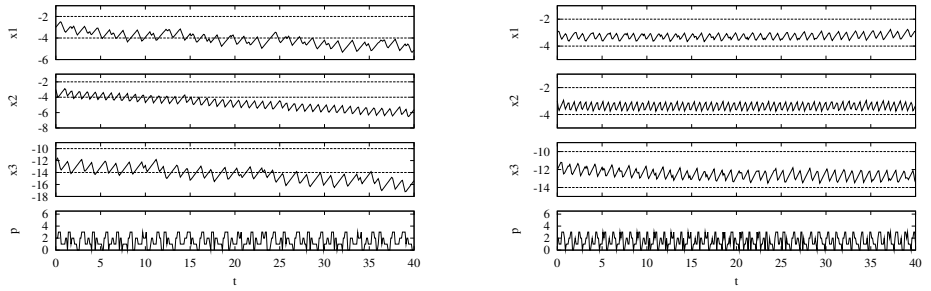


Figure 6.2: Comparison between systems affected by errors driven by a open-loop RTPS scheduler (left) and an adaptive RTPS scheduler (right). The adaptive scheduler is able to maintain the state variables within the working range.

to +4% of the original value. This means that real-time parameters are calculated using the values of α_i^{on} depicted in Table 6.1, giving the results reported in Table 6.2. However, the true value that determines the state variable behavior is slightly larger. The result is depicted in Figure 6.2(a): all state variables drift from their desired working range.

Figure 6.2(b) shows the schedule of the same load set using the adaptive technique proposed in Section 4.3. The time quantum is set as $\tau = 0.1$h (i.e., 6 minutes). Therefore, there is a minimum distance in time of 12 between two consecutive load activation, which is a reasonable value for common electric motors[1]. In this case, activation times are dynamically adapted at every request time, with the result that all state variables remain within their working ranges. The achieved peak load of power consumption is the same as in Figure 6.2(a) since, being $U^{\text{tot}} \approx 91\% < 1$, all loads can

[1]Electric motors suffer from too frequent activations due to the high current required during the start up.

i	U_i	$\tilde{T}_i$	$\tilde{C}_i$
1	0.33	0.6	0.2
2	0.25	0.4	0.1
3	0.33	0.6	0.2

Table 6.3: Values of real-time parameters calculated in presence of errors.

always be activated by EDF at non-overlapping time instants. Notice that, as depicted in Table 6.3, the total utilization of each load is larger than in case of absence of errors.

6.2 Uncoordinated vs RTPS scheduling

This section provides a statistical evaluation of benefits derived by using RTPS scheduling technique with FFDH grouping (see Section 5.2.3) against the average behavior of load activations in absence of coordination.

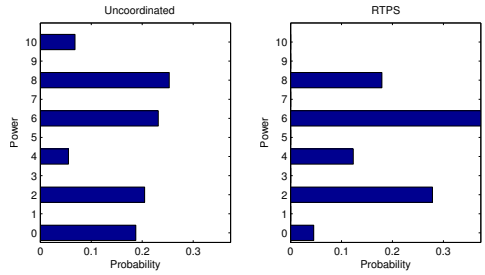
We study the distribution of the total power consumption during the system lifespan. For example, the maximum power consumption occurs when all the loads are active at the same time instant. This event occurs in a certain time instant with a probability that depends on the loads' utilization. Assuming an infinite system lifespan, this probability matches the percentage of time in which the event takes place.

If load activations are independent, the occurrence of the maximum power consumption is the product of utilization of active loads. Conversely, a load activation schedule based on RTPS technique prevents the simultaneous activation of loads in the same scheduling group.

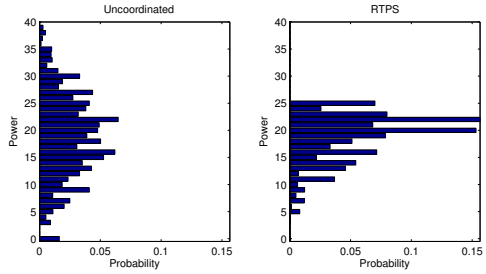
Figure 6.3 shows the distribution of the total power consumption during the system lifespan in systems composed by three different load sets,

one for each plot. Nominal parameters P and U are shown in each caption. These examples show that the RTPS scheduling method reduces the peak load while compressing the power consumption distribution around the average value. It follows the list of parameters used: $P = (2, 2, 6)$kW and $U = (0.38, 0.32, 0.55)$ for Figure 6.3(a), $P = (1, 3, 3, 4, 4, 5, 6, 7)$kW and $U = (0.57, 0.38, 0.45, 0.21, 0.49, 0.14, 0.67, 0.43)$ for Figure 6.3(b), $P = (1, 2, 2, 3, 4, 4, 5, 7, 8, 8, 9)$kW and $U = (0.24, 0.28, 0.56, 0.54, 0.51, 0.62, 0.30, 0.46, 0.16, 0.69, 0.44)$ for Figure 6.3(c).

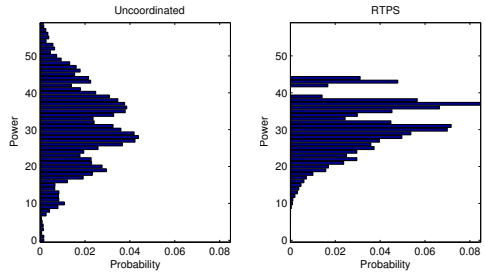
Finally, Figure 6.4 shows the power distributions with and without using the scheduling techniques as a function of the number of loads. Each boxplot is the average of the power distribution of 100 randomly generated load sets. Nominal power and utilization have been generated with a uniform distribution in $[1, 10]$ and $[0.1, 0.7]$, respectively. Again, this plot confirms that RTPS technique compresses the power consumption distribution and significantly reduces the peak load.



(a) Results with $n = 3$ loads.



(b) Results with $n = 7$ loads.



(c) Results with $n = 11$ loads.

Figure 6.3: Example of power distribution over time consumed by 3, 7 and 11 loads, without using RTPS technique (left) and using RTPS technique (right). The peak is reduced by 20%, 37% and 26%, respectively.

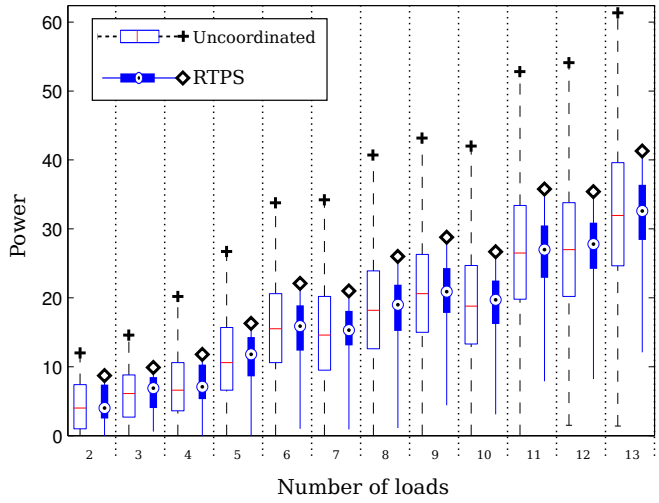


Figure 6.4: Distribution of power consumption over time as a function of the number of loads. For each number of loads, the average of 100 random load sets is plotted.

6.3 Large HVAC systems

This section evaluates the performance of the proposed approach using realistic parameters. In this scenario, a set of on/off air-conditioning systems described by the model in (5.13) is considered. Similarly to (Ramanathan and Vittal [61]), from the viewpoint of generating different realistic operating parameters, Monte Carlo simulations are applied in this paper. By assuming uncertainties in different variables (τ, X, X_o, X_g) that closely resemble real-life operating conditions, Monte Carlo simulations are performed through repeated sampling of uncertain variables.

Each simulation run has been initialized with pre-specified stochastic parameters, chosen with the following method:

- internal temperature distribution – Normal distribution with mean 72 F and standard deviation 12 F, i.e. $X(0) \in N(74, 12)$ [F];
- air-conditioner model parameters – $\tau \in N(64, 5)$ [min], $X_g \in N(30, 10)$ [F];
- outside air temperature – uniform distribution between 75 F and 90 F $X_o \in [75, 90]$ [F];
- desired temperature range – $X^{\min} = 70$ [F], $X^{\max} = 76$ [F].

Regarding the power request of each load, 5 different air-conditioner sizes are assumed, installed depending on the room dimension – $P \in (1.2, 2.5, 3.0, 4.5, 6.0)$ [kW].

For the generated load set of 10 000 air-conditioners, the timing parameters values of each device are calculated, i.e., utilization, period and activation time, (U, T, C) for each load in the way illustrated in Section 5.3.1.

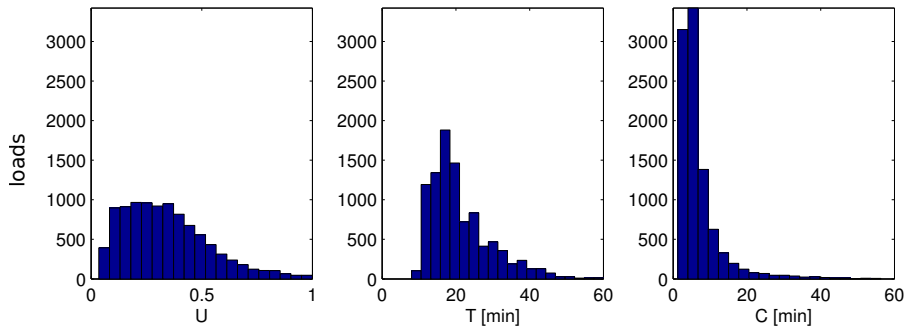


Figure 6.5: Distribution of timing parameters values obtained from physical parameters in the 10k-loads Monte Carlo simulation.

In particular, we set $U = \frac{U^{\max}+U^{\min}}{2}$ and then T is selected as the maximum value that satisfies (5.16). Figure 6.5 depicts the resulting histograms of timing parameters values distributions. Utilizations range from 0, i.e. always off, to 1, i.e. always on. Periods vary from 10 to 60 minutes.

Based on above parameters, the adopted scheduling approach is based on the FFDH partitioning scheme (see Section 5.2.3). Under this scheme, a dedicated heuristic is used to divide the set of loads into a set of independent groups (called scheduling groups), where the total utilization of each group is $U \leq 1$. In this way, the EDF scheduling algorithm can be used to optimally manage the loads in each scheduling group.

In order to evaluate the performance of the proposed approach, namely RTPS-controlled system, a comparison is carried out with a traditional hysteresis control. In a hysteresis-controlled system, each air-conditioner is turned on when the internal room air temperature is greater than the upper thermostat set-point $X^{\max}$ and turned off whenever this temperature reaches the lower thermostat set-point $X^{\min}$.

Figure 6.6 shows the results of a simulation run having a 1 000 hours

timespan, a time-base of 1 second, and the aforesaid 10 000 loads. The first note is that this simulation, performed as a compiled C program on a common notebook, took only a few tens of seconds to complete. This fast execution is due to the low computational complexity of the proposed method. The figure reports the comparison of the achieved peak load with respect to the theoretical peak load bound and the registered peak load. It also compares the average power demand. The theoretical peak load bound is equal to the sum of all loads power in the hysteresis-controlled system, which is the worst-case condition that may arise when load activations are not coordinated. On the other hand, the registered peak load is the highest power demand registered during the simulation. As expected, the average power demand is quite similar between the two methods, in fact the average power demand is directly related to the total amount of energy required to the system and it does not depend on the control strategy. The registered values and the theoretical bounds for the peak load are instead significantly different. It may be argued that the theoretical bound for the hysteresis-controlled system, which is given by a very unlikely worst-case condition, is not meaningful. However, on one hand it represents a possible situation, whose likelihood increases with the lifetime of the system. On the other hand, it allows a comparison between worst-case and average-case conditions.

As a result, even if the peak load actually registered in the simulation is lower than the worst-case bound in the hysteresis case, it is significantly greater than the one registered in the RTPS-controlled case. The improvement on the peak load registered in the simulation is around 37%. The overall effect of the RTPS control is to maintain the power demand steadily close to the average demand. Therefore, the "peak to average power ratio", which is a well-known parameter for Demand-Side Management systems,

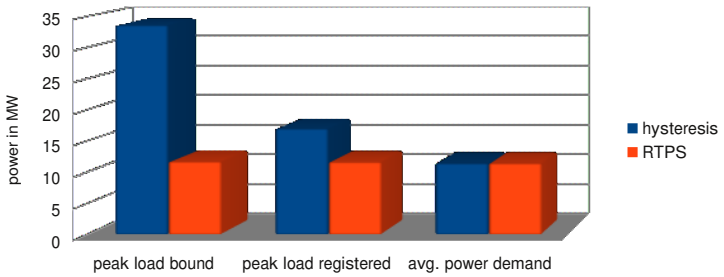


Figure 6.6: Comparison between hysteresis-controlled and RTPS-controlled systems: aggregate results of Monte Carlo simulations.

can significantly improve.

In the next experiment, it has been verified that the peak load bound reduction depends on the load utilizations. In particular, the lower the average loads utilization, the greater the improvement on the peak load bound given by the proposed approach (Figure 6.7). The rational behind this behavior is that, when the utilization of a load increases, the flexibility to schedule its activation within a period time frame decreases. Therefore, this reduces the effectiveness of the peak load reduction. In other words, this effect is due to the fact that the load partitioning technique is more effective when loads have small utilization factors.

In Figure 6.8 it is shown the peak load as a function of the total number of loads. We limit the chart between 1 and 100 loads for giving and idea of the trends, which increase linearly. Loads' parameters have been generated with the same random functions described above. The actual (recorded) peak load generated by the RTPS-based control is able to reduce the peak load in average by 35% with respect to the actual (recorded) peak load in absence of coordination, which represents a normal working condition.

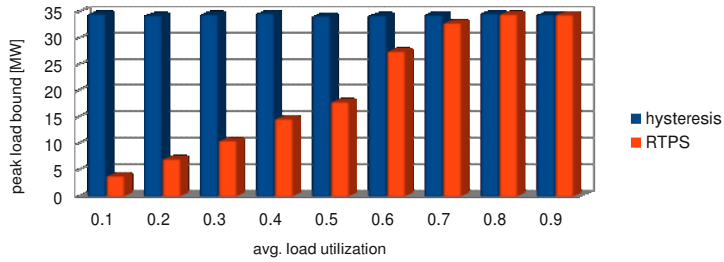


Figure 6.7: Comparison between hysteresis-controlled and RTPS-controlled systems varying average loads utilization.

On the other hand, the peak load is reduced by up to 60% with respect to the worst possible case (theoretical bound) of the hysteresis-controlled case. Finally, it is worth to note that the recorded peak load of the RTPS-controlled system is very close to its theoretical bound. This means that the worst situation in the partitioned scheme (i.e., when the most power-consuming loads in every scheduling group are simultaneously activated) do happen often.

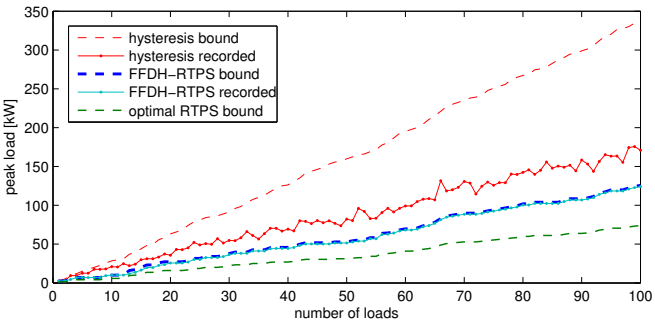


Figure 6.8: Comparison between hysteresis- and RTPS-control methods. The RTPS-controlled actual behavior, during the simulation of 120 hours, reduces the peak load of the 35% in respect of hysteresis control in regular working conditions. The improvement on the theoretical bound associated with worst case working conditions is around 60%.

6.4 Effect on power demand curve

The following example aims to show the effect of the proposed method on the power demand curve by controlling the schedule of 100 on/off devices. Each device acts on a subsystem (system model of "Switched integrators with modeling errors" Section 4.3), whose parameters have been randomly generated from a uniform distribution within the following intervals:

- $\alpha_i^{\text{on}} \in [1.0, 10.0] \pm 5\%$,
- $\alpha_i^{\text{off}} \in [3.0, 5.0] \pm 2\%$,
- $\delta_i^x \in [0.01, 0.1]$,
- $x_i^{\min} = -1$,
- $x_i^{\max} = 1$,
- $\bar{x}_i = 0$.

Operationally, the RT parameters C_i^* and T_i have been calculated for each device as described in the aforesaid modeling section. Then, the load set has been partitioned into scheduling groups using the FFDH heuristic. Scheduling groups are built such that the utilization of each group is less than 1. This allows to successfully use the EDF scheduling algorithm to schedule the loads in each group. The result of the power consumed by the system over time is depicted in Figure 6.9. The performance in terms of peak load reduction is compared with a common hysteresis controller. An hysteresis controller simply turns on the load when the state variable gets to the upper bound, while it turns the device off when the lower bound is reached. For instance, this is the typical behavior of a thermostat. There is no coordination between loads. Figure 6.9 shows a smoother behavior of the power demand in case of control using RTPSs. Numerical results reported in Table 6.4 show an actual reduction of the peak load of around 30%. The smoother behavior of the power profile is confirmed by a smaller standard deviation of the distribution of samples. As expected, the mean power is substantially the same in the two cases, since it is essentially related with the consumed energy. Finally, the worst-case peak load bound is much lower. The bound of the hysteresis method is calculated in the case of having all loads active at the same time, while in case of the RTPS control the bound is equal to the sum of the most power-consuming load in each scheduling group.

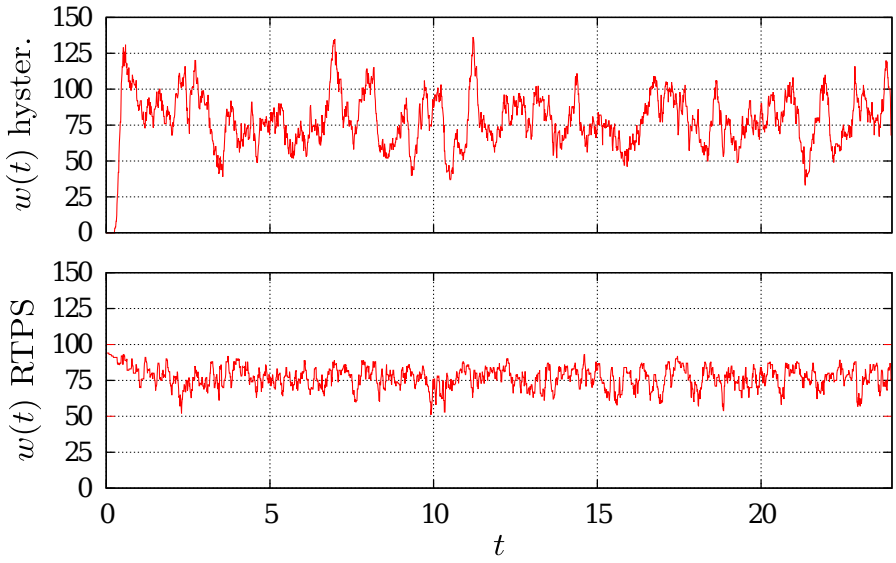


Figure 6.9: Comparison of the total power consumption of 100 randomly generated loads controlled by an hysteresis controller, and a RTPS controller. It is remarkable the flattening effect due to the coordinated scheduling.

	hysteresis	RTPS	improvement
$\max w(t)$	136.0	94.0	-31%
mean $w(t)$	77.9	77.1	$==$
std. dev. $w(t)$	19.4	7.4	-62%
peak load bound	337.0	95.0	-71%

Table 6.4: Summary of relevant values extracted from Figure 6.9

7 Conclusion

This thesis presented Real-Time Physical Systems: a new class of real-time systems where dynamical system are scheduled instead of computing tasks. Those systems are the theoretical basis for a novel Demand-Side Management framework based on real-time scheduling algorithms. The goal is to limit the peak load of the power demand of a set of electric devices, while meeting user requirements, that is, without compromising the quality of service.

A detailed systematic literature review in Chapter 2 has given an overview over the electric load management state of the art. It has shown, among other things, that there is room for new DSM approaches that can address the need of low complexity algorithms that are able to manage large set of heterogeneous devices. The proposed approach goes in this direction.

We have shown that the RTPS method is able to find an approximate solution to the peak load minimization problem. Unlike traditional optimization procedures, it has a very low computational cost that enable the proposed approach to manage hundreds of devices. Besides excellent

scalability properties, the proposed framework can also boast of flexibility, because RTPS models can capture many different types of loads. Those models include: affine systems, switched systems with integrator dynamics, stochastic disturbance, modeling errors, and industrial processes. Therefore, RTPS framework can manage heterogeneous electric loads together.

Simulation results based on realistic parameters showed outstanding performance for the proposed method, such as the ability to reduce the peak load of a large air conditioning system by the 35% in normal working conditions, and up to 60% with respect to worst case situations, without affecting thermal comfort for occupants. Hence, the RTPS approach proves to be a promising solution to realize an effective Demand-Side Management framework.

List of Figures

List of Definitions and Theorems

Definitions

Theorems

Bibliography

[1] S. Ashok and R. Banerjee. An optimization mode for industrial load management. *Power Systems, IEEE Transactions on*, 16(4):879–884, nov 2001. ISSN 0885-8950. doi: 10.1109/59.962440. (Cited on page 133)

[2] S. Baruah. The limited-preemption uniprocessor scheduling of sporadic task systems. In *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, pages 137–144, July 2005. doi: http://doi.ieeecomputersociety.org/10.1109/ECRTS.2005.32. (Cited on pages 63 and 64)

[3] 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. (Cited on page 155)

[4] S. K. Baruah, A. K. Mok, and L. E. Rosier. Preemptively scheduling hard-real-time sporadic tasks on one processor. In *Real-Time Sys-*

tems Symposium, 1990. Proceedings., 11th, pages 182–190, Lake Buena Vista (FL), U.S.A., dec 1990. (Cited on page 61)

[5] M. Bertogna and S. Baruah. Limited preemption edf scheduling of sporadic task systems. *Industrial Informatics, IEEE Transactions on*, 6(4):579–591, 2010. ISSN 1551-3203. doi: 10.1109/TII.2010.2049654. (Cited on page 63)

[6] E. Bini and G. C. Buttazzo. Biasing effects in schedulability measures. In *Proceedings of the* 16^{th} *Euromicro Conference on Real-Time Systems*, pages 196–203, Catania, Italy, jun 2004. (Cited on pages 158 and 160)

[7] G. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer, 2011. (Cited on pages 51, 52, 53, and 166)

[8] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni. Elastic scheduling for flexible workload management. *IEEE Transactions on Computers*, 51(3):289–302, mar 2002. (Cited on page 100)

[9] G. C. Buttazzo, M. Bertogna, and G. Yao. Limited preemptive scheduling for real-time systems. a survey. *Industrial Informatics, IEEE Transactions on*, 9(1):3–15, 2013. ISSN 1551-3203. doi: 10.1109/TII.2012.2188805. (Cited on page 65)

[10] D. Callaway and I. Hiskens. Achieving controllability of electric loads. *Proceedings of the IEEE*, 99(1):184–199, jan. 2011. ISSN 0018-9219. doi: 10.1109/JPROC.2010.2081652. (Cited on page 103)

[11] J. Carpenter, S. Funk, P. Holman, A. Srinivasan, J. Anderson, and S. Baruah. A categorization of real-time multiprocessor scheduling problems and algorithms. In *HANDBOOK ON SCHEDULING ALGORITHMS, METHODS, AND MODELS*. Chapman Hall/CRC, Boca, 2004. (Cited on page 65)

[12] J.-L. Chen and Y.-Y. Hsu. An expert system for load allocation in distribution expansion planning. *Power Delivery, IEEE Transactions on*, 4(3):1910–1918, Jul 1989. ISSN 0885-8977. doi: 10.1109/61.32690. (Cited on page 47)

[13] C.-M. Chu and T.-L. Jong. A novel direct air-conditioning load control method. *IEEE Transactions on Power Systems*, 23(3):1356–1363, 2008. (Cited on page 48)

[14] C.-M. Chu, T.-L. Jong, and Y.-W. Huang. A direct load control of air-conditioning loads with thermal comfort control. In *IEEE Power Engineering Society General Meeting*, pages 664–669Vol.1, June 2005. doi: 10.1109/PES.2005.1489078. (Cited on page 47)

[15] E. Coffman, Jr., M. Garey, D. Johnson, and R. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826, 1980. doi: 10.1137/0209062. (Cited on page 156)

[16] J. Daafouz, M. D. D. Benedetto, V. D. Blondel, G. Ferrari-Trecate, L. Hetel, M. Johansson, A. L. Juloski, S. Paoletti, G. Pola, E. D. Santis, and R. Vidal. Switched and piecewise affine systems. In *Handbook of Hybrid Systems Control*. Cambridge University Press, 2009. ISBN 9780511807930. (Cited on pages 70 and 71)

[17] 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*, pages 487–488, oct. 2008. doi: 10.1109/SASO.2008.64. (Cited on page 47)

[18] M. L. Della Vedova and T. Facchinetti. Real-time scheduling for industrial load management. In *Proceeding of the 2nd ENERGYCON Conference & Exhibition*, September 2012. (Cited on page 131)

[19] M. L. Della Vedova and T. Facchinetti. Feedback scheduling of real-time physical systems with integrator dynamics. In *Proceedings of 17th*

IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), September 2012. (Cited on pages 80 and 90)

[20] M. L. Della Vedova and T. Facchinetti. Real-time scheduling for peak load reduction in a large set of HVAC loads. In *ENERGY 2013, The Third International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, pages 161–166, Lisbon, Portugal, March 2013. IARIA. ISBN 978-1-61208-259-2. (Cited on pages 116 and 149)

[21] 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)*, pages 41–49, November 2010. (Cited on page 116)

[22] 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)*, pages 1117–1123, July 2011. doi: 10.1109/IWCMC.2011.5982697. (Cited on page 162)

[23] M. Dertouzos and A. Mok. Multiprocessor online scheduling of hard-real-time tasks. *Software Engineering, IEEE Transactions on*, 15(12): 1497–1506, 1989. ISSN 0098-5589. doi: 10.1109/32.58762. (Cited on page 67)

[24] M. L. Dertouzos. Control robotics: The procedural control of physical processes. *Information Processing*, 74, 1974. (Cited on page 60)

[25] S. K. Dhall and C. L. Liu. On a real-time scheduling problem. *Operation Research*, 26(1):127–140, jan 1978. (Cited on page 65)

[26] K. Dietrich, J. Latorre, L. Olmos, and A. Ramos. Demand response in an isolated system with high wind integration. *IEEE Transactions on Power Systems*, 27(1):20–29, 2012. (Cited on page 48)

[27] K. Elgazzar, H. Li, and L. Chang. A centralized fuzzy controller for aggregated control of domestic water heaters. pages 1141–1146, 2009. (Cited on page 47)

[28] T. Ericson. Direct load control of residential water heaters. *Energy Policy*, 37(9):3502–3512, 2009. (Cited on page 47)

[29] T. Facchinetti and M. L. Della Vedova. Real-time modeling and control of a cyber-physical energy system. In *Proceedings of the First International Workshop on Energy Aware Design and Analysis of Cyber Physical Systems (WEA-CPS)*, April 2010. (Cited on page 80)

[30] T. Facchinetti and M. L. 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*, 7(4):689–698, November 2011. ISSN 1551-3203. doi: 10.1109/TII.2011.2166787. (Cited on page 102)

[31] T. Facchinetti, E. Bini, and M. Bertogna. Reducing the peak power through real-time scheduling techniques in cyber-physical energy systems. In *Proceedings of the First International Workshop on Energy Aware Design and Analysis of Cyber Physical Systems (WEA-CPS)*, April 2010. (Cited on page 149)

[32] M. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, San Francisco, 1979. ISBN 0-7167-1045-5. (Cited on page 57)

[33] Y. He, B. Venkatesh, and L. Guan. Optimal scheduling for charging and discharging of electric vehicles. *Smart Grid, IEEE Transactions on*, PP(3):1–11, 2012. ISSN 1949-3053. doi: 10.1109/TSG.2011.2173507. (Cited on page 167)

[34] IEEE. IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power

System (EPS), End-Use Applications, and Loads. *IEEE Std 2030-2011*, pages 1–126, 2010. doi: 10.1109/IEEESTD.2011.6018239. (Cited on page 29)

[35] A. Kailas, V. Cecchi, and A. Mukherjee. A survey of communications and networking technologies for energy management in buildings and home automation. *Journal of Computer Networks and Communications*, 2012. (Cited on page 32)

[36] I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, second ed. edition, 1991. ISBN 0387976558. (Cited on page 103)

[37] W. Kempton and J. Tomić. Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *Journal of Power Sources*, 144:280–294, June 2005. (Cited on page 167)

[38] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15, 2009. (Cited on page 34)

[39] G. Koutitas and L. Tassiulas. A delay based optimization scheme for peak load reduction in the smart grid. In *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*, page 7. ACM, 2012. (Cited on page 162)

[40] C. Kurucz, D. Brandt, and S. Sim. A linear programming model for reducing system peak through customer load control programs. *Power Systems, IEEE Transactions on*, 11(4):1817–1824, nov 1996. ISSN 0885-8950. doi: 10.1109/59.544648. (Cited on page 47)

[41] M. Kuzlu, M. Pipattanasomporn, and S. Rahman. Hardware demonstration of a home energy management system for demand response applications. *Smart Grid, IEEE Transactions on*, 3(4):1704–1711, dec.

2012. ISSN 1949-3053. doi: 10.1109/TSG.2012.2216295. (Cited on page 50)

[42] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems – A Cyber-Physical Systems Approach*. http://LeeSeshia.org, UC Berkeley, 2011. doi: ISBN978-0-557-70857-4. (Cited on pages 25 and 70)

[43] T.-F. Lee, M.-Y. Cho, Y.-C. Hsiao, P.-J. Chao, and F.-M. Fang. Optimization and implementation of a load control scheduler using relaxed dynamic programming for large air conditioner loads. *Power Systems, IEEE Transactions on*, 23(2):691–702, may 2008. ISSN 0885-8950. doi: 10.1109/TPWRS.2008.919311. (Cited on page 47)

[44] Z. Li, P.-C. Huang, A. K. Mok, T. Nghiem, M. Behl, G. Pappas, and R. Mangharam. On the feasibility of linear discrete-time systems of the green scheduling problem. In *Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd*, pages 295–304, 29 2011-dec. 2 2011. doi: 10.1109/RTSS.2011.34. (Cited on page 49)

[45] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the Association for Computing Machinery*, 20(1):46–61, January 1973. (Cited on page 59)

[46] A. Lodi, S. Martello, and M. Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2): 241–252, 2002. ISSN 0377-2217. doi: 10.1016/S0377-2217(02)00123-6. (Cited on page 155)

[47] A. Lodi, S. Martello, and D. Vigo. Models and bounds for two-dimensional level packing problems. *Journal of Combinatorial Optimization*, 8(3):363–379, 2004. (Cited on page 150)

[48] B. Lu and M. Shahidehpour. Short-term scheduling of battery in a grid-connected pv/battery system. *IEEE Transactions on Power Systems*, 20(2):1053–1061, 2005. (Cited on page 103)

[49] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves. Model predictive control for the operation of building cooling systems. *Control Systems Technology, IEEE Transactions on*, 20(3):796–803, 2012. ISSN 1063-6536. doi: 10.1109/TCST.2011.2124461. (Cited on page 49)

[50] D. Maly and K. Kwan. Optimal battery energy storage system (bess) charge scheduling with dynamic programming. *Science, Measurement and Technology, IEE Proceedings*, 142(6):453–458, nov 1995. ISSN 1350-2344. doi: 10.1049/ip-smt:19951929. (Cited on page 103)

[51] A. Molina-Garcia, M. Kessler, J. Fuentes, and E. Gomez-Lazaro. Probabilistic characterization of thermostatically controlled loads to model the impact of demand response programs. *Power Systems, IEEE Transactions on*, 26(1):241–251, Feb. 2011. ISSN 0885-8950. doi: 10.1109/TPWRS.2010.2047659. (Cited on page 47)

[52] 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. (Cited on page 25)

[53] MVV Energie Group. MVV Energie Group, 2009. http://www.mvv-energie.de/cms/konzernportal/de/mvv_energie_gruppe/mvv_energie_/innovation/energiebutler/Energiebutler.jsp. (Cited on page 165)

[54] G. Newsham, B. Birt, and I. Rowlands. A comparison of four methods to evaluate the effect of a utility residential air-conditioner load control program on peak electricity use. *Energy Policy*, 39(10):6376–6389, 2011. (Cited on page 48)

[55] T. Nghiem, M. Behl, R. Mangharam, and G. Pappas. Green scheduling of control systems for peak demand reduction. In *Decision and Control*

and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, pages 5131–5136, 2011. doi: 10.1109/CDC.2011.6161164. (Cited on page 49)

[56] T. Nghiem, M. Behl, G. Pappas, and R. Mangharam. Green scheduling: Scheduling of control systems for peak power reduction. In *Green Computing Conference and Workshops (IGCC), 2011 International*, pages 1–8, July 2011. doi: 10.1109/IGCC.2011.6008555. (Cited on page 49)

[57] T. X. Nghiem. *Green Scheduling of Control Systems*. PhD thesis, Electrical and Systems Engineering Department, University of Pennsylvania, 2012. (Cited on page 49)

[58] Y. Oh and S. Son. Allocating fixed-priority periodic tasks on multiprocessor systems. *Real-Time Systems*, 9(3):207–239, 1995. ISSN 0922-6443. doi: 10.1007/BF01088806. (Cited on page 158)

[59] P. Palensky and D. Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7(3):381–388, August 2011. ISSN 1551-3203. doi: 10.1109/TII.2011.2158841. (Cited on page 31)

[60] M. Pipattanasomporn, M. Kuzlu, and S. Rahman. An algorithm for intelligent home energy management and demand response analysis. *Smart Grid, IEEE Transactions on*, PP(99):1, 2012. ISSN 1949-3053. doi: 10.1109/TSG.2012.2201182. (Cited on page 50)

[61] B. Ramanathan and V. Vittal. A framework for evaluation of advanced direct load control with minimum disruption. *IEEE Transactions on Power Systems*, 23(4):1681–1688, November 2008. ISSN 0885-8950. doi: 10.1109/TPWRS.2008.2004732. (Cited on pages 47, 162, and 178)

[62] S. Shao, M. Pipattanasomporn, and S. Rahman. Demand response as a load shaping tool in an intelligent grid with electric vehicles. *Smart Grid, IEEE Transactions on*, 2(4):624–631, dec. 2011. ISSN 1949-3053. doi: 10.1109/TSG.2011.2164583. (Cited on page 50)

[63] S. Shao, M. Pipattanasomporn, and S. Rahman. Development of physical-based demand response-enabled residential load models. *Power Systems, IEEE Transactions on*, PP(99):1, 2012. ISSN 0885-8950. doi: 10.1109/TPWRS.2012.2208232. (Cited on pages 50 and 164)

[64] S. Shao, M. Pipattanasomporn, and S. Rahman. Grid integration of electric vehicles and demand response with customer choice. *IEEE Transactions on Smart Grid*, 3(1):543–550, 2012. (Cited on page 50)

[65] J. A. Stankovic and K. Ramamritham. *Tutorial: hard real-time systems*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1989. ISBN 0-8186-0819-6. (Cited on page 52)

[66] J. A. Stankovic, I. Lee, A. Mok, and R. Rajkumar. Opportunities and obligations for physical computing systems. *Computer*, 38(11):23–31, 2005. (Cited on page 25)

[67] G. A. Subramanian, A. Dominguez-Garcia, D. Callaway, K. Poolla, and P. Varaiya. Real-time scheduling of deferrable electric loads. In *Proceedings of the 2012 American Control Conference*, June 2012. (Cited on page 49)

[68] J. R. Taylor. *An Introduction to Error Analysis, The Study of Uncertainties in Physical Measurements. Second Edition*. University Science Books, 1997. (Cited on page 93)

[69] US Dept. of Energy. Benefits of demand response in electricity markets and recommendations for achieving them, Feb 2006. Report to the United States Congress. (Cited on page 31)

[70] A. Vojdani. Smart integration. *IEEE Power and Energy Magazine*, 6 (6):71–79, 2008. (Cited on pages 46 and 47)

[71] D.-C. Wei and N. Chen. Air conditioner direct load control by multi-pass dynamic programming. *IEEE Transactions on Power Systems*, 10(1):307–313, 1995. (Cited on page 47)