# Resource-Aware Control

Luca Greco

HABILITATION À DIRIGER DES RECHERCHES

université PARIS-SACLAY

# Resource-Aware Control

*Commande consciente des ressources*

**Habilitation à diriger des recherches de l'Université Paris-Saclay**

**présentée et soutenue à Paris-Saclay, le 7 octobre 2024, par**

**Luca GRECO**

## Composition du jury

| | |
|---|---|
| **Elena PANTELEY**<br>Directrice de Recherches CNRS, L2S, Université Paris-Saclay | Rapporteuse |
| **Maria Domenica DI BENEDETTO**<br>Professeure, DEIE, DEWS, Université de l'Aquila | Rapporteuse |
| **Luca ZACCARIAN**<br>Directeur de Recherches CNRS, LAAS-CNRS, Professeur Université de Trento | Rapporteur |
| **Mohammed CHADLI**<br>Professeur, Université Paris-Saclay, IBISC, University d'Evry Val d'Essone | Examinateur |
| **Constantin MORARESCU**<br>Professeur, CRAN, Université de Lorraine | Examinateur |

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

**Titre:** Commande consciente des ressources

**Résumé:** Les dernières décennies ont été marquées par une croissance spectaculaire de l'intérêt pour les systèmes complexes distribués et interconnectés et pour leur contrôle. Le dénominateur commun de tous les systèmes susmentionnés est l'interaction entre des objets physiques et des entités numériques abstraites que nous concevons pour détecter, surveiller et commander les premiers. Ces systèmes s'accompagnent d'un niveau important de complexité et d'hétérogénéité qui peut être très difficile à gérer. Pour faciliter cette tâche ardue, il est utile d'adopter un point de vue de haut niveau fondé sur la distinction entre la plateforme et les applications de commande. La plate-forme est une couche d'abstraction qui fournit aux applications de commande des ressources telles que des canaux de communication et des capacités de calcul et qui impose des contraintes réalistes. Les services garantis par la plateforme peuvent être caractérisés en termes de qualité de service (QoS) quantifiable (par exemple, largeur de bande, délais, garanties d'ordonnancement, contraintes d'équité). D'autre part, les applications de commande exploitent les ressources fournies par la plateforme pour accomplir leurs tâches avec une qualité de commande (QoC) souhaitable (par exemple, stabilité, robustesse, performance). La commande consciente des ressources consiste à étudier cette relation profonde entre la qualité de service et la qualité de commande, entre la plateforme et les applications de commande. L'objectif est d'étudier l'impact des limitations de la plateforme sur la QoC et, inversement, la manière dont les stratégies de commande peuvent être conçues pour atteindre la QoC souhaitée si elles sont conscientes de la QoS de la plateforme.

La qualité de service reçue par les applications de contrôle peut varier considérablement dans le temps (même avec une échelle de temps courte). Afin de faire face à cette variabilité inhérente, nous poursuivons deux voies de recherche : la commande flexible et la plateforme flexible. La première vise à développer des lois de commande dont la structure ou les paramètres caractéristiques peuvent changer dynamiquement pour s'adapter aux variations de la qualité de service, en garantissant toujours la QoC souhaitée. La seconde vise à développer des politiques flexibles pour gérer les paramètres de la plateforme (par exemple, le temps de calcul alloué aux tâches de commande par l'ordonnanceur du système d'exploitation, les créneaux temporels dans l'accès à un canal de communication) afin de garantir une qualité de commande souhaitée.

Enfin, nous illustrerons quelques perspectives de recherche futures. La première est liée à la commande consciente des ressources. Les deux autres concernent de nouveaux sujets de recherche dans le domaine des soins de santé et des neurosciences. Dans le domaine des soins de santé, nous abordons le problème de la garantie de la synchronisation entre un patient et un ventilateur mécanique. Dans le domaine des neurosciences, nous cherchons à comprendre les mécanismes de base d'un phénomène intriguant appelé « criticalité auto-organisée » (Self-Organised Criticality, SOC).

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

**Title:** Resource-Aware Control

**Abstract:** The past decades have witnessed a dramatic growth in interest for complex distributed interconnected systems and their control. The common denominator of all aforementioned systems is the interaction between physical objects and abstract digital entities that we devise in order to sense, monitor and control the former ones. These systems come with a significant level of complexity and heterogeneity that can be very hard to deal with. To aid in this daunting task, it is beneficial to adopt a high level point of view based on the distinction between platform and control applications. The platform is an abstraction layer providing control applications with resources such as communication channels and computation capabilities and imposing realistic constraints. The services guaranteed by the platform can be characterised in terms of a quantifiable Quality of Service (QoS) (e.g. bandwidth, delays, scheduling guarantees, fairness constraints). On the other hand, control applications exploit the resources provided by the platform to accomplish their tasks with a desirable Quality of Control (QoC) (e.g. stability, robustness, performance). Resource-aware control is all about the study of this deep relation between QoS and QoC, between platform and control applications. The goal is to investigate how platform limitations impact the QoC and conversely how control strategies can be designed to achieve some desired QoC if they are aware of the platform's QoS.

The QoS received by the control applications can widely change in time (even with a short time scale). In order to cope with this inherent variability, we pursue two dual research paths: flexible control and flexible platform. The former aims at developing control laws whose structure or characteristic parameters can dynamically change to adapt to the varying QoS, always guaranteeing a desired QoC. The latter aims at developing flexible policies to manage platform parameters (e.g. computation time allotted to control tasks by the operating system scheduler, time slots in the access to a communication channel) to ensure a desired QoC.

Finally, we will illustrate some future research perspectives. The first is related to resource-aware control. The other two concern new research topics in the field of health care and neuroscience. In health care we address the problem of guaranteeing the synchronisation between a patient and a mechanical ventilator. In neuroscience we aim to understand the basic mechanisms behind an intriguing phenomenon called Self-Organised Criticality (SOC).

Luca Greco

Thèse d'Habilitation à Diriger des Recherches

# *Resource–Aware Control*

12th July 2024

# CONTENTS

# 1

# INTRODUCTION

## 1.1 QUALITY OF SERVICE AND QUALITY OF CONTROL

The past decades have witnessed a dramatic growth in interest for complex distributed interconnected systems and their control. The pervasive penetration of control, computation and communication capabilities in human artefacts of any scale is inducing a deep architectural change: from closed, isolated devices performing individual tasks, we are migrating toward highly interconnected systems comprised of a large number of heterogeneous devices cooperating for achieving common goals. Modern automobiles are small-scale systems hosting from 20 to 80 communicating microprocessors to support a large variety of functionalities from critical engine control tasks to safety and comfort duties [57]. Automated factories are an excellent example of mid-scale plants adopting a distributed networked architecture accounting for sensors, actuators, robotic arms, process machines, etc. [155, 222]. As concerns (very) large-scale plants, electric power systems represent one of the largest and most complex engineering systems created by humans. A huge number of widely different users are dynamically connected to the power grid along with large power generators and an increasing number of medium and small renewable source generators.

The common denominator of all aforementioned systems is the interaction between physical objects and abstract digital entities that we devise in order to sense, monitor and control the former ones. Thus the name Cyber Physical Systems (CPS). The distributed nature of many CPS points to another relevant and unavoidable aspect: the network. A wide gamut of devices and physical systems are connected and exchange information through a shared, digital, finite capacity channel, hence the name Networked Control Systems (NCS).

NCS come with a significant level of complexity and heterogeneity that can be

very hard to deal with. To aid in this daunting task, it is beneficial to adopt a high level point of view based on the *distinction between platform and control applications*. The platform is an abstraction layer providing control applications with resources such as communication channels and computation capabilities and imposing realistic constraints. The services guaranteed by the platform can be characterised in terms of a quantifiable Quality of Service (QoS) (e.g. bandwidth, delays, scheduling guarantees, fairness constraints). On the other hand, control applications exploit the resources provided by the platform to accomplish their tasks with a desirable Quality of Control (QoC) (e.g. stability, robustness, performance). *Resource-aware control* is all about the study of this deep relation between QoS and QoC, between platform and control applications. The goal is to investigate how platform limitations impact the QoC and conversely how control strategies can be designed to achieve some desired QoC if they are aware of the platform's QoS.

Promptly reacting to asynchronous events (e.g. unexpected obstacles for autonomous guided vehicles, abrupt variations in the environmental conditions) induces fluctuations both in the time required to process sensor data and in the bandwidth required to communicate the results of such computations among the subsystems composing the networked system. Besides, on a shared computation platform, different tasks compete for the same computation resources. Some of these tasks may experience unpredictable fluctuations in their execution time due to the data they have to process, the specific algorithm they have to employ or the interference they suffer from other concurrent tasks. As a consequence, the QoS received by the control applications can widely change in time (even with a short time scale). In order to cope with this inherent variability, we pursue two dual research paths: flexible control and flexible platform. The former aims at developing control laws whose structure or characteristic parameters can dynamically change to adapt to the varying QoS, always guaranteeing a desired QoC. The latter aims at developing flexible policies to manage platform parameters (e.g. computation time allotted to control tasks by the operating system scheduler, time slots in the access to a communication channel) to ensure a desired QoC. Both approaches can be applied to communication and computation problems, however in this thesis we focus mainly on the constraints engendered by the computation platform and we propose a set of methods which explicitly take into account the availability of computation resources.

## 1.2 COMPUTATION–AWARE CONTROL METHODS

Given a plant and some control goals, it is essential to establish if the computational resources are able to provide the desired services. If some classical requirements such as stability, optimal indices or $H_2/H_\infty$ minimisation are chosen to measure the QoC, the designer has to check if the chosen computation platform can ensure the desired QoC. If, for instance, the embedded platform is too simple to provide the necessary computation power, or cannot satisfy the time sampling requirements, a different solution accounting for the employment of some remote computation platforms has

to be sought for. On the other hand, if the embedded platform is powerful enough to be endowed with a Real Time Operating System (RTOS), economical reasons usually push towards an intense sharing of its resources among different tasks. As an example, the BMW research division [92] estimates that in a modern car more than 800 control functions share 70 Embedded Computation Unit (ECU), which corresponds to an average allocation of more than 10 concurrent applications per ECU. New standards (such as AUTOSAR) have been defined to guarantee a safe coexistence of several concurrent applications at the software level, but the implications of this coexistence on control performance are not yet understood in depth.

The price to pay for resource sharing is a reduced predictability of the timing behaviour: an application receives a different availability of resources and suffers time-varying delays depending on the interference from other applications. The ever increasing use of sophisticated sensing devices such as cameras, LIDARS, LASER scanners in such systems as autonomous robots and self-driving cars, adds complexity to the picture. The time needed to extract the information from such sensors can be conspicuous and it fluctuates significantly depending on the input data set. A standard way to keep in check the delays suffered by a control task is by the combination of the time–triggered approach [121] with the real–time scheduling theory [139]. The former ensures constant delays by forcing the input-output operations of a task to take place at precise points in time (e.g., the sampling period expiration), even if the result is produced earlier. The latter offers conditions to guarantee that all tasks will deliver their results before the planned instants (deadlines), given the scheduling parameters and the Worst Case Execution Time (WCET) of the tasks.

Therefore, other than coping with varying delays, resource sharing raises the problem of guaranteeing schedulability of the different tasks (avoiding deadline misses). This problem has been tackled in the literature [186, 163, 211, 212] resorting to real-time preemptive algorithms, such as e.g. Rate Monotonic and Earliest Deadline First [139, 142, 44], that can suspend the execution of a task in the presence of requests by other, higher priority tasks. Unfortunately, these results are hardly applicable to the important class of application mentioned above, where the probability distributions of the computation time have long tails. Indeed, if classical WCET estimates are used, the QoC that can be ensured is usually very poor as minimal controllers only can be implemented to ensure the schedulability. The evident drawback is that the control designer cannot capitalise on the additional availability of resources when the system workload is low. In modern control systems strong fluctuations in the workload are not infrequent. As a specific example, suppose, in an automotive application, that an ECU is used both for spark ignition control and for other control applications. The spark ignition task is activated by an ''angular'' event (i.e., the piston reaching the dead-end). Therefore, it generates a workload increasing with the rotation per minute of the engine. Designing the remaining applications for the worst case, is in this case tantamount to assuming that the engine always rotates at the maximum speed, which is clearly an infrequent occurrence. Even more evident are the workload fluctuations generated by control tasks using multimedia data (e.g., visual servoing).

While empirical approaches can be used to fine-tune the application operating with the prototypical implementations, the result rarely meets the robustness and portability requirements posed by modern industry. Indeed, the empirical fine-tuning is made only with a specific architecture and it is not guaranteed that the specific workload conditions considered in the prototype tests will be the ones actually encountered in the production system. This problem calls for non conventional approaches in which stochastic fluctuations in the delays are accepted and harnessed [91].

Some authors have investigated on how to make the control design robust against an irregular timing behaviour of the implementation, focusing on such effects as packet dropout [137], jitter in computation [147] and time varying delays [116]. More recently, the onset of a new class of control algorithms, event-triggered [205] self-triggered [7, 202] dismisses the very idea of periodic sampling and advocates a different idea: the execution of the control action should take place only when necessary. Significant recent work has been made to close the gap between event-triggered control and a periodic implementation [37], pushing the applicability boundary of this idea to the domain of non-linear systems.

In this thesis, we advocate a different theoretical approach to this problem based on the design of flexible controllers and on a flexible management of platform parameters.

### 1.2.1 *Anytime Control algorithms (flexible control)*

As mentioned before, when the QoS offered by the platform is not constant but can significantly vary over time, ensuring a guaranteed QoC is a relevant problem. Classical methods based on worst-case estimates, while simple to implement, are clearly not ideal in terms of performance. A way to overcome these limitations is to render the control structure or the control algorithm flexible, capable to adapt to the ever-changing QoS. The Anytime Control paradigm represents an attempt to provide a rigorous answer to this need.

According to the anytime approach, controllers must be designed such that the control algorithm is always able to provide a minimal QoC when run for a minimal guaranteed time (as in the worst-case design). However, the more the control algorithm is allowed to execute, the better the QoC it provides. The idea of an algorithm refining the quality of its results according to the availability of computation time is borrowed by the field of imprecise computation [143, 144]. The characteristic of anytime algorithms is to always give an answer on demand; however, the longer they are allowed to compute, the more accurate the answer they return. Exporting the general anytime philosophy to control algorithms is not a trivial task as done, for instance, in digital filter design [166]. There, a full-order filter is decomposed in a series of lower order filters whose execution is prioritised. While very attractive, such an approach cannot be directly applied to the control setting. The main reason is that controllers are in closed loop and their performance does not increase, in general, with the quality of the functional approximation of the control law. Even worst, the simple stability ensured at a certain level of approximation can be lost with a more precise approximation! The functional approximation approach was pursued in [29, 30]. Apart for the aforementioned prob-

lems, the controllers presented in these works suffer of an intrinsic limitation which make them impractical: the implementation of their simplest controller is almost always more computational demanding than the full-order controller implemented with a classical design.

We present, in this thesis, our take on the anytime control philosophy as formalised in [99, 83, 100, 170, 101, 171]. We opt for a bottom-up, compositional approach to the construction of an hierarchy of control tasks of increasing complexity and, accordingly, of increasing performance. The simplest control task in the hierarchy could be designed to guarantee only stability, while whenever the scheduler provides additional time, other more sophisticated control algorithms could be executed to obtain better QoCs. The ensuing closed loop system switches among different dynamics (one for each controller in the hierarchy), thus inducing non trivial stability problems. We provide a stochastic description of the time allotted to the execution of the task computing the anytime control and, accordingly, we give stochastic guarantees on the stability of the switched closed loop system.

### 1.2.2 *Optimal bandwidth allocation (flexible platform)*

When the computation platform (ECU or CPU) is complex enough to be endowed with a modern Real-Time Operating System (RTOS), it is customary to have several different tasks running concurrently and competing for the computational resources. As discussed before, the classical approach to real-time scheduling is based on WCET estimates, however, important resource savings can be obtained if the hard real-time constraints are adequately weakened (see [48]). A current trend in embedded system design is to relax hard schedulability constraints and introduce *softer* models of computation (viz. "Resource Reservations" [160], "weakly hard" [25] and "firm" [177] RTOSs). Here, occasional deadline misses are tolerated for tasks whose execution time may largely vary (e.g. due to data-driven branching and/or to the use of caches and pipelines). Instead of using gross WCET bounds, the random distribution pattern and average of such misses can be quantified by suitable QoS metrics, in terms of which design constraints are typically set [140]. Several studies [161, 49] have shown, indeed, that there can be a sensible performance gain in aggressive choices for the scheduling parameters (e.g. the sampling period) even in the face of a moderate number of deadline misses.

The point of view pursued here is dual and complementary to that of the Anytime Control. With the Anytime Control the platform is fixed and the control task is made flexible to adapt to varying QoS, while here the control is fixed and the QoS is manipulated, subject to some constraints, to achieve a desired QoC. We devise different optimisation algorithms whose goal is to strike a good compromise between resource exploitation and closed loop system performance. In particular, we consider different realistic scheduling scenarios and we provide two families of bandwidth allocation policies. The aim of the policies is to find a balanced allocation of computation resources among different concurrent tasks and to ensure guaranteed QoCs. The two computation/scheduling models we propose are based on the Constant Bandwidth

Server (CBS) [2] and, consequently, enjoy the temporal isolation property: the ability for a task to respect a temporal constraint is independent of the behaviour of the other tasks in the system.

The first model (proposed in [87, 89]) is simpler and allows to assign a bandwidth to each task according to their needs and a *control-oriented* QoC metric. Each task is soft real-time as it could occasionally miss its deadline. The goal is to maximise the overall QoC of all the closed loop system while allowing as much task as possible to execute.

The second model (presented in [84, 86, 85]) is more flexible as it allows the tasks to terminate after their deadlines (with a bounded delay) and it offers the designer a finer granularity in the allocation of the bandwidth by exposing a parameter known as server period. The cost function is, in this case, more *platform-oriented* as it aims at maximising the number of tasks concurrently executing. The QoC is clearly present, in the form of a stochastic stability, but is limited to the constraints. The goal here is almost reversed, that is to maximise the number of tasks while ensuring a certain QoC. The allocation policies developed with the second model belong to the class of resource reservations ([176, 2]). We call them *Adaptive Reservations* as they are able to manipulate (like in a feedback loop) the scheduling parameters (our virtual actuators) to minimise the scheduling error, which accounts for the delay introduced by the task computation.

Very interesting is the combination of the two models in a framework called *continuous stream* (see [88]), which further increases the level of flexibility offered to the designer by almost removing the periodicity in the tasks' activation.

### 1.2.3 ω-regular constraints (flexible platform)

In many computation and communication platforms the QoS cannot be simply described in terms of minimal and maximal delays, dwell times, number of packet drops in a given period, etc. On the other hand, a stochastic description is not always well adapted or desirable. Consider, for instance, the case where the constraints imposed by (or, dually, the requirements demanded to) a platform are of the fairness type, i.e. concurrent users must be granted access to a shared resource infinitely often. It is not uncommon to have different controllers for the same plant, each designed to ensure different performance. The scheduling algorithm can then be asked to ensure that each control in the set execute an infinite number of times. Similarly, different agents may be in need to cyclically access to the same communication channel. In both cases the overall system is switching among different dynamics (modes). A way to model the QoS in such platforms is via a special class of constrained switching signals called *shuffled* ([103, 206, 95]). A switching signal is said to be shuffled if all the modes of the switched systems are activated infinitely often. The set of shuffled switching signals constitutes an example of an $\omega$-regular language, which, in turn, can always be characterised using Büchi automata [14].

A practical application of systems driven by $\omega$-regular languages is consensus/synchronisation over time-varying (undirected) graphs, which can be seen

as a switched system. It is well known (see e.g. [152, 33]) that consensus/synchronisation can be reached if and only if, at every time instant, the union of future interaction graphs is connected. This connectivity condition cannot be described using labeled graphs as in [169], but can be specified using deterministic Büchi automata.

In general, $\omega$-regular languages are well suited to characterise a wide range of properties or constraints through, for instance, linear temporal logic specifications [14]. In this vein, an interesting example is represented by tasks whose execution is subject to precedence constraints. In NCSs a task can be responsible for retrieving measurements from sensors and another for sending control values to actuators. The access to the network by these two tasks can be managed asymmetrically: the measurement task can access the network more frequently as the fresher the data the better; on the other hand, there is no gain for the control task to access the network if it has not received a new measurement. The measurement task must have executed at least once between two consecutive executions of the control task.

A fundamental step in the direction of achieving a flexible management of platform parameters is to characterise what is the minimal QoS necessary to ensure a desired QoC. In the present line of research we have chosen as (minimal) QoC the stability of the switched closed loop system. We provide some sufficient (in a special case also necessary) results for the stability of switched linear systems driven by an $\omega$-regular switching signal. Moreover, we exploit such results to tackle an interesting applicative problem: designing a switched observer to reconstruct the continuous state of the system. To this aim, we devise a procedure to build a Büchi automaton generating the required $\omega$-regular language. The automaton is then used to drive the discrete state of the switched observer. Two methods for computing off-line the gains of the observer are also proposed.

## 1.3 OUTLINE OF THE THESIS

In this section we provide a brief outline of the thesis. A detailed analysis of the individual contributions is deferred to the section Conclusions of each chapter.

Chapter 2 introduces models, definitions and tools that prove useful in Chapter 3 and Chapter 5. In particular, we present a model of Stochastic Jump Linear System (SJLS) encompassing stochastic switched systems driven by Markov Chains and independent and identically distributed processes. We also introduce definitions and Lyapunov tools to ascertain the robust stability of the SJLS in the spirit of classic ISS formulation, but in a stochastic context.

Chapter 3 is devoted to the presentation of our approach to the Anytime Control philosophy. We illustrate the principles of a compositional design technique guiding the construction of an hierarchy of controllers with increasing complexity and performance. We describe how to model a scheduler in terms of a SJLS and we give a rigorous method to design a stochastic switching law that guarantees the stability of the closed loop system. Robustness issues are considered both in the case of imperfect knowledge of the scheduler parameters and in the presence of exogenous disturbances.

Chapters 4 and 5 together address the problem of how to optimally allocate the computation bandwidth to different control tasks. They both assume the adoption of a scheduling algorithm known as the Constant Bandwidth Server (CBS), which greatly simplifies the description of the temporal behaviour of concurrent tasks. In Chapter 4 a simpler model of computation platform and QoS is presented, where a control task unable to complete its execution within its deadline is simply dropped. In Chapter 5 a much more refined model is presented based on the Resource Reservation policy. As a result the designer is endowed with a more control and a finer granularity in the choice of the design parameters. Both computation models are amenable of a stochastic description upon which an optimisation problem can be formulated to allocate the scheduler bandwidth. Different QoC measures are considered: a more control-oriented in Chapter 4 and a more platform-oriented in Chapter 5. In Chapter 5 we also present an interesting development of these models called continuous stream which provides even more degrees of freedom to the designer.

Chapter 6 presents a different point of view to the idea of flexible platform. It discusses the cases where the QoS, or the platform constraints, are better described in terms of $\omega$-regular languages (or equivalently by Büchi automata). The considered switched systems are now discrete and deterministic. We provide necessary and sufficient stability results based on the construction of a suitable Büchi automaton and associated Lyapunov function. We extend these results to the design of a switched asymptotic observer.

Finally, Chapter 7 illustrates the future research perspectives. They follow three main directions. The first is in the track of the topic this thesis is devoted to, namely resource-aware control. The other two directions account for new research topics to which we are presently focusing the most of our efforts.

One is in the field of neurosciences and aims to understand the basic mechanisms behind an intriguing phenomenon called Self-Organised Criticality (SOC). In Chapter 7 the SOC phenomenon is thoroughly described and put into context wrt the brain criticality hypothesis and the recent literature.

The other direction concerns mechanical ventilation. A fundamental problem in this domain is to avoid deleterious asynchronies between the patient and the ventilator. The main goal, in this line of research, is to develop an algorithm ensuring the correct patient/ventilator synchronisation for the most used ventilation mode.

# 2

# SOME ROBUST STABILITY RESULTS FOR STOCHASTIC DISCRETE-TIME SWITCHED LINEAR SYSTEMS

## 2.1 INTRODUCTION

A current trend in embedded system design is to relax hard schedulability constraints and introduce "softer" models of computation. The aim is to overcome the limitations induced by deterministic, WCET-based models in favour of more flexible *stochastic* models, more keen on capturing the intrinsic fluctuations of the available resources (se for instance [146, 26, 27]). Many authors have embraced models of Real-Time systems entailing Markov chains (MC) (cf. e.g. [71, 140, 117]) for their ability to accurately describe the response time distribution and deadline miss probabilities of the concurrent tasks. This direction is also followed by Lemmon and co-workers in a series of papers [138, 140, 51] where they consider performance of NCSs and embedded systems in a Firm Real-Time systems framework ([177, 25]) and introduce a stochastic model to describe the task dropout process based on MCs.

All the previous models belong to the class of Stochastic Jump Systems (SJSs), namely switching systems whose commutations are ruled by a stochastic process. Commonly adopted stochastic processes are MCs and independent and identically distributed (i.i.d.) processes. Recently the need for a more general model has also emerged. In many practical cases (cf. e.g. [157, 101, 171, 84, 86, 172]), the evolution of a system involving shared resources cannot be adequately described in terms of a unique stochastic source. For instance, let us consider a system where different dynamics switch according to which (control) task is executed on a computational platform. The commutations among the different dynamics are induced by two distinct stochastic

sources: the available computation time and the execution time of each task. Whence the need for a model accounting for the two stochastic sources.

As customary in control domain, finding conditions for ensuring the stability is the first concern for SJSs. There exists a wide gamut of stability definitions for stochastic systems (see for instance [124, 128, 125, 54]), whose precise comparison is usually a problematic task. Since our interest is mainly devoted to Stochastic Jump Linear Systems (SJLSs) we will refer to the widely adopted stability definitions in [76, 80, 81, 79, 154, 82, 78]. However, these definitions do not explicitly take into account a prominent problem in control: ensuring robustness of the stability property with respect to perturbations or model inaccuracies. Input-to-state stability (ISS [188]) is a powerful tool to that aim. Traditional extensions of the deterministic ISS approach to stochastic systems mainly focus on continuous-time systems with a stochastic input (see for instance [197, 198, 189, 218]). Recently an important first step towards stochastic ISS for continuous-time SJS with stochastic commutations has appeared in [53], even if the general nonlinear framework there considered prevents from obtaining tight results for linear systems. A related robustness problem is addressed in [192], where a state feedback control is designed to ensure the almost sure stability of a continuous-time Markov Jump Linear System (MJLS) affected by structured uncertainties.

The present chapter serves as an introduction to models, definitions and tools that will prove useful in Chapter 3 and Chapter 5. However, many of the results presented here deserve their own attention. More precisely, the goal of this chapter is twofold.

First, we provide a formal description of a model of SJLS encompassing as special cases switching systems whose commutations are governed by MCs and i.i.d. processes. We enlighten the tight relation bonding the stochastic process adopted in the presented SJLS and the well known Hidden Markov Models (HMMs). We briefly present some concrete cases, where such model turns out essential to catch the complex and varying behaviour of shared resource availability.

Second, we make a connection between deterministic ISS results and the stochastic stability definitions in [76, 80, 81, 79, 154, 82, 78]. In particular we provide stochastic ISS definitions and easily testable conditions for the aforementioned more general model of discrete-time SJLSs. These definitions and conditions enjoy the nice property to coincide with the classical definitions and conditions in [76, 80, 81, 79, 154, 82, 78], when they are specialised to SJLSs governed by MCs or i.i.d. processes in the absence of disturbances.

We stress that, in this chapter, we focus on discrete-time SJLSs with deterministic disturbances and stochastic commutations, while some classical MJLS models (see for instance [59, 125, 197, 198, 218]) consider both stochastic inputs and a stochastic source of commutations.

NOTATION: The set $\mathcal{S}^{N-1} \triangleq \left\{ s = (s_1, \dots, s_N) \in [0,1]^N \mid \sum_{i=1}^N s_i = 1 \right\}$ represents the $(N-1)$-dimensional canonical stochastic simplex. The symbol $\otimes$ denotes the Kronecker product. A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of class $\mathcal{K}$ if it is continuous, zero at zero and strictly increasing. It is said of class $\mathcal{K}_\infty$ if it is also unbounded. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to be of class $\mathcal{KL}$ if, given any $t \geq 0$, $\beta(\cdot, t) \in \mathcal{K}$ and,

given any $s \geq 0$, $\beta(s,\cdot)$ is continuous, non–increasing and tends to zero as its argument tends to infinity. Given a symmetric positive definite matrix $M \in \mathbb{R}^{n\times n}$ we denote with $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$ the smallest and the largest eigenvalue of $M$ respectively. Throughout the chapter, $\|\cdot\|$ denotes both the Euclidean norm and the induced matrix norm: for $x \in \mathbb{R}^n$, $\|x\| = \sqrt{x_1^2 + \ldots + x_n^2}$ and, for $A \in \mathbb{R}^{n\times m}$, $\|A\| = \max_{\|x\|=1} \|Ax\|$. Given a locally bounded signal $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^n$, we define $\|d\|_\infty := \sup_{k\in\mathbb{Z}_{\geq 0}} \|d(k)\|$.

## 2.2 BASIC DEFINITIONS AND MOTIVATIONS

### 2.2.1 *System under study*

In this note we focus on the following stochastic jump linear system (SJLS)

$$x(k+1) = A_{\varphi(k)}x(k) + B(x(k),k)d(k), \tag{2.2.1}$$

where $k \in \mathbb{Z}_{\geq 0}$, $x(0) = x_0 \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n\times n}$ for each $i \in J_\varphi \triangleq \{1,\ldots,N\}$, $N \in \mathbb{Z}_{>0}$ and $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^p$ is a deterministic, unknown but locally bounded disturbance. We assume that $B(x(k),k) \in \mathbb{R}^{n\times p}$ is a time-varying vector field for which[1] there exists $\bar{b} > 0$ such that

$$\|B(x(k),k)\| \leq \bar{b}, \qquad \forall k \in \mathbb{Z}_{\geq 0},\ \forall x \in \mathbb{R}^n. \tag{2.2.2}$$

The stochastic process $\{\varphi(\cdot)\}$, governing the index switching, takes values in the set $J_\varphi$ and is defined on the probability space $(\Omega_\varphi, \mathcal{F}_\varphi, \mathsf{P}_\varphi)$ where $\Omega_\varphi$ is the space of elementary events, $\mathcal{F}_\varphi$ is the associated sigma-algebra and $\mathsf{P}_\varphi$ is the probability measure. In order to fully describe the process $\{\varphi(\cdot)\}$ and motivated by several applications that we describe below (see Section 2.2.2), let us introduce a discrete-time finite-state time-homogeneous[2] Markov Chain (FSH-MC) $\{\sigma(\cdot)\}$ taking values in the set $J_\sigma \triangleq \{1,\ldots,M\}$, and defined on the probability space $(\Omega_\sigma, \mathcal{F}_\sigma, \mathsf{P}_\sigma)$. Due to the finite cardinality of the sets $J_\sigma$ and $J_\varphi$, the discrete probability distributions associated to the processes $\{\varphi(\cdot)\}$ and $\{\sigma(\cdot)\}$ can be compactly represented at each step $k \in \mathbb{Z}_{\geq 0}$ by the row vectors $\pi_\sigma(k) \in \mathcal{S}^{M-1}$ and $\pi_\varphi(k) \in \mathcal{S}^{N-1}$. The joint process $\{\sigma(\cdot),\varphi(\cdot)\}$, taking values in the set $J_\sigma \times J_\varphi$, is defined on the probability space $(\Omega, \mathcal{F}, \mathsf{P})$, where $\Omega \triangleq \Omega_\sigma \times \Omega_\varphi$, $\mathcal{F}$ is the sigma-algebra generated by the measurable rectangles $H_\sigma \times H_\varphi$ with $H_\sigma \in \mathcal{F}_\sigma$ and $H_\varphi \in \mathcal{F}_\varphi$ and $\mathsf{P}$ is a probability measure guaranteeing $\mathsf{P}\{H_\sigma \times H_\varphi\} = \mathsf{P}_\sigma\{H_\sigma\}\mathsf{P}_\varphi\{H_\varphi\}$ (see [31] page 231). Let us also denote with $\mathsf{E}\{\cdot\}$ the associated expectation operator.

By the Markov property we have that $\mathsf{P}\{\sigma(k+1) \mid \sigma(k),\ldots,\sigma(0)\} = \mathsf{P}\{\sigma(k+1) \mid \sigma(k)\}$, hence a complete description of the FSH-MC $\{\sigma(\cdot)\}$ is given in terms of its transition probability matrix $P = (p_{li})_{l,i=1,\ldots,M}$, with $p_{li} \triangleq \mathsf{P}\{\sigma(k+1) = i \mid \sigma(k) = l\}$, and its initial probability distribution $\pi_\sigma(0) = \pi_{\sigma 0}$. We consider a process $\{\varphi(\cdot)\}$

[1] An interesting case is obtained when the input matrix switches in the same way as the $A$ matrix, namely for $B(x(k),k) = B_{\varphi(k)}$, in which case $\bar{b} = \max_{i\in J_\varphi} \|B_i\|$.

[2] Time-homogeneous Markov Chains (sometimes referred to as stationary Markov Chains) are characterised by transition probabilities which are independent of time.

whose probability distribution $\pi_\varphi \in \mathcal{S}^{N-1}$ is linearly related to the probability distribution $\pi_\sigma \in \mathcal{S}^{M-1}$ of $\{\sigma(\cdot)\}$ by means of a row stochastic matrix $L = (\ell_{lj})_{\substack{l=1,\dots,M \\ j=1,\dots,N}}$, with $\ell_{lj} \triangleq \mathsf{P}\{\varphi(k) = j \mid \sigma(k) = l\}$. Actually, we are assuming that the distribution of the process $\{\varphi(\cdot)\}$ at any step $k \in \mathbb{Z}_{\geq 0}$ depends only on $\sigma(k)$ and not on the previous values of $\{\sigma(\cdot)\}$ ($\sigma(i)$, $i < k$) or $\{\varphi(\cdot)\}$ ($\varphi(i)$, $i < k$). This means that $\mathsf{P}\{\varphi(k) \mid \varphi(k-1), \dots, \varphi(0), \sigma(k), \dots, \sigma(0)\} = \mathsf{P}\{\varphi(k) \mid \sigma(k)\}$. Therefore, the probability distribution of the joint process $\{\sigma(\cdot), \varphi(\cdot)\}$ evolves according to the following dynamics

$$\pi_\sigma(k+1) = \pi_\sigma(k)P \tag{2.2.3}$$

$$\pi_\varphi(k) = \pi_\sigma(k)L. \tag{2.2.4}$$

The equations (2.2.3)-(2.2.4) can be regarded as the description of a discrete-time LTI system without control, where $\pi_\sigma$ is the state vector and $\pi_\varphi$ is the output. The present stochastic model encompasses as special cases both FSH-MCs (if $J_\varphi = J_\sigma$ and $L$ is chosen as the identity matrix) and, in particular, independent and identically distributed (i.i.d.) processes. It is worth noting that the model (2.2.3)-(2.2.4) can be readily interpreted as a Hidden Markov Model (HMM) [74], where the (hidden) state of the MC $\{\sigma(\cdot)\}$ is not directly observable and the observation process $\{\varphi(\cdot)\}$ is used to reconstruct such state. We stress, however, that our interest here is not about the estimation and reconstruction problems, which are already widely studied in literature (see for instance [175, 221, 72]), but rather in robust stability problems arising when the commutations of a switching system are ruled by a HMM.

If the FSH-MC $\{\sigma(\cdot)\}$ is also irreducible and aperiodic (FSHIA-MC) (see [150, Section 8.4] and [31, Section 8]), then there exists a unique invariant probability distribution (i.p.d.) $\overline{\pi}_\sigma$ such that $\lim_{k\to\infty} \pi_\sigma(k) = \overline{\pi}_\sigma$ for any $\pi_\sigma(0)$. Namely, $\{\sigma(\cdot)\}$ is an ergodic process with a unique ergodic class (see [31, Section 24] and [150, Section 8.4]). In this case the process $\{\varphi(\cdot)\}$ inherits the ergodic property of $\{\sigma(\cdot)\}$ and a unique i.p.d. $\overline{\pi}_\varphi \triangleq \overline{\pi}_\sigma L$ exists for that process.

### 2.2.2 *Motivating examples*

The stochastic model (2.2.3)-(2.2.4) describes a wide range of different systems. It is especially relevant in modelling a limited resource shared among competing users. The MC $\{\sigma(\cdot)\}$ can represent the quantised amount of resource, in the finite set $J_\sigma$, available at each instant $k \in \mathbb{Z}_{\geq 0}$, while the process $\{\varphi(\cdot)\}$ can represent the index of the user using the resource at that instant. With such an interpretation $\pi_\varphi(k)$ is the probability of each user to exclusively use the resource at time $k$ and the $(i, j)$-th entry of $L$ represents the probability of the $j$-th user to need an amount of resource equal to the $i$-th value in the set $J_\sigma$. We further illustrate this in practical examples in control with shared resources.

**Example 2.1.** The system presented in Chapter 3 perfectly fits the previous description. In this system, the shared resource is the computation time offered by a multitasking

embedded unit for the execution of a periodic control task. In order to cope with the fluctuations in the available computation time, the control task is designed according to the anytime paradigm. Namely, it is decomposed in a finite set of ordered subroutines $\Gamma_j$, $j \in J_\varphi$, guaranteeing, as $j$ increases, an increasing performance but requiring an increasing computation time. Due to the lack of deterministic information on the computation time availability in each period, the subroutines are forced to execute sequentially, i.e. the computation of $\Gamma_j$ cannot start until the computation of $\Gamma_{j-1}$ has terminated. The computation begins with $\Gamma_1$ and proceeds by progressively adding modules according to the available computation time. In case sufficient computation time is available, the subroutine $\Gamma_j$ is executed, otherwise, in case of premature interruption, the subroutine $\Gamma_{j-1}$ is executed. The ensuing closed-loop system is described by the SJLS (2.2.1), (2.2.3)-(2.2.4). In this setting the FSHIA-MC $\{\sigma(\cdot)\}$ describes the amount of available computation time and the process $\{\varphi(\cdot)\}$ the highest index of a schedulable controller, i.e. the subroutine whose execution time is the largest among those shorter than the available time. The execution time $T^j$ required to compute sequentially all component subroutines from $\Gamma_1$ to $\Gamma_j$ is modelled as a discrete time i.i.d. process taking values in the set $J_\sigma$. The explicit construction of the matrix $L$ in (2.2.4) in terms of the stationary probability distribution of $T^j$ is detailed in Chapter 3.

The model (2.2.3)-(2.2.4) is open to a dual interpretation. Still in the case of a shared resource among different users, the MC $\{\sigma(\cdot)\}$ can represent the index of the user exploiting the resource at any $k \in \mathbb{Z}_{\geq 0}$, while the process $\{\varphi(\cdot)\}$ may represent the amount of resource used at time $k$.

**Example 2.2.** In Section IV of [171] this model is used to describe the overall computation time used by a set of concurrent periodic tasks. For the sake of simplicity, let us focus on a single task only. Such a task is assumed to have a finite number of working modes, which, in this description, play the role of the users. In each period $k$ the task can be in one of $M$ possible modes defined by the set $J_\sigma$. The change in time of the modes is assumed to be ruled by a FSHIA-MC $\{\sigma(\cdot)\}$. According to each mode, the task can require a certain amount of computation time assuming values in the finite set $J_\varphi$ and described by an i.i.d. process. In this case, the $(i, j)$-th entry of $L$ represents the probability of the task to require an amount $j$ of computation time during the mode $i$. In the cited paper this model is used to provide a stochastic description of the computation time left available by higher priority tasks to the execution of a lower priority control task.

A similar model is presented in Section VII of [172], where a FSHIA-MC rules the transition among processor states (or modes) and a stochastic process (analogous to $\{\varphi(\cdot)\}$ here) describes the length of the control sequence computed at each step.

We remark that the model (2.2.3)-(2.2.4) is flexible enough to also encompass embedded control systems affected by variable computation delays and networked systems.

**Example 2.3.** In Chapter 5 a class of scheduling algorithms, known as adaptive reservations, is applied to control problems. Adaptive reservations allow the formulation of a discrete time dynamic model for the evolution of the delay potentially experienced by a control task. If the computation time of the task is an i.i.d. process and the bandwidth allocation policy is time invariant but function of the previously experienced delay, then the delay is described by a FSH-MC $\{\sigma(\cdot)\}$. According to the delay, namely to the state of the MC $\{\sigma(\cdot)\}$, a scheduling policy decides whether executing the delayed control task or dropping the computation. At each time step the system can be either in open loop (drop case) or in closed loop with a specific delay. Then, if a finite discrete set of values is considered for the delay, the switching among the different systems is ruled by a stochastic process $\{\varphi(\cdot)\}$, which encodes the outcome of the scheduling policy.

**Example 2.4.** In [157] a simple model of network delay dynamics is presented. In such a model a three-state MC is used to describe different network load conditions (modes): low, medium and high. Associated to each mode there is an i.i.d. process describing the delay probability for different values of delays. Clearly larger delays are more likely to occur in high than in low load conditions. If we consider a discrete and finite set of possible delays, we can easily fit the model in [157] with the one here presented. It suffices to identify the MC with $\{\sigma(\cdot)\}$ and to stack the three delay distributions (considered as vectors) to build the $L$ matrix. The ensuing process $\{\varphi(\cdot)\}$ will then represent at each step $k$ the transmission delay experienced by a measurement or a control packet.
Many other works modelling the network traffic fit, or can be adapted to fit, the present stochastic framework. Here we concisely refer to some of them. In [181] a HMM is proposed to model the packet loss process in TCP channels. The MC describes the evolution of the network state, while the block/pass event is determined by different binary probability distributions associated with each state. The paper [182] aims at modelling the IP traffic exhibiting long-range dependence. There, a discrete-time batch Markov arrival process is used to jointly characterise the packet arrival process and the packet size distribution. In particular, a discrete-time Markov modulated Poisson process (dMMPP) accounts for the packet arrivals and a probability distribution associated to the MC state of the dMMPP accounts for the packet size. In [180], the authors employ a HMM to describe the joint delay and loss dynamics in periodic UDP traffic. A HMM is used also in [64] to provide a characterisation of Internet traffic at a packet level, namely to jointly describe the Inter Packet Time and the Packet Size processes.

### 2.2.3 *Stability and robustness definitions*

In Definition 2.1 we adapt classical stochastic stability definitions, such as those in [76, 80, 81, 79], to fit the SJLS system (2.2.1) governed by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$. We denote with $x(k, d)$ the stochastic process solution $x(k, x_0, d, \omega)$, $\omega \in \Omega$ of the difference equation (2.2.1) with non random initial condition $x(0) = x_0 \in \mathbb{R}^n$ and input

signal $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^p$. when the case is explicit enough, we might just write $x(k)$ to denote $x(k, d)$.

**Definition 2.1.** Given a constant $\delta \in \mathbb{R}_{>0}$ and the set of admissible initial distributions $\Phi \triangleq \left\{ \left( \pi_{\sigma 0}, \pi_{\varphi 0} \right) \in S^{N+M-2} \mid \pi_{\varphi 0} = \pi_{\sigma 0} L,\ \pi_{\sigma 0} \in \mathcal{S}^{M-1} \right\}$, the system (2.2.1) with $d = 0$ and driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$ with dynamics (2.2.3)-(2.2.4) is said to be:

1. *$\delta$-moment globally asymptotically stable ($\delta$-GAS)* if, for any $x_0 \in \mathbb{R}^n$ and any initial distribution $\left( \pi_{\sigma 0}, \pi_{\varphi 0} \right) \in \Phi$ of $\{\sigma(\cdot), \varphi(\cdot)\}$,
$$\lim_{k \to \infty} \mathsf{E} \left\{ \| x(k, 0) \|^{\delta} \right\} = 0 ;$$

2. *$\delta$-moment globally exponentially stable ($\delta$-GES)* if there exist constants $a, b \in \mathbb{R}_{>0}$ such that for any $x_0 \in \mathbb{R}^n$ and any initial distribution $\left( \pi_{\sigma 0}, \pi_{\varphi 0} \right) \in \Phi$ of $\{\sigma(\cdot), \varphi(\cdot)\}$
$$\mathsf{E} \left\{ \| x(k, 0) \|^{\delta} \right\} \leq a \, \| x_0 \|^{\delta} \, \mathrm{e}^{-bk}, \quad \forall k \in \mathbb{Z}_{\geq 0};$$

3. *almost surely* (with probability one) *globally stable (as-GS)* if for any $x_0 \in \mathbb{R}^n$ and any initial distribution $\left( \pi_{\sigma 0}, \pi_{\varphi 0} \right) \in \Phi$ of $\{\sigma(\cdot), \varphi(\cdot)\}$
$$\mathsf{P} \left\{ \lim_{k \to \infty} \| x(k, 0) \| = 0 \right\} = 1.$$

We stress that, despite their names, $\delta$-GAS and as-GS actually address the *convergence* of solutions of the system (2.2.1) to the origin, rather than *stability* issues. Nonetheless, for stochastic linear systems $\delta$-GAS and $\delta$-GES are equivalent properties (see for instance [154, 82, 78, 81]). Hence the convergence property in Definition 2.1.1 actually implies the stability contained in Definition 2.1.2. In order to analyse the robustness of system (2.2.1), we introduce the following new stability definitions.

**Definition 2.2.** Given a constant $\delta \in \mathbb{R}_{>0}$ and the set $\Phi$ of admissible initial distributions as in Definition 2.1, the system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$ with dynamics (2.2.3)-(2.2.4) is said to be:

1. *$\delta$-moment input-to-state stable ($\delta$-ISS)* if there exist functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_{\infty}$ such that for any $x_0 \in \mathbb{R}^n$, any initial distribution $\left( \pi_{\sigma 0}, \pi_{\varphi 0} \right) \in \Phi$ of $\{\sigma(\cdot), \varphi(\cdot)\}$ and any locally bounded $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^p$
$$\mathsf{E} \left\{ \| x(k, d) \|^{\delta} \right\} \leq \beta( \| x_0 \|, k) + \gamma( \| d \|_{\infty} ), \quad \forall k \in \mathbb{Z}_{\geq 0};$$

2. *$\delta$-moment exponentially input-to-state stable ($\delta$-EISS)* if there exist constants $a, b \in \mathbb{R}_{>0}$ and a function $\gamma \in \mathcal{K}_{\infty}$ such that for any $x_0 \in \mathbb{R}^n$, any initial distribution $\left( \pi_{\sigma 0}, \pi_{\varphi 0} \right) \in \Phi$ of $\{\sigma(\cdot), \varphi(\cdot)\}$ and any locally bounded $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^p$
$$\mathsf{E} \left\{ \| x(k, d) \|^{\delta} \right\} \leq a \, \| x_0 \|^{\delta} \, \mathrm{e}^{-bk} + \gamma( \| d \|_{\infty} ), \quad \forall k \in \mathbb{Z}_{\geq 0};$$

3. *almost surely* (with probability one) *input-to-state stable (as-ISS)* if there exist functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that for any $x_0 \in \mathbb{R}^n$, any initial distribution $\left(\pi_{\sigma 0}, \pi_{\varphi 0}\right) \in \Phi$ of $\{\sigma(\cdot), \varphi(\cdot)\}$ and any locally bounded $d : \mathbb{Z}_{\geq 0} \to \mathbb{R}^p$

$$\mathsf{P}\left\{\|x(k,d)\| \leq \beta(\|x_0\|, k) + \gamma(\|d\|_\infty)\right\} = 1, \quad \forall k \in \mathbb{Z}_{\geq 0}.$$

As pointed out in [82], the Definitions 2.1 are stronger than the weak stability definitions in [124], as they require to be independent of the initial distribution. Definitions 2.2 inherit the same property. It is also worth noting that the above $\delta$-ISS definition is close to the ISS in $L_1$ estimate at switching instants introduced for continuous time switching systems in [53].

The results derived in this chapter aim at providing some robust stability guarantees for the SJLS (2.2.1), (2.2.3)-(2.2.4). They can be used to ensure that the presence of disturbances do not disrupt the stability properties enjoyed by the considered systems. Once established that the system of interest fits the presented model, the tools can be used to verify the robust stability property for a given distribution of the shared resource. It is worth noting that for "given distribution" we mean a distribution whose stochastic characterisation is known. There is no need for the knowledge of any specific realisation (i.e. sampling path) of the stochastic process.

*Remark* 2.1. Due to the special structure of $\Phi$, which imposes a linear relation between $\pi_{\varphi 0}$ and $\pi_{\sigma 0}$, in the following we will slightly abuse the notation and refer to the probability measure $\mathsf{P}_{(\pi_{\sigma 0}, \pi_{\varphi 0})}\{\cdot\}$, induced by any initial distribution $\left(\pi_{\sigma 0}, \pi_{\varphi 0}\right) \in \Phi$, as $\mathsf{P}_{\pi_{\sigma 0}}\{\cdot\}$.

## 2.3 MAIN RESULTS

On the one hand, the systems evoked in Section 2.2.2 demonstrate the need to develop analytical tools guaranteeing stability and robustness of SJLS of the form (2.2.1). On the other hand, Section 2.2.3 underlines the diversity of existing notions of stability for such systems. This section therefore aims at exhibiting the tight links existing between the stability and robustness notions recalled above for the specific class of systems (2.2.1), (2.2.3)-(2.2.4), and at developing practical conditions to establish them.

### 2.3.1 *δ-ISS and 2-ISS*

Asymptotic second moment stability (2-GAS) and exponential second moment stability (2-GES), for systems without input and driven by a Markov chain, have been shown to be equivalent and to imply the almost sure stability (as-GS) ([154, 82, 78]). Similar properties hold for $\delta$-GAS and $\delta$-GES (see for instance [123, 81]). In this section we prove similar equivalences for the more general class of systems (2.2.1), (2.2.3)-(2.2.4).

**Theorem 2.1.** *For the system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4), $\delta$-GAS, $\delta$-GES, $\delta$-ISS and $\delta$-EISS are equivalent.*

We are now ready to provide sufficient conditions for the $\delta$-EISS (hence also for $\delta$-GAS, $\delta$-GES and $\delta$-ISS) of the system (2.2.1). As a first step, we introduce a Lyapunov characterisation for the $\delta$-EISS of system (2.2.1), which is in the same spirit of some classical results such as [127, 28].

**Lemma 2.1.** *The system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4), is $\delta$-EISS if there exist functions $V_i : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, constants $\underline{\alpha}_i, \bar{\alpha}_i, \alpha_i \in \mathbb{R}_{>0}$, $i \in J_\sigma$, and a function $\chi \in \mathcal{K}_\infty$ such that for all $x \in \mathbb{R}^n$, all $d \in \mathbb{R}^p$ and all $k \in \mathbb{Z}_{\geq 0}$,*

**i)** $\underline{\alpha}_i \|x\|^\delta \leq V_i(x) \leq \bar{\alpha}_i \|x\|^\delta$ for all $i \in J_\sigma$;

**ii)** $\mathsf{E}\left\{V_{\sigma(k+1)}\left(A_{\varphi(k)}x + B(x,k)d(k)\right) - V_{\sigma(k)}(x)\right\} \leq -\alpha_i \|x\|^\delta + \chi(\|d(k)\|)$.

The next result provides an alternative Lyapunov condition for $\delta$-EISS. Contrarily to Lemma 2.1, it does not rely on the decrease of the Lyapunov function expectation, but rather involves the transition probabilities $p_{li}$.

**Lemma 2.2.** *The system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4), is $\delta$-EISS if there exist $M$ matrices $R_l = R_l^T > 0$ and $\alpha \in \mathbb{R}_{>0}$ such that for all $x \in \mathbb{R}^n$ and for all $l \in J_\sigma$*

$$\sum_{h=1}^{N}\sum_{j=1}^{M} p_{lj}\ell_{lh}\left(x^T A_h^T R_j A_h x\right)^{\frac{\delta}{2}} - \left(x^T R_l x\right)^{\frac{\delta}{2}} \leq -\alpha \|x\|^\delta, \tag{2.3.1}$$

*where we recall that $p_{lj} \triangleq \mathsf{P}\{\sigma(k+1) = j \mid \sigma(k) = l\}$ and $\ell_{lh} \triangleq \mathsf{P}\{\varphi(k) = h \mid \sigma(k) = l\}$.*

For the special case of 2-EISS ($\delta = 2$), the following LMI condition can be obtained using Lemma 2.2.

**Theorem 2.2.** *The system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4), is 2-EISS if there exist $M$ matrices $R_l = R_l^T > 0$ such that*

$$\sum_{h=1}^{N} \ell_{lh} A_h^T \tilde{R}_l A_h - R_l < 0, \qquad \forall l \in J_\sigma \tag{2.3.2}$$

*with*

$$\tilde{R}_l \triangleq \sum_{j=1}^{M} p_{lj} R_j, \qquad \forall l \in J_\sigma, \tag{2.3.3}$$

*where we recall that $p_{lj} \triangleq \mathsf{P}\{\sigma(k+1) = j \mid \sigma(k) = l\}$ and $\ell_{lh} \triangleq \mathsf{P}\{\varphi(k) = h \mid \sigma(k) = l\}$.*

The proof is straightforward in view of Lemma 2.2. It is worth stressing that, if the matrix $L = (\ell_{lh})_{\substack{l=1,\dots,M \\ h=1,\dots,N}}$ is the identity matrix, then the LMI conditions (2.3.2)-(2.3.3) become the one of the second moment stability [154, 114, 58, 81].

### 2.3.2 Almost Sure ISS

The previous section shows, for the class of systems (2.2.1), the equivalence between $\delta$-stability properties (namely, $\delta$-GAS and $\delta$-GES) and robustness to exogenous inputs ($\delta$-ISS and $\delta$-EISS), and provides natural Lyapunov tools to guarantee them. We now proceed in conducting a similar analysis for as-ISS. As a first step, we prove the following result.

**Theorem 2.3.** *For the system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4), any property among $\delta$-GAS, $\delta$-GES, $\delta$-ISS and $\delta$-EISS implies as-ISS.*

Exploiting the previous theorem and Lemma 2.2 and proceeding as in Theorem 2.1 in [77], we can provide sufficient conditions for the as-ISS of our system. In case $L$ is the identity matrix, such conditions turn to those provided in [77] for as-GS.

**Corollary 2.1.** *The system (2.2.1) driven by the stochastic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4), is as-ISS if there exist $M$ matrices $R_l = R_l^T > 0$ such that one of the following conditions is verified*

- $$\max_{\|x\|=1} \prod_{h=1}^{N} \prod_{j=1}^{M} \left( \frac{x^T A_h^T R_j A_h x}{x^T R_l x} \right)^{p_{lj}\ell_{lh}} < 1, \qquad \forall l \in J_\sigma$$
- $$\prod_{h=1}^{N} \prod_{j=1}^{M} \lambda_{\max} \left( A_h^T R_j A_h R_l^{-1} \right)^{p_{lj}\ell_{lh}} < 1, \qquad \forall l \in J_\sigma.$$

As remarked in Section 2.3, if $\{\sigma(\cdot)\}$ is a FSHIA-MC, thus an ergodic process with a unique ergodic class and a unique i.p.d. $\overline{\pi}_\sigma$, the process $\{\varphi(\cdot)\}$ inherits the same ergodic property and it has a unique i.p.d. $\overline{\pi}_\varphi = \overline{\pi}_\sigma L$. Therefore, for ergodic processes we have the following condition (see also [80, 77]).

**Corollary 2.2.** *The system (2.2.1) driven by the stochastic ergodic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4) and whose i.p.d. is given by $\left(\overline{\pi}_\sigma, \overline{\pi}_\varphi\right)$ with $\overline{\pi}_\varphi \triangleq \left[\overline{\pi}_{\varphi_1}, \ldots, \overline{\pi}_{\varphi_N}\right] = \overline{\pi}_\sigma L$, is as-ISS if*

$$\prod_{j=1}^{N} \left\| A_j \right\|^{\overline{\pi}_{\varphi_j}} < 1.$$

### 2.3.3 Lifted systems

Many of the stability results presented in the previous sections can be straightforwardly extended to lifted systems with the advantage of producing less restrictive conditions. To ease the exposition, we will focus on autonomous systems, namely we drop the presence of the disturbance $d$. We briefly extend here some concepts related to lifted MJLS (see for instance [35]).

We define "lifting of period $m$" associated to the SJLS (2.2.1) (or simply $m$-lifted system) the SJLS

$$\widetilde{x}(h+1) = \widetilde{A}_{\tilde{\varphi}(h)}\widetilde{x}(h) \tag{2.3.4}$$

with $\widetilde{x}(h) = x(mh)$, $\widetilde{\varphi}(h) = [\varphi(mh), \cdots, \varphi(mh+m-1)]$, $\widetilde{A}_{\tilde{\varphi}(h)} = A_{\varphi(mh+m-1)}A_{\varphi(mh+m-2)}\cdots A_{\varphi(mh)}$. Such a system can be interpreted as the sampling of the original one at time instants $hm$, $h \in \mathbb{Z}_{\geq 0}$.

In order to characterise the process $\{\widetilde{\varphi}(\cdot)\}$, which takes values in the set $J_{\widetilde{\varphi}} = J_{\varphi}^{m}$, we define the $m$-lifted FSH-MC $\{\widetilde{\sigma}(\cdot)\}$, taking values in the set $J_{\widetilde{\sigma}} = J_{\sigma}^{m}$, as follows: $\widetilde{\sigma}(h) = [\sigma(mh), \cdots, \sigma(mh+m-1)]$. By using the usual vector notation, we associate with $\{\widetilde{\sigma}(\cdot)\}$ the discrete distribution at time $h$ $\pi_{\tilde{\sigma}}(h) \in \mathcal{S}^{mM-1}$, where $\pi_{\tilde{\sigma}_{\widetilde{\imath}}}(h) = \mathsf{P}\{\widetilde{\sigma}(h) = \widetilde{\imath}\}$ for every $\widetilde{\imath} = (i_1, i_2, \ldots, i_m) \in J_{\widetilde{\sigma}}$. It is immediate to verify that the FSH-MC $\{\widetilde{\sigma}(\cdot)\}$ has transition probability matrix $\widetilde{P}$ given by $\widetilde{p}_{\widetilde{\imath}\widetilde{\jmath}} = p_{i_m j_1}\prod_{l=1}^{m-1} p_{j_l j_{l+1}}$, where $\widetilde{p}_{\widetilde{\imath}\widetilde{\jmath}} = \mathsf{P}\{\widetilde{\sigma}(h+1) = \widetilde{\jmath} \mid \widetilde{\sigma}(h) = \widetilde{\imath}\}$ for $\widetilde{\imath}, \widetilde{\jmath} \in J_{\widetilde{\sigma}}$ and $P = (p_{li})_{l,i=1,\ldots,M}$ is the transition probability matrix of $\{\sigma(\cdot)\}$. If $\{\sigma(\cdot)\}$ is a FSHIA-MC with i.p.d. $\overline{\pi}_{\sigma}$, then $\{\widetilde{\sigma}(\cdot)\}$ also is a FSHIA-MC whose i.p.d. has elements $\overline{\pi}_{\tilde{\sigma}_{\widetilde{\imath}}} = \prod_{l=1}^{m-1} p_{i_l i_{l+1}}\overline{\pi}_{\sigma_{i_1}}$.

We can now associate with $\{\widetilde{\varphi}(\cdot)\}$ its discrete probability distribution at time $h$ $\pi_{\tilde{\varphi}}(h) \in \mathcal{S}^{mN-1}$ as follows:

$$\begin{aligned}\pi_{\tilde{\varphi}_{\widetilde{\jmath}}}(h) &= \mathsf{P}\{\widetilde{\varphi}(h) = \widetilde{\jmath}\} \\ &= \mathsf{P}\{\widetilde{\varphi}(h) = \widetilde{\jmath} \mid \widetilde{\sigma}(h) = \widetilde{\imath}\}\,\mathsf{P}\{\widetilde{\sigma}(h) = \widetilde{\imath}\} \\ &= \widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}}\pi_{\tilde{\sigma}_{\widetilde{\imath}}}(h)\end{aligned}$$

where $\widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}} \triangleq \mathsf{P}\{\widetilde{\varphi}(h) = \widetilde{\jmath} \mid \widetilde{\sigma}(h) = \widetilde{\imath}\}$ for $\widetilde{\imath} = (i_1, i_2, \ldots, i_m) \in J_{\widetilde{\sigma}}$ and $\widetilde{\jmath} = (j_1, j_2, \ldots, j_m) \in J_{\widetilde{\varphi}}$. The terms $\widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}}$ can be expressed as products of the entries of the matrix $L = (\ell_{lj})_{\substack{l=1,\ldots,M \\ j=1,\ldots,N}}$ in (2.2.4):

$$\begin{aligned}\widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}} &= \mathsf{P}\{\varphi(mh+m-1) = j_m, \cdots, \varphi(mh) = j_1 \mid \sigma(mh+m-1) = i_m, \cdots, \sigma(mh) = i_1\} \\ &= \mathsf{P}\{\varphi(mh+m-1) = j_m \mid \varphi(mh+m-2) = j_{m-1}, \cdots, \varphi(mh) = j_1, \\ &\quad \sigma(mh+m-1) = i_m, \cdots, \sigma(mh) = i_1\} \\ &\quad \times \mathsf{P}\{\varphi(mh+m-2) = j_{m-1}, \cdots, \varphi(mh) = j_1 \mid \sigma(mh+m-1) = i_m, \cdots, \sigma(mh) = i_1\} \\ &= \mathsf{P}\{\varphi(mh+m-1) = j_m \mid \sigma(mh+m-1) = i_m\} \\ &\quad \times \mathsf{P}\{\varphi(mh+m-2) = j_{m-1}, \cdots, \varphi(mh) = j_1 \mid \sigma(mh+m-1) = i_m, \cdots, \sigma(mh) = i_1\},\end{aligned}$$

where we used the fact that the information of $\varphi(mh+m-1)$ is entirely encapsulated in $\sigma(mh+m-1)$, thus allowing to express its dependence only in terms of $\sigma(mh+m-1)$ without the need for the previous values of $\{\varphi(\cdot)\}$ and $\{\sigma(\cdot)\}$. By iterating the previous relation, we find $\widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}} = \ell_{i_1 j_1}\cdots\ell_{i_m j_m}$. The previous expression can be compactly written as $\widetilde{L} = L \otimes \cdots \otimes L$ (repeated $m$ times), where $\widetilde{L}$ is the matrix collecting all the terms $\widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}}$ and $\otimes$ denotes the Kronecker product.

Summing up, for the joint lifted process $\{\widetilde{\sigma}(\cdot), \widetilde{\varphi}(\cdot)\}$ we have a stochastic characterisation equal to that in (2.2.3)-(2.2.4) for the non lifted process:

$$\pi_{\tilde{\sigma}}(h+1) = \pi_{\tilde{\sigma}}(h)\widetilde{P} \tag{2.3.5}$$

$$\pi_{\tilde{\varphi}}(h) = \pi_{\tilde{\sigma}}(h)\widetilde{L}. \tag{2.3.6}$$

As an example of stability results given in terms of the lifted system, we provide a condition less restrictive than that in Corollary 2.2. The condition is formally almost identical to that presented in [35].

**Corollary 2.3.** *The system (2.2.1) driven by the stochastic ergodic process $\{\sigma(\cdot), \varphi(\cdot)\}$, whose distributions are described by the evolutions (2.2.3) and (2.2.4) and whose m-lifted system is given by* (2.3.4)-(2.3.5)-(2.3.6) *is as-ISS if*

$$\prod_{\tilde{j}\in J_{\widetilde{\varphi}}} \|\widetilde{A}_{\tilde{j}}\|^{\overline{\pi}_{\tilde{\varphi}_{\tilde{j}}}} < 1,$$

*where $\overline{\pi}_{\tilde{\varphi}} = \overline{\pi}_{\tilde{\sigma}}\widetilde{L}$ and $\overline{\pi}_{\tilde{\sigma}_{\tilde{i}}} = \prod_{l=1}^{m-1} p_{i_l i_{l+1}} \overline{\pi}_{\sigma_{i_1}}$, $\widetilde{\imath} = (i_1, i_2, \ldots, i_m) \in J_{\widetilde{\sigma}}$.*

## 2.4 CONCLUSIONS

This Chapter was devoted to the introduction of some models, definitions and tools useful in the analysis of wide variety of cyber-physical or network controlled systems entailing shared resources. More precisely, the main contributions are:

*A model of SJLS.* We proposed a new model of SJLS encompassing, as special cases, switched systems driven by MCs and i.i.d. processes. Such a model proves very useful in describing the varying behaviour of shared resource availability, which characterises many cyber-physical systems.

*Robust stochastic stability definitions.* We presented stochastic ISS definitions which smoothly connect deterministic ISS results and classical stochastic stability definitions for SJLSs.

*Tools for stability.* We provided a Lyapunov characterisation for the aforementioned definitions and some easily testable sufficient conditions, often in the form of LMIs.

# 3

# ANYTIME CONTROL

## 3.1 INTRODUCTION

On a shared platform (CPU or ECU) where different tasks compete for the access to the same computation resource, the time allotted to the execution of each task can strongly fluctuates. A way to cope with this variability is to design flexible control laws, whose structure can change and adapt to the varying Quality of Service (QoS) provided by the platform. The goal is to be always capable of guaranteeing a desired Quality of Control (QoC) in spite of all such variability.

The key idea behind Anytime Control is to design controllers such that a useful result is yielded whenever the algorithm is run for at least a minimal guaranteed time, while better results can be provided if longer times are allowed.

The concept of anytime algorithms is borrowed from the field of *imprecise computation*, that has been proposed in the real-time systems literature [143, 144]. The characteristic of anytime algorithms is to always return an answer on demand; however, the longer they are allowed to compute, the better (e.g. more precise) an answer they will return. Thus, an anytime algorithm can be interrupted prematurely, still providing a valid result and improving the output accuracy as the available time increases. A periodic task is split in a *mandatory* part and one or more *optional* parts. Functionally critical subtasks are considered in the mandatory part. If all mandatory parts of a set of tasks are schedulable, the *feasible mandatory constraint* is satisfied [143].

In digital filter design [166], this philosophy has been pursued by decomposing the full-order filter in a series of lower order filters whose execution is prioritised. Execution of code implementing the first block is always guaranteed within $\tau_{min}$; code for blocks in the cascade is then executed sequentially, until a deadline event takes over. The latest computed block output is used as the anytime filter output. The overall

performance of the filter was shown in [166] to be superior to the conservative solution of always using only the first filter block.

To adopt the anytime approach in the control domain, a classical monolithic control task should be replaced by a hierarchy of control tasks of increasing complexity, each providing a correspondingly increasing performance of the controlled system. For instance, the simplest control task in the hierarchy, could be designed to guarantee only stability of the closed loop system, while whenever the scheduler provides "surplus" time, other more sophisticated control algorithms could be executed to obtain better quality of control.

However, application of the anytime algorithm idea to control is much more challenging than it may superficially appear. The main conceptual roadblock is that, as opposed to most anytime computation and filtering algorithms, anytime controllers interact in feedback with dynamic systems, which fact entails issues such as:

- *Switched System Performance:* unpredictable preemption events introduce stochastic switching among different closed-loop systems, which can subvert naïve expectations — e.g., switching between stabilising controllers may well result in overall instability. More generally, closed-loop performance is strongly influenced by switching;
- *Practicality:* implementation of both control and scheduling algorithms must be numerically accurate, yet very simple and noninvasive, not to contradict the very nature of the limited-resource, embedded control problem;
- *Hierarchical Design:* the design of a set of controllers as progressive approximations towards a given target design does not typically provide the desired performance hierarchy. Indeed, performance of closed-loop systems are not trivially related to how close controller approximations are to the target, as it is e.g. in filter design;
- *Modularity:* the computational structure of control algorithms should be inherited through the hierarchy levels, so that the computation of higher controllers in the hierarchy exploits results of computations executed for lower controllers. Although this property is not strictly required, it can greatly enhance effectiveness of anytime control.

We will discuss these issues further in the rest of the chapter.

### 3.1.1 Related work and comparison

A first attempt to apply the anytime computation paradigm to control is reported in [29, 30], where standard system reduction methods (balanced truncation or modal decomposition) are used to decompose a target controller in simpler ones. If the simplified controllers are individually stabilising, stability under switching is guaranteed by [29, 30] only under the implicit assumption that a long enough (but unquantified) dwelling time is allowed by the scheduler between switches.

On the other hand, the substantial literature on *switching system* stability (see e.g. [39, 69, 134, 214] and references therein) provides much inspiration and ideas for the problem at hand, but few results can be used directly. For instance, application of the important results of [109] would provide state-space realisations of different stabilising controllers such that the overall closed-loop systems would remain stable under *any* switching law. Unfortunately, however, the method is thought for a different application, and assumes that all controllers are designed by the internal-model approach, thus having the same (rather heavy) computational complexity. Most importantly, at each switching instant, a state-space transformation has to be applied, whose complexity is comparable to that of the controllers themselves. By the same practicality argument, algorithms for *switched system* stabilisation (such as e.g. [209, 208, 168, 122]) requiring the computation of complex functions of the state to ascertain which subsystem can be activated next time, are not applicable to the anytime control problem as we consider it here.

The thread of work closest to ours is probably that related to Firm Real Time Systems (FRTSs) [177, 25]. In a series of papers [137, 138, 141, 140], Lemmon and co-workers consider performance of Networked Control Systems (NCSs) in a FRTS framework, introduce a stochastic model to describe the task dropout process, and provide a general QoS constraint encompassing classical metrics such as average- or window-based dropout measures.

Probabilistic modelling of real-time systems is by now a widely accepted approach to avoid over-conservatism of deterministic (WCET-based) models, to which an ample and growing literature [146, 26, 27] is devoted. Within stochastic models of Real-Time (RT) systems, the use of Markov chains (cf. e.g. [71, 141, 140, 117]) is one of the most promising avenues to accurately compute the response time distribution (and deadline miss probabilities) of different tasks in systems ranging from fixed-priority (e.g. RM) to dynamic-priority (e.g. EDF).

In this chapter we also adopt a Markov Chain (MC) model to describe the sequence of time slots allotted by a scheduler for the execution of a control task (for methods to infer the parameters of the stochastic model of the scheduler, see e.g. [71, 27]). Our model differs from the one used in the FRTSs literature cited above, as we define our probabilities on the space of execution times rather than on deadline misses. A more substantial difference, however, is that we regard the scheduler characteristics to be a given in our problem, rather than a design objective. In our anytime control set-up, different control subroutines can be alternatively activated in different periods to control the same plant, and we develop a switching policy that affords better exploitation of the computational platform while guaranteeing overall closed-loop stability. The ensuing closed loop switched system is aptly described by the model of Stochastic Jump Linear Systems (SJLSs) introduced in Chapter 2 and we will make reference to the definitions and results there presented on almost sure input to state stability (as-ISS) and second-moment exponential input to state stability (2-EISS). We stress that those conditions are quite close to the well known conditions for almost sure stability ([80, 35]) and for second moment stability ([154, 114, 58, 81]) of Markov Jump Linear Systems (MJLSs).

## 3.2 THE CONTROL PROBLEM

Let $\Sigma = (A, B, C, D)$ be a given linear, discrete time, invariant SISO plant and let $\Gamma_i$, $i \in I \triangleq \{1, 2, \ldots, N\}$, be a family of controllers for $\Sigma$ such that the feedback connection of $\Sigma$ with $\Gamma_i$ is asymptotically stable for all $i$. Let the closed-loop system thus obtained be denoted as $\Sigma_i$ and let its dynamics be

$$x(t+1) = \widehat{A}_i x(t), \tag{3.2.1}$$

with $t \in \mathbb{Z}_{\geq 0}$. The set of controllers is assumed to provide a hierarchy of performance and complexity. In other terms, we assume that application of controller $i$ provides better closed-loop performance than controller $j$ if $i > j$, and that its software implementation typically requires longer execution time. Which controller will be actually executed at each period depends on the time allotted by the Real Time Operating System (RTOS) scheduler and on the execution time of different controllers, both nondeterministic quantities. An example of typical set of control goals for the system (3.2.1) is as follows:

1. Closed loop asymptotic stability of the nominal system,
2. Rejection of some class of external disturbances,
3. Robust stability w.r.t. model uncertainties.

## 3.3 STOCHASTIC MODELLING OF THE SCHEDULING PROCESS

We consider a single-processor platform with a multitasking RTOS, and a periodic control task with period $T_g$. We further assume that control inputs and outputs are time-triggered and synchronised, i.e. measurements are acquired at the beginning of each period and control inputs are released at the end. As a consequence, the control task is not affected by jitter, while the constant unit delay can be easily accounted for directly in the design of the individual controllers [50]. Assume the period $T_g$ to be divided in $M < \infty$ time slices of length $\Delta T$, and that each controller $\Gamma_j$ is implemented by a homonymous software subroutine. Both the time allotted by the scheduler to the control task, and the execution times of any subroutine, are finite multiples of $\Delta T$.

Let the time allotted to the control task during the $t$-th sampling period be described by the discrete random variable $\sigma(t)$ taking on values in the set $J_\sigma \triangleq \{1, \cdots, M\}$. When $\sigma(t) = i \in J_\sigma$ the time allotted to the control task during the $t$-th sampling period is $l_i \Delta T$, $l_i \in \{l_{min}, \cdots, l_{max}\}$ with $l_{min} \Delta T = \tau_{min}$. A simple stochastic description of the random sequence $\{\sigma(t)\}_{t \in \mathbb{Z}_{\geq 0}}$ can be given in terms of an independently and identically distributed (i.i.d.) process, with probability distribution denoted by $\overline{\pi}_\sigma = [\overline{\pi}_{\sigma_1}, \cdots, \overline{\pi}_{\sigma_M}] \in \mathcal{S}^{M-1}$. Motivated by the stochastic model of RT systems in [71, 117], we adopt in what follows a more general model based on the SJLS introduced in Chapter 2. Accordingly, $\{\sigma(\cdot)\}$ will denote a discrete-time Finite-State time-Homogeneous Irreducible Aperiodic Markov Chain (FSHIA-MC) taking values in $J_\sigma$, with transition probability matrix $P$ and i.p.d. $\overline{\pi}_\sigma$.

Since it is assumed that no deterministic information is available in advance on the effective time allowance for the control task, subroutines $\Gamma_j$ must be computed sequentially, i.e. the computation of $\Gamma_i$ cannot start until the computation of $\Gamma_{i-1}$ has terminated. Therefore, the (normalised) execution time from $\Gamma_1$ to $\Gamma_j$ is modelled as a discrete random variable $T^j$ defined on $J_\sigma$, with probability distribution $\mathsf{P}\left\{T^j = i\right\} \triangleq \overline{\pi}_{T_i^j}$, with $0 \leq \overline{\pi}_{T_i^j} \leq 1$ and $\sum_i \overline{\pi}_{T_i^j} = 1$. If $T^j = i \in J_\sigma$, then the execution time from $\Gamma_1$ to $\Gamma_j$ is equal to $l_i \Delta T$. The (time-independent) probability distribution of $T^j$ is described by $\overline{\pi}_{T^j} = [\overline{\pi}_{T_1^j}, \cdots, \overline{\pi}_{T_M^j}] \in \mathcal{S}^{M-1}$.

We now combine the probabilistic models above into a stochastic description of the highest-index executable controller in each period. This will be referred to as the (unconditioned) scheduler process $\{\varphi(\cdot)\}$. Indeed, $\{\varphi(\cdot)\}$ represents the highest index of a schedulable controller, i.e. the controller such that its execution time is the largest among those shorter than the available time. Assume first that the simplest controller $\Gamma_1$ (whose worst-case execution time is $WCET_1$) is considered to be mandatory in the anytime scheme, and that the *feasible mandatory constraint* $WCET_1 \leq \tau_{min}$ is satisfied. Let the random process $\{\varphi(\cdot)\}$ taking on values in $J_\varphi \triangleq I = \{1, \cdots, N\}$ be defined as "$\varphi(t) = i$ iff in the $t$-th period all controllers $\Gamma_j$, $j \leq i$ but no controller $\Gamma_k$, $k > i$ can be executed." Because controllers are computed sequentially, this is equivalent to stating that "$\varphi(t) = i$ iff in the $t$-th period $\Gamma_i$ can be executed but $\Gamma_{i+1}$ can not".

Let $\pi_{\varphi_i}(t) \triangleq \mathsf{P}\{\varphi(t) = i\}$. From $\pi_{\varphi_i}(t) = \sum_j \mathsf{P}\{\varphi(t) = i \mid \sigma(t) = j\}\, \pi_{\sigma_j}(t)$, it follows

$$\pi_{\varphi_i}(t) = \sum\nolimits_j \mathsf{P}\left\{T^i \leq j \cap T^{i+1} > j\right\} \pi_{\sigma_j}(t).$$

To express $\mathsf{P}\left\{T^i \leq j \cap T^{i+1} > j\right\}$ in terms of distributions $\overline{\pi}_{T^i}$, let us write

$$\begin{aligned}\mathsf{P}\left\{T^i \leq j\right\} &= \mathsf{P}\left\{\left(T^i \leq j \cap T^{i+1} \leq j\right) \cup \left(T^i \leq j \cap T^{i+1} > j\right)\right\} \\ &= \mathsf{P}\left\{T^i \leq j \cap T^{i+1} \leq j\right\} + \mathsf{P}\left\{T^i \leq j \cap T^{i+1} > j\right\},\end{aligned}$$

because the two events are mutually exclusive. Recalling that controllers are computed sequentially, $T^{i+1} \leq j$ implies $T^i \leq j$, hence $\mathsf{P}\left\{T^i \leq j \cap T^{i+1} \leq j\right\} = \mathsf{P}\left\{T^{i+1} \leq j\right\}$ and $\mathsf{P}\left\{T^i \leq j \cap T^{i+1} > j\right\} = \mathsf{P}\left\{T^i \leq j\right\} - \mathsf{P}\left\{T^{i+1} \leq j\right\}$. Defining the cumulative distribution of $T^i$ as $\kappa_{T^i} = [\overline{\pi}_{T_1^i}, \overline{\pi}_{T_1^i} + \overline{\pi}_{T_2^i}, \cdots, \sum_{k=1}^M \overline{\pi}_{T_k^i}]$, we can write $\mathsf{P}\left\{T^i \leq j \cap T^{i+1} > j\right\} = \left(\kappa_{T^i} - \kappa_{T^{i+1}}\right)_j$. The relation between the stochastic processes $\{\sigma(\cdot)\}$ and $\{\varphi(\cdot)\}$ is now in the form (2.2.3)-(2.2.4)

$$\pi_\sigma(t+1) = \pi_\sigma(t) P \tag{3.3.1}$$

$$\pi_\varphi(t) = \pi_\sigma(t) L, \tag{3.3.2}$$

with

$$L = \begin{bmatrix} \kappa_{T^1} - \kappa_{T^2} \\ \vdots \\ \kappa_{T^{n-1}} - \kappa_{T^n} \\ \kappa_{T^n} \end{bmatrix}^T$$

a row stochastic matrix.

*Remark* 3.1. The assumption $WCET_1 \leq \tau_{min}$ can be easily dropped. Indeed, if $WCET_1 > \tau_{min}$, it is sufficient to add to $J_\varphi$ the event 0, with the meaning "$\tau(t) = 0$ iff in the $t$-th period no controllers can be executed." In this case, matrix $L^T$ is modified by adding on its top a row $\kappa_{T^0} - \kappa_{T^1}$, where $\kappa_{T^0} = [1, \ldots, 1]$ is the cumulative distribution that no controller can be executed.

*Remark* 3.2. In our previous work [99], a deterministic model of the controller execution time was considered. This can be regarded as a particular case of the present treatment, where $\overline{\pi}_{T^i}$ is chosen as a vector having a 1 in $j$-th position if $j\Delta T = WCET_i$, and 0 elsewhere. Moreover, if $J_\sigma$ is limited to the set of all the WCETs, then $N = M$, $J_\sigma = J_\varphi$, $L$ becomes the identity matrix and $\{\varphi(\cdot)\}$ coincides with the FSHIA-MC $\{\sigma(\cdot)\}$.

## 3.4 PROBLEM FORMULATION

Given a set of controllers $\Gamma_i$, the associated closed loop dynamics $\widehat{A}_i$, and a probabilistic description of the process $\{\varphi(\cdot)\}$ modelling the maximum schedulable controller, a degree of freedom is left to the control task designer to make an explicit choice of the controller to be actually executed in each period. We will refer to such choice as the *switching policy*, which is defined as a map $s : \mathbb{Z}_{\geq 0} \to J_\varphi$, $t \mapsto s(t)$ determining the upper bound $i \leq s(t)$ to the index $i$ of the controller to be executed at time $t$. In other terms, at time $tT_g$, the control task computes controllers $\Gamma_1$, $\Gamma_2$, ... until $\Gamma_s$, unless a deadline event occurs forcing it to provide only $\Gamma_\varphi$, the highest controller computed before the deadline.

Application of a switching policy $s$ to a set of feedback systems $\Sigma_i$, $i \in J_\varphi$ under a given scheduler $\{\varphi(\cdot)\}$ generates a switched linear system $(\Sigma_i, \varphi, s)$ which, under general hypotheses, is a SJLS with a conditioned i.p.d. $\overline{\pi}_{\varphi|s}$.

As an example, the most conservative policy is to set $s(t) \equiv 1$, i.e. forcing always the execution of the simplest controller $\Gamma_1$, regardless of the probable availability of more computational time ($\overline{\pi}_{\varphi|s} = [1, 0, \cdots, 0]$). If the feasible mandatory constraint $WCET_1 \leq \tau_{min}$ is satisfied, this (non-switching) policy guarantees stability of the resulting closed loop system.

On the opposite, a "greedy" strategy would set $s(t) \equiv N$, which leads to computing $\Gamma_\varphi$ for all $t$ (hence $\overline{\pi}_{\varphi|s} = \overline{\pi}_\varphi$). Although this policy attempts at maximising the utilisation of the most performing controller, it is well known that switching arbitrarily among asymptotically stable systems $\Sigma_i$ may easily result in an unstable behaviour [133]. A sufficient condition for the greedy switching policy to provide an AS-stable (actually as-ISS) system is provided by Corollary 2.2, where $A_j = \widehat{A}_j$. Notice that here we assume that all closed-loop systems have the same number of states, which coincides with the sum of the number of states of the plant and of the largest controller (the actual arrangement of the state vectors for closed-loop systems is illustrated in detail in the implementation Section 3.7).

If this condition is not verified, it is important to investigate whether there exist scheduling policies which can ensure AS-stability other than the over-conservative non-switching choice. To do so, assume that the set

$$\overline{\Pi} = \left\{ \overline{\pi} \mid \prod_{i \in J_\varphi} \left\| \widehat{A}_i \right\|^{\overline{\pi}_i} < 1 \right\}$$

is not empty for some matrix norm. Given a scheduler i.p.d. $\overline{\pi}_\varphi$, in order for a compatible and stabilising conditioned i.p.d. $\overline{\pi}_{\varphi|s} \in \overline{\Pi}$ to exist, further constraints have to be satisfied. Indeed, let $\overline{\pi}_d \in \overline{\Pi}$. For a switching policy to exist which can alter a given scheduler probability distribution $\overline{\pi}_\varphi$ into $\overline{\pi}_{\varphi|s} = \overline{\pi}_d$, the following must hold:

$$\begin{aligned}
&\overline{\pi}_{d_N} \leq \overline{\pi}_{\varphi_N} && (C.1)\\
&\overline{\pi}_{d_{N-1}} \leq \overline{\pi}_{\varphi_{N-1}} + \left(\overline{\pi}_{\varphi_N} - \overline{\pi}_{d_N}\right) && (C.2)\\
&\quad \vdots \\
&\overline{\pi}_{d_1} \leq \overline{\pi}_{\varphi_1} + \left(\overline{\pi}_{\varphi_2} - \overline{\pi}_{d_2}\right) + \cdots + \left(\overline{\pi}_{\varphi_N} - \overline{\pi}_{d_N}\right) && (C.N)
\end{aligned}$$

where $\overline{\pi}_\varphi = \left[\overline{\pi}_{\varphi_1}, \cdots, \overline{\pi}_{\varphi_N}\right] \in \mathcal{S}^{N-1}$, and $\overline{\pi}_d = \left[\overline{\pi}_{d_1}, \cdots, \overline{\pi}_{d_N}\right] \in \mathcal{S}^{N-1}$.

Inequalities $(C.1)$–$(C.N)$ take into account the fact that no switching law can alter the scheduler so as to give more computational time to control tasks than it is made available by the scheduler. Furthermore, constraints $(C.2)$–$(C.N)$ model the fact that the probability $\overline{\pi}_{d_i}$ of the $i$-th controller can be increased only at the expenses of a reduction of the probabilities $\overline{\pi}_{d_j}$, $j > i$ of more complex controllers.

**Definition 3.1.** Given a set of matrices $\widehat{A}_i$, $i \in J_\varphi$, and a scheduler i.p.d. $\overline{\pi}_\varphi$, the set of feasible AS-stabilising scheduling probabilities is defined as $\overline{\Pi}_d = \{\overline{\pi}_d\}$, with

$$\begin{aligned}
&D2.1) && \prod_{i=1}^{N} \left\| \widehat{A}_i \right\|^{\overline{\pi}_{d_i}} < 1\\
&D2.2) && 0 \leq \overline{\pi}_{d_i} \leq 1\\
&D2.3) && \sum_{i=1}^{N} \overline{\pi}_{d_i} = 1\\
&D2.4) && \overline{\pi}_{d_i} \leq \overline{\pi}_{\varphi_i} + \sum_{j=i+1}^{N} \overline{\pi}_{\varphi_j} - \sum_{j=1+1}^{N} \overline{\pi}_{d_j} \,.
\end{aligned}$$

Assuming that $\overline{\Pi}_d \neq \varnothing$, it is natural to consider a QoC metric within the set of feasible stabilising schedules. Based on the hypothesis that controllers are hierarchically ordered by their closed-loop performance, so that use of controller $\Gamma_i$ provides better results than $\Gamma_j$, $j < i$, a natural QoC metric for the problem at hand can be simply given as

$$\mathcal{J}(\overline{\pi}_d) = \sum_{i=1}^{N} d_i^2 \overline{\pi}_{d_i}, \tag{3.4.1}$$

i.e. the second moment of a random variable $d \in \{d_1, \dots, d_n\}$ with i.p.d. $\overline{\pi}_d$, where $d_i$ is the performance index associated with $\Gamma_i$, $0 < d_i < d_j$, $i < j$.

In conclusion, we state the following problem

**Problem 3.1** (Optimal Switching Policy (OSP) Problem)**.** For a given plant $\Sigma$, a set of controllers $\Gamma_i$, with associated performance index $d_i$ and probabilistic execution time $T^i$, and a probabilistic scheduler process $\{\varphi(\cdot)\}$, find a switching policy $s(\cdot)$ that maximises $\mathcal{J}(\overline{\pi}_{\varphi|s})$, subject to $\overline{\pi}_{\varphi|s} \in \overline{\Pi}_d$.

## 3.5 INDEPENDENT STOCHASTIC SWITCHING POLICY

In this section, we tackle problem 3.1 above by introducing a switching law which is itself stochastic, and is based on the concept of a *conditioning* Markov chain. The stochastic properties of the scheduler process and conditioning chain interact to produce a resulting switched system. To study such interaction, we will make use of the operations of *merging* and *aggregating* stochastic processes.

The merging of two finite state Markov chains $\{\alpha(\cdot)\}$ and $\{\beta(\cdot)\}$, defined on the state spaces $J_\alpha$ and $J_\beta$, respectively, is the stochastic process $\{\alpha\beta(\cdot)\}$ defined on $J_{\alpha\beta} \triangleq J_\alpha \times J_\beta$ such that $\alpha\beta(t) = (\alpha(t), \beta(t))$ for every $t \in \mathbb{Z}_{\geq 0}$. The following holds:

**Proposition 3.1.** *For two independent, FSHIA-MCs $\{\alpha(\cdot)\}$ and $\{\beta(\cdot)\}$ with transition probability matrices $P_\alpha = \left({}^{\alpha}p_{ij}\right)_{n\times n}$ and $P_\beta = \left({}^{\beta}p_{ij}\right)_{l\times l}$, and initial probability distributions $\pi_\alpha(0)$ and $\pi_\beta(0)$, let $\overline{\pi}_\alpha$ and $\overline{\pi}_\beta$ denote their respective (unique) invariant probability distributions. Then, for the merging $\{\alpha\beta(\cdot)\}$, it holds*

**i)** $\{\alpha\beta(\cdot)\}$ is a FSHIA-MC whose statistics are given by the transition probability matrix $P_{\alpha\beta} = \left({}^{\alpha\beta}p_{ij}\right)_{nl\times nl} = P_\alpha \otimes P_\beta$ and by the initial probability distribution $\pi_{\alpha\beta}(0) = \pi_\alpha(0) \otimes \pi_\beta(0)$ ($\otimes$ denotes the Kronecker product);

**ii)** the evolution of the chain $\{\alpha\beta(\cdot)\}$ is given by

$$\pi_{\alpha\beta}(t) = \pi_\alpha(t) \otimes \pi_\beta(t) = \left(\pi_\alpha(0) \otimes \pi_\beta(0)\right) P_{\alpha\beta}^t \tag{3.5.1}$$

with $t \in \mathbb{Z}_{\geq 0}$. Moreover, $\pi_{\alpha\beta}(t)$ converges to the unique i.p.d.

$$\overline{\pi}_{\alpha\beta} = \overline{\pi}_\alpha \otimes \overline{\pi}_\beta \tag{3.5.2}$$

for any initial distribution $\pi_{\alpha\beta}(0)$.

Consider further a stochastic process $\{\rho(\cdot)\}$ defined on the finite state space $J_\rho$ and a function $g : J_\rho \to G$ mapping $J_\rho$ in another finite state space $G$. The *aggregation* of $\{\rho(\cdot)\}$ with respect to $g$ is the stochastic process $\{\rho^*(\cdot)\}$ defined on the quotient space of $J_\rho$ by the equivalence relation $\rho_i \equiv \rho_j$ iff $g(\rho_i) = g(\rho_j)$. Notice that the aggregated process is not necessarily Markovian even if the original process is.

Coming back to the stochastic processes describing the scheduling and execution of controllers, consider an independent FSHIA-MC $\{\varsigma(\cdot)\}$ defined on a finite state space

$J_\varphi$. We will associate a switching law to the stochastic process $\{\varsigma(\cdot)\}$ in the following explicit sense:

> **Independent Stochastic Switching Policy (ISSP)**. At the $t$–th period, the control task produces the results of the $k$-th controller $\Gamma_k$ iff

1. $\varphi(t) \geq k$ and $\varsigma(t) = k$; or
2. $\varsigma(t) \geq k$ and $\varphi(t) = k$.

The fact that process $\{\varphi(\cdot)\}$ may not be a Markov chain implies that the merged process $\{\varphi\varsigma(\cdot)\}$ is also not a Markov chain in general. However, the probability distribution $\pi_{\varphi\varsigma}(t)$ of $\{\varphi\varsigma(\cdot)\}$ is linearly related to the distribution $\pi_{\sigma\varsigma}(t)$ of the merged Markov chain $\{\sigma\varsigma(\cdot)\}$. Indeed, from the independence of the random variables $\varphi(t)$ and $\varsigma(t)$ for every $t \in \mathbb{Z}_{\geq 0}$, by the mixed product rule we can write

$$\begin{aligned} \pi_{\varphi\varsigma}(t) &= \pi_\varphi(t) \otimes \pi_\varsigma(t) = (\pi_\sigma(t)L) \otimes \pi_\varsigma(t) \\ &= (\pi_\sigma(t) \otimes \pi_\varsigma(t))\,(L \otimes I_N) \\ &= \pi_{\sigma\varsigma}(t)\,(L \otimes I_N)\,, \end{aligned} \tag{3.5.3}$$

with $I_N$ the identity matrix.

We identify states of $\{\varphi\varsigma(\cdot)\}$ resulting in the execution of the same controller by introducing the aggregating function $g : J_{\varphi\varsigma} \to J_\varphi$ with $g(i,j) \triangleq \min\{i,j\}$ for any $i, j \in J_\varphi$. The switching law choice above amounts to setting the conditioned probability distribution $\pi_{\varphi|\varsigma}$ equal to $\pi^\star$, which is the probability distribution of the aggregated process $\{(\varphi\varsigma)^\star(\cdot)\}$. It can be easily verified that the evolution of $\pi^\star(t) = [\pi_1^\star(t), \ldots, \pi_N^\star(t)]$ is given by

$$\pi^\star(t) = \big(\pi_\varphi(t) \otimes \pi_\varsigma(t)\big)\,H \tag{3.5.4}$$

with $H \in \{0,1\}^{N^2 \times N}$ such that $H = [H_1^T, H_2^T, \ldots, H_N^T]^T$ and

$$H_i^T = \left[\begin{array}{c|c} I_{i,i} & \begin{matrix} 0_{i-1,N-i} \\ 1_{1,N-i} \end{matrix} \\ \hline 0_{N-i,i} & 0_{N-i,N-i} \end{array}\right].$$

*Remark* 3.3. Because of the linear mappings (3.5.3) and (3.5.4) the process $\{(\varphi\varsigma)^\star(\cdot)\}$ admits an i.p.d. to which each initial distribution of type $(\pi_\sigma(0) \otimes \pi_\varsigma(0))\,(L \otimes I_N)\,H$ converges. In particular, we have

$$\begin{aligned} \overline{\pi}^\star &= \big(\overline{\pi}_\varphi \otimes \overline{\pi}_\varsigma\big)\,H, \\ &= (\overline{\pi}_\sigma \otimes \overline{\pi}_\varsigma)\,(L \otimes I_N)\,H \end{aligned} \tag{3.5.5}$$

In view of the linear relation between $\overline{\pi}^\star$ and $\overline{\pi}_\sigma \otimes \overline{\pi}_\varsigma$ through the row stochastic matrix $(L \otimes I_N)\,H$, the contractivity condition of the Corollary 2.2 applies to $\overline{\pi}^\star$.

*Remark* 3.4. It can be observed that the choice of a switching law by a Markov chain $\{\varsigma(\cdot)\}$ independent of the scheduler process $\{\varphi(\cdot)\}$ is not the most general possible choice. However, besides simplifying the analysis considerably, this choice has the advantage of not requiring on-line computations. Indeed, an arbitrary realisation of the process $\{\varsigma(\cdot)\}$ can be computed off-line and used as a switching law $s(\cdot)$, which fact is crucial from a practicality point of view.

### 3.5.1 One-step average contractivity condition

Based on results of the previous section, we seek a solution of Problem 3.1 with a structure such as in (3.5.5). The conditioned i.p.d. $\overline{\pi}_{\varphi|\varsigma}$, which for notational simplicity we will denote henceforth as $\overline{\pi}^\star = \overline{\pi}_{\varphi|\varsigma}$, can be written more explicitly as

$$\overline{\pi}^\star_i = \sum_{(h,k)\in\chi_i} \overline{\pi}_{\varphi_h}\overline{\pi}_{\varsigma_k} \tag{3.5.6}$$

where $\chi_i = \{(h,k) \in J_{\varphi\varsigma} \mid g(h,k) = i\}$.

It actually turns out that the structure of $\overline{\pi}^\star$ described in (3.5.6), resulting from the choice of an independent conditioning chain, simplifies the formulation of the synthesis problem substantially. Indeed, it can be proven by simple if lengthy arguments, that constraints *D*2.2, *D*2.3, and *D*2.4 in Definition 3.1 are automatically satisfied by an i.p.d. $\overline{\pi}^\star$ as in (3.5.6). Furthermore, for such $\overline{\pi}^\star$, constraint *D*2.1 can be rewritten (with the proviso that $\|\widehat{A}_i\| \neq 0 \ \forall i$) as

$$\begin{aligned}\ln\left(\prod_{i=1}^{N}\left\|\widehat{A}_i\right\|^{\overline{\pi}^\star_i}\right) &= \ln\left(\prod_{(h,k)\in J_{\varphi\varsigma}}\left\|\widehat{A}_{g(h,k)}\right\|^{\overline{\pi}_{\varphi_h}\overline{\pi}_{\varsigma_k}}\right)\\ &= \sum_{k=1}^{N}\overline{\pi}_{\varsigma_k}\sum_{h=1}^{N}\overline{\pi}_{\varphi_h}\ln\left(\left\|\widehat{A}_{g(h,k)}\right\|\right) < 0.\end{aligned}$$

By introducing the QoC vector $c_d \in \mathbb{R}^N$, $c_{d_i} = d_i^2$, such that $\mathcal{J}(\overline{\pi}^\star) = \sum_{i=1}^N c_{d_i}\overline{\pi}^\star_i = c_d\overline{\pi}^{\star\,T}$, using (3.5.6) and rearranging terms, the OSP problem restricted to the ISSP class can be stated as a classical linear programming problem in the unknown $\overline{\pi}_\varsigma$:

*OSP Problem*

$$\begin{aligned}\max_{\overline{\pi}_\varsigma}\ \mathcal{J}(\overline{\pi}^\star) &= \max_{\overline{\pi}_\varsigma}\ \overline{\pi}_\varphi M_d\,\overline{\pi}_\varsigma^T\\ 1)\quad &\overline{\pi}_\varphi M_c\overline{\pi}_\varsigma^T \le -\varepsilon < 0\\ 2)\quad &0 \le \overline{\pi}_{\varsigma_i} \le 1\\ 3)\quad &\textstyle\sum_{i=1}^N \overline{\pi}_{\varsigma_i} = 1,\end{aligned}$$

where $(M_c)_{ij} = \ln\left(\left\|\widehat{A}_{\min(i,j)}\right\|\right)$ and $(M_d)_{ij} = c_{d_{\min(i,j)}}$. Notice that $\varepsilon > 0$ indicates here a desired contractivity margin, while the use of non-strict inequalities in 2) implies that a non-irreducible Markov chain $\varsigma(t)$ is accepted as a possible solution.

It is also worth noting that a necessary condition for OSP to be feasible is that there exists $\widehat{A}_i$ such that $\|\widehat{A}_i\| < 1$. This may easily not be the case for an arbitrary

matrix norm. On the other hand, if a matrix norm for which $\|\widehat{A}_1\| < 1$ is chosen, and $WCET_1 \leq \tau_{min}$ (which is reasonable to assume), then feasibility is guaranteed. While finding such a norm is always possible if $\|\widehat{A}_1\|$ is Schur, the choice of such an $\widehat{A}_1$–adapted norm tends to bias solutions of OSP towards using the simplest controller more often. This in turn tends to produce low performance index $\mathcal{J}$. Therefore, it can be useful to consider a more general position of the scheduling problem, which is described next.

### 3.5.2 *Multi-step average contractivity condition*

In this section we exploit the concept of $m$-lifted SJLS system presented in Section 2.3.3. If the one-step average contractivity solution is based on the Corollary 2.2, the multi-step solution is based on the Corollary 2.3. The advantage of using $m$-step contractivity is illustrated by the following Lemma:

**Lemma 3.1.** *If there exists $i \in J_\varphi$ such that $\widehat{A}_i$ is Schur, then for any given matrix norm $\|\cdot\|$, there exists $m \in \mathbb{N}$ and $\widetilde{\imath} = (i_1, \dots, i_m) \in J_{\widetilde{\varphi}} = J_\varphi^m$ such that the solution set is not empty, i.e.*

$$\overline{\Pi} = \left\{ \overline{\pi} \mid \prod_{\widetilde{\imath} \in J_{\widetilde{\varphi}}} \left\| \widehat{A}_{\widetilde{\imath}} \right\|^{\overline{\pi}_{\widetilde{\imath}}} < 1 \right\} = \varnothing,$$

*where $\widehat{A}_{\widetilde{\imath}} = \widehat{A}_{i_m} \cdots \widehat{A}_{i_1}$.*

This result guarantees that a $m$-step stabilising switching policy exists, for large enough $m$: by the assumption that $WCET_1 \leq \tau_{min}$, it is indeed sufficient to choose $i = 1$ in the Lemma above. To exploit the more general $m$-step contractivity condition given in Corollary 2.3, we introduce a switching law such that some control patterns, i.e. sequences of symbols in $J_\varphi$, are preferentially used with respect to others. This may imply associating to a matrix sequence a steady-state probability of occurrence different from the product of the probabilities of each matrix. Accordingly, an unconstrained chain $\{\widetilde{\varsigma}(\cdot)\}$ will be used for conditioning (as opposed to using a lifted version of the one-step conditioning chain $\{\varsigma(\cdot)\}$), which can be interpreted as suggesting the sequence of controllers to be executed in the next $m$ steps.

Let $\{\widetilde{\sigma}(\cdot)\}$ denote the lifted Markov chain describing the time allotted to control task computation in the next $m$ periods, whose states are sequences $\widetilde{\imath} = (i_1, \dots, i_m)$ of the original symbols $i_j \in J_\sigma$ defined in the new state space $\widetilde{J}_\sigma = J_\sigma^m$ with cardinality $M^m$. Also let $\{\widetilde{\varphi}(\cdot)\}$ denote the lifted scheduler process taking values in the set $J_{\widetilde{\varphi}} = J_\varphi^m$.

To compute the distribution of the lifted process $\{\widetilde{\varphi}(\cdot)\}$, consider the matrix $\widetilde{L} = (\widetilde{\ell}_{\widetilde{\imath}\widetilde{\jmath}})_{\substack{\widetilde{\imath} \in \widetilde{J}_\sigma \\ \widetilde{\jmath} \in J_{\widetilde{\varphi}}}}$, whose $(\widetilde{\imath}, \widetilde{\jmath})$ entry is $\mathsf{P}\{\widetilde{\varphi}(h) = \widetilde{\jmath} \mid \widetilde{\sigma}(h) = \widetilde{\imath}\}$. Recalling the meaning of the process $\{\varphi(\cdot)\}$ and the stationarity of the random variables $T^{i_k}$, this probability can be written as $\mathsf{P}\{\widetilde{\varphi}(h) = \widetilde{\jmath} \mid \widetilde{\sigma}(h) = \widetilde{\imath}\} = \mathsf{P}\left\{\bigcap_{k=1}^m \left(T^{i_k} \leq j_k \cap T^{i_k+1} > j_k\right)\right\}$. Thanks to the independence of the variables $T^{i_k}$ and $T^{i_r}$ for every $i_k, i_r \in J_\sigma$ $k \neq r$, we find, as in Section 2.3.3, $\widetilde{L} = L \otimes \cdots \otimes L$ (repeated $m$ times).

The distribution of $\{\widetilde{\varphi}(\cdot)\}$ and $\{\widetilde{\sigma}(\cdot)\}$ are hence linearly related through $\widetilde{L}$ as shown in equation (2.3.6). A $m$-step equivalent of Problem 3.1 is directly obtained by replacing $\widehat{A}_i$, $\overline{\pi}_i^\star$ and $\overline{\pi}_{\varphi_i}$ with $\widehat{A}_{\widetilde{\imath}}$, $\overline{\widetilde{\pi}}_{\widetilde{\imath}}^\star$ and $\overline{\pi}_{\widetilde{\varphi}_{\widetilde{\imath}}}$, respectively.

If a set of steady-state probability distributions $\overline{\widetilde{\Pi}}_d$ exists solving the $m$-step version of Problem 3.1, the synthesis problem is again to find a steady-state probability distribution $\overline{\pi}_{\widetilde{\varsigma}}$ for the chain $\{\widetilde{\varsigma}(\cdot)\}$ such that the aggregated process $\{(\widetilde{\varphi\varsigma})^\star(\cdot)\}$ has steady-state distribution $\overline{\widetilde{\pi}}^\star \in \overline{\widetilde{\Pi}}_d$, with the aggregation induced by the element-wise minimum function.

Finally, a QoC metric generalising (3.4.1) is obtained by setting $\mathcal{J}(\overline{\widetilde{\pi}}^\star) = \widetilde{c}_d \overline{\widetilde{\pi}}^{\star\,T}$, with $\widetilde{c}_d = c_d \otimes \cdots \otimes c_d$ ($m$ times). Indeed, in this case any weight $\widetilde{c}_{d_{\tilde{\imath}}}$ is the product of weights related to the sequence $d_{\tilde{\imath}} = d_{i_1} \cdots d_{i_m}$. With these stipulations, an optimal multistep switching problem for fixed $m$ can be formulated in the parameters $\overline{\pi}_{\widetilde{\varsigma}}$ in the same terms as the OSP.

## 3.6 Robust synthesis w.r.t. imperfect knowledge of the scheduler

We consider now the case that an exact stochastic model of the scheduler is not available, rather it is subject to Unknown But Bounded (UBB) uncertainties. For the sake of simplicity, we limit our robustness analysis to the one–step average contractivity condition.

Assume first that UBB uncertainties affect the steady-state probabilities of the process $\{\sigma(\cdot)\}$ establishing how much time is allotted to the control task in each period, but that the probabilistic computational model of the different controllers is known. In other terms, we assume that the matrix $L$ is given, while $\overline{\pi}_\sigma$ is only constrained to belong to a polytopic set with a finite number of vertices, i.e. $\overline{\pi}_\sigma \in \Pi_\sigma$, with $\Pi_\sigma = \operatorname{conv}\left\{\pi_\gamma^1, \ldots, \pi_\gamma^q\right\}$ (conv denotes the convex hull operator).

Due to the linearity of the mapping between $\overline{\pi}_\sigma$ and $\overline{\pi}_\varphi$, $\Pi_\sigma$ is mapped in the polytopic set $\Pi_\varphi = L\Pi_\sigma = \operatorname{conv}\left\{\pi_\varphi^1, \ldots, \pi_\varphi^r\right\}$, $r \leq q$, where $\pi_\varphi^i \in \left\{L\pi_\sigma^1, \ldots, L\pi_\sigma^q\right\}$, $i = 1, \ldots, r$.

This uncertainty description lends itself to a very simple robustness problem formulation. Indeed, being that problem OSP is linear, hence convex, w.r.t. $\overline{\pi}_\varphi$, the set of $\overline{\pi}_\varsigma$ solving OSP for every $\overline{\pi}_\varphi \in \Pi_\varphi$ can be characterised simply by replacing the first inequality in OSP with $r$ inequalities, one for each vertex $\pi_\varphi^i$. Hence, the new solution set is the polytope

$$\Pi_\varsigma = \begin{cases} \left(M_{\pi_\varphi} M_c\right) \overline{\pi}_\varsigma^T \leq -\varepsilon < 0 \\ 0 \leq \overline{\pi}_{\varsigma_i} \leq 1 \\ \sum_{i=1}^N \overline{\pi}_{\varsigma_i} = 1, \end{cases}$$

with $M_{\pi_\varphi} = \begin{bmatrix} \pi_\varphi^{1T} & \pi_\varphi^{2T} & \cdots & \pi_\varphi^{rT} \end{bmatrix}^T$. Exploiting the linearity of the index function in OSP w.r.t. both $\overline{\pi}_\varphi$ and $\overline{\pi}_\varsigma$, the search for a robust optimal switching policy can be cast as a bilinear programming problem with disjoint constraints:

*ROSP Problem*

$$\max_{\overline{\pi}_\varsigma \in \Pi_\varsigma} \min_{\overline{\pi}_\varphi \in \Pi_\varphi} \overline{\pi}_\varphi M_d \overline{\pi}_\varsigma^T. \tag{3.6.1}$$

Such a problem admits an optimal solution $(\widehat{\overline{\pi}}_\varsigma, \widehat{\overline{\pi}}_\varphi)$ where $\widehat{\overline{\pi}}_\varsigma \in \text{vert}\{\Pi_\varsigma\}$ and $\widehat{\overline{\pi}}_\varphi \in \text{vert}\{\Pi_\varphi\}$ [120] (with $\text{vert}\{\Pi\}$ denoting the set of vertices of $\Pi$). Applying von Neumann's minimax theorem [203], an equivalent formulation of ROSP is given by

$$\min_{\overline{\pi}_\varphi \in \Pi_\varphi} \max_{\overline{\pi}_\varsigma \in \Pi_\varsigma} \overline{\pi}_\varphi M_d \overline{\pi}_\varsigma^T.$$

An optimal solution $(\widehat{\overline{\pi}}_\varsigma, \widehat{\overline{\pi}}_\varphi)$ to ROSP can then be found by the following algorithm:

1. for every $\pi_\varphi^i \in \text{vert}\{\Pi_\varphi\} = \{\pi_\varphi^1, \dots, \pi_\varphi^r\}$, solve the linear program

$$\pi_\varsigma^i = \arg\max_{\overline{\pi}_\varsigma \in \Pi_\varsigma} \pi_\varphi^i M_d \overline{\pi}_\varsigma^T,$$

   and build the set $\Pi_{\varphi\varsigma} = \{(\pi_\varphi^i, \pi_\varsigma^i), i = 1, \dots, r\}$;

2. Exhaustively search $\Pi_{\varphi\varsigma}$ and find

$$(\widehat{\overline{\pi}}_\varsigma, \widehat{\overline{\pi}}_\varphi) = \arg\min_{(\pi_\varphi, \pi_\varsigma) \in \Pi_{\varphi\varsigma}} \pi_\varphi^i M_d \left(\pi_\varsigma^i\right)^T.$$

Consider now the more realistic case that the transition probability matrix $P$ of the Markov chain $\{\sigma(\cdot)\}$ is also affected by UBB uncertainties, described by a polytope of stochastic matrices $\mathcal{P} = \text{conv}\{P_1, \dots, P_m\}$ such that $P \in \mathcal{P}$. The set of steady-state distributions $\overline{\pi}_\sigma$ corresponding to $\mathcal{P}$, i.e.

$$\mathcal{Q} = \left\{\overline{\pi}_\sigma \in \mathcal{S}^{M-1} \mid \overline{\pi}_\sigma (P - I) = 0,\ P \in \mathcal{P}\right\}$$

is unfortunately not convex in general. In fact, if $\overline{\pi}_\sigma^i$ denote the steady-state vectors corresponding to vertex matrices $P_i$, there is no guarantee that $\mathcal{Q}$ is included in $\text{conv}\{\overline{\pi}_\sigma^1, \dots, \overline{\pi}_\sigma^m\}$.

A numerically tractable condition for robust AS-stability (or as-ISS) can be obtained by constructing a polytope $\Pi_\mathcal{P}$ providing an outer (conservative) approximation of $\mathcal{Q}$, i.e. $\Pi_\mathcal{P} \supseteq \mathcal{Q}$.

Consider the image under $\mathcal{P}$ of a set $W \subseteq \mathcal{S}^{M-1}$, denoted as $\mathcal{P}(W) = \text{conv}\{WP_1, \dots, WP_m\}$. As a consequence of the linearity of the $\mathcal{P}$ map, if $W$ is a polytope $W = \text{conv}\{w^1, \dots, w^r\}$, then also $\mathcal{P}(W)$ is a polytope, described by $\mathcal{P}(W) = \text{conv}_{\substack{i=1,\dots,m \\ j=1,\dots,r}} \{w^j P_i\}$. Because $\mathcal{Q}$ is the set of all fixed points for the map $\mathcal{P}$, i.e. for every $\overline{\pi}_\sigma \in \mathcal{Q}$, there exists $P \in \mathcal{P}$ such that $\overline{\pi}_\sigma P = \overline{\pi}_\sigma$, it holds $\mathcal{P}(\mathcal{Q}) \supseteq \mathcal{Q}$. For the same reason, for $W \supseteq \mathcal{Q}$, it holds $\mathcal{P}(W) \supseteq \mathcal{Q}$. It is worth noting that in general $\mathcal{P}(W) \not\supseteq W$, but, being $W \supseteq \mathcal{Q}$ and $\mathcal{P}(W) \supseteq \mathcal{Q}$, clearly $\mathcal{P}(W) \cap W \supseteq \mathcal{Q}$. This suggests the following iterative algorithm:

$$\begin{aligned} \Pi_\mathcal{P}(0) &\triangleq \mathcal{S}^{M-1} \\ \Pi_\mathcal{P}(k+1) &= \mathcal{P}\left(\Pi_\mathcal{P}(k)\right) \cap \Pi_\mathcal{P}(k), \end{aligned}$$

providing a non-increasing sequence of outer polytopic approximations for $\mathcal{Q}$ as desired. If UBB uncertainties also affect the mapping $L$, i.e. if we assume $L \in \mathcal{L} = \text{conv}\left\{L^1, \ldots, L^p\right\}$, then the approximating polytope $\Pi_{\mathcal{P}}$ is mapped by means of all the mappings in $\mathcal{L}$ in the polytope $\mathcal{L}\Pi_{\mathcal{P}} = \text{conv}_{\substack{i=1,\ldots,p \\ j=1,\ldots,q}} \left\{\pi_\sigma^j L^i\right\}$. The general ROSP problem is therefore restated by simply replacing the polytope $\Pi_\varphi$ with $\mathcal{L}\Pi_{\mathcal{P}}$.

## 3.7 IMPLEMENTATION OF ANYTIME CONTROL SYSTEMS

In this section we discuss the problem of designing an ordered set of feedback control algorithms for the given plant $\Sigma$, whose state $w \in \mathbb{R}^n$ and output $y \in \mathbb{R}$ evolve with the input $u \in \mathbb{R}$ as

$$\begin{aligned} w(t+1) &= Aw(t) + Bu(t) \\ y(t) &= Cw(t). \end{aligned} \tag{3.7.1}$$

A first, straightforward approach consists of designing a set of $N$ independent controllers $\Gamma_i$:

$$\begin{aligned} z_i(t+1) &= F_i z_i(t) + G_i y(t) \\ u(t) &= R_i z_i(t) + S_i y(t), \end{aligned}$$

where $z_i \in \mathbb{R}^{n_i}$. A hierarchical design, with $n_i < n_{i+1}$ and increasing performance, can be made e.g. by adopting a sequence of design procedures of increasing sophistication and/or by introducing successive refinements of design specifications. Controllers are then computed sequentially in the order $\Gamma_1, \Gamma_2, \ldots, \Gamma_\varsigma$ according to the switching policy $\varsigma$ proposed above.

For AS stability analysis purposes only, it is convenient to set up a compound state vector $x \in \mathbb{R}^{n_c}$, with $n_c = n + n_1 + \cdots + n_N$, which includes the states of the plant and those of each controller. The corresponding closed-loop matrices $\widehat{A}_i \in \mathbb{R}^{n_c \times n_c}$ are computed as follows

$$\widehat{A}_i = \begin{bmatrix} A - BS_iC & -B\widetilde{R}_i \\ \widetilde{G}_iC & \widetilde{F}_i \end{bmatrix}, \tag{3.7.2}$$

where, assuming that the "sleeping" states corresponding to inactive controllers $\Gamma_i$, $i \neq \varsigma(t)$ are reset to zero, we have

$$\begin{aligned} \widetilde{R}_i &= [0, \cdots, 0, R_i, 0, \cdots, 0] \\ \widetilde{F}_i &= \text{diag}\,(0, \cdots, 0, F_i, 0, \cdots, 0) \\ \widetilde{G}_i^T &= \left[0, \cdots, 0, G_i^T, 0, \cdots, 0\right] \end{aligned}$$

with the non-zero blocks in the $i$–th block positions. It should be noticed that this independent controller design approach is the simplest but does not provide any modularity: indeed, in such a mutually exclusive scheme, all computation results for $\Gamma_1, \ldots, \Gamma_{\varsigma-1}$ are eventually discarded.

To obtain a modular design for anytime control, one could pursue a top-down approach, starting with the design of a complex, high-performance controller $\Gamma_N$ (by e.g. a $H_\infty$ technique applied to the full set of performance requirements). Progressively

| | Comp. Compl. | Num. Reliab. |
|---|---|---|
| Generic | $N\,(N+2)$ | – |
| Companion | $2N$ | low |
| Jordan | $2N$ to $3N$ | high |

Table 3.7.1: Computational complexity and numerical reliability of different state-space realisations of a strictly proper transfer function $G(z)$ with $N$ poles.

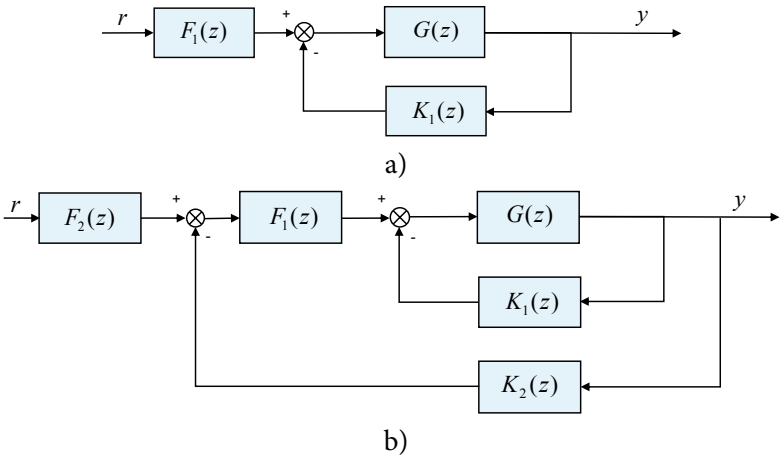


Figure 3.7.1: Stages of a classical cascade design procedure

simpler controllers $\Gamma_i$, $N-1 \geq i \geq 1$ may then be obtained by e.g. model reduction techniques. As already remarked, however, this approach does not systematically guarantee closed-loop performance under switching.

Moreover, most model reduction techniques require state-space realisations with full dynamic matrices $F_i$, which makes them impractical in real-time embedded applications. Indeed, practicality requirements imply careful consideration of algorithmic implementations of control laws [10]. Table 3.7.1 reports a comparison among three different state space representations for SISO systems w.r.t. computational complexity (considered as the number of multiplications except by 0 or 1) and numerical reliability. The generic case assumes no particular structure in the systems matrices.

We propose a simple, bottom-up design technique which is suitable for addressing the main requirements of anytime control algorithms. The method is based on classical cascade design. Consider the two design stages illustrated in Figure 3.7.1, in which controllers are designed to ensure increasing performance by any classical synthesis technique (system, controllers and filters are represented by their discrete transfer functions).

The scheme in Figure 3.7.1 cannot be implemented as a modular anytime control, because after computation of the a) scheme, the input to the $F_1(z)$ block needs to be recomputed completely if the b) scheme is to be applied. However, by simple block manipulations, the scheme in Figure 3.7.2 can be obtained, where we set $\widehat{K}_2(z) =$

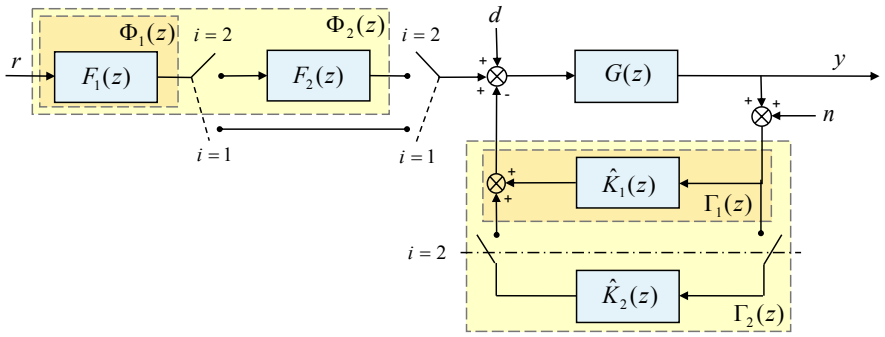


FIGURE 3.7.2: A switched control scheme suitable for anytime control implementation. The scheme is equivalent to Figure 3.7.1-a when the switches are in the $i = 1$ position, and to Figure 3.7.1-b for $i = 2$.

$F_1(z)K_2(z)$.

The scheme in Figure 3.7.2 is suitable for anytime implementation. Indeed the series of $F_1(z)$ and $F_2(z)$ is in open-loop (hence equivalent to an anytime filter $F_1(z)F_2(z)$), while the parallel connection in the feedback loop is simply obtained by summing the value computed in $\widehat{K}_2(z)$ to the value computed previously in $\widehat{K}_1(z) = K_1(z)$. Using Jordan form realisations of the blocks provides good numerical accuracy as well as low computational complexity. The cascade design method can be applied iteratively to provide a complete hierarchy of $N$ controllers, satisfactorily addressing the issues of hierarchical design, practicality, and modularity.

For AS stability analysis, consider the sequence of feedback controllers $\Gamma_1(z) = \widehat{K}_1(z), \cdots, \Gamma_{i+1}(z) = \Gamma_i(z) + \widehat{K}_{i+1}(z)$. Let the dynamics of the component controllers $\widehat{K}_i$ be realised in a $n_i$–dimensional state space with $(A_{K_i}, B_{K_i}, C_{K_i}, D_{K_i})$. A state-space realisation for $\Gamma_i$ is then given by

$$\begin{aligned} F_i &= \operatorname{diag}(A_{K_1}, \cdots, A_{K_i}) \\ G_i^T &= \begin{bmatrix} B_{K_1}^T & \cdots & B_{K_i}^T \end{bmatrix} \\ R_i &= \begin{bmatrix} C_{K_1} & \cdots & C_{K_i} \end{bmatrix} \\ S_i &= D_{K_1} + \cdots + D_{K_i}. \end{aligned}$$

Accordingly, and assuming again that the states of inactive controllers are reset to zero, the corresponding closed-loop matrices $\widehat{A}_i \in \mathbb{R}^{n_c \times n_c}$ have the same structure as in (3.7.2) with the new block definitions

$$\widetilde{R}_i = \begin{bmatrix} R_i & 0 & \cdots & 0 \end{bmatrix} \tag{3.7.3}$$

$$\widetilde{F}_i = \operatorname{diag}(F_i, 0, \cdots, 0). \tag{3.7.4}$$

$$\widetilde{G}_i^T = \begin{bmatrix} G_i^T & 0 & \cdots & 0 \end{bmatrix} \tag{3.7.5}$$

## 3.7.1 *Tracking control and bumpless transfer*

The schedule conditioning technique of Section 3.5 is able to address the switched system performance issue satisfactorily when a regulation problem is considered. In

reference-tracking tasks, however, performance can be severely impaired by switching between different controllers. Indeed, some of the controllers may remain idle for a certain number of periods, with their states in a "sleeping" condition. Sudden re-activation of sleeping states may produce "bumps" in the states and outputs, and degrade performance, if the active part of the system has gone through significant changes in the meantime.

The design of bumpless-control techniques has been extensively considered in the literature since at least the 80's ([106]), and is still an active area of research ([9, 216, 110]). However, most of the existing results are not meant for resource-limited applications as those targeted here, and do not comply with the practicality requirements which are at a premium in anytime control. We describe below a simple method for bumpless control that applies to the modular anytime structure in Figure 3.7.2, and that generalises it to tracking problems for set-point references $r(t)$ which vary slowly with respect to the scheduling switches.

We assume in what follows that the input reference is scaled, as customary in tracking problems, in the prefilter block $\Phi_i$ (see Figure 3.7.2), by the steady-state gain of the corresponding closed loop system with controller $\Gamma_i$, $1 \leq i \leq N$. Let $w$ denote the state of the controlled plant, and $x_{K_i}$ be the state of the $i$–th controller component $\widehat{K}_i$. Suppose now that, at some instant in time $t^* T_g$, the $i$–th level controller is active and the system components have reached a steady-state equilibrium under a constant reference $\bar{r}$. Let $\bar{w}$, $\bar{x}_{K_i}$, and $\bar{y} = \bar{r}$ denote the corresponding equilibrium values of the states and output.

Consider the event that, at time $(t^* + 1) T_g$, the execution of the $j$–th level controller is imposed, by either the occurrence of a deadline or a conditioned schedule switch, while the reference remains unchanged from $\bar{r}$. If $j \leq i$ (high-to-low level switching), it can be easily verified that the active parts of the system state remain at $\bar{w}, \bar{x}_{K_1}, \cdots, \bar{x}_{K_j}$. If instead $j > i$ (low-to-high level switching), the evolution of the whole system for $t > t^*$ depends upon the values of the sleeping states at time $t^*$, i.e. $x_{K_k}(t^*)$, $i < k \leq j$. The dynamics of controller states during their idle periods is therefore an important degree-of-freedom in switching control design. Straightforward policies for managing sleeping states, such as e.g. keeping them constant during sleep, may be adequate for the regulation problem, but not for tracking (if the reference changes even slowly during the idle time of a controller component, output bumps will necessarily result at re-activation of that component). It is worthwhile noticing explicitly that zeroing the sleeping states, either instantaneously - as assumed in the previous section - or progressively with a simple and computationally inexpensive dynamics as e.g. in [29], effectively avoid bumps only for zero-regulation problems.

When the active part of the system is at steady state in $t = t^*$ under a constant reference $\bar{r}$, perfectly smooth low-to-high switching would be achieved if and only if the state of the $k$-th component controller ($i < k \leq j$) is re-initialised as

$$(I - A_{K_k}) x_{K_k}(t^* + 1) = B_{K_k} \bar{y} \tag{3.7.6}$$

Observing that $y(t^*) = \bar{y}$ and setting $W_k \triangleq (I - A_{K_k})^{-1} B_{K_k}$, one can rewrite (3.7.6) as

$$x_{K_k}(t^* + 1) = W_k y(t^*). \tag{3.7.7}$$

We replace this re-initialisation formula with the following one

$$x_{K_k}(t^* + 1) = A_{K_k} W_k y(t^* - 1) + B_{K_k} y(t^*), \tag{3.7.8}$$

which is equivalent to (3.7.6) provided that $y(t^* - 1) = y(t^*) = \bar{y}$. This holds true under the hypothesis that the system is at steady-state before the switch. Indeed, using the definition of $W_k$ and Woodbury's matrix inversion lemma, it is straightforward to prove that $\left(A_{K_k} W_k + B_{K_k}\right) = W_k$. If the system is not exactly at steady-state at $t^*$, but the reference is varying slowly, the difference $y(t^* + 1) - y(t^*)$ is small and application of the two formulae will slightly differ.

From a computational viewpoint, this re-initialisation scheme introduces negligible overhead, because matrices $W_k$ can be easily pre-computed, and (3.7.8) needs to be evaluated only once when a low-to-high transition occurs for the reactivated controllers. However, for the sake of AS stability analysis, it is instrumental to consider an equivalent system where all sleeping states are re-initialised at every time instant as

$$x_{K_k}(t) = W_k y(t - 1)$$

(the re-initialisation of states clearly has no effect on the closed-loop system until they remain inactivated). Therefore, the closed-loop matrices $\widehat{A}_i \in \mathbb{R}^{n_c \times n_c}$ have the same structure as in (3.7.2), (3.7.3), and (3.7.4), but with (3.7.5) replaced by

$$\widetilde{G}_i^T = \begin{bmatrix} G_i^T & W_{i+1}^T & \cdots & W_n^T \end{bmatrix}. \tag{3.7.9}$$

In conclusion, the solution of the OSP (ROSP) problem with $\widehat{A}_i$ matrices computed as in (3.7.2), (3.7.3), (3.7.4), and (3.7.9), provides the (robust) optimal switching policy for the intended controller hierarchy $\Gamma_i$, while also guaranteeing AS stability of the bumpless transfer technique (3.7.8).

## 3.8 CASE STUDIES

The control of two mechanical systems will be used to illustrate the application of the proposed technique. In the first example we consider a regulation problem for both a nominal and an uncertain scheduler model. A conditioning Markov chain solving the OSP and ROSP problems, respectively, is obtained which satisfies the 1-step average contractivity condition for AS-stability. The second example illustrates application of $m$-step average contractive solutions for AS-stability to the OSP problem, and the effectiveness of the proposed bumpless transfer technique to track piecewise-constant references. As a third example we stick to the tracking problem, but we show the robustness properties guaranteed by the 2-EISS conditions in Theorem 2.2 in case of an external disturbance. In all cases, the unit delay between the output sampling instant and the application of the control action has been explicitly considered.

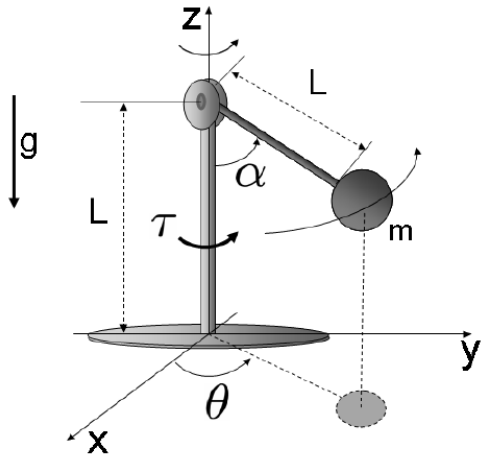


FIGURE 3.8.1: Model of Furuta pendulum with zero offset ([93]).

$$K_1(z) = \frac{31.6(z^2-1.8z+1.1)}{(z-0.1)(z-0.5)} \qquad F_1(z) = 21.28$$
$$K_2(z) = \frac{136.5(z-1.3)(z-4.7\,10^{-3})(z^2-1.4z+0.7)}{(z-0.5)(z+0.3)(z-0.1)(z^2+0.6z+0.8)} \qquad F_2(z) = 0.54$$
$$K_3(z) = \frac{1370.9(z-0.4)(z-0.6)(z-0.2)(z+0.3)(z-4.7\,10^{-3})(z^2-0.7z+0.2)(z^2-0.4z+0.3)}{(z-0.5)^2(z+0.3)^2(z-0.1)^2(z^2+0.6z+0.8)(z^2+3.4z+4.7)} \qquad F_3(z) = 2.73$$

TABLE 3.8.1: Hierarchical controllers and prefilters used in simulations for the Furuta pendulum with zero offset.

### 3.8.1 *One-step contractivity with and without UBB uncertainties on the scheduler*

#### *Plant and control description*

A model of Furuta pendulum with zero offset ([93]) is depicted in Figure 3.8.1. After linearisation about the origin and sampled-time discretisation with sampling time $T = 0.1\,\mathrm{s}$, we obtain the following transfer function from the torque $\tau$ to the angle $\alpha$

$$G(z) = \frac{1.2\,10^{-3}(z+3.7)(z+0.3)}{(z-1)(z^2-1.7z+1)}.$$

We assume measurements to be acquired at the beginning of each period and control inputs to be released at the end, thus the controller is not affected by jitter but experiences a constant unit delay. In order to account for the unit delay, controllers are designed for the transfer function $G(z)\frac{1}{z}$ instead of $G(z)$.

With reference to the connection scheme in Figure 3.7.2 and the table 3.8.1, the first controller $\Gamma_1(z) = K_1(z)$ has been designed to ensure the stability requirement, while the controllers $\Gamma_2(z) = K_1(z) + \widehat{K}_2(z)$ and $\Gamma_3(z) = K_1(z) + \widehat{K}_2(z) + \widehat{K}_3(z)$ are obtained applying LQG design techniques (we recall that $\widehat{K}_i(z) = F_{i-1}(z)K_i(z)$, $i = 2, 3$). Prefilters $\Phi_i(z) = \prod_{l=1}^{i} F_l(z)$ are constants used to adapt the steady-state gain and ensure static requirements.

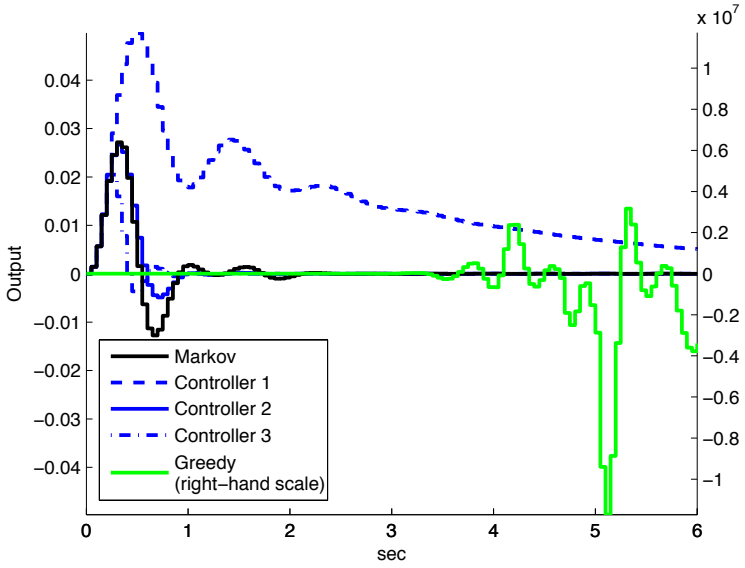


FIGURE 3.8.2: Regulation results for the Furuta pendulum example: output signals.

### *Stochastic description of the computation platform*

For the sake of brevity, we consider a deterministic WCET model of controller execution time, so that, in the light of Remark 3.2, the process $\{\varphi(\cdot)\}$ is described by a FSHIA-MC. The transition probability matrix of the nominal scheduler process is

$$P_\varphi = \begin{bmatrix} 0.2744 & 0.342 & 0.3836 \\ 0.0881 & 0.3443 & 0.5676 \\ 0.0204 & 0.2097 & 0.7699 \end{bmatrix}. \tag{3.8.1}$$

The corresponding steady-state probability distribution is given by $\overline{\pi}_\varphi = [1/20, 5/20, 14/20]$.

In the application of the contractivity condition for solving (R)OSP problems, a suitable choice of the adopted matrix norm can considerably help. In our examples we use the method proposed by [80] to choose a vector norm $|x| = |Tx|_2$ and induced matrix norm $\|A\| = \|TAT^{-1}\|_2$, where $T$ is a non-singular matrix satisfying the condition that there exists $\widehat{A}_i$ such that $\|T\widehat{A}_i T^{-1}\|_2 < 1$ (this condition generalises directly to the multistep problem).

### *Simulation results*

For the Furuta pendulum a solution to the 1-step OSP with QoC index vector $c_d = [1, 4, 9]$ is $\overline{\pi}_\varsigma = [0.006, 0.972, 0.022]$, resulting in $\overline{\pi}_{\varphi|\varsigma} = [0.056, 0.929, 0.015]$ and $\mathcal{J}(\overline{\pi}_{\varphi|\varsigma}) \approx 3.91$.

In Figure 3.8.2, the output signals obtained by different controllers are shown, corresponding to regulation errors for initially perturbed angular velocity $\pi/10$ rad/s

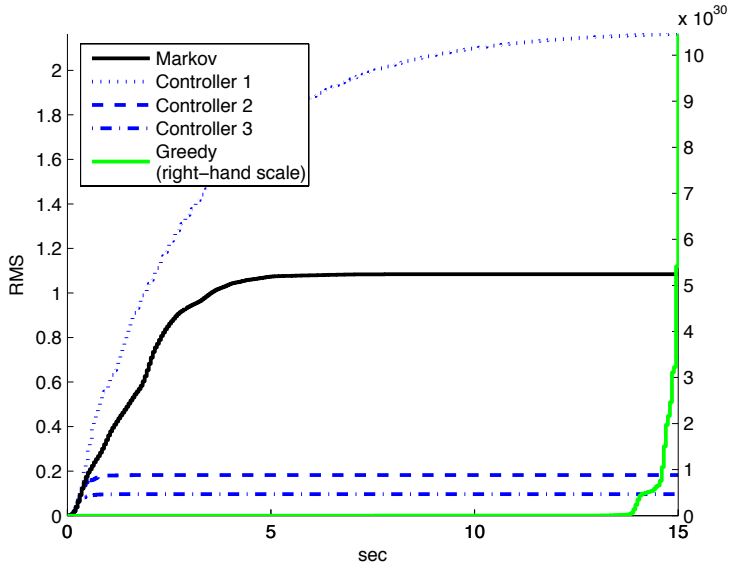


FIGURE 3.8.3: Regulation results for the Furuta pendulum example: mean values of RMS errors (for one thousand runs) of the closed loop system with different control schedulings.

with respect to an equilibrium velocity of 3.71 rad/s, and an equilibrium pendulum displacement of $\pi/4$ rad from the vertical. It can be noticed in this case that the application of the greedy switching policy causes instability.

In Figure 3.8.3, the *Root Mean Squares* (RMSs) of the regulation error are shown for different controllers corresponding to the same perturbation. Data plotted represent the average RMS error of one thousand simulations performed for different independent realisations of the scheduler and conditioning processes. Plots labeled Controller 1, 2, and 3, corresponding to results obtained without switching, are reported for reference. Notice the performance increase which would be obtained by the (unschedulable) higher-level controllers. The average behaviour of the greedy switching policy is clearly unstable. On the same Figure 3.8.3, the plot labeled "Markov" shows the average RMS error obtained by the stochastically conditioned scheduler. This example shows how the proposed stochastic switching policy ensures the AS-stability of the closed loop system (which is not guaranteed by the greedy policy), while it obtains a definite performance increase (of the order of 50%) with respect to the conservative scheduling policy consisting in using only Controller 1 (see Figure 3.8.3.

Consider now the case that the actual transition probability matrix of the scheduler is affected by UBB uncertainties. In practical cases, a rough description of the uncertainty may be available in terms of a bounding box centred in the nominal value $P_\varphi$ and described by an additive uncertainty matrix $\Delta P_\varphi$, i.e. $\tilde{\mathcal{P}} = \{\tilde{P}_\varphi = P_\varphi + \delta P_\varphi \mid \delta P_\varphi \in \Delta P_\varphi\}$ where $\Delta P_\varphi = \left(\rho_{ij}\right)_{N\times N}$. Notice however that $\tilde{\mathcal{P}}$ is not in general a set of stochastic matrices, so that a polytopic description $\mathcal{P}$ of the possible perturbations is obtained as the intersection of $\tilde{\mathcal{P}}$ with the set of stochastic matrices (this can be done e.g. by

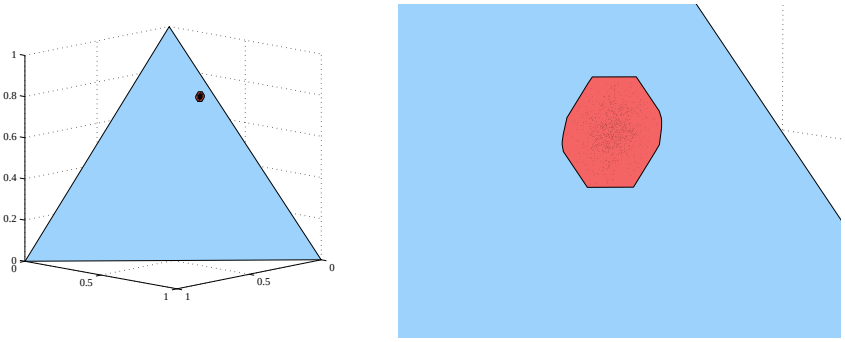

FIGURE 3.8.4: Samples of steady-state probability distributions in $\mathcal{Q}$ for $P \in \mathcal{P}$ and the outer approximating polytope $\Pi_{\mathcal{P}}(8)$. On the right, the approximating polytope is located on the canonical 3-D simplex.

using the Polytope Library of the `Multi-Parametric Toolbox` [129]). In our example, taking $P_\varphi$ as in (3.8.1) and $\rho_{1j} \in [-0.025, 0.025]$, $\rho_{2j} \in [-0.02, 0.02]$, and $\rho_{3j} \in [-0.01, 0.01]$, a matrix polytope $\mathcal{P}$ is obtained which has 216 vertices (not reported here). Using the algorithm presented in Section 3.6, an approximating polytope of the steady-state probability set $\mathcal{Q}$ with 30 vertices is obtained after 8 iterations (see Figure 3.8.4)

A solution to the ROSP problem for this example with QoC index vector $c_d = [1, 4, 9]$ is $\overline{\pi}_\varsigma = [10^{-6}, 0.992, 0.007]$, resulting in $\mathcal{J}(\overline{\pi}_{\varphi|\varsigma}) \approx 3.88$ (i.e., slightly worse than the QoC index in the nominal case).

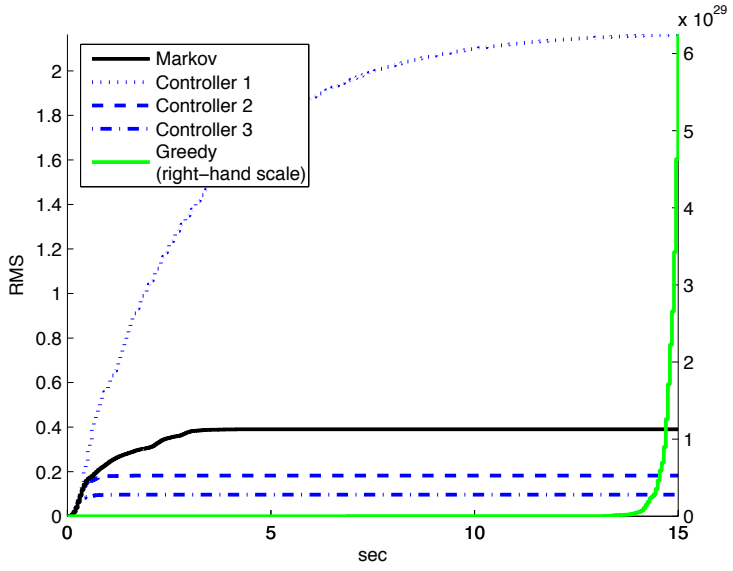


FIGURE 3.8.5: Regulation results for the Furuta pendulum example and scheduler transition probability matrix affected by UBB uncertainties: mean value of the RMS errors (for 2500 runs) of the closed loop system with different schedules.

In Figure 3.8.5, the RMSs of the regulation error for different closed loop controllers is shown. The perturbed initial conditions are the same of the previous case (with known scheduler). The RMS regulation error is obtained as the mean of the RMS errors for 2500 simulations, whereby the scheduler stochastic description is obtained by randomly choosing 50 transition probability matrices $\tilde{P}_\varphi \in \mathcal{P}$ and considering 50 different sample realisations for each matrix.

The closed loop system driven by the greedy switching policy still appears unstable. The example shows results very close to the ones obtained for the nominal scheduler.

### 3.8.2 *Multi-step contractivity and tracking*

#### *Plant and control description*

The Translational Oscillator/Rotational Actuator (TORA) system in Figure 3.8.6 (see [43] for further details) can be described as follows

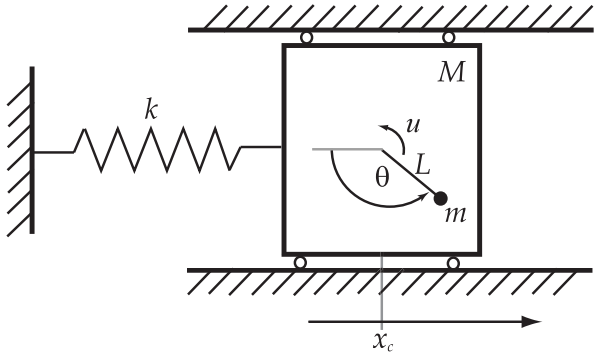


FIGURE 3.8.6: Model of a Translational Oscillator/Rotational Actuator (TORA) system ([43]).

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{x}_c \end{bmatrix} = \frac{1}{\Delta(\theta)} \begin{bmatrix} m+M & -mL\cos\theta \\ -mL\cos\theta & I+mL^2 \end{bmatrix} \begin{bmatrix} u+d \\ mL\dot{\theta}^2\sin\theta - \kappa x \end{bmatrix}$$
$$y = \theta + n$$

where $\Delta(\theta) = (I+mL^2)(m+M) - m^2L^2\cos^2\theta > 0$, $M$ is the mass of the translational oscillator, $m$ and $I$ are the mass and the inertia of the rotational actuator located at a distance $L$ from the centre of rotation, $\kappa$ is the stiffness of the spring, $\theta$ is the angle of the actuator, $x_c$ is the horizontal displacement of the oscillator, $u$ is the control torque, $d$ is the torque disturbance and $n$ is the measurement error. The previous parameters are assumed to take the following values: $m = 1\,\text{kg}$, $M = 5\,\text{kg}$, $L = 0.1\,\text{m}$, $I = 0.01\,\text{kg}\,\text{m}^2$ and $\kappa = 20\,\text{N/m}$. We remark that the TORA is a planar system where the rotating arm is implanted above the oscillating mass. Hence, the gravity force is orthogonal to the plan.

$$\begin{array}{l|l} K_1(z) = \frac{2.0895(z-0.75)}{(z+0.3761)} & F_1(z) = 0.38 \\ K_2(z) = \frac{0.8(z-0.4)}{(z+0.6)} & F_2(z) = 1.79 \\ K_3(z) = \frac{0.73(z^2-0.76z+0.2228)}{(z+0.3)^2} & F_3(z) = 1.29 \end{array}$$

TABLE 3.8.2: Hierarchical controllers and prefilters used in simulations for the TORA system (AS-stability).

After linearisation about the origin and sampled-time discretisation with sampling time $T = 0.1\,\mathrm{s}$, we obtain the following transfer function from $u$ to $\theta$

$$G(z) = \frac{0.27266(z+1)(z^2-1.967z+1)}{(z-1)^2(z^2-1.964z+1)}.$$

In this section we consider the disturbance $d$ and the noise $n$ to be null and we focus on the AS-stability through the multi-step contractivity condition shown in Section 3.5.2 and the tracking problem discussed in Section 3.7.1.

We make the same assumptions on the unit delay made before. The expressions of the hierarchical controllers used in this section are reported in table 3.8.2. The first (simplest) controller ensures only the stability requirement, while other controllers are designed by hand to achieve performance enhancements with minimal complexity increase.

### *Stochastic description of the computation platform*

We adopt here the same model for the computation platform as in the previous example. In particular we assume the process $\{\varphi(\cdot)\}$ to be a FSHIA-MC whose transition probability matrix is given by (3.8.1).

### *Simulation results*

Figure 3.8.7 and Figure 3.8.8 report plots to illustrate application of the proposed methodology to the TORA example. A 4-steps lifted version of the OSP problem with index vector $c_d = \otimes_{i=1}^{4}[1,4,9]$ admits a solution resulting in $\mathcal{J}(\overline{\widetilde{\pi}}_{\overline{\varphi|\varsigma}}) \approx 382.49$, according to which the conditioning sequence of controllers is a concatenation of the 81 possible combinations of length 4 of the three controllers. The steady state conditioning probability distribution $\overline{\widetilde{\pi}}_{\overline{\varsigma}} \in (0,1)^{81}$ is not reported here; it is however worth noticing that the two controller sequences $\Gamma_2-\Gamma_2-\Gamma_2-\Gamma_2$ and $\Gamma_2-\Gamma_2-\Gamma_2-\Gamma_3$ turn out to be by far the most likely, being used in more than the 51.5% and the 40.5% of cases, respectively. Results of simulations of the different controllers and scheduling policies are reported in Figure 3.8.7-a, for a regulation problem from perturbed initial angular velocity of 0.1 rad/s.

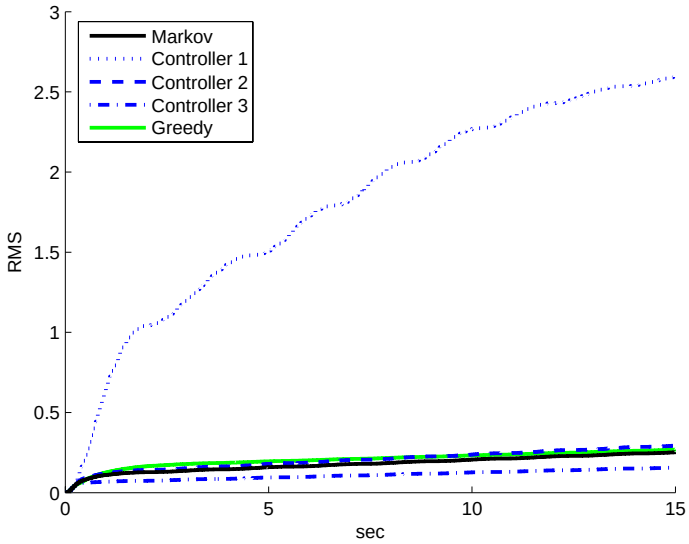


FIGURE 3.8.7: Regulation results for the TORA example: mean values of RMS errors (for one thousand runs) of the closed loop system with different control schedulings.

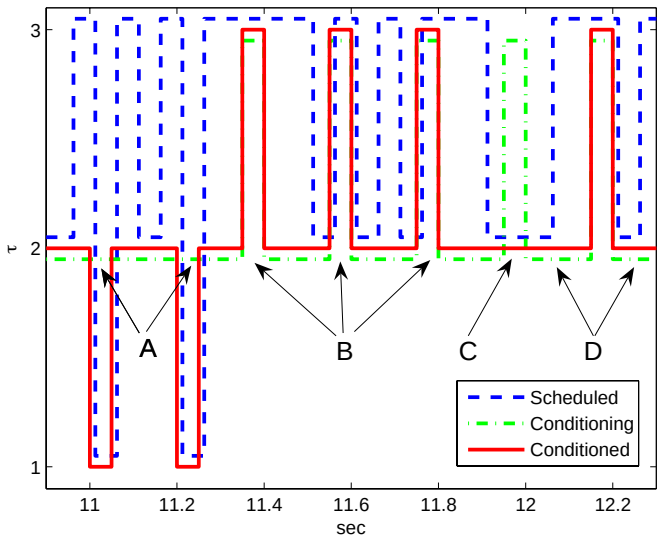


FIGURE 3.8.8: Sample realisation of the scheduler, conditioning and conditioned stochastic processes for the TORA example. At instants labeled as *A* the conditioning process (i.e., the switching policy) calls for the execution of Controller 2, while the scheduler gives time only for the execution of Controller 1, hence the conditioned process executes Controller 1. Similarly, at instant *C*, the switching policy call for Controller 3 is overridden and Controller 2 is executed. At instants tagged as *D*, the opposite occurs, i.e. the conditioning process forces execution of Controller 2, while the scheduler could allow Controller 3 to be executed. Finally, at instants labeled as *B*, the switching policy and the scheduler agree on the execution of Controller 3.

The RMS performance plots in Figure 3.8.7 show that the greedy policy (in this particular case) does not lead to divergence. In Figure 3.8.8, sample realisations of scheduler, conditioning and conditioned processes are depicted: the prevalence of the preferred patterns is apparent.

Finally, results of application of the proposed technique for a tracking control problem for the TORA example are reported in Figures 3.8.9 and 3.8.10. The reference to be tracked by the angular position is a piecewise constant signal of amplitude $\pi/4$ rad, period 10 s and pulse width of 30%.

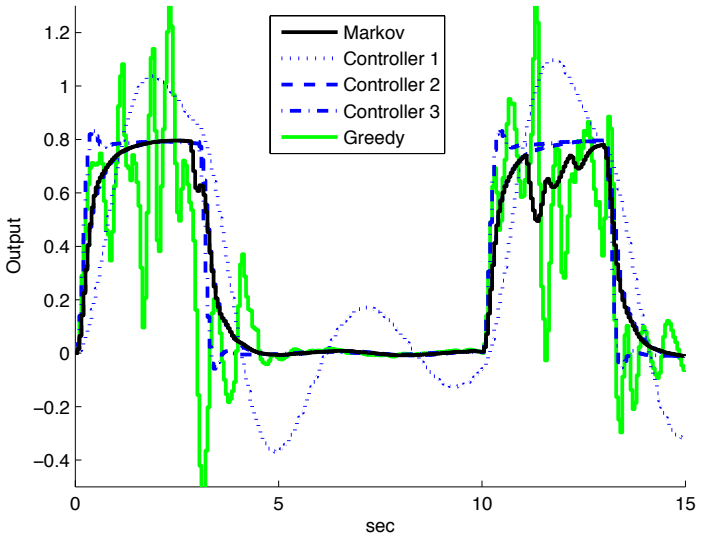


FIGURE 3.8.9: Tracking results for the TORA system: output signals.

The comparison of RMS errors shows that the simple bumpless switching technique proposed in Section 3.7.1 ensures good performance, as shown in both Figures 3.8.9 and 3.8.10. The RMS performance of the conditioned switching policy is better than both the conservative and the greedy approaches and is quite close to the results given by using always Controller 2, which is not a feasible choice.

## 3.8.3 *Robustness w.r.t. external perturbations*

### *Plant and control description*

In this section we illustrate the robustness properties guaranteed by the 2-EISS conditions in Theorem 2.2 on a tracking problem for the model of the TORA system introduced in the previous section. The expressions of the hierarchical controllers and prefilters used in this section are reported in table 3.8.2. We recall that, according to the connection scheme in Figure 3.7.2, the actual controllers and prefilters are given by $\Gamma_i(z) = \sum_{l=1}^{i} \widehat{K}_l(z)$ and $\Phi_i(z) = \prod_{l=1}^{i} F_l(z)$.

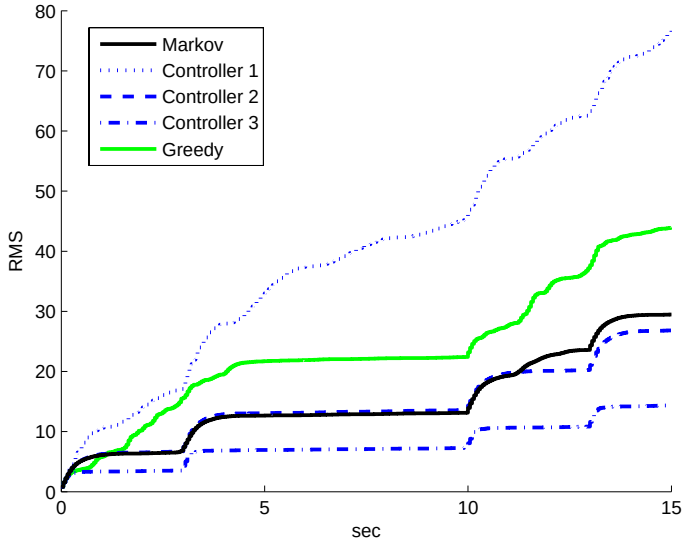


FIGURE 3.8.10: Tracking results for the TORA system: RMS errors (one thousand runs).

$$\begin{array}{l|l} K_1(z) = \frac{3.04(z-0.97)}{z+0.9} & F_1(z) = 0.0480 \\ K_2(z) = \frac{-0.021(z-2)}{z-0.76} & F_2(z) = 2.8146 \end{array}$$

TABLE 3.8.3: Hierarchical controllers and prefilters used in simulations for the TORA system (2-EISS).

The controller $\Gamma_1(z)$ is designed to ensure only stability requirement, while the second controller $\Gamma_2(z)$ to enhance performance in terms of rise time and settling time (see Figure 3.8.11 for a graphical comparison). The prefilters are used to adapt the steady-state gain and ensure static requirements.

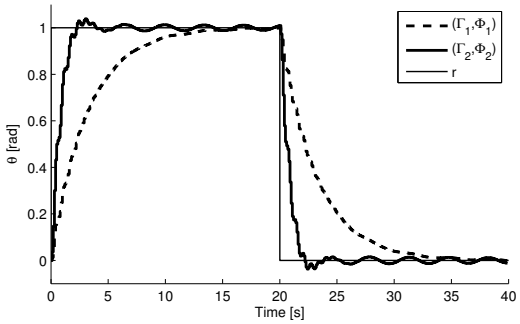


FIGURE 3.8.11: Outputs of the TORA system in closed loop with $(\Gamma_1, \Phi_1)$ and $(\Gamma_2, \Phi_2)$ for a square wave reference $r$ of period 40 s, duty cycle 50% and amplitude 1 rad.

### *Stochastic description of the computation platform*

In this case study we have two controllers $\Gamma_1$ and $\Gamma_2$, hence $J_\varphi \triangleq \{1, 2\}$. We also assume that the available computation time can take four different values, hence $J_\sigma \triangleq \{1, \dots, 4\}$. We assume the FSH-MC $\{\sigma(\cdot)\}$ to be described by the following transition probability matrix

$$P = \begin{bmatrix} 0.5 & 0.1 & 0.4 & 0 \\ 0.2 & 0.1 & 0.5 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{bmatrix}$$

and the following initial probability distribution $\pi_{\sigma 0} = \frac{1}{4}[1,\ 1,\ 1,\ 1]$. The stationary probability distributions of the execution times $T^1$ and $T^2$ are given by the row vectors $\pi_{T^1} = [1,\ 0,\ 0,\ 0]$ and $\pi_{T^2} = [0,\ 0.33,\ 0.5,\ 0.17]$ respectively. Therefore, the matrix $L$ in (3.3.2) is

$$L = \begin{bmatrix} 1 & 0 \\ 0.67 & 0.33 \\ 0.17 & 0.83 \\ 0 & 1 \end{bmatrix}.$$

The controller actually executed at each time step $k$ is the largest possible one within the available computation time. That is, if the available computation time $\sigma(k)$ at step $k$ is such that $\sigma(k) \geq T^2(k)$, then $\varphi(k) = 2$, otherwise $\varphi(k) = 1$. We assume here that $\sigma(k) \geq T^1(k)$ for every $k$.

### *Simulation results*

The closed–loop SJLS with the computation platform described in the previous sections satisfies the LMI conditions of Theorem 2.2, thus it turns out to be 2-EISS. For simulation purposes we have considered a tracking problem with a square wave reference of period 40 s, duty cycle 50% and amplitude 1 rad. We assume also that the input disturbance $d$ and the measurement error $n$ affecting the SJLS (see the scheme in Figure 3.7.2) are bounded as $|d| \leq 0.05\,\mathrm{N\,m}$ and $|n| \leq 0.05\,\mathrm{rad}$. Figure 3.8.12 shows a simulation run for the TORA system in closed loop with $(\Gamma_1, \Phi_1)$, with $(\Gamma_2, \Phi_2)$ and with the anytime control (SJLS). The vertical black segments of the graph on the bottom represent the instants where $(\Gamma_1, \Phi_1)$ is executed, that is the steps $k$ where $\varphi(k) = 1$. The empty spaces among those segments represent, instead, the steps $k$ where $\varphi(k) = 2$. Simulations show a bounded response in presence of the bounded disturbances, which is typical of an ISS behaviour.

Figure 3.8.13 illustrates the first 4 seconds of temporal evolution of a sample path for the processes $T^1(\cdot)$, $T^2(\cdot)$ and $\sigma(\cdot)$. The available ($\sigma(\cdot)$) and required ($T^1(\cdot)$, $T^2(\cdot)$) computation time range in the discrete set $J_\sigma = \{1, \dots, 4\}$. Whenever $\sigma(k) \geq T^2(k)$, it holds that $\varphi(k) = 2$, otherwise $\varphi(k) = 1$. The process $\varphi(\cdot)$ is represented at the bottom of the figure as a bichromatic thick line: dark grey segments accounts for the activation periods of $(\Gamma_2, \Phi_2)$, light grey segments indicate those of $(\Gamma_1, \Phi_1)$.

This example shows the applicability of our result and its ability to formally guarantee robustness properties.

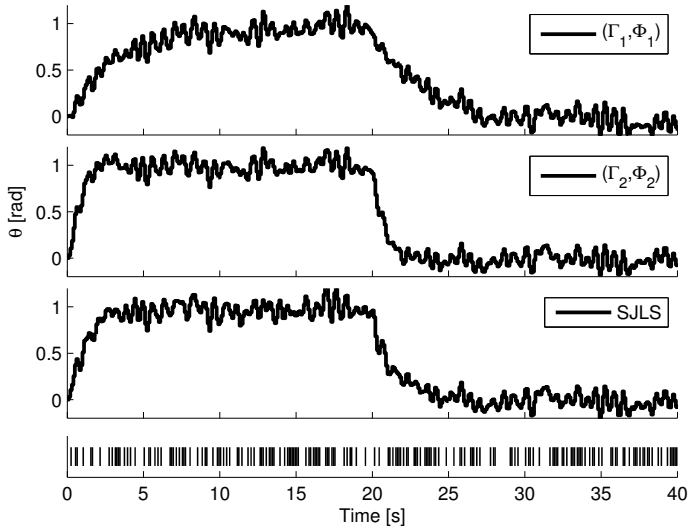


FIGURE 3.8.12: Outputs of the TORA system in closed loop with $(\Gamma_1, \Phi_1)$, $(\Gamma_2, \Phi_2)$ and with the anytime control for a square wave reference of period 40 s, duty cycle 50% and amplitude 1 rad. The system is affected by bounded input disturbance $d$ and measurement error $n$ with $|d| \leq 0.05$ N m and $|n| \leq 0.05$ rad. The bottom graph portraits the activation instants of $(\Gamma_1, \Phi_1)$.

## 3.9 CONCLUSIONS AND RECENT LITERATURE

In this Chapter we considered the problem of scheduling the execution of several control tasks on a multi-tasking, preemptive RTOS, according to an Anytime Control philosophy. The main contributions are:

*A structural approach.* We proposed an approach to the synthesis of anytime controllers which incorporates the flexibility requirements in the very architecture of the controllers. To this aim we introduced three fundamental requirements to satisfy: practicality, hierarchical design and modularity.

*A compositional design technique.* Even if we could not provide a general synthesis methodology, we were able to present a simple design technique ensuring robustness to numerical errors as well as reduction of computational burden w.r.t. worst case execution time design. A bottom-up approach was described, where closed-loop performance are used to drive the practical design of a compositional hierarchy of controllers complying with the aforementioned requirements.

*A model for the scheduler.* We introduced a stochastic model of the scheduler, which generalises and encompasses, as special cases, both FSH-MCs and i.i.d. processes.

*A stabilising stochastic switching law.* Given the scheduler and the set of controllers, we formulated a linear programming problem whose solutions provide stochastic

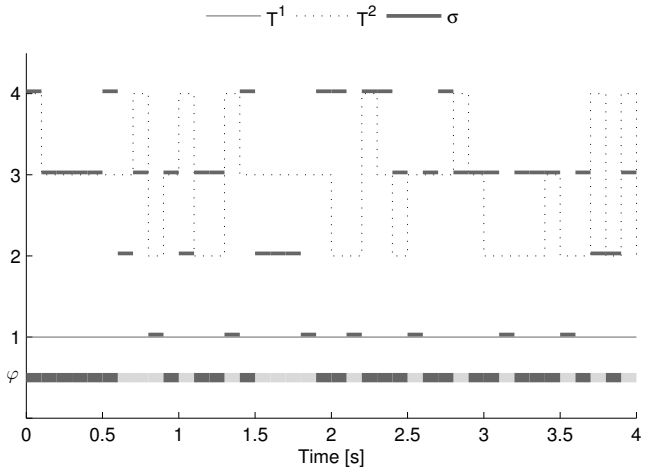


FIGURE 3.8.13: First 4 seconds of temporal evolution of a sample path for the processes $T^1(\cdot)$, $T^2(\cdot)$ and $\sigma(\cdot)$. Whenever $\sigma(k) \geq T^2(k)$, it holds that $\varphi(k) = 2$, otherwise $\varphi(k) = 1$. The bottom graph represents the process $\varphi(\cdot)$: dark grey segments accounts for the activation periods of $(\Gamma_2, \Phi_2)$, light grey segments indicate those of $(\Gamma_1, \Phi_1)$.

switching laws that condition the scheduler in such a way that the resulting switched system is stable in a probabilistic sense.

*Robust guarantees.* We considered both the case of a scheduler model affected by unknown but bounded uncertainties and the case of a system affected by a locally bounded exogenous disturbance. Accordingly, we employed two very different definitions of robustness and we provided corresponding guarantees.

*Bumpless-transfer tracking.* We presented a practical bumpless-transfer technique to address the problems related to the re-activation of idle controllers in a tracking context. Despite its simplicity and computational inexpensiveness, such approach proved very effective in improving the performance while still guaranteeing stability.

After the publications of a series of papers ([99, 83, 100, 170, 101, 171]) where the results presented in this chapter first appeared, other contributions to the anytime control problem came to enrich the literature.

The very recent paper [204] assumes, without giving details, the existence of an hierarchy of controllers similar to that presented here, but extends the anytime formulation to continuous-time systems.

Another stream of interesting works ([172, 173, 174, 65, 145]) takes a different approach to the anytime control which is modular but not structural in nature. They all revolve around the concept of sequence-based anytime control: at each time step, instead of computing just one control value, the algorithm uses a model of the system to compute a sequence of future control values for the number of steps allowed by the available computation time. The sequence is stored in a buffer, that is used to provide a control value in those steps where the computation resource is not available. Different

policies are considered for the management of the buffer, when a new sequence is computed, and for the control, when the buffer is empty. The evolution of the size (number of steps) of the sequence is described in [172] as an i.i.d. process or as a process equal to those presented here in (2.2.3)-(2.2.4); in [174] it is described as a Markov chain. Subsequent works employ the sequence-based anytime control in a networked environment with event-triggered sensor updates with single [173] or multiple control laws [65] and with a realistic model of network on both sensor and actuator channels [145]. The advantage of the sequence-based anytime control is its applicability to nonlinear systems, even if with important restrictions.

In [102] the author considers a multi-input linear system and proposes an LQR and a constrained receding-horizon optimal control. For the former, the most important component of control vector is computed first and the other components are progressively computed according to the available time. For the latter, an approach very similar to the sequence-based anytime control is adopted. Finally, a worth mentioning, strictly related work is [8].

# 4

# CONTROL–ORIENTED OPTIMAL BANDWIDTH ALLOCATION

## 4.1 INTRODUCTION

In embedded control systems a control loop is usually implemented as a software application, called *control task* (or simply *task*). Upon the periodic arrival of new sensor data, a control task executes *a job* in order to compute the control value and deliver it to the actuators. An important requirement motivated by cost and engineering reasons is that several control tasks share the same computing hardware.

The control performance obtained by the computer implementation of a controller is strongly related to the delays introduced in the computation. When a control task shares the CPU with other tasks, its jobs incur two types of delays: the first one caused by their own execution time and the second one by the time spent awaiting their turn to be scheduled and take control of the CPU (scheduling interference). Such delays are often time–varying and undermine the typical assumptions of digital control design: regular sampling and negligible (or at least constant) input-to-output latency for control computation. This justifies its modelling as a stochastic process with potentially long tails in the probability distributions [85].

An important design problem that emanates from this approach is how to choose activation periods and scheduling parameters in order to maximise some "global" control performance metrics, which means how to optimally share the CPU time between different control tasks.

Following the steps of other authors [186, 163, 211, 212], we seek the choice of scheduling parameters for a set of control task that maximises their global control performance. However, in our setting, the computation time of the tasks is not characterised by

a single number (the worst case), but by a probability distribution. Based on the observation that occasional delays or data losses can be tolerated by many control systems [47, 85, 151], we consider a model in which a job executing beyond its deadline is cancelled without closing the control loop (the same approach used in networked control to deal with packet dropouts by [157]). A key element of our strategy is the use of a scheduling algorithm known as the Constant Bandwidth Server (CBS) [2]. Server-based scheduling for control tasks is proposed by other authors [6] in a deterministic setting. Our specific motivation is the possibility to control the fraction of CPU time (bandwidth) allocated to each task, and hence the probability that a job of the task will miss its deadline and be cancelled.

In view of these choices, our problem bears a close resemblance to the allocation of bandwidth to channels with stochastic performance, well known in the communication community [179]. The performance metric is in our case the control performance, or Quality of Control (QoC), and the way it is evaluated is an additional contribution of this work. Assuming that the task controls a plant affected by process noise, we express the QoC as the trace of the steady state covariance of the system's state. This way, the control design can be made using the classical machinery of digital control, and the choice of the sampling period by the well known rules-of-the-thumb developed in decades of scientific literature and industrial practice. Then by assigning a bandwidth, we control the probability of deviating from the design assumption, and hence the QoC degradation. In particular, we show the functional dependence between this task level QoC metric and the bandwidth allocated to the task. By consolidating the QoC of the different control loops into a global QoC metric, we set up an optimisation problem where the values of bandwidth assigned to the tasks are decision variables. The most important contribution of the chapter hinges on a few theoretical results on the properties of the optimal solution that exploit the specific structure of the problem and lead to a very efficient numeric algorithm.

## 4.2 PLATFORM MODEL

We consider a scenario where a set of real-time tasks $\mathcal{T} = \{\tau_i\}\ i \in \{1, \ldots, n\},\ n \in \mathbb{Z}_{>1}$ share the same computation platform. Each task $\tau_i$ is used to control a different system $\Sigma_i$ (see Section 4.3). A control task $\tau_i$ consists of a stream of jobs $J_i(k)$, $k \in \mathbb{Z}_{\geq 0}$. Each job $J_i(k)$ is activated at time $r_i(k)$ and finishes at time $f_i(k)$ after executing for a time $c_i(k)$. Job $J_i(k)$ is also characterised by a deadline $d_i(k)$, that is respected if $f_i(k) \leq d_i(k)$ and is missed if $f_i(k) > d_i(k)$. We focus on *periodic* tasks with *task period* $T_i$ (i.e. $r_i(k+1) = r_i(k) + T_i$), where each activation time is also the deadline of the previous instance (i.e. $d_i(k) = r_i(k+1)$). The task is said *hard* real–time if all deadlines are met, and is said *soft* real–time if deadlines are met with a given probability. In this chapter we focus on soft real-time implementation of control tasks.

### 4.2.1 Scheduling algorithm

In order to allocate the CPU to the different tasks, we adopt the Constant Bandwidth Server (CBS) scheduler [2]. The CBS allows us to assign a *bandwidth* $B_i$ to each task sharing the CPU. A task receiving bandwidth $B_i$ can be thought of as executing on a processor whose speed is a fraction $B_i$ of the speed of the actual processor. The total allocated bandwidth cannot exceed 100%:

$$\sum_{i=1}^{n} B_i \leq 1. \tag{4.2.1}$$

A fundamental property of the CBS is the *temporal isolation*, meaning that the ability for a task to respect a temporal constraint only depends on its execution requirement and on its allocated bandwidth and is independent of the behaviour of the other tasks in the system.

We remark that this scheduling technology is currently implemented in a popular real–time patch of the Linux Kernel[1] and is likely to become mainstream in the future releases.

### 4.2.2 QoS model for the control tasks

A control task $\tau_i$ produces, at its $k$-th activation, a job $J_i(k)$ that takes a sample $y_i(k)$ of the output of the system $\Sigma_i$ and produces a control value $u_i(k)$. The control value $u_i(k)$ is held constant for the entire period $T_i$ (ZoH policy).

To construct a QoS model that can be used in the control design, we need to know the instant $s_i(k)$ when $y_i(k)$ is sampled (*sampling time*) and the instant $v_i(k)$ when $u_i(k)$ is released (*release time*). As shown next, this is possible by the combination of an appropriate model of computation with a CBS scheduler.

#### Model of computation

Despite the periodical activation of the task, the collection of input samples and the release of the computation results are affected by several sources of timing fluctuations (jitter) and computation time variability. To address this problem, we follow the time–triggered model of computation [121] and we introduce a set of rules (a model of computation) that specify the possible sampling times $s_i(k)$ and the release times $v_i(k)$:

*Rule 1:* If a job $J_i(k)$ finishes *before* its deadline (i.e. $f_i(k) < d_i(k) = r_i(k) + T_i$), the release of the output $u_i(k)$ is deferred to the end of the period (i.e. $v_i(k) = r_i(k) + T_i$ is forced).

*Rule 2:* If a job $J_i(k)$ *can not* finish *before* its deadline, it is cancelled and a new job $J_i(k+1)$ is activated without updating the control value (i.e. $u_i(k) = u_i(k-1)$).

*Rule 3:* The output $y_i(k)$ is sampled at the same time the control $u_i(k-1)$ is released (at instant $s_i(k) = v_i(k-1)$).

[1] `http://gitorious.org/sched_deadline/pages/Home`

The adoption of these rules allows us to assume a fixed delay between sampling and actuation (assumed equal to the period $T_i$) without any jitter or cumulated delay whenever the task finishes its job before the deadline, with the control value held constant when it does not.

## 4.3 CONTROL PROBLEM

Our goal is to control a number of independent linear systems $\Sigma_i$, with $i = 1, \dots, n$, sharing the same CPU. Each system $\Sigma_i$ is assumed to be completely observable and controllable, and is described by the discrete-time dynamic equations:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + F_i u_i(k) + w_i(k) \\ y_i(k) &= C_i x_i(k) \end{aligned} \tag{4.3.1}$$

where $x_i \in \mathbb{R}^{n_{x_i}}$ represents the system state, $u_i \in \mathbb{R}^{m_i}$ the control input, $w_i \in \mathbb{R}^{n_{x_i}}$ a zero-mean white Gaussian noise term and $y_i \in \mathbb{R}^{p_i}$ the output variables. One step transition for this discrete time system refers to the evolution of the system step across one temporal unit, which is equal to the task period $T_i$.

In the light of the model of computation introduced in Section 4.2.2, the system (4.3.1) becomes

$$\begin{bmatrix} x_i(k+1) \\ \zeta_i(k+1) \end{bmatrix} = \begin{bmatrix} A_i & F_i \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_i(k) \\ \zeta_i(k) \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} u_i(k) + \begin{bmatrix} w_i(k) \\ 0 \end{bmatrix}, \tag{4.3.2}$$

where the additional state variable $\zeta_i \in \mathbb{R}^{m_i}$ is used to store the previous control value. Such system is controlled by the stabilising dynamic controller:

$$\begin{aligned} z_i(k+1) &= H_i z_i(k) + K_i y_i(k) \\ u_i(k) &= N_i z_i(k) + G_i y_i(k) \end{aligned} \tag{4.3.3}$$

where $z_i \in \mathbb{R}^{n_{z_i}}$.

In the considered scenario, the derivation of the output $y_i(k)$ is a time-varying computation demanding activity (e.g., the sample could be derived from processing an image frame), therefore the computation time $\{c_i(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ of job $J_i(k)$ can be modelled as a stochastic process.

A job $J_i(k)$ could occasionally terminate beyond its deadline and be cancelled. As a result, if we define the augmented state $\tilde{x}_i = [x_i^T, \zeta_i^T, z_i^T]^T \in \mathbb{R}^{n_{\tilde{x}_i}}$ with $n_{\tilde{x}_i} = n_{x_i} + n_{z_i} + m_i$, the system closed loop evolution switches between $\tilde{x}_i(k+1) = A_{c_i}\tilde{x}_i(k) + \omega(k)$, when $J_i(k)$ finishes before the deadline, and $\tilde{x}_i(k+1) = A_{o_i}\tilde{x}_i(k) + \omega(k)$, when it is cancelled, where $\omega(k) \triangleq [w(k)^T, 0]^T \in \mathbb{R}^{n_{\tilde{x}_i}}$ and

$$A_{c_i} = \begin{bmatrix} A_i & F_i & 0 \\ G_i C_i & 0 & N_i \\ K_i C_i & 0 & H_i \end{bmatrix}, \; A_{o_i} = \begin{bmatrix} A_i & F_i & 0 \\ 0 & I_{m_i} & 0 \\ 0 & 0 & I_{n_{z_i}} \end{bmatrix}. \tag{4.3.4}$$

The matrix $A_{c_i}$ is associated with the nominal closed loop evolution and is Schur-stable[2], while $A_{o_i}$ could be not.

**Example 4.1.** We start here a simple example that will be developed throughout the chapter with the aim to elucidate some key aspects of the proposed method.

Let us consider the following (randomly generated) system in the form (4.3.1):

$$\begin{aligned} x_i(k+1) &= \begin{bmatrix} 1.0077 & -0.0008 \\ 0.0062 & 1.0154 \end{bmatrix} x_i(k) + \begin{bmatrix} 0.0023 \\ 0.0189 \end{bmatrix} u_i(k) + w_i(k) \\ y_i(k) &= \begin{bmatrix} 0.4957 & 0.2867 \\ 0.7671 & 0.7342 \end{bmatrix} x_i(k) \end{aligned} \tag{4.3.5}$$

having two unstable poles in 1.01 and 1.02. The LQG regulator in the form (4.3.3) is given by:

$$\begin{aligned} z_i(k+1) &= \begin{bmatrix} 0.04783 & -0.05361 & 0.002259 \\ -0.05404 & 0.06377 & 0.01888 \\ 9.3 & -3.6 & -0.04591 \end{bmatrix} z_i(k) + \begin{bmatrix} 4.612 & -1.729 \\ -4.761 & 3.155 \\ 0 & 0 \end{bmatrix} y_i(k) \\ u_i(k) &= \begin{bmatrix} 9.3 & -3.6 & -0.04591 \end{bmatrix} z_i(k) \end{aligned} \tag{4.3.6}$$

being $G_i = 0$. The corresponding closed and open loop matrices in (4.3.4) can be readily obtained.

## 4.4 OPTIMISATION PROBLEM

### 4.4.1 *Problem formulation*

Let us assume the computation time $\{c_i(k)\}_{k\in\mathbb{Z}_{\geq 0}}$ of the $i$-th task be described by a random process and let $\mathcal{E}_i(k)$ denote the event "job $J_i(k)$ finishes before the deadline". As a consequence of the temporal isolation property, the event $\mathcal{E}_i(k)$ is independent of any event $\mathcal{E}_j(k)$ with $j \neq i$. Furthermore, in view of the cancellation policy described in Rule 2 above, there is no accumulation of delays between jobs. Therefore, for any $\overline{k}$, the event $\mathcal{E}_i(\overline{k})$ is independent of any event $\mathcal{E}_i(k)$ for $k \neq \overline{k}$ and the probability of meeting the deadline is given by (see [87]):

$$\mu_i \triangleq \mathsf{P}\left\{\mathcal{E}_i(k)\right\} = \mathsf{P}\left\{\frac{c_i(k)}{T_i} \leq B_i\right\}. \tag{4.4.1}$$

Let $P_i(k) \triangleq \mathsf{E}\left\{\tilde{x}_i(k)\tilde{x}_i(k)^T\right\}$ be the variance of the augmented state. When $\tilde{P}_i \triangleq \lim_{k\to\infty} P_i(k) < +\infty$ (i.e., the variance converges), we will say that the closed loop system is *mean square stable* [59]. We will consider $\mathrm{tr}\left\{\tilde{P}_i\right\}$ (i.e., the sum of the mean squared values of the state variables) as a QoC metric for control task $\tau_i$. This is a

[2] We recall that a discrete–time linear system is said Schur-stable if the absolute value of all eigenvalues of its dynamical matrix is strictly less than 1.

function of the probability $\mu_i$ and hence of the bandwidth $B_i$, which we will denote by $\phi_i(B_i) \triangleq \operatorname{tr}\left\{\tilde{P}_i\right\}$.

The QoC functions for each system can be consolidated into a global cost function $\Phi(B_1, B_2, \ldots, B_n)$. We do this by computing the infinity norm of the different QoC functions: $\Phi(B_1, \ldots, B_n) \triangleq \max_i |\phi_i(B_i)|$, where the absolute value can be omitted because the $\phi_i$ functions are non negative by definition. The infinity norm is a sensible choice if the designer seeks a balanced performance between the different tasks (avoiding the case of one control loop performing exceedingly well at the expenses of the other ones). If the designer prefers an unbalanced performance, where some higher importance tasks receive larger bandwidths, the previous global cost function can be adapted accordingly. A suitable rescaling of the individual $\phi_i$ functions by means of some positive coefficients $q_i$ can easily achieve this goal. In this case, the ensuing global cost function will read $\Phi(B_1, \ldots, B_n) \triangleq \max_i |q_i \phi_i(B_i)|$.

The optimisation problem can be formally stated as follows[3]:

$$\min_{B \in \mathcal{B}} \Phi(B) = \min_{B \in \mathcal{B}} \max_{i \in \{1,\ldots,n\}} \phi_i(B_i) \tag{4.4.2}$$

where $B \triangleq (B_1, \ldots, B_n)$. The set $\mathcal{B}$ is the set of feasible solutions and is given by the polytope:

$$\mathcal{B} \triangleq \left\{ (B_1, \ldots, B_n) \in \mathbb{R}^n \mid B_i^{(m)} \leq B_i \leq B_i^{(M)}, \sum_{i=1}^{n} B_i \leq 1 \right\}, \tag{4.4.3}$$

where the upper bound $B_i^{(M)}$ is to ensure that the task does not receive more bandwidth than it needs to achieve probability 1 of meeting the deadline and the lower bound $B_i^{(m)}$ identifies a minimal critical bandwidth to ensure mean square stability of the feedback loop, and $\sum_{i=1}^{n} B_i \leq 1$ comes from condition (4.2.1). In the following section we derive an expression for both $B_i^{(M)}$ and $B_i^{(m)}$.

### 4.4.2 Constraints

In this section we show how to identify the constraints for Problem (4.4.2). For notational simplicity, we will drop the $i$ subscript whenever the discussion refers to a specific task. In the following, we assume that the noise $w(\cdot)$ is an i.i.d. process with zero mean and variance $\mathsf{E}\left\{w(k)w(k)^T\right\} = W \in \mathbb{R}^{n_x \times n_x}$. Also the computation time $\{c(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ of the tasks is an i.i.d. random process, which is mutually independent of $w(\cdot)$. Both processes $w(\cdot)$ and $c(\cdot)$ are assumed independent of the state.

The dynamic evolution of the variance $P \in \mathbb{R}^{n_{\tilde{x}} \times n_{\tilde{x}}}$ of the state $\tilde{x}$ is given by

$$\begin{aligned} P(k+1) =& \mathsf{E}\left\{\tilde{x}(k+1)\tilde{x}(k+1)^T\right\} = \\ =& \mu \mathsf{E}\left\{\left(A_c\tilde{x}(k) + v(k)\right)\left(A_c\tilde{x}(k) + \omega(k)\right)^T\right\} + \\ &+ (1-\mu)\mathsf{E}\left\{\left(A_o\tilde{x}(k) + \omega(k)\right)\left(A_o\tilde{x}(k) + \omega(k)\right)^T\right\}, \end{aligned}$$

[3] For sake of exposition, we present here the balanced performance case only.

where the probability of meeting the deadline $\mu$ stochastically rules the switching between the different (closed and open loop) dynamics. Taking into account the mutual independence of the stochastic processes and the fact that $w(\cdot)$ has zero mean and constant variance $W$, the equation above can be written as

$$P(k+1) = \mu A_c P(k) A_c^T + (1-\mu)\, A_o P(k) A_o^T + H, \tag{4.4.4}$$

where $H \triangleq \left[\begin{smallmatrix} W & 0 \\ 0 & 0 \end{smallmatrix}\right]$.

*An algorithmic estimate of the critical probability*

In order to identify the constraints of Problem (4.4.2) (and in particular the lower bound $B_i^{(m)}$), we need to identify the critical probability, defined as the infimum of $\mu$ for which the system is mean square stable. Using Kronecker product properties we can write the dynamics (4.4.4) as

$$\mathrm{vec}(P(k+1)) = \left(\mu A_c^{[2]} + (1-\mu) A_o^{[2]}\right) \mathrm{vec}(P(k)) + \mathrm{vec}(H), \tag{4.4.5}$$

where $M^{[2]} \triangleq M \otimes M$ and $\mathrm{vec}(\cdot)$ is the linear operator producing a vector by stacking the columns of a matrix. This is a discrete–time linear time–invariant system in the state $\mathrm{vec}(P(j)) \in \mathbb{R}^{n_x^2}$. Hence, it admits a steady state solution $\tilde{P}(\mu)$ w.r.t. the constant input $\mathrm{vec}(H) \in \mathbb{R}^{n_x^2}$ if

$$\max_i \left|\lambda_i \left(\mu A_c^{[2]} + (1-\mu) A_o^{[2]}\right)\right| < 1, \tag{4.4.6}$$

with $\lambda_i(M)$ denoting $i$-th eigenvalue of $M$.

In view of these considerations the critical probability can be defined as

$$\widetilde{\mu} \triangleq \inf_{\bar{\mu}\in[0,1]} \left\{ \bar{\mu} \mid \max_i \left|\lambda_i \left(\mu A_c^{[2]} + (1-\mu) A_o^{[2]}\right)\right| < 1, \forall\, \mu \geq \bar{\mu} \right\}. \tag{4.4.7}$$

Note that a solution with $\widetilde{\mu} < 1$ always exists, since $A_c$ is Schur-stable and the eigenvalues are continuous functions of $\mu$. In practice, instead of looking for the infimum in the continuum set $[0,1]$, we can fix an accuracy level $t_\mu$ and adopt Algorithm 4.1 which implements a dichotomic search in the $\mu$ space. The function `SchurStableSegment(A,B)` implements the necessary and sufficient algebraic conditions in Theorem 4.*iii* of [75] and provides a positive answer if all the matrices given by the convex combination of $A$ and $B$ are Schur-stable.

### 4.4.3 Cost function

In this section we discuss an analytical expression of the QoC function $\phi(\cdot)$ for Problem (4.4.2) in terms of the probability $\mu$ of meeting the deadline, and then (via equation (4.4.1)) of the bandwidth $B$. Here again we are dropping the $i$ subscript for notational simplicity. The following Lemma shows an analytic computation for the function $\phi(\cdot)$.

**Algorithm 4.1** Find Critical Probability

```
function FindMu(A_c,A_o)
    A_C = A_c^[2]
    A_O = A_o^[2]
    μ̃ = 1
    μ_0 = 0
    while not(μ̃ − μ_0 < t_μ ∧ μ̃ < 1 − t_μ) do
        μ = (μ̃ − μ_0)/2 + μ_0
        A_2 = (1 − μ) ∗ A_O + μ ∗ A_C
        if SchurStableSegment(A_C, A_2) then
            μ̃ = μ
        else
            μ_0 = μ
        end if
    end while
    return μ̃
end function
```

**Lemma 4.1.** *The QoC function* $\phi : [\widetilde{\mu}, 1] \to \mathbb{R}_{>0} \cup \{+\infty\}$, $\mu \mapsto \operatorname{tr}\left\{\tilde{P}(\mu)\right\}$ *can be expressed as the ratio of two polynomials in* $\mu$*:*

$$\operatorname{tr}\left\{\tilde{P}(\mu)\right\} = \frac{n_{\tilde{x}}^2 \sum_{j=0}^{n_{\tilde{x}}^2-1} \alpha_j \mu^j}{\sum_{j=0}^{n_{\tilde{x}}^2-1} \gamma_j \mu^j}, \tag{4.4.8}$$

*where the* $\alpha_j$ *and* $\gamma_j$ *coefficients can be found as algebraic functions*[4] *of the* $A_c$ *and* $A_o$ *matrices.*

The QoC function $\phi(\cdot)$ is continuous with respect to $\mu$ because so are the eigenvalues of $\tilde{P}(\mu)$. Moreover, for any $\mu \in [\widetilde{\mu}, 1]$ equation (4.4.6) is verified by definition and $\operatorname{tr}\left\{\tilde{P}(\mu)\right\}$ is also finite.

**Example 4.2.** Coming back to the system of Example 4.1, let us consider a set of possible sampling periods $T_i = \{20, 24, 32, 40, 48, 56\}$ ms. The corresponding set of critical probabilities provided by the Algorithm 4.1 is then given by $\widetilde{\mu} = \{0.18, 0.2, 0.23, 0.26, 0.29, 0.31\}$.

Let us assume now that the system is affected by two mutually independent noise inputs (gathered in the vector $w$) of zero mean and standard deviation $\sigma = 0.01$. The associated covariance matrix is $W = \sigma^2 I_2$. The QoC function $\phi$ in (4.4.8) can be explicitly computed by means of the expression (4.4.8). For the system in exam, we have $n_{\tilde{x}} = 6$, hence 36 coefficients $\alpha_j$ and 36 coefficients $\gamma_j$. However, most of the coefficients turn out to be practically zero and the function $\phi$ becomes a ratio of polynomials of degree 8. The non zero coefficients are reported in Table 4.4.1. A graph of the $\phi$ for $\mu \in [0.18, 1]$ is shown in Figure 4.4.1.

[4] An explicit expression for these coefficients is provided in [89].

| $\alpha_{27} \to \alpha_{35}$ | $\gamma_{27} \to \gamma_{35}$ |
|---|---|
| 0.0000005 | 0.0000029 |
| −0.0000227 | −0.0002319 |
| 0.0007560 | 0.0093476 |
| −0.0148452 | −0.2007313 |
| 0.1475369 | 2.0286992 |
| −0.5625039 | −7.4926220 |
| −0.0150922 | 0.2150213 |
| 0.0092984 | −0.0782990 |
| 0.0390932 | 0.3470355 |

TABLE 4.4.1: Coefficients $\alpha_j$ and $\gamma_j$ for the QoC function $\phi$ defined in (4.4.8). They have been all multiplied by $10^6$.

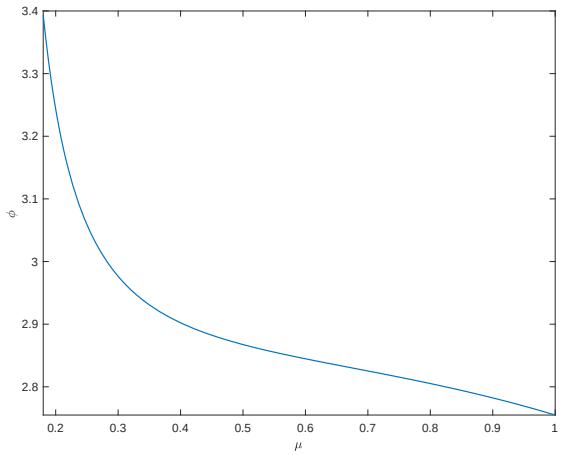


FIGURE 4.4.1: Graph of the function $\phi$ computed in the Example 4.2.

We stress that in deriving all the previous quantities we did not make any assumption on the distribution of the computation time (except from its independence of the noise $w$), as they are all independent of it.

### 4.4.4 *Formalisation of the optimisation problem*

Let $\Delta_{c_i} : \mathbb{R}_{\geq 0} \to [0,1]$ denote the cumulative distribution function (cdf) of the computation time $\{c_i(\cdot)\}$ for the task $\tau_i$. In view of (4.4.1), the probability $\mu_i$ to finish before the deadline is given by $\mu_i = \mathsf{P}\{c_i(k) \leq T_i B_i\} = \Delta_{c_i}(T_i B_i)$. By definition $\Delta_{c_i}$ is monotone non–decreasing. For the sake of simplicity, we will also let it to be strictly increasing and hence invertible. This leads us to an easy expression for the bandwidth $B_i$ that attains the probability $\mu_i$ in Problem (4.4.2): $B_i = \frac{\Delta_{c_i}^{-1}(\mu_i)}{T_i}$, for every $\mu_i \in [0,1]$.

| $\eta_i$ [ms] | 6 | 8 | 10 | 12 | 14 | 16 | 20 | 24 | 28 |
|---|---|---|---|---|---|---|---|---|---|
| $B_i^{(m)}$ % | 23.6 | 27.2 | 30.8 | 34.4 | 38 | 41.6 | 48.8 | 56 | 63.2 |
| $B_i^{(M)}$ % | 40 | 60 | 80 | 100 | 120 | 140 | 180 | 220 | 260 |

TABLE 4.4.2: Minimum and maximum bandwidth computed on the system of Example 4.1.

Consequently, the minimum bandwidth to achieve mean square stability is given by

$$B_i^{(m)} \triangleq \frac{\Delta_{c_i}^{-1}(\widetilde{\mu}_i)}{T_i}, \tag{4.4.9}$$

where $\widetilde{\mu}_i$ is the critical probability given by (4.4.7). Likewise, the bandwidth necessary for the $i$-th task to finish within its deadline with probability 1 is given

$$B_i^{(M)} \triangleq \frac{\Delta_{c_i}^{-1}(1)}{T_i}.$$

In our framework the deadlines $T_i$ are fixed, thus the probability $\mu_i$ is a strictly increasing function of the bandwidth $B_i$: $\mu_i = \Delta_{c_i}(B_i)$. The composition $\phi_i \circ \Delta_{c_i}(\cdot)$ is a function of the bandwidth $B_i$ in the set $[B_i^{(m)}, B_i^{(M)}]$. With a slight abuse of notation, in what follows we will write $\phi_i(\cdot)$ directly as a function of $B_i$ to refer such a composition. The expressions we have just shown for $B_i^{(m)}$, $B_i^{(M)}$ and $\phi_i(\cdot)$ allow us to give Problem (4.4.2) an explicit form. We will require below that polytope $\mathcal{B}$ be not empty, and therefore the point $B^{(m)} = (B_1^{(m)}, \ldots, B_n^{(m)})$ be feasible.

**Example 4.3.** Still with reference to the system in Example 4.1 and 4.2, let us assume now that the computation time $c_i$ is described by a uniform distribution $\mathfrak{U}_{[\eta_i, b_c]}$ parametrised by the mean computation time $\eta_i$ and by the best-case execution time (BCET) $b_c$ ($c_i \geq b_c$). The distribution is defined in the range $[b_c, w_c]$, where $w_c = 2\eta_i - b_c$ is the worst case execution time (WCET) and we fix here $b_c = 4$ ms. Fixing the sampling time to $T = 20$ ms we obtain the corresponding critical probability $\widetilde{\mu}_i = 0.18$. Once the mean value $\eta_i$ is given, the distribution and the cumulative distribution $\Delta_{c_i}$ of the computation time $c_i$ are fixed. Hence, the minimum and maximum bandwidth $B_i^{(m)}$, $B_i^{(M)}$ can be easily computed. The Table 4.4.2 reports the values of $B_i^{(m)}$ and $B_i^{(M)}$ for different mean values $\eta_i$.

Notice that for $\eta_i > 14$ ms, the task WCET $w_c$ is greater than the deadline $T_i$ and indeed the maximum bandwidth $B_i^{(M)}$ is larger than 100%.

## 4.5 SOLUTION OF THE OPTIMISATION PROBLEM

In this section we first state some fundamental results on the structure of the solutions of the optimisation problem introduced above. Then we show how such properties can be used to build an efficient optimisation algorithm.

### 4.5.1 Degenerate Problem

The optimisation Problem (4.4.2) can present some special cases that require a specific consideration.

**Definition 4.1.** The optimisation Problem (4.4.2) is said to be *degenerate* if there exist $i, j \in \{1, \dots, n\}$, $i \neq j$ such that $\phi_i(B_i) \geq \phi_j(B_j)$ for every $B_i \in [B_i^{(m)}, B_i^{(M)}]$ and $B_j \in [B_j^{(m)}, B_j^{(M)}]$. In such a case $\phi_i(\cdot)$ is said *to dominate* $\phi_j(\cdot)$.

If $\phi_j(\cdot)$ is a dominated function the optimal value of the cost function is not influenced by the bandwidth $B_i$: $\Phi(B) = \max_{h\in\{1,\dots,n\}} \phi_h(B_h) = \max_{h\in\{1,\dots,n\}\setminus\{j\}} \phi_h(B_h)$. Therefore, we can simply set $B_j = B_j^{(m)}$ to secure the largest possible feasibility set for the remaining variables, which will be an $(n-1)$–dimensional facet of the original polytope $\mathcal{B}$. In general, if there are $n' < n$ dominated functions with indices in the set $I' \subset \{1, \dots, n\}$, we can reduce the search for the optimal solution to the $(n - n')$–dimensional facet:

$$\mathcal{B}_{n'} \triangleq \mathcal{B} \cap \left\{ B \in \mathbb{R}^n_{>0} \mid B_h = B_h^{(m)}, \forall h \in I' \right\}. \tag{4.5.1}$$

### 4.5.2 Optimal solution with rectified functions

The non monotonicity of the functions $\phi_i(\cdot)$ complicates the analysis of the solution set of the optimisation Problem (4.4.2), even in the non degenerate case. Therefore, we first study the solution set of an auxiliary optimisation problem, obtained by forcing all the $\phi_i(\cdot)$ to be non increasing functions, and then we exploit such knowledge to address the original one.

We define the *rectified* functions $\phi_i^{(rect)} : [B_i^{(m)}, B_i^{(M)}] \to \mathbb{R}_{>0} \cup \{+\infty\}$ for every $i \in \{1, \dots, n\}$ as follows (see Figure 4.5.1)

$$\phi_i^{(rect)}(B_i) \triangleq \min_{b\in[B_i^{(m)}, B_i]} \phi_i(b), \tag{4.5.2}$$

and we consider the associated optimisation problem

$$\min_{B\in\mathcal{B}} \Phi^{(rect)}(B) = \min_{B\in\mathcal{B}} \max_{i\in\{1,\dots,n\}} \phi_i^{(rect)}(B_i). \tag{4.5.3}$$

*Remark* 4.1. Differently from the functions $\phi_i(\cdot)$ (which are ratio of polynomials), their rectified counterparts $\phi_i^{(rect)}(\cdot)$, can be constant over intervals. Such functions hold on the values taken by the corresponding functions $\phi_i(\cdot)$ in their local minima. Indeed, if $B_i \in [B_i^{(m)}, B_i^{(M)}]$ is such that $\phi_i^{(rect)}(B_i) \neq \phi_i(B_i)$ then it can be easily verified that $B_i^\star = \min_{\alpha\in\mathcal{A}_i\left(\phi_i^{(rect)}(B_i)\right)} \alpha$, where $\mathcal{A}_i(x) \triangleq \{b \in [B_i^{(m)}, B_i^{(M)}] \mid \phi_i^{(rect)}(b) = x\}$, is a local minimum for $\phi_i(\cdot)$ and $\phi_i^{(rect)}(B_i) = \phi_i(B_i^\star)$ (see also Figure 4.5.1).

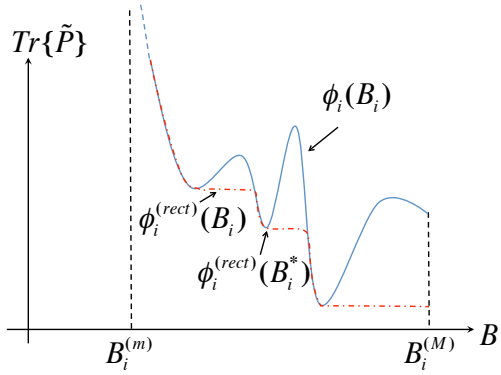


FIGURE 4.5.1: Graphical representation of the rectification of $\phi_i(\cdot)$. $\phi_i^{(rect)}(\cdot)$ holds on the values taken by the corresponding function $\phi_i(\cdot)$ in its local minima.

We now state a result on the structure of the solution for the optimisation Problem (4.5.3).

**Theorem 4.1.** *Assume that the optimisation Problem (4.5.3) is not degenerate, the functions $\phi_i : [B_i^{(m)}, B_i^{(M)}] \to \mathbb{R}_{>0} \cup \{+\infty\}$ are continuous and differentiable for every $i \in \{1, \dots, n\}$ and the functions $\phi_i^{(rect)} : [B_i^{(m)}, B_i^{(M)}] \to \mathbb{R}_{>0} \cup \{+\infty\}$ are defined as in (4.5.2). Define the optimal solution set as $\mathcal{X}^{(rect)^*} \triangleq \arg\min_{B \in \mathcal{B}} \Phi^{(rect)}(B)$, and the following quantities:*

$$\bar{t}^{(M)} \triangleq \max_{i \in \{1,\dots,n\}} \phi_i^{(rect)}(B_i^{(M)}), \quad \underline{t}^{(m)} \triangleq \min_{i \in \{1,\dots,n\}} \phi_i^{(rect)}(B_i^{(m)}),$$
$$\bar{t}^{(m)} \triangleq \max_{i \in \{1,\dots,n\}} \phi_i^{(rect)}(B_i^{(m)}).$$

*Let $\mathcal{A}_i(x) \triangleq \{b \in [B_i^{(m)}, B_i^{(M)}] \mid \phi_i^{(rect)}(b) = x\}$ and the vectors $\check{B}^{(m)} = (\check{B}_1^{(m)}, \dots, \check{B}_n^{(m)})$ and $\check{B}^{(M)} = (\check{B}_1^{(M)}, \dots, \check{B}_n^{(M)})$ be defined as:*

$$\check{B}_i^{(m)} \triangleq \min_{\alpha \in \mathcal{A}_i(\underline{t}^{(m)})} \alpha, \quad \check{B}_i^{(M)} \triangleq \min_{\alpha \in \mathcal{A}_i(\bar{t}^{(M)})} \alpha, \tag{4.5.4}$$

*for every $i \in \{1, \dots, n\}$. The following mutually exclusive cases are given:*

*i) $\sum_{i=1}^n \check{B}_i^{(M)} \leq 1$, then $\check{B}^{(M)} \in \mathcal{X}^{(rect)^*}$ and the optimal value is $t^* = \bar{t}^{(M)}$;*

*ii) $\sum_{i=1}^n \check{B}_i^{(m)} \leq 1$ and $\sum_{i=1}^n \check{B}_i^{(M)} > 1$, then there exists $\tilde{B} = (\tilde{B}_1, \dots, \tilde{B}_n)$, $\tilde{B} \in \mathcal{X}^{(rect)^*}$ such that $\phi_i^{(rect)}(\tilde{B}_i) = \phi_j^{(rect)}(\tilde{B}_j)$ for all $i$ and $j$, and $\sum_{i=1}^n \tilde{B}_i = 1$. The optimal value is $t^* \in (\bar{t}^{(M)}, \underline{t}^{(m)}]$;*

*iii) $\sum_{i=1}^n \check{B}_i^{(m)} > 1$, then there exists $B^* = (B_1^*, \dots, B_n^*)$ such that for $\bar{h} = \arg\min_{j \in \{1,\dots,n\}} \phi_j^{(rect)}(B_j^{(m)})$, the element $B_{\bar{h}}^* = B_{\bar{h}}^{(m)}$ and $B^* \in \mathcal{X}^{(rect)^*}$. The optimal values is $t^* \in (\underline{t}^{(m)}, \bar{t}^{(m)}]$.*

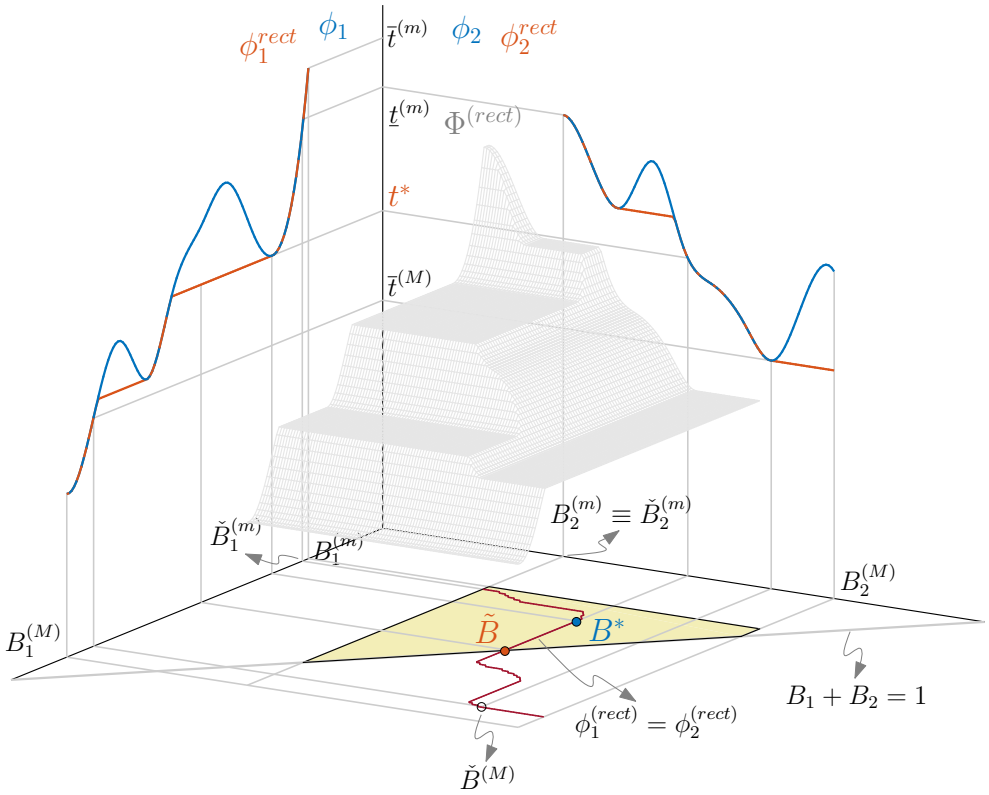


FIGURE 4.5.2: Example of optimisation in the case of two functions $\phi_i$. The picture reports all the key quantities defined in Theorem 4.1 point *ii)* and Corollary 4.1 point *i)*. In blue are the $\phi_i$ functions and in red their rectified version. The overall function $\Phi^{(rect)}$ is represented in faint grey. The yellow polytope represents the set of admissible bandwidths (notice the line $B_1 + B_2 = 1$) and the dark magenta curve represents the set of bandwidth for which the $\phi_i^{(rect)}$ functions assume the same value. The optimal value $t^*$ and the optimal solution $\tilde{B}$ of the Problem (4.5.3) are reported in red, while the optimal solution $B^*$ of the Problem (4.4.2) is reported in blue.

The intuition behind the rationale of the Theorem is as follows. The optimal solution of Problem (4.5.3) belongs to the region in which all the functions $\phi_i^{(rect)}(\cdot)$ assume the same value, if such a region exists. In that case, the optimal solution is either where one of the $\phi_i^{(rect)}(\cdot)$ has reached its minimum value (point *i)*) or where it is not possible to further decrease it (point *ii)*). If the region in which the functions $\phi_i^{(rect)}(\cdot)$ assume the same value does not exist, then there exists at least one function that will not play a role in the minimisation (point *iii)*). A detailed representation of the optimal solution $\tilde{B}$ of type *ii)* and related optimal value $t^*$ is depicted in Figure 4.5.2. All the key quantities defined in the Theorem 4.1 are reported, along with the region of admissible bandwidths and the set where the $\phi_i^{(rect)}$ functions assume the same value.

*Remark* 4.2. The degenerate case can be addressed by restricting the optimisation domain to the facet $\mathcal{B}_{n'}$ in (4.5.1), which is an $(n - n')$–dimensional polytope. In the

application of Theorem 4.1, the sum of the remaining bandwidths $B_i$ has to be set to $\sum_{i\in\{1,\dots,n\}\setminus I'} B_i = 1-\sum_{i\in I'} B_i^{(m)}$ (rather than 1) to account for the bandwidth distributed to the tasks with degenerate functions $\phi_i(\cdot)$.

*Remark* 4.3. The case *iii)* in Theorem 4.1 is similar to the degenerate case. Indeed, once the $\bar{h}$-th component of $B^*$ is fixed to $B^*_{\bar{h}} = B^{(m)}_{\bar{h}}$, the other components are found solving the optimal problem on the sub-polytope $\mathcal{B}' \triangleq \mathcal{B} \cap \{(B_1,\dots,B_n) \in \mathbb{R}^n_{\geq 0} \mid B_h = B^{(m)}_h\}$ with the constraint $\sum_{i\in\{1,\dots,n\}\setminus\{\bar{h}\}} B_i = 1 - B^{(m)}_{\bar{h}}$.

### 4.5.3 Optimal solutions of Problem (4.4.2)

If the $\phi_i(\cdot)$ functions are strictly decreasing, they coincide with their rectified counterpart in Theorem 4.1. In this case, it is possible to show that if the situation of point *ii)* takes place, then the optimal value is unique. In the discussion below, we show a Corollary tackling the generic case of non-monotone functions $\phi_i(\cdot)$. The characterisation of the optimal set of the Problem (4.4.2) is given in terms of the optimal set of the Problem (4.5.3). A representation of the optimal solution $B^*$ of the point *i)* of the following Corollary is shown in Figure 4.5.2.

**Corollary 4.1.** *Assume the optimisation Problem (4.4.2) and (4.5.3) are not degenerate. Define the corresponding optimal solution sets as* $\mathcal{X}^* \triangleq \arg\min_{B\in\mathcal{B}} \Phi(B)$ *and* $\mathcal{X}^{(rect)^*} \triangleq \arg\min_{B\in\mathcal{B}} \Phi^{(rect)}(B)$. *Given* $\widetilde{B}^* = (\tilde{B}^*_1,\dots,\tilde{B}^*_n) \in \mathcal{X}^{(rect)^*}$, *we have:*

*i) the point* $B^* = (B^*_1,\dots,B^*_n)$ *such that* $B^*_i = \min_{\alpha\in\mathcal{A}_i\left(\phi_i^{(rect)}(\tilde{B}^*_i)\right)} \alpha$ *for every* $i \in \{1,\dots,n\}$, *is optimal for the Problem (4.4.2), namely* $B^* \in \mathcal{X}^*$. *Moreover, for the optimal value we have* $\Phi(B^*) = \Phi^{(rect)}(\tilde{B}^*)$;

*ii) if* $\phi_i(\tilde{B}^*_i) = \phi_i^{(rect)}(\tilde{B}^*_i)$ *for every* $i \in \{1,\dots,n\}$, *then* $\tilde{B}^* \in \mathcal{X}^*$; *otherwise the optimal point* $B^* \in \mathcal{X}^*$ *defined at point i) is such that there exists at least one index* $\bar{h} \in \{1,\dots,n\}$ *for which* $B^*_{\bar{h}} = B^\dagger_{\bar{h}}$ *with* $B^\dagger_{\bar{h}}$ *local minimum of* $\phi_{\bar{h}}(\cdot)$.

### 4.5.4 The Solution Algorithm

Theorem 4.1 and Corollary 4.1 are exploited to devise the Algorithm 4.2 for the computation of an optimal solution for the Problem (4.4.2). First the quantities $\check{B}^{(M)}$ and $\check{B}^{(m)}$ are computed (Lines 6-9), then condition at point *i)* of Theorem 4.1 is readily checked (Lines 10-11) and in case the optimal solution immediately returned. If the condition at point iii) holds, the sub-polytope defined in Remark 4.2 is in case determined (Lines 13-18). Then, for condition at point *ii)*, two nested binary search algorithms are employed (Lines 20-32). The inner one, implemented in the function `FIND POINTS` (not reported due to its simplicity), is responsible for computing the generalised inverse of the functions $\phi_i^{(rect)}(\cdot)$: given a value $\check{t} \in [\bar{t}^{(M)}, \underline{t}^{(m)}]$, the algorithm provides the

**Algorithm 4.2** Solution Algorithm

**function** OPTIMALSOLUTION($\phi^{(rect)}$, $B^{(m)}$, $B^{(M)}$, $n$)
  $\text{Set}_I = [1, \dots, n]$; $\text{Set}_H = \varnothing$; $\text{Max}_B = 1$
  $\overline{t}^{(M)} = \max_{i \in \text{Set}_I} \phi_i^{(rect)}(B_i^{(M)})$
  $\underline{t}^{(m)} = \min_{i \in \text{Set}_I} \phi_i^{(rect)}(B_i^{(m)})$
  $\overline{t}^{(m)} = \max_{i \in \text{Set}_I} \phi_i^{(rect)}(B_i^{(m)})$
  **for** $i \in \text{Set}_I$ **do**
    $[\check{B}_i^{(m)}, a] = \text{FINDPOINTS}(\underline{t}^{(m)}, \phi_i^{(rect)}, B_i^{(m)}, B_i^{(M)})$
    $[\check{B}_i^{(M)}, a] = \text{FINDPOINTS}(\overline{t}^{(M)}, \phi_i^{(rect)}, B_i^{(m)}, B_i^{(M)})$
  **end for**
  **if** $\sum_{i \in \text{Set}_I} \check{B}_i^{(M)} \leq \text{Max}_B$ **then**
    **return** $[\check{B}^{(M)}, \overline{t}^{(M)}]$
  **else**
    $t_a = \overline{t}^{(M)}$; $t_b = \underline{t}^{(m)}$
    **while** $\sum_{i \in \text{Set}_I} \check{B}_i^{(m)} > \text{Max}_B$ **do**
      $\bar{h} = \arg\min_{j \in \text{Set}_I} \phi_j^{(rect)}(B_j^{(m)})$
      $\text{Set}_I = \text{Set}_I \backslash \bar{h}$; $\text{Set}_H = \text{Set}_H \cup \bar{h}$; $\text{Max}_B = \text{Max}_B - B_{\bar{h}}^{(m)}$
      $\underline{t}^{(m)} = \min_{i \in \text{Set}_I} \phi_i^{(rect)}(B_i^{(m)})$; $t_b = \underline{t}^{(m)}$
    **end while**
    $\check{t} = \frac{t_a + t_b}{2}$
    **while** $\sum_{i \in \text{Set}_I} \underline{B}_i^{\check{t}} > \text{Max}_B \vee (\sum_{i \in \text{Set}_I} \underline{B}_i^{\check{t}} < \text{Max}_B \wedge \sum_{i \in \text{Set}_I} \overline{B}_i^{\check{t}} \leq \text{Max}_B)$ **do**
      **for** $i \in \text{Set}_I$ **do**
        $[\underline{B}_i^{\check{t}}, \overline{B}_i^{\check{t}}] = \text{FINDPOINTS}(\check{t}, \phi_i^{(rect)}, B_i^{(m)}, B_i^{(M)})$
      **end for**
      **if** $\sum_{i \in \text{Set}_I} \underline{B}_i^{\check{t}} > \text{Max}_B$ **then**
        $t_a = \check{t}$
      **else**
        **if** $\sum_{i \in \text{Set}_I} \overline{B}_i^{\check{t}} < \text{Max}_B$ **then**
          $t_b = \check{t}$
        **end if**
      **end if**
      $\check{t} = \frac{t_a + t_b}{2}$
    **end while**
    **return** $[\underline{B}_{\text{Set}_I}^{\check{t}} \bigcup B_{\text{Set}_H}^{(m)}, \check{t}]$
  **end if**
**end function**

extreme points $\underline{B}_i^{\check{t}} \triangleq \min_{\alpha \in \mathcal{A}_i(\check{t})} \alpha$ and $\overline{B}_i^{\check{t}} \triangleq \max_{\alpha \in \mathcal{A}_i(\check{t})} \alpha$ (if $\phi_i^{(rect)}(\cdot)$ is locally invertible in $\check{t}$, then $\underline{B}_i^{\check{t}} = \overline{B}_i^{\check{t}}$). Notice that the algorithm is proved to converge to the optimal solution in this case in light of point *i)* of Corollary 4.1. Moreover, the problem solution strategy is particularly efficient for the function $\phi_i$ chosen in this paper, which links the bandwidth $B_i$ to the QoC with an explicit analytic form. Other choices of $\phi_i$ are legitimated and perfectly compatible with the approach. As an example, if no explicit form is available, $\phi_i$ could be numerically computed "by points" and interpolated.

| Discrete-time linear systems | | Controllers | | | |
|---|---|---|---|---|---|
| $(n_{x_i}, m_i, p_i)$ | Unst. poles | Task | $T_i$ | Distr. | $\tilde{\mu}$ |
| $(2,1,2)$ | $1.01, 1.02$ | $\tau_1$ | 20 | $\mathfrak{U}_{[\eta^\star,4]}$ | 0.18 |
| $(3,1,3)$ | $1.03, 1.01, 1.01$ | $\tau_2$ | 56 | $\mathfrak{B}_{[6,4]}$ | 0.33 |
| $(4,1,3)$ | 1.08 | $\tau_3$ | 56 | $\mathfrak{E}_{[6,4]}$ | 0.33 |

TABLE 4.6.1: Systems adopted in the simulations. Mean and best case computation times as well as periods are expressed in milliseconds. The system executed by task $\tau_1$ is the one described in Example 4.1.

## 4.6 NUMERIC EVALUATION

In order to show a numeric validation of the bandwidth optimal synthesis described in Section 4.5, we consider randomly generated, open-loop unstable, reachable and observable linear discrete time systems subject to a linear combination of discrete time noises. The number of independent noise sources equals the number of states and their stochastic processes are normally distributed with zero mean and standard deviations equal to $\sigma = 0.01$. Among the hundreds of test cases synthesised, we report, as an example, three different systems, for which number of states $n_{x_i}$, inputs $m_i$, outputs $p_i$ and the open loop unstable poles are reported in Table 4.6.1.

The controllers are designed using the systematic LQG optimal control synthesis. They are executed in the tasks $\tau_1$, $\tau_2$ and $\tau_3$, respectively, with execution period $T_i$. Three different probability density functions (pdf) have been chosen for the computation times of the control tasks, which are parametrised by the mean value $\eta_i$ and by the best-case execution time $b_c$ ($c_i \geq b_c$): $\mathfrak{U}_{[\eta_i, b_c]}$ is a uniform distribution defined as in Example 4.3; $\mathfrak{E}_{[\eta_i, b_c]} = \frac{1}{\eta_i} \mathrm{e}^{-\frac{c_i - b_c}{\eta_i}}$ is an exponential distribution, where the WCET is $w_c = +\infty$; $\mathfrak{B}_{[\eta_i, b_c]} = \frac{1}{N} \left(\frac{c_i - b_c}{w_c - b_c}\right)^{\alpha-1} \left(1 - \frac{c_i - b_c}{w_c - b_c}\right)^{\beta-1}$, with normalisation factor $N = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\mathrm{d}x$, is a beta distribution, having WCET $w_c = 60$ ms and parameters $\alpha = 2$ and $\beta = \alpha \frac{w_c - \eta_i}{\eta_i - b_c}$. For all the distributions, the $b_c = 4$ ms. The task set is not hard real-time schedulable since the modelled WCETs are so high that no guarantee to finish the job within the activation period $T_i$ can be given. Table 4.6.1 reports the critical probabilities $\tilde{\mu}$ computed with Algorithm 4.1.

To highlight the relevant features of the solution algorithm presented in Section 4.5, the possible mean values $\eta^\star$ of the execution times of task $\tau_1$ are chosen in the set $\mathcal{N} = \{6, 8, 10, 12\}$ ms (see also Example 4.3), while $\eta_i$ is fixed to 6 ms for $\tau_2$ and $\tau_3$. Figure 4.6.1 shows the optimal bandwidths, minimising the $\mathrm{tr}\, \tilde{P}(\mu)$, in the considered cases.

For $\eta^\star \leq 10$ ms, the optimal solution is given by $B_1 = B_1^{(M)}$, while $B_i < B_i^{(M)}$, $i = 2, 3$, since point *ii)* of Theorem 4.1 applies. By the optimality of the solution, it is evident that the minimum of $\mathrm{tr}\left\{\tilde{P}(\mu)\right\}$ is the same for any $\eta^\star \leq 10$ ms. Notice how for increasing $\eta^\star$, the $B_1^{(M)}$ increases as well. When $\eta^\star = 12$ ms, we have a full exploitation of resources with $B_i < B_i^{(M)}$, $i = 1, 2, 3$, $B_1 > B_2 > B_3$ and $B_1 + B_2 + B_3 = 1$ (recall the

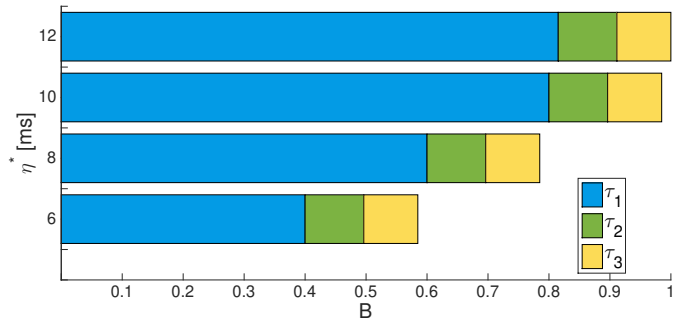


FIGURE 4.6.1: Bandwidth optimal allocation minimising tr $\left\{\tilde{P}(\mu)\right\}$ with respect to the systems specified in Table 4.6.1 with $\eta^\star \in \{6, 8, 10, 12\}$.

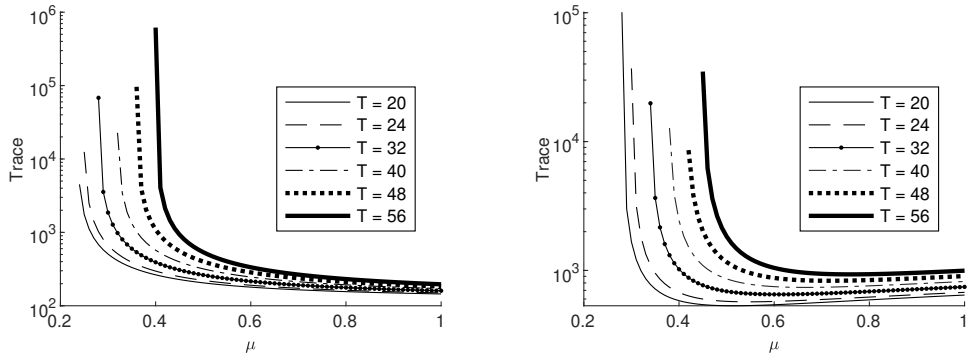


FIGURE 4.6.2: tr $\left\{\tilde{P}(\mu)\right\}$ as a function of the probability $\mu$ for the systems scheduled by task $\tau_2$ (a) and $\tau_3$ (b) specified in Table 4.6.1. The effect of variable execution periods $T_i$ is also considered.

schedulability condition (4.2.1)) and, again, point *ii)* applies. By further increasing $\eta^\star$, we observe a slight increase of $B_1$ still preserving full exploitation of computing resources, i.e. $B_1 + B_2 + B_3 = 1$. When $\eta^\star = 28$ ms, $B_3 = B_3^{(m)}$, hence point *iii)* of Theorem 4.1 applies. As a consequence, for $\eta^\star > 28$ ms, the optimal allocation for a degenerate case applies.

To further substantiate the numerical evaluation, we first present in Figure 4.6.2 the value of tr $\left\{\tilde{P}(\mu)\right\}$ as a function of the probability $\mu$ for the systems scheduled by task $\tau_2$ and $\tau_3$.

It is evident that for the four-dimensional system (Figure 4.6.2-b), increasing the bandwidth (and hence $\mu$) may lead to worse performance. Moreover, Figure 4.6.2 clearly shows the effects of an increased execution period $T_i$ that leads to: i) an higher critical probability (recall Example 4.2); ii) an higher value of the trace (since the model noise $w_i(k)$ in (4.3.1) is integrated for a longer period).

We finally present a set of interesting experiments, where the task $\tau_1$ has fixed

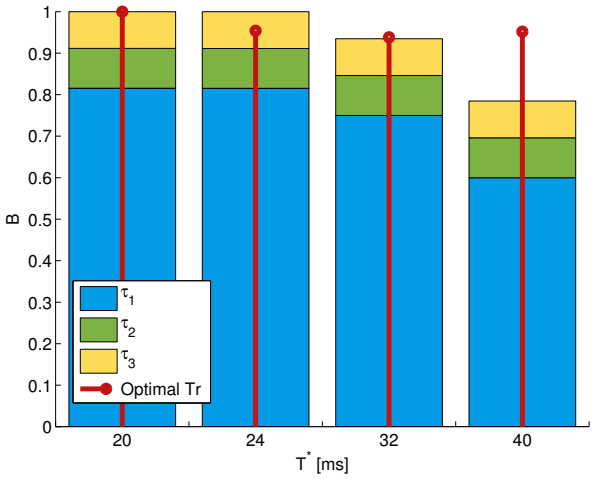


FIGURE 4.6.3: Bandwidth optimal allocation minimising tr $\{P\}$ with respect to the task set specified in Table 4.6.1 with $\eta^\star = 14$ ms and $T_1 \in \{20, 24, 32, 40\}$ ms. The solid thick line represents the normalised value of the adopted metrics.

mean value $\eta^\star = 14$ ms and variable period $T_1 \in \{20, 24, 32, 40\}$ ms. The results of the optimisation is hence influenced by the different periods (see Figure 4.6.3).

This figure shows how increasing the period has evident benefits on the total allocated bandwidth with a tolerable price to pay on the control performance (the tr $\left\{\tilde{P}(\mu)\right\}$ remains almost constant across the selected periods).

## 4.7 CONCLUSIONS

In this Chapter we considered an application scenario where multiple tasks, running on a shared computation platform, are used to implement independent feedback loops. In the spirit of soft real-time systems, a task is allowed to occasionally require more computation time than the amount specified by its deadline. If the computation has completed within the deadline, the control value is updated, otherwise the task is dropped and the control retains the previous value. Here are the main contributions:

*A flexible QoS model.* We introduced a stochastic, flexible QoS model for the control tasks scheduled on a CBS scheduler. The scheduler can vary the QoS by allotting different bandwidths to the various tasks.

*A realistic QoC metric.* We presented a sensible measure of the overall performance of the control tasks in the form of the trace of the steady state covariance.

*An optimal bandwidth allocation algorithm.* We tackled the problem of bandwidth allocation showing its formulation as an optimisation problem. We provided a formal characterisation of the optimal solutions and developed an efficient solution algorithm.

# 5

# PLATFORM–ORIENTED OPTIMAL BANDWIDTH ALLOCATION

## 5.1 INTRODUCTION

In this chapter we refine the scheduling model introduced in the previous chapter. The starting point is always a set of concurrent real-time tasks competing for the computation resources provided by a unique CPU. In classical digital control design, the QoS of a computation platform is generally associated with the $(T, D)$ pair, where $T$ denotes the sampling time (or period) of a control task and $D$ is its loop delay. The standard tools of digital control allow the designer to identify the QoC performance *given* a $(T, D)$ QoS specification. But the $(T, D)$ pair is easy to identify only when the controller is implemented using dedicated hardware resources. As we have seen, on a shared CPU with tasks whose execution times can significantly fluctuate, this is no longer the case. A more sophisticated model for the QoS is needed, which exposes some scheduling parameters to be tuned (or co-tuned) to combine QoC guarantees with an efficient resource utilisation.

The model presented in Chapter 4 is a first attempt in the direction of a *flexible platform,* allowing a designer to ensure both a reasonable QoC over all control tasks and a fair distribution of the computation resources. That model is based on the knowledge of the period $T$ and on a set of rules inspired by the time–triggered computation (see Section 4.2.2). In particular, the second rule states that, if a task is unable to complete its execution within the deadline, it is cancelled. Motivated by the practical observation that the strict respect of every deadline is not needed in several control systems [49], we aim to enrich our model with one more tuning parameter: the delay $D$. Indeed, further flexibility can be achieved if the control tasks unable to complete their execution within

the deadline are allowed to terminate after the deadline with a bounded delay.

The overall objective is still to ensure a safe coexistence in the same CPU of control tasks with *guaranteed QoC* with the largest possible number of concurrent activities. In the wide range of possible applications of our technique, we have isolated two significant scenarios for illustrative purposes. In the first one, we aim for the maximum number of applications that receive real–time guarantees along with the control task. In the second one, the CPU is shared with best effort activities and we aim to give the latter optimised chance of execution. The cornerstone of our approach is a generalisation of the classical $(T, D)$ QoS model. In the model presented in this chapter, we still have a period $T$ used to trigger the activation of the control tasks, but samples are not always collected at the timer expiration. The loop delay $D$ is not fixed, but it evolves according to a stochastic model $\mathcal{F}_D$. The stochastic switches in the loop delay induce a closed loop dynamics which turns out to be described by the Stochastic Jump Linear System model introduced in Chapter 2. The construction of this model allows us to set up an optimisation problem where the decision variables are the scheduling parameters of the control task, and the goal is to minimise the CPU utilisation under the constraint that the system attains some specified QoC goals. The result of this process can be different for different application scenarios. In some cases, we generate a static scheduling policy, where the scheduling parameters are fixed, in others a dynamic policy, where they are subject to change depending on the delay accumulated during the execution. The two key enablers for this approach are the adoption of an analytically tractable notion of QoC (2-GAS of Chapter 2) and of a predictable model of computation. The latter, in combination with a soft real–time scheduling algorithm ([176, 2]), allows us to assign scheduling parameters so that delays evolve according to the model $\mathcal{F}_D$ resulting from the optimisation.

### 5.1.1 Related work and comparison

A first piece of work this chapter is related to is [48]. In this interesting paper, Cervin et al. propose a feedback scheduler capable of adapting the sampling frequency of the tasks on the basis of different types of analytical dependence between QoC and frequency. More recently, Buttazzo et al. [45] suggest to re-modulate the sampling periods when the system undergoes an overload condition. Our work differs from these in two essential aspects: 1) the scheduling mechanism, which in our case adapts the bandwidth leaving the period unchanged, 2) the evaluation of the QoC, which in our case is rooted in system theoretic properties.

Even more closely connected is the approach developed in [140]. The authors take to an interesting development their previous work [137], by considering a $(n, m)$ firm real–time tasking model: in each window of $n$ tasks at least $m$ complete within their deadline. Given a QoC optimisation problem, they derive a Markov chain describing the evolution of the $(n, m)$ model and propose a heuristic scheduling algorithm that approximates this evolution. The approach presented in this chapter differs, however, for a couple of reasons. First, we do not consider an all-or-nothing model of execution, where a job is forced to terminate within the deadline or dropped. We allow for a job

to complete after the deadline with a bounded delay. Second, our scheduling algorithm determines an evolution of the delays which is exactly captured by the $\mathcal{F}_D$ model, and is not a heuristic approximation.

## 5.2 PLATFORM MODEL

In this chapter we consider a scenario very close to that of Chapter 4. We have a set of tasks $\mathcal{T} = \{\tau_i\}$ $i \in \{1, \ldots, n_\tau\}$, $n_\tau \in \mathbb{Z}_{>1}$ sharing the same computation platform. Some elements of $\mathcal{T}$ are *real–time tasks* (which include the tasks used for control purposes), the others are *best–effort tasks*. The former have temporal constraints on their execution, while the latter do not receive any kind of temporal guarantees: their execution is optional but adds value to the quality perceived by the user. A real–time task $\tau_i$ consists of a stream of jobs $J_{i,j}$, $j \in \mathbb{Z}_{\geq 0}$. Each job $J_{i,j}$ is activated at time $r_{i,j}$ and finishes at time $f_{i,j}$ after executing for a time $c_{i,j}$. Job $J_{i,j}$ is also characterised by a deadline $d_{i,j}$, that is respected if $f_{i,j} \leq d_{i,j}$ and is missed if $f_{i,j} > d_{i,j}$. We focus on *periodic* tasks with *task period* $T_i$ (i.e. $r_{i,j+1} = r_{i,j} + T_i$), where each activation time is also the deadline of the previous instance (i.e. $d_{i,j} = r_{i,j+1}$). The task is said *hard* real–time if all deadlines are met, and is said *soft* real–time if deadlines are met with a given probability. The quantity $U_i = \max_j \frac{c_{i,j}}{T_i}$ is defined *worst case utilisation*.

### 5.2.1 Scheduling algorithm

The model of scheduler adopted in this chapter is still based on the Constant Bandwidth Server (CBS) [2] as that of the previous chapter and it also aims to allocate a bandwidth $B_i$ to a task $\tau_i$, provided that the full occupation constraint

$$\sum_i B_i \leq 1 \tag{5.2.1}$$

is respected. However, the version of scheduler we consider here is more refined as it ensures a finer granularity in the choice of the tuneable parameters. In particular, we assume a scheduler employing the *Resource Reservations (RR)* policy. Resource Reservations are shortly explained as follows. Each task $\tau_i$ is associated with a reservation pair $(Q_i, R_i)$, meaning that it is allowed to execute for $Q_i$ (*budget*) time units in every interval of length $R_i$ (*reservation period*). Clearly, the budget $Q_i$ has to be chosen in the range $\{0, \ldots, R_i\}$. The bandwidth allocated to the task is defined as $B_i = Q_i/R_i$ and corresponds to the fraction of CPU time allocated to the task. The reservation period $R_i$ is typically chosen as an integer sub-multiple of the task period: $T_i = N_i R_i$, $N_i \in \mathbb{Z}_{>0}$.

#### *Choosing the reservation parameters*

When condition (5.2.1) is respected, the CBS enjoys the *temporal isolation property*, meaning that temporal guarantees can be given to each task based on its computation requirements and on its scheduling parameters, regardless of the parameters of the other tasks in the system. Therefore, the policy used to choose the reservation parameters

$(Q_i, R_i)$ depends on the real–time constraints of the task and on its computation requirements.

A first possibility is to use a *static policy*, whereby such parameters are chosen once and for all and kept constant throughout the execution of the task. For instance, for an hard real–time task, we can guarantee the respect of all its deadlines as long as it is assigned a bandwidth greater than or equal to its worst case utilisation [2]: $B_i \geq U_i$. For a soft real–time task with known distributions of the computation time, we can compute a bound for the probability that it will meet its deadlines for any choice of the budget and of the reservation period [3].

A second possibility is to use a *dynamic policy*, where the bandwidth is changed in every job to adapt the QoS provided by the task. In principle, it is possible to change the bandwidth by operating on the budget and/or on the reservation period. However, for periodic tasks, the reservation period is set to an integer sub-multiple of the task period and adaptations are made by changing the budget $Q_i$. This particular choice is mandated by efficiency reasons because it reduces the waste of computing power in the scheduling process [61]. The adaptation of the budget is typically made by a feedback control loop, where measurements collected inside the OS are used to assess the QoS and to fine-tune it as required. For this reason such policies are usually referred to as *adaptive reservations* [162]. In the following we will provide a concrete example of such a policy.

As far *best effort* activities are concerned, the CBS can be complemented with additional mechanisms that enable an effective *reclaiming* of unused bandwidth. As discussed in Section 5.4, an effective solution of this kind can be built relying on an adaptive reservation mechanism for the real–time tasks.

### 5.2.2 QoS model for the control task

We now focus on a specific task implementing the controller, which will be denoted by $\tau$ removing the $i$ subscript. Our objective is now to construct a QoS model $(T, \mathcal{F}_D)$ for the control task that describes the evolution of the delays introduced in its computations for different choices of the scheduling parameters. In our framework, the control task is soft real–time and shares the CPU with the other tasks in the set $\mathcal{T}$.

The task is scheduled by a reservation $(Q, R)$ and, in view of the temporal isolation property, evolves independently of the other tasks. For the choice of $Q$, we will consider dynamic policies (in which the budget $Q_j$ changes for every job $J_j$) and static policies. The model below is derived for the case of dynamic policies, and the case of static policies is easily recovered by imposing the constraint $Q_j = Q_{j-1}$ for all $j$. The task $\tau$ is activated with period $T = NR$; each activation produces a job $J_j$ that takes a sample $y_j$ of the plant output and produces the control value $u_j$. As done in Section 4.2.2, we construct a QoS model in terms of the instant $s_j$ when $y_j$ is sampled (*sampling time*) and the instant $v_j$ when $u_j$ is released (*release time*).

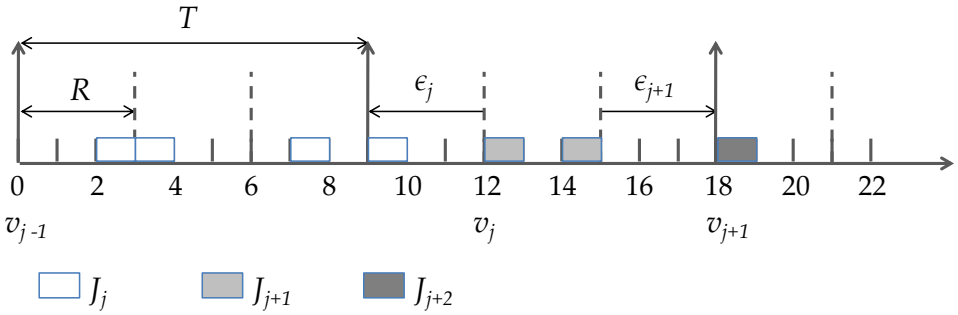


FIGURE 5.2.1: Example of schedule of a task ($T = 9$, $R = 3$, $N = 3$).

### *Model of computation*

In order to keep in check the jitter and the computation time variability, we introduce the following set of rules:

*Rule 1:* If a job $J_j$ finishes *before* its deadline (i.e. $f_j < d_j = r_j + T$), the release of the output $u_j$ is deferred to the end of the period (i.e. $v_j = r_j + T$ is forced).

*Rule 2:* If a job $J_j$ finishes *after* its deadline (i.e. $r_j + T + \left(D_j - 1\right) R < f_j \leq r_j + T + D_j R$ with $D_j \in \mathbb{Z}_{\geq 0}$), the release of the output $u_j$ takes place immediately at the end of the present reservation period (i.e. $v_j = r_j + T + D_j R$ is enforced and $D_j$ is the delay of job $J_j$).

*Rule 3*: If a job $J_j$ experiences a delay $D_j$ greater than a threshold, it is cancelled and a new job $J_{j+1}$ is activated.

*Rule 4:* The output $y_j$ is sampled at the same time the control $u_{j-1}$ is released (at instant $s_j = v_{j-1}$).

The Rule 1 and Rule 4 are identical to the Rule 1 and Rule 3 adopted in Chapter 4. As mentioned in the Introduction, we want to extend the previous model of computation by allowing the cumulation of a finite delay due to late completion of the control task. To this aim, we replaced Rule 2 of Chapter 4 (i.e. the cancelling rule) with the present Rule 2 and Rule 3.

The purpose of Rule 1 is to reduce the output jitter due to early terminations. As an illustrative example, consider the schedule in Figure 5.2.1. The job $J_{j+1}$ finishes at time 15 but releases its output only at time 18. Rule 2, together with Rule 1, prevents output releases that are not aligned with a reservation period. As a consequence the delay experienced by a job is always an integer multiple $D_j$ of the reservation period. With reference to Figure 5.2.1, job $J_j$ finishes with a delay $D_j = 1$. Rule 3 avoids the accumulation of very large delays which are impossible to be compensated. In what follows we assume that the maximum tolerated delay is one task period $T = NR$ (i.e. $\max D_j = N$). As a consequence of Rule 1, Rule 2 and Rule 3 the release time for the output of $J_j$ is given by

$$v_j = r_j + T + D_j R = (j+1)T + D_j R, \tag{5.2.2}$$

with $D_j \in \{0, 1, \ldots, N\}$. In view of such expression for $v_j$, Rule 4 ensures that output sampling always occurs at the beginning of a reservation period, even if the task can receive its budget later in the same reservation period (the job $J_j$ in Figure 5.2.1 samples $y_j$ at time 0, but starts at time 2).

A few remarks on this model of computation are in order. If the computation consistently terminates within the deadline, the application of the rules recovers the standard time–triggered behaviour [121, 108], with periodic sampling and a fixed delay equal to $T$ in the computation. On the contrary, when the task execution gets delayed, Rule 4 allows a more flexible management of the sampling instants than a time–triggered approach: when the next job starts its execution, it is given the possibility of using fresher data. For instance, job $J_{j+1}$ in Figure 5.2.1 will use the data sampled at time 12 rather than the one sampled at time 9. One potential drawback of sampling on $v_{j-1}$ instead of sampling on $r_j$ is that the classical analysis based on periodic updates for $y_j$ and $u_{j-1}$ is not applicable. However, as discussed in the next section, this problem is easy to solve and its importance is outweighed by the advantages. In some respects, we could say that the periodic activation is used to set a periodic reference pace for the control task. However, the communication between the control task and the environment is event–triggered, and the communication events are generated the earliest instant between the periodic activation of the job and the termination of the previous one. As a final remark, this model of computation is easy to implement using the techniques proposed in the literature on time–triggered systems on top of an operating system equipped with a CBS scheduler [108, 121, 162]. We will not delve into these implementation details for the sake of brevity.

### *Evolution of the delays*

Let $Q_j$ denote the budget allocated to job $J_j$. The reservation period $R$ is held constant (and equal to an integer submultiple of the task period $T$) to decide the granularity of the CPU allocation (a smaller value for $R$ corresponds to a more fluid allocation but to a greater OS overhead). The temporal evolution of the task is described by introducing a state variable, called *scheduling error*, given by the difference between the output release time $v_j$ and the soft deadline $d_j$:

$$\epsilon_j \triangleq v_j - d_j = v_j - r_{j+1}. \tag{5.2.3}$$

As an example, the job $J_j$ in Figure 5.2.1 uses four reservation periods to complete. So its output release time is $v_j = 12$ and the resulting scheduling error $\epsilon_j = 3$ time units, i.e. a reservation period $R$. A positive value for $\epsilon_j$ means that $J_j$ finished beyond the deadline, i.e. it received less bandwidth than it needed. Conversely, a negative value means that it finished before its deadline (the assigned bandwidth was greater than the task needed, as happened to job $J_{j+1}$ in Figure 5.2.1).
The dynamic evolution of the scheduling error can be found in [162]. The combination of such dynamics with the rules of the model of computation produces the following

expression for the evolution of the delay $D_j$:

$$D_{j+1} = \frac{\epsilon_{j+1}}{R} = S(D_j) + \left\lceil \frac{c_{j+1}}{Q_{j+1}} \right\rceil - N, \tag{5.2.4}$$

where $S(x) \triangleq \max(\min(x, N), 0)$. The loop delays evolve in a bounded and discrete set: $D_j \in \{0, 1, \ldots, N\}$. In this model the computation time $c_{j+1}$ (which is unknown before the execution of the job) plays the role of an exogenous disturbance term. For the case of dynamic policies, $Q_j$ can be seen as an input variable and used to control the evolution of the delays.

## 5.3 CONTROL PROBLEM

The control task $\tau$ is used to control a linear time–invariant and strictly proper plant. We can describe the plant by a discrete–time model obtained using the reservation period $R$ as sampling time[1]:

$$x(k+1) = Ax(k) + Bu(k) \tag{5.3.1}$$

with $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $k \in \mathbb{Z}_{\geq 0}$. The control $u(k)$ is held constant between two successive release times, that is $u(k) = u_{j-1}$ for $k \in \{v_j/R, \ldots, v_{j+1}/R\}$, where $u_{j-1}$ is the control value released at $v_j$. The output $y_j$ sampled at $v_j$ is given by $y_j = Cx\left(v_j/R\right)$ with $y \in \mathbb{R}^{n_y}$ and $C \in \mathbb{R}^{n_y \times n_x}$ output matrix of the plant (5.3.1). The task $\tau$ implements the following linear controller

$$\begin{aligned} z_{j+1} &= A_c z_j + B_c y_j \\ u_j &= C_c z_j + D_c y_j, \end{aligned} \tag{5.3.2}$$

where $j$ is the index of the $j$-th job releasing its output at $v_j$, $z_j \in \mathbb{R}^{n_z}$, $A_c \in \mathbb{R}^{n_z \times n_z}$, $B_c \in \mathbb{R}^{n_z \times n_y}$, $C_c \in \mathbb{R}^{n_u \times n_z}$ and $D_c \in \mathbb{R}^{n_u \times n_y}$. The controller (5.3.2) does not evolve on a periodical time basis, as it is updated at each new release time instant of $\tau$ and $v_j - v_{j-1}$ is not constant for $j \in \mathbb{Z}_{>0}$. Nonetheless, such a controller is designed assuming *a nominal periodic behaviour* with period $T = NR$, computation delay equal to the period, and a ZoH semantics. Let $F_j \in \{0, \ldots, 2N\}$ denote the integer variable describing the number of reservation periods in which the control value $u_{j-1}$ is held constant. Recalling (5.2.2), it is immediate to see that $F_j \triangleq \frac{v_j - v_{j-1}}{R} = N + D_j - D_{j-1}$. In the nominal condition where both jobs $J_{j-1}$ and $J_j$ do not suffer delays we have $F_j = N$. According to this definition, the controlled system dynamics between the release time of job $J_{j-1}$ and the release time of job $J_j$ is given by

$$\begin{aligned} x_{j+1} &= A^{F_j} x_j + \sum_{t=0}^{F_j - 1} A^{F_j - t - 1} B u_{j-1} \\ y_j &= C x_j, \end{aligned} \tag{5.3.3}$$

[1]Indeed, according to the rules in the previous section, each output sampling and control release operation takes place at some release time $v_j$, which is always an integer multiple of the reservation period.

where $x_j \triangleq x\left(v_j/R\right)$. If the delay is greater than $N$, then a drop event takes place limiting the delay to $N$ (Rule 3). The drop event can be managed either by holding the previous control value (drop and hold), or by zeroing it (drop and zero). In both cases we consider the controller state $z_j$ to be held ($z_j = z_{j-1}$). Therefore, in the drop case, the dynamics of the controller (5.3.2) has to be modified, while the dynamics (5.3.3) remains unchanged. In order to model the constant delay (of $N$ reservation periods) and the drop event, we introduce the state variable $\zeta_j \in \mathbb{R}^{n_u}$. In this paper, we adopt the drop and hold semantics, with the following dynamics for the controller:

$$\underbrace{\begin{aligned} z_{j+1} &= A_c z_j + B_c y_j \\ \zeta_{j+1} &= C_c z_j + D_c y_j \\ u_j &= C_c z_j + D_c y_j \end{aligned}}_{\text{no drop}} \quad \text{and} \quad \underbrace{\begin{aligned} z_{j+1} &= z_j \\ \zeta_{j+1} &= \zeta_j \\ u_j &= \zeta_j \end{aligned}}_{\text{drop and hold}}, \tag{5.3.4}$$

The drop and zero case is obtained by simply replacing $\zeta_{j+1} = \zeta_j$, $u = \zeta_j$ with $\zeta_{j+1} = u_j = 0$.

The resulting closed loop system has the following switching dynamics

$$\begin{aligned} \xi_{j+1} &= \tilde{A}_{\phi(j)} \xi_j \\ y_j &= \tilde{C} \xi_j \end{aligned} \tag{5.3.5}$$

where $\xi_j = \left[x_j^T, z_j^T, \zeta_j\right]^T \in \mathbb{R}^{n_x+n_z+n_u}$ and $\tilde{C} = \left[C, 0, 0\right] \in \mathbb{R}^{n_y \times (n_x+n_z+n_u)}$. The piecewise constant function $\phi : \mathbb{Z}_{\geq 0} \to \{0, \ldots, 3N+1\}$ rules the switchings among the different subsystems according to the delay evolution (5.2.4) and the drop policy. Indeed, we have $2N+1$ possible values of $F_j$ for the case of regular evolution (no drop event, i.e. $\phi(j) = 0, \ldots, 2N$), and each of them generates a possible closed loop dynamics. Additionally, we have to account for $N+1$ dynamics stemming from a drop event (i.e. $\phi(j) = 2N+1, \ldots, 3N+1$). The expression of matrices $\tilde{A}_{\phi(j)}$ can be easily derived combining (5.3.3) and (5.3.4). For $\phi(j) = 0, \ldots, 2N$, we get $2N+1$ closed loop matrices for the regular dynamics

$$\tilde{A}_{\phi(j)} = \begin{bmatrix} A^{F_j} & 0 & \tilde{B}_{\phi(j)} \\ B_c C & A_c & 0 \\ D_c C & C_c & 0 \end{bmatrix}, \tag{5.3.6}$$

with $\tilde{B}_{\phi(j)} = \sum_{t=0}^{F_j-1} A^{F_j-t-1} B$ and $F_j = 0, \ldots, 2N$. Additionally, we have $N+1$ dynamics after a drop and hold event given by indices $\phi(j) = 2N+1, \ldots, 3N+1$ that generate

$$\tilde{A}_{\phi(j)} = \tilde{A}^{\mathrm{dh}}_{\phi(j)} = \begin{bmatrix} A^{F_j} & 0 & \tilde{B}_{\phi(j)} \\ 0 & I & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.3.7}$$

with $F_j = N, \ldots, 2N$ and the $\tilde{B}_{\phi(j)}$ is formally equal to the one given previously. The drop and zero matrix $\tilde{A}^{\mathrm{dz}}_{\phi(j)}$ can be obtained by $\tilde{A}^{\mathrm{dh}}_{\phi(j)}$ zeroing the last element of the last row.

### 5.3.1 Problem formulation

We are now able to state the following problem:

**Problem 5.1.** Design an *adaptive reservation* such that: 1) the QoS $(T, \mathcal{F}_D)$ that it offers to the control process is sufficient to sustain a QoC specification, 2) the bandwidth devoted to the control task is minimised.

QoC and bandwidth consumption are in evident trade-off. Indeed, if the bandwidth devoted to the control task $\tau$ is permanently greater than the worst case utilisation, all jobs complete within their deadline [2], and the controlled system always evolves with its nominal dynamics $\tilde{A}_N$ ($F_j = N$ for all jobs $J_j$) where the QoC specifications are respected by design. Conversely, if the bandwidth granted to $\tau$ is small, the jobs are often dropped and the system ends up executing with its open loop dynamics ($F_j = 2N$) most of the times.

## 5.4 STOCHASTIC MODEL

The sequence of the computation times of the control task $\tau$ is modelled as an independent and identically distributed (i.i.d.), real–valued, continuous stochastic process denoted by $\{c_j\}_{j\in\mathbb{Z}_{\geq 0}}$. The random variable $c_j$ takes values in the set $L_c = \{\underline{c}, \dots, \overline{c}\}$, where $\overline{c}$ is the worst case execution time (WCET) and $\underline{c}$ is the best case execution time (BCET) and is distributed according to the probability density function (pdf) $f(c_j)$. From (5.2.4), it follows that $\{D_j\}_{j\in\mathbb{Z}_{\geq 0}}$ is itself a stochastic process that is related to the process $\{c_j\}_{j\in\mathbb{Z}_{\geq 0}}$, the budget $Q_j$ and the value of the previous delay $D_{j-1}$. Contrary to $\{c_j\}_{j\in\mathbb{Z}_{\geq 0}}$ this process is discrete valued. In the QoS model, $\mathcal{F}_D$ is defined by the set of all possible values of the delay ($\{0, \dots, N\}$) and by the associated probability.

In order to attain the desired QoC goals, we look for a time invariant dynamic policy for the QoS $(T, \mathcal{F}_D)$, with the budget chosen as a function of the delay experienced in the previous job $Q_j = Q\{D_{j-1}\}$ (which, as a special case, can be constant). With this choice and assuming the process $\{c_j\}_{j\in\mathbb{Z}_{\geq 0}}$ stationary (it is indeed i.i.d.), the process describing the evolution of $D_j$ and also the drop events is a finite-state homogeneous, discrete-time Markov chain (FSH-MC). Denote by $\{\sigma(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ such a MC, which takes on values in the set $L_\sigma = \{0, \dots, N+1\}$, with the meaning that if $\sigma(j) = q$, $q \in \{0, \dots, N\}$ then $D_j = q$, if $\sigma(j) = N+1$ then $D_j = N$ and a drop event takes place. The stochastic characterisation of $\{\sigma(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ is given by the transition probability matrix $P = (p_{qg})_{(N+2)\times(N+2)}$, $p_{qg} \triangleq \mathsf{P}\{\sigma(j+1) = g \mid \sigma(j) = q\}$ and by the initial probability measure $\pi_\sigma(0) \in \mathcal{S}^{N+1}$, where $\mathcal{S}^{N+1}$ is, as usual, the $(N+1)$-dimensional canonical stochastic simplex. The evolution of the probability distribution $\pi_\sigma(\cdot)$ of the MC $\{\sigma(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ is then given by $\pi_\sigma(j+1) = \pi_\sigma(j)P$.

Recalling (5.2.4), for every $q$ and $g$ such that $0 \leq q \leq N$ and $1 \leq g \leq N$, we have

$$p_{qg} = \mathsf{P}\left\{g = q + \left\lceil \frac{c_{j+1}}{Q\{q\}} \right\rceil - N\right\}$$

and

$$p_{qg} = \mathsf{P}\left\{c_{j+1} \in V(q,g)\right\} = \int_{V(q,g)} f(x)dx,$$

where the set $V(q,g) \triangleq (\underline{a}_{qg}, \overline{a}_{qg}]$, and the values $\underline{a}_{qg} \triangleq \min\left(Q\{q\}(g-q+N-1), \bar{c}\right)$ and $\overline{a}_{qg} \triangleq \min\left(Q\{q\}(g-q+N), \bar{c}\right)$.

A special case is related to the probability to finish with a state $g = 0$ associated with the event $D_{j+1} = 0$ (i.e., the control task terminates before the deadline). In this case, due to the presence of the $S(\cdot)$ operator, we can compute $V(q,0) \triangleq (0,\ \min\left(Q\{q\}(N-q), \bar{c}\right)]$. From this expression, we can easily see that by increasing the budget $Q\{q\}$ we can increase the probability of meeting the deadline, whilst a large accumulated delay $q$ plays adversely. Another important special case is related to the state $g = N+1$ (corresponding to a drop event), with the probability $p_{q\,N+1}$ given by:

$$p_{q\,N+1} = \mathsf{P}\left\{q + \left\lceil \frac{c_{j+1}}{Q\{q\}} \right\rceil - N > N\right\},$$

resulting into $V(q,N+1) \triangleq \left(\min\left(Q\{q\}(2N-q), \bar{c}\right), \bar{c}\right]$ for $0 \le q \le N$. The set $V(q,N+1)$ can be empty if $\min\left(Q\{q\}(2N-q), \bar{c}\right) = \bar{c}$.

An important question is ascertaining whether the FSH-MC $\sigma$ is also irreducible and aperiodic (FSHIA-MC), that is if there exists a unique invariant probability distribution (i.p.d.) $\bar{\pi}_\sigma$ corresponding to the steady-state probability distribution of the MC (i.e. $\lim_{h\to\infty} \pi_\sigma(h) = \bar{\pi}_\sigma$ for any $\pi_\sigma(0)$). Conditions for this are stated in the following.

**Lemma 5.1.** *The FSH-MC $\sigma$ is irreducible and aperiodic if*

$$\begin{aligned} &0 < Q\{q\} < \left\lfloor \frac{\bar{c}}{N} \right\rfloor,\ \forall q \in \{0,\dots,N\}, \\ &0 < Q\{N+1\} < \left\lfloor \frac{\bar{c}}{N-1} \right\rfloor. \end{aligned} \tag{5.4.1}$$

### *Stochastic description of the switching process*

In this section we derive a stochastic description for the process $\{\phi(j)\}_{j\in\mathbb{Z}_{\ge 0}}$ ruling the switchings of the closed loop system (5.3.5), which is thus a Stochastic Jump Linear System (SJLS). The process $\{\phi(j)\}_{j\in\mathbb{Z}_{\ge 0}}$ takes values in the set $L_\phi = \{0,\dots,3N+1\}$ as follows:

1. $\phi(j) = N - \sigma(j-1) + \sigma(j)$ for $\sigma(j-1), \sigma(j) < N+1$;
2. $\phi(j) = 2N - \sigma(j-1)$ for $\sigma(j-1) < N+1$ and $\sigma(j) = N+1$;
3. $\phi(j) = 2N+1+\sigma(j)$ for $\sigma(j-1) = N+1$ and $\sigma(j) < N+1$;
4. $\phi(j) = 3N+1$ for $\sigma(j-1), \sigma(j) = N+1$.

According to this definition, we can define for each $i \in L_\phi$ a set of pair(s) $O_i = \{(a,b) \in L_\sigma \times L_\sigma \mid \sigma(j-1) = a,\ \sigma(j) = b \text{ s.t. } \phi(j) = i\}$. Hence we can write

$$\pi_{\phi_i}(j) = \mathsf{P}\{\phi(j) = i\} = \sum_{(a,b)\in O_i} \mathsf{P}\{\sigma(j) = b, \sigma(j-1) = a\}. \tag{5.4.2}$$

In order to provide the stochastic characterisation of $\phi$, we define another process describing the evolution of two consecutive steps of the MC $\sigma$: $\hat{\sigma}(j) = (\sigma(j), \sigma(j-1))$, taking values in the set $L_{\hat{\sigma}} = L_\sigma \times L_\sigma$. Hence we have

$$\begin{aligned}\hat{\pi}_{ab}(j) &\triangleq \mathsf{P}\{\hat{\sigma}(j) = (a,b)\}\\ &= \mathsf{P}\{\sigma(j) = b,\ \sigma(j-1) = a\}\\ &= \mathsf{P}\{\sigma(j) = b \mid \sigma(j-1) = a\}\mathsf{P}\{\sigma(j-1) = a\}\\ &= p_{ab}\pi_{\sigma_a}(j-1).\end{aligned}$$

Recalling the definition of $\pi_\sigma$ we have $\pi_{\sigma_a}(j-1) = \sum_{c=0}^{N+1} p_{ca}\pi_{\sigma_c}(j-2) = \sum_{c=0}^{N+1} \hat{\pi}_{ca}(j-1)$, and

$$\hat{\pi}_{ab}(j) = p_{ab}\sum_{c=0}^{N+1} \hat{\pi}_{ca}(j-1) = \hat{\pi}(j-1)v_a p_{ab}, \tag{5.4.3}$$

with $\hat{\pi} \triangleq [\hat{\pi}_{00}, \hat{\pi}_{01}, \ldots, \hat{\pi}_{N+1,N}, \hat{\pi}_{N+1,N+1}]$ and $v_a$ is the $a$-th column of the matrix $V \in \{0,1\}^{(N+2)^2\times(N+2)}$ given by $V = [I_{N+2}, \ldots, I_{N+2}]^T$. Equation (5.4.3) can be written as

$$\hat{\pi}(j) = \hat{\pi}(j-1)\hat{P}, \tag{5.4.4}$$

with $\hat{P} \triangleq [v_0 p_{00}, v_0 p_{01}, \ldots, v_{N+1}p_{N+1,N+1}]$, thus revealing that $\hat{\sigma}$ is a FSH-MC. A relevant issue is establishing if $\hat{\sigma}$ admits a unique i.p.d.. A potential problem is that $\hat{\sigma}$ is not always irreducible even if $\sigma$ is irreducible, which could lead to zero steady state probability for some initial state of $\hat{\sigma}$. However, the following proposition holds.

**Proposition 5.1.** *If $\sigma$ is a FSHIA-MC with unique steady state distribution $\bar{\pi}_\sigma = [\bar{\pi}_{\sigma_0}, \bar{\pi}_{\sigma_1}, \ldots, \bar{\pi}_{\sigma_{N+1}}]$ and transition probability matrix $P = (p_{ab})_{(N+2)\times(N+2)}$, then $\hat{\sigma}$ is a FSH-MC with a unique steady state distribution $\hat{\bar{\pi}} = [\hat{\bar{\pi}}_{00}, \hat{\bar{\pi}}_{01}, \ldots, \hat{\bar{\pi}}_{N+1,N}, \hat{\bar{\pi}}_{N+1,N+1}]$ with $\hat{\bar{\pi}}_{ab} = p_{ab}\bar{\pi}_{\sigma_a}$.*

The process $\{\phi(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ is not a FSH-MC, but its distribution $\pi_\phi(j)$ is linearly related to the one of the FSH-MC $\hat{\sigma}$. Indeed, recalling (5.4.2) we can write

$$\pi_\phi(j) = \hat{\pi}(j)W, \tag{5.4.5}$$

for a suitable matrix $W = (w_{lh})_{(N+2)^2\times(3N+2)}$, with $w_{lh} \in \{0,1\}$. Therefore, the probability distribution of the joint process $\{\hat{\sigma}(\cdot), \phi(\cdot)\}$ evolves according to the dynamics (2.2.3)-(2.2.4) introduced in Chapter 2, where $\hat{P}$ and $W$ play the role of $P$ and $L$ respectively. The process $\{\phi(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ admits a unique steady state distribution $\bar{\pi}_\phi$ which, recalling Proposition 5.1 and Chapter 2, is given by

$$\bar{\pi}_\phi = \hat{\bar{\pi}}W, \tag{5.4.6}$$

where $\hat{\bar{\pi}}$ is the steady state probability of the FSH-MC $\hat{\sigma}$.

### 5.4.1 The notion of QoC

The description of process $\{\phi(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ allows us to establish a link between the QoS model $(T, \mathcal{F}_D)$ and the QoC. We use the system stability as our notion of QoC. To this end, we adopt the 2-GAS (sometimes referred to as second moment stability) presented in Definition 2.1. We recall that, in the light of Theorem 2.1, 2-GAS, 2-GES, 2-ISS and 2-EISS are all equivalent, thus ensuring also robustness w.r.t. a locally bounded exogenous disturbance. In order to ensure 2-GAS (or equivalently 2-GES, 2-ISS, 2-EISS), we will use the sufficient condition proved in Theorem 2.2.

### 5.4.2 Optimal Reservation Policies

We are now in condition to solve Problem 5.1. Let us define the stochastic process $\{q(j)\}_{j\in\mathbb{Z}_{\geq 0}}$ as the sequence of budgets assigned to the control task $\tau$ by the scheduler. Such a process takes values in the set $\{Q\{0\},\dots,Q\{N+1\}\}$ representing all the possible state dependent budget values. All these values can be stacked in the optimisation vector $Q \triangleq [Q\{0\},\dots,Q\{N+1\}]^T$. The solution set of the optimisation problem is given by the constraints (5.4.1) on the values of the budget and by the 2-GAS condition of Theorem 2.2. Assuming that a fraction of each reservation period $R$ has to be reserved for the execution of other tasks than $\tau$, then each $Q\{q\}$ must be less than a pre-specified $Q_{\max}$. Therefore, defining the vectors $\underline{Q} \triangleq [0,\dots,0]^T$ and $\overline{Q} \triangleq [\alpha,\dots,\alpha,\beta]^T$, where $\alpha \triangleq \min\left(\left\lfloor \frac{\bar{c}}{N}\right\rfloor, Q_{\max}\right)$ and $\beta \triangleq \min\left(\left\lfloor \frac{\bar{c}}{N-1}\right\rfloor, Q_{\max}\right)$, the budget is constrained by: $\underline{Q} < Q < \overline{Q}$. Problem 5.1 can be formalised by the following nonlinear optimisation program, the *Optimal Reservation Policy - 2-GAS (ORP-2GAS)* problem in the vector $Q$ and the $(N+2)^2$ matrices $G_l$:

*ORP-2GAS Problem*

$$
\begin{aligned}
&\min_{Q} g(\bar{\pi}_\sigma, Q) \text{ s.t.}\\
&\quad \underline{Q} < Q < \overline{Q}\\
&\quad G_l = G_l^T > 0\\
&\quad \sum_{h=0}^{3N+1} w_{lh}\tilde{A}_h^T\hat{G}_l\tilde{A}_h - G_l < 0, \qquad \forall l \in L_{\hat{\sigma}}\\
&\quad \hat{G}_l = \sum_{i=0}^{(N+2)^2-1} \hat{p}_{li}(Q)G_i\\
&\quad \hat{P}(Q) = (\hat{p}_{li}(Q))_{(N+2)^2\times(N+2)^2} \text{ as in (5.4.4)}\\
&\quad W = (w_{lh})_{(N+2)^2\times(3N+2)} \text{ as in (5.4.5).}
\end{aligned}
$$

The computation budget is usually chosen as an integer multiple of a time interval (which is dictated by the granularity of the clock used in the OS). For this reason the problem ORP-2GAS is essentially a mixed integer optimisation problem which can, in principle, be solved by exhaustively searching on all the possible combination of

discrete values for the vector of budget $Q$. For a given choice of the elements of the vector $Q$, the problem reduces to an LMI feasibility check, which is convex and can be solved very efficiently.

### 5.4.3 Optimisation goals

The ORP-2GAS solution depends on the cost index $g(\bar{\pi}_\sigma, Q)$, which represents the bandwidth minimisation criterion according to the second point of Problem 5.1. For a real–time platform, we identify two main problems: 1) given the control task set $\mathcal{T} = \{\tau_i\}$, with $i \in \{1, \ldots, n_\tau\}$, maximise the number of tasks $n_\tau$ that can be hosted on the platform; 2) maximise the amount of CPU time allotted to other best effort (non critical) activities sharing the platform.

#### 5.4.3.1 Maximise $n_\tau$

In order to maximise the number of control tasks that can be executed on the platform, we need to minimise the *maximum* bandwidth that each one uses. This way we can comply with the schedulability constraint (5.2.1) for a larger number $n_\tau$. In formal terms, we can optimise w.r.t. the following cost function $g(\bar{\pi}_\sigma, Q) = \|Q\|_\infty$. When the purpose is to minimise the maximum bandwidth, dynamic adaptation of the bandwidth is immaterial, hence a *static* scheduling policy is sufficient. Therefore, the vector $Q$ of decision variables reduces to a scalar.

#### 5.4.3.2 Optimise for best effort tasks

A best-effort task is piece of code that makes the most of the computing power received. Examples of such tasks without real-time requirements are code execution profiling algorithms, supervising and security programs, platform hardware management tasks, etc. However, in the realm of digital processing and control, imprecise computation approaches [143, 144] or Anytime Control algorithms, such as those presented in Chapter 3, can be considered as best-effort tasks. More precisely, best-effort tasks are the lowest priority tasks in the platform that execute whenever possible.

For best-effort tasks, our goal is to maximise the chance of execution whenever possible, i.e. whenever the processor is not utilised by tasks receiving real–time guarantees (first and foremost the control task). To this end, we need a reclaiming mechanism operating hand-in-hand with the CBS scheduler. There are several solutions available in the literature. The one that we advocate here lies in the track opened by Cucinotta et al. [60]. The solution is compounded of an adaptive mechanism in the scheduler (an adaptive reservation) and of an additional mechanism to redistribute the unused bandwidth to the best effort activities. The purpose of the adaptive reservation is to track the computing requirements of the control task (and of other soft real–time tasks) and to allocate it a bandwidth very close to its actual needs.

An adaptive scheduler of this kind can be obtained, in our setting, by solving the ORP-2GAS problem with a cost index that minimises the mean bandwidth allocated to

the control task. A suitable cost index, which provides a stochastic average of the budget provided by the scheduler, is the long-run expected value of the budget itself. Assuming $\sigma$ is a FSHIA-MC, the cost index can be written as $g(\bar{\pi}_\sigma, Q) = \lim_{j\to\infty} \mathrm{E}\{q(j)\} = \bar{\pi}_\sigma(Q)Q$. By definition $\bar{\pi}_\sigma$ is a function of the budget $Q$, hence this is a nonlinear cost index.

The solution of the optimisation problem produces in this case a *dynamic* scheduling policy, where the budget is adjusted for each job based on the delay measured in the OS. Clearly, the optimisation of the average bandwidth comes at the price of a possible growth of the maximal bandwidth required by the task. This can reduce the number of tasks requiring temporal guarantees (i.e. a minimum availability of bandwidth).

### 5.4.4 Comparison between ORP-2GAS and Problem (4.4.2)

It is interesting to compare the ORP-2GAS problem with Problem (4.4.2) described in Chapter 4 from the point of view of both cost functions and constraints.

The cost functions adopted in problem ORP-2GAS are directly related to the platform parameters $Q$, thus to the QoS. In Problem (4.4.2), instead, the cost function is a measure of the overall QoC (minimum of the maximal variance of the closed loop systems), thus related to the platform bandwidth $B$ only through the dynamics of the SJLSs. In other words, the measure of performance in problem ORP-2GAS is more platform-oriented, while in Problem (4.4.2) is more control-oriented.

The constraints also are quite different. In Problem (4.4.2) the most important constraint is provided by the minimum bandwidth $B_i^{(m)}$ in (4.4.9), that is necessary to ensure the mean square stability of the closed loop systems. In the ORP-2GAS problem the stability of the closed loop systems (in the 2-GAS flavour) is the chosen QoC and it enters the constraints through the condition provided by Theorem 2.2. Moreover, the solution set of Problem (4.4.2) is a polytope computed before the optimisation takes place, while the solution set of the ORP-2GAS problem is in terms of the unknown matrices $G_i$ and is also function of the optimisation vector $Q$, as a consequence it cannot be computed in advance. As a result, the optimisation problem ORP-2GAS is, in general, more difficult to characterise and to solve than Problem (4.4.2). As mentioned in Section 5.4.2, it is a mixed-integer optimisation problem that reduces to a LMI feasibility check for any fixed $Q$.

## 5.5 SIMULATION RESULTS

This section reports simulations that show the effectiveness of the proposed approach in the two scenarios discussed earlier. In all the experiments reported below, we synthesised a Linear Quadratic Gaussian (LQG) controller, using the same weights and assuming a nominal condition with a constant delay of one sampling period (see Rule 1). The computation model presented in Section 5.2.2 is adopted. In the different experiments, we have used computation times given by i.i.d. stochastic processes with

three different distributions: a uniform distribution (UD), an exponential distribution (ED) and a beta distribution (BD).

### *Hosting multiple tasks with real–time guarantees*

In the first set of experiments, we have considered a platform hosting three control tasks $\mathcal{T} = \{\tau_1, \tau_2, \tau_3\}$ controlling three open–loop unstable, observable and controllable plants. The task periods have been fixed to $T_1 = 10$ ms, $T_2 = 2$ ms and $T_3 = 18$ ms, each one being four times the respective reservation periods $R_i$. The BCETs have been chosen as $\underline{c}_1 = 125\mu$s, $\underline{c}_2 = 25\mu$s and $\underline{c}_3 = 225\mu$s. The WCETs have been fixed to a ratio of the reservation periods: $\bar{c}_i = \frac{T_i+R_i}{3} = \frac{5}{3}R_i$.

By using the classic approach the three tasks cannot be executed sharing a single real–time platform. Indeed, the strict respect of every deadline requires an allocation of bandwidth equal to the worst case utilisation [2] ($B_i = \frac{\bar{c}_i}{T_i}$), and this assignment violates the schedulability constraint in (5.2.1) ($\sum_i \frac{\bar{c}_i}{T_i} = \frac{15}{12} > 1$). However, by exploiting the knowledge on the distribution of the computation times, this limit has been overcome. For example, we have chosen the UD and BD defined in the range given by the BCET and WCET, while the ED has been truncated (and re-normalised) in the same range. The mean values $\mu_i$ and standard deviations $\sigma_i$ have been chosen as follows: $\mu_1 = 1.37$ ms, $\mu_2 = 0.27$ ms, $\mu_3 = 2.46$ ms, $\sigma_1 = 0.67$ ms, $\sigma_2 = 0.13$ ms and $\sigma_3 = 1.22$ ms for the BD; $\mu_1 = 1.53$ ms, $\mu_2 = 0.31$ ms, $\mu_3 = 2.75$ ms, $\sigma_1 = 1.06$ ms, $\sigma_2 = 0.21$ ms and $\sigma_3 = 1.9$ ms for the ED. The mean values of UD are $\mu_1 = 2.15$ ms, $\mu_2 = 0.43$ ms and $\mu_3 = 3.86$ ms. By solving the ORP-2GAS problem assuming a *static* allocation of the bandwidth minimising the maximum bandwidth for each task in isolation (see Section 5.4.3.1), the bandwidths allocated to the three tasks were respectively: $B_1 = B_2 = 0.3$ and $B_3 = 0.35$ for UD, $B_1 = B_2 = 0.2$ and $B_3 = 0.3$ for ED, and $B_1 = B_2 = 0.2$ and $B_3 = 0.25$ for BD. As it may be expected, the UD is the distribution that required the highest allocation of bandwidth, since the shape of the distribution is the less favourable. Since the sum of the bandwidths for the three tasks was less than one for any possible combination of the distributions, it was possible to host the three tasks on the same CPU.

The previous simulations show that our approach allows us to schedule a control task even in the face of a computation time greater than the period (utilisation greater than one), which is clearly out of the question in a hard real–time setting. To investigate further on this issue, we have devised a specific experiment where our scheduling proposal is confronted with a naive approach where the task is allowed to execute until the deadline with full bandwidth and, if the execution is not finished, the job is simply dropped (*drop–out* policy). To this end, we designed a linear stabilising controller for a Furuta pendulum with zero offset [93] (as seen also in Chapter 3), whose linearised dynamic is represented by the continuous-time transfer function $G(s) = \frac{7.435}{s(s^2+34.63)}$ between the input torque $\tau$ and the output arm angle $\alpha$. The task period was fixed to $T = 10$ ms (reservation period $R = T/6 = 1.66$ ms). The computation times were given by an i.i.d. process with uniform distribution defined in the range between BCET $\underline{c} = 1.66$ ms and WCET $\bar{c} = 11.66$ ms. In this situation, the single task was not hard

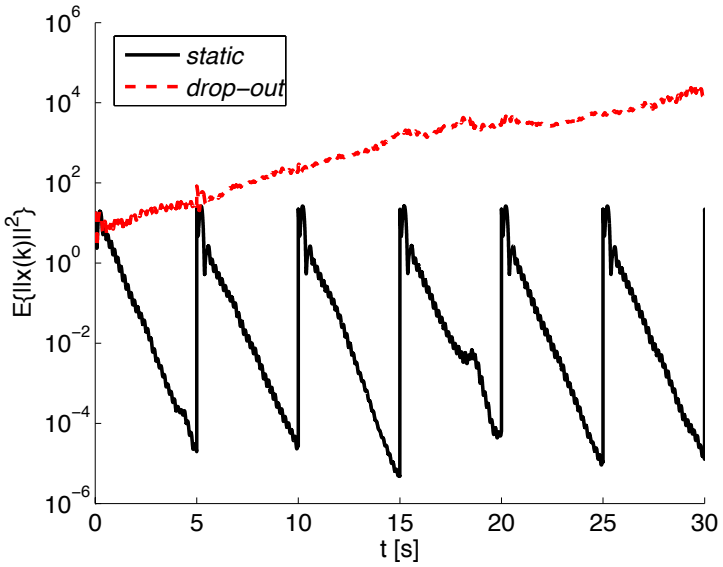


FIGURE 5.5.1: $\mathsf{E}\left\{\|x(k)\|^2\right\}$ of the Furuta pendulum in case of *static* reservation policy (solid) and *drop-out* policy (dashed). The mean is computed over 200 independent runs.

real–time schedulable even if 100% of the computation time were granted. We have carried out simulations using the drop–out policy outlined above and compared it with the *static* reservation policy (Section 5.4.3.1). To show the generality of our approach, we have considered here a problem of tracking a piecewise constant reference, which can be easily re-cast into a stabilisation problem. The reference signal was a square wave of period 10 s, $\pi/4$ amplitude and duty cycle of 50%. Figure 5.5.1 reports the logarithm of $\mathsf{E}\left\{\|x(k)\|^2\right\}$ on 200 runs with different UD sample paths and different initial conditions for both the *drop-out* (dashed) and the *static* (solid) reservation policy.

The evident convergence of the mean squared tracking error is a direct consequence of the 2-GAS. Interestingly the *drop-out* policy led to instability. As a final remark, even if large fluctuations of the state variables cannot be ruled out under 2-GAS, in these experiments the transient behaviour was always satisfactory, i.e., the deviation of the system behaviour with respect to the ideal execution (where the control data is always delivered on the deadline) was consistently below 0.1 rad for all simulations run.

### *Best-Effort Activities*

In this set of experiments 100 unstable plants, whose order ranges between two and six, have been generated. The task period has been fixed to $T = 10$ ms, which is four times the reservation period $R = 2.5$ ms. For the computation times, the UD and BD are defined in the range $[2.5, 10]$ ms (i.e. BCET of 2.5 ms and WCET of $T = 10$ ms), while the ED is truncated (and re–normalised) in the same range. The mean value is

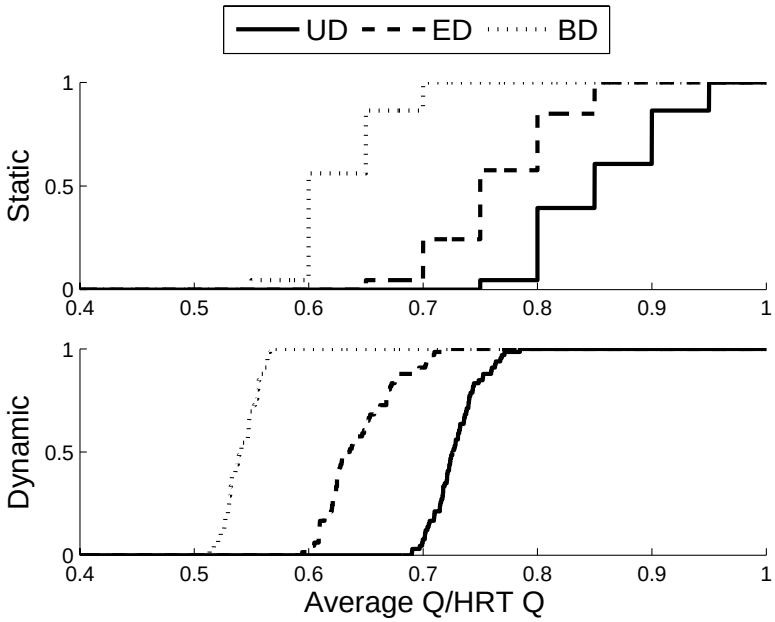


FIGURE 5.5.2: CDFs representing the experimental probability of using a fraction of the budget allocated to *HRT* for both *static* and *dynamic* policies.

$\mu$ = 4.86 ms (standard deviation $\sigma$ = 1.25 ms) for the BD and $\mu$ = 5.62 ms ($\sigma$ = 2.1 ms) for the ED. We draw a comparison between a hard real–time policy (*HRT*) and the two proposed policies solution of the ORP-2GAS problem: the *static* policy in Section 5.4.3.1 and the *dynamic* policy in Section 5.4.3.2. Note that with the *HRT* policy the 100% of the CPU utilisation would be required. The objective of this set of experiments is to quantify the amount of bandwidth saving. To this aim, for each experiment run we measure the ratio between the average allocated budget (i.e. total budget over number of jobs) and the budget required by the *HRT* (which is constant). In Figure 5.5.2, we report the experimental probability of this ratio in form of Cumulative Density Functions (CDFs). In plain words, for a given value $x$ of the ratio, the plot shows the experimental probability of using less than $x$% of the *HRT* budget.

This set of experiments underscores the impact of the distributions of computation times on the QoS improvement, the UD being the most challenging and the BD the most favourable. The *dynamic* policy is the most effective in exploiting the knowledge of the distribution, e.g. for the BD the bandwidth savings are closely concentrated around 50%.

## 5.6 AN INTERESTING EXTENSION

Comparing the approaches presented in this chapter and in Chapter 4, a question may naturally arise: can we merge them? In other words, can we adopt the more control-oriented performance measure of Problem (4.4.2), still retaining the more refined and flexible model of computation of the ORP-2GAS problem? A positive answer has been proposed in [88] and is based on the *continuous stream* model of computation [90].

In both Chapter 4 and Chapter 5 the tasks are triggered in a strictly periodic fashion ($r_{j+1} = r_j + T$). Despite such periodic triggering, a task following the model of computation in Section 5.2.2, cannot always immediately execute. Indeed, a job $J_j$ can be triggered at $r_j$ even if the job $J_{j-1}$ has not finished yet, because the latter is executing beyond its deadline $d_{j-1}$ (recall that $r_j = d_{j-1}$). This means that a job can be affected by a backlog due to late terminations of previous jobs. Such backlog is kept under control only by the application of the cancelling Rule 3.

The continuous stream model is an "extreme" evolution of the soft real–time model described in the present chapter in which the period $T$ is relegated to the role of just suggesting a nominal or ideal pace. Indeed, when a job $J_j$ is activated at time $r_j$, it is assigned a relative deadline $D$ to complete, which is a multiple of the reservation period $R$, usually fixed to $D = T = NR$. But even when $D = T$, a job triggering is not necessarily periodic, that is $r_j$ is not necessarily a multiple of $T$. More precisely, the continuous stream model is defined by the following new set of rules:

*Rule 1:* If a job $J_j$ finishes *before* its deadline (i.e. $f_j < d_j = r_j + D$), the release of the output $u_j$ is deferred to the time $r_j + D$ (i.e. $v_j = r_j + D$ is forced). A new job is immediately activated (i.e. $r_{j+1} = v_j = r_j + D$).

*Rule 2:* If a job $J_j$ finishes *after* its deadline (i.e. $f_j > d_j = r_j + D$), the release of the output $u_j$ takes place immediately at the end of the present reservation period (i.e. $v_j = r_j + \left\lceil \frac{f_j - r_j}{R} \right\rceil R$ is enforced). A new job is immediately activated (i.e. $r_{j+1} = v_j = r_j + \left\lceil \frac{f_j - r_j}{R} \right\rceil R$).

*Rule 3*: If a job $J_j$ experiences a delay greater than a threshold, it is cancelled and a new job $J_{j+1}$ is activated.

*Rule 4:* The output $y_j$ is sampled at the same time the control $u_{j-1}$ is released (at instant $s_j = v_{j-1}$).

When $D$ is fixed equal to the nominal period $T$, these rules look similar to those presented in this chapter. However, there is a relevant difference. Except for the case of a delay larger than the threshold (Rule 3), the activation of a new job is now aligned to the release time $v_j$ (Rules 1 and 2). In other words, the idea of time-triggered activation is relaxed as a new job is activated immediately after the completion or cancellation of the previous one (hence the name ''continuous stream''). The absence of carry-on execution between adjacent jobs prevents an accumulation of delays (backlog). This drastically simplifies the model expressing the evolution of the delays, which becomes an i.i.d. process (see [90]) and is no longer a FSH-MC as in Section 5.4. Also the switching process simplifies to an i.i.d. process as in Chapter 4 (compare again to the more complicated model in Section 5.4).

With the adoption of the continuous stream, the dynamics of the closed loop system can evolve from the all-or-nothing model of Chapter 4 to a model akin to that presented in this chapter and taking into account all the possible delays a job can experience. The probability of meeting the deadline (cf. equation (4.4.1)) is then replaced by a vector of probabilities of a job to execute exactly in a given number of reservation periods.

On the other hand, the addition of a stochastic input in the form of a white Gaussian noise, makes the optimisation problem amenable of a formulation where the cost index is also the QoC (minimum of the maximal variance of the closed loop systems) as in Chapter 4. The solution set is again a polytope that can be computed before the optimisation is performed. As a consequence, a characterisation of the optimal solution set, very close to that presented in Section 4.5, can be provided (see [88] for details).

## 5.7 CONCLUSIONS

This Chapter considered the problem of scheduling soft real-time tasks, possibly accompanied by best effort tasks, on a shared computation platform. The tasks are allowed to terminate after their deadline with a bounded delay. If a fixed maximum delay is reached, the task is dropped and the control value is not updated (or even zeroed). The main contribution are as follows:

*A refined QoS model.* We presented a stochastic flexible QoS model still based on the CBS scheduler as that of Chapter 4, but, the adoption of a Resource Reservation policy, allowed us to make available a finer granularity in the choice of the design parameters.

*Adaptive Reservation policies.* The use of a more refined QoS model enabled the definition of dynamic policies that ensure a more flexible allocation of the resources and that better adapt to fluctuations.

*An optimal allocation with guaranteed QoC.* We formulated an optimal allocation problem with two cost functions corresponding to two concrete application cases. We embedded in the optimisation problem the QoC guarantees in the form of stochastic stability (2-GAS).

*A continuous stream extension.* We briefly illustrated an extension to the present model which gets rid almost completely of the very idea of periodic tasks. The continuous stream model simplifies the overall problem formulation still enabling an efficient solution to the optimisation problem and the use of control-oriented QoC metrics.

# 6

# SWITCHED SYSTEMS WITH $\omega$-REGULAR CONSTRAINTS

## 6.1 INTRODUCTION

When dealing with a multitude of agents competing for the access to resources provided by a shared platform, it is sometimes necessary to specify special kinds of constraints e.g. fairness, precedence, or any constraint expressed in terms of linear temporal logic specifications [14]. Classical descriptions involving maximum delays, maximum latency between two communications, dwell time, etc. are not well adapted. In these instances, a suitable way to build a QoS model is by means of $\omega$-regular languages. More specifically, the agents, the platform and their interaction can be described in terms of a switched system where the switching signals belong to an $\omega$-regular language. An interesting aspect of $\omega$-regular languages is that they can always be characterised using Büchi automata [14]. As an example, a particular class of switching signals with $\omega$-regular constraints studied in literature is that of shuffled switching signals ([103, 206, 95]), namely signals where all the modes of the switched system are activated infinitely often. We remark that shuffled switching signals cannot be generated by the labeled graphs considered in the works [130, 63, 11, 169, 165].

In view of their applicative relevance, it is important to analyse the stability properties associated with switched systems driven by $\omega$-regular signals. This is the stepping stone to any procedure aiming to strike a trade-off between QoS and QoC. Of particular concern here is the question: what are the minimal requirements in terms of QoS to ensure the desired QoC in the form of stability or convergence of the switched system?

This chapter is made up of two parts.

**The first part** provides an answer to the previous question for discrete-time switched linear systems via sufficient Lyapunov conditions. Such conditions are amenable of an

LMI formulation when quadratic Lyapunov functions are chosen, and turn out to be necessary when the state matrices of each subsystem are invertible. Our approach is illustrated using a simple example of oscillators synchronisation over a communication network.

We stress that classical results developed for arbitrary switching signals [135, 190, 136] are still applicable, but prove very conservative, while results with dwell-time conditions are not adapted. Our work build on the paper [169], where the authors consider switching signals generated by labeled graphs, that correspond to non-deterministic Büchi automata where all states are accepting. However, not all $\omega$-regular languages can be generated using labeled graphs. Hence, the results here presented subsume those of [169] (and of [95]) and also apply to systems that cannot be handled by any of these methods.

Finally, stability analysis of discrete-time switched linear systems constrained by $\omega$-regular languages have already been considered in the literature in [206], where it is shown that the stability is equivalent to the stability of a lifted system driven by shuffled switching signals. In comparison, our approach works directly on the original state-space, and thus results in more tractable conditions. Our Lyapunov functions also resembles that considered in [126]. However, in that work, the connection to $\omega$-regular languages has not been investigated.

**The second part** builds upon the results of the first part and addresses a relevant problem: designing asymptotic observers for discrete-time switched linear systems.

For this class of systems, there exists a diversity of observability notions depending on whether the switching signal is known [94, 12, 17] or need to be estimated [13, 67]. Here, we assume that the switching signal is known. In that context, there have been several works for characterising observable or reconstructible switching sequences [94, 191], i.e. finite switching sequences that make it possible to estimate the initial or the current state of the system, respectively. Relevant work also includes the characterisation of observability for continuous-time switched systems with jumps at the switching instants [210, 193], which can be seen as a generalisation of discrete-time switched systems. Finally, when the switching signal is constrained by some automaton, necessary and sufficient conditions for observability have been given in [17, 115].

The design of asymptotic observers itself has also been considered in several papers. Observers that are convergent for arbitrary switching sequences have been designed in [5, 62] using mode dependent observer gains and quadratic or poly-quadratic Lyapunov functions. Note that this design requires that the dynamics in each mode is observable, since constant switching signals are allowed. The case where individual dynamics are unobservable but some observable (or reconstructible) switching sequences exist is also of interest. To be able to estimate the state of the system, it is necessary that the switching signal contains at least one reconstructible sequence. Actually, to be able to design observers that are robust to unmodelled disturbances or to measurement noises, it is necessary to consider switching signals containing an infinite number of reconstructible sequences. This is the setting considered in this second part.

Here we exploit the $\omega$-regular language formalism to characterise the set of switch-

ing signals containing an infinite number of reconstructible sequences. More precisely, we provide a constructive procedure to build a Deterministic Büchi Automaton (DBA) generating the required $\omega$-regular language. We then consider the problem of designing an asymptotic observer. We propose a switched observer with an internal discrete state whose dynamics is given by the transition map of the DBA. By adapting the results presented in the first part of the chapter, we then establish sufficient conditions to design the observer gains such that the resulting observer is convergent for a class of constrained switching signals. The constraint imposed on the switching signals is the following: the occurrence rate of reconstructible sequences must be higher than a certain tuneable parameter. Such parameter represents the adaptation term offered by the flexible platform to modulate the QoS.

In the case where all state matrices of the switched system are invertible, we present an explicit construction of suitable gains, showing that the proposed observer structure is universal in the sense that one can always find an observer of the proposed form. We also show an alternative design based on a set of LMIs that always admit a feasible solution under the same invertibility assumption. Finally, we extend our approach to take into account additional constraints on the switching signals. The effectiveness of the proposed methodology is illustrated through a practical case study in which we design an observer for a multicellular converter [201, 200].

The most related works in the observer literature are the following. In [131], a switched observer is presented where the observer gain at some instant $t$ depends on the sequence of modes at time $t-L, \ldots, t$. Similar to our construction, this can be seen as a switched observer with an internal discrete state, though the discrete dynamics is different from that presented here. In [131], the design of the observer gains is done by solving LMIs. However, contrarily to our approach, there is no clear characterisation of the cases when the proposed design can be successful. Another approach can be to consider a switched linear system as a time-varying linear system and to apply associated observer design techniques, such as Kalman filters [217]. Alternatively, the design proposed in [193] for observers of continuous-time switched linear systems with jumps at the switching instants can be adapted to discrete-time systems. The observers designed in [217] and [193] can be shown to be convergent for switching signals containing an infinite number of reconstructible sequences. However, their gains have to be computed online. In comparison, the gains of our observer are computed offline, resulting in reduced requirements for its implementation.

**Notations:** We denote by $\varnothing$ the empty set, by $\mathbb{R}$, $\mathbb{R}_0^+$ and $\mathbb{N}$ the sets of real numbers, nonnegative real numbers and nonnegative integers, respectively. $\|.\|$ denotes an arbitrary norm on $\mathbb{R}^n$ and the associated induced matrix norm defined for $M \in \mathbb{R}^{n\times n}$ by $\|M\| = \sup_{x\neq 0} \frac{\|Mx\|}{\|x\|}$. $I_n \in \mathbb{R}^{n\times n}$ denotes the $n$ dimensional identity matrix. We use $|.|$ to denote the cardinality of a finite set.

## 6.2 BÜCHI AUTOMATA AND STABILITY DEFINITIONS

In this section we introduce the Büchi automaton, a key tool for the description of constraints in switching signals, along with the class of systems under study. We also define several associated notions related to stability.

### 6.2.1 Büchi automata

Given a finite set $\Sigma$ called *alphabet*, a *word* is either a finite or an infinite sequence of elements of $\Sigma$, i.e. $w = \sigma_1\sigma_2\ldots$, or the *empty word* $w = \epsilon$. $\Sigma^+$ denotes the set of all finite words over the alphabet $\Sigma$ exempted from the empty word; $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$ and $\Sigma^\omega$ denotes the set of all infinite words over the alphabet $\Sigma$. The *length* of a word $w \in \Sigma^*$, denoted by $|w|$, is $n$ if $w = \sigma_1\cdots\sigma_n$, or 0 if $w = \epsilon$. For $w \in \Sigma^*$, we say that $w'$ is a *prefix* of $w$ if it belongs to the set $\mathsf{P}(w) = \{w' \in \Sigma^* \mid \exists w'' \in \Sigma^*, w = w'w''\}$. Similarly, we say that $w'$ is a *suffix* of $w$ if it belongs to the set $\mathsf{S}(w) = \{w' \in \Sigma^* \mid \exists w'' \in \Sigma^*, w = w''w'\}$. We stress that, in the previous definitions, $w'$ and $w''$ can be the empty word. Let $S_1, S_2 \subseteq \Sigma^*$, then the set $S_1S_2$ is the set of words consisting of the concatenation of words of $S_1$ and $S_2$, i.e. $S_1S_2 = \{w = w_1w_2 | \, w_1 \in S_1, w_2 \in S_2\}$. Let $S \subseteq \Sigma^+$, then $S^\omega$ is the set of words consisting of the concatenation of an infinite sequence of words of $S$, i.e. $S^\omega = \{w_1w_2\cdots | \, w_i \in S, i = 1,\ldots\}$.

A *non-deterministic Büchi automaton* (NBA, see e.g. [14]) is a tuple $\mathcal{B} = (Q, \Sigma, \delta, Q_0, F)$ where $Q$ is a finite set of *states*, $\Sigma$ is the *alphabet*, $\delta : Q \times \Sigma \to 2^Q$ is a *transition function*, $Q_0$ is the set of *initial states* and $F$ is the set of *accepting states*. A *run* associated with a finite or an infinite word $\sigma_1\sigma_2\cdots \in \Sigma^* \cup \Sigma^\omega$ is a sequence of states $q_0q_1q_2\ldots$ in $Q$ such that $q_0 \in Q_0$ and $q_{i+1} \in \delta(q_i, \sigma_i)$ for all $i = 0, 1, \ldots$. A run $q_0q_1q_2\ldots$ associated with an infinite word $\sigma_1\sigma_2\cdots \in \Sigma^\omega$ is said to be *accepting* if $q_i \in F$ for infinitely many indices $i \in \mathbb{N}$. The language of $\mathcal{B}$, denoted by $Lang(\mathcal{B})$, is the set of all words in $\Sigma^\omega$ which have an associated accepting run in $\mathcal{B}$. We note that an NBA is called *deterministic Büchi automaton* (DBA) if $|\delta(q, i)| \leq 1$ for all $q \in Q$, $i \in \Sigma$ and $|Q_0| = 1$. It is well known that $Lang(\mathcal{B})$ is an $\omega$-regular language (see again [14]). Given $q', q'' \in Q$ and $\sigma_1\cdots\sigma_n \in \Sigma^*$, we write $q' \xrightarrow{\sigma_1\cdots\sigma_n} q''$ if $q''$ is obtained as a concatenation of transitions from $q'$, namely $q'' = \delta(\cdots\delta(\delta(q', \sigma_1), \sigma_2), \ldots, \sigma_n)$.

### 6.2.2 Stability of switched systems with $\omega$-regular switching sequences

Let us consider a discrete-time switched linear system in which the switching sequences are infinite words accepted by a given NBA. Specifically, given a Büchi automaton $\mathcal{B} = (Q, \mathcal{I}, \delta, Q_0, F)$ where the alphabet is the set $\mathcal{I} = \{1, \ldots, m\}$, given a finite set of matrices $\mathcal{A} = \{A_1, \ldots, A_m\}$ with $A_i \in \mathbb{R}^{n\times n}$, $i \in \mathcal{I}$, the discrete-time switched linear system with $\omega$-regular switching sequences $(\mathcal{A}, \mathcal{B})$ is described by the equation

$$x(t+1) = A_{\theta(t)}x(t), \tag{6.2.1}$$

where $t \in \mathbb{N}$, $x(t) \in \mathbb{R}^n$ is the state and $\theta : \mathbb{N} \to \mathcal{I}$ is the switching signal with $\theta \in Lang(\mathcal{B})$ where, by abuse of notation, we say that $\theta \in Lang(\mathcal{B})$ if the infinite word

$\theta(0)\theta(1)\cdots \in Lang(\mathcal{B})$. For $\theta \in Lang(\mathcal{B})$, we denote by $q_0, q_1, \ldots$ the associated accepting run. We note that, given an initial condition $x_0 \in \mathbb{R}^n$, and a switching signal $\theta \in Lang(\mathcal{B})$, the trajectory with $x(0) = x_0$ is unique, denoted by $\mathbf{x}(., x_0, \theta)$ and given by

$$\forall t \geq 1 : \mathbf{x}(t, x_0, \theta) = \prod_{i=0}^{t-1} A_{\theta(i)} x_0, \tag{6.2.2}$$

where $\prod_{i=0}^{t-1} A_{\theta(i)} = A_{\theta(t-1)} \times \cdots \times A_{\theta(0)}$.

Now we introduce the following running Assumption:

**Assumption 6.1.** *All the states of the Büchi automaton $\mathcal{B}$ are reachable from at least one initial state and for any finite run $q_0 q_1 q_2 \ldots q_k$ there exists an infinite sequence of states $q_{k+1} q_{k+2} \ldots$ such that $q_0 q_1 q_2 \ldots$ is an accepting run.*

Note that there is no loss of generality to suppose that Assumption 6.1 holds true since it can be shown easily that for any NBA (resp. DBA), there exists an NBA (resp. DBA) with the same language and satisfying Assumption 6.1. Hence, in the rest of the paper, Assumption 6.1 is always supposed to be satisfied.

We start by defining some stability notions.

**Definition 6.1.** The system $(\mathcal{A}, \mathcal{B})$ is *globally attractive (GA)* if for all switching signals $\theta \in Lang(\mathcal{B})$ and for all initial conditions $x_0 \in \mathbb{R}^n$, we have

$$\lim_{t \to \infty} \|\mathbf{x}(t, x_0, \theta)\| = 0.$$

**Definition 6.2.** The system $(\mathcal{A}, \mathcal{B})$ is *globally uniformly stable (GUS)* if there exists a scalar $\alpha \geq 1$ such that for all switching signals $\theta \in Lang(\mathcal{B})$ and for all initial conditions $x_0 \in \mathbb{R}^n$, we have

$$\|\mathbf{x}(t, x_0, \theta)\| \leq \alpha \|x_0\|, \forall t \in \mathbb{N}.$$

Now, we define some quantities specific to a DBA $\mathcal{B}$. For $\theta \in Lang(\mathcal{B})$, we denote by $q_0 q_1 \ldots$ its unique accepting run in $\mathcal{B}$ and we define the sequence of *return instants* $(\tau_k^{\theta,\mathcal{B}})_{k \in \mathbb{N}}$ by $\tau_0^{\theta,\mathcal{B}} = 0$ and, for $k \in \mathbb{N}$,

$$\tau_{k+1}^{\theta,\mathcal{B}} = \min \left\{ t > \tau_k^{\theta,\mathcal{B}} \,|\, q_t \in F \right\},$$

and the *return index* $\kappa^{\theta,\mathcal{B}} : \mathbb{N} \to \mathbb{N}$ by

$$\kappa^{\theta,\mathcal{B}}(t) = \max \left\{ k \in \mathbb{N} \,|\, \tau_k^{\theta,\mathcal{B}} \leq t \right\}.$$

Intuitively, $\tau_k^{\theta,\mathcal{B}}$ is the first instant where the run associated with $\theta$ has visited the set $F$ of accepting states $k$ times, and $\kappa^{\theta,\mathcal{B}}(t)$ is the number of times the run associated with $\theta$ has visited $F$ up to time $t$. Since $\theta \in Lang(\mathcal{B})$, the set $F$ will be visited an infinite

number of times, so $\tau_k^{\theta,\mathcal{B}}$ is well defined for every $k \in \mathbb{N}$ and $\lim_{t\to\infty} \kappa^{\theta,\mathcal{B}}(t) = \infty$. We also define the *accepting rate* by

$$\gamma^{\theta,\mathcal{B}} = \liminf_{t\to\infty} \frac{\kappa^{\theta,\mathcal{B}}(t)}{t}.$$

We note that an infinite word can have multiple runs in an NBA, therefore $\kappa^{\theta,\mathcal{B}}(t)$, $\tau_k^{\theta,\mathcal{B}}$ and $\gamma^{\theta,\mathcal{B}}$ cannot be defined as above for an NBA.

Let us define the following stability notion for a DBA.

**Definition 6.3.** The system $(\mathcal{A},\mathcal{B})$, where $\mathcal{B}$ is a DBA, is *globally uniformly exponentially stable (GUES)* if there exist a scalar $C \geq 1$ and a scalar $0 < \lambda < 1$ such that for all switching signals $\theta \in Lang(\mathcal{B})$ and for all initial conditions $x_0 \in \mathbb{R}^n$, we have

$$\|\mathbf{x}(t, x_0, \theta)\| \leq C\lambda^{\kappa^{\theta,\mathcal{B}}(t)}\|x_0\|, \forall t \in \mathbb{N}.$$

It is clear that if the system $(\mathcal{A},\mathcal{B})$ is *GUES* then it is *GA* and *GUS*, we note that this type of stability cannot be defined for an NBA since $\kappa^{\theta,\mathcal{B}}(t)$ is not defined for these automata.

We have seen in this section several notions of stability of switched linear systems with $\omega$-regular switching sequences, some of them concern the general case of an NBA, and the rest concerns the DBA only. In the next sections, we will develop sufficient conditions for stability using a Lyapunov approach and we will give a converse result for a specific class of systems.

## 6.3 SUFFICIENT CONDITIONS FOR STABILITY

In this section, for a system $(\mathcal{A},\mathcal{B})$, we establish sufficient conditions for the notions of stability defined in the previous section based on the following type of Lyapunov functions:

**Definition 6.4.** For the system $(\mathcal{A},\mathcal{B})$, the function $V : Q \times \mathbb{R}^n \to \mathbb{R}_0^+$, is called *Lyapunov function* if there exist scalars $\alpha_1, \alpha_2 > 0$ and $0 < \lambda < 1$ such that for all $x \in \mathbb{R}^n$, the following hold:

$$\alpha_1\|x\| \leq V(q,x) \leq \alpha_2\|x\|, \qquad q \in Q \tag{6.3.1}$$

$$V(q', A_i x) \leq V(q,x), \qquad q \in Q, i \in \mathcal{I}, q' \in \delta(q,i) \setminus F \tag{6.3.2}$$

$$V(q', A_i x) \leq \lambda V(q,x), \qquad q \in Q, i \in \mathcal{I}, q' \in \delta(q,i) \cap F \tag{6.3.3}$$

**Theorem 6.1.** *If there exists a Lyapunov function for the system $(\mathcal{A},\mathcal{B})$ then $(\mathcal{A},\mathcal{B})$ is* GA *and* GUS. *If in addition $\mathcal{B}$ is a DBA then $(\mathcal{A},\mathcal{B})$ is* GUES.

If we restrain to Lyapunov functions of the form $V(q,x) = \sqrt{x^T P_q x}$ where, for every $q \in Q$, $P_q$ is a positive definite matrix, then the conditions in Definition 6.4

are equivalent to a set of linear matrix inequalities (LMI). In that case, stability of the switched system $(\mathcal{A}, \mathcal{B})$ can be verified by solving a convex optimisation problem.

It is worth noting that inequality (6.3.2) requires a non-increasing function $V$, which excludes the possibility of having matrices $A_i$ whose spectral radius is larger than 1. This is necessary to prove *GA* and *GUS* in Theorem 6.1, as there is no bound on the length of a sequence between two consecutive visits to the set $F$. Moreover, our definition of *GUES* also imposes the same constraint on the Lyapunov function. However, in the case of DBA, a result allowing for unstable matrices $A_i$ can be proved. In particular, *GA* can still be guaranteed if the set $F$ is visited frequently enough.

**Theorem 6.2.** *If there exist* $V : Q \times \mathbb{R}^n \to \mathbb{R}_0^+$, $\alpha_1, \alpha_2, \rho > 0$ *and* $0 < \lambda < 1$ *such that for all* $x \in \mathbb{R}^n$, *the following hold:*

$$\alpha_1 \|x\| \leq V(q,x) \leq \alpha_2 \|x\|, \qquad q \in Q \tag{6.3.4}$$

$$V(q', A_i x) \leq \rho V(q,x), \qquad q \in Q, i \in \mathcal{I}, \delta(q,i) = q' \notin F \tag{6.3.5}$$

$$V(q', A_i x) \leq \rho\lambda V(q,x), \qquad q \in Q, i \in \mathcal{I}, \delta(q,i) = q' \in F \tag{6.3.6}$$

*then there exists* $C \geq 1$ *such that for all switching signals* $\theta \in Lang(\mathcal{B})$ *and for all initial conditions* $x_0 \in \mathbb{R}^n$, *we have*

$$\|\mathbf{x}(t, x_0, \theta)\| \leq C\rho^t \lambda^{\kappa^{\theta,\mathcal{B}}(t)} \|x_0\|, \forall t \in \mathbb{N}.$$

*Moreover, whenever the accepting rate satisfies* $\gamma^{\theta,\mathcal{B}} > \frac{\ln(\rho)}{-\ln(\lambda)}$, *we have*

$$\lim_{t\to\infty} \|\mathbf{x}(t, x_0, \theta)\| = 0.$$

## 6.4 NECESSARY CONDITIONS FOR STABILITY

In this section we show that, if all matrices in $\mathcal{A}$ are invertible, then the existence of a Lyapunov function for $(\mathcal{A}, \mathcal{B})$ is not only sufficient but also necessary for the attractivity and uniform stability of the system. We introduce some quantities related to the NBA $\mathcal{B}$. We define $\mathcal{L}_{qq_f}$ as the set of all words in $\Sigma^+$ corresponding to a run starting from $q \in Q$ and reaching $q_f \in F$ without visiting any accepting state between $q$ and $q_f$. Formally

$$\mathcal{L}_{qq_f} = \left\{ \sigma_1 \dots \sigma_k \in \Sigma^+ \;\middle|\; \begin{array}{c} q_{i+1} \in \delta(q_i, \sigma_i) \setminus F, 1 \leq i < k \\ \text{where } q_1 = q \text{ and } q_f \in \delta(q_k, \sigma_k) \end{array} \right\}.$$

Now we define $\mathcal{L}^{\times}_{qq_f}$ as the set of all products of matrices in $\mathcal{A}$ associated with words in $\mathcal{L}_{qq_f}$:

$$\mathcal{L}^{\times}_{qq_f} = \left\{ A_{\sigma_k} \times \dots \times A_{\sigma_1} \;\middle|\; \sigma_1 \dots \sigma_k \in \mathcal{L}_{qq_f} \right\}.$$

Given a word $l \in \mathcal{L}_{qq_f}$ the corresponding matrix in $\mathcal{L}^{\times}_{qq_f}$ is denoted $M_l$.

In order to establish necessary conditions, we apply a result obtained in [169] based on labeled graphs. A labeled graph is a tuple $\mathcal{G} = (V, L, E)$ where $V$ is a set of nodes, $L$ is

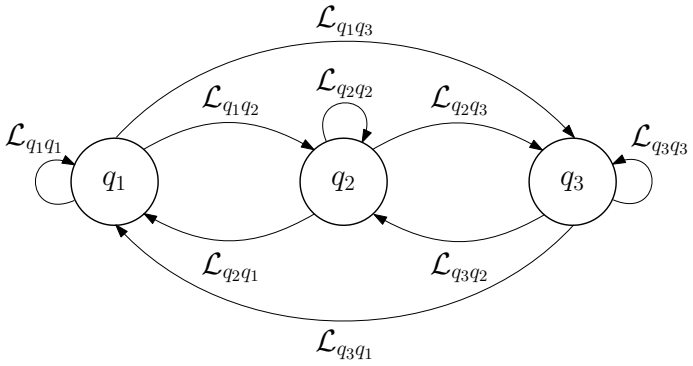


FIGURE 6.4.1: The labeled graph $\mathcal{G}$ corresponding to an NBA $\mathcal{B}$ with 3 accepting states $q_1, q_2$ and $q_3$.

a set of labels, $E \subset V \times L \times V$ is a set of edges or transitions. For an edge $e = (v, l, v') \in E$, the node $v \in V$ is the origin, $l \in L$ is the label, $v' \in V$ is the end. A path in the labeled graph $\mathcal{G}$ is a set of consecutive edges $(v_0, l_1, v_1), (v_1, l_2, v_2), \ldots$ where $v_i \in V$ for all $i = 0, 1, \ldots$, its label is the word $w = l_1 l_2 \ldots$.

In order to analyse the dynamics of the system $(\mathcal{A}, \mathcal{B})$, we use the concept of constrained system [169]. A constrained system is formed from a labeled graph $\mathcal{G}$, and a set of matrices $\mathcal{M}$ where each matrix of $\mathcal{M}$ corresponds to an element of the labels of $\mathcal{G}$. In our case, with a switched system $(\mathcal{A}, \mathcal{B})$ one can associate a constrained system $(\mathcal{M}_{\mathcal{B}}^F, \mathcal{G})$, where $\mathcal{M}_{\mathcal{B}}^F = \bigcup_{q,p \in F} \mathcal{L}_{qp}^{\times}$ and the nodes of $\mathcal{G}$ are $V = F$, the labels $L = \bigcup_{q,p \in F} \mathcal{L}_{qp}$, and the set of edges is $E = \left\{ (q, l, p) \,\middle|\, q, p \in F, l \in \mathcal{L}_{qp} \right\}$.

*Remark* 6.1. In [169], the set of labels $L$ is assumed to be finite. In our case, this set can be infinite.

Figure 6.4.1 shows the labeled graph corresponding to an NBA $\mathcal{B}$ with three accepting states $q_1, q_2$ and $q_3$. In the figure, by an abuse of notation, for $q_i, q_j \in F$, the edge $(q_i, \mathcal{L}_{q_i q_j}, q_j)$, denotes the set of edges $\left\{ (q_i, l, q_j) \,\middle|\, l \in \mathcal{L}_{q_i q_j} \right\}$.

Note that, unlike [169], in our case the set of edges of the labeled graph is usually infinite. We consider the following stability notion.

**Definition 6.5.** We say that the constrained system $(\mathcal{M}_{\mathcal{B}}^F, \mathcal{G})$ is attractive if for all paths $(q_{f_0}, l_1, q_{f_1}), (q_{f_1}, l_2, q_{f_2}), \ldots$ we have

$$\lim_{k \to \infty} \| M_{l_k} \cdots M_{l_1} \| = 0.$$

In the following we will make use of the following assumption.

**Assumption 6.2.** *All the matrices in $\mathcal{A}$ are invertible.*

We will analyse the attractivity of the constrained system $(\mathcal{M}_{\mathcal{B}}^F, \mathcal{G})$ by making use of the concept of multinorm, defined below.

**Definition 6.6.** (Definition 1 in [169]) A multinorm of the constrained system $(\mathcal{M}_{\mathcal{B}}^{F}, \mathcal{G})$, denoted by $\mathcal{H}$, is a set of $|F|$ norms in $\mathbb{R}^n$, that is $\mathcal{H} = \{\|.\|_q, q \in F\}$. The value of the multinorm $\nu^*(\mathcal{H})$ is defined as

$$\nu^*(\mathcal{H}) = \inf \left\{ \nu > 0 \;\middle|\; \begin{array}{c} \|Mx\|_p \leq \nu \|x\|_q, \ \forall x \in \mathbb{R}^n, \\ \forall q, p \in F \text{ s.t. } \mathcal{L}_{qp} \neq \varnothing, \ \forall M \in \mathcal{L}_{qp}^{\times} \end{array} \right\}.$$

This definition coincides with Definition 1 in [169] except for the fact that here the matrix $M$ takes values on a possibly infinite set.

**Theorem 6.3.** *Suppose that Assumption 6.2 holds true and that* $(\mathcal{A}, \mathcal{B})$ *is* GUS. *Then the constrained system* $(\mathcal{M}_{\mathcal{B}}^{F}, \mathcal{G})$ *is attractive iff it admits a multinorm* $\mathcal{H}$ *with value* $\nu^*(\mathcal{H}) < 1$.

Now we relate the attractivity of $(\mathcal{A}, \mathcal{B})$ with that of $(\mathcal{M}_{\mathcal{B}}^{F}, \mathcal{G})$ using the following lemma.

**Lemma 6.1.** *Under Assumption 6.2, if the switched system* $(\mathcal{A}, \mathcal{B})$ *is* GA *then the constrained system* $(\mathcal{M}_{\mathcal{B}}^{F}, \mathcal{G})$ *is attractive.*

We next provide a converse result to Theorem 6.1.

**Theorem 6.4.** *Under Assumption 6.2, if the switched system* $(\mathcal{A}, \mathcal{B})$ *is* GUS *and* GA, *then it admits a Lyapunov function.*

**Corollary 6.1.** *Under Assumption 6.2, let* $\mathcal{B}$ *be deterministic. Then, the* $(\mathcal{A}, \mathcal{B})$ *is* GUES *if and only if it is* GA *and* GUS.

## 6.5 NUMERICAL EXAMPLE

We consider a multi-agent system consisting of 3 discrete-time oscillators whose dynamics is given by:

$$z_i(t+1) = Rz_i(t) + u_i(t), \ i = 1, 2, 3 \tag{6.5.1}$$

where $z_i(t) \in \mathbb{R}^2, u_i(t) \in \mathbb{R}^2$ and $R = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix}$ with $\phi = \frac{\pi}{6}$. The input $u_i(t)$ is used for synchronisation purpose and is based on the available information at time $t$. There exist 3 communication channels between agent 1 and agent 2 (channel 1), 2 and 3 (channel 2) and 1 and 3 (channel 3). At each instant, only one of these channels is active and the active channel is selected by a switching signal $\theta : \mathbb{N} \to \mathcal{I} = \{1, 2, 3\}$.

Then, the input value is given as follows:

$$u_1(t) = \begin{cases} k(z_2(t) - z_1(t)), & \text{if } \theta(t) = 1 \\ 0, & \text{if } \theta(t) = 2 \\ k(z_3(t) - z_1(t)), & \text{if } \theta(t) = 3 \end{cases}$$

$$u_2(t) = \begin{cases} k(z_1(t) - z_2(t)), & \text{if } \theta(t) = 1 \\ k(z_3(t) - z_2(t)), & \text{if } \theta(t) = 2 \\ 0, & \text{if } \theta(t) = 3 \end{cases}$$

$$u_3(t) = \begin{cases} 0, & \text{if } \theta(t) = 1 \\ k(z_2(t) - z_3(t)), & \text{if } \theta(t) = 2 \\ k(z_1(t) - z_3(t)), & \text{if } \theta(t) = 3 \end{cases}$$

where $k = 0.05$ is a control gain. Denoting the vector of synchronisation errors as $x(t) = (x_1(t)^T, x_2(t)^T)^T$ with $x_i(t) = z_{i+1}(t) - z_i(t)$, the error dynamics is described by a 4-dimensional switched linear system of the form:

$$x(t+1) = A_{\theta(t)} x(t)$$

where the 3 matrices describing the 3 modes of communication are given by:

$$A_1 = \begin{bmatrix} R - 2kI_2 & 0 \\ kI_2 & R \end{bmatrix}, A_2 = \begin{bmatrix} R & kI_2 \\ 0 & R - 2kI_2 \end{bmatrix}, A_3 = \begin{bmatrix} R - kI_2 & -kI_2 \\ -kI_2 & R - kI_2 \end{bmatrix}.$$

As for the communication protocol, we impose a fairness constraint that the switching signal cannot keep activating the same communication channel:

$$\forall t \in \mathbb{N}, \exists t' \geq t,\ \theta(t') \neq \theta(t).$$

This is equivalently described by a DBA $\mathcal{B}$, where the set of states is $Q = \{q_0, q_1, q_2, q_3\}$, the alphabet $\mathcal{I} = \{1, 2, 3\}$, $Q_0 = \{q_0\}$, $F = Q_0$. Figure 6.5.1 shows the corresponding Büchi automaton which describes the switching logic in this system.

We want to show that the agents synchronise if the state $q_0$ in $\mathcal{B}$ is visited infinitely often. This can be done by studying the stability of $(\mathcal{A} = \{A_1, A_2, A_3\}, \mathcal{B})$. We then look for a Lyapunov function of the form $V(q, x) = \sqrt{x^T P_q x}$ where $P_q$ is a positive definite symmetric matrix. The conditions in Theorem 6.1 translate into the following linear matrix inequalities:

$$\begin{aligned} & I_4 \leq P_q, && q \in Q \\ & A_i^T P_{q'} A_i \leq P_q, && q \in Q, i \in \mathcal{I}, q' \in \delta(q, i) \setminus F \\ & A_i^T P_{q'} A_i \leq \lambda^2 P_q && q \in Q, i \in \mathcal{I}, q' \in \delta(q, i) \cap F. \end{aligned}$$

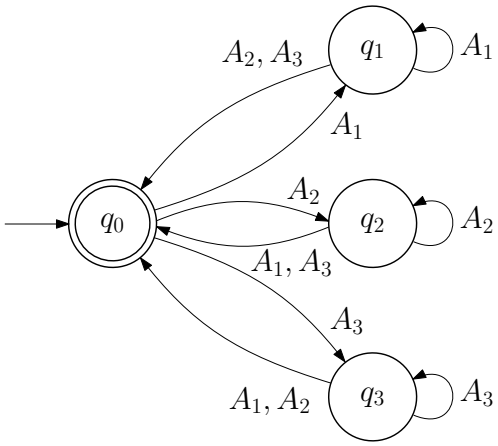


FIGURE 6.5.1: Deterministic Büchi automaton $\mathcal{B}$.

By solving these 16 LMIs, we find for $\lambda = 0.96$ :

$$P_{q_0} = \begin{bmatrix} 1.98I_2 & 0.98I_2 \\ 0.98I_2 & 1.98I_2 \end{bmatrix}, \qquad P_{q_1} = \begin{bmatrix} 2.26I_2 & 0.99I_2 \\ 0.99I_2 & 1.98I_2 \end{bmatrix},$$
$$P_{q_2} = \begin{bmatrix} 1.98I_2 & 0.99I_2 \\ 0.99I_2 & 2.26I_2 \end{bmatrix}, \qquad P_{q_3} = \begin{bmatrix} 2.22I_2 & 1.22I_2 \\ 1.22I_2 & 2.22I_2 \end{bmatrix}.$$

From Theorem 6.1, we get that the switched system $(\mathcal{A}, \mathcal{B})$ is *GUES* which means the oscillators synchronise well after sufficient time.

We now consider the following scenario: for the first 50 time units, the communication channel 1 is constantly active, then at $t = 50$ a switch occurs and for the next 50 time units the channel 2 is active, then at $t = 100$ another switch occurs and the communication channel 3 stays active for the next 50 time units. After $t = 150$, the switching signal randomly activates channel 1 and channel 2 with equal probability so that the accepting state $q_0$ is visited infinitely often. The simulation results are shown in Figure 6.5.2. It is interesting to remark that when the switching signal remains constant the synchronisation error does not go to zero, however after $t = 150$, when $q_0$ is visited more frequently, the synchronisation error starts to converge towards zero. As expected, the Lyapunov function $V(q_t, x(t))$ is non-increasing and, as soon as the state $q_0$ is visited frequently enough, it starts approaching zero.

## 6.6 THE OBSERVABILITY PROBLEM

We consider now the problem of reconstructing the state of system (6.2.1) from the knowledge of the switching signal $\theta$ and of a measured output $y$. More precisely, the goal is to propose a general procedure for designing asymptotic observers for the

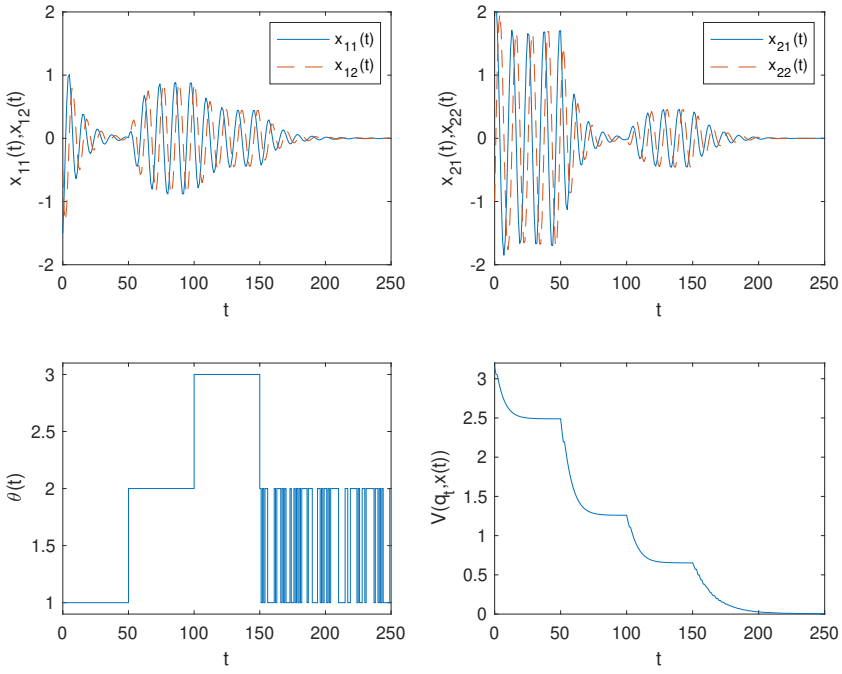


FIGURE 6.5.2: Time evolution of the synchronisation error $x(t)$ (top figures), switching signal $\theta(t)$ (bottom left), and the Lyapunov function $V(q_t, x(t))$ (bottom right).

system:

$$\begin{aligned} x(t+1) &= A_{\theta(t)}x(t), \\ y(t) &= C_{\theta(t)}x(t) \end{aligned} \tag{6.6.1}$$

where $x(t) \in \mathbb{R}^n$ is the state of the system, $\theta : \mathbb{N} \to \mathcal{I}$ is the switching signal, $\mathcal{I} = \{1, \dots, m\}$, $\mathcal{A} = \{A_1, \dots, A_m\} \subseteq \mathbb{R}^{n\times n}$. The output vector is $y(t) \in \mathbb{R}^p$ and the output matrices are in the set $\mathcal{C} = \{C_1, \dots, C_m\} \subseteq \mathbb{R}^{p\times n}$.

## 6.6.1 *Observability of discrete-time switched systems*

By following [190], we recall a definition and a characterisation of observability for the switched system (6.6.1).

**Definition 6.7.** The switched system (6.6.1) is *observable* (resp., *reconstructible*), if there exist a switching signal $\theta$ and $k \in \mathbb{N}$ such that the knowledge of the output sequence $y(0), \dots, y(k)$ is sufficient to determine the initial condition $x_0$ (resp., the state $x(k)$). We refer to the finite sequence of modes $\theta(0), \dots, \theta(k)$ as an observable (resp., reconstructible) sequence.

We recall that an observable system is also reconstructible, while the opposite is true if all the matrices in $\mathcal{A}$ are invertible. We define the observability matrix corresponding

to a sequence of modes $i_1, \dots, i_j \in \mathcal{I}$, as

$$\Omega(i_1, \dots, i_j) = \begin{bmatrix} C_{i_1}^T & A_{i_1}^T C_{i_2}^T & \cdots & A_{i_1}^T \cdots A_{i_{j-1}}^T C_{i_j}^T \end{bmatrix}^T.$$

Necessary and sufficient conditions for observability and reconstructibility are recalled in the following theorem.

**Theorem 6.5.** *[190] A sequence of modes $i_1, \dots, i_j \in \mathcal{I}$ is observable if and only if*

$$\operatorname{rank}\Big(\Omega(i_1, \dots, i_j)\Big) = n. \tag{6.6.2}$$

*It is reconstructible if and only if*

$$\ker\Big(\Omega(i_1, \dots, i_j)\Big) \subseteq \ker(A_{i_1} \cdots A_{i_j}). \tag{6.6.3}$$

As far as the design of asymptotic observers is concerned, it is sufficient to consider the notion of reconstructibility. However, let us remark that the conditions in (6.6.2) and (6.6.3) are equivalent if all matrices in $\mathcal{A}$ are invertible.

## 6.7 A DBA FOR RECONSTRUCTIBLE SWITCHING SIGNALS

From Definition 6.7, it is clear that one can reconstruct the state $x$ of the system whenever the switching signal $\theta$ contains one reconstructible sequence. However, if one wants to design an observer that is robust to unmodelled disturbances or to measurement noises, it becomes necessary to consider switching signals that contain an infinite number of reconstructible sequences. We characterise such switching sequences using the notion of deterministic Büchi automaton introduced in Section 6.2.

### *Construction of the automaton*

Using Theorem 6.5, one can efficiently compute reconstructible sequences up to a specific length for a switched linear system given by $(\mathcal{A}, \mathcal{C})$ and described by equation (6.6.1). The set of reconstructible sequences of length $j \geq 1$ is defined by

$$\mathcal{O}_j = \left\{ \sigma_1 \cdots \sigma_j \in \mathcal{I}^* \;\middle|\; \ker\Big(\Omega(\sigma_1, \dots, \sigma_j)\Big) \subseteq \ker(A_{\sigma_1} \cdots A_{\sigma_j}) \right\}$$

Moreover, the set of reconstructible sequences of length up to $k \geq 1$ is given by

$$\mathcal{O}^{[k]} = \bigcup_{j=1,\dots,k} \mathcal{O}_j.$$

It is easy to see that if $w$ is a reconstructible sequence, then any sequence containing $w$ as a subsequence is also reconstructible. Therefore it is useful to define the following reduced set of reconstructible sequences:

$$\mathcal{O}'^{[k]} = \left\{ w \in \mathcal{O}^{[k]} \;\middle|\; \nexists w_1, w_2 \in \mathcal{I}^*, w' \in \mathcal{O}^{[k]} \setminus \{w\} \quad \text{s.t.} \quad w = w_1 w' w_2 \right\}.$$

**Algorithm 6.1** Construction of DBA $\mathcal{B}_k$

**Inputs:** alphabet $\mathcal{I}$, reconstructible sequences $\mathcal{O}'^{[k]}$
**Output:** DBA $\mathcal{B}_k$ such that $Lang(\mathcal{B}_k) = (\mathcal{I}^*\mathcal{O}'^{[k]})^\omega$

$q_0 := \epsilon, F := \{\epsilon\}, Q := \{\epsilon\}$
**for** $\sigma \in \mathcal{I}$ **do**
  **if** $\sigma \in \mathcal{O}'^{[k]}$ **then**
    $\delta(\epsilon, \sigma) := \epsilon$
  **else**
    $Q := Q \cup \{\sigma\}$
    $\delta(\epsilon, \sigma) := \sigma$
  **end if**
**end for**
**for** $\sigma_1 \cdots \sigma_l \in \mathcal{O}'^{[k]}, l \geq 2$ **do**
  **if** $l = 2$ **then**
    $\delta(\sigma_1, \sigma_2) := \epsilon$
  **else**
    **for** $j \in \{2, \ldots, l-1\}$ **do**
      $Q := Q \cup \{\sigma_1 \cdots \sigma_j\}$
      $\delta(\sigma_1 \cdots \sigma_{j-1}, \sigma_j) := \sigma_1 \cdots \sigma_j$
    **end for**
    $\delta(\sigma_1 \cdots \sigma_{l-1}, \sigma_l) := \epsilon$
  **end if**
**end for**
$\mathcal{W} := \left\{(w, \sigma) \in Q \times \mathcal{I} \,\middle|\, (w, \sigma) \notin \delta^{-1}(Q)\right\}$
**for** $(w, \sigma) \in \mathcal{W}$ **do**
  **if** $\exists w_s \in \mathsf{S}(w)$ s.t. $w_s\sigma \in \mathcal{O}'^{[k]}$ **then**
    $\delta(w, \sigma) := \epsilon$
  **else**
    $\delta(w, \sigma) := \arg \max_{v \in Q \cap \mathsf{S}(w\sigma)} |v|$
  **end if**
**end for**

Intuitively, $\mathcal{O}'^{[k]}$ consists of *minimal reconstructible sequences*, i.e. reconstructible sequences that do not contain reconstructible subsequences. It is easy to see that for any $k_1 \leq k_2$, $\mathcal{O}'^{[k_1]} \subseteq \mathcal{O}'^{[k_2]}$.

Let us consider $k \geq 1$, such that $\mathcal{O}'^{[k]} \neq \varnothing$. Our goal is to generate words containing an infinite number of reconstructible sequences in $\mathcal{O}'^{[k]}$. To this aim, we provide Algorithm 6.1 which outputs a DBA $\mathcal{B}_k = (Q, \mathcal{I}, \delta, q_0, F)$ generating the language $(\mathcal{I}^*\mathcal{O}'^{[k]})^\omega$. The main idea is to build an automaton which visits an accepting state each time a reconstructible sequence occurs. We note that in this automaton the states are words in $\mathcal{I}^*$, i.e. $Q \subseteq \mathcal{I}^*$.

Algorithm 6.1 is comprised of three parts:

- In the first part, from step 2 to step 9, we check if there are reconstructible sequences of length 1. If it is the case we create a self transition to the accepting state, otherwise we add a new state and we define a transition from the accepting state to the newly added state.
- In the second part, from step 10 to step 20, we consider reconstructible sequences $w$ in $\mathcal{O}'^{[k]}$ of length $l \geq 2$. We add to the set $Q$ all the prefixes of $w$, except $w$ itself, and we define the corresponding transitions between consecutive prefixes. The last transition, which happens when the reconstructible sequence $w$ has occurred, leads to the accepting state.
- Finally, in the third part, from step 21 to step 28, we focus on the couples $(w, \sigma) \in Q \times \mathcal{I}$ for which no transition is defined yet. In this case, if there exists a suffix $w_s$ of $w$ such that $w_s\sigma \in \mathcal{O}'^{[k]}$, then we add a transition from $w$ to the accepting state. Otherwise, we look for the longest suffix $v$ of $w\sigma$ such that $v \in Q$ and we define a transition toward this state.

*Remark* 6.2. The construction of the DBA $\mathcal{B}_k$ mostly relies on the set $\mathcal{O}'^{[k]}$. Actually, it is readily seen that the number of states of $\mathcal{B}_k$ is upper bounded by $m + kc_k$, where $c_k$ denotes the number of elements of $\mathcal{O}'^{[k]}$. Note that $c_k$ may grow unbounded with $k$. A trivial bound can be given by $c_k \leq m^k$, however this bound appears to be conservative in practice, as will be seen later on numerical examples. The derivation of tight bounds on the cardinality of $\mathcal{O}'^{[k]}$ remains an open question. Also relevant is the minimal value of $k$ for which $\mathcal{O}'^{[k]} \neq \varnothing$. If all matrices in $\mathcal{A}$ are invertible, it is easy to see from (6.6.3) that $\mathcal{O}'^{[k]} = \varnothing$ for all $k < \frac{n}{p}$. Moreover, it follows from Theorem 3 in [220] that there exist observable switched systems such that $\mathcal{O}'^{[k]} = \varnothing$ for all $k \leq \frac{n(n+1)}{2} - 1$. Conversely, it can be obtained from the proof of Theorem 1 in [219] that if a switched system is observable then $\mathcal{O}'^{[k]} \neq \varnothing$ for all $k \geq \frac{n(n^2-1)}{2}$.

**Example 6.1.** Consider the switched linear system with two modes, i.e. $\mathcal{I} = \{1, 2\}$, defined by the set of matrices $\mathcal{A} = \{A_1, A_2\}, \mathcal{C} = \{C_1, C_2\}$, where

$$A_1 = I_3,\ A_2 = 1.5 \times \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$C_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ and $C_2 = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$. For $k = 3$, we get that

$$\mathcal{O}'^{[k]} = \{121, 122, 212, 221\}.$$

Applying Algorithm 6.1, we obtain the DBA represented in Figure 6.7.1 where transitions are represented with arrows of different types depending on which phase of the algorithm they have been added. Let us comment the influence of $k$ on our construction. It is not hard to check that, for $k \geq 3$,

$$\mathcal{O}'^{[k]} = \{121, 122, 221\} \cup \{21^l 2 \mid l = 1, \ldots, k-2\}.$$

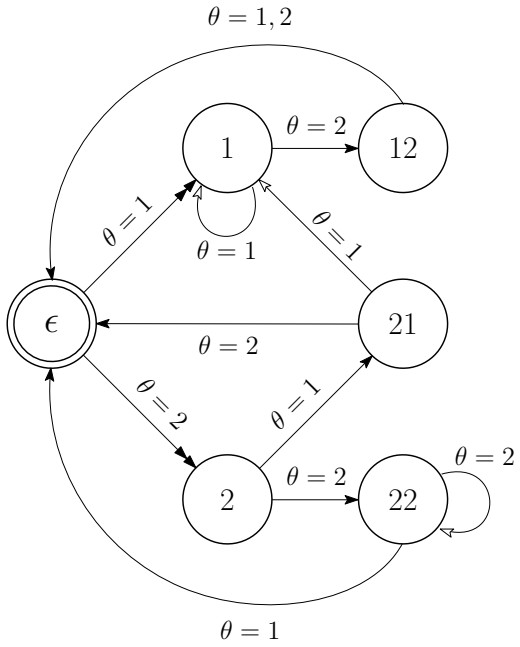


FIGURE 6.7.1: The generated DBA $\mathcal{B}_k$ corresponding to $\mathcal{O}'^{[k]} = \{121, 122, 212, 221\}$ for $k = 3$. The transitions built in part 1, 2 and 3 of Algorithm 1 are represented respectively by arrows with double head, single filled head and single empty head.

So the cardinalities of $\mathcal{O}'^{[k]}$ and of the set of states of $\mathcal{B}_k$ are $k+1$ and $k+3$, respectively. This corresponds to linear growths far from the exponential growth bounds discussed in Remark 6.2.

### *Characterisation of the language*

The following result provide a characterisation of the language generated by the DBA $\mathcal{B}_k$.

**Theorem 6.6.** *Let $\mathcal{B}_k$ be the DBA constructed using Algorithm 6.1 for some $k \geq 1$, then $Lang(\mathcal{B}_k) = (\mathcal{I}^* \mathcal{O}'^{[k]})^\omega$.*

## 6.8 ASYMPTOTIC OBSERVER DESIGN

In this section, we present an approach for designing asymptotic observers for system (6.6.1). The proposed design results in a switched observer where the switching is driven by the DBA $\mathcal{B}_k$ presented in the previous section. We first introduce the structure of the observer and then propose two approaches for designing the gains of the observer.

### 6.8.1 Observer structure

Let us consider the system (6.6.1) and let $\mathcal{B}_k = (Q, \mathcal{I}, \delta, q_0, F)$ be the DBA constructed using Algorithm 6.1, for some $k \geq 1$ such that $\mathcal{O}'^{[k]} \neq \varnothing$. We consider the following switched observer where $\hat{x}(t) \in \mathbb{R}^n$ is the estimate of the state of (6.6.1) and $q(t) \in Q$ is an internal discrete state:

$$\begin{aligned} q(t+1) &= \delta(q(t), \theta(t)),\ q(0) = q_0, \\ \hat{x}(t+1) &= A_{\theta(t)}\hat{x}(t) + L_{(q(t),\theta(t))}\left(y(t) - C_{\theta(t)}\hat{x}(t)\right). \end{aligned} \tag{6.8.1}$$

Let us remark that the dynamics of $q$ is given by the transition function $\delta$ of the DBA $\mathcal{B}_k$. The structure of the observer being given by (6.8.1), it remains to design the observer gains $L_{(w,i)}$, for all $w \in Q$ and $i \in \mathcal{I}$.

Let $e(t) = x(t) - \hat{x}(t)$ be the estimation error of the observer, then the dynamics of $e(t)$ is given by:

$$e(t+1) = (A_{\theta(t)} - L_{(q(t),\theta(t))}C_{\theta(t)})e(t).$$

We note that, given an initial condition $e_0 \in \mathbb{R}^n$, and a switching signal $\theta$, the trajectory with $e(0) = e_0$ is unique and denoted by $\mathbf{e}(., e_0, \theta)$. In order to guarantee the convergence of the asymptotic observer, one needs to prove that $\mathbf{e}(t, e_0, \theta)$ goes to 0 as $t$ tends to infinity. For this purpose, we adapt Theorem 6.2 to the estimation error dynamics.

**Theorem 6.7.** *Let us assume that there exist a function $V : Q \times \mathbb{R}^n \to \mathbb{R}_0^+$, and scalars $\alpha_1, \alpha_2, \rho > 0$, and $\lambda \in (0,1)$, such that for all $e \in \mathbb{R}^n$, the following inequalities hold:*

$$\alpha_1\|e\| \leq V(w,e) \leq \alpha_2\|e\|, \qquad w \in Q \tag{6.8.2}$$

$$V\left(w', (A_i - L_{(w,i)}C_i)e\right) \leq \rho V(w,e), \qquad w \in Q, i \in \mathcal{I}, \delta(w,i) = w' \notin F \tag{6.8.3}$$

$$V\left(w', (A_i - L_{(w,i)}C_i)e\right) \leq \rho\lambda V(w,e), \qquad w \in Q, i \in \mathcal{I}, \delta(w,i) = w' \in F \tag{6.8.4}$$

*Then, there exists $C \geq 1$ such that for all $e_0 \in \mathbb{R}^n$, and for all $\theta \in Lang(\mathcal{B}_k)$:*

$$\|\mathbf{e}(t, e_0, \theta)\| \leq C\rho^t\lambda^{\kappa^{\theta,\mathcal{B}_k}(t)}\|e_0\|,\ \forall t \in \mathbb{N}. \tag{6.8.5}$$

*Moreover, whenever the accepting rate satisfies $\gamma^{\theta,\mathcal{B}_k} > \frac{\ln(\rho)}{-\ln(\lambda)}$, we have*

$$\lim_{t\to\infty} \|\mathbf{e}(t, e_0, \theta)\| = 0. \tag{6.8.6}$$

Equation (6.8.5) relates the convergence rate of the asymptotic observer to the return index $\kappa^{\theta,\mathcal{B}_k}(t)$ associated with the DBA $\mathcal{B}_k$, which essentially counts the number of reconstructible sequences (i.e. elements of $\mathcal{O}'^{[k]}$) that occur up to time $t$. Note that $\rho$ may be greater than 1. In that case, in order to guarantee the convergence of the observer, one must consider switching signals where reconstructible sequences occur sufficiently often. This occurrence rate is measured by the accepting rate $\gamma^{\theta,\mathcal{B}_k}$. If $\gamma^{\theta,\mathcal{B}_k}$ is sufficiently large, we get from (6.8.6) that the observer asymptotically converges. In general the set of switching signals $\theta$ satisfying $\gamma^{\theta,\mathcal{B}_k} > \frac{\ln(\rho)}{-\ln(\lambda)}$ is not an $\omega$-regular language anymore but belongs to the class of quantitative languages, studied in [55].

*Remark* 6.3. Theorem 6.7 shows convergence of the observer for all switching signals $\theta$ such that $\gamma^{\theta,\mathcal{B}_k} > \frac{\ln(\rho)}{-\ln(\lambda)}$. We briefly discuss how this inequality can be verified for important classes of switching signals. Firstly, let us assume that the switching signal is a stochastic process generated by a finite state Markov chain $\mathcal{C}$. Then, by the ergodic theorem (see e.g. Theorem 1.10.2 in [158]) applied to the Markov chain obtained by the synchronous product of $\mathcal{C}$ and $\mathcal{B}_k$, one can compute a value $\gamma^*$ such that $\gamma^{\theta,\mathcal{B}_k} = \gamma^*$, almost surely. Secondly, let us assume that the switching signal is non-stochastic but constrained by some automaton $\mathcal{A}$ (see e.g. [169]). Let us remark that this case includes classical minimal or maximal dwell time constraints. Using algorithms from model checking [14], one can check if $Lang(\mathcal{A}) \subseteq Lang(\mathcal{B}_k)$ where $Lang(\mathcal{A})$ denotes the set of switching signals generated by automaton $\mathcal{A}$. Then, by using the techniques for quantitative languages developed in [55] applied to the synchronous product of $\mathcal{A}$ and $\mathcal{B}_k$, it is possible to compute the value $\gamma^* = \inf\{\gamma^{\theta,\mathcal{B}_k} | \ \theta \in Lang(\mathcal{A})\}$. Hence, we can see that for two broad classes of switching signals, there exist effective procedures to verify whether the condition $\gamma^{\theta,\mathcal{B}_k} > \frac{\ln(\rho)}{-\ln(\lambda)}$ holds.

### 6.8.2 *Explicit design of observer gains*

In this subsection, we present a particular design of the observer gains $L_{(w,i)}$, $w \in Q$ and $i \in \mathcal{I}$, and of the associated Lyapunov function $V$. The proposed design requires the following property which is assumed to hold throughout the subsection:

**Assumption 6.3.** *$A_i$ is invertible for all $i \in \mathcal{I}$.*

Let us fix $\rho > \rho_e(\mathcal{A})$ where $\rho_e(\mathcal{A})$ denotes the ellipsoid norm approximation of the joint spectral radius of $\mathcal{A}$ [34]:

$$\rho_e(\mathcal{A}) = \inf\left\{\rho \geq 0 \;\middle|\; \begin{array}{c} \exists M > 0,\ M^T = M, \\ \forall A \in \mathcal{A},\ A^T M A \leq \rho^2 M \end{array}\right\}.$$

Then, by definition, there exists $M \in \mathbb{R}^{n\times n}$ symmetric positive definite such that

$$A_i^T M A_i \leq \rho^2 M,\ \forall i \in \mathcal{I}. \tag{6.8.7}$$

Let us also consider an arbitrary $\lambda \in (0,1)$.

*Remark* 6.4. The results presented in this section can be extended to the case $\rho = \rho_e(\mathcal{A})$ if there exists a symmetric positive definite matrix $M$ such that (6.8.7) holds.

In the following, we provide an explicit construction of observer gains $L_{(w,i)}$ and of a Lyapunov function $V$ under the form $V(w,e) = \sqrt{e^T P_w e}$.

Let us associate a matrix $P_w \in \mathbb{R}^{n\times n}$ with every word $w \in \mathcal{I}^*$, defined recursively as follows:

$$P_\epsilon = M \tag{6.8.8}$$

$$P_{w\sigma} = \rho^2 A_i^{-T} P_w A_i^{-1} + \frac{\rho^2}{\gamma\lambda^2} A_i^{-T} C_i^T C_i A_i^{-1}, \quad \forall w \in \mathcal{I}^*, i \in \mathcal{I}, \tag{6.8.9}$$

and define the gain matrices $L_{(w,i)} \in \mathbb{R}^{n\times p}$, for all $w \in \mathcal{I}^*$, $i \in \mathcal{I}$ as follows:

$$L_{(w,i)} = A_i P_w^{-1} C_i^T \left(\gamma\lambda^2 I_p + C_i P_w^{-1} C_i^T\right)^{-1}. \tag{6.8.10}$$

We can now prove the following result.

**Theorem 6.8.** *Let $\mathcal{B}_k$ be the DBA constructed using Algorithm 6.1 for some $k \geq 1$, let $\rho > \rho_e(\mathcal{A})$ and $\lambda \in (0,1)$. Under Assumption 6.3, for all $w \in Q$ and $i \in \mathcal{I}$ let the matrices $P_w$ and $L_{(w,i)}$ be defined as in (6.8.8), (6.8.9) and (6.8.10). Then, $P_w > 0$, for all $w \in Q$, and*

$$(A_i - L_{(w,i)}C_i)^T P_{w'} (A_i - L_{(w,i)}C_i) \leq \rho^2 P_w, \qquad w \in Q, i \in \mathcal{I}, \delta(w,i) = w' \notin F \tag{6.8.11}$$

$$(A_i - L_{(w,i)}C_i)^T P_{w'} (A_i - L_{(w,i)}C_i) \leq \rho^2\lambda^2 P_w, \qquad w \in Q, i \in \mathcal{I}, \delta(w,i) = w' \in F \tag{6.8.12}$$

*In particular, the function $V(w,e) = \sqrt{e^T P_w e}$ satisfies inequalities (6.8.2), (6.8.3) and (6.8.4).*

Hence, in this section, we have shown that when all matrices in $\mathcal{A}$ are invertible, for all $\rho > \rho_e(\mathcal{A})$ and $\lambda \in (0,1)$ it is possible to design observer gains such that conditions of Theorem 6.7 are satisfied. In particular, for a given $\gamma^* > 0$, chosen for instance according to the approach described in Remark 6.3, one can appropriately choose the parameter $\lambda$ to achieve any arbitrary convergence rate for the error dynamics for all switching signal $\theta$ such that $\gamma^{\theta,\mathcal{B}_k} \geq \gamma^*$. However, it should be noted that the faster the convergence rate, the larger the magnitude of the observer gains is.

### 6.8.3 LMI-based design of observer gains

In this subsection, we present an alternative design of the observer gains based on solving a set of linear matrix inequalities.

**Proposition 6.1.** *Let $\mathcal{B}_k$ be the DBA constructed using Algorithm 6.1 for some $k \geq 1$, let $\rho > 0$ and $\lambda \in (0,1)$. Let us assume that there exist matrices $P_w \in \mathbb{R}^{n\times n}$, $Y_{(w,i)} \in \mathbb{R}^{n\times p}$, for $w \in Q$, $i \in \mathcal{I}$, such that the following LMIs hold:*

$$P_w > 0, \qquad w \in Q, \tag{6.8.13}$$

$$\begin{bmatrix} P_{w'} & P_{w'}A_i - Y_{(w,i)}C_i \\ \star & \rho^2 P_w \end{bmatrix} \geq 0 \qquad w \in Q, i \in \mathcal{I}, \delta(w,i) = w' \notin F \tag{6.8.14}$$

$$\begin{bmatrix} P_{w'} & P_{w'}A_i - Y_{(w,i)}C_i \\ \star & \rho^2\lambda^2 P_w \end{bmatrix} \geq 0 \qquad w \in Q, i \in \mathcal{I}, \delta(w,i) = w' \in F \tag{6.8.15}$$

*Then, the function $V(w,e) = \sqrt{e^T P_w e}$ satisfies inequalities (6.8.2),(6.8.3) and (6.8.4) with observer gains*

$$L_{(w,i)} = P_{\delta(w,i)}^{-1} Y_{(w,i)}, \ w \in Q, \ i \in \mathcal{I}. \tag{6.8.16}$$

*Moreover, under Assumption 6.3, LMIs (6.8.13), (6.8.14), (6.8.15) have a feasible solution for all* $\rho > \rho_e(\mathcal{A})$ *and* $\lambda \in (0,1)$.

From Theorem 6.8 and Proposition 6.1, it appears that under Assumption 6.3, it is always possible to design gains for the switched observer (6.8.1), either using the explicit construction or by solving LMIs. In this sense, the proposed observer structure given by (6.8.1) is universal. Let us remark that the observer gains need not be computed online as opposed to approaches such as [193] or based on Kalman filters [217], resulting in much simpler implementations in practice.

### 6.8.4 *Extension to switched systems with constrained switching*

In this subsection, we briefly explain how our approach can be adapted when the switched system (6.6.1) is subject to switching constraints. Let us assume that switching signals $\theta$ belong to a subset $S$ of $\mathcal{I}^\omega$. We start by remarking that we can use the exact same approach as described in the previous subsections, since the conclusions of Theorem 6.7 hold for all $\theta \in S \cap Lang(\mathcal{B}_k)$. However, we can use the additional information provided by $S$ to design observers with lower complexity or higher performance.

Firstly, due to switching constraints, not all reconstructible sequences may appear in the switching signals. We denote by $S_k$ the set of subsequences of length up to $k$ of sequences in $S$. Formally, let

$$S_k = \left\{ w \in \mathcal{I}^* \;\middle|\; \begin{array}{c} |w| \leq k,\ \exists w_1 \in \mathcal{I}^*, w_2 \in \mathcal{I}^\omega, \\ w_1 w w_2 \in S \end{array} \right\}.$$

Then, it is sufficient to consider the set of reconstructible sequences given by $\tilde{O}'^{[k]} = O'^{[k]} \cap S_k$. Then, we can design an observer using the same approach as before, simply building the automaton $\mathcal{B}_k$ from the set of reconstructible sequences $\tilde{O}'^{[k]}$ instead of $O'^{[k]}$.

Additionally, if $S$ is itself generated by a DBA $\mathcal{B}$, i.e. $S = Lang(\mathcal{B})$, then we can use the following approach. We start by computing a DBA $\tilde{\mathcal{B}}_k$ such that $Lang(\tilde{\mathcal{B}}_k) = Lang(\mathcal{B}_k) \cap S$. This is always possible, following the approach described in [14, Lemma 4.59 and Theorem 4.56]. One can then adapt the approach described in Subsection 6.8.3 with the DBA $\tilde{\mathcal{B}}_k$ instead of $\mathcal{B}_k$.

## 6.9 CASE STUDY: THE MULTICELLULAR CONVERTER

Let us consider the circuit of the multicellular converter taken from [201, 200] and shown in Figure 6.9.1. This system is an example of switched dynamical systems due to several commutation cells in the circuit. The dynamics of this converter is described

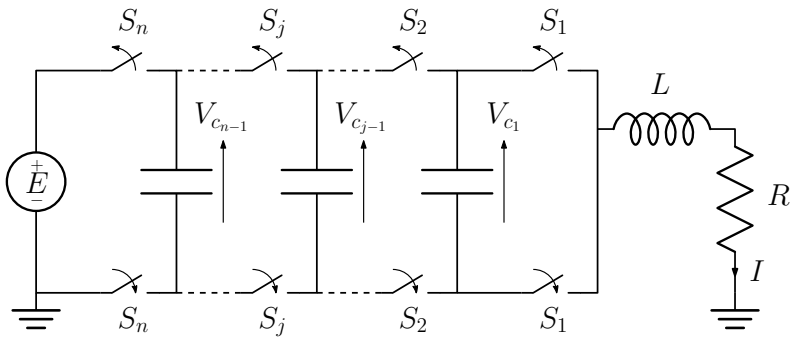


FIGURE 6.9.1: Multicellular converter with an inductive load.

by the following equations:

$$\dot{I} = -\frac{R}{L}I + \frac{E}{L}S_n - \sum_{j=1}^{n-1}\frac{V_{c_j}}{L}(S_{j+1} - S_j)$$
$$\dot{V}_{c_j} = \frac{I}{c_j}(S_{j+1} - S_j), \qquad j = 1, \ldots, n-1$$

where $c_j$ and $V_{c_j}$ are the capacitance and the voltage of the $j$-th capacitor respectively, $I$ is the current passing through the load consisting of the resistor $R$ and the inductor $L$, $E$ is the voltage of the source. $S_j \in \{0,1\}$ is a binary signal corresponding to the $j$-th commutation cell. When $S_j = 1$, the upper switch of the $j$-th cell is ''on'' and the lower switch is ''off'' and vice versa in the other case.

We consider the bijection between the set $\{0,1\}^n$ and the set $\{1, \ldots, 2^n\}$ which maps the binary vector $(S_1, \ldots, S_n)$ to $\theta = 1 + \sum_{j=0}^{n-1} 2^j S_{j+1}$. Writing $x = (V_{c_1}, \ldots, V_{c_{n-1}}, I)^T \in \mathbb{R}^n$, $u = E$, and assuming only $y = I$ is measured, the system dynamics can be described by a continuous time switched system of the form:

$$\begin{aligned} \dot{x}(t) &= A_{\theta(t)}x(t) + B_{\theta(t)}u(t), \\ y(t) &= C_{\theta(t)}x(t) \end{aligned} \tag{6.9.1}$$

For numerical experiments, we consider $n = 3$ commutation cells, with the following parameter values $E = 1500\,\mathrm{V}, c_1 = c_2 = 40\,\mu\mathrm{F}, R = 10\,\Omega, L = 0.5\,\mathrm{mH}$. For such

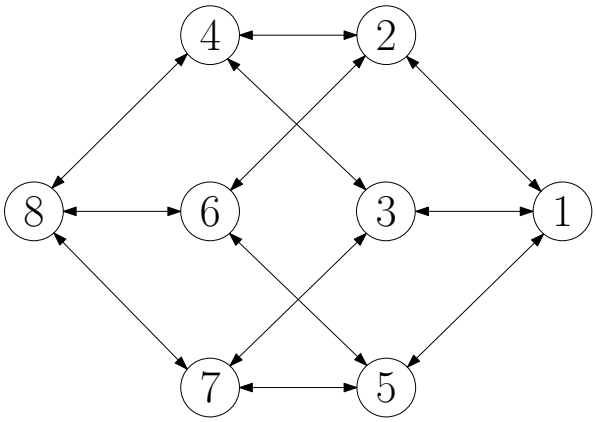


FIGURE 6.9.2: Adjacency relations between modes describing the constraints on the switching signal $\theta$. When a mode change occurs in the switching signal $\theta$, the new mode must be adjacent to the previous one.

numerical values, the state matrices are given by

$$A_1 = 10^3 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -20 \end{bmatrix}, \qquad A_2 = 10^3 \times \begin{bmatrix} 0 & 0 & -25 \\ 0 & 0 & 0 \\ 2 & 0 & -20 \end{bmatrix}$$

$$A_3 = 10^3 \times \begin{bmatrix} 0 & 0 & 25 \\ 0 & 0 & -25 \\ -2 & 2 & -20 \end{bmatrix}, \qquad A_4 = 10^3 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -25 \\ 0 & 2 & -20 \end{bmatrix}$$

$$A_5 = 10^3 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 25 \\ 0 & -2 & -20 \end{bmatrix}, \qquad A_6 = 10^3 \times \begin{bmatrix} 0 & 0 & -25 \\ 0 & 0 & 25 \\ 2 & -2 & -20 \end{bmatrix}$$

$$A_7 = 10^3 \times \begin{bmatrix} 0 & 0 & 25 \\ 0 & 0 & 0 \\ -2 & 0 & -20 \end{bmatrix}, \qquad A_8 = 10^3 \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -20 \end{bmatrix}$$

while the output matrices are $C_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, for all $i = 1, \ldots, 8$. In the following, we consider a sampled version of (6.9.1) with a sampling period $T = 0.3$ ms.

Let us remark that the dynamics in all modes are unobservable. Hence, it is not possible to build an observer that is convergent for arbitrary switching signals. Additionally, the system is subject to the following constraint on the switching signal [201, 200]: only one commutation cell can be switched at a time. This constraint translates to some adjacency relations between modes described by the graph in Figure 6.9.2. When a mode change occurs in the switching signal $\theta$, the new mode must be adjacent to the previous one. Our goal is to design an observer for this switched system with constrained switching. The approach presented in [201, 200] is similar to that of [193] and requires computing the observer gains online.

As described in Subsection 6.8.4, we compute the set $\tilde{\mathcal{O}}'^{[k]}$ of reconstructible sequences satisfying the switching constraints, which, for $k = 3$, contains 48 sequences. We also checked numerically that the cardinality of this set satisfies $|\tilde{\mathcal{O}}'^{[k]}| = 48(k-2)$ for $k = 3, 4, 5, 6, 7$. This suggests that this set increases linearly with $k$. Then, we use Algorithm 6.1 to build the corresponding Büchi automaton $\mathcal{B}_3$. As (6.9.1) possesses a non-strict common quadratic Lyapunov function (the energy of the system) and thanks to Remark 6.4, we can apply the explicit design of observer gains presented in Subsection 6.8.2 for $\rho = 1$ and $\lambda = 0.1$. Let us remark that switching sequences satisfying adjacency constraints can be generated using a DBA $\mathcal{B}$ (see e.g. [207]). Then, as explained in Subsection 6.8.4, we can further compute the DBA $\tilde{\mathcal{B}}_3$ shown in Figure 6.9.3 and such that $Lang(\tilde{\mathcal{B}}_3) = Lang(\mathcal{B}_3) \cap Lang(\mathcal{B})$. Then, we solve the corresponding LMIs (6.8.13), (6.8.14), (6.8.15) with $\rho = 1$ and $\lambda = 0.1$ to design observer gains.

Numerical simulations are shown in Figure 6.9.4. The figure shows the switching signal $\theta$; the return instants (i.e. instants where $q$ visits an accepting state) are indicated with red circles. We also show the evolution in logarithmic scale of the norm of the

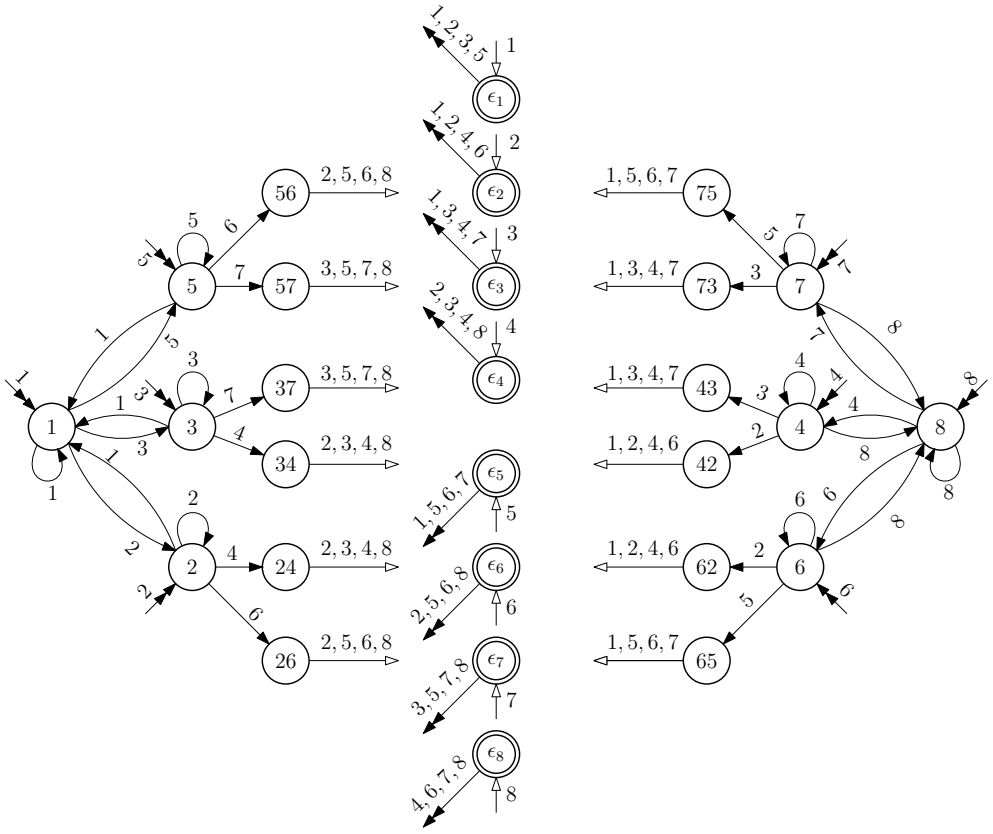


FIGURE 6.9.3: DBA $\tilde{\mathcal{B}}_3$ that generates switching sequences satisfying the adjacency constraints shown in Figure 6.9.2 and with an infinite number of reconstructible subsequences in $\tilde{\mathcal{O}}'^{[k]}$. Arrows with double head and labelled by $i$ lead to state $i$, arrows with empty head and labelled by $i$ lead to accepting state $\epsilon_i$.

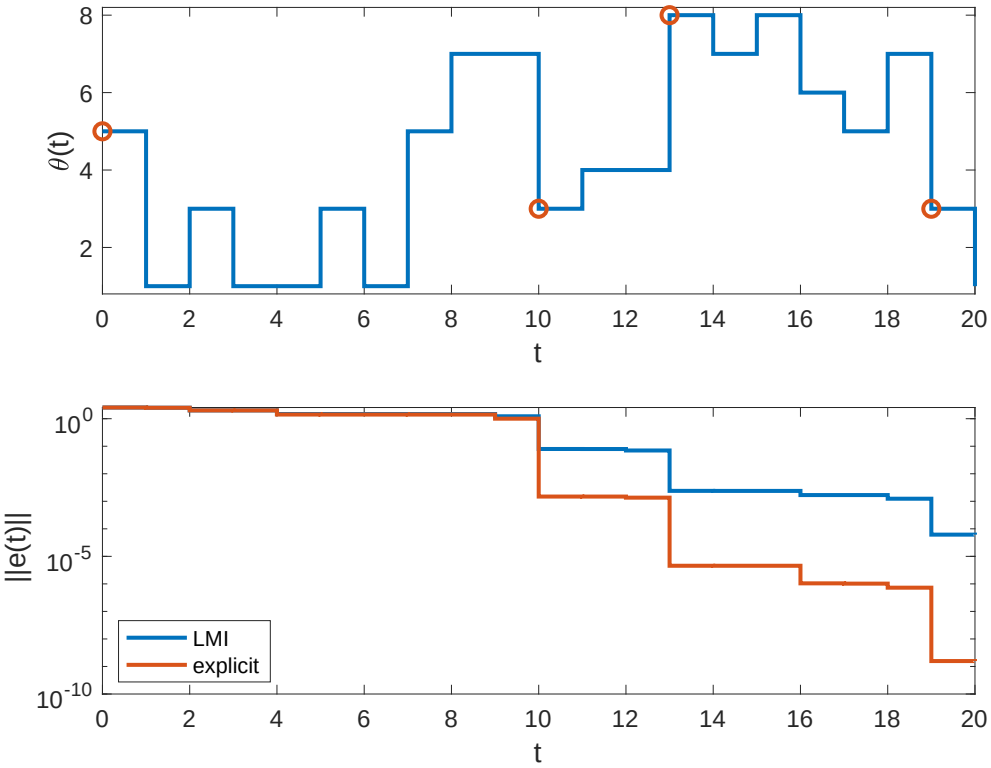


FIGURE 6.9.4: Simulations for the multicellular converter: top - switching signal $\theta$, red circles indicate return instants; bottom - norm of the estimation error $e$ in logarithmic scale for both type of gains.

estimation error $e$ for both designs. We can see again that the observer based on the explicit design converges faster than the one based on LMIs whose convergence rate is more consistent with the theoretical guarantees provided by Theorem 6.7.

We end this section by mentioning that in [201, 200], they consider a sampling period $T = 1\,\text{ms}$. In that case, the state matrices of the sampled system are numerically close to singularity and thus Assumption 6.3 is not robustly satisfied. This results in the explicit design of Subsection 6.8.2 to be numerically ill-conditioned. In practice, due to numerical errors, the ensuing observer is divergent. However, it is important to remark that even in that case, the observer based on the LMI design still works smoothly.

## 6.10 CONCLUSIONS

In this chapter, we considered discrete-time switched linear systems whose switching signals are generated by a Büchi automaton. This class of signals allows to describe constraints defining the QoS of a platform that cannot be modelled with classical approaches (e.g. dwell time, maximum delays or latency, etc.). We analysed the problem of ensuring the stability of this class of switched systems and we applied those results to

the design of a switched observer for the estimate of the continuous state of the system. The main contributions of the chapter are as follows.

*Suitable stability definitions.* We defined for this class of systems the notions of attractivity, uniform stability, and in the case of deterministic Büchi automaton, of uniform exponential stability.

*Sufficient and necessary conditions.* We established sufficient conditions for attractivity and uniform stability and also of uniform exponential stability when the considered Büchi automaton is deterministic, all based on Lyapunov arguments. Note that these conditions can readily be extended to the case of nonlinear systems. Moreover, we proved that these conditions are also necessary for a subclass of such systems with invertible matrices. We also showed, through a numerical example, how these Lyapunov functions can actually be computed using a convex optimisation problem based on LMIs.

*A Büchi automaton-based observer.* We developed a procedure to build a deterministic Büchi automaton (DBA) generating the set of switching signals containing an infinite number of reconstructible sequences. We proposed a switched observer driven by the switching signals produced by the DBA.

*A universal observer structure.* The most important feature of the proposed observer structure is that it is universal as we show that it is always possible to design observer gains that make this observer convergent.

*Two methods for computing off-line the gains.* We presented two approaches to compute off-line the observer gains. The first is LMI-based and the second is direct, even if viable only when all state matrices of the switched system are invertible. We also showed that the LMIs of the first method always admit a feasible solution under the same invertibility assumption. We stress that the gains of our observer are computed off-line, which is a highly attractive feature from the point of view of their implementation.

# 7

# FUTURE RESEARCH DIRECTIONS

## 7.1 INTRODUCTION

This Chapter presents the main research directions I aim to pursue in the future.

In the latest years I have begun a thematic shift towards topics involving systems and control theory applied to neuroscience and health care. While still working on resource-aware control and switched systems, my efforts are now chiefly devoted to the new topics. This change in interests is clearly visible in the structure of the present Chapter.

Section 7.2 briefly presents some questions related to resource-aware control I want to investigate. They focus on the development of the topics introduced in Chapter 6: switched systems driven by Büchi automata.

Section 7.3 introduces a very interesting phenomenon, the self-organised criticality, arising in many physical and biological systems. After a general presentation of the concepts of criticality and self-organised criticality in physical systems, the tantalising, and very debated, hypothesis of critical brain is discussed. Then, the main theses in favour and against the criticality hypothesis are summarised according to the current literature. Finally, a mathematical description of the self-organised criticality in the framework of systems and control theory is proposed and a class of simple, but physiologically meaningful, systems is introduced. The section ends with the presentation of the main questions I would like to answer from the most specific to the most general ones.

Finally, Section 7.4 presents the relevant problem of synchronising the natural breath of a patient with the respiratory support provided by a mechanical ventilator. After a quick overview of mechanical ventilation in general, the analysis focuses on a widely adopted ventilation mode (Pressure Support Ventilation). The main types of asynchrony are defined and the synchronisation problem formulated. Due to a patent

application still pending, no details can be provided on the synchronisation strategy. However, many future lines of research related to this topic are discussed.

## 7.2 RESOURCE-AWARE CONTROL

My latest developments concerning resource-aware control deals with discrete-time switched systems whose switching signals are described in terms of $\omega$-regular languages. This section presents some interesting open problems in this topic I would like to investigate.

*Occurrence frequencies.* Let us consider the switched system (6.2.1) driven by a deterministic Büchi automaton (DBA) $\mathcal{B}$ and where the matrices $A_i \in \mathcal{A}$ can be unstable. Given a specific DBA the stability conditions are provided by Theorem 6.2. The DBA is a way to describe special constraints, like in the case of shuffling systems. However, in Chapter 6, no particular knowledge was assumed on the frequency of occurrence of the individual modes (matrices $A_i$). Nevertheless, in many practical applications some information on the frequency is actually available. For instance, some modes could require a periodic activation, e.g. once each $N$ steps for some fixed $N > 0$, or an average periodic activation, e.g. $N_1$ times on any window of $N$ steps. The results in Theorem 6.2 are still valid, but potentially conservative. Explicitly taking into account the knowledge of the frequency of occurrence should lead to less restrictive stability conditions, e.g. a smaller acceptance rate $\gamma$ required to achieve the same convergence rate $\lambda$, or a smaller $\lambda$ for the same $\gamma$. Those frequencies, instead of being considered just as constraints, can also be used as QoS parameters to tune in order to obtain a desired rate of convergence (QoC) of the switched system. Exploiting classical compositional approaches, a suitable Büchi automaton incorporating the information on modes frequencies can easily be built. However, it is not apparent how to modify the stability result of Theorem 6.2 to make this information directly accessible through $\lambda$ and $\gamma$.

Related to the occurrence problem there is a more general question deserving investigation. Given some accepting constraints, such as the shuffling one, how can one choose the composition of the switching signals to obtain a desired $\lambda$ and $\gamma$? In other words, which restrictions can be applied to the language generated by the original Büchi automaton to obtain a desired QoC?

*Performance.* Up to now, we have considered stability problems for switched systems driven by a Büchi automaton. A relevant aspect, in particular from the applicative point of view, is how to incorporate performance requirements in this class of systems. Even if the rate of convergence can be considered a performance metrics, other QoC are commonly adopted, e.g. $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norms. A good track in this direction could be the results in [68]. Even if they are developed for periodic switching laws similar in spirit to the min projection strategy, the main tools should be amenable of an adaptation to suit our class of systems.

*Switched controllers.* In Chapter 6 the design of an asymptotic observer was considered as an application of the stability results on switched systems driven by $\omega$-regular languages. Using a similar approach one can design switched controllers to stabilise

discrete-time switched linear systems. However, the problem is more complex than just exploiting the duality between observability and controllability. One can characterise the controllable sequences of given length and also prove a sufficient condition for stability like that in Theorem 6.7 (even in the form of LMI conditions). Problems arise when one wants to prove that, if controllable sequences of a given length exist, then any convergence rate $\lambda$ is attainable through a suitable design of feedback gains. This seems not to be possible in general. In the case of observability, the knowledge of the past (i.e. previously activated modes) was sufficient to build an observer whose error can be made to converge arbitrarily fast. In the case of controllability, ensuring stability with an arbitrary convergence rate requires, in general, the knowledge of the modes that will be activated in the future. This fact does not exclude the possibility of designing such controllers for particular classes of systems on the basis of worst case estimation of the future modes. One can, for instance, exploit some partial knowledge on the structure of the sequences in terms periodicity or average periodicity. Generally speaking, as discussed for the occurrence frequencies problem, the question is how can restrictions be imposed on the language generated by the original Büchi automaton to allow the design of feedback gains ensuring a prescribed convergence rate?

*Continuous shuffling systems.* So far, the set $\mathcal{A}$ in which the matrices of the modes take value has been discrete and finite. What does it happen if $\mathcal{A}$ is a compact set? This direction is more exploratory and requires the definition of a new concept of shuffling signal. Roughly speaking, to be shuffling, the function $A(\cdot)$ describing the active system matrix at any instant $t$, must pass an infinite number of times by any subset of $\mathcal{A}$. The idea is akin to the ergodicity property of stochastic processes. The problem is to find conditions for the stability of such continuous shuffling system. If $\mathcal{A}$ is a polytope, then a sufficient condition is given by the stability for arbitrary switching, namely by ensuring the stability of the system that arbitrarily switches among the matrices that are vertices of the polytope. However, this is an over-conservative condition that ignores the shuffling constraint. An idea worth exploration is to partition the set $\mathcal{A}$ in a finite number of sub-polytopes $\mathcal{A}_i$. With each sub-polytope $\mathcal{A}_i$ we can associate a system $\Sigma_i$ switching among the vertex matrices of $\mathcal{A}_i$. Then we can impose a shuffling constraint on the switched sub-systems, that is each $\Sigma_i$ must be activated an infinite number of times. By mixing conditions of stability for arbitrary switching on each $\Sigma_i$ wrt $\mathcal{A}_i$ and condition of shuffling stability among the different sub-systems, one could find sufficient conditions for the stability of the continuous shuffling systems. Those conditions should be the less restrictive the more the partition is refined. Given a certain partition, one can compute the Shuffle Joint Spectral Radius (SJSR) [1] of the ensuing system and, when the number of sub-polytopes tends to infinity, it should tend to the SJSR of the continuous shuffling system from above. For each partition a Büchi automaton with a finite number of states can be built and one can look for a trade-off between the number of states and the precision of the approximation of the continuous SJSR.

## 7.3 A CONTROL THEORY VIEW ON SELF-ORGANISED CRITICALITY

Self-organised criticality (SOC) is a phenomenon observed in many physical and biological systems. It characterises dynamic systems which tend autonomously towards a critical equilibrium, lying at the boundary between two qualitatively distinct behaviours. SOC plays a crucial role in the emergence of complexity since, for such systems, small disturbances can generate radically different responses. A classic example of SOC is the pile of sand to which grains are added one by one: the pile of sand will eventually reach a critical state and the addition of a grain of sand will (or not) lead to an avalanche of a more or less strong intensity.

Many theoretical ([18, 187, 111]) and experimental studies suggest that the brain may operate near criticality, which would contribute to its optimal computational and memory capabilities relative to its metabolic resources. Some studies (see for instance the discussion at the end of [187]) also propose that some pathological conditions, like epilepsy or autism, could be related to a cortex not perfectly tuned to criticality. Indicators of the critical regime have been found in cell cultures ([21, 22, 194]) animals ([167, 105]) and humans ([119, 148]). These observations have led to the hypothesis that the brain is capable of self-regulating through homeostatic mechanisms around a critical configuration.

Despite these studies, the critical brain hypothesis remains a subject of ongoing debate in the neuroscience community ([23, 19]), particularly highlighted by recent studies ([196, 70]) that have identified non-critical neural networks that also exhibit avalanches with power law distributions, a characteristic previously considered as a signature of critical systems. If it is nowadays generally accepted that these power law distributions are not the ultimate proof of criticality, it is not clear which and how many indicators are needed to draw a definitive conclusion.

One of the difficulties associated with the current study of the SOC phenomenon is the adoption of large-dimensional models, whose dynamics are simulated and then analysed with statistical tools. Due to their size, these models are not very well suited for a formal analysis of their dynamic properties. For this reason, we want to bring to this debate the point of view of the systems and control theory, by interpreting the criticality as a bifurcation and the SOC as a feedback. This control-oriented vision of the SOC has received few attention to date, with the notable exception of [153, 73], which however consider very specific classes of systems. The difficulty comes from the intrinsically non-linear nature of the dynamics involved, as well as the non-standard nature of the control objective, which can be seen neither as stabilisation nor as trajectory following.

To tackle this complex problem we plan to start with simple, low-dimensional (possibly deterministic) neuronal population models and explore the feedback mechanisms that can lead the system to self-regulate around a bifurcation point. It will be a question, for these systems, of developing biologically realistic strategies for regulating the bifurcation parameter (the synaptic weight linking one population to the other) and of determining what minimum information on each of the populations is necessary

to obtain the SOC. Later on, we want to extend our research to large-dimensional models, where the knowledge acquired on the small models will be exploited to try and reconcile the point of view of statistical physics and neuroscience with that of control and systems theory.

In the following subsections we will introduce the concept of criticality and self-organised criticality as they are commonly intended in the community of statistical physics and neuroscience. We will present the brain criticality hypothesis as well as the main arguments in favour and against such hypothesis. We will propose a formulation of the SOC as a bifurcation tuning problem and we will conclude presenting the main research directions we intend to pursue in this topic.

### 7.3.1 Criticality as a bridge between microscopic and macroscopic models

What is criticality? What does it mean for a system to be in a critical state?

The concept of criticality is deeply rooted in thermodynamics and statistical physics and is strictly related to that of *phase transition* (see for instance [32]). In studying complex systems one can adopt a *microscopic* description considering a huge number of constituent elements (particles) and all their interactions. Gases are made up of molecules moving and colliding in a given volume, crystals are described in terms of their reticular structure and the forces of their bondings, magnets are described by spins and nearest neighbourhood interactions, etc. A microscopic model can give a very precise description of the basic mechanisms dictating the different behaviours of a system, but its complexity becomes quickly overwhelming if a significant number of particles are considered. On the other hand, many properties and behaviours of such systems are emergent phenomena, which are more aptly described by *macroscopic* models. Properties of fluids like density and viscosity do not belong to any specific molecule in the fluids, but are the macroscopic (emergent) result of all their interactions. Macroscopic models are defined in terms of properties belonging to the system as a whole and in terms of (simple) laws describing how those properties evolve. A key concept in statistical physics linking the microscopic and macroscopic descriptions is that of *phase*. Phases represent qualitatively different behaviours (or macroscopic states) of a system; water, for instance, can be in gaseous (vapour), liquid or solid (ice) state. By varying some global parameter such as temperature or pressure, one can induce a phase transition in water from liquid to solid or from gas to liquid, etc. In general, phase transitions are described in terms of an *order parameter* accounting for the macroscopic state (for instance the aggregation state of molecules in a crystal structure) and of a *control parameter* (for instance temperature), whose variation determines the change of phase. It is important to stress that the control parameter does not make sense in microscopic terms: there is no molecular temperature. It is an emergent property of the collective behaviour of the particles: the motion and related collisions in the case of molecules. Order parameters have at least one *critical value* marking the boundary between two distinct phases.

Phase transitions are organised in classes according to the way the order parameter changes in response to a variation in the control parameter. *First order transitions*

are characterised by a discontinuity at the critical value. In other words, the order parameter makes a finite jump when the order parameter crosses the critical value. The transition from liquid to solid in water is a first order transition: above the freezing temperature there is no crystal structure, below it all the molecules are packed in a crystal structure. *Second order transitions* have a continuous nature, even if they are usually not differentiable at the critical value. A peculiarity of second order transition is that the system can dwell exactly at the critical value. In this case the system is said to be in a *critical state* (or to be *at criticality*), i.e. a state at the boundary of two qualitatively different behaviours ([111]). Systems in critical state can manifest very peculiar properties.

### 7.3.2 Power laws and criticality

Natural systems are often empirically characterised and analysed in terms of statistics of some physically relevant quantities. In earth-quake and fires, for instance, the histogram of events ranked by their magnitude is particularly important. Studying those statistics, physicists realised a striking constant: they are distributed as power laws. By the years, researchers have found a surprising number of phenomena exhibiting power law distributions: other than earthquakes and fires, also solar flares, crackling systems, percolation in porous media, snow avalanches, neural spikings, word distributions, city populations, etc. (see for instance [15, 32, 223, 185, 21]). For a quantity $N$, say the number of earthquakes in a given interval of years, to be expressed as a power law of another quantity $m$, say the magnitude of the earthquake, it means that $N(m) = m^{-\tau}$ for some exponent $\tau$. Represented in a doubly logarithmic scale, power laws appear as straight lines with slope $\tau$. The presence of power law statistics is usually interpreted as a sign of the *scale invariance* (or self-similarity) of the studied phenomenon (see for instance [185]). Indeed, a power law distribution for the quantity $N$ means that its graph, hence the underlying phenomenon, is essentially the same at any scale (in the admissible range of scales[1]).

Physical systems that undergo a second order phase transition exhibit power law statistics when they are at criticality. However, even if the presence of power laws is often considered an hallmark of criticality, not all the phenomena having power law statistics underpin a critical system. For instance, word distributions and city populations can be explained by non critical mechanisms ([36]). Critical systems, and sometimes also non critical ones (see criticisms in Section 7.3.5.2), have often a wealth of different quantities showing power law statistics. Physicists have scrutinised these quantities and collected the different exponents (sometimes called *critical exponents*) arising in the various statistics. An account of the main exponents can be found in [223, 32]. They discovered that, wildly different phenomena have the same critical exponents, thus opening the

[1] Any real physical system has a finite characteristic scale or length accounting for its finite spatial extension. In this case, the scale invariance property inevitably breaks out for scales larger or comparable to the characteristic one. It often breaks out also for very small scales as the individual microscopic effects are no longer negligible and the macroscopic model based on statistical abstraction is no longer valid. For this reason, even when we say 'at any scale' we really mean a large but finite interval of scales.

door to a potentially unified description of those phenomena in terms of *universality classes* (see again[223, 32]).

### 7.3.3 A problematic definition

Criticality, in the physical sense is an empirical, observational concept. In a real physical system or in a distributed model, accounting for the microscopic constituents and their interactions, it is necessary to identify two macroscopic quantities, i.e. the order and the control parameter, and at least two qualitatively different phases corresponding to distinct sets of values of the order parameter. Moreover, a continuous variation of the control parameter must induce a continuous (possibly non smooth) transition between the two phases. The criticality is the state of the system at the edge between these two phases where the control parameter assumes the critical value.

This definition can be problematic both from a mathematical and an operative point of view.

The function expressing the order parameter in terms of the control parameter is often represented through a graph called the *phase diagram*. This graph is, in general, constantly zero below the critical value (subcritical region), positive and increasing above the critical value (supercritical region) and presents a discontinuous derivative at the critical value. However, there is no clear indication of what mathematical properties such a function should enjoy.

From an operative point of view, it is not always clear how or even if the control parameter is accessible to manipulation in order to produce the graph of the aforementioned function. The latter is not a negligible point and represents one of the reason for the existence of the huge literature on scale invariance. The argument is as follows: if a phase diagram cannot be produced, then one can look for the presence of the scale invariance property as a sign of criticality.

Unfortunately, such kind of analysis is not less problematic. The scale invariance is the property of some phenomenon or physical quantity to present essentially the same patterns at any scale. This is again a purely empirical definition that requires, to be supported, a certain number of indicators. And there are a lot of them. We mentioned the power law statistics and the critical exponents with associated universality classes, but there are also mathematical relations among the exponents ([32, 185]), shape properties on some physical quantities often referred to as *scaling (or shape) collapse* ([185, 23]), the *critical slowing-down* criterion[2] ([184, 111, 196]), etc. The problem is that there is no clear consensus on which or how many of these indicators are necessary to properly define the scale invariance property. Moreover, even if all known critical systems exhibit some scale invariance property, there is no proof that all critical systems must do the same. Even worst, the presence of scale invariance does not necessarily imply that the underlying system is critical. In fact, there are many ways a system can

[2] The critical slowing-down is a phenomenon arising in dynamical systems close to a phase transition. According to it, the time taken by the system to recover from small perturbations slows down in the vicinity of and diverges at criticality.

produce a certain number (but maybe not all) of good indicators of scale invariance without being critical (see Section 7.3.5.2).

All these criticisms do not diminish the appeal and meaningfulness of the criticality from a physical point of view. Many physical systems are indeed critical and the brain could too, thus it makes perfectly sense to study and characterise such phenomenon. The criticisms should be taken as a warning that great caution must be exerted whenever dealing with critical phenomena. A more rigorous definition of criticality and, above all, more stringent tests, capable of ascertaining whether a system is actually critical or not, are much needed.

### 7.3.4 Fine tuning vs SOC and SOqC

Criticisms apart, the number of systems both in a critical state and exhibiting scale invariance seems to be substantial. So much so, that the pervasiveness of scale-invariant phenomena in nature is bewildering. Indeed, criticality would require a fine tuning of the control parameter to the critical value, which is often incompatible with physical systems subject to any kind of perturbation and fluctuation. A different mechanism must be in place to keep a system exactly at the criticality.

In the seminal paper [16], the authors proposed a toy system (the sandpile), which has since become paradigmatic, to elucidate a new concept dubbed *Self-Organised Criticality* (SOC). According to SOC, some systems posses, per their structure, the ability to recover from possible deviations and to auto-regulate about the criticality. The original sandpile model is described as follows (see for instance [16, 36]): consider a finite 2-dimensional lattice; each node (site) in the lattice can store a certain amount of sand grains up to a fixed threshold (identical for all the sites); an external driving mechanism exists to inject a small number of grains, usually one grain in a unique site; if the number of grains in a site exceed the threshold, the site is emptied and its grains redistributed among the neighbouring sites (toppling event); when the grains reach the boundary of the lattice, they are lost (open boundary conditions).

A toppling event can produce other toppling events in the neighbourhoods and so on, thus provoking an *avalanche*. The concept of avalanche must be intended in a general sense. It can go from a single toppling event to a very large number of consecutive events producing an effect that is both temporally and spatially extended. During avalanches no new grains are introduced in the system, the driving mechanism is activated when all events have ceased. After some time, the system reaches a statistically stationary state where the avalanches show some scale-invariant property, that is the distributions of their time duration and spatial extension are power laws, up to the characteristic size of the lattice.

The original sandpile has been shown to not present perfectly scale-free phenomena, but rather anomalous multi-scaling or no scaling at all (see for instance comments and references in [36]). Nonetheless, it is still regarded by physicist as very useful to elucidate some of the crucial ingredients of SOC mechanisms. The first, and most important, ingredient is the s*low driving/fast relaxation* couple. In other words, there must exists a significant separation in the time scales of the slow driving mechanism

and the fast avalanches. The separation should theoretically be infinite as it is in the original sandpile model, where a new grain is injected only after all avalanches have ceased. The second ingredient is some kind of *energy conservation* in the so-called bulk dynamics. During the redistribution phase, *all* the grains removed from the toppling site are transferred on the neighbouring sites, thus their quantity is preserved. It is worth noting that the overall quantity of grains in the sandpile is not constant, but fluctuates due to the driving mechanism and the losses at the open boundaries.

In order to study the criticality of the sandpile model one has to resort to a fixed energy sandpile, where both the driving mechanism and the boundary dissipation are absent. In this case the self-regulation mechanism is deactivated and the total amount of grains in the sandpile stays constant (fully conserved quantity). The total number of grains can be used as the control parameter and the number of sites above the threshold as the order parameter.

Systems with dissipative bulk dynamics have been shown (see for instance the forest-fires and earthquake models in [36]) to not exhibit true scale-invariance, even in the limit of infinite time-scale separation. This leads some physicists to maintain that true criticality is impossible without a conservative dynamics. Such systems are usually endowed with a *loading mechanism,* similar to the driving mechanism seen before, whose goal is to compensate for the loss in the relevant quantity due to the dissipative dynamics and boundaries. The result is usually a system that hovers around the critical state, with large excursions in the subcritical and supercritical phases. In these systems a self-regulating mechanism is clearly at work, but, due to the non perfect criticality, Bonachela and Muñoz proposed in [36] to dub it *Self-Organised quasi Criticality* (SOqC). SOqC systems can show true criticality only when the loading mechanism is finely tuned to perfectly compensate (equate) the dissipation. However, this would defeat the very role of a self-regulating mechanism.

### *7.3.5 Is brain critical?*

This is one of the most important question in modern neuroscience and a very debated one. We will briefly summarise here the main theses supporting the so-called *criticality hypothesis* and those against. For a thorough discussion on this complex topic, we refer the reader to the excellent surveys [23, 187, 19] and the book [20].

#### *7.3.5.1 The criticality hypothesis*

The criticality hypothesis states that neurons in many brain tissues collectively organise to be close to a critical state and in this condition they optimise many information transmission, processing and storage functions.

To better grasp how it can happen, one can build a very simple model of a network of neurons and analyse its behaviour for different levels of excitation propagation (see [187, 20]). Consider a network of $N$ identical neurons connected according to an all-to-all graph. When a neuron spikes it has a probability $p$ to induce a neuron, to which it is connected, to fire. In other words, an active neuron can propagate its

excitation to an average of $pN$ other neurons. If $p$ is low ($p \ll 1/N$), the excitation propagates to few neurons and the network activity quickly dies (subcritical regime). If $p$ is high ($p \gg 1/N$), each firing neuron excites a significant number of other neurons, quickly leading to a surge of activity in the entire network (supercritical regime). If $p$ is close to $1/N$, each spiking neuron activates, in average, one other neuron, thus producing a sustained but not overwhelming activity in the network (critical regime). Reinterpreting this model in terms of the theory of branching processes, one can easily recognise in the quantity $pN$ the *branching parameter* $\sigma$ and recall that $\sigma = 1$ is usually considered an hallmark of criticality (see for instance [111]). In this simplistic model one can perform an information transmission experiment [20]. A certain number of neurons is randomly excited at time $t_0$ and the state of the network is probed at a later time $t_p$. One can say that the original information is transmitted with fidelity if it is possible to guess the number of neurons initially excited at $t_0$ by the number of active neurons at the instant $t_p$. It is clear that, if the network is in a subcritical regime few, if any, neurons are active at $t_p$. In the supercritical regime, on the contrary, the number of neurons firing at time $t_p$ is much larger than the number of excited neurons at $t_0$. In both cases, it is impossible to guess the original number of excited neurons: the information is lost. In the case of critical regime, the number of active neurons stays, in average, constant, so the information is transmitted through the network. Therefore, a graph of the fidelity of transmission wrt the probability $p$ would have a peak for $p = 1/N$.

A similarly simple model, where each neuron is connected only to a limited number of neighbouring neurons (short distance connections), helps illustrate other interesting features of the critical state. Suppose that, also in this model, $p$ can be tuned to induce a subcritical, supercritical or critical regime. In the subcritical regime the original excitation dies rapidly, thus propagating only on short spatial and temporal scales: few neighbouring neurons are interested. In the supercritical regime a lot of neurons are excited and for a long time, but the original information is lost in a deafening noise. In the critical regime, instead, the original excitation propagates with a reasonable fidelity through the network for potentially long distances, much longer than the short connections of the individual neurons. Moreover, the excitation can persist for a long time. The ability of the network to transmit information on large spatial and temporal scales is an emergent phenomenon, that manifests only in the critical state. Its importance resides in the fact that it can potentially explain how the brain maintain a (short term) memory of a stimulus and how it communicate such stimulus to vastly distant regions.

The real scenario is clearly much more complex; however, as clearly surveyed in [187], many theoretical and experimental studies have shown that information functions such as dynamic range, transmission fidelity, information capacity and information storage may be maximised in critical neural networks. The literature on the topic is vast, we refer the reader to the surveys [18, 187] for a detailed picture of the results on information functions maximisation. A more general analysis of the literature on criticality in networks of neurons is performed in the survey [111] and in the more

recent book [20]. Nonetheless, we would like to cite here a few papers concerning experimental evidences of the critical regime (or at least the presence of some of its hallmarks) in cell cultures ([21, 22, 194]) animals ([167, 105]) and humans ([119, 148]). The majority of the cited literature assumes the existence of a homeostatic mechanism ensuring the self-organised nature of the critical state. Explicit biologically plausible mechanisms for SOC and SOqC have been proposed in the literature, see for instance the pioneering work [132] and the survey [118].

#### 7.3.5.2 *Criticisms to the criticality hypothesis*

Since its formulation, the criticality hypothesis has been hotly debated and also criticised. Arguments and counterarguments have grown over the years in a continuous exchange among scientists. Many non trivial aspects have been clarified and more rigorous tools and tests developed. The debate, however, is far from being settled yet. In illustrating the main criticisms and potential replies, we will follow the papers [23, 19].

*Avalanches and power laws are artefacts.* One of the first criticisms was related to the observation of avalanches and power laws in empirical data. Concerns were raised on the methods used to detect the power laws, on the quality of the measurements and even on the very definition of avalanches (see for instance [195]). The definition of an avalanche, indeed, relies on the choice of a threshold and the observed statistics, like the power laws and their exponents, could change or even disappear depending on the threshold. This ambiguity is not entirely solved yet and great caution has to be observed. In general, any conclusion based on the observation of avalanches should be considered reliable only if it persists unchanged across a not so narrow interval of threshold values. The discovery of the limits of the initial analyses and their potentially spurious results has led to the adoption of more rigorous statistical tests for the detection of power laws, such as, for instance, the one proposed in [56]. Moreover, better data are now available thanks to recent technological advances like the increase in the number of electrodes used to record neurons' activity.

*Power laws do not imply criticality.* This also is one of the first criticisms moved by skeptics. It is known, since some time now, that power law distributions can be produced by mechanisms such as successive fractionation or combination of exponentials ([178, 36]) not entailing any criticality. In [195] it was also shown how thresholding a specific random process can produce power law distributions. These objections had the effect of refining the analysis of supposedly critical systems by requiring the simultaneous presence of multiple hallmarks: relations among critical exponents [32, 185], scaling collapse [185, 23], critical slowing-down [184, 111, 196], long-range temporal correlations [107]. Another merit of this criticism has been to bring into focus a specific property of critical systems: their tunability. Both models and physical systems must have a (control) parameter that can be varied such that avalanches and related signatures can be produced or destroyed at will. Some level of tunability has been practically verified *in vivo* and *in vitro* experiments by using pharmacological agents capable of disrupting synaptic transmission (see again [19] for a brief review).

*Non-critical systems exhibiting signatures of criticality.* There are physiologically meaningful systems that can be tuned far from criticality and still showing many of the characteristic signatures of criticality (not just power laws). An example was proposed by Benayoun and co-workers in [24]. There, a stochastic version of the celebrated Wilson-Cowan model was tuned such that excitation and inhibition are closely balanced. The model was tested with many different connectivity configurations always generating avalanches with power laws distributions having desired exponents. The balanced structure of the network was shown to be equivalent to an underlying functional feedforward connectivity, however, the equivalent deterministic model was not able to produce any avalanche. So the avalanches are both the result of the balanced synaptic weights and of the stochastic nature of the individual neurons. A limit of [24] is that it analyses only few of the indicators of criticality, hence the model could fail some of the other, more stringent, tests for *bona fide* criticality. Recently, in [66] the same model, but with a slightly different tuning, has been shown to actually be critical! In this configuration many of the characteristic signatures of criticality (i.e. exponent relation, scaling collapse, long-range temporal correlations) distinctly appear.

Another system was proposed by Touboul and Destexhe in [196]. They adopted a classical Brunel model, which is known to be able to display many of the regimes commonly found in experiments with cortical tissues. They considered an external random input whose intensity is equal to the drive from the excitatory neurons. The results was that, in a regime far from any criticality, the system exhibits avalanches with power laws distributions, good exponents and even scaling collapse. Only the exponent relation is not satisfied, however the values of exponents are all consistent with *in vitro* data. They also showed that the maximal information capacity is maximised in the non critical regime and stay maximal even for a regime where no avalanches were present. Therefore showing that some information functions are optimised even in non critical systems and in the absence of avalanches.

This is one of the strongest objection to the criticality hypothesis, to which Beggs responded in [19]. According to the author, the external input is too strong and when it is zeroed and the source of excitation is internal (randomly chosen neurons spike), the system is actually critical with the relative strength of inhibition wrt excitation as the control parameter. This configuration produces a lot of characteristic indicators of criticality. Moreover, the author pointed out that the information capacity considered in [196] may be not a relevant quantity. It is large when there is a large variety of responses to external stimuli, but it is unable to capture the correlation between the stimulus and the response. In other words, it can be high because of the high variability of responses to the *same* stimulus, implying that the reconstruction of the stimulus by means of the measurement of the response is impossible. In order to actually measure the information transmitted through the network, Beggs proposes to use the mutual information. This quantity, contrary to the information capacity employed in [196], manifests a clear peak in correspondence of the critical value of the Brunel model analysed by Beggs.

Another strong objection was put forward again in [196]. The authors showed

that a large system of weakly correlated (almost unconnected) units driven by special stochastic processes with similar statistics, can produce almost all the aforementioned signatures of criticality. The main reply to this criticism, given again in [19], is that all the signatures are already present in the external process driving the system and are only minimally influenced by the system itself. This objection calls for a sharper definition of criticality in neuronal networks. According to Beggs, in real cortical networks the criticality must be an emergent phenomenon. Its signatures must emerge as a result of a collective interaction of the neurons making up the network, they can not be already present in the external input. A network of (almost) disconnected neurons do not process any information and is not a adequate model of real neuronal tissues. To support his assertions, the author recalls that in many experimental studies where the synaptic transmission (hence the functional connection among neurons) is disrupted via drugs, the signatures of criticality disappear. Connection and interaction among neurons is thus a fundamental requirement. Beggs also notes that the unconnected model presented in [196], as well as similar ones, fails to show any long-range temporal correlation, which, instead, is a signature present in real tissues.

It is apparent from this section that the debate is still very active and will surely continue in the forthcoming years. New tools or signatures will probably be proposed to settle the question whether criticality in biological neural networks is a real or just an apparent phenomenon.

### *7.3.6 A control theory point of view*

Much of the literature on criticality concerns the empirical observation of potential signatures or statistical analysis of a variety of models. We believe that the theory of dynamical systems and control has the potential to provide a different and valuable perspective to the ongoing debate. The SOC (or SOqC) mechanism is, indeed, a special type of feedback. The phase diagram of the order parameter wrt the control parameter and the very idea of a state at the edge between two distinct behaviours seem to clearly allude to the presence of a bifurcation. The tools of the theory of dynamical systems and control could help answering profound questions concerning the existence of limitations or obstructions to the emergence of the criticality phenomenon, or the type of information and the mechanism needed to support the criticality and its self-regulating behaviour. This is not a completely unknown point of view in the physics literature, but very few results appeal to concept directly related to the control theory. To the best of our knowledge, only two studies in our community have addressed a similar question [153, 73]. In the following sections we introduce a systems and control theory-oriented formulation of the SOC problem that constitutes the basis of our future research on this topic.

#### *7.3.6.1 The SOC as a bifurcation tuning problem*

Inspired by the pioneering paper [153], we consider an ordinary differential equation depending on a parameter $\mu$ with unknown bifurcation value $\mu^*$. More precisely, let

us consider the following dynamic system

$$\dot{x} = f(x, \mu, \mu^*) \tag{7.3.1}$$

with $x, \mu \in \mathbb{R}$ and $\mu^* \in \mathbb{R}$, a fixed but unknown parameter (called *critical parameter* or *critical value*) and $f : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ a continuously differentiable function in $x$ and Lipschitz in $\mu$ and $\mu^*$.

**Definition 7.1.** We say that system (7.3.1) undergoes a critical transition through a point $(x^*, \mu^*)$ if the latter is a bifurcation point for the former. More precisely, we mean that $f(x^*, \mu^*, \mu^*) = 0$ and one of the following two things happens:

- $\partial_x f(x^*, \mu^*, \mu^*) = 0$ and for every $\varepsilon > 0$ small enough $\partial_x f(x^*, \mu^* - \varepsilon, \mu^*)\partial_x f(x^*, \mu^* + \varepsilon, \mu^*) < 0$;
- there exists an open interval $\mathcal{I} \subseteq \mathbb{R}$ with $\mu^* \in \mathcal{I}$ such that, for every $\mu \in \mathcal{I}$ there exists a point $\bar{x}(\mu)$, with $\bar{x}(\mu) \neq x^*$ for all $\mu \neq \mu^*$, such that $f(\bar{x}(\mu), \mu, \mu^*) = 0$ and $\lim_{\mu \to \mu^*} \bar{x}(\mu) = x^*$.

The first item in the previous definition corresponds to the case where the eigenvalue associated to the linearised system about $(x^*, \mu^*)$ crosses the imaginary axis for small variation around $\mu^*$. The second item represents the case where there exists at least a second equilibrium point $\bar{x}(\mu)$, function of $\mu$, which collapses to $x^*$ for $\mu$ tending to its critical value. The latter case describe a bifurcation where the number of equilibria changes.

Concerning system (7.3.1), we assume the following.

**Assumption 7.1.** *We do the following assumptions:*

- *there exists $x^* \in \mathbb{R}$ such that $f(x^*, \mu, \mu^*) = 0$ for any $\mu \in \mathbb{R}$;*
- *the system (7.3.1) undergoes a critical transition through the point $(x^*, \mu^*)$.*

*Remark* 7.1. The first point of the Assumption 7.1 is verified by all the relevant models of critical systems studied in literature. In the case of a second order transition, indeed, at least two equilibria collapse (see second item in Definition 7.1). One of the two, i.e. $x^*$, is in general fixed and independent of the value of $\mu$.

The problem we want to address here is to design a control law for the parameter $\mu$ that makes the closed-loop system self-regulate towards criticality, although $\mu^*$ is unknown. More precisely, we aim the following.

**Problem 7.1.** Given a system of the form (7.3.1) satisfying Assumption 7.1, design a dynamic feedback for $\mu$ independent of $\mu^*$ ensuring that $x \to x^*$ if and only if $\mu \to \mu^*$.

To tackle this problem we plan to extend the dynamics (7.3.1) as follows:

$$\begin{aligned} \dot{x} &= f(x, \mu, \mu^*) \\ \dot{z} &= g(x, z, \mu) \\ \dot{\mu} &= h(x, z, \mu) \end{aligned} \tag{7.3.2}$$

where $z \in \mathbb{R}^p$ and $g, h$ functions defined accordingly. We stress that the critical parameter $\mu^*$ is unknown, hence both $g$ and $h$ are not aware of its value. The extension allows a more general dynamic feedback that can also encompass controllers based on observers or on models.

### *7.3.6.2 A special system*

In [153] the following system is considered:

$$\dot{x} = (\mu - \mu^*)x \tag{7.3.3}$$

with $x \geq 0$, $\mu \in \mathbb{R}$ and $\mu^* \in \mathbb{R}$ fixed but unknown. The system is in the form (7.3.1) and satisfies Assumption 7.1 with $x^* = 0$ and any $\mu^*$. In particular, one can verify that the system (7.3.3) undergoes a critical transition like that in the first item of Definition 7.1.

In [153] no dynamic extension is employed and a feedback dynamics $\dot{\mu} = h(x, \mu)$ for the variable $\mu$ is designed to ensure the exponential stability of an isolated equilibrium point $(\bar{x}, \mu^*)$ of the complete system with $\bar{x} \neq x^* = 0$. We stress that the proposed solution does not solve Problem 7.1 since it does not work for $\bar{x} = 0$. The system (7.3.3) is quite special as it satisfies $f(\bar{x}, \mu^*, \mu^*) = 0$ for *any* value of $\bar{x}$, not only $x^*$. Therefore, any point $(\bar{x}, \mu^*)$ with $\bar{x} \geq 0$ can be considered critical, not only the point $(0, \mu^*)$. This is the reason why in [153] a dynamics for $\mu$ can be designed to drive the system to the point $(\bar{x}(\mu^*), \mu^*)$ where $\bar{x}(\mu^*)$ is a function of $\mu^*$.

In [73], a wider class of systems in Lurie form is considered and an adaptive (observer-based) control law is developed. However, also in this paper, while $\mu$ is driven towards its critical value $\mu^*$, the state $x$ converges to an equilibrium which does not correspond necessarily to $x^*$.

Problem 7.1 is more difficult to solve, but we stick with it because it is the one appearing in physiological meaningful systems.

### *7.3.6.3 A more general system*

We would like to study the following scalar nonlinear system:

$$\dot{x} = (\mu - \mu^*)x - q(\mu)x^2, \tag{7.3.4}$$

with $x \geq 0$, $\mu \in \mathbb{R}$, $\mu^* \in \mathbb{R}$ fixed but unknown and $q$ is a globally Lipschitz function satisfying $q(\mu) > 0$ for all $\mu \neq 0$ or $q \equiv 0$. For $q \equiv 0$ we obtain the system (7.3.3). For $q > 0$ the system (7.3.4) has two equilibria: $x = 0$ (the quiescent state) and $x = \frac{\mu - \mu^*}{q(\mu)}$ (the

active state). For $\mu \to \mu^*$ the two equilibria collapse, therefore, system (7.3.4) satisfies Assumption 7.1 w.r.t. the point $x^* = 0$.

Such model encompasses many (simple) systems relevant to statistical physics and computational neuroscience. As already noted, for $q \equiv 0$, this model turns out to be the first-order linear system considered in [153]. For a constant, strictly positive $q$, we recover the simplest continuous-time model for a mean-field approximation of a sandpile: see for instance [42]. For $q(\mu) = \mu$, one obtains the neuronal mean-field models presented in [111, 96, 149].

More precisely, in [111] (see also Section 7.3.5.1) a macroscopic model of a network of neurons is presented, where the variable $A(t)$ describes the mean proportion of active neurons at time $t$:

$$\dot{A} = \frac{1}{\tau}\left((p\kappa - 1)A - p\kappa A^2\right). \tag{7.3.5}$$

The constant $p$ is the probability that an active presynaptic node activates a postsynaptic node in a time interval of length $\tau$. The constant $\kappa$ represents the average number of outgoing links for each node (network connectivity). One can easily see that (7.3.5) results in (7.3.4) for $\mu := p\kappa/\tau$ and $\mu^* := 1/\tau$. Concerning the models presented in [96, 149], we limit ourselves to the first equation, ignoring the mean-field dynamics of the neuronal gain $\Gamma$ and of the synaptic weight $W$:

$$\rho(t+1) = (1 - \rho(t))\Gamma(W\rho(t) + h). \tag{7.3.6}$$

Here $\rho$ is the firing rate of a network of neurons with a sigmoidal activation function having neuronal gains $\Gamma_i$ and synaptic thresholds $\theta_i$.[3] The combined input $h$ is equal to $h = I - \theta$, with $I$ the average external input current. The system (7.3.6) undergoes a second-order phase transition only for $h = 0$, so we limit our analysis to this value. As customary in the neuroscience literature, the discrete-time model (7.3.6) has been derived via the Euler approximation with unit time interval: $\dot{\rho}(t) \approx \rho(t+1) - \rho(t)$. By reverting the approximation and rearranging the terms, one can find

$$\dot{\rho} = (\Gamma W - 1)\rho - \Gamma W \rho^2,$$

which is again the model (7.3.4) for $\mu := \Gamma W$ and $\mu^* := 1$.

## 7.3.7 Research directions

In view of the complexity of the debate on the critical brain hypothesis, the range of problems that need a solution is quite large. Here, we briefly present the research directions we aim to follow in the form of questions needing an answer. They are organised in chronological order from the closest to the furthest in time. Some of them are purely theoretical in nature, other will require a tight interaction with neuroscientists and physicists. To this aim, we have already started a collaboration with Alain Destexhe

[3] $i$ is the index of the neuron while $\Gamma = \langle \Gamma_i \rangle$, $\theta = \langle \theta_i \rangle$ are the average gain and threshold respectively.

(DR CNRS at NeuroPSI), which is at the origin of the tougher criticisms to the criticality hypothesis.

*Structural obstructions.* The first questions that arise concern obstacles and limitations to the solution of Problem 7.1. A preliminary analysis seems to suggest the existence of some general obstructions on the nature of the critical points defining the bifurcation and on the structure of the closed loop system. The determination of potential obstructions is of great relevance as it can guide the (experimental) research for the actual physiological mechanisms that could support the criticality.

*Observer or not.* Starting with the simple, but still physiologically meaningful, system (7.3.4), we want to understand what kind of structure the dynamic feedback $(g, h)$ in (7.3.2) can take. We are confident that a solution can be found in the form of an observer. The unknown parameter can be reconstructed by the measurements of the variables $x$ and $\mu$ and the estimated critical value used to build a feedback law. However, is this structure mandatory? Is it possible to tune the bifurcation without an explicit observer or without an extended dynamics at all? Indeed, while an explicit observer is of interest for the theoretical problem and, maybe, for the statistical physics, it is not clear how plausible it is from a neurophysiological point of view.

*Richer mean-field models.* The system (7.3.4) is very simple and scalar. We want to extend our study to other mean-field models studied in literature. In particular we will consider models that include the simultaneous dynamics of other relevant physiological quantities like the neuronal gain $\Gamma$ and the synaptic weight $W$ (see for instance [96, 149]).

*Information and robustness.* In the system (7.3.2) we are assuming that $(g, h)$ have continuous access to the variables $x$ and $\mu$. We don't know if this is the case in real systems. Questions that deserve to be considered concern what kind of information is strictly necessary to solve Problem 7.1: Is the knowledge of only one of the variables $x$ or $\mu$ enough? Is a partial knowledge of those variables, like an average or a pick-to-pick measure on a period, sufficient? Moreover, is this information affected by delays or by some kind of noise? Connected to this question is also the robustness requirement. To be physiological plausible, the dynamic feedback we develop must exhibit a certain degree of robustness to perturbations: a fragile mechanism would have never been selected by nature.

*Reconcile macro with micro.* One of the most ambitious directions of this line of research is to reconcile the two points of view on criticality. From one side, the empirical definition based on the statistical signatures associated to avalanches coming from statistical physics and neuroscience, on the other side, the definition based on bifurcation and structural stability coming from systems theory. This is a complex point requiring many steps.

First of all, it is necessary to establish which mean-field model is capable of generating avalanches.

Second, the critical behaviour is always measured when the feedback mechanism is deactivated, but in cortical tissues it is supposed to be active, so: What temporal scale separation is necessary between the system and the feedback? What kind of

approximate behaviour is actually visible when the feedback is active? Is it similar to the aforementioned quasi-criticality?

Third, real cortical neural networks are large, extended systems, while the mean-field approximation is a macroscopic description at the level of neuronal population. The order and the control parameters are global (emergent) quantities. If the brain is critical and a mechanism is responsible for its self-regulation, then such mechanism must act on a local scale. In other words, a neuron has no access to the global order parameter but only to the state of a limited number of neurons connected to it. The question is: how can we adapt a mechanism developed for a mean-field model to work on a microscopic scale on the basis of only local measurements?

Four, if such adaptation succeed and the time scales are correctly tuned, the large system should generate avalanches. How many signatures of criticality can in this case be measured in simulation? Are they consistent with those experimentally measured in real networks?

## 7.4 MECHANICAL VENTILATION

Mechanical ventilation is used to assist patients with respiratory failures. Its goal is to ensure adequate tissues oxygenation, carbon dioxide elimination and muscular effort reduction. There exist many ventilation modes distinguished by the way the respiration cycle is managed and which variable (pressure or volume) is controlled. The choice of the appropriate ventilation mode is left to trained clinicians, according to the patient condition and pathology ([112]). Despite its unquestionable benefits, however, mechanical ventilation is associated to the risk of developing lung ([40, 46]) and diaphragm ([97]) injuries due to prolonged ventilation duration ([97]), especially when ventilators are not optimally tuned.

When a patient is not severely injured or deeply sedated, he/she retains a respiratory drive. In this case a ventilation mode capable of following and amplifying his/her breath pattern is recommended. The aim is to avoid discomfort, patient fighting against the ventilator and even lung injuries. Some ventilation modes, as a consequence, are meant to assist more than to replace the spontaneous breath. Among these modes we focus on the Pressure Support Ventilation (PSV) by reason of its wide adoption in invasive as well as non invasive ventilation units. A crucial issue for any assistive mode is to ensure, as much as possible, the synchronisation between the patient effort and the ventilator support. Asynchronies are indeed correlated with longer permanence in intensive care units and potential pulmonary injuries.

The main goal of this line of research is to improve the PSV mode synchrony. We want to provide a finer and automatic tuning of a critical ventilator parameter, whose manual setting is usually complex and requires a significant training of the clinicians.

In the following subsections we present a brief overview of mechanical ventilation in general and of PSV in particular. We provide a definition of patient/ventilator asynchrony and we focus on how the most common asynchronies can be avoided. In particular, we discuss the general ideas behind the optimal tuning of the expiratory

trigger, which is the crucial parameter in PSV to circumvent asynchronies. We end with an overview of the future research directions we want to pursue in the field of mechanical ventilation control.

The present line of research is carried out in collaboration with William Pasillas Lepine, DR CNRS at L2S, and the anaesthesiologists of the AP-HP/IMRB (Institut Mondor de Recherche Biomédicale) Guillaume Carteux and Samuel Tuffet. It is also partially supported by the AirPad project funded by the SATT Paris-Saclay[4] in the framework of the Poc'Up program. One of the objective of the AirPad project is to finalise the patent application for the synchronisation strategy we propose. For this reason, we can not give here any detail on such strategy.

## 7.4.1 Background on mechanical ventilation

### 7.4.1.1 Preliminaries on ventilation modes

A respiration cycle is made up of an *inspiratory phase*, characterised by positive flow, and an *expiratory phase*, characterised by negative flow. Hence, the total cycle time $T_{\text{tot}}$, also known as ventilation period, is the sum of the inspiratory time $T_{\text{I}}$ and expiratory time $T_E$. The respiratory frequency, usually measured in breaths per minute, is $f = 60/T_{\text{tot}}$ bpm. The total amount of inhaled air is measured through the *tidal volume* $V_{\text{T}}$, which is the integral of the positive flow over $T_{\text{I}}$. For a normal adult it is approximatively $7\ {}^{\text{mL}}/_{\text{kg}}$ times the ideal body weight.

Mechanical ventilation works by imposing a desired, predetermined reference for the pressure or the volume (control variable). Some ventilation modes allow to automatically switch from one variable to the other, but no mode allows to control both at the same time since they are related variables in the respiratory mechanical equations (see later equation (7.4.1)). A ventilation mode can assist the patient in his/her effort by sustaining the *spontaneous breath* or it can replace altogether the patient by imposing a *mandatory breath* (for instance in case of severely injured or deeply sedated patients). Ventilation modes are also classified in terms of the way the inspiratory phase is initiated *(triggering event)* and terminated *(cycling event)*. Those events can be determined by the crossing of some threshold on flow, volume, pressure or time and can be dependent or independent of the patient effort. For a taxonomy of the different ventilation modes see e.g., [52].

### 7.4.1.2 Pressure Support Ventilation

Pressure Support Ventilation (PSV) is defined as a patient-triggered, pressure-limited (or targeted), flow-cycled ventilation mode (see e.g. [41]). In order to understand this definition, we make reference to the standard model for respiratory mechanics:

$$P_{\text{ao}}(t) = RQ(t) + EV(t) + P_{\text{mus}}(t) + P_0 \tag{7.4.1}$$

[4] SATT Paris-Saclay is the technology transfer accelerator office of the Paris-Saclay cluster.

where $P_{ao}$ is the *airways opening pressure*, that is the pressure at the entrance of the patient airways, $P_{mus}$ is the *muscular pressure* exerted by the patient (namely the *patient effort*) and $P_0$ is the *offset pressure* balancing the pressure at the airways opening at $t = 0$. $V$ is the *volume* of the patient's lungs, $Q = \dot{V}$ is the *flow*, while $E$ and $R$ are lumped parameters accounting for the *elastance* of the lungs and chest and *resistance* of the airways and chest, respectively. The parameters $E$ and $R$ are unknown and usually set by clinicians on the basis of their experience and commonly accepted practice. Their estimation is a relevant problem as it usually requires the measurement of the patient effort. The latter is available only in case an oesophageal catheter is used, which is clearly an invasive method. A novel approach for a noninvasive estimation of respiratory effort is presented and compared with reference methods in [164].

At a time $t_{str}$ the patient starts his/her effort (spontaneous breath) by increasing the flow. The instant $t_{on} > t_{str}$ when the flow exceeds a threshold $Q_{on}$, usually fixed by the clinicians, signal the start of the inspiratory phase. Thus, the triggering event $Q(t_{on}) \geq Q_{on}$ is patient-initiated. Similarly, the end of the inspiratory phase, hence the start of the expiratory phase, is triggered at an instant $t_{off} > t_{on}$ by the event $Q(t_{off}) \leq Q_{off}$, namely when the flow goes below a given threshold. Whence the denotation ''flow-cycled'' in the PSV definition. Differently from $Q_{on}$, $Q_{off}$ is not fixed but is defined as a percentage of the peak value $Q_{max}$ during inspiration: $Q_{off} = r_{off} Q_{max}$. The ratio $r_{off}$ is usually dubbed *expiratory* (or cycling) *trigger* and can be adjusted by clinicians in all modern ventilators. The aforementioned quantities are represented in Figure 7.4.1 through a simulated respiration cycle.

The parameter $r_{off}$ is critical in determining the synchronisation between patient and ventilator. Indeed, we consider the pressure support synchronised with the patient effort if the cut–off instant $t_{off}$ coincides with the instant $t_{end}$ when the patient ends his/her effort (see Figure 7.4.2-a). There exist many types of asynchronies (see for instance [156, 41]), however we consider two of them: *short cycles* when $t_{end} > t_{off}$ (see Figure 7.4.2-b) and *prolonged insufflations* when $t_{end} < t_{off}$ (see Figure 7.4.3-a). Particularly relevant for PSV, as the most frequently occurring, are the *inefficient efforts* (see Figure 7.4.3-b). These are extreme case of prolonged insufflations where, during the expiratory phase, the patient effort is not able to trigger a new inspiration phase. In this case the flow stays below the threshold $Q_{on}$ for the entire duration of the patient effort.

During inspiration, the PSV mode sustains the patient effort by guaranteeing a level of pressure defined according to the amount of required support and to the desired tidal volume. The imposed level of pressure explains the ''pressure-limited'' denotation in the definition of PSV. A piecewise-constant *pressure reference* $P_{ref}$ is provided by the ventilator. $P_{ref}$ is equal to the *positive end-expiratory pressure* (PEEP) during expiration. During the inspiration, the difference between $P_{ref}$ and the PEEP determines the level of pressure support and is adjusted by clinicians. The PEEP (also known as *extrinsic PEEP*[5]) is used to keep the airways open and to improve gas distribution. The pressure

[5]It has to be distinguished from the *intrinsic PEEP*, which is due to incomplete airways emptying during expiration. The intrinsic PEEP is a negative phenomenon that can lead to overinflation and a

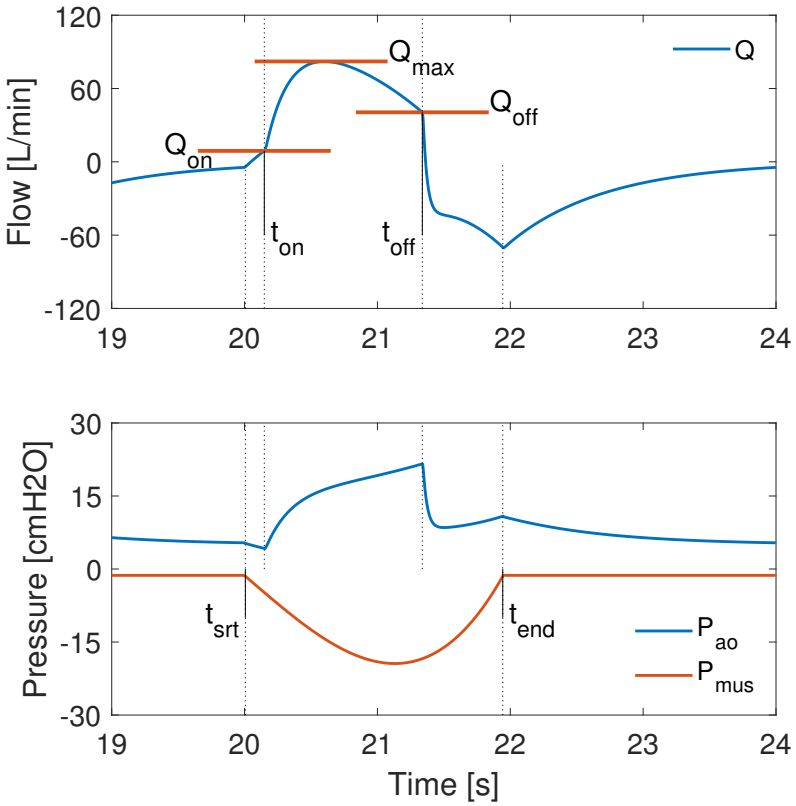


FIGURE 7.4.1: Triggering thresholds and times for inspiratory and expiratory phases in a respiration cycle under PSV.

reference is not directly applied to the patient, a smoothed version of $P_{\text{ref}}$, namely the *filtered reference* $P_{\text{flt}}$, is used instead to define a flow reference $Q_{\text{ref}}$. The flow is then assumed to follow such reference, which is proportional to $P_{\text{flt}}$ (see [38])

$$Q(t) = Q_{\text{ref}}(t) = K_s(P_{\text{flt}}(t) - P_{\text{ao}}(t)) \tag{7.4.2}$$

with $K_s > 0$. $P_{\text{flt}}$ is designed to converge to $P_{\text{ref}}$ and sometimes it is just taken as a piecewise linear version of $P_{\text{ref}}$ with adjustable slopes. In any case the convergence speed is tuneable through a parameter (settling time) of the ventilator interface.

A differential equation for the volume is obtained by replacing (7.4.1) in (7.4.2) and recalling that $Q = \dot{V}$:

$$\dot{V}(t) = \alpha V(t) + \beta(P_{\text{flt}}(t) - P_0 - P_{\text{mus}}(t)) \tag{7.4.3}$$

where

$$\alpha := -\frac{E}{R + \frac{1}{K_s}}, \quad \beta := \frac{1}{R + \frac{1}{K_s}}. \tag{7.4.4}$$

The constant $-1/\alpha$ identifies the *Expiratory-Time Constant (ETC)* (see e.g. [104]).

difficulty for the patient to trigger the ventilator insufflation.

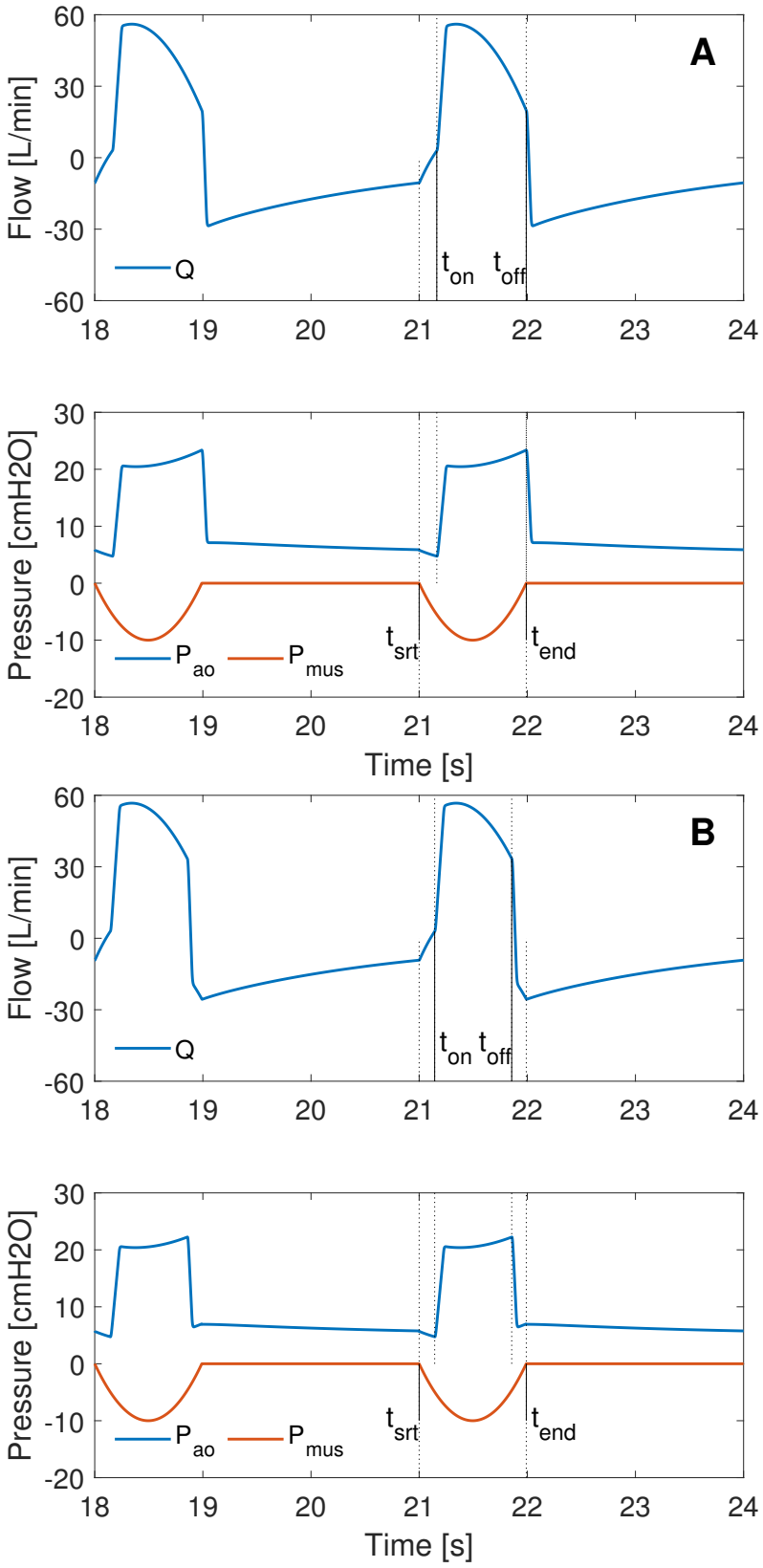


FIGURE 7.4.2: Impact of the expiratory trigger in PSV: (a) synchronisation for $r_{off}$ = 36%; (b) short cycle for $r_{off}$ = 60%.

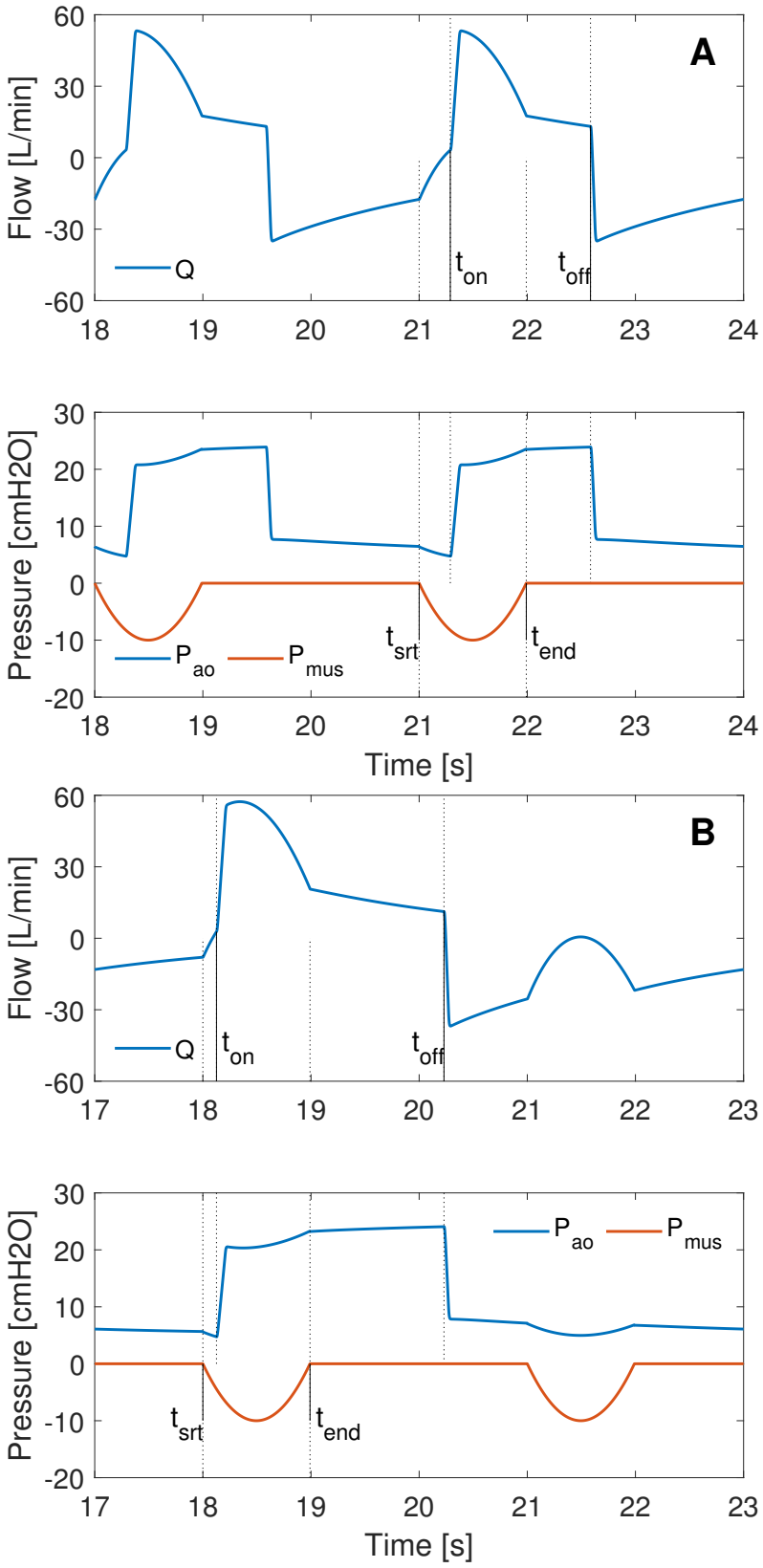


FIGURE 7.4.3: Impact of the expiratory trigger in PSV: (a) prolonged insufflation for $r_{off}$ = 25%; (b) inefficient effort for $r_{off}$ = 20%.

### 7.4.2 The synchronisation problem

By virtue of its ability to assist a patient in his/her spontaneous respiratory effort, PSV is often regarded as an ideal ventilation mode capable of reducing respiratory distress while keeping a good level of synchrony with the patient ([41]). It is usually considered by patients more comfortable than other ventilation modes and it has been shown to reduce the weaning times. Moreover, it has the potential to limit the occurrence of ventilator-induced pathologies. For all these reasons PSV is still the most widely used method of assisted ventilation in the world ([41]). Despite these advantages, PSV is far from perfect as it can still present asynchronies like those discussed in Section 7.4.1.2.

In PSV the key parameter to avoid asynchronies is the expiratory trigger $r_{\text{off}}$. Today, the common practice is to manually adjust $r_{\text{off}}$. To be optimal, this adjustment requires considerable expertise and regular reassessment (difficult to reconcile with clinical activity). An automatic adjustment of the expiratory trigger has the potential to improve the patient-ventilator synchrony, thus facilitating the clinician's work and reducing the risk of setting errors.

The state of the art approach ([213]) to optimise the choice of $r_{\text{off}}$ assumes the simplifying hypothesis $t_0 := t_{\text{on}} - t_{\text{str}} = 0$. That is, the triggering of insufflation is considered to take place instantly, at the start of the patient effort ($t_{\text{on}} = t_{\text{str}}$). With this simplification, it is relatively easy to calculate the value of $r_{\text{off}}$ for which $t_{\text{off}} = t_{\text{end}}$ by integrating the differential equation (7.4.3). Even if this approach can provide reasonable results for low ETC values, it proves remarkably imprecise for high values of ETC (case where synchronisation between patient and ventilator is harder to achieve). Because of this difficulty, instead of looking for the exact value of $r_{\text{off}}$ ensuring synchronisation, more recent works (e.g. [183, 159]) provide general considerations on the $r_{\text{off}}$ ranges to use. For example, they propose a $r_{\text{off}}$ of 15% for ETC values less than 0.4 s, of 25% for ETCs between 0.4 and 0.8 s and of 45% for ETCs greater than 0.8 s. These indications are quite relevant and useful for clinicians. However, they do not take into account the details of the patient's specific conditions: is he/she capable of exerting a significant muscular effort? What level of pressure support is used? They also neglect the settings of the other ventilator parameters: What is the slope (or settling time) in the filtered pressure reference? What is the value of $Q_{\text{on}}$?

The aim of our research in this topic is to propose an automatic adjustment of $r_{\text{off}}$ that takes into account all these considerations.

From a mathematical point of view, the calculation of the $r_{\text{off}}$ corresponding to synchronisation can be done using the Poincaré first return map ([199]). For a fixed $r_{\text{off}}$, we can integrate (7.4.3) and compute, for any value of $t_0$ of a respiration cycle, the corresponding value of $t_0$ of the next cycle. As a result, for each value of $r_{\text{off}}$ in a given interval, we can compute the corresponding $t_0$ and, from the latter, the difference between $t_{\text{off}}$ and $t_{\text{end}}$. Finally, we can choose the value of $r_{\text{off}}$ that minimises such difference. The problem with this approach is that it is computationally impractical. The method we have developed, on the contrary, allows for an almost instantaneous computation. This requirement is strictly necessary if a concrete implementation of the algorithm on a mechanical ventilator is aimed at. The idea is to construct what we

call the *synchronisation map*, which, unlike the Poincaré map, does not need to fix a specific value of $r_{\text{off}}$. Like for the Poincaré map, the fixed points of the synchronisation map determine the admissible values of $t_0$. From the value of $t_0$ ensuring the synchronisation, the computation of $r_{\text{off}}$ is then immediate. Unfortunately, we can not provide a mathematical description of our method as it is presently subject to a patent pending declaration. However, we can show in Figure 7.4.4 a couple of diagrams, produced by our method, for two possible patients with different resistance $R$ and respiratory frequency $f$. These diagrams show the value of $r_{\text{off}}$ ensuring synchronisation for different values of patient effort (peak muscular pressure) and of pressure support.

### 7.4.3 Research directions

The first goal of this line of research is to produce an effective algorithm that can help or replace the clinicians in setting the optimal expiratory trigger. A completely automatic setting algorithm could be included in any mechanical ventilator, thus relieving the clinicians of the burden of continuously readjusting $r_{\text{off}}$. An non automatic help can also be provided for any ventilator not directly supporting the algorithm. In this case an application (e.g. on a smart phone) can be queried by clinicians whenever they need the right value of $r_{\text{off}}$. To achieve this objective, many steps are required.

*Test campaigns.* As for now, a prototypical version of the algorithm exists in Matlab language. It is therefore not suitable for a realtime implementation on a ventilator in the present form. However, it can, and will, be used for testing. The first test campaign will be performed on at least two different models of ventilators by using an artificial lung type ASL-5000. The ASL allows the simulation of many different patient profiles, thus providing a wealth of useful data to validate or to improve our algorithm. This first campaign is strictly necessary before launching much more demanding, and potentially risky, clinical trials on human patients. Both campaigns will be performed in collaboration with our medical partners of the Institut Henri Mondor.

*Low level flow control.* In order to induce a ventilators' manufacturer to include our algorithm on their ventilators' models, we must develop a convincing prototype. Unfortunately, commercial ventilators are closed systems that do not allow any change to their proprietary algorithms. For this reason we designed a test platform, based on an open source ventilator, which will allow a real time implementation of the algorithm as well as the future development of other, more sophisticated ventilation modes. The fundamental components of our platform are a blower, injecting air in the airways, and two pinch valves (one for the inspiratory circuit and one for the expiratory circuit) actuated by two electric motors. The first problem to consider here is the improvement of flow regulation in the valves, to achieve a similar or better performance to that of commercial ventilators. The model having the angle of the motors as input and the flow as output is highly non linear, affected by saturation and delay. Moreover, the control low must be implemented in discrete time, so the system must be correctly sampled. Many existing controllers are linear (see for instance [113]), while we look for a nonlinear controller for its potentially superior performance. The basic ingredients

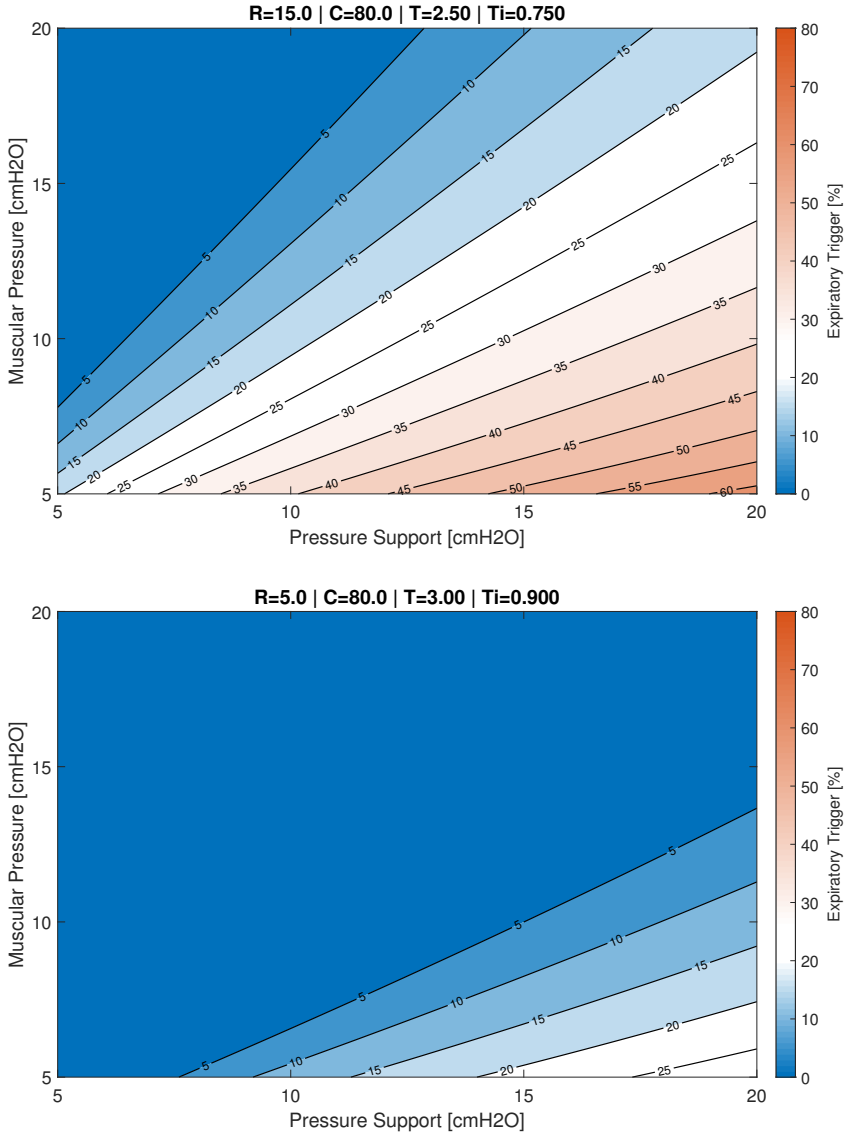


FIGURE 7.4.4: Diagrams of the optimal $r_{\text{off}}$: Up: patient with a resistance of 15 cmH2O/L/s an elastance of 1/80 cmH20 · L and a respiration frequency of 24 bpm; Down: patient with a resistance of 5 cmH2O/L/s an elastance of 1/80 cmH20 · L and a respiration frequency of 20 bpm.

of the control law will be a feedback linearising controller and a Smith predictor to compensate the delay of the linearised system.

*ETC estimation.* Our synchronisation method requires the knowledge of the mechanical parameters $R$ and $E$ as well as of the peak of the patient effort. They can be provided by the clinicians on the basis of their experience, however they may significantly vary during the period a patient is ventilated. A fully automated algorithm would ideally be able to cope with the changes in the patient conditions and to readjust $r_{\text{off}}$ accordingly. This calls for a continuous estimation of the required parameters. To this aim, a technique like that presented in [164] can be used. However, such technique, as well as other available in literature, requires a precise knowledge of the ETC. We want to develop a robust estimation technique for the ETC constant. The technique will be based on two steps: the first steps aims to recognise and discard degenerate respiration cycles such as those including inefficient efforts; the second step realises the actual estimation of ETC on regular cycles. The exclusion of degenerate cycles is not problematic since the regular cycles are generally more frequent.

*Other ventilation modes.* While PSV mode is widely used and generally appreciated, there is clearly room for better ventilation modes, more capable of following the patient effort instead of imposing a respiration pattern. In this vein the Proportional Assist Ventilation (PAV) has already shown to be theoretically superior to PSV. In the words of his proposer [215], with PAV ''the ventilator simply amplifies patient instantaneous effort throughout inspiration while leaving the patient with complete control over all aspects of breathing pattern (tidal volume, inspiratory and expiratory durations, and flow patterns)''. Despite all these promises, PAV is presently not significantly widespread. One reason is that early clinical trials have proved disappointing not showing a significant benefit wrt PSV ([41]). It is important to stress that PAV is very reliant on precise knowledge of $R$ and $E$ ([98]), which, as already mentioned, is not easy to obtain. Moreover, those parameters change over time. Thus, the disappointing results could be related to an imprecise knowledge of $R$ and $E$. Another obstacle to a larger adoption seems to be the setting of the PAV: it is more time consuming than that of PSV in reason of its complexity ([41]). Besides, PAV can also generate a specific type of asynchronies.

While a better and continuous estimation of $E$ and $R$ has the potential to increase the acceptance of PAV, there is still the possibility to improve the very ventilation mode. The PAV as well as the other modes are based on linear equations for the respiratory mechanics and mainly linear controllers to follow the patient effort. In many critical respiratory pathologies some nonlinear phenomena characterising the pulmonary mechanics are no longer negligible. More accurate models and nonlinear controllers could lead to better ventilation modes. Once the low-level controls of our test platform will be in place, we plan to study such innovative modes.

*Extension of the synchronisation map.* At the very essence, the synchronisation problem we consider here arises from the interaction of two hybrid systems. Each system has its guard conditions, influences and is influenced by the dynamics of the other and ideally performs a periodical activity. One of the system (i.e. the patient

in our case) dictates the pattern of the periodic activity, while the other (i.e. the ventilator) must achieve some objectives (e.g. ensuring a desired tidal volume, reducing discomfort) while synchronising with the first.

This scheme is rather general and can appear in many physiological systems where periodic activities are widespread. The goal of this research direction is to extend and generalise the ideas behind the synchronisation application to such general class of systems. A first step in this exploration will be to focus on the human cardiovascular and respiratory systems. They are, indeed, an example of interacting systems performing cyclic activities that often need to synchronise. In some meditation exercises, the respiration cadence is even used, among other things, to control the hearth rate. The models usually found in literature are, however, too complex to obtain analytical results (see for instance [4]). Therefore, an initial effort will be devoted to distill a simpler model, namely with few state variables, still physiologically meaningful, but allowing the construction of the synchronisation application.

# BIBLIOGRAPHY


[1] G. Aazan, A. Girard, L. Greco, and P. Mason, ''Stability of shuffled switched linear systems: A joint spectral radius approach,'' *Automatica*, vol. 143, p. 110434, 2022.

[2] L. Abeni and G. Buttazzo, ''Integrating multimedia applications in hard real-time systems,'' in *Proc. IEEE Real-Time Sys. Symp.*, Dec. 1998, pp. 4–13.

[3] L. Abeni and G. C. Buttazzo, ''QoS guarantee using probabilistic deadlines,'' in *Proc. Euromicro Conf. on Real-Time Sys.*, York, England, UK, Jun. 1999, pp. 242–249.

[4] A. Albanese, L. Cheng, M. Ursino, and N. W. Chbat, ''An integrated mathematical model of the human cardiopulmonary system: model development,'' *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 310, no. 7, pp. H899–H921, Apr. 2016.

[5] A. Alessandri and P. Coletta, ''Switching observers for continuous-time and discrete-time linear systems,'' in *Proc. American Control Conf.*, vol. 3, 2001, pp. 2516–2521.

[6] A. Aminifar, E. Bini, P. Eles, and Z. Peng, ''Analysis and design of real-time servers for control applications,'' *IEEE Trans. on Computers*, vol. 65, no. 3, pp. 834–846, Mar. 2016.

[7] A. Aminifar, P. Tabuada, P. Eles, and Z. Peng, ''Self-triggered controllers and hard real-time guarantees,'' in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2016, pp. 636–641.

[8] A. Anta and P. Tabuada, ''On the minimum attention and anytime attention problems for nonlinear systems,'' in *Proc. IEEE Int. Conf. on Decision and Control*, 2010, pp. 3234–3239.

[9] A. B. Arehart and W. A. Wolovich, ''Bumpless switching controllers,'' in *Proc. IEEE Int. Conf. on Decision and Control*, vol. 2, Kobe, Japan, 1996, pp. 1654–1655.

[10] K. J. Åström and B. Wittenmark, *Computer Controlled Systems*. Prentice Hall Inc., November 1996.

[11] N. Athanasopoulos and M. Lazar, ''Stability analysis of switched linear systems defined by graphs,'' in *Proc. IEEE Int. Conf. on Decision and Control*, 2014, pp. 5451–5456.

[12] M. Babaali and M. Egerstedt, ''Pathwise observability and controllability are decidable,'' in *Proc. IEEE Int. Conf. on Decision and Control*, vol. 6, 2003, pp. 5771–5776.

[13] ——, ''Observability of switched linear systems,'' in *Int. Wor. on Hybrid Sys. Comp. and Contr.*, 2004, pp. 48–63.

[14] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[15] P. Bak, *How nature works: the science of self-organized criticality*. Copernicus New York, 1996.

[16] P. Bak, C. Tang, and K. Wiesenfeld, ''Self-organized criticality: an explanation of the 1/f noise,'' *Physical Review Letters*, vol. 59, no. 4, pp. 381–384, Jul. 1987.

[17] M. Baştuğ, M. Petreczky, R. Wisniewski, and J. Leth, ''Reachability and observability reduction for linear switched systems with constrained switching,'' *Automatica*, vol. 74, pp. 162–170, 2016.

[18] J. M. Beggs, ''The criticality hypothesis: how local cortical networks might optimize information processing,'' *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1864, pp. 329–343, Aug. 2008.

[19] ——, ''Addressing skepticism of the critical brain hypothesis,'' *Frontiers in Computational Neuroscience*, vol. 16, Sep. 2022.

[20] ——, *The cortex and the critical point: understanding the power of emergence*. MIT Press, 2022.

[21] J. M. Beggs and D. Plenz, ''Neuronal avalanches in neocortical circuits,'' *The Journal of Neuroscience*, vol. 23, no. 35, pp. 11 167–11 177, Dec. 2003.

[22] ——, ''Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures,'' *The Journal of Neuroscience*, vol. 24, no. 22, pp. 5216–5229, Jun. 2004.

[23] J. M. Beggs and N. Timme, ''Being critical of criticality in the brain,'' *Frontiers in Physiology*, vol. 3, 2012.

[24] M. Benayoun, J. D. Cowan, W. van Drongelen, and E. Wallace, ''Avalanches in a stochastic model of spiking neurons,'' *PLoS Computational Biology*, vol. 6, no. 7, p. e1000846, Jul. 2010.

[25] G. Bernat, A. Burns, and A. Llamosi, ''Weakly hard real-time systems,'' *IEEE Trans. on Computers*, vol. 50, no. 4, pp. 308–321, April 2001.

[26] G. Bernat, A. Colin, and S. M. Petters, ''WCET analysis of probabilistic hard real-time systems,'' in *Proc. IEEE Real-Time Sys. Symp.*, 2002, pp. 279–288.

[27] ——, ''pWCET: A tool for probabilistic worst-case execution time analysis of real-time systems,'' University of York, Department of Computer Science, Tech. Rep. YCS353, 2003.

[28] F. J. Beutler, ''On two discrete-time system stability concepts and supermartingales,'' *J. Math. Anal. Applicat.*, vol. 44, no. 2, pp. 464–471, 1973.

[29] R. Bhattacharya and G. J. Balas, ''Implementation of control algorithms in an environment of dynamically scheduled CPU time using balanced truncation,'' in *AIAA Guid. Navig. and Contr. Conf. and Exhibit*, Monterey, CA, 2002.

[30] ——, ''Anytime control algorithm: model reduction approach,'' *J. of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 767–776, 2004.

[31] P. Billingsley, *Probability and measure*, 3rd ed. John Wiley & Sons, 1995.

[32] J. J. Binney, N. J. Dowrick, A. J. Fisher, and M. E. J. Newman, *The theory of critical phenomena: an introduction to the renormalization group*. Clarendon Press, 1992.

[33] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, ''Convergence in multiagent coordination, consensus, and flocking,'' in *Proc. IEEE Int. Conf. on Decision and Control*, 2005, pp. 2996–3000.

[34] V. D. Blondel, Y. Nesterov, and J. Theys, ''On the accuracy of the ellipsoid norm approximation of the joint spectral radius,'' *Lin. Alg. Appl.*, vol. 394, pp. 91–107, 2005.

[35] P. Bolzern, P. Colaneri, and G. De Nicolao, ''On almost sure stability of discrete-time Markov jump linear systems,'' in *Proc. IEEE Int. Conf. on Decision and Control*, vol. 3, 2004, pp. 3204–3208.

[36] J. A. Bonachela and M. A. Muñoz, ''Self-organization without conservation: true or just apparent scale-invariance?'' *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 09, p. P09009, Sep. 2009.

[37] D. P. Borgers, R. Postoyan, A. Anta, P. Tabuada, D. Nešić, and W. P. M. H. Heemels, ''Periodic event-triggered control of nonlinear systems using overapproximation techniques,'' *Automatica*, vol. 94, pp. 81–87, 2018.

[38] M. Borrello, ''Modeling and control of systems for critical care ventilation,'' in *Proc. of the American Control Conf.*, Portland, Oregon, 2005, pp. 2166–2180.

[39] M. S. Branicky, ''Stability of hybrid systems: State of the art,'' in *Proc. IEEE Int. Conf. on Decision and Control*, San Diego, California, USA, 1997, pp. 120–125.

[40] L. Brochard, A. Slutsky, and A. Pesenti, ''Mechanical ventilation to minimize progression of lung injury in acute respiratory failure,'' *American Journal of Respiratory and Critical Care Medicine*, vol. 195, no. 4, pp. 438–442, Feb. 2017.

[41] L. J. Brochard and F. Lellouche, *Pressure-Support Ventilation*. New York, NY: The McGraw-Hill Companies, 2013, ch. 8, pp. 199–225.

[42] V. Buendía, S. di Santo, J. A. Bonachela, and M. A. Muñoz, ''Feedback mechanisms for self-organization to the edge of a phase transition,'' *Frontiers in Physics*, vol. 8, Sep. 2020.

[43] R. T. Bupp, D. S. Bernstein, and V. T. Coppola, ''A benchmark problem for nonlinear control design,'' *Int. J. Rob. and Nonlin. Contr.*, vol. 8, no. 4-5, pp. 307–310, 1998.

[44] G. Buttazzo, *Hard real-time computing systems: predictable scheduling algorithms and applications*. Boston: Kluwer Academic Publishers, 1997.

[45] G. Buttazzo, M. Velasco, and P. Marti, ''Quality-of-Control management in overloaded real-time systems,'' *IEEE Trans. on Computers*, vol. 56, no. 2, pp. 253–266, Feb. 2007.

[46] G. Carteaux, M. Parfait, M. Combet, A.-F. Haudebourg, S. Tuffet, and A. Mekontso Dessap, ''Patient-self inflicted lung injury: a practical review,'' *Journal of Clinical Medicine*, vol. 10, no. 12, p. 2738, Jun. 2021.

[47] A. Cervin and J. Eker, ''The control server: A computational model for real-time control tasks,'' in *ECRTS*, 2003, pp. 113–120.

[48] A. Cervin, J. Eker, B. Bernhardsson, and K. E. Årzén, ''Feedback–feedforward scheduling of control tasks,'' *Real-Time Systems*, vol. 23, no. 1, pp. 25–53, 2002.

[49] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen, ''How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime,'' *IEEE Contr. Syst. Mag.*, vol. 23, no. 3, pp. 16–30, Jun. 2003.

[50] A. Cervin, B. Lincoln, J. Eker, K.-E. Årzén, and G. Buttazzo, ''The Jitter Margin and its application in the design of real-time control systems,'' in *Proc. Int. Conf. on Real-Time and Embedded Computing Systems and Applications*, Gothenburg, Sweden, August 2004.

[51] T. Chantem, X. S. Hu, and M. D. Lemmon, ''Generalized elastic scheduling for real-time tasks,'' *IEEE Trans. on Computers*, vol. 58, no. 4, pp. 480–495, Apr. 2009.

[52] R. L. Chatburn, M. El-Khatib, and E. Mireles-Cabodevila, ''A taxonomy for mechanical ventilation: 10 fundamental maxims,'' *Respiratory Care*, vol. 59, no. 11, pp. 1747–1763, Aug. 2014.

[53] D. Chatterjee and D. Liberzon, ''Towards ISS disturbance attenuation for randomly switched systems,'' in *Proc. IEEE Int. Conf. on Decision and Control*, New Orleans, LA, USA, 2007, pp. 5612–5617.

[54] ——, ''Stabilizing randomly switched systems,'' *SIAM J. Contr. Optim.*, vol. 49, no. 5, pp. 2008–2031, 2011.

[55] K. Chatterjee, L. Doyen, and T. A. Henzinger, ''Quantitative languages,'' *ACM Trans. on Comput. Logic*, vol. 11, no. 4, pp. 1–38, 2010.

[56] A. Clauset, C. R. Shalizi, and M. E. J. Newman, ''Power-law distributions in empirical data,'' *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[57] J. A. Cook, I. V. Kolmanovsky, D. McNamara, E. C. Nelson, and K. V. Prasad, ''Control, computing and communications: technologies for the twenty-first century model T,'' *Proc. of IEEE*, vol. 95, no. 2, pp. 334–355, Feb. 2007.

[58] O. L. V. Costa and M. D. Fragoso, ''Stability results for discrete-time linear systems with Markovian jumping parameters,'' *J. Math. Anal. Applicat.*, vol. 179, pp. 154–178, 1993.

[59] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*. Springer & Verlag, 2005.

[60] T. Cucinotta, L. Abeni, L. Palopoli, and G. Lipari, ''A robust mechanism for adaptive scheduling of multimedia applications,'' *ACM Trans. on Embed. Comp. Sys.*, vol. 10, no. 4, p. 46, 2011.

[61] T. Cucinotta, F. Checconi, L. Abeni, and L. Palopoli, ''Self-tuning schedulers for legacy real-time applications,'' in *Proc. of Eur. Conf. on Computer Syst.*, 2010, pp. 55–68.

[62] J. Daafouz, G. Millerioux, and C. Iung, ''A poly-quadratic stability based approach for linear switched systems,'' *Int. J. Contr.*, vol. 75, no. 16-17, pp. 1302–1310, 2002.

[63] X. Dai, ''A Gel'fand-type spectral radius formula and stability of linear constrained switching systems,'' *Lin. Alg. Appl.*, vol. 436, no. 5, pp. 1099–1113, 2012.

[64] A. Dainotti, A. Pescapé, P. S. Rossi, F. Palmieri, and G. Ventre, ''Internet traffic modeling by means of Hidden Markov Models,'' *Comput. Netw.*, vol. 52, no. 14, pp. 2645–2662, Oct. 2008.

[65] T. V. Dang, K.-V. Ling, and D. E. Quevedo, ''Stability analysis of event-triggered anytime control with multiple control laws,'' *IEEE Trans. on Automat. Contr.*, vol. 64, no. 1, pp. 420–426, Jan. 2019.

[66] A. de Candia, A. Sarracino, I. Apicella, and L. de Arcangelis, ''Critical behaviour of the stochastic Wilson-Cowan model,'' *PLOS Computational Biology*, vol. 17, no. 8, pp. 1–23, 08 2021.

[67] E. De Santis, M. D. Di Benedetto, and G. Pola, ''Observability of discrete time linear switching systems,'' *IFAC Proceedings Volumes*, vol. 40, no. 6, pp. 259–264, 2007.

[68] G. S. Deaecto and J. C. Geromel, ''Stability and performance of discrete-time switched linear systems,'' *Systems & Control Letters*, vol. 118, pp. 1–7, Aug. 2018.

[69] R. A. Decarlo, M. S. Branicky, S. Pettersson, and B. Lennartson, *Proc. of the IEEE - special issue on hybrid systems: theory and applications*, P. J. Antsaklis, Ed., 2000, vol. 88, no. 7.

[70] A. Destexhe and J. D. Touboul, ''Is there sufficient evidence for criticality in cortical systems?'' *eneuro*, vol. 8, no. 2, Mar. 2021.

[71] J. L. Diaz, D. F. Garcia, K. Kim, C.-G. Lee, L. L. Bello, J. M. Lopez, S. L. Min, and O. Mirabella, ''Stochastic analysis of periodic real-time systems,'' in *Proc. IEEE Real-Time Sys. Symp.*, 2002, pp. 289–300.

[72] P. Dymarski, Ed., *Hidden Markov Models, theory and applications*. InTech, 2011.

[73] D. V. Efimov and A. L. Fradkov, ''Adaptive tuning to bifurcation for time-varying nonlinear systems,'' *Automatica*, vol. 42, no. 3, pp. 417–425, Mar. 2006.

[74] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: estimation and control*, ser. Applications of mathematics. Springer-Verlag, 1995.

[75] L. Elsner and T. Szulc, ''Convex sets of Schur stable and stable matrices,'' *Linear and Multilinear Algebra*, vol. 48, no. 1, pp. 1–19, 2000.

[76] Y. Fang, ''Stability analysis of linear control systems with uncertain parameters,'' Ph.D. dissertation, Case Western Reserve University, 1994.

[77] ——, ''A new general sufficient condition for almost sure stability of jump linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 42, no. 3, pp. 378–382, 1997.

[78] Y. Fang and K. A. Loparo, ''Stabilization of continuous-time jump linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 47, no. 10, pp. 1590–1603, 2002.

[79] ——, ''Stochastic stability of jump linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 47, no. 7, pp. 1204–1208, 2002.

[80] Y. Fang, K. A. Loparo, and X. Feng, ''Almost sure and $\delta$-moment stability of jump linear systems,'' *Int. J. Contr.*, vol. 59, no. 5, pp. 1281–1307, 1994.

[81] ——, ''Stability of discrete time jump linear systems,'' *J. Math. Sys. Estim. Contr*, vol. 5, no. 3, pp. 275–321, 1995.

[82] X. Feng, K. A. Loparo, Y. Ji, and H. J. Chizeck, ''Stochastic stability properties of jump linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 37, no. 1, pp. 38–53, 1992.

[83] D. Fontanelli, L. Greco, and A. Bicchi, ''Anytime Control algorithms for embedded real-time systems,'' in *Proc. Int. Conf. on Hybrid Systems: Computation and Control*, St. Louis, MO, Apr. 2008, pp. 158–171.

[84] D. Fontanelli, L. Greco, and L. Palopoli, ''Adaptive reservations for feedback control,'' in *Proc. IEEE Int. Conf. on Decision and Control*, Atlanta, GA, USA, Dec. 2010, pp. 4236–4243.

[85] ——, ''Soft real-time scheduling for embedded control systems,'' *Automatica*, vol. 8, pp. 2330–2338, 2013.

[86] D. Fontanelli, L. Palopoli, and L. Greco, ''Deterministic and stochastic QoS provision for real-time control systems,'' in *17th IEEE RTAS*, Chicago, IL, USA, Apr. 2011, pp. 103–112.

[87] ——, ''Optimal CPU allocation to a set of control tasks with soft real–time execution constraints,'' in *Proc. Int. Conf. on Hybrid Systems: Computation and Control.* Philadelphia, PA, USA: ACM, Apr. 2013, pp. 233–242.

[88] D. Fontanelli, L. Greco, and L. Palopoli, ''Optimal mean square control using the continuous stream model of computation,'' in *Proc. IEEE Int. Conf. on Decision and Control*, Osaka, Japan, Dec. 2015, pp. 1958–1965.

[89] ——, ''Optimal resource allocation for stochastic systems performance optimisation of control tasks undergoing stochastic execution times,'' *Int. J. Contr.*, vol. 95, no. 2, pp. 461–472, 2022.

[90] D. Fontanelli, L. Palopoli, and L. Abeni, ''The Continuous Stream model of computation for real–time control,'' in *Proc. IEEE Real-Time Sys. Symp.*, Vancouver, Canada, Dec. 2013, pp. 150–159.

[91] B. V. Frías, L. Palopoli, L. Abeni, and D. Fontanelli, ''The PROSIT tool: Towards the optimal design of probabilistic soft real–time systems,'' *Journal of Software: Practice and Experience*, vol. 48, no. 11, pp. 1940–1967, Nov. 2018.

[92] S. Fürst, ''Challenges in the design of automotive software,'' in *Design, Automation Test in Europe Conference Exhibition*, Mar. 2010, pp. 256–258.

[93] K. Furuta, M. Yamakita, and S. Kobayashi, ''Swing-up control of inverted pendulum using pseudo-state feedback,'' *Proc. of the Inst. of Mechanical Engineers. Pt.I. J. of Sys. and Contr. Engineering*, vol. 206, no. I4, pp. 263–269, 1992.

[94] S. S. Ge, Z. Sun, and T. H. Lee, ''Reachability and controllability of switched linear discrete-time systems,'' *IEEE Trans. on Automat. Contr.*, vol. 46, no. 9, pp. 1437–1441, 2001.

[95] A. Girard and P. Mason, ''Lyapunov functions for shuffle asymptotic stability of discrete-time switched systems,'' *IEEE Contr. Sys. Letters*, vol. 3, no. 3, pp. 499–504, 2019.

[96] M. Girardi-Schappo, E. F. Galera, T. T. A. Carvalho, L. Brochini, N. L. Kamiji, A. C. Roque, and O. Kinouchi, ''A unified theory of E/I synaptic balance, quasicritical neuronal avalanches and asynchronous irregular spiking,'' *J. of Physics: Complexity*, vol. 2, no. 4, p. 045001, Oct. 2021.

[97] E. C. Goligher, M. Dres, E. Fan, G. D. Rubenfeld, D. C. Scales, M. S. Herridge, S. Vorona, M. C. Sklar, N. Rittayamai, A. Lanys, A. Murray, D. Brace, C. Urrea, W. D. Reid, G. Tomlinson, A. S. Slutsky, B. P. Kavanagh, L. J. Brochard, and N. D. Ferguson, ''Mechanical ventilation-induced diaphragm atrophy strongly impacts clinical outcomes,'' *American Journal of Respiratory and Critical Care Medicine*, vol. 197, no. 2, pp. 204–213, Jan. 2018.

[98] S. Grasso and V. Ranieri, ''Proportional assist ventilation,'' *Respiratory Care Clinics of North America*, vol. 7, no. 3, pp. 465–473, Sep. 2001.

[99] L. Greco, D. Fontanelli, and A. Bicchi, ''Almost sure stability of Anytime Controllers via stochastic scheduling,'' in *Proc. IEEE Int. Conf. on Decision and Control*, New Orleans, LO, Dec. 2007, pp. 5640–5645.

[100] ——, ''Robust almost sure stability for uncertain stochastically scheduled anytime controllers,'' in *Proc. IEEE Med. Conf.*, Ajaccio, France, Jun. 2008, pp. 249–254.

[101] ——, ''Design and stability analysis for anytime control via stochastic scheduling,'' *IEEE Trans. on Automat. Contr.*, vol. 56, no. 3, pp. 571–585, 2010.

[102] V. Gupta, ''On an anytime algorithm for control,'' in *Proc. IEEE Int. Conf. on Decision and Control and Chinese Control Conf.*, 2009, pp. 6218–6223.

[103] L. Gurvits, ''Stability of discrete linear inclusion,'' *Lin. Alg. Appl.*, vol. 231, pp. 47–85, 1995.

[104] J. Guttmann, L. Eberhard, B. Fabry, W. Bertschmann, J. Zeravik, M. Adolph, J. Eckart, and G. Wolff, ''Time constant/volume relationship of passive expiration in mechanically ventilated ards patients,'' *European Respiratory Journal*, vol. 8, no. 1, pp. 114–120, Jan. 1995.

[105] G. Hahn, T. Petermann, M. N. Havenith, S. Yu, W. Singer, D. Plenz, and D. Nikolić, ''Neuronal avalanches in spontaneous activity in vivo,'' *Journal of Neurophysiology*, vol. 104, no. 6, pp. 3312–3322, 2010.

[106] R. Hanus, M. Kinnaert, and J. L. Henrotte, ''Conditioning technique, a general anti-windup and bumpless transfer method,'' *Automatica*, vol. 23, no. 6, pp. 729–739, 1987.

[107] R. Hardstone, S.-S. Poil, G. Schiavone, R. Jansen, V. V. Nikulin, H. D. Mansvelder, and K. Linkenkaer-Hansen, ''Detrended fluctuation analysis: a scale-free view on neuronal oscillations,'' *Frontiers in Physiology*, vol. 3, 2012.

[108] T. A. Henzinger, B. Horowitz, and C. M. Kirsch, ''Giotto: a time-triggered language for embedded programming,'' *Proc. of IEEE*, vol. 91, no. 1, pp. 84–99, Jan. 2003.

[109] J. P. Hespanha and A. S. Morse, ''Switching between stabilizing controllers,'' *Automatica*, vol. 38, no. 11, pp. 1905–1917, Nov. 2002.

[110] J. P. Hespanha, P. Santesso, and G. Stewart, ''Optimal controller initialization for switching between stabilizing controllers,'' in *Proc. IEEE Int. Conf. on Decision and Control*, New Orleans, LO, 2007, pp. 5634–5639.

[111] J. Hesse and T. Gross, ''Self-organized criticality as a fundamental property of neural systems,'' *Frontiers in Systems Neuroscience*, vol. 8, Sep. 2014.

[112] G. A. Holt, S. A. Habib, and D. C. Shelledy, *Principles of mechanical ventilation*. Jones & Bartlett Learning, 2019, ch. 3, pp. 95–154.

[113] B. Hunnekens, S. Kamps, and N. Van De Wouw, ''Variable-gain control for respiratory systems,'' *IEEE Transactions on Control Systems Technology*, vol. 28, no. 1, pp. 163–171, Jan. 2020.

[114] Y. Ji and H. J. Chizeck, ''Jump linear quadratic gaussian control: steady-state solution and testable conditions,'' *Control Theory Adv. Technol.*, vol. 6, no. 3, pp. 289–318, 1990.

[115] R. M. Jungers, A. Kundu, and W. P. M. H. Heemels, ''Observability and controllability analysis of linear systems subject to data losses,'' *IEEE Trans. on Automat. Contr.*, vol. 63, no. 10, pp. 3361–3376, 2017.

[116] C. K. Kao and A. Rantzer, ''Stability analysis of systems with uncertain time-varying delays,'' *Automatica*, vol. 43, no. 6, pp. 959–970, Jun. 2007.

[117] K. Kim, J. L. Diaz, L. L. Bello, J. M. Lopez, C.-G. Lee, and S. L. Min, ''An exact stochastic analysis of priority-driven periodic real-time systems and its approximations,'' *IEEE Trans. on Computers*, vol. 54, no. 11, pp. 1460–1466, Nov. 2005.

[118] O. Kinouchi, R. Pazzini, and M. Copelli, ''Mechanisms of self-organized quasicriticality in neuronal network models,'' *Frontiers in Physics*, vol. 8, Dec. 2020.

[119] M. Kitzbichler, M. Smith, S. Christensen, and E. Bullmore, ''Broadband criticality of human brain network synchronization,'' *PLoS computational biology*, vol. 5, no. 3, pp. 1–13, Mar. 2009.

[120] H. Konno, ''A cutting plane algorithm for solving bilinear programs,'' *Mathematical Programming*, vol. 11, no. 1, pp. 14–27, 1976.

[121] H. Kopetz and G. Bauer, ''The time-triggered architecture,'' *Proc. of IEEE*, vol. 91, no. 1, pp. 112–126, 2003.

[122] X. D. Koutsoukos and P. J. Antsaklis, ''Design of stabilizing switching control laws for discrete-and continuous-time linear systems using piecewise-linear Lyapunov functions,'' *Int. J. Contr.*, vol. 75, no. 12, pp. 932–945, Jan. 2002.

[123] F. Kozin, ''On relations between moment properties and almost sure Lyapunov stability for linear stochastic systems,'' *J. Math. Anal. Applicat.*, vol. 10, pp. 324–353, 1965.

[124] ——, ''A survey of stability of stochastic systems,'' *Automatica*, vol. 5, no. 1, pp. 95–112, 1969.

[125] C. Kubrusly and O. L. V. Costa, ''Mean square stability conditions for discrete stochastic bilinear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 30, no. 11, pp. 1082–1087, Nov. 1985.

[126] A. Kundu and D. Chatterjee, ''On stability of discrete-time switched systems,'' *Nonlinear Analysis: Hybrid Systems*, vol. 23, pp. 191–210, 2017.

[127] H. J. Kushner, ''On the stability of stochastic dynamical systems,'' *Proc. Nat. Acad. of Sciences USA*, vol. 53, no. 1, pp. 8–12, 1965.

[128] ——, *Stochastic stability and control*, ser. Mathematics in science and engineering. Academic Press, 1967.

[129] M. Kvasnica, P. Grieder, and M. Baotić, ''Multi-Parametric Toolbox (MPT),'' 2004. [Online]. Available: http://control.ee.ethz.ch/~mpt/

[130] J.-W. Lee and G. E. Dullerud, ''Uniformly stabilizing sets of switching sequences for switched linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 52, no. 5, pp. 868–874, 2007.

[131] J.-W. Lee and P. P. Khargonekar, ''Detectability and stabilizability of discrete-time switched linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 54, no. 3, pp. 424–437, 2009.

[132] A. Levina, J. M. Herrmann, and T. Geisel, ''Dynamical synapses causing self-organized criticality in neural networks,'' *Nature Physics*, vol. 3, no. 12, pp. 857–860, Nov. 2007.

[133] D. Liberzon, J. P. Hespanha, and A. S. Morse, ''Stability of switched systems: a Lie-algebraic condition,'' *Systems & Control Letters*, vol. 37, no. 3, pp. 117–122, 1999.

[134] D. Liberzon and A. S. Morse, ''Basic problems in stability and design of switched systems,'' *IEEE Contr. Syst. Mag.*, vol. 19, no. 5, pp. 59–70, 1999.

[135] D. Liberzon, *Switching in systems and control*. Springer Science & Business Media, 2003.

[136] H. Lin and P. J. Antsaklis, ''Stability and stabilizability of switched linear systems: a survey of recent results,'' *IEEE Trans. on Automat. Contr.*, vol. 54, no. 2, pp. 308–322, 2009.

[137] Q. Ling and M. D. Lemmon, ''Robust performance of soft real-time networked control systems with data dropouts,'' in *Proc. IEEE Int. Conf. on Decision and Control*, vol. 2, 10–13 Dec. 2002, pp. 1225–1230.

[138] ——, ''Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain,'' in *Proc. American Control Conf.*, vol. 6, 4–6 June 2003, pp. 4845–4850.

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

[140] D. Liu, X. S. Hu, M. D. Lemmon, and Q. Ling, ''Firm real-time system scheduling based on a novel QoS constraint,'' *IEEE Trans. on Computers*, vol. 55, no. 3, pp. 320–333, Mar. 2006.

[141] ——, ''Scheduling tasks with Markov-chain based constraints,'' in *Proc. Euromicro Conf. on Real-Time Sys.*, 6–8 July 2005, pp. 157–166.

[142] J. W. S. Liu, *Real–time systems*, T. Holm, Ed. Upper Saddle River, NJ: Prentice Hall Inc., 2000.

[143] J. W. S. Liu, W.-K. Shih, K.-J. Lin, R. Bettati, and J.-Y. Chung, ''Imprecise computation,'' *Proc. of IEEE*, vol. 82, no. 1, pp. 83–93, January 1994.

[144] J. W. S. Liu, K.-J. Lin, W.-K. Shih, A. C.-S. Yu, J.-Y. Chung, and W. Zhao, ''Algorithms for scheduling imprecise computations,'' *Computer*, vol. 24, no. 5, pp. 58–68, 1991.

[145] W. Liu, D. E. Quevedo, Y. Li, and B. Vucetic, ''Anytime control under practical communication models,'' *IEEE Trans. on Automat. Contr.*, vol. 67, no. 10, pp. 5400–5407, Oct. 2022.

[146] S. Manolache, P. Eles, and Z. Peng, ''Memory and time-efficient schedulability analysis of task sets with stochastic execution time,'' in *Proc. Euromicro Conf. on Real-Time Sys.*, 2001, pp. 19–26.

[147] P. Marti, J. M. Fuertes, G. Fohler, and K. Ramamritham, ''Jitter compensation for real-time control systems,'' in *Proc. IEEE Real-Time Sys. Symp.*, 2001, pp. 39–48.

[148] C. Meisel, A. Storch, S. Hallmeyer-Elgner, E. Bullmore, and T. Gross, ''Failure of adaptive self-organized criticality during epileptic seizure attacks,'' *PLOS Computational Biology*, vol. 8, no. 1, pp. 1–8, Jan. 2012.

[149] G. Menesse, B. Marin, M. Girardi-Schappo, and O. Kinouchi, ''Homeostatic criticality in neuronal networks,'' *Chaos, Solitons & Fractals*, vol. 156, p. 111877, Mar. 2022.

[150] C. D. Meyer, *Matrix analysis and applied linear algebra*. Philadelphia, PA, USA: SIAM, 2000.

[151] S. Mohamed, A. U. Awan, D. Goswami, and T. Basten, ''Designing image-based control systems considering workload variations,'' in *Proc. IEEE Int. Conf. on Decision and Control*, Nice, France, 2019.

[152] L. Moreau, ''Stability of multiagent systems with time-dependent communication links,'' *IEEE Trans. on Automat. Contr.*, vol. 50, no. 2, pp. 169–182, 2005.

[153] L. Moreau, E. Sontag, and M. Arcak, ''Feedback tuning of bifurcations,'' *Systems & Control Letters*, vol. 50, no. 3, pp. 229–239, Oct. 2003.

[154] T. Morozan, ''Optimal stationary control for dynamic systems with Markov perturbations,'' *Stoch. Anal. Appl.*, vol. 3, no. 1, pp. 299–325, 1983.

[155] J. R. Moyne and D. M. Tilbury, ''The emergence of industrial control networks for manufacturing control, diagnostics, and safety data,'' *Proc. of IEEE*, vol. 95, no. 1, pp. 29–47, 2007.

[156] G. Murias, U. Lucangelo, and L. Blanch, ''Patient-ventilator asynchrony,'' *Current Opinion in Critical Care*, vol. 22, no. 1, pp. 53–59, Feb. 2016.

[157] J. Nilsson and B. Bernhardsson, ''Analysis of real-time control systems with time delays,'' in *Proc. IEEE Int. Conf. on Decision and Control*, 1996, pp. 3173–3178.

[158] J. R. Norris, *Markov chains*. Cambridge university press, 1998, no. 2.

[159] A. Orlando, ''How to improve patient-ventilator synchrony — waveform analysis and optimization of ventilator settings,'' Hamilton Medical AG, White paper, 2018.

[160] L. Palopoli, L. Abeni, G. Bolognini, B. Allotta, and F. Conticelli, ''Novel scheduling policies in real-time multithread control system design,'' *Control Engineering Practice*, vol. 10, no. 10, pp. 1095–1110, November 2002.

[161] L. Palopoli, L. Abeni, G. Buttazzo, F. Conticelli, and M. Di Natale, ''Real-time control system analysis: an integrated approach,'' in *Proc. IEEE Real-Time Sys. Symp.*, Orlando, FL , USA, Nov. 2000, pp. 131–140.

[162] L. Palopoli, T. Cucinotta, L. Marzario, and G. Lipari, ''AQuoSA - adaptive quality of service architecture,'' *Software: Practice and Experience*, vol. 39, no. 1, pp. 1–31, 2009.

[163] L. Palopoli, C. Pinello, A. Bicchi, and A. Sangiovanni-Vincentelli, ''Maximizing the stability radius of a set of systems under real-time scheduling constraints,'' *IEEE Trans. on Automat. Contr.*, vol. 50, no. 11, pp. 1790–1795, Nov. 2005.

[164] W. Pasillas-Lépine, S. Tuffet, C. Soussen, S. Gendreau, M. A. Boujelben, A. Mekontso-Dessap, and G. Carteaux, ''A novel method for noninvasive estimation of respiratory effort during pressure support ventilation,'' *Biomedical Signal Processing and Control*, vol. 93, p. 106176, Jul. 2024.

[165] P. Pepe, ''Converse Lyapunov theorems for discrete-time switching systems with given switches digraphs,'' *IEEE Trans. on Automat. Contr.*, 2018.

[166] N. Perrin and B. H. Ferri, ''Digital filters with adaptive length for real–time applications,'' in *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium*, K. E. Le Royal Meridien, Ed., Toronto, Canada, May 2004.

[167] T. Petermann, T. C. Thiagarajan, M. A. Lebedev, M. A. L. Nicolelis, D. R. Chialvo, and D. Plenz, ''Spontaneous cortical activity in awake monkeys composed of neuronal avalanches,'' *Proceedings of the National Academy of Sciences*, vol. 106, no. 37, pp. 15 921–15 926, 2009.

[168] S. Pettersson and B. Lennartson, ''Stabilization of hybrid systems using a min-projection strategy,'' in *Proc. American Control Conf.*, Arlington, Virginia, 2001, pp. 223–228.

[169] M. Philippe, R. Essick, G. E. Dullerud, and R. M. Jungers, ''Stability of discrete-time switching systems with constrained switching sequences,'' *Automatica*, vol. 72, pp. 242–250, 2016.

[170] A. Quagli, D. Fontanelli, L. Greco, L. Palopoli, and A. Bicchi, ''Designing real-time embedded controllers using the anytime computing paradigm,'' in *Proc.*

*IEEE Conf. on Emerging Technologies and Factory Automation ETFA*, Mallorca, Spain, Sep. 2009, pp. 1–8.

[171] ——, ''Design of embedded controllers based on anytime computing,'' *IEEE Trans. on Industrial Infor.*, vol. 6, no. 4, pp. 492–502, Nov. 2010.

[172] D. E. Quevedo and V. Gupta, ''Sequence-based Anytime Control,'' *IEEE Trans. on Automat. Contr.*, vol. 58, no. 2, pp. 377–390, Feb. 2013.

[173] D. E. Quevedo, V. Gupta, W.-J. Ma, and S. Yüksel, ''Stochastic stability of event-triggered Anytime Control,'' *IEEE Trans. on Automat. Contr.*, vol. 59, no. 12, pp. 3373–3379, Dec. 2014.

[174] D. E. Quevedo, W.-J. Ma, and V. Gupta, ''Anytime Control using input sequences with Markovian processor availability,'' *IEEE Trans. on Automat. Contr.*, vol. 60, no. 2, pp. 515–521, Feb. 2015.

[175] L. Rabiner, ''A tutorial on Hidden Markov Models and selected applications in speech recognition,'' *Proc. of IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[176] R. Rajkumar, K. Juvva, A. Molano, and S. Oikawa, ''Resource kernels: A resource-centric approach to real-time and multimedia systems,'' in *Proc. of the SPIE/ACM Conf. on Multimedia Comp. and Netw.*, Jan. 1998.

[177] P. Ramanathan, ''Overload management in real-time control applications using (m, k)-firm guarantee,'' *IEEE Trans. on Parallel and Distrib. Syst.*, vol. 10, no. 6, pp. 549–559, 1999.

[178] W. J. Reed and B. D. Hughes, ''From gene families and genera to incomes and internet file sizes: Why power laws are so common in nature,'' *Physical Review E*, vol. 66, p. 067103, Dec. 2002.

[179] J. W. Roberts, ''A survey on statistical bandwidth sharing,'' *Computer Networks*, vol. 45, no. 3, pp. 319–332, 2004.

[180] P. S. Rossi, G. Romano, F. Palmieri, and G. Iannello, ''Joint end-to-end loss-delay hidden Markov model for periodic UDP traffic over the internet,'' *IEEE Trans. on Signal Proc.*, vol. 54, no. 2, pp. 530–541, Feb. 2006.

[181] K. Salamatian and S. Vaton, ''Hidden Markov modeling for network communication channels,'' in *Proc. of ACM SIGMETRICS Int. Conf. on Measur. and Modeling of Comp. Sys.*, New York, NY, USA, 2001, pp. 92–101.

[182] P. Salvador, A. Pacheco, and R. Valadas, ''Modeling IP traffic: Joint characterization of packet arrivals and packet sizes using BMAPs,'' *Comput. Netw.*, vol. 44, no. 3, pp. 335–352, Feb. 2004.

[183] C. S. H. Sassoon, ''Triggering of the ventilator in patient-ventilator interactions,'' *Respiratory Care*, vol. 56, no. 1, pp. 39–51, Jan. 2011.

[184] M. Scheffer, S. R. Carpenter, T. M. Lenton, J. Bascompte, W. Brock, V. Dakos, J. van de Koppel, I. A. van de Leemput, S. A. Levin, E. H. van Nes, M. Pascual, and J. Vandermeer, ''Anticipating critical transitions,'' *Science*, vol. 338, no. 6105, pp. 344–348, 2012.

[185] J. P. Sethna, K. A. Dahmen, and C. R. Myers, ''Crackling noise,'' *Nature*, vol. 410, no. 6825, pp. 242–250, Mar. 2001.

[186] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, ''On task schedulability in real-time control systems,'' in *Proc. IEEE Real-Time Sys. Symp.*, 1996, pp. 13–21.

[187] W. L. Shew and D. Plenz, ''The functional benefits of criticality in the cortex,'' *The Neuroscientist*, vol. 19, no. 1, pp. 88–100, May 2013.

[188] E. D. Sontag, ''Smooth stabilization implies coprime factorization,'' *IEEE Trans. on Automat. Contr.*, vol. 34, no. 4, pp. 435–443, apr 1989.

[189] J. Spiliotis and J. Tsinias, ''Notions of exponential robust stochastic stability, ISS and their Lyapunov characterizations,'' *Int. J. Rob. and Nonlin. Contr.*, vol. 13, no. 2, pp. 173–187, 2003.

[190] Z. Sun, *Switched linear systems: control and design*. Springer Science & Business Media, 2006.

[191] S. Sutrisno and S. Trenn, ''Observability and determinability characterizations for linear switched systems in discrete time,'' in *Proc. IEEE Int. Conf. on Decision and Control*, 2021, pp. 2474–2479.

[192] M. Tanelli, B. Picasso, P. Bolzern, and P. Colaneri, ''Almost sure stabilization of uncertain continuous-time Markov jump linear systems,'' *IEEE Trans. on Automat. Contr.*, vol. 55, no. 1, pp. 195–201, 2010.

[193] A. Tanwani, H. Shim, and D. Liberzon, ''Observability for switched linear systems: characterization and observer design,'' *IEEE Trans. on Automat. Contr.*, vol. 58, no. 4, pp. 891–904, 2012.

[194] C. Tetzlaff, S. Okujeni, U. Egert, F. Wörgötter, and M. Butz, ''Self-organized criticality in developing neuronal networks,'' *PLOS Computational Biology*, vol. 6, no. 12, pp. 1–18, Dec. 2010.

[195] J. Touboul and A. Destexhe, ''Can power-law scaling and neuronal avalanches arise from stochastic dynamics?'' *PLOS ONE*, vol. 5, no. 2, pp. 1–14, Feb. 2010.

[196] ——, ''Power-law statistics and universal scaling in the absence of criticality,'' *Physical Review E*, vol. 95, no. 1, Jan. 2017.

[197] J. Tsinias, ''Stochastic input-to-state stability and applications to global feedback stabilization,'' *Int. J. Contr.*, vol. 71, no. 5, pp. 907–930, Jan. 1998.

[198] ——, ''The concept of "exponential input to state stability" for stochastic systems and applications to feedback stabilization,'' *Systems & Control Letters*, vol. 36, pp. 221–229, 1999.

[199] W. Tucker, ''Computing accurate Poincaré maps,'' *Physica D: Nonlinear Phenomena*, vol. 171, no. 3, pp. 127–137, Oct. 2002.

[200] J. van Gorp, ''Diagnostic et observation d'une classe de systèmes dynamiques hybrides. Application au convertisseur multicellulaire série,'' Ph.D. dissertation, Université de Valenciennes et du Hainaut-Cambresis, 2013.

[201] J. Van Gorp, M. Defoort, M. Djemai, and N. Manamanni, ''Hybrid observer for the multicellular converter,'' *IFAC Proceedings Volumes*, vol. 45, no. 9, pp. 259–264, 2012.

[202] M. Velasco, P. Martí, and E. Bini, ''Control-driven tasks: modeling and analysis,'' in *Proc. IEEE Real-Time Sys. Symp.*, 2008, pp. 280–290.

[203] J. von Neumann, ''Über ein ökonomisches gleichungs-system und eine verallgemeinerung des brouwerschen fixpunktsatzes,'' *Ergebnisse eines Mathmatischen Kolloquiums*, vol. 8, pp. 1945–1946, 1937.

[204] G. Wang and Y. Sun, ''Almost sure stabilization of continuous-time jump linear systems via a stochastic scheduled controller,'' *IEEE Trans. on Cyber.*, vol. 52, no. 5, pp. 2712–2724, May 2022.

[205] X. Wang and M. D. Lemmon, ''Event-triggering in distributed networked control systems,'' *IEEE Trans. on Automat. Contr.*, vol. 56, no. 3, pp. 586–601, 2011.

[206] Y. Wang, N. Roohi, G. E. Dullerud, and M. Viswanathan, ''Stability analysis of switched linear systems defined by regular languages.'' *IEEE Trans. on Automat. Contr.*, vol. 62, no. 5, pp. 2568–2575, 2017.

[207] G. Weiss and R. Alur, ''Automata based interfaces for control and scheduling,'' in *Int. Wor. on Hybrid Sys. Comp. and Contr.*, 2007, pp. 601–613.

[208] M. A. Wicks and R. A. DeCarlo, ''Solution of coupled Lyapunov equations for the stabilization of multimodal linear systems,'' in *Proc. American Control Conf.*, Albuquerque, NM, 1997, pp. 1709–1713.

[209] M. A. Wicks, P. Peleties, and R. A. DeCarlo, ''Construction of piecewise Lyapunov functions for stabilizing switched systems,'' in *Proc. IEEE Int. Conf. on Decision and Control*, Lake Buena Vista, FL, 1994, pp. 3492–3497.

[210] G. Xie and L. Wang, ''Necessary and sufficient conditions for controllability and observability of switched impulsive control systems,'' *IEEE Trans. on Automat. Contr.*, vol. 49, no. 6, pp. 960–966, 2004.

[211] Y. Xu, A. Cervin, and K. E. Årzén, ''Harmonic scheduling and control co-design,'' in *IEEE Int. Conf. on Embedded and Real-Time Computing Systems and Applications*, Aug. 2016, pp. 182–187.

[212] Y. Xu, A. Cervin, and K. Årzén, ''Jitter-robust LQG control and real-time scheduling co-design,'' in *Proc. American Control Conf.*, Jun. 2018, pp. 3189–3196.

[213] Y. Yamada and H.-L. Du, ''Analysis of the mechanisms of expiratory asynchrony in pressure support ventilation: a mathematical approach,'' *Journal of Applied Physiology*, vol. 88, no. 6, pp. 2143–2150, Jun. 2000.

[214] H. Ye, A. N. Michel, and L. Hou, ''Stability theory for hybrid dynamical systems,'' *IEEE Trans. on Automat. Contr.*, vol. 43, no. 4, pp. 461–474, 1998.

[215] M. Younes, ''Proportional Assist Ventilation, a new approach to ventilatory support: theory,'' *American Review of Respiratory Disease*, vol. 145, no. 1, pp. 114–120, Jan. 1992.

[216] L. Zaccarian and A. R. Teel, ''The $\mathcal{L}_2(l_2)$ bumpless transfer problem: its definition and solution,'' in *Proc. IEEE Int. Conf. on Decision and Control*, vol. 5, Atlantis, Paradise Island, Bahamas, 2004, pp. 5505–5510.

[217] Q. Zhang, ''On stability of the Kalman filter for discrete time output error systems,'' *Systems & Control Letters*, vol. 107, pp. 84–91, 2017.

[218] P. Zhao, W. Feng, and Y. Kang, ''Stochastic input-to-state stability of switched stochastic nonlinear systems,'' *Automatica*, vol. 48, no. 10, pp. 2569–2576, Oct. 2012.

[219] Y. Zhu and Z. Sun, ''Constructive design for reachability of reversible discrete-time switched linear systems,'' in *Chinese Control Conf.*, 2019, pp. 1825–1828.

[220] ——, ''Minimum realization for controllability/observability of switched linear systems,'' *Nonlinear Analysis: Hybrid Systems*, vol. 51, p. 101426, 2024.

[221] W. Zucchini and I. L. MacDonald, *Hidden Markov Models for time series: an introduction using R*, ser. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2009.

[222] R. Zurawski, ''Special issue on industrial communication systems,'' *Proc. of IEEE*, vol. 93, no. 6, 2005.

[223] G. Ódor, ''Universality classes in nonequilibrium lattice systems,'' *Reviews of Modern Physics*, vol. 76, no. 3, pp. 663–724, Aug. 2004.