

THÈSE DE DOCTORAT

de
l'UNIVERSITÉ PARIS-SACLAY

École doctorale de mathématiques Hadamard (EDMH, ED 574)

Établissement d'inscription : École Normale Supérieure Paris-Saclay

Établissement d'accueil : École Normale Supérieure Paris-Saclay

Laboratoire d'accueil : Centre de Mathématiques et de Leurs Applications (UMR CNRS 8536)

Spécialité de doctorat : Mathématiques appliquées

Adrien LE COËNT

Guaranteed control synthesis for switched space-time
dynamical systems

Date de soutenance : 02 Octobre 2017

Après avis des rapporteurs : THIERRY HORSIN (CNAM Paris)
LUC JAULIN (Lab-STICC, ENSTA Bretagne)

LUDOVIC CHAMOIN	(LMT) Codirecteur de thèse
THAO DANG	(Verimag) Examinatrice
FLORIAN DE VUYST	(CMLA) Directeur de thèse
LAURENT FRIBOURG	(LSV) Codirecteur de thèse
Jury de soutenance :	THIERRY HORSIN (CNAM) Rapporteur
	LUC JAULIN (Lab-STICC) Rapporteur
	SYLVIE PUTOT (LIX) Examinatrice
	CHRISTIAN REY (SafranTech) Invité
	EMMANUEL TRÉLAT (LJLL) Examinateur

Title : Guaranteed control synthesis for switched space-time dynamical systems

Keys words : Control synthesis, Model reduction, Switched control systems, Compositional synthesis, Partial differential equations

Abstract : In this thesis, we focus on switched control systems described by partial differential equations, and investigate the issues of guaranteed control of such systems using state-space decomposition methods. The use of state-space decomposition methods requires model order reduction, control of the different sources of error for quantities of interest, and measure of uncertainties on the states and parameters of the system. We are considering using set-based computation methods, in association with model order reduction techniques, along with the use of state-observers for on-line estimation of the system.



Summary

In this thesis, we focus on switched control systems and investigate the issue of guaranteed (correct- by-design) control of such systems. More specifically, we focus on control synthesis, and wish to extend the field of application of the existing algorithms, notably for problems described by partial differential equations. Indeed, the existing algorithms mainly rely on a state-space decomposition or discretization, associated to reachable set computations, and their computational complexity is exponential with respect to the dimension of the system, which strongly restricts the complexity of the systems one can study. A first issue tackled in this thesis is the improvement of the reachable set computations, by extending them to non-linear systems with the use of guaranteed numerical schemes. We also propose an extremely fast approach based on the Euler method associated to a hypothesis close to incremental stability. Secondly, in order to increase the dimension of the systems handled by such methods, we propose distributed (compositional) versions of the synthesis algorithms, allowing to break the exponential complexity by synthesizing controllers on sub-parts of the system, but implying additional constraints which can be handled by approaches such as assume/guarantee reasoning. Lastly, the direct application to partial differential equations, even in their discretized form, is still intractable for such algorithms. To reach this goal, we propose approaches based on model order reduction methods, allowing to decrease the complexity of the studied system by approaching it with a low dimensional system, but which require taking the different sources of error into account. While the first applications of correct-by-design methods successfully synthesized robust controllers for systems such as power converters modeled switched control systems of dimension 2, we applied our methods to case studies such as the floor heating of an eleven room house (concrete case study proposed by the Danish company Seluxit), boundary control of the heat equation, or vibration control of metal plates.

Contents

1	Introduction	1
1.1	Control of dynamical systems	2
1.2	Symbolic methods and switched systems	3
1.2.1	State of the art	3
1.2.2	Motivations	4
1.3	The reachable set computation	5
1.3.1	Guaranteed Runge-Kutta schemes	6
1.3.2	The Euler method	7
1.4	Compositional approaches	8
1.5	Model order reduction methods	9
1.6	Contributions	10
2	Switched systems	13
2.1	Introduction	14
2.2	Switched systems	14
2.3	General principle	18
2.3.1	The state-space bisection algorithm	19
2.3.2	A covering of balls	22
2.4	Improving the research of patterns	24
2.5	Computational cost	25
2.6	Final remarks	26
3	Reachable set computation	29
3.1	Zonotopes and linear systems	30
3.2	Validated simulation and state-space bisection	31
3.2.1	Validated simulation	31
3.2.2	Experimentations	34
3.2.3	Performance tests	39
3.2.4	Final remarks	42
3.3	Sampled switched systems with one-sided Lipschitz conditions	42
3.3.1	Lipschitz and one-sided Lipschitz condition	42
3.3.2	A note on the OSL constant for linear systems	44
3.3.3	Euler approximate solutions	46
3.3.4	Application to control synthesis	49
3.3.5	Numerical experiments and results	51
3.3.6	Final remarks	57

4 Disturbances and distributed control	59
4.1 Linear systems and disturbances	60
4.2 Distributed control using zonotopes	62
4.2.1 State-dependent Switching Control	62
4.2.2 Control synthesis using tiling	64
4.2.3 Centralized control	67
4.2.4 Distributed control	70
4.2.5 Case study	74
4.2.6 Continuous-time case	76
4.2.7 Reachability in continuous time	76
4.2.8 Final remarks	80
4.3 Perturbed and distributed Euler scheme	80
4.3.1 Distributed synthesis	84
4.3.2 Application	87
4.3.3 Final remarks and future work	88
5 Control of high dimensional ODEs	91
5.1 Background	93
5.2 Problem setting	94
5.3 Model order reduction	96
5.3.1 The balanced truncation	98
5.3.2 Error bounding	99
5.4 Reduced order control	100
5.4.1 Offline procedure	101
5.4.2 Semi-online procedure	102
5.5 Numerical results	104
5.5.1 Thermal problem on a metal plate	104
5.5.2 Vibrating beam	106
5.5.3 Vibrating aircraft panel	108
5.6 Extension to output feedback control	110
5.6.1 Partial observation	111
5.6.2 Convergence of the observer	112
5.6.3 Observer based decomposition	114
5.6.4 Reduced output feedback control	114
5.7 Final remarks	116
6 Control of PDEs	119
6.1 Introduction	120
6.2 Setting of the problem	121
6.3 Spectral decomposition and EIM	122
6.3.1 Problem statement	122
6.3.2 Spectral Model Reduction	123
6.3.3 Error bounding	126
6.4 L^2 guaranteed control	127
6.4.1 Transformation of the problem	128
6.4.2 Stability requirements	128
6.4.3 Strategy for stability control	131
6.4.4 Certified reduced basis for control	133

6.4.5 Numerical experiment for the L^2 guaranteed control synthesis by stability of error balls	135
6.5 Reliable measurements, online control, and other applications	136
7 Conclusions and perspectives	139
Appendices	143
A Case studies modeled by ODEs	145
A.1 Boost DC-DC converter	145
A.2 Two-room apartment	145
A.3 A polynomial example	146
A.4 Four room apartment	146
A.5 Linearized four room apartment	148
A.6 A path planning problem	148
A.7 Two-tank system	148
A.8 Helicopter	149
A.9 Eleven room house	149
B Proof of Lemma 1	151
Bibliography	153

Chapter 1

Introduction

In recent years, there has been an increasing interest in studying hybrid systems, which allow to model a wide range of cyber-physical systems. These models have been applied with success in various domains such as automotive industry, power electronics, smart houses, medical monitoring, robotic systems... Switched control systems (switched systems for short) are a sub-class of hybrid systems, and their importance has grown considerably over the last years because of their ease of implementation for controlling cyber-physical systems. One of the main issues raised in the study of switched systems is the improvement of robust and flexible control techniques in order to increase reliability and safety of operation. A switched system is constituted of two parts: a family of continuous systems called *modes*, each having its own dynamics; and a *switching signal* that selects which mode is active. We suppose that only one mode is active at a given time. The switching signal can be state dependent and/or time dependent. Switched systems are thus merely described by piecewise dynamics.

The dynamics of the modes of switched systems is usually described by ordinary differential equations (ODEs), and many tools exist to control such systems. But the complexity of the systems describing nowadays problems grows more and more, and switching modes described by partial differential equations (PDEs) are being paid more attention. We should point out that one of the main difficulties arising in switched systems with respect to classical systems is that the state of the system cannot usually be asymptotically stabilized by a continuous feedback control law [38], whether the dynamics is described by ODEs or PDEs. Therefore, the stability notions that we define in this thesis are closer to invariance than classical stability.

1.1 Control of dynamical systems

While many tools and methods successfully manage to provide control laws ensuring some properties for the controlled systems, such as stability or reachability, the approach to be used often depends on the particular application aimed by the model. For example, optimal control approaches, basically aimed at minimizing a cost function and permitting to reach a target state under given constraints, are often used in aerospace engineering [113, 171]. They have also been applied on infinite dimensional cases for PDEs [62, 90, 91, 158]. They are however often computationally expensive and require sophisticated numerical methods to be applied online. Lyapunov theory approaches provide ways to analyze and stabilize controlled systems. They merely rely on energy (Lyapunov) functions, characterizing the state of the system, and ensure stability when they reach a 0 level. These type of approaches have been applied to nonlinear control systems [104, 174], hybrid systems [79], and for infinite dimensional systems [25, 49, 87, 131]. Note that the case of PDEs is still an actual challenge since every method is different depending on the type of equa-

tion. For the case of switched systems, the use of common Lyapunov functions also provides efficient control laws [124, 170]. Some recent works also give stability and stabilization results for switched partial differential equations [111, 128, 150]. We should however point out that there is no general method for determining a suitable Lyapunov function, whether it is for ODEs or PDEs, which makes these types of approaches more tied to given case studies, and harder to apply in a general case. Furthermore, even though all these methods provide strong results for the controlled systems, the online application is often performed with digital devices, involving a discretization of the state and/or control input. Numerical schemes can also be used, and these additional tools inevitably imply numerical errors that are not taken into account, and could thus lead to safety problems, particularly in safety critical systems. This is why we focus here on guaranteed, or “correct-by-design” methods. A correct-by-design method ensures that, with respect to a mathematical model, every possible working case or behavior of a system is taken into account and made safe. It should include all the possible perturbations induced by the external environment. The appropriate tool for this purpose is symbolic methods, which exhaustively control all the possible states of the system, and can be associated to guaranteed numerical schemes, i.e., which take all the numerical errors into account. They also provide the advantage of being fully automated, and do not require, for example, the estimation of a Lyapunov function.

1.2 Symbolic methods and switched systems

In this thesis, we focus on the subclass of sampled switched systems, for which switches occur periodically at a fixed switched period denoted by τ . These switching signals are very common because of their ease of implementation. Given that a physical actuator cannot change its state infinitely fast, it is also realistic to consider a fixed period at which the actuator can change its state. This sub-class is particularly adapted to the use of numerical schemes, and in general, numerical methods allowing to synthesize controllers offline. Note however that [2] provides a symbolic method allowing to have variable time periods.

1.2.1 State of the art

Symbolic methods for controlling sampled switched systems are numerous, rely on different tools, and often require some hypotheses on the dynamics of the system. Note that symbolic methods also apply to classical (finite dimensional) control systems, but generally discretize the control input, which finally comes back to switched system models. Most methods rely on finite state abstractions, which basically discretize (abstract) the state space of the system in order to transform it into a finite state automaton, for which multiple tools exist for performing control

synthesis (e.g. BDDs: binary decision diagrams). The states of the automaton are then called symbols, and the finite state automaton is a symbolic, or abstract model of the system. However, the guaranteed aspect still depends on the abstraction method. For example, the tool PESSOA [132] synthesizes a finite state abstraction which is (alternatingly) approximately bisimilar to the original model. It basically ensures that the trajectories of the real system stay close to those of the symbolic model with a given precision. This tool is available for linear systems, but nonlinear extensions are available with additional hypotheses such as incremental global asymptotic stability or incremental input-to-state stability [149]. Roughly speaking, incremental stability is quite a strong hypothesis which assumes, for each mode, that two trajectories always get exponentially closer within time. More information on the incremental stability property is detailed in [13]. The tool CoSyMA [142] uses approximate bisimulation as well and assumes that the system is incrementally stable, but includes multi scale abstractions, which means that the discretization adapts to the system and uses more discrete states where needed. More information on abstractions using approximate simulations is given in [75, 77]. The tool SCOTS [159] also relies on finite state abstractions but uses a different concept named feedback refinement relations developed in [154]. Another class of finite state abstractions relies on tilings of the state space. Associated to the hypothesis of monotonicity, which assumes that the trajectories of the system stay ordered, it is possible to simply compute the image of a set by computing the images of the extreme points of the tiles. Finite state abstractions can then be constructed for control synthesis. These approaches are used in [103, 136]. A quite different type of abstraction is used in [122], where the symbolic states are mode sequences, but this method also requires the hypothesis of incremental stability. A recent abstraction approach [153] uses robust control Lyapunov-like functions, which are automatically computed using a counter-example inductive synthesis by means of an SMT solver (which solves a decision problem).

1.2.2 Motivations

While all these approaches are efficient on practical case studies, most of them make strong assumptions on the dynamics of the system (such as incremental stability or monotonicity). In this thesis, we develop methods that do not require such strong assumptions. While we first introduce methods for linear systems, the application to nonlinear systems is made possible with guaranteed numerical schemes that require the weakest hypotheses possible, such as locally Lipschitz dynamics. We will base our developments on the tool MINIMATOR [106], which synthesizes controllers with an adaptive tiling of the state space, associated to an exhaustive research of possible control sequences (up to a given length) which either succeeds to find an admissible control sequence, or fails and decomposes further the state

space (adaptation). This procedure developed by Romain Soulat, called state-space decomposition, is presented for linear finite dimensional systems in [66, 68]. It actually provides for an efficient way to compute state-dependent controllers ensuring discrete-time properties, i.e. ensured at the switching instants $\tau, 2\tau\dots$. Note that the use of polyhedral symbolic states, as used here, is classical (see e.g. [17, 72]), and the use of tiling or partitioning of the state-space using bisection is also classical (see e.g. [76, 94]). One of the objectives of this thesis is to apply this procedure to nonlinear systems, while also ensuring continuous time properties. In order to apply this approach with safety properties ensured for all time, one first needs to compute a tube of reachability, and no longer just an image at discrete instants of an initial set (easily computable for linear systems). In other words, we have to compute a solution of a nonlinear ODE with an initial condition given as a set. The extension to nonlinear systems thus requires new tools for the computation of the reachable sets, namely, guaranteed numerical schemes.

An inevitable drawback of symbolic methods is their computational complexity, subject to the so-called “curse of dimensionality”. Indeed, most of them are based on finite state abstractions, and the resulting size of the symbolic models is exponential with respect to the dimension of the system. While our method of adaptive tiling manages to keep the number of symbols quite low, it still struggles to synthesize controllers for systems of dimensions larger than 8 in reasonable amounts of time. In order to overcome this issue, we propose to apply compositional principles, and develop distributed versions of these algorithms.

Finally, symbolic approaches have never been applied to switched systems described by PDES. We aim at providing formal safety or reachability guarantees for such systems by using symbolic methods. In their discretized forms (using for example finite element methods), PDEs lead to high dimensional ODEs, and the straightforward application of a symbolic method is irrelevant. Fortunately, reducing the dimension of a PDE model is an important issue in the field of computational mechanics, with many applications (optimization of a process, storage reduction, virtual abacus...). We thus propose to use such techniques in association to symbolic methods to reach this goal.

1.3 The reachable set computation

Computing the solution at discrete times of a linear ODE when the initial condition is given as a box can be easily done using *zonotopes* [10, 73, 105, 109], and this, because we know exactly the solution of the ODE, and can be written as an affine transformation. Yet, generally, the exact solution of nonlinear differential equations cannot be obtained, and a numerical integration scheme is used to approximate the state of the system. With the objective of computing a guaranteed control, which ensures continuous time properties, the computation of a reachability tube is

mandatory.

Given an ODE of the form $\dot{x}(t) = f(t, x(t))$, and a set of initial values X_0 , a symbolic (or “set-valued” since the symbols used here are sets) integration method consists in computing a sequence of approximations (t_n, \tilde{x}_n) of the solution $x(t; x_0)$ of the ODE with $x_0 \in X_0$ such that $\tilde{x}_n \approx x(t_n; x_{n-1})$. Symbolic integration methods extend classical *numerical* integration methods which correspond to the case where X_0 is just a singleton $\{x_0\}$. The simplest numerical method is Euler’s method in which $t_{n+1} = t_n + h$ for some step-size h and $\tilde{x}_{n+1} = \tilde{x}_n + h f(t_n, \tilde{x}_n)$; so the derivative of x at time t_n , $f(t_n, x_n)$, is used as an approximation of the derivative on the whole time interval. This method is very simple and fast, but requires small step-sizes h . More advanced methods coming from the Runge-Kutta family use a few intermediate computations to improve the approximation of the derivative. The general form of an explicit s -stage Runge-Kutta formula of the form $\tilde{x}_{n+1} = \tilde{x}_n + h \sum_{i=1}^s b_i k_i$ where $k_i = f(t_n + c_i h, \tilde{x}_n + h \sum_{j=1}^{i-1} a_{ij} k_j)$ for $i = 2, 3, \dots, s$. A challenging question is then to compute a bound on the distance between the true solution and the numerical solution, i.e.: $\|x(t_n; x_{n-1}) - x_n\|$. This distance is associated to the *local truncation error* of the numerical method.

We develop two approaches relying on this type of numerical schemes. The first one makes use of Runge-Kutta type schemes and interval methods. The second one is a renewal of the Euler method, with a new error bound allowing to compute reachability tubes using balls.

1.3.1 Guaranteed Runge-Kutta schemes

Most of the recent work on the symbolic (or set-valued) integration of nonlinear ODEs is based on the upper bounding of the Lagrange remainders either in the framework of Taylor series or Runge-Kutta schemes [6, 8, 35, 37, 42, 43, 56, 130]. Sets of states are generally represented as vectors of intervals (or “rectangles”) and are manipulated through interval arithmetic [141] or affine arithmetic [54]. Taylor expansions with Lagrange remainders are also used in the work of [8], which uses “polynomial zonotopes” for representing sets of states in addition to interval vectors.

The *guaranteed* or *validated* solution of ODEs using interval arithmetic is studied in the framework of Taylor series in [59, 125, 141, 144], and Runge-Kutta schemes in [6, 35, 36, 71]. The former is the oldest method used in interval analysis community because the expression of the remainder of Taylor series is simple to obtain. Nevertheless, the family of Runge-Kutta methods is very important in the field of numerical analysis. Indeed, Runge-Kutta methods have several interesting stability properties which make them suitable for an important class of problems. The recent work [5] implements Runge-Kutta based methods which prove their efficiency at low orders and for short simulations (fixed by the sampling period of the controller).

In the methods of symbolic analysis and control of hybrid systems, the way

of representing sets of state values and computing reachable sets for systems defined by autonomous ordinary differential equations (ODEs) is fundamental (see for example [9, 74]). Many tools using, among other techniques, linearization or hybridization of these dynamics are now available (*e.g.*, SpaceEx [64], Flow* [43], iSAT-ODE [61]). An interesting approach appeared recently, based on the propagation of reachable sets using guaranteed Runge-Kutta methods with adaptive step-size control (see [35, 92]). An originality of our work is to use such guaranteed integration methods in the framework of switched systems. This notion of guarantee of the results is very interesting, because it allows applications in critical domains, such as aeronautical, military and medical ones.

1.3.2 The Euler method

In the end, the Runge-Kutta based method of [5] remains an elaborated method that requires the use of affine arithmetic, application of the Banach's fixpoint theorem and Picard-Lindelöf operator, see [144]. Despite being very efficient and accurate, it still requires a lot of computations for every time step. In contrast, our second approach uses ordinary arithmetic (instead of affine arithmetic) and a basic Euler scheme (instead of Runge-Kutta schemes). We neither need to estimate Lagrange remainders nor perform Picard iteration in combination with Taylor series. Our simple Euler-based approach is made possible by resorting to the notion of *one-sided Lipschitz* (OSL) function [57]. This allows us to bound directly the *global error*, *i.e.* the distance between the approximate point $\tilde{x}(t)$ computed by the Euler scheme and the exact solution $x(t)$ for all $t \geq 0$. Note that the bound we establish is more precise than the classical one found in [20], which is also used in hybridization methods in [18, 44]. An appropriate way to exploit this new bound is balls, and the formula established is available for all time in the switching period. It allows us to compute reachability tubes in an extremely fast way compared to Runge-Kutta methods, although it can lack accuracy for certain values of OSL constant.

None of the works of guaranteed integration above mentioned uses the Euler scheme nor the notion of one-sided Lipschitz constant. In the literature on symbolic integration, the Euler scheme with OSL conditions is explored in [57, 123]. Our approach is similar but establishes an *analytical* result for the global error of Euler's estimate rather than analyzing, in terms of complexity, the speed of convergence to zero, the accuracy and the stability of Euler's method. In the control literature, OSL conditions have been recently applied to control and stabilization [1, 39], but do not make use of Euler's method. To our knowledge, our work applies for the first time Euler's scheme with OSL conditions to the symbolic control of hybrid systems.

1.4 Compositional approaches

As mentioned earlier, the complexity of abstractions of switched systems by symbolic methods are subject to the curse of dimensionality. Actually, this exponential cost is twofold. On the first hand, the size of the abstractions grows exponentially with the dimension of the system. Indeed, most symbolic control methods rely on discretizations or tilings of the state-space. If we consider a system of dimension n , and if each dimension is discretized with m points or tiles, then the resulting number of symbols is in $O(m^n)$. On the other hand, the number of control sequences to be explored is exponential with the size of the sequences and depends on the number of switched modes. Actually, if N is the number of switched modes, the number of control sequences of length up to k is in $O(N^k)$.

It is therefore essential to design compositional analysis techniques in order to obtain control methods for switching systems with formal correctness guarantees. The aim is to split the system in smaller systems (components), and synthesize controllers for these sub-systems of smaller dimension. With simple techniques of over-approximation, it allows one component to estimate the symbolic states of the other components, in presence of partial information. This is similar in spirit to an *assume-guarantee* (or *contract-based*) reasoning, where the controller synthesis for each sub-system assumes that some safety properties are satisfied by the other sub-systems [11, 34, 53, 65, 100, 135, 161, 167]. This approach is a continuation of [65]. In contrast to [65], we do not need, for the mode selection of a sub-system, to blindly explore all the possible modes selected by the other sub-system. This yields a drastic reduction of the complexity. This approach allows us to treat a real case study, which is intractable using a centralized approach. This case study proposed by the Danish company Seluxit comes from [112], it models an eleven room house heated by geothermal energy. In contrast to the work of [112] which uses an on-line and heuristic approach with no formal guarantees, we use here an off-line formal method which guarantees reachability and stability properties.

This compositional approach is applied for linear systems using zonotopes, and for nonlinear systems using the Runge-Kutta and Euler based approaches. While the extension to nonlinear systems using the Runge-Kutta approach is nearly straightforward thanks to its handling of perturbations, the Euler based approach requires further developments. We explain how such an Euler-based method can be extended to synthesize safety controllers in a *distributed* manner. In order to perform such a distributed synthesis, we will see the components of the global systems as being *interconnected* (see, *e.g.*, [173]), and use (a less restrictive variant of) the notions of *incremental input-to-state stability (δ -ISS)* and *ISS Lyapunov functions* [96] instead of the notion of OSL used in the centralized framework.

1.5 Model order reduction methods

Model order reduction methods are aimed at representing the solutions of partial differential equations with few basis functions. They are extensively used in the field of structural and computational mechanics. Of course, with such methods, one loses a part of the information contained in the exact solution, and bounding the error between the reduced and full order models is mandatory to induce guaranteed control laws. One of the oldest methods might be the spectral decomposition [40], basically based on truncated Fourier decompositions, and which already allows to accurately represent solutions of a wide range of PDEs with reasonable amounts of basis functions. They present the advantage of being applicable *a priori*, i.e., without having to compute solutions of the PDE, and also come with various error bounds. More elaborated and accurate methods can rely on *a posteriori* model reduction, by extracting relevant information out of solution samples (*snapshots*). The idea is to perform a singular value decomposition on a matrix of snapshots, associated with a relevant normalization. The Proper Orthogonal Decomposition (POD) methods [48, 98] generally fit this type of methods. Although the construction of the basis functions can require a lot of time because of the need to compute snapshots, this type of approach is not prohibitive when using offline control syntheses. An important type of model reductions in structural mechanics is the one associated with Galerkin projections [28, 157], which allow to establish L^2 error bounds in a natural manner, and POD methods are often applied in this framework [107]. While all these approaches are applicable on a wide range of PDEs (excluding e.g. transport equations, which are still highly difficult to reduce), many nonlinear extensions of these methods have been proposed [23, 81, 162].

Even though the use of model reduction techniques is not classical when it comes to control systems, there are many works on the subject. The Gramian based approach (Gramians are, roughly speaking, functions that characterize energy of the state and output of the system, their computation generally requires finding solutions of Lyapunov equations) is for example used for switched systems in [165]. The balanced truncation [15, 29, 30, 140], a Gramian based approach quite close to the POD in spirit, allows to reduce linear high dimensional systems with outputs, and this technique is applied here to the case of discretized PDEs. There exist nonlinear versions [31, 110], but their application is often difficult on concrete case studies. For example [31] requires the computation of over-approximations of the generalized Gramians which are not computable in the general case. Note that interesting combinations of computational mechanics and control based approaches have been proposed, see for example [21] which proposes an application of the POD to infer reduced order controllers, or the works of [172] mixing Gramians and POD.

Our objective is to use such techniques to apply symbolic methods to PDEs, and the main issue to be dealt with is providing guaranteed controllers. This can be done

by appropriately bounding the error between the trajectories of the full and reduced order systems, and taking this bound into account in the synthesis. Of course, the choice of the reduction technique is not trivial and should be adapted to the PDE. The construction of a proper error bound highly depends on this previous choice.

1.6 Contributions

In Chapter 2, we first formally define the class of systems considered before introducing the algorithms used in the remainder of the thesis. These algorithms are highly inspired by those of [66, 68, 106], and simply extend them to continuous time properties. We however provide a non negligible improvement for the research of control sequences which highly reduces the computation times.

In Chapter 3, we consider the problem of reachability analysis. We first present the method used for linear systems in [68], and then introduce the method used in [5, 6, 56], which is mainly due to Julien Alexandre dit Sandretto and Alexandre Chapoutot. The application to nonlinear systems is however entirely new and provides competitive results with respect to the state-of-the-art tools. These works led to a conference paper [115] and an extended journal paper [116]. We finally present the Euler based method, which is an entirely novel approach and gives very promising results. It led to the conference paper [118].

In Chapter 4, we propose the compositional versions of the algorithms of Chapters 2 and 3. The synthesis procedure presented for linear discrete-time systems is an extension of [66, 68], which provides a new iterative backward filling of the state space. It is then applied with an over-approximation method allowing distributed computations, which allowed to synthesize a controller for a system of dimension eleven. This is, to our knowledge, the first time that a system of such dimension is handled with formal methods. These works have been published in a conference paper [120] and submitted in an extended version [121]. The extension to nonlinear systems with continuous time properties is made possible with the use of validated simulation. We then present a distributed version of the Euler method approach, relying on weaker variants of δ -ISS properties. These works led to the conference paper [114].

In Chapter 5, we present a symbolic approach for the control of discretized PDEs, relying on the balanced truncation. We give two possible procedures for application of the control, and propose some initiating works towards partial observation with the use of reduced order observers. This approach has been published in [117] and applied more generally to mechanical systems in [119].

In Chapter 6, we introduce a first possible approach for the control of undiscretized PDEs, relying on a spectral decomposition and an interpolation method particularly efficient for representing continuous functions with few basis functions due to [129]. We give a second approach aimed at using Galerkin projections for

the reduction and the Euler based method. It provides a guaranteed L^2 control for a coupled ODE-PDE system, thanks to an appropriate error bounding and decomposition of the terms involved in the solution. This approach is very promising but might require further developments in order to be applied to a wider range of systems.

Chapter 2

Switched systems

2.1 Introduction

In this chapter, we introduce the class of systems we are interested in, and present the principles of the algorithms we use, as well as some results on the computational costs, highlighting the need of further developments for widening the types of systems supported by the method. Most of the algorithms presented here are based on the works of Romain Soulat and Laurent Fribourg [67–69, 169]. They provide algorithms allowing to synthesize *state-dependent* controllers ensuring discrete time properties, they are based on an adaptive tiling of the state-space. We extend this approach to ensuring continuous time properties, and present the different types of heuristics and sets which can be used with this method. We also give a new algorithm for the research of admissible control sequences. Although being theoretically of the same complexity, it drastically lowers the computation times in practice. The class of systems considered is presented in Section 2.2, and we give the adaptations of the algorithms of [68] in Section 2.3. We then present the improved research of admissible controls in Section 2.4, and conclude with the computational cost of the method in Section 2.5.

2.2 Switched systems

We are interested in continuous-time switched systems subject to disturbances, described by the set of nonlinear ordinary differential equation:

$$\dot{x} = f_j(x, d), \quad (2.1)$$

where $x \in \mathbb{R}^n$ is the state of the system, $j \in U$ is the mode of the system, and $d \in \mathbb{R}^m$ is a bounded perturbation. The finite set $U = \{1, \dots, N\}$ is the set of switching modes of the system. The functions $f_j : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, with $j \in U$, are the vector fields describing the dynamics of each mode j of the system. The system can be in only one mode at a time. Such systems can be schemed as in Figure 2.1, where we have several working modes for a system, and one has to choose which working mode j is active within time, in order to ensure some properties for the state x . A supervisor applies a switching rule deciding when to change the working mode, which one should be applied next.

We focus on sampled switched systems: given a sampling period $\tau > 0$, switchings will occur periodically at times $\tau, 2\tau, \dots$. A switching rule $\sigma(\cdot) : \mathbb{R}^+ \rightarrow U$ associates to each time $t > 0$ the active mode $j \in U$. A switched system is thus a dynamical system with piecewise dynamics, and the switching rule selects which mode is active. The switching rule is thus piecewise constant. Given a switching rule $\sigma(\cdot) : \mathbb{R}^+ \rightarrow U$, and a perturbation $w(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}^m$, we will denote by $\phi(t; t_0, x_0, \sigma, w)$ the state reached by the system at time $t > t_0$, from the initial state $x_0 \in \mathbb{R}^n$ at time $t_0 \geq 0$, and under control input and perturbation σ and

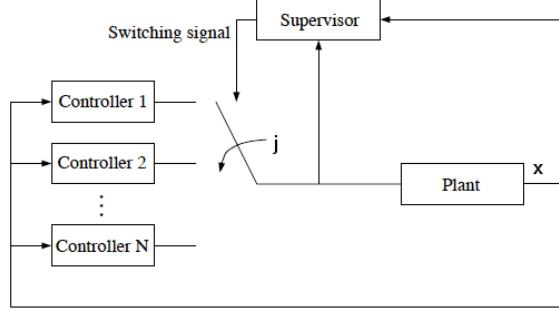


Figure 2.1: Scheme of a switched system.

w respectively. For a given control $\sigma(\cdot)$ and perturbation $w(\cdot)$, we will often refer to function ϕ as the solution of equation (2.1). Note that for a given $w(\cdot)$ such that $f_j(\cdot, w(\cdot))$ is continuous with respect to both variables and locally Lipschitz with respect to the first variable, the existence and uniqueness of ϕ is given by the Cauchy-Lipschitz theorem. In a more general case, we will just suppose that σ and w are such that ϕ exists and is continuous with respect to time. One can note that this notion of solution differs from the classical (mathematical) definition of the solution of a differential equation.

Often, we will consider $\phi(t; t_0, x^0, \sigma, w)$ on the interval $0 \leq t < \tau$ for which $\sigma(t)$ is equal to a constant, say $j \in U$. In this case, we will abbreviate $\phi(t; t_0, x^0, \sigma, w)$ as $\phi_j(t; t_0, x^0, w)$. We will also consider $\phi(t; t_0, x^0, \sigma, w)$ on the interval $0 \leq t < k\tau$ where k is a positive integer, and $\sigma(t)$ is equal to a constant, say $j_{k'}$, on each interval $[(k'-1)\tau, k'\tau]$ with $1 \leq k' \leq k$; in this case, we will abbreviate $\phi(t; t_0, x^0, \sigma, w)$ as $\phi_\pi(t; t_0, x^0, w)$, where π is a sequence of k modes, also denoted as a control pattern (pattern for short), of the form $\pi = j_1 \cdot j_2 \cdot \dots \cdot j_k \in U^k$.

We will assume that $\phi(\cdot; 0, x_0, \sigma, w)$ is *continuous* at time $k\tau$ for all positive integer k (assuming that $t_0 = 0$ for the sake of simplicity). This means that there is no “reset” at time $k'\tau$ ($1 \leq k' \leq k$); the value of $\phi(t; t_0, x^0, \sigma, w)$ for $t \in [(k'-1)\tau, k'\tau]$ corresponds to the solution of $\dot{x}(u) = f_{\sigma((k'-1)\tau+u)}(x(u), w(u))$ for $u \in [0, \tau]$ with initial value $\phi((k'-1)\tau; t_0, x^0, \sigma, w)$.

Given a “recurrence set” $R \subset \mathbb{R}^n$ and a “safety set” $S \subset \mathbb{R}^n$ which contains R ($R \subseteq S$), we are interested in the synthesis of a control such that: starting from any initial point $x \in R$, the controlled trajectory always returns to R within a bounded time while never leaving S . We suppose that sets R and S are compact. Furthermore, we suppose that S is convex.

This is formalized as follows, note that Problem 1 is the continuous time version of the control problem considered in [67]:

Problem 1 ((R, S)-Stability problem). *Given a switched system of the form (2.1), a recurrence set $R \subset \mathbb{R}^n$ and a safety set $S \subset \mathbb{R}^n$, find a control rule $\sigma : \mathbb{R}^+ \rightarrow U$*

such that, for any initial condition $x_0 \in R$ and any perturbation $w : \mathbb{R}^+ \rightarrow U$, the following holds:

- *Recurrence in R :* there exists a monotonically strictly increasing sequence of (positive) integers $\{k_l\}_{l \in \mathbb{N}}$ such that for all $l \in \mathbb{N}$, $\phi(k_l \tau; t_0, x^0, \sigma, w) \in R$
- *Stability in S :* for all $t \in \mathbb{R}^+$, $\phi(t; t_0, x^0, \sigma, w) \in S$

We also define a similar problem for reachability from a set $R_1 \subset \mathbb{R}^n$ to a set $R_2 \subset \mathbb{R}^n$, where both R_1 and R_2 are subsets of $S \subseteq \mathbb{R}^n$.

Problem 2 $((R_1, R_2, S)$ -Reachability problem). *Given a switched system of the form (2.1), two sets $R_1 \subset \mathbb{R}^n$, and $R_2 \subset \mathbb{R}^n$, and a safety set $S \subset \mathbb{R}^n$, find a control rule $\sigma : \mathbb{R}^+ \rightarrow U$ such that, for any initial condition $x_0 \in R_1$ and any perturbation $w : \mathbb{R}^+ \rightarrow U$, the following holds:*

- *Reachability from R_1 to R_2 :* there exists an integer $k \in \mathbb{N}_{>0}$ such that we have $\phi(k \tau; t_0, x^0, \sigma, w) \in R_2$
- *Stability in S :* for all $t \in \mathbb{R}^+$, $\phi(t; t_0, x^0, \sigma, w) \in S$

Another interesting problem is the avoid problem, where one has to ensure (R, S) -stability while avoiding an obstacle, given as a set B .

Problem 3 $((R, B, S)$ -Avoid problem). *Given a switched system of the form (2.1), and given three sets $R \subset \mathbb{R}^n$, $S \subset \mathbb{R}^n$, and $B \subset \mathbb{R}^n$, with $R \cup B \subset S$ and $R \cap B = \emptyset$, find a rule $\sigma : \mathbb{R}^+ \rightarrow U$ such that, for any initial condition $x_0 \in R$ and any perturbation $w : \mathbb{R}^+ \rightarrow U$, the following holds:*

- *Recurrence in R :* there exists a monotonically strictly increasing sequence of (positive) integers $\{k_l\}_{l \in \mathbb{N}}$ such that for all $l \in \mathbb{N}$, $\phi(k_l \tau; t_0, x^0, \sigma, w) \in R$
- *Stability in S :* for all $t \in \mathbb{R}^+$, $\phi(t; t_0, x^0, \sigma, w) \in S$
- *Avoid B :* for all $t \in \mathbb{R}^+$, $\phi(t; t_0, x^0, \sigma, w) \notin B$.

In the rest of this chapter, we focus on solving Problem 1 of synthesizing controllers for (R, S) -stability for systems of the form (2.1). Note that solving Problem 2 can be done in a very similar manner (see for example Chapter 3). As a matter of fact, we will not look for *time dependent* switching rules $\sigma : \mathbb{R}^+ \rightarrow U$ returning the mode to be applied for a given time, but rather look for *state-dependent* switching rules which, for every state x of the system, return a pattern $\pi \in U^k$ to be applied in the next time interval $[t, t + k\tau]$. The set of admissible state-dependent control laws is thus $\{\tilde{\sigma} : \mathbb{R}^n \rightarrow U^k \text{ for } k \in \mathbb{N}\}$. Such laws can be computed *offline*.

Under the above-mentioned notation, we propose the main procedure of our approach which solves this problem by constructing a state-dependent law $\tilde{\sigma}(\cdot)$, such that for all $x_0 \in R$, and under the unknown bounded perturbation w , there exists $\pi = \tilde{\sigma}(x_0) \in U^k$ for some k such that:

$$\begin{cases} \phi_\pi(t_0 + k\tau; t_0, x_0, w) \in R, \\ \forall t \in [t_0, t_0 + k\tau], \quad \phi_\pi(t; t_0, x_0, w) \in S. \end{cases}$$

Such a law permits to perform an infinite-time state-dependent control. The synthesis algorithm is described in Section 2.3.1 and involves guaranteed set-based integration presented in the next chapter. Before presenting the algorithms, we introduce some definitions abstracting the set-based integration.

Definition 1 (Post operator). *Let $X \subset \mathbb{R}^n$ be a box of the state space. Suppose perturbation w lies in a compact $D \subset \mathbb{R}^m$. Let $\pi = (i_1, i_2, \dots, i_k) \in U^k$. The successor set of X via π , denoted by $Post_\pi(X)$, is the (over-approximation of the) image of X induced by application of the pattern π , i.e., the solution at time $t = k\tau$ of*

$$\begin{aligned} \dot{x}(t) &= f_{\sigma(t)}(x(t), w(t)), \\ x(0) &= x_0 \in X, \\ \forall t \geq 0, \quad w(t) &\in D, \\ \forall j \in \{1, \dots, k\}, \quad \sigma(t) &= i_j \in U \text{ for } t \in [(j-1)\tau, j\tau]. \end{aligned} \tag{2.2}$$

Note that D is absent from the notation $Post_\pi(X)$. When it is relevant, we will rather use the notation $Post_\pi^D(X)$ to clarify where the perturbation lies. The *Post* operator can also be defined, when the perturbation is omitted, as

$$Post_\pi(X) = \bigcup_{x_0 \in X} \phi_\pi(t; t_0, x_0).$$

With a bounded perturbation $w : \mathbb{R}^+ \rightarrow D$, it can be defined as:

$$Post_\pi^D(X) = \bigcup_{x_0 \in X} \bigcup_{w \in D^{\mathbb{R}^+}} \phi_\pi(t; t_0, x_0, w).$$

In a set-based computation application, the perturbation is just defined by the whole set D at every time $t \in \mathbb{R}^+$.

Definition 2 (Tube operator). *Let $X \subset \mathbb{R}^n$ be a box of the state space. Suppose perturbation w lies in a compact $D \subset \mathbb{R}^m$. Let $\pi = (i_1, i_2, \dots, i_k) \in U^k$. We denote by $Tube_\pi(X)$ the union of the trajectories of IVP (2.2), i.e.:*

$$Tube_\pi(X) = \bigcup_{t \in [0, k\tau]} \bigcup_{x_0 \in X} \bigcup_{w \in D^{\mathbb{R}^+}} \phi_\pi(t; t_0, x_0, w).$$

In the same manner as the Post operator, we will use the notation $Tube_\pi^D(X)$ when it is relevant. An illustration of these definitions is shown in Figure 2.2, the Post and Tube operators are computed numerically on a case-study described in Chapter 3. It is applied to the synthesis of an (R_1, R_2, S) -reachability controller.

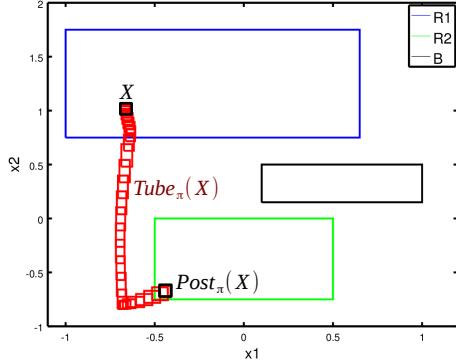


Figure 2.2: Functions $Post_\pi(X)$ and $Tube_\pi(X)$ for the initial box $X = [-0.69, -0.64] \times [1, 1.06]$, with a pattern $\pi = (1, 3, 0)$.

2.3 General principle

We introduce a first basic procedure permitting to perform (R, S) -stability, and omit the perturbation in a first time. Given a set R , let $\{W_i\}_{i \in I}$ be a family of sets such that $R \subseteq \bigcup_{i \in I} W_i \subseteq S$ as illustrated in Figure 2.3 (a). If one can find, for each W_i for $i \in I$, a pattern π_i such that $Post_{\pi_i}(W_i) \subseteq R$, then we can induce an infinite-time switching rule permitting to return infinitely often in R (such a pattern is illustrated for W_1 in Figure 2.3 (b)).

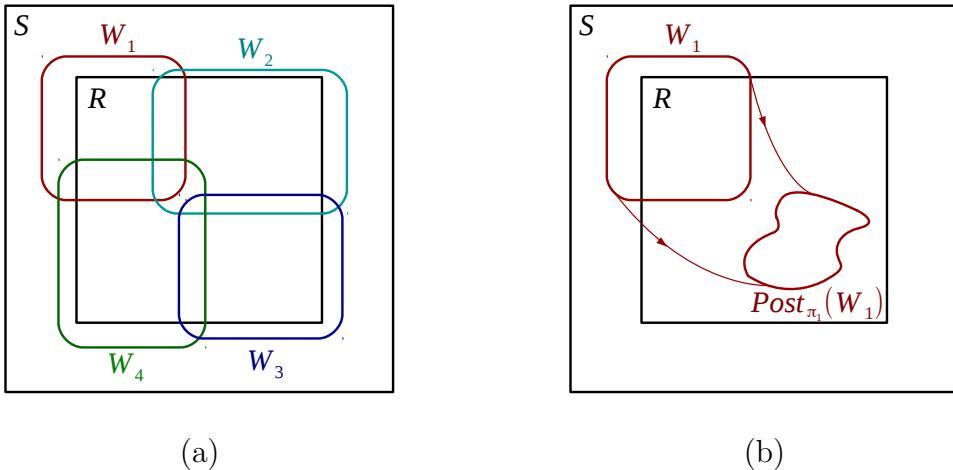


Figure 2.3: (a): A family of sets $\{W_i\}_{i=1,\dots,4}$ covering R ; (b): a pattern π_1 such that $Post_{\pi_1}(W_1) \subset R$.

Theorem 1. Let $R \subseteq \mathbb{R}^n$, suppose we are given a switched system satisfying (2.1).

A family of sets $\{W_i\}_{i \in I}$ associated to patterns $\{\pi_i\}_{i \in I}$ such that

- $R \subseteq \bigcup_{i \in I} W_i \subseteq S$
- for all $i \in I$, $Post_{\pi_i}(W_i) \subseteq R$

induces an infinite-time control ensuring recurrence in R .

Proof. Let $x_0 \in R$, there exists $i_0 \in I$ such that $x_0 \in W_{i_0}$ since $R \subseteq \bigcup_{i \in I} W_i$. Application of pattern π_{i_0} leads to a state $x_1 = \phi(\tau; 0, x_0, \pi_{i_0})$ also belonging to

R since $\text{Post}_{\pi_{i_0}}(W_{i_0}) \subseteq R$. State x_1 thus belongs to W_{i_1} for some $i_1 \in I$, and by recurrence, one can obtain a sequence of points x_0, x_1, \dots all belonging to R . The induced trajectory thus returns infinitely often in R . \square

A simple extension of this procedure, relying on the computation of reachability tubes, allows to ensure safety in $S \subseteq \mathbb{R}^n$ as follows.

Theorem 2. *Let $R \subseteq \mathbb{R}^n$, $S \subseteq \mathbb{R}^n$, suppose we are given a switched system satisfying (2.1). A family of sets $\{W_i\}_{i \in I}$ associated to patterns $\{\pi_i\}_{i \in I}$ such that*

- $R \subseteq \bigcup_{i \in I} W_i \subseteq S$
- for all $i \in I$, $\text{Post}_{\pi_i}(W_i) \subseteq R$
- for all $i \in I$, $\text{Tube}_{\pi_i}(W_i) \subseteq S$

induces an infinite-time control ensuring recurrence in R and safety in S .

Proof. The recurrence in R is proved with the same arguments as the proof of Theorem 1. The safety in S is ensured by the definition of $\text{Tube}_{\pi_i}(W_i)$, which permits to ensure that for all $x_0 \in R$, $i \in I$, $t \in k_i \tau$, where k_i is the length of pattern π_i , we have $\phi(t; 0, x_0, \pi_i) \in S$. \square

Having defined the principle of the procedure, we now present how controllers can be numerically computed using Theorem 1 and 2. At this point, two main problems arise. The first is the construction of a family $\{W_i\}_{i \in I}$ covering R , the second is ensuring that for all $i \in I$, $\text{Post}_{\pi_i}(W_i) \subseteq R$ and $\text{Tube}_{\pi_i}(W_i) \subseteq S$. The first problem can be solved using heuristics, but depends of the type of sets one uses, the second is actually impossible to ensure exactly, in the sense that solutions of ODEs are not known in general (particularly when the initial condition is a set). Supposing that one can compute reachability sets and tubes, the procedure works as follows in practice. First, we generate a coarse covering of R (starting for example by considering the whole set R), we then try to compute patterns associated to each set of the covering. If this last step fails, we generate another finer tiling, performing for example a bisection of each dimension of R , and one now has to control each bisected part of R . This is a simple heuristics, but which works well in practice (as seen in the following Chapters). In the following, we use a uniform covering of R with boxes and balls of \mathbb{R}^n . If each box or ball is controlled, the problem is solved, otherwise, we use a finer covering. We address the problem of computing reachability sets and tubes in the following chapters. We now present in details the possible heuristics and associated algorithms for control synthesis, supposing that one can compute the Post and Tube operators.

2.3.1 The state-space bisection algorithm

We describe the algorithm solving the control synthesis problem for nonlinear switched systems (see Problem 3, Section 2.2). Given the input boxes R , S , B , and

given two positive integers K and D , the algorithm provides, when it succeeds, a decomposition Δ of R of the form $\{V_i, \pi_i\}_{i \in I}$, with the properties:

- $\bigcup_{i \in I} V_i = R$,
- $\forall i \in I, Post_{\pi_i}(V_i) \subseteq R$,
- $\forall i \in I, Tube_{\pi_i}(V_i) \subseteq S$,
- $\forall i \in I, Tube_{\pi_i}(V_i) \cap B = \emptyset$.

The sub-boxes $\{V_i\}_{i \in I}$ are obtained by repeated bisection. At first, function *Decomposition* calls sub-function *Find_Pattern* which looks for a pattern π of length at most K such that $Post_\pi(R) \subseteq R$, $Tube_\pi(R) \subseteq S$ and $Tube_\pi(R) \cap B = \emptyset$. If such a pattern π is found, then a uniform control over R is found (see Figure 2.4(a)). Otherwise, R is divided into two sub-boxes V_1 , V_2 , by bisecting R w.r.t. its longest dimension. Patterns are then searched to control these sub-boxes (see Figure 2.4(b)). If for each V_i , function *Find_Pattern* manages to get a pattern π_i of length at most K verifying $Post_{\pi_i}(V_i) \subseteq R$, $Tube_{\pi_i}(V_i) \subseteq S$ and $Tube_{\pi_i}(V_i) \cap B = \emptyset$, then it is a success and algorithm stops. If, for some V_j , no such pattern is found, the procedure is recursively applied to V_j . It ends with success when every sub-box of R has a pattern verifying the latter conditions, or fails when the maximal degree of decomposition D is reached. The algorithmic form of functions *Decomposition* and *Find_Pattern*, adapted from [68], are given in Algorithm 1 and Algorithm 2 respectively. Note that a special form of Algorithm 2 for linear ODEs can be found in [67].

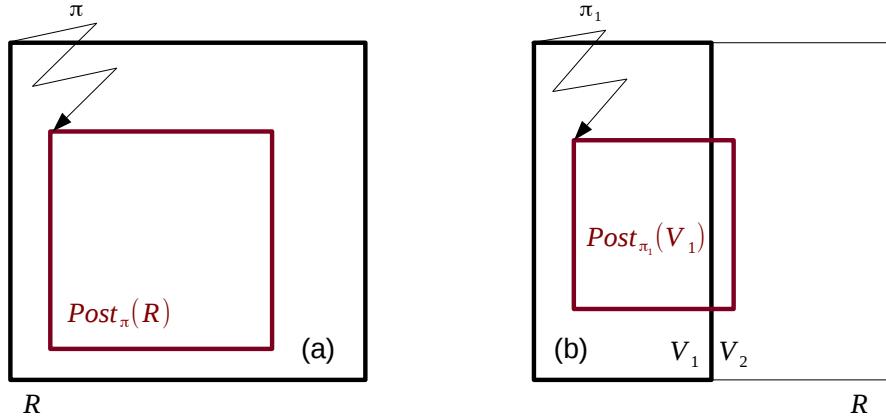


Figure 2.4: Principle of the bisection method.

Our control synthesis method being well defined, we introduce the main result of this section (initially formalized in [67]), stated as follows:

Proposition 1. *Algorithm 1 with input (R, R, S, B, D, K) returns, when it successfully terminates, a decomposition $\{V_i, \pi_i\}_{i \in I}$ of R which solves Problem 3.*

Proof. Let $x_0 = x(t_0 = 0)$ be an initial condition belonging to R . If the decomposition has terminated successfully, we have $\bigcup_{i \in I} V_i = R$, and x_0 thus belongs to V_{i_0}

Algorithm 1 Algorithmic form of Function *Decomposition*.

Function: *Decomposition*(W, R, S, B, D, K)

Input: A box W , a box R , a box S , a box B , a degree D of bisection, a length K of input pattern

Output: $\langle \{(V_i, \pi_i)\}_i, \text{True} \rangle$ or $\langle _, \text{False} \rangle$

```

 $(\pi, b) := \text{Find\_Pattern}(W, R, S, B, K)$ 
if  $b = \text{True}$  then
    return  $\langle \{(W, Pat)\}, \text{True} \rangle$ 
else
    if  $D = 0$  then
        return  $\langle \_, \text{False} \rangle$ 
    else
        Divide equally  $W$  into  $(W_1, W_2)$ 
        for  $i = 1, 2$  do
             $(\Delta_i, b_i) := \text{Decomposition}(W_i, R, S, B, D - 1, K)$ 
        end for
        return  $(\bigcup_{i=1,2} \Delta_i, \bigwedge_{i=1,2} b_i)$ 
    end if
end if

```

for some $i_0 \in I$. We can thus apply the pattern π_{i_0} associated to V_{i_0} . Let us denote by k_0 the length of π_{i_0} . We have:

- $\phi_{\pi_{i_0}}(k_0\tau; 0, x_0, d) \in R$
- $\forall t \in [0, k_0\tau], \phi_{\pi_{i_0}}(t; 0, x_0, d) \in S$
- $\forall t \in [0, k_0\tau], \phi_{\pi_{i_0}}(t; 0, x_0, d) \notin B$

Let $x_1 = \phi_{\pi_{i_0}}(k_0\tau; 0, x_0, d) \in R$ be the state reached after application of π_{i_0} and let $t_1 = k_0\tau$. State x_1 belongs to R , it thus belongs to V_{i_1} for some $i_1 \in I$, and we can apply the associated pattern π_{i_1} of length k_1 , leading to:

- $\phi_{\pi_{i_1}}(t_1 + k_1\tau; t_1, x_1, d) \in R$
- $\forall t \in [t_1, t_1 + k_1\tau], \phi_{\pi_{i_1}}(t; t_1, x_1, d) \in S$
- $\forall t \in [t_1, t_1 + k_1\tau], \phi_{\pi_{i_1}}(t; t_1, x_1, d) \notin B$

We can then iterate this procedure from the new state

$$x_2 = \phi_{\pi_{i_1}}(t_1 + k_1\tau; t_1, x_1, d) \in R.$$

This can be repeated infinitely, yielding a sequence of points belonging to R x_0, x_1, x_2, \dots attained at times t_0, t_1, t_2, \dots , when the patterns $\pi_{i_0}, \pi_{i_1}, \pi_{i_2}, \dots$ are applied.

We furthermore have that all the trajectories stay in S and never cross B :

$$\forall t \in \mathbb{R}^+, \exists k \geq 0, t \in [t_k, t_{k+1}]$$

and

$$\forall t \in [t_k, t_{k+1}], \phi_{\pi_{i_k}}(t; t_k, x_k, d) \in S, \phi_{\pi_{i_k}}(t; t_k, x_k, d) \notin B.$$

The trajectories thus return infinitely often in R , while always staying in S and never crossing B . \square

Remark 1. Note that it is possible to perform reachability from a set R_1 to another set R_2 by computing $\text{Decomposition}(R_1, R_2, S, B, D, K)$. The set R_1 is thus decomposed with the objective to send its sub-boxes into R_2 , i.e., for a sub-box V of R_1 , patterns π are searched with the objective $\text{Post}_\pi(V) \subseteq R_2$ (see Example 3.2.2).

Algorithm 2 Algorithmic form of Function *Find_Pattern*.

Function: *Find_Pattern*(W, R, S, B, K)

Input: A box W , a box R , a box S , a box B , a length K of input pattern

Output: $\langle \pi, \text{True} \rangle$ or $\langle _, \text{False} \rangle$

```

for  $i = 1 \dots K$  do
     $\Pi :=$  set of input patterns of length  $i$ 
    while  $\Pi$  is non empty do
        Select  $\pi$  in  $\Pi$ 
         $\Pi := \Pi \setminus \{\pi\}$ 
        if  $\text{Post}_\pi(W) \subseteq R$  and  $\text{Tube}_\pi(W) \subseteq S$  and  $\text{Tube}_\pi(W) \cap B = \emptyset$  then
            return  $\langle \pi, \text{True} \rangle$ 
        end if
    end while
end for
return  $\langle \_, \text{False} \rangle$ 

```

In Algorithms 1 and 2, we use a bisection of uncontrolled tiles into two parts (by bisecting the greatest dimension). But another possible heuristics is to divide uncontrolled parts into 2^n parts, by bisecting each dimension (i.e. replacing “Divide equally W into (W_1, W_2) ” by “Divide equally W into (W_1, \dots, W_{2^n}) ” in Algorithm 1). This leads to a faster growing of the number of tiles to be controlled, but can sometimes lead to lower computation times, when the system requires a fine tiling. The two possible heuristics are schemed in Figure 2.5.

2.3.2 A covering of balls

So far, we used boxes of \mathbb{R}^n to represent sets of states. Balls of \mathbb{R}^n are actually another useful way of representing it, since we provide an efficient way of performing reachability analysis with such sets (see Chapter 3). A covering of R can be

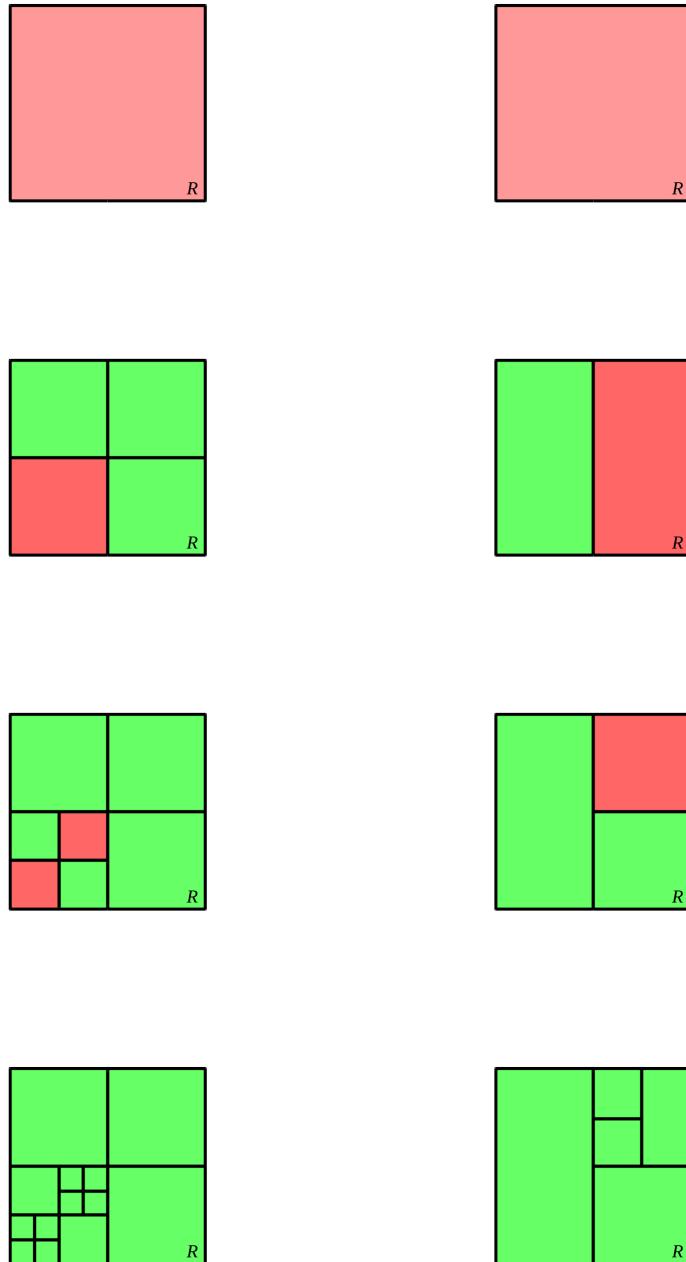


Figure 2.5: Scheme of the two possible heuristics: green tiles have been controlled (associated to a pattern), and red tiles have yet to be controlled and bisected. Left: bisection of all the dimensions; right: bisection of the largest dimension

performed as schemed in Figure 2.6. Let δ be a radius, each set $W_i = B(\tilde{x}_i, \delta)$ has to be controlled, otherwise, a finer covering (using more balls) should be used. Actually, the same heuristics as boxes could be used, since these balls can be built as circumscribed balls of the boxes.

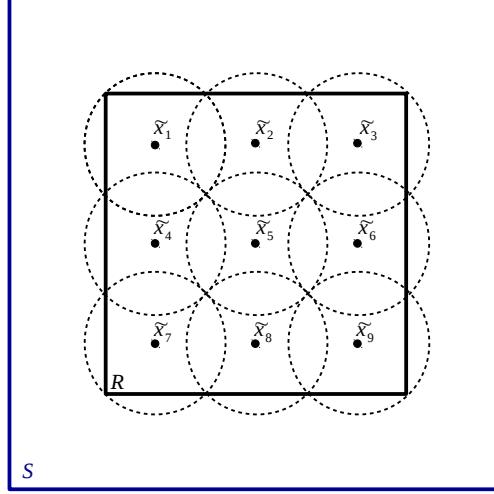


Figure 2.6: Scheme of a covering of $R \subset \mathbb{R}^2$ with balls.

2.4 Improving the research of patterns

We propose in this section an improvement of the function *Find_Pattern* given in [67], which is a naive testing of all the patterns of growing length (up to K).

The improved function, denoted here by *Find_Pattern2*, exploits heuristics to prune the search tree of patterns. We present it with boxes of \mathbb{R}^n , but can also be used with balls. The algorithmic form of *Find_Pattern2* is given in Algorithm 3. It relies on a new data structure consisting of a list of triplets containing:

- An initial box $V \subset \mathbb{R}^n$,
- A *current* box $Post_\pi(V)$, image of V by the pattern π ,
- The associated pattern π .

For any element e of a list of this type, we denote by $e.Y_{\text{init}}$ the initial box, $e.Y_{\text{current}}$ the *current* box, and by $e.\Pi$ the associated pattern. We denote by $e_{\text{current}} = \text{takeHead}(\mathcal{L})$ the element on top of a list \mathcal{L} (this element is removed from list \mathcal{L}). The function $\text{putTail}(\cdot, \mathcal{L})$ adds an element at the end of the list \mathcal{L} .

Let us suppose one wants to control a box $X \subseteq R$. The list \mathcal{L} of Algorithm 3 is used to store the intermediate computations leading to possible solutions (patterns sending X in R while never crossing B or $\mathbb{R}^n \setminus S$). It is initialized as $\mathcal{L} = \{(X, X, \emptyset)\}$. First, a testing of all the control modes is performed (a set simulation starting from X during time τ is computed for all the modes in U). The first level of branches is thus tested exhaustively. If a branch leads to crossing B or $\mathbb{R}^n \setminus S$, the branch is cut. Indeed, no following branch can be accepted if a previous one crosses B . Otherwise, either a solution is found or an intermediate state is added to \mathcal{L} . The next level of branches (patterns of length 2) is then explored from branches that are not cut. And so on iteratively. At the end, either the tree is explored up to level K (avoiding the cut branches), or all the branches have been cut at lower levels. List \mathcal{L} is thus

of the form $\{(X, Post_{\pi_i}(X), \pi_i)\}_{i \in I_X}$, where for each $i \in I_X$ we have $Post_{\pi_i}(X) \subseteq S$ and $Tube_{\pi_i}(X) \cap B = \emptyset$. Here, I_X is the set of indices associated to the stored intermediate solutions, $|I_X|$ is thus the number of stored intermediate solutions for the initial box X . The number of stored intermediate solutions grows as the search tree of patterns is explored, then decreases as solutions are validated, branches are cut, or the maximal level K is reached.

The storage of the intermediate solutions $Post_{\pi_i}(X)$ allows to reuse the computations already performed. Even if the search tree of patterns is visited exhaustively, it already allows to obtain much better computation times than with Function *Find_Pattern*.

A second list, denoted by *Sol* in Algorithm 3, is used to store the validated patterns associated to X , *i.e.*, a list of patterns of the form $\{\pi_j\}_{j \in I'_X}$, where for each $j \in I'_X$ we have $Post_{\pi_j}(X) \subseteq R$, $Tube_{\pi_j}(X) \cap B = \emptyset$ and $Tube_{\pi_j}(X) \subseteq S$. Here, I'_X is the set of indices associated to the stored validated solutions, $|I'_X|$ is thus the number of stored validated solutions for the initial box X . The number of stored validated solutions can only increase, and we hope that at least one solution is found, otherwise, the initial box X is split in two sub-boxes.

Remark that several solutions can be returned by *Find_Pattern2*, so further optimizations could be performed, such as returning the pattern minimizing a given cost function. In practice, and in the examples given below, we return the first validated pattern and stop the computation as soon as it is obtained (see commented line in Algorithm 3). Compared to [67], this new function highly improves the computation times, even though the complexity of the two functions is theoretically the same, at most in $O(N^K)$. A comparison between functions *Find_Pattern* and *Find_Pattern2* is given in Section 3.2.3.

2.5 Computational cost

The computational cost of the synthesis method depends on the heuristics, but in every case, if M is the number of sets used to cover R , N is the number of switched modes, and k is the maximal length of explored control patterns, then the computational complexity is in $O(MN^k)$ (see [68]). Note that in practice, M grows exponentially with the dimension n of the system. Indeed, using the adaptive box bisection heuristics, if D is the maximal depth of bisection, using the bisection of each dimension, we have a complexity in $O(2^{nD})N^k$. Using a uniform tiling, by dividing each dimension in p , we get a complexity in $O(p^n N^k)$. We thus see that the computation cost is exponential with the dimension, but also with the length of the patterns and number of modes, and this has to be multiplied by the cost of reachability computations. We thus see two aspects have to be dealt with to improve the efficiency of the method: the dimension, and the reachability computations. We will thus present in Chapter 3 methods to perform reachability analysis in the most

accurate and fast possible ways (note that there is a tradeoff to make between accuracy and speed). In the following chapters, we propose methods to extend the approach to systems of greater dimensions, by using

- compositional approaches: dividing a system into several sub-systems of lower dimension (see Chapter 4)
- model order reduction: approximating a high dimensional system with a lower dimensional one (see Chapter 5 and 6)

Of course, these two last approaches introduce new issues: accuracy of the models, efficiency of the induced control laws for the original system...

2.6 Final remarks

We have now introduced the class of systems considered in this thesis and the main ideas of the control synthesis method for switched systems represented by ODEs. In order to complete the method, what remains to be studied first is the computation of the *Post* and *Tube* operators, this is tackled in Chapter 3. However, as mentioned above, the computational complexity is still a very limiting factor for the application to systems of greater dimensions, and we thus propose distributed versions of the algorithms presented here in Chapter 4, and reduced order approaches in Chapters 5 and 6.

Algorithm 3 Algorithmic form of Function *Find_Pattern2*.

Function: *Find_Pattern2*(*W, R, S, B, K*)

Input: A box *W*, a box *R*, a box *S*, a box *B*, a length *K* of input pattern**Output:** $\langle \pi, \text{True} \rangle$ or $\langle _, \text{False} \rangle$

```
Sol = { $\emptyset$ }
L = {(W, W,  $\emptyset$ )}
while L  $\neq \emptyset$  do
    ecurrent = takeHead(L)
    for i  $\in U$  do
        if Posti(ecurrent.Ycurrent)  $\subseteq R$  and Tubei(ecurrent.Ycurrent)  $\cap B = \emptyset$  and
            Tubei(ecurrent.Ycurrent)  $\subseteq S$  then
                putTail(Sol, ecurrent.Π + i) /* or also “return  $\langle e_{\text{current}}.\Pi + i, \text{True} \rangle$ ” */
            else
                if Tubei(ecurrent.Ycurrent)  $\cap B \neq \emptyset$  or Tubei(ecurrent.Ycurrent)  $\not\subseteq S$  then
                    discard ecurrent
                end if
            else
                if Tubei(ecurrent.Ycurrent)  $\cap B = \emptyset$  and Tubei(ecurrent.Ycurrent)  $\subseteq S$  then
                    if Length(Π) + 1 < K then
                        putTail(L, (ecurrent.Yinit, Posti(ecurrent.Ycurrent), ecurrent.Π + i))
                    end if
                end if
            end if
        end if
    end for
end while
return  $\langle \_, \text{False} \rangle$  if no solution is found, or  $\langle \pi, \text{True} \rangle$ ,  $\pi$  being any pattern
validated in Solution.
```

Chapter 3

Reachable set computation

In this chapter, we present practical ways to compute the Post and Tube operators when sets are represented with boxes or balls. We first give some results for linear systems. We then present approaches relying on Runge-Kutta schemes, allowing to compute accurately images of box sets for nonlinear ODEs. We then introduce some hypotheses to use a simple Euler scheme, associated to a new error bound, permitting to compute the Post and Tube operators for balls in a very fast way, even though the accuracy can fall down in some cases. We present the approach for linear systems in Section 3.1, we then introduce the Runge-Kutta approach in Section 3.2, and we finally present the Euler scheme for balls in Section 3.3.

3.1 Zonotopes and linear systems

Let us first introduce *zonotopes*, a type of symmetrical polytopes, allowing to represent efficiently boxes of \mathbb{R}^n , and thus very useful for performing tilings of the state-space. Furthermore, there exist multiple ways to compute images of zonotopes by linear or nonlinear transformations.

Definition 3. A zonotope is a set:

$$Z = \{x \in \mathbb{R}^n : x = c + \sum_{i=1}^p \beta^{(i)} g^{(i)}, -1 \leq \beta^{(i)} \leq 1\}$$

with $c, g^{(1)}, \dots, g^{(p)} \in \mathbb{R}^n$.

The vectors $g^{(1)}, \dots, g^{(p)}$ are referred to as the *generators* and c as the center of a zonotope. A zonotope is thus a symmetric polytope in dimension n . It is convenient to represent the set of generators as an $n \times p$ matrix G , of columns $g^{(1)}, \dots, g^{(p)}$. The notation is $Z = \langle c, G \rangle$. Note that if G is an $n \times n$ diagonal matrix, then the zonotope Z is a box of \mathbb{R}^n .

Given a zonotope $\langle c, G \rangle$, the transformation of Z via an affine function $x \rightarrow Cx + d$ is a zonotope of the form $\langle Cc + d, CG \rangle$. More information and properties on zonotopes can be found in [10, 73, 105]. Besides, being given a linear switched system satisfying

$$\dot{x} = A_j x + b_j,$$

and an initial condition $x_0 \in \mathbb{R}^n$ at time $t = 0$, if mode $j \in U$ is applied on $[0, \tau]$, then the solution at time $t = \tau$ is given by

$$\phi(t; 0, x_0, j) = e^{A_j \tau} x_0 + \int_0^\tau e^{A_j(t-s)} b_j ds. \quad (3.1)$$

In the case where A_j is invertible, we furthermore have

$$\phi(t; 0, x_0, j) = e^{A_j \tau} x_0 + (e^{A_j \tau} - I_n) A_j^{-1} b_j$$

where I_n is the identity matrix of size n . In both cases we have an affine transformation. One can thus compute exactly the image of a set using zonotopes. Take an initial set given at time $t = 0$ as a zonotope $Z = \langle c, G \rangle$, its image (successor set) at time $t = \tau$ is (for A_j invertible) $Z' = Post_j(Z) = \langle e^{A_j\tau}c + (e^{A_j\tau} - I_n)A_j^{-1}b_j, e^{A_j\tau}G \rangle$. This formula can be iterated to obtain the successor set at time $t = k\tau$ of Z via a pattern $\pi = (j_1, \dots, j_k)$ for $k \in \mathbb{N}_{>0}$: $Post_\pi(Z) = Post_{j_k}(Post_{j_{k-1}}(\dots Post_{j_1}(Z)))$.

While computing the Tube operator is still a difficult task for linear systems, computing the Post operator in this way, associated to Algorithm 1 and 3 (without the safety property relying on the Tube), we can compute controllers permitting to return infinitely often in a set R thanks to Theorem 1. This approach can also be used to ensure discrete-time properties, i.e., which are not ensured between switchings but at discrete times $\tau, 2\tau\dots$. This approach is efficient and useful in practice, all the more so as the Post operator is computed exactly.

3.2 Validated simulation and state-space bisection

3.2.1 Validated simulation

In this subsection, we describe our approach for validated simulation based on Runge-Kutta methods [6, 35]. The goal is obviously to obtain a solution of the differential equations describing the modes of the nonlinear switched systems. Before presenting the method, we introduce some definitions.

In the following, we will often use the notation $[x] \in \mathbb{IR}$ (the set of intervals with real bounds) where

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\}$$

denotes an interval. By an abuse of notation $[x]$ will also denote a vector of intervals, *i.e.*, a Cartesian product of intervals, also known as a *box*. In the following, the sets R , S and B are given under the form of boxes. With interval values, it comes an associated interval arithmetic.

Interval arithmetic extends to \mathbb{IR} elementary functions over \mathbb{R} . For instance, the interval sum, *i.e.*, $[x_1] + [x_2] = [\underline{x}_1 + \underline{x}_2, \bar{x}_1 + \bar{x}_2]$, encloses the image of the sum function over its arguments. The enclosing property basically defines what is called an *interval extension* or an *inclusion function*.

Definition 4 (Inclusion function). *Consider a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, then $[f] : \mathbb{IR}^n \rightarrow \mathbb{IR}^m$ is said to be an extension of f to intervals if*

$$\forall [x] \in \mathbb{IR}^n, \quad [f]([x]) \supseteq \{f(x), x \in [x]\} .$$

It is possible to define inclusion functions for all elementary functions such as \times , \div , \sin , \cos , \exp , and so on. The *natural* inclusion function is the simplest to obtain:

all occurrences of the real variables are replaced by their interval counterpart and all arithmetic operations are evaluated using interval arithmetic. More sophisticated inclusion functions such as the centered form, or the Taylor inclusion function may also be used (see [93] for more details).

We now introduce the Initial Value Problem, which is one of main ingredients of our approach.

Definition 5 (Initial Value Problem (IVP)). *Consider an ODE with a given initial condition*

$$\dot{x}(t) = f(t, x(t), d(t)) \quad \text{with} \quad x(0) \in X_0, \quad d(t) \in [d], \quad (3.2)$$

with $f : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ assumed to be continuous in t and d and globally Lipschitz in x . We assume that parameters d are bounded (used to represent a perturbation, a modeling error, an uncertainty on measurement, ...). An IVP consists in finding a set-valued function $X(t)$ which contains any trajectory of the ODE (3.2), for any $d(t)$ lying in $[d]$ and for any initial condition in X_0 .

A numerical integration method computes a sequence of values (t_n, x_n) approximating the solution $x(t; x_0)$ of the IVP defined in Equation (3.2) such that $x_n \approx x(t_n; x_{n-1})$. The simplest method is Euler's method in which $t_{n+1} = t_n + h$ for some step-size h and $x_{n+1} = x_n + h \times f(t_n, x_n, d)$; so the derivative of x at time t_n , $f(t_n, x_n, d)$, is used as an approximation of the derivative on the whole time interval to perform a linear interpolation. This method is very simple and fast, but requires small step-sizes. More advanced methods, coming from the Runge-Kutta family, use a few intermediate computations to improve the approximation of the derivative. The general form of an explicit s -stage Runge-Kutta formula, that is using s evaluations of f , is

$$\begin{aligned} x_{n+1} &= x_n + h \sum_{i=1}^s b_i k_i , \\ k_1 &= f(t_n, x_n, d) , \\ k_i &= f\left(t_n + c_i h, x_n + h \sum_{j=1}^{i-1} a_{ij} k_j, d\right), \quad i = 2, 3, \dots, s . \end{aligned} \quad (3.3)$$

The coefficients c_i , a_{ij} and b_i fully characterize the method. To make Runge-Kutta validated, the challenging question is how to compute guaranteed bounds of the distance between the true solution and the numerical solution, defined by $x(t_n; x_{n-1}) - x_n$. This distance is associated to the *local truncation error* (LTE) of the numerical method.

To bound the LTE, we rely on *order condition* [84] respected by all Runge-Kutta methods. This condition states that a method of this family is of order p iff the $p+1$ first coefficients of the Taylor expansion of the solution and the Taylor expansion

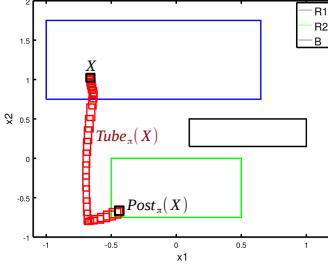


Figure 3.1: Functions $Post_\pi(X)$ and $Tube_\pi(X)$ for the initial box $X = [-0.69, -0.64] \times [1, 1.06]$, with a pattern $\pi = (1, 3, 0)$.

of the numerical methods are equal. In consequence, LTE is proportional to the Lagrange remainders of Taylor expansions. Formally, LTE is defined by (see [35]):

$$x(t_n; x_{n-1}) - x_n = \frac{h^{p+1}}{(p+1)!} \left(f^{(p)}(\xi, x(\xi; x_{n-1}), d) - \frac{d^{p+1}\varphi}{dt^{p+1}}(\eta) \right) \quad \xi \in]t_n, t_{n+1}[\text{ and } \eta \in]t_n, t_{n+1}[. \quad (3.4)$$

The function $f^{(n)}$ stands for the n -th derivative of function f w.r.t. time t that is $\frac{d^n f}{dt^n}$ and $h = t_{n+1} - t_n$ is the step-size. The function $\varphi : \mathbb{R} \rightarrow \mathbb{R}^n$ is defined by $\varphi(t) = x_n + h \sum_{i=1}^s b_i k_i(t)$ where $k_i(t)$ are defined as in Equation (3.3).

The challenge to make Runge-Kutta integration schemes safe w.r.t. the true solution of IVP is then to compute a bound of the result of Equation (3.4). In other words, we do have to bound the value of $f^{(p)}(\xi, x(\xi; x_{n-1}), d)$ and the value of $\frac{d^{p+1}\varphi}{dt^{p+1}}(\eta)$ with numerical guarantee. The latter expression is straightforward to bound because the function φ only depends on the value of the step-size h , and so does its $(p+1)$ -th derivative. The bound is then obtained using the affine arithmetic [7, 54].

However, the expression $f^{(p)}(\xi, x(\xi; x_{n-1}), d)$ is not so easy to bound as it requires to evaluate f for a particular value of the IVP solution $x(\xi; x_{n-1})$ at an unknown time $\xi \in]t_n, t_{n+1}[$. The solution used is the same as the one found in [36, 144] and it requires to bound the solution of IVP on the interval $[t_n, t_{n+1}]$. This bound is usually computed using the Banach's fixpoint theorem applied with the Picard-Lindelöf operator, see [144]. This operator is used to compute an enclosure of the solution $[\tilde{x}]$ of IVP over a time interval $[t_n, t_{n+1}]$, that is for all $t \in [t_n, t_{n+1}]$, $x(t; x_{n-1}) \in [\tilde{x}]$. We can hence bound $f^{(p)}$ substituting $x(\xi; x_{n-1})$ by $[\tilde{x}]$. This general approach used to solve IVPs in a validated way is called Lohner two step approach [127].

For a given pattern of switched modes $\pi = (i_1, \dots, i_k) \in U^k$ of length k , we are able to compute, for $j \in \{1, \dots, k\}$, the enclosures:

— $[x_j] \ni x(j\tau)$;

— $[\tilde{x}_j] \ni x(t)$, for $t \in [(j-1)\tau, j\tau]$.
with respect to the system of IVPs:

$$\left\{ \begin{array}{l} \dot{x}(t) = f_{\sigma(t)}(t, x(t), d(t)), \\ x(t_0 = 0) \in [x_0], d(t) \in [d], \\ \sigma(t) = i_1, \forall t \in [0, t_1], t_1 = \tau \\ \vdots \\ \dot{x}(t) = f_{\sigma(t)}(t, x(t), d(t)), \\ x(t_{k-1}) \in [x_{k-1}], d(t) \in [d], \\ \sigma(t) = i_k, \forall t \in [t_{k-1}, t_k], t_k = k\tau \end{array} \right.$$

Thereby, the enclosure $Post_\pi([x_0])$ is included in $[x_k]$ and $Tube_\pi([x_0])$ is included in $\bigcup_{j=1,\dots,k} [\tilde{x}_j]$. This applies for all initial states in $[x_0]$ and all disturbances $d(t) \in [d]$. A view of enclosures computed by the validated simulation for one solution obtained for Example 3.2.2 is shown in Figure 3.1.

Control synthesis

If we now associate computation of the Post and Tube operators to Algorithm 1 and 3, and using Theorem 2, we can now perform control synthesis ensuring (R, S) -stability, as well as (R_1, R_2, S) -reachability and (R, B, S) -avoidance.

3.2.2 Experiments

In this subsection, we apply our approach to different case studies taken from the literature. In every case study, a second order Runge-Kutta method is applied. Our solver prototype is written in C++ and based on DynIBEX [5]. The computations times given in the following have been performed on a 2.80 GHz Intel Core i7-4810MQ CPU with 8 GB of memory. Note that our algorithm is mono-threaded so all the experimentation only uses one core to perform the computations. The results given in this subsection have been obtained with Function *Find_Pattern2* of Chapter 2.

A linear example: boost DC-DC converter

This linear example is taken from [27] and has already been treated with the state-space bisection method in a linear framework in [67]. This running example is used to verify that our approach is still valid for linear case, and also to show the strong improvement in term of computation time.

The system is a boost DC-DC converter with one switching cell. There are two switching modes depending on the position of the switching cell. The dynamics is given by the equation $\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}$ with $\sigma(t) \in U = \{1, 2\}$. The two modes are given by the matrices:

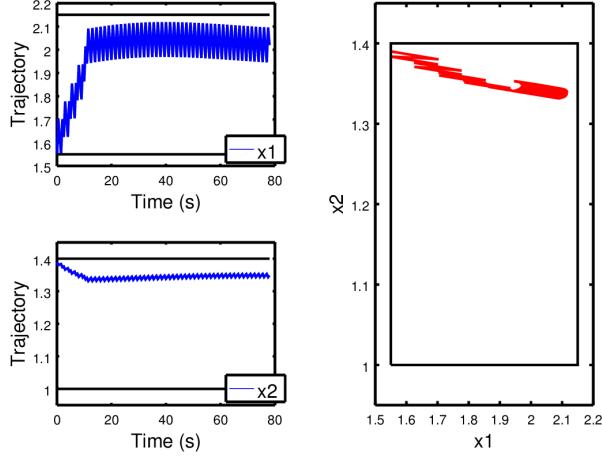


Figure 3.2: Simulation from the initial condition $(1.55, 1.4)$. The box R is in plain black. The trajectory is plotted within time for the two state variables on the left, and in the state-space plane on the right.

$$A_1 = \begin{pmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c} \frac{1}{r_0+r_c} \end{pmatrix} \quad B_1 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} -\frac{1}{x_l} (r_l + \frac{r_0 \cdot r_c}{r_0+r_c}) & -\frac{1}{x_l} \frac{r_0}{r_0+r_c} \\ \frac{1}{x_c} \frac{r_0}{r_0+r_c} & -\frac{1}{x_c} \frac{r_0}{r_0+r_c} \end{pmatrix} \quad B_2 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

with $x_c = 70$, $x_l = 3$, $r_c = 0.005$, $r_l = 0.05$, $r_0 = 1$, $v_s = 1$. The sampling period is $\tau = 0.5$. The parameters are exact and there is no perturbation. We want the state to return infinitely often to the region R , set here to $[1.55, 2.15] \times [1.0, 1.4]$, while never going out of the safety set $S = [1.54, 2.16] \times [0.99, 1.41]$. The goal of this example is then to synthesize a controller with intrinsic stability. The dynamics of the system is recalled in Appendix A.1.

The decomposition was obtained in less than one second with a maximum length of pattern set to $K = 6$ and a maximum bisection depth of $D = 3$. A simulation is given in Figure 3.2.

A polynomial example

We consider the polynomial system taken from [126], presented as a difficult example:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 - 1.5x_1 - 0.5x_1^3 + u_1 + d_1 \\ x_1 + u_2 + d_2 \end{bmatrix}. \quad (3.5)$$

The control inputs are given by $u = (u_1, u_2) = K_{\sigma(t)}(x_1, x_2)$, $\sigma(t) \in U = \{1, 2, 3, 4\}$, which correspond to four different state feedback controllers $K_1(x) = (0, -x_2^2 + 2)$, $K_2(x) = (0, -x_2)$, $K_3(x) = (2, 10)$, $K_4(x) = (-1.5, 10)$. We thus have four switching modes. The disturbance $d = (d_1, d_2)$ lies in $[-0.005, 0.005] \times [-0.005, 0.005]$. The

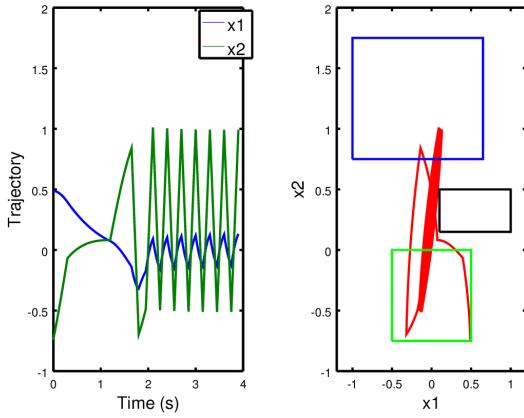


Figure 3.3: Simulation from the initial condition $(0.5, -0.75)$. The trajectory is plotted within time on the left, and in the state space plane on the right. In the state space plane, the set R_1 is in plain green, R_2 in plain blue, and B in plain black.

dynamics of the system is recalled in Appendix A.3. The objective is to visit infinitely often two zones R_1 and R_2 , without going out of a safety zone S , and while never crossing a forbidden zone B . Two decompositions are performed:

- a decomposition of R_1 which returns $\{(V_i, \pi_i)\}_{i \in I_1}$ with:
 - $\bigcup_{i \in I_1} V_i = R_1$,
 - $\forall i \in I_1, \text{Post}_{\pi_i}(V_i) \subseteq R_2$,
 - $\forall i \in I_1, \text{Tube}_{\pi_i}(V_i) \subseteq S$,
 - $\forall i \in I_1, \text{Tube}_{\pi_i}(V_i) \cap B = \emptyset$.
- a decomposition of R_2 which returns $\{(V_i, \pi_i)\}_{i \in I_2}$ with:
 - $\bigcup_{i \in I_2} V_i = R_2$,
 - $\forall i \in I_2, \text{Post}_{\pi_i}(V_i) \subseteq R_1$,
 - $\forall i \in I_2, \text{Tube}_{\pi_i}(V_i) \subseteq S$,
 - $\forall i \in I_2, \text{Tube}_{\pi_i}(V_i) \cap B = \emptyset$.

The input boxes are the following:

- $R_1 = [-0.5, 0.5] \times [-0.75, 0.0]$,
- $R_2 = [-1.0, 0.65] \times [0.75, 1.75]$,
- $S = [-2.0, 2.0] \times [-1.5, 3.0]$,
- $B = [0.1, 1.0] \times [0.15, 0.5]$.

The sampling period is set to $\tau = 0.15$. The decompositions were obtained in 2 minutes and 30 seconds with a maximum length of pattern set to $K = 12$ and a maximum bisection depth of $D = 5$. A simulation is given in Figure 3.3 in which the disturbance d is chosen randomly in $[-0.005, 0.005] \times [-0.005, 0.005]$ at every time step. We see that the trajectories do visit alternately R_1 and R_2 while staying in S and avoiding B .

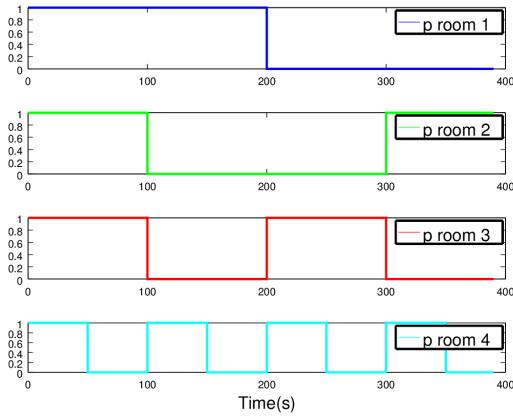


Figure 3.4: Perturbation (presence of humans) imposed within time in the different rooms.

Four-room apartment

We consider a building ventilation application adapted from [134]. The system is a four room apartment subject to heat transfer between the rooms, with the external environment, with the underfloor, and with human beings. The dynamics of the system is given by the following equation:

$$\frac{dT_i}{dt} = \sum_{j \in \mathcal{N}^* \setminus \{i\}} a_{ij}(T_j - T_i) + \delta_{s_i} b_i(T_{s_i}^4 - T_i^4) + c_i \max \left(0, \frac{V_i - V_i^*}{\bar{V}_i - V_i^*} \right) (T_u - T_i).$$

The state of the system is given by the temperatures in the rooms T_i , for $i \in \mathcal{N} = \{1, \dots, 4\}$. Room i is subject to heat exchange with different entities stated by the indexes $\mathcal{N}^* = \{1, 2, 3, 4, u, o, c\}$.

The heat transfer between the rooms is given by the coefficients a_{ij} for $i, j \in \mathcal{N}^2$, and the different perturbations are the following:

- The convective heat transfer with the external environment: it has an effect on room i with the coefficient a_{io} and the outside temperature T_o , varying between $27^\circ C$ and $30^\circ C$.
- The convective heat transfer through the ceiling: it has an effect on room i with the coefficient a_{ic} and the ceiling temperature T_c , varying between $27^\circ C$ and $30^\circ C$.
- The convective heat transfer with the underfloor: it is given by the coefficient a_{iu} and the underfloor temperature T_u , set to $17^\circ C$ (T_u is constant, regulated by a PID controller).
- The perturbation induced by the presence of humans, modeled by a radiation term: it is given in room i by the term $\delta_{s_i} b_i(T_{s_i}^4 - T_i^4)$, the parameter δ_{s_i} is equal to 1 when someone is present in room i , 0 otherwise, and T_{s_i} is a given identified parameter.

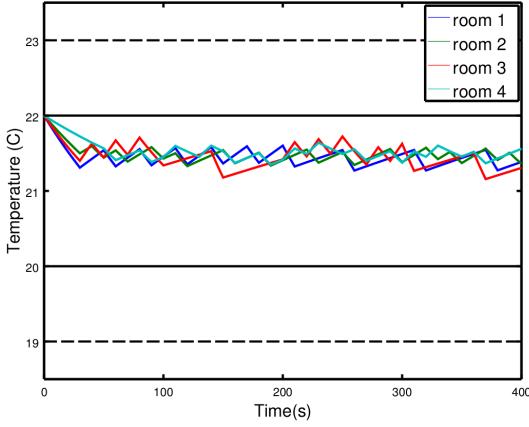


Figure 3.5: Simulation from the initial condition $(22, 22, 22, 22)$. The objective set R is in plain black and the safety set S is in dotted black.

The control V_i , $i \in \mathcal{N}$, is applied through the term $c_i \max(0, \frac{V_i - V_i^*}{\bar{V}_i - V_i^*})(T_u - T_i)$. A voltage V_i is applied to force ventilation from the underfloor to room i , and the command of an underfloor fan is subject to a dry friction. Because we work in a switched control framework, V_i can take only discrete values, which removes the problem of dealing with a “max” function in interval analysis. In the experiment, V_1 and V_4 can take the values 0V or 3.5V, and V_2 and V_3 can take the values 0V or 3V. This leads to a system of the form of Equation (2.1) with $\sigma(t) \in U = \{1, \dots, 16\}$, the 16 switching modes corresponding to the different possible combinations of voltages V_i . The sampling period is $\tau = 30$ s. The dynamics of the system is recalled in Appendix A.4.

The parameters T_{s_i} , V_i^* , \bar{V}_i , a_{ij} , b_i , c_i are given in [134] and have been identified with a proper identification procedure detailed in [137]. Note that here we have neglected the term $\sum_{j \in \mathcal{N}} \delta_{d_{ij}} c_{i,j} * h(T_j - T_i)$ of [134], representing the perturbation induced by the open or closed state of the doors between the rooms. Taking a “max” function into account with interval analysis is actually still a difficult task. However, this term could have been taken into account with a proper regularization (smoothing).

The main difficulty of this example is the large number of modes in the switched system, which induces a combinatorial issue.

The decomposition was obtained in 4 minutes with a maximum length of pattern set to $K = 2$ and a maximum bisection depth of $D = 4$. The perturbation due to human beings has been taken into account by setting the parameters δ_{s_i} equal to the whole interval $[0, 1]$ for the decomposition, and the imposed perturbation for the simulation is given Figure 3.4. The temperatures T_o and T_c have been set to the interval $[27, 30]$ for the decomposition, and are set to $30^\circ C$ for the simulation. A simulation of the controller obtained with the state-space bisection procedure is given in Figure 3.5, where the control objective is to stabilize the temperature in

$[20, 22]^4$ while never going out of $[19, 23]^4$.

A path planning problem

This last case study is based on a model of a vehicle initially introduced in [19] and successfully controlled in [154, 175] with the tools PESSOA and SCOTS. In this model, the motion of the front and rear pairs of wheels are approximated by a single front wheel and a single rear wheel. The dynamics of the vehicle is given by:

$$\begin{aligned}\dot{x} &= v_0 \frac{\cos(\alpha+\theta)}{\cos(\alpha)} \\ \dot{y} &= v_0 \frac{\sin(\alpha+\theta)}{\cos(\alpha)} \\ \dot{\theta} &= \frac{v_0}{b} \tan(\delta)\end{aligned}\tag{3.6}$$

where $\alpha = \arctan(a \tan(\delta)/b)$. The system is thus of dimension 3, (x, y) is the position of the vehicle, while θ is the orientation of the vehicle. The control inputs are v_0 , an input velocity, and δ , the steering angle of the rear wheel. The parameters are: $a = 0.5$, $b = 1$. Just as in [154, 175], we suppose that the control inputs are piecewise constant, which leads to a switched system of the form of Equation (2.1) with no perturbation. The objective is to send the vehicle into an objective region $R_2 = [9, 9.5] \times [0, 0.5] \times]-\infty, +\infty[$ from an initial region $R_1 = [0, 0.5] \times [0, 0.5] \times [0, 0]$. The safety set is $S = [0, 10] \times [0, 10] \times]-\infty, +\infty[$. There is in fact no particular constraint on the orientation of the vehicle, but multiple obstacles are imposed for the two first dimensions, they are represented in Figure 3.6. The input velocity v_0 can take the values in $\{-0.5, 0.5, 1.0\}$. The rear wheel orientation δ can take the values in $\{0.9, 0.6, 0.5, 0.3, 0.0, -0.3, -0.5, -0.6, -0.9\}$. The sampling period is $\tau = 0.3$. The dynamics of the system is recalled in Appendix A.6.

Note that for this case study we used an automated pre-tiling of the state-space permitting to decompose the reachability problem in a sequence of reachability problems. Using patterns of length up to $K = 10$, we managed to successfully control the system in 3619 seconds. In this case, the pattern is computed until almost the end without bisection as shown in Figure 3.6. To obtain the last steps, the box is bisected in four ones by Algorithm 1. After that, patterns are found for the four boxes:

- $[8.43, 8.69]; [2.52, 2.78] : \{7000166\}$
- $[8.43, 8.69]; [2.78, 3.03] : \{7000256\}$
- $[8.69, 8.94]; [2.52, 2.78] : \{00055\}$
- $[8.69, 8.94]; [2.78, 3.03] : \{000265\}$

The four set simulations obtained for the last steps are given in Figure 3.7.

3.2.3 Performance tests

We present a comparison of functions *Find_Pattern*, *Find_Pattern2* w.r.t. the computation times obtained, and with the state-of-the-art tools PESSOA [132] and

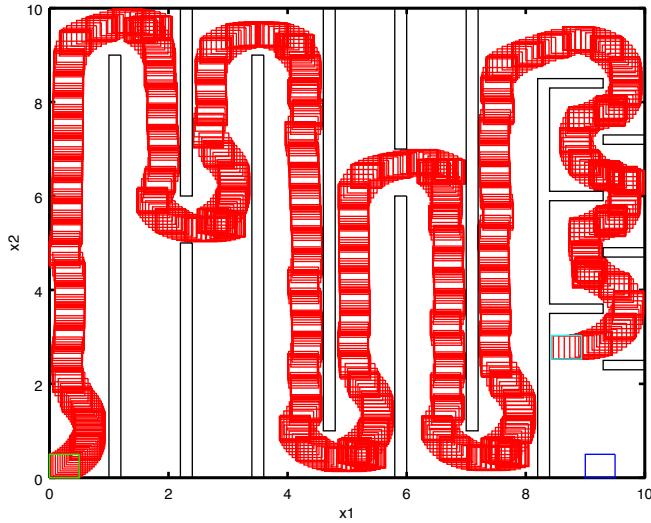


Figure 3.6: Set simulation of the path planning example. The green box is the initial region R_1 , the blue box is the target region R_2 . The union of the red boxes is the reachability tube. In this case, the target region is not attained without bisection.

Table 3.1: Comparison of *Find_Pattern* and *Find_Pattern2*.

Example	Computation time	
	<i>Find_Pattern</i>	<i>Find_Pattern2</i>
DC-DC Converter	1609 s	< 1 s
Polynomial example	Time Out	150 s
Building ventilation	272 s	228 s
Path planning	Time Out	3619 s

SCOTS [159].

Table 3.1 shows a comparison of functions *Find_Pattern* and *Find_Pattern2*, which shows that the new version highly improves computation time (Time Outs refer to computation times exceeding 10 hours). We can note that the new version is all the more efficient as the length of the patterns increases, and as obstacles cut the research tree of patterns. This is why we observe significant improvements on the examples of the DC-DC converter and the polynomial example, and not on the building ventilation example, which only requires patterns of length 2, and presents no obstacle.

Table 3.2 shows of comparison of function *Find_Pattern2* with state-of-the-art tools SCOTS and PESSOA. On the example of the DC-DC converter, our algorithm manages to control the whole state-space $R = [1.55, 2.15] \times [1.0, 1.4]$ in less than one second, while SCOTS and PESSOA only control a part of R , and with greater computation times. Note that these computation times vary with the number of discretization points used in both, but even with a very fine discretization, we never managed to control the whole box R . For the polynomial example, we manage to

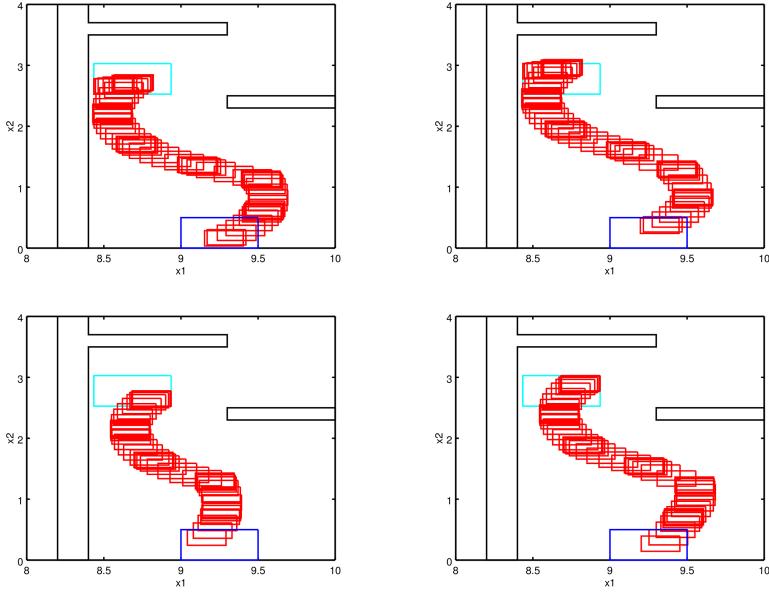


Figure 3.7: Set simulation of the path planning example after bisection. The green boxes are the initial regions obtained by bisection, the blue box is the target region R_2 . The union of the red boxes is the reachability tube.

Table 3.2: Comparison with state-of-the-art tools.

Example	Computation time		
	FP2	SCOTS	PESSOA
DC-DC Converter	< 1 s	43 s	760 s
Polynomial example	150 s	131 s	--
Path planning	3619 s	492 s	516 s

control the whole boxes R_1 and R_2 , such as SCOTS and in a comparable amount of time. However, PESSOA does not support natively this kind of nonlinear systems. For path planning case study, on which PESSOA and SCOTS perform well, we have not obtained as good computations times as [132, 159]. This comes from the fact that this example requires a high number of switched modes, long patterns, as well as a high number of boxes to tile the state-space. This is in fact the most difficult case of application of our method. This reveals that our method is more adapted when either the number of switched modes or the length of patterns is not high (though it can be handled at the cost of high computation times). Another advantage is that we do not require a homogeneous discretization of the state space. We can thus tile large parts of the state-space using only few boxes, and this often permits to consider much less symbolic states than with discretization methods, especially in higher dimensions.

3.2.4 Final remarks

We presented a method of control synthesis for nonlinear switched systems, based on a simple state-space bisection algorithm, and on validated simulation. The approach permits to deal with stability, reachability, safety and forbidden region constraints. Varying parameters and perturbations can be easily taken into account with interval analysis. The approach has been numerically validated on several examples taken from the literature, a linear one with constant parameters, and two nonlinear ones with varying perturbations. Our approach compares well with the state-of-the art tools SCOTS and PESSOA.

We would like to point out that the exponential complexity of the algorithms presented here, which is inherent to guaranteed methods, is not prohibitive. Two approaches have indeed been developed to overcome this exponential complexity. A first approach is the use of compositionality, presented in Chapter 4, which permits to split the system in two (or more) sub-systems, and to perform control synthesis on these sub-systems of lower dimensions. This approach has been successfully applied in [120] to a system of dimension 11, and we are currently working on applying this approach to the more general context of contract-based design [161]. A second approach, developed in Chapters 5 and 6, is the use of Model Order Reduction, which allows to approximate the full-order system (2.1) with a reduced-order system, of lower dimension, on which it is possible to perform control synthesis.

3.3 Sampled switched systems with one-sided Lipschitz conditions

3.3.1 Lipschitz and one-sided Lipschitz condition

Let us consider a nonlinear switched system of the form (2.1). We make the following hypothesis:

(H0) For all $j \in U$, f_j is a locally Lipschitz continuous map.

We recall the definition of locally Lipschitz:

Definition 6. A function $f : A \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ is locally Lipschitz at $x_0 \in A$ if there exist constants $\eta > 0$ and $M > 0$ such that

$$\|x - x_0\| < \eta \rightarrow \|f(x) - f(x_0)\| \leq M\|x - x_0\|$$

As in [78], we make the assumption that the vector field f_j is such that the solutions of the differential equation (2.1) are defined, e.g. by assuming that the support of the vector field f_j is compact.

We denote by T a compact overapproximation of the image by ϕ_j of S for $0 \leq t \leq \tau$ and $j \in U$, i.e. T is such that

$$T \supseteq \{\phi_j(t; x^0) \mid j \in U, 0 \leq t \leq \tau, x^0 \in S\}.$$

The existence of T is guaranteed by assumption (H0). We know furthermore by (H0), Definition 6 and the compactness of the support of f_j that, for all $j \in U$, there exists a constant $L_j > 0$ such that:

$$\|f_j(y) - f_j(x)\| \leq L_j \|y - x\| \quad \forall x, y \in S. \quad (3.7)$$

Let us define C_j for all $j \in U$:

$$C_j = \sup_{x \in S} L_j \|f_j(x)\| \quad \text{for all } j \in U. \quad (3.8)$$

We make the additional hypothesis that the mappings f_j are *one-sided Lipschitz* (OSL) [57].

Formally:

(H1) For all $j \in U$, there exists a constant $\lambda_j \in \mathbb{R}$ such that

$$\langle f_j(y) - f_j(x), y - x \rangle \leq \lambda_j \|y - x\|^2 \quad \forall x, y \in T, \quad (3.9)$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product of two vectors of \mathbb{R}^n . Constant $\lambda_j \in \mathbb{R}$ is called one-sided Lipschitz (OSL) constant, and can also be found in the literature as Dahlquist's constant [168]. Note that in practice, hypotheses H0 and H1 are not strong. Hypothesis H0 just ensures the existence of solutions for the system, and constants L_j and λ_j can always be found if the state of the system stays in a compact set (e.g. the set T).

Computation of constants λ_j , L_j and C_j The computation of constants L_j , C_j , λ_j ($j \in U$) are realized with a constrained optimization algorithm. They are performed using the “sqp” function of Octave, applied on the following optimization problems:

— Constant L_j :

$$L_j = \max_{x, y \in S, x \neq y} \frac{\|f_j(y) - f_j(x)\|}{\|y - x\|}$$

— Constant C_j :

$$C_j = \max_{x \in S} L_j \|f_j(x)\|$$

— Constant λ_j :

$$\lambda_j = \max_{x, y \in T, x \neq y} \frac{\langle f_j(y) - f_j(x), y - x \rangle}{\|y - x\|^2}$$

We could point out that the computation of the constants is not guaranteed, in the sense that the results given by optimization algorithms do not provide a guarantee that an underapproximation of the constants is computed. However, some works have been done for computing over and under approximation of Lipschitz constants in [148], and could be used here. This approach can be extended to the OSL constant. In the following, we consider that we can compute these constants exactly.

Origin of the OSL property This notion has been used for the first time by [58] in order to treat “stiff” systems of differential equations for which the explicit Euler method is numerically “unstable” (unless the step size is taken to be extremely small). Unlike Lipschitz constants, OSL constants can be *negative*. In the case where an OSL constant λ_j is negative, it is said that the vector field f_j is strongly monotone [166], which expresses a form of contractivity of the system dynamics: a strongly monotone system presents trajectories getting exponentially closer together within time. Even if the OSL constant is positive, it is in practice much lower than the Lipschitz constant [52]. The use of OSL thus allows us to obtain a much more precise upper bound for the global error. We believe that this notion is also closely related to the notion of incremental stability [13, 77]. We think that it could be shown that any system presenting a negative OSL constant is incrementally stable, since it is already the case for linear systems. Indeed, a system presenting a negative OSL constant actually admits $\|\cdot\|^2$ as a stable Lyapunov function [13]. However, this OSL Lipschitz property has never been used in the context of switched systems and symbolic control.

3.3.2 A note on the OSL constant for linear systems

We show here a result giving an exact expression for the OSL constant for linear vector fields.

Proposition 2. *Let $X \subset \mathbb{R}^n$ be a (non trivial) compact set. Let $A \in \mathcal{M}_n(\mathbb{R})$, $b \in \mathbb{R}^n$ and $f(x) = Ax + b$. The OSL constant of f is equal to the greatest eigenvalue of $\frac{A+A^\top}{2}$.*

Proof. First

$$\exists \lambda \in \mathbb{R} \text{ s.t. } \langle f(y) - f(x), y - x \rangle \leq \lambda \|y - x\|^2 \quad \forall x, y \in X,$$

is equivalent to

$$\exists \lambda \in \mathbb{R} \text{ s.t. } \langle A(y - x), y - x \rangle \leq \lambda \|y - x\|^2 \quad \forall x, y \in X,$$

and is equivalent to (the case $x = y$ being trivial)

$$\exists \lambda \in \mathbb{R} \text{ s.t. } \langle A \frac{y - x}{\|y - x\|}, \frac{y - x}{\|y - x\|} \rangle \leq \lambda \quad \forall x, y \in X, x \neq y, \quad (3.10)$$

and it is thus equivalent to

$$\exists \lambda \in \mathbb{R} \text{ s.t. } \langle Az, z \rangle \leq \lambda \quad \forall z \in S(0, 1), \quad (3.11)$$

where $S(0, 1)$ is the sphere of center 0 and radius 1 in \mathbb{R}^n , and because X is non trivial.

Let us then remark that we have

$$\langle Az, z \rangle = \left\langle \frac{A + A^\top}{2} z, z \right\rangle \quad (3.12)$$

Indeed, if $A = (a_{ij})_{ij}$ and $z = (z_i)_i$:

$$\begin{aligned} \langle Az, z \rangle &= \sum_{i=1}^n \sum_{j=1}^n z_i a_{ij} z_j = \sum_{i=1}^n \sum_{j=1}^n a_{ij} z_i z_j \\ \left\langle \frac{A + A^\top}{2} z, z \right\rangle &= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n a_{ij} z_i z_j + \sum_{i=1}^n \sum_{j=1}^n a_{ji} z_i z_j \right) \end{aligned}$$

The sums on the last term can be exchanged, it yields

$$\begin{aligned} \left\langle \frac{A + A^\top}{2} z, z \right\rangle &= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n a_{ij} z_i z_j + \sum_{j=1}^n \sum_{i=1}^n a_{ji} z_i z_j \right) \\ &= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n a_{ij} z_i z_j + \sum_{i=1}^n \sum_{j=1}^n a_{ij} z_i z_j \right) \\ &= \langle Az, z \rangle \end{aligned}$$

We thus have equivalence of (3.11) and

$$\exists \lambda \in \mathbb{R} \text{ s.t. } \left\langle \frac{A + A^\top}{2} z, z \right\rangle \leq \lambda \quad \forall z \in S(0, 1), \quad (3.13)$$

Now, $\frac{A+A^\top}{2}$ is a symmetric matrix, let us denote by $\lambda_1^s, \dots, \lambda_n^s$ its (real) eigenvalues. Let us denote by λ_{min}^s the minimum one, and by λ_{max}^s the maximum one. We can apply the known result (using for example Rayleigh quotient's properties [147]):

$$\forall z \in S(0, 1), \quad \lambda_{min}^s \leq \left\langle \frac{A + A^\top}{2} z, z \right\rangle \leq \lambda_{max}^s$$

and equality is attained in both sides for z (normalized) eigenvector of $\frac{A+A^\top}{2}$ corresponding to eigenvalues λ_{min}^s and λ_{max}^s , which proves the result.

□

Remark 2. Function $\phi : z \mapsto \langle Az, z \rangle$ is a quadratic form. There is thus a unique symmetric matrix M such that $\phi(z) = \langle Mz, z \rangle$, this unique symmetric matrix is $\frac{A+A^\top}{2}$.

3.3.3 Euler approximate solutions

Having defined OSL conditions, we now present an original method allowing to compute reachability sets and tubes, relying on the Euler method. The introduction of OSL conditions actually allows to establish a new global error bound, permitting the computation of overapproximation of reachability sets and tubes, precise enough to be used for control synthesis. In the remainder of this chapter, we consider, without loss of generality, that $t_0 = 0$, and omit its notation in the trajectory ϕ_j .

Given an initial point $\tilde{x}^0 \in S$ and a mode $j \in U$, we define the following “linear approximate solution” $\tilde{\phi}_j(t; \tilde{x}^0)$ for t on $[0, \tau]$ by:

$$\tilde{\phi}_j(t; \tilde{x}^0) = \tilde{x}^0 + t f_j(\tilde{x}^0). \quad (3.14)$$

Note that formula (3.14) is nothing else but the explicit forward Euler scheme with “time step” t . It is thus a consistent approximation of order 1 in t of the exact trajectory of (2.1) under the hypothesis $\tilde{x}^0 = x^0$.

More generally, given an initial point $\tilde{x}^0 \in S$ and pattern π of U^k , we can define a “(piecewise linear) approximate solution” $\tilde{\phi}_\pi(t; \tilde{x}^0)$ of ϕ_π at time $t \in [0, k\tau]$ as follows:

- $\tilde{\phi}_\pi(t; \tilde{x}^0) = t f_j(\tilde{x}^0) + \tilde{x}^0$ if $\pi = j \in U$, $k = 1$ and $t \in [0, \tau]$, and
- $\tilde{\phi}_\pi(k\tau + t; \tilde{x}^0) = t f_j(\tilde{z}) + \tilde{z}$ with $\tilde{z} = \tilde{\phi}_{\pi'}((k-1)\tau; \tilde{x}^0)$, if $k \geq 2$, $t \in [0, \tau]$, $\pi = j \cdot \pi'$ for some $j \in U$ and $\pi' \in U^{k-1}$.

We wish to synthesize a guaranteed control σ for ϕ_σ using the approximate functions $\tilde{\phi}_\pi$. We define the closed ball of center $x \in \mathbb{R}^n$ and radius $r > 0$, denoted $B(x, r)$, as the set $\{x' \in \mathbb{R}^n \mid \|x' - x\| \leq r\}$.

Given a positive real δ , we now define the expression $\delta_j(t)$ which, as we will see in Theorem 3, represents (an upper bound on) the error associated to $\tilde{\phi}_j(t; \tilde{x}^0)$ (i.e. $\|\tilde{\phi}_j(t; \tilde{x}^0) - \phi_j(t; x^0)\|$).

Definition 7. Let us consider a switched system verifying hypotheses (H0) and (H1), associated to constants λ_j , L_j and C_j for each mode $j \in U$, such that equations (3.7), (3.8) and (3.9) hold. Let δ be a positive constant. We define, for all $0 \leq t \leq \tau$, function $\delta_j(t)$ as follows:

- if $\lambda_j < 0$:

$$\delta_j(t) = \left(\delta^2 e^{\lambda_j t} + \frac{C_j^2}{\lambda_j^2} \left(t^2 + \frac{2t}{\lambda_j} + \frac{2}{\lambda_j^2} (1 - e^{\lambda_j t}) \right) \right)^{\frac{1}{2}}$$

- if $\lambda_j = 0$:

$$\delta_j(t) = (\delta^2 e^t + C_j^2 (-t^2 - 2t + 2(e^t - 1)))^{\frac{1}{2}}$$

— if $\lambda_j > 0$:

$$\delta_j(t) = \left(\delta^2 e^{3\lambda_j t} + \frac{C_j^2}{3\lambda_j^2} \left(-t^2 - \frac{2t}{3\lambda_j} + \frac{2}{9\lambda_j^2} (e^{3\lambda_j t} - 1) \right) \right)^{\frac{1}{2}}$$

Note that $\delta_j(t) = \delta$ for $t = 0$. The function $\delta_j(\cdot)$ depends implicitly on two parameters: $\delta \in \mathbb{R}$ and $j \in U$. In Section 3.3.4, we will use the notation $\delta'_j(\cdot)$ where the parameters are denoted by δ' and j .

Theorem 3. *Given a sampled switched system satisfying (H0-H1), consider a point \tilde{x}^0 and a positive real δ . We have, for all $x^0 \in B(\tilde{x}^0, \delta)$, $t \in [0, \tau]$ and $j \in U$:*

$$\phi_j(t; x^0) \in B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t)).$$

Proof. Consider on $t \in [0, \tau]$ the differential equations

$$\frac{dx(t)}{dt} = f_j(x(t))$$

and

$$\frac{d\tilde{x}(t)}{dt} = f_j(\tilde{x}^0).$$

with initial points $x^0 \in S, \tilde{x}^0 \in S$ respectively. We will abbreviate $\phi_j(t; x^0)$ (resp. $\tilde{\phi}_j(t; \tilde{x}^0)$) as $x(t)$ (resp. $\tilde{x}(t)$). We have

$$\frac{d}{dt}(x(t) - \tilde{x}(t)) = (f_j(x(t)) - f_j(\tilde{x}^0)),$$

then

$$\begin{aligned} \frac{1}{2} \frac{d}{dt}(\|x(t) - \tilde{x}(t)\|^2) &= \langle f_j(x(t)) - f_j(\tilde{x}^0), x(t) - \tilde{x}(t) \rangle \\ &= \langle f_j(x(t)) - f_j(\tilde{x}(t)) + f_j(\tilde{x}(t)) - f_j(\tilde{x}^0), x(t) - \tilde{x}(t) \rangle \\ &= \langle f_j(x(t)) - f_j(\tilde{x}(t)), x(t) - \tilde{x}(t) \rangle \\ &\quad + \langle f_j(\tilde{x}(t)) - f_j(\tilde{x}^0), x(t) - \tilde{x}(t) \rangle \\ &\leq \langle f_j(x(t)) - f_j(\tilde{x}(t)), x(t) - \tilde{x}(t) \rangle \\ &\quad + \|f_j(\tilde{x}(t)) - f_j(\tilde{x}^0)\| \|x(t) - \tilde{x}(t)\|. \end{aligned}$$

The last expression has been obtained using the Cauchy-Schwarz inequality. Using (H1) and (3.7), we have

$$\begin{aligned} \frac{1}{2} \frac{d}{dt}(\|x(t) - \tilde{x}(t)\|^2) &\leq \lambda_j \|x(t) - \tilde{x}(t)\|^2 + \|f_j(\tilde{x}(t)) - f_j(\tilde{x}^0)\| \|x(t) - \tilde{x}(t)\| \\ &\leq \lambda_j \|x(t) - \tilde{x}(t)\|^2 + L_j \|\tilde{x}(t) - \tilde{x}^0\| \|x(t) - \tilde{x}(t)\| \\ &\leq \lambda_j \|x(t) - \tilde{x}(t)\|^2 + L_j t \|f_j(\tilde{x}^0)\| \|x(t) - \tilde{x}(t)\|. \end{aligned}$$

Using (3.8) and a Young inequality, we then have

$$\begin{aligned} \frac{1}{2} \frac{d}{dt}(\|x(t) - \tilde{x}(t)\|^2) &\leq \lambda_j \|x(t) - \tilde{x}(t)\|^2 + C_j t \|x(t) - \tilde{x}(t)\| \\ &\leq \lambda_j \|x(t) - \tilde{x}(t)\|^2 + C_j t \frac{1}{2} \left(\alpha \|x(t) - \tilde{x}(t)\|^2 + \frac{1}{\alpha} \right) \end{aligned}$$

for all $\alpha > 0$.

— In the case $\lambda_j < 0$:

For $t > 0$, we choose $\alpha > 0$ such that $C_j t \alpha = -\lambda_j$, i.e. $\alpha = -\frac{\lambda_j}{C_j t}$. It follows, for all $t \in [0, \tau]$:

$$\frac{1}{2} \frac{d}{dt} (\|x(t) - \tilde{x}(t)\|^2) \leq \frac{\lambda_j}{2} \|x(t) - \tilde{x}(t)\|^2 - \frac{C_j t}{2\alpha} = \frac{\lambda_j}{2} \|x(t) - \tilde{x}(t)\|^2 - \frac{(C_j t)^2}{2\lambda_j}.$$

We thus get:

$$\|x(t) - \tilde{x}(t)\|^2 \leq \|x^0 - \tilde{x}^0\|^2 e^{\lambda_j t} + \frac{C_j^2}{\lambda_j^2} \left(t^2 + \frac{2t}{\lambda_j} + \frac{2}{\lambda_j^2} (1 - e^{\lambda_j t}) \right).$$

— In the case $\lambda_j > 0$:

For $t > 0$, we choose $\alpha > 0$ such that $C_j t \alpha = \lambda_j$, i.e. $\alpha = \frac{\lambda_j}{C_j t}$. It follows, for all $t \in [0, \tau]$:

$$\frac{1}{2} \frac{d}{dt} (\|x(t) - \tilde{x}(t)\|^2) \leq \frac{3\lambda_j}{2} \|x(t) - \tilde{x}(t)\|^2 + \frac{C_j t}{2\alpha} = \frac{3\lambda_j}{2} \|x(t) - \tilde{x}(t)\|^2 + \frac{(C_j t)^2}{2\lambda_j}.$$

We thus get:

$$\|x(t) - \tilde{x}(t)\|^2 \leq \|x^0 - \tilde{x}^0\|^2 e^{3\lambda_j t} + \frac{C_j^2}{3\lambda_j^2} \left(-t^2 - \frac{2t}{3\lambda_j} + \frac{2}{9\lambda_j^2} (e^{3\lambda_j t} - 1) \right)$$

— In the case $\lambda_j = 0$:

For $t > 0$, we choose $\alpha = \frac{1}{C_j t}$. It follows:

$$\frac{d}{dt} (\|x(t) - \tilde{x}(t)\|^2) \leq \|x(t) - \tilde{x}(t)\|^2 + C_j t^2$$

We thus get:

$$\|x(t) - \tilde{x}(t)\|^2 \leq \|x^0 - \tilde{x}^0\|^2 e^t + C_j^2 (-t^2 - 2t + 2(e^t - 1))$$

In every case, since by hypothesis $x^0 \in B(\tilde{x}^0, \delta)$ (i.e. $\|x^0 - \tilde{x}^0\|^2 \leq \delta^2$), we have, for all $t \in [0, \tau]$:

$$\|x(t) - \tilde{x}(t)\| \leq \delta_j(t).$$

It follows: $\phi_j(t; x^0) \in B(\tilde{\phi}_j(t; \tilde{x}^0), \delta)$ for $t \in [0, \tau]$.

□

Remark 3. In Theorem 3, we have supposed that the step size h used in Euler's method was equal to the sampling period τ of the switching system. Actually, in order to have better approximations, it is sometimes convenient to consider a uniform subdivision of $[0, \tau]$ and apply the Euler's method for a time step h equal to e.g. $h = \frac{\tau}{10}$. Such a splitting is called "sub-sampling" in numerical methods. See Section 3.3.5 for details.

Corollary 1. *Given a sampled switched system satisfying (H0-H1), consider a point $\tilde{x}^0 \in S$, a real $\delta > 0$ and a mode $j \in U$ such that:*

1. $B(\tilde{x}^0, \delta) \subseteq S$,
2. $B(\tilde{\phi}_j(\tau; \tilde{x}^0), \delta_j(\tau)) \subseteq S$, and
3. $\frac{d^2(\delta_j(t))}{dt^2} > 0$ for all $t \in [0, \tau]$.

Then we have, for all $x^0 \in B(\tilde{x}^0, \delta)$ and $t \in [0, \tau]$: $\phi_j(t; x^0) \in S$.

Proof. By items 1 and 2, $B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t)) \subseteq S$ for $t = 0$ and $t = \tau$. Since $\delta_j(\cdot)$ is convex on $[0, \tau]$ by item 3, and S is convex, we have $B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t)) \subseteq S$ for all $t \in [0, \tau]$. It follows from Theorem 3 that $\phi_j(t; x^0) \in B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t)) \subseteq S$ for all $1 \leq t \leq \tau$. \square

Remark 4. *Condition 3 of Corollary 1 on the convexity of $\delta_j(\cdot)$ on $[0, \tau]$ can be established again using an optimization function. Since we have an exact expression for $\delta_j(\cdot)$, its second derivative (w.r.t. time) can be computed using a computer algebra software. Using an optimization algorithm then allows to verify that its minimum is positive.*

3.3.4 Application to control synthesis

Consider a point $\tilde{x}^0 \in S$, a positive real δ and a pattern π of length k . Let $\pi(k')$ denote the k' -th element (mode) of π for $1 \leq k' \leq k$. Let us abbreviate the k' -th approximate point $\tilde{\phi}_\pi(k'\tau; \tilde{x}^0)$ as $\tilde{x}_\pi^{k'}$ for $k' = 1, \dots, k$, and let $\tilde{x}_\pi^{k'} = \tilde{x}^0$ for $k' = 0$. It is easy to show that $\tilde{x}_\pi^{k'}$ can be defined recursively for $k' = 1, \dots, k$, by: $\tilde{x}_\pi^{k'} = \tilde{x}_\pi^{k'-1} + \tau f_{\pi(k')}(\tilde{x}_\pi^{k'-1})$ with $j = \pi(k')$.

Let us now denote by $\delta_\pi^{k'}$ (an upper bound on) the error associated to $\tilde{x}_\pi^{k'}$, i.e. $\|\tilde{x}_\pi^{k'} - \phi_\pi(k'\tau; x^0)\|$. Using repeatedly Theorem 3, $\delta_\pi^{k'}$ can be defined recursively as follows:

For $k' = 0$: $\delta_\pi^{k'} = \delta$, and for $1 \leq k' \leq k$: $\delta_\pi^{k'} = \delta'_j(\tau)$ where δ' denotes $\delta_\pi^{k'-1}$, and j denotes $\pi(k')$.

Likewise, for $0 \leq t \leq k\tau$, let us denote by $\delta_\pi(t)$ (an upper bound on) the global error associated to $\tilde{\phi}_\pi(t; \tilde{x}^0)$ (i.e. $\|\tilde{\phi}_\pi(t; \tilde{x}^0) - \phi_\pi(t; x^0)\|$). Using Theorem 3, $\delta_\pi(t)$ can be defined itself as follows:

- for $t = 0$: $\delta_\pi(t) = \delta$,
- for $0 < t \leq k\tau$: $\delta_\pi(t) = \delta'_j(t')$ with $\delta' = \delta_\pi^{\ell-1}$, $j = \pi(\ell)$, $t' = t - (\ell - 1)\tau$ and $\ell = \lceil \frac{t}{\tau} \rceil$.

Note that, for $0 \leq k' \leq k$, we have: $\delta_\pi(k'\tau) = \delta_\pi^{k'}$. We have:

Theorem 4. *Given a sampled switched system satisfying (H0-H1), consider an initial point $\tilde{x}^0 \in S$, a positive real δ and a pattern π of length k such that, for all $1 \leq k' \leq k$:*

1. $B(\tilde{x}_\pi^{k'}, \delta_\pi^{k'}) \subseteq S$ and
2. $\frac{d^2(\delta'_j(t))}{dt^2} > 0$ for all $t \in [0, \tau]$, with $j = \pi(k')$ and $\delta' = \delta_\pi^{k'-1}$.

Then we have, for all $x^0 \in B(\tilde{x}^0, \delta)$ and $t \in [0, k\tau]$: $\phi_\pi(t; x^0) \in S$.

Proof. By induction on k using Corollary 1. \square

The statement of Theorem 4 is illustrated in Figure 3.8 for $k = 2$. From Theorem 4, it easily follows:

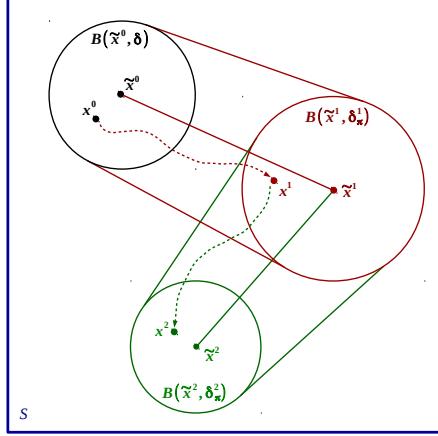


Figure 3.8: Illustration of Theorem 4.

Corollary 2. Given a switched system satisfying (H0-H1), consider a positive real δ and a finite set of points $\tilde{x}_1, \dots, \tilde{x}_m$ of S such that all the balls $B(\tilde{x}_i, \delta)$ cover R and are included into S (i.e. $R \subseteq \bigcup_{i=1}^m B(\tilde{x}_i, \delta) \subseteq S$). Suppose furthermore that, for all $1 \leq i \leq m$, there exists a pattern π_i of length k_i such that:

1. $B((\tilde{x}_i)^{k'}_{\pi_i}, \delta_{\pi_i}^{k'}) \subseteq S$, for all $k' = 1, \dots, k_i - 1$
2. $B((\tilde{x}_i)^{k_i}_{\pi_i}, \delta_{\pi_i}^{k_i}) \subseteq R$.
3. $\frac{d^2(\delta'_j(t))}{dt^2} > 0$ with $j = \pi_i(k')$ and $\delta' = \delta_{\pi_i}^{k'-1}$, for all $k' \in \{1, \dots, k_i\}$ and $t \in [0, \tau]$.

These properties induce a control σ which guarantees

- (safety): if $x \in R$, then $\phi_\sigma(t; x) \in S$ for all $t \geq 0$, and
- (recurrence): if $x \in R$ then $\phi_\sigma(k\tau; x) \in R$ for some $k \in \{k_1, \dots, k_m\}$.

Corollary 2 gives the theoretical foundations of the following method for synthesizing σ ensuring recurrence in R and safety in S :

- we (pre-)compute λ_j, L_j, C_j for all $j \in U$;
- we find m points $\tilde{x}_1, \dots, \tilde{x}_m$ of S and $\delta > 0$ such that $R \subseteq \bigcup_{i=1}^m B(\tilde{x}_i, \delta) \subseteq S$;

-
1. Given an initial point $x \in R$, the induced control σ corresponds to a sequence of patterns $\pi_{i_1}, \pi_{i_2}, \dots$ defined as follows: Since $x \in R$, there exists a point \tilde{x}_{i_1} with $1 \leq i_1 \leq m$ such that $x \in B(\tilde{x}_{i_1}, \delta)$; then using pattern π_{i_1} , one has: $\phi_{\pi_{i_1}}(k_{i_1}\tau; x) \in R$. Let $x' = \phi_{\pi_{i_1}}(k_{i_1}\tau; x)$; there exists a point \tilde{x}_{i_2} with $1 \leq i_2 \leq m$ such that $x' \in B(\tilde{x}_{i_2}, \delta)$, etc.

- we find m patterns π_i ($i = 1, \dots, m$) such that conditions 1-2-3 of Corollary 2 are satisfied.

A covering of R with balls as stated in Corollary 2 is illustrated in Figure 3.9. The control synthesis method based on Corollary 2 is illustrated in Figure 3.10 (left) together with an illustration of the validated simulation approach of Section 3.2 (right).

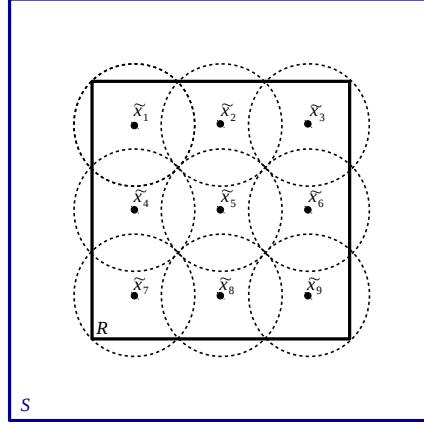


Figure 3.9: A set of balls covering R and contained in S .

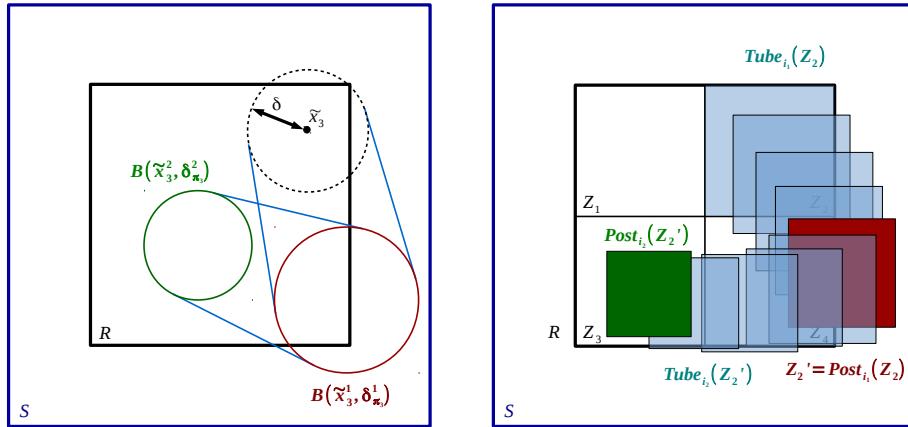


Figure 3.10: Control of ball $B(\tilde{x}_3, \delta)$ with our method (left); control of tile Z_2 with the method of Section 3.2 (right).

This theorem is actually an equivalent of Theorem 2 using balls, it thus solves Problem 1.

3.3.5 Numerical experiments and results

This method has been implemented in the interpreted language Octave, and the experiments performed on a 2.80 GHz Intel Core i7-4810MQ CPU with 8 GB of

memory.

Note that in some cases, it is advantageous to use a time sub-sampling to compute the image of a ball. Indeed, because of the exponential growth of the radius $\delta_j(t)$ within time, computing a sequence of balls can lead to smaller ball images. It is particularly advantageous when a constant λ_j is negative. We illustrate this with the example of the DC-DC converter. It has two switched modes, for which we have $\lambda_1 = -0.014215$ and $\lambda_2 = 0.142474$. In the case $\lambda_j < 0$, the associated formula $\delta_j(t)$ has the behavior of Figure 3.11 (a). In the case $\lambda_j > 0$, the associated formula $\delta_j(t)$ has the behavior of Figure 3.11 (b). In the case $\lambda_j < 0$, if the time sub-sampling is small enough, one can compute a sequence of balls with reducing radius, which makes the synthesis easier.

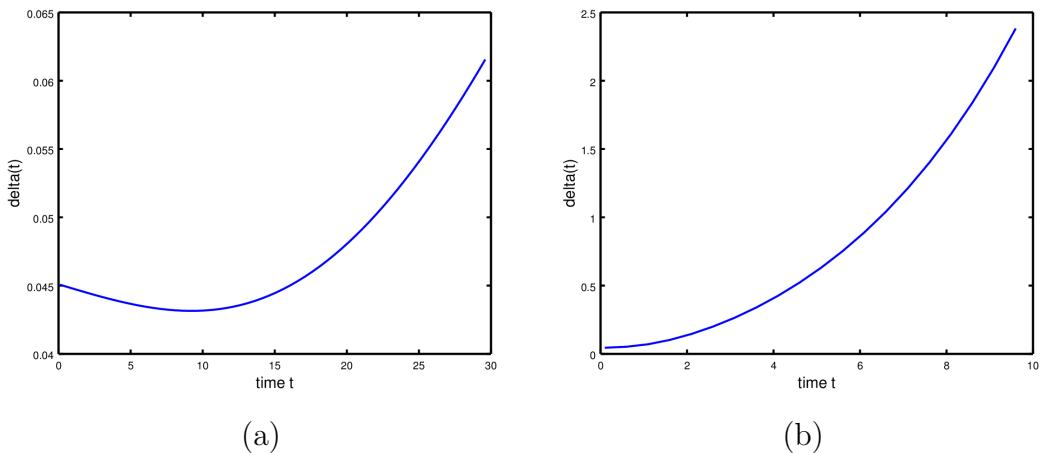


Figure 3.11: Behavior of $\delta_j(t)$ for the DC-DC converter with $\delta_j(0) = 0.045$. (a) Evolution of $\delta_1(t)$ (with $\lambda_1 < 0$); (b) Evolution of $\delta_2(t)$ (with $\lambda_2 > 0$).

In the following, we give the results obtained with our Octave implementation of this Euler-based method on 5 examples, and compare them with those given by the C++ implementation *DynIBEX* [5] of the Runge-Kutta based method used in Section 3.2.

Four-room apartment

We describe a first application on the 4-room 16-switch building ventilation case study adapted from [134], recalled in Appendix A.4. The model has been simplified in order to get constant parameters. To get constant parameters, we took $T_o = 30$, $T_c = 30$, $T_u = 17$, $\delta_{s_i} = 1$ for $i \in \mathcal{N}$. Compared simulations are given in Figure 3.12. On this example, the Euler-based method works better than *DynIBEX* in terms of CPU time.

DC-DC converter

This linear example is recalled in Appendix A.1.

	Euler	DynIBEX
R	$[20, 22]^2 \times [22, 24]^2$	
S	$[19, 23]^2 \times [21, 25]^2$	
τ	30	
Time subsampling	No	
Complete control	Yes	Yes
$\max_{j=1,\dots,16} \lambda_j$	-6.30×10^{-3}	
$\max_{j=1,\dots,16} C_j$	4.18×10^{-6}	
Number of balls/tiles	4096	252
Pattern length	1	1
CPU time	63 seconds	249 seconds

Table 3.3: Numerical results for the four-room example.

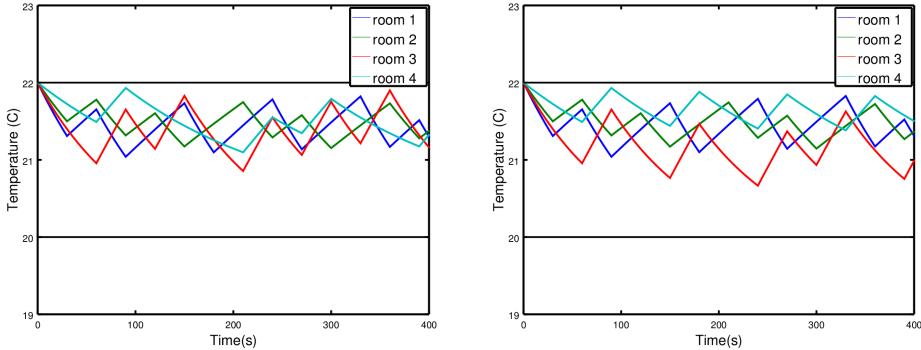


Figure 3.12: Simulation of the four-room case study with our synthesis method (left) and with the synthesis method of Section 3.2 (right).

On this example, the Euler-based method *fails* while *DynIBEX* succeeds rapidly.

Polynomial example

We consider the polynomial system taken from [126], recalled in Appendix A.3. The disturbances are not taken into account. The objective is to visit infinitely often *two* zones R_1 and R_2 , without going out of a safety zone S .

For Euler and *DynIBEX*, the table indicates *two* CPU times corresponding to the reachability from R_1 to R_2 and vice versa. On this example, the Euler-based method is much slower than *DynIBEX*.

Two-tank system

The two-tank system is a linear example taken from [89]. The system consists of two tanks and two valves. The first valve adds to the inflow of tank 1 and the second valve is a drain valve for tank 2. There is also a constant outflow from tank 2 caused by a pump. The system is linearized at a desired operating point. The objective

	Euler	DynIBEX
R	$[1.55, 2.15] \times [1.0, 1.4]$	
S	$[1.54, 2.16] \times [0.99, 1.41]$	
τ	0.5	
Complete control	No	Yes
λ_1	-0.014215	
λ_2	0.142474	
C_1	6.7126×10^{-5}	
C_2	2.6229×10^{-2}	
Number of balls/tiles	x	48
Pattern length	x	6
CPU time	x	~ 1 second

Table 3.4: Numerical results for the DC-DC converter example.

is to keep the water level in both tanks within limits using a discrete open/close switching strategy for the valves. Let the water level of tanks 1 and 2 be given by x_1 and x_2 respectively. The behavior of x_1 is given by $\dot{x}_1 = -x_1 - 2$ when the tank 1 valve is closed, and $\dot{x}_1 = -x_1 + 3$ when it is open. Likewise, x_2 is driven by $\dot{x}_2 = x_1$ when the tank 2 valve is closed and $\dot{x}_2 = x_1 - x_2 - 5$ when it is open. The dynamics of the system is recalled in Appendix A.7 On this example, the Euler-based method works better than *DynIBEX* in terms of CPU time.

Helicopter

The helicopter is a linear example taken from [55]. The problem is to control a quadrotor helicopter toward a particular position on top of a stationary ground vehicle, while satisfying constraints on the relative velocity. Let g be the gravitational constant, x (reps. y) the position according to x -axis (resp. y -axis), \dot{x} (resp. \dot{y}) the velocity according to x -axis (resp. y -axis), ϕ the pitch command and ψ the roll command. The possible commands for the pitch and the roll are the following: $\phi, \psi \in \{-10, 0, 10\}$. Since each mode corresponds to a pair (ϕ, ψ) , there are nine switched modes. The dynamics of the system is given by the equation:

$$\dot{X} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} X + \begin{pmatrix} 0 \\ g \sin(-\phi) \\ 0 \\ g \sin(\psi) \end{pmatrix}$$

where $X = (x \ \dot{x} \ y \ \dot{y})^\top$. Since the variables x and y are decoupled in the equations and follow the same equations (up to the sign of the command), it suffices to study the control for x (the control for y is the opposite). The dynamics of the system is recalled in Appendix A.8. On this example again, the Euler-based method works

	Euler	DynIBEX
R_1	$[-1, 0.65] \times [0.75, 1.75]$	
R_2	$[-0.5, 0.5] \times [-0.75, 0.0]$	
S	$[-2.0, 2.0] \times [-1.5, 3.0]$	
τ	0.15	
Time subsampling	$\tau/20$	
Complete control	Yes	Yes
λ_1	-1.5	
λ_2	-1.0	
λ_3	-1.1992×10^{-8}	
λ_4	-5.7336×10^{-6}	
C_1	641.37	
C_2	138.49	
C_3	204.50	
C_4	198.64	
Number of balls/tiles	16 & 16	1 & 1
Pattern length	8	7
CPU time	29 & 4203 seconds	0.1 & 329 seconds

Table 3.5: Numerical results for the polynomial example.

better than *DynIBEX* in terms of CPU time.

Analysis and comparison of results

This method presents a great advantage over the recent work [119]: no numerical integration is required for the control synthesis. The computations just require the evaluation of given functions f_j and (global error) functions δ_j at sampling times. The synthesis is thus *a priori* cheap compared to the use of numerical integration schemes (and even compared to exact integration for linear systems). However, most of the computation time is actually taken by the search for an appropriate radius δ of the balls B_i ($1 \leq i \leq m$) that cover R , and the search for appropriate patterns π_i that make the trajectories issued from B_i return to R .

Furthermore, the method lacks accuracy when the error bound $\delta_j(t)$ grows fast, this is particularly the case when $\lambda_j > 0$. A high number of balls may be required to counteract this drawback, as well as using time sub-sampling, and both increase the computational cost, but as seen on the helicopter example, it can still be cheaper than classical methods. Moreover, we can use the fact that some modes make the error grow, while others make it decrease, like in the two tank example. On systems for which the error does not grow fast, we perform very well as the computation of the image of a ball is very inexpensive. This is very often the case on thermal heating applications, for which the system usually has $\lambda_j < 0$ (see for example the

	Euler	DynIBEX
R	$[-1.5, 2.5] \times [-0.5, 1.5]$	
S	$[-3, 3] \times [-3, 3]$	
τ	0.2	
Time subsampling	$\tau/10$	
Complete control	Yes	Yes
λ_1	0.20711	
λ_2	-0.50000	
λ_3	0.20711	
λ_4	-0.50000	
C_1	11.662	
C_2	28.917	
C_3	13.416	
C_4	32.804	
Number of balls/tiles	64	10
Pattern length	6	6
CPU time	58 seconds	246 seconds

Table 3.6: Numerical results for the two-tank example.

four room case study).

Note that for systems presenting negative λ_j , if the sampling time is not imposed by the system, it is possible to choose an optimal sampling time minimizing the radius of the ball images (see Figure 3.11 (a)), and thus maximizing the chance of finding controllers fast.

The method presents a specific fault for synthesizing a controller for the DC-DC converter. Because we use balls to tile a box R , parts of some balls (crescent-shaped) are not included in the initial box, and these parts are particularly hard to steer inside R , because the dynamics of the system generates trajectories which are nearly horizontal. The fact that λ_2 is strictly positive makes it even harder to control these balls. This explains why we obtain controllable regions which look like Figure 3.13. Note that the same kind of results are obtained with state-of-the-art tools such as SCOTS [159] and PESSOA [132]. The use of zonotopes which perfectly tile the region R does not present this fault for this particular system.

We observe on the examples that the resulting control strategies synthesized by our method are quite different from those obtained by the Runge-Kutta method of Section 3.2 (which uses in particular rectangular tiles instead of balls). This may explain why the experimental results are here contrasted: Euler's method works better on 3 examples and worse on the 2 others. Besides the Euler method fails on one example (DC-DC converter) while *DynIBEX* succeeds on all of them. Note however that our Euler-based implementation is made of a few hundreds lines of in-

	Euler	DynIBEX
R	$[-0.3, 0.3] \times [-0.5, 0.5]$	
S	$[-0.4, 0.4] \times [-0.7, 0.7]$	
τ	0.1	
Time subsampling	$\tau/10$	
Complete control	Yes	Yes
λ_1	0.5	
λ_2	0.5	
λ_3	0.5	
C_1	1.77535	
C_2	0.5	
C_3	1.77535	
Number of balls/tiles	256	35
Pattern length	7	7
CPU time	539 seconds	1412 seconds

Table 3.7: Numerical results for the helicopter motion example.

terpreted code Octave while *DynIBEX* is made of around five thousands of compiled code C++.

3.3.6 Final remarks

We have given a new Euler-based method for controlling sampled switched systems, and compared it with the Runge-Kutta method of [115]. The method is remarkably simple and gives already promising results. In future work, we plan to explore the use of the *backward* Euler method instead of the forward Euler method used here (cf [32]). We plan also to give general sufficient conditions ensuring the convexity of the error function $\delta_j(\cdot)$; this would allow us to get rid of the convexity tests that we perform so far numerically for each pattern.

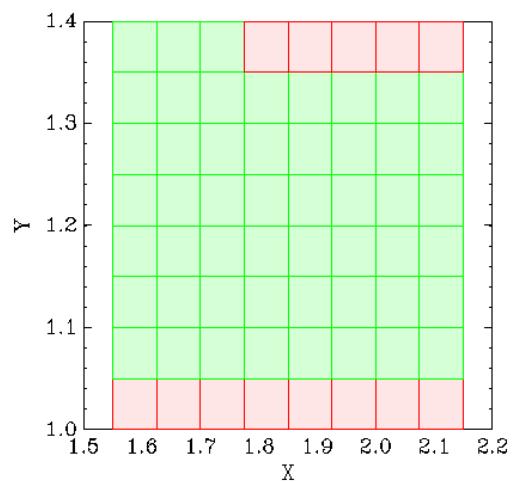


Figure 3.13: Controlled region of R using the Euler method for the DC-DC converter.

Chapter 4

Disturbances and distributed control

In this chapter, we extend the results of the previous chapter to systems subject to disturbances and varying parameters. We present how disturbances can be used to perform distributed (also called compositional) control synthesis, allowing to overcome the exponential complexity of the algorithms of Chapter 2. Provided that the modes do not affect each dimension of the system, system (2.1) can be rewritten as two sub-systems with independent control modes, but sharing some state variables. Those shared state variables can be viewed as disturbances, and using a method close to assume-guarantee reasoning [12, 34, 41, 108], we synthesize two controllers, much cheaper to compute than a centralized one. This distributed approach is applied with sets represented by zonotopes and balls, and made available for nonlinear systems using Runge-Kutta and Euler schemes.

This chapter is divided as follows. We present some results for linear systems subject to disturbances using zonotopes in Section 4.1. We introduce a backward reachability procedure relying on zonotopes and apply it in a centralized and distributed manner in Section 4.2. We then present in Section 4.3 an approach relying on a notion close to incremental input-to-state stability [13] which, associated to an Euler scheme and balls of \mathbb{R}^n , allows to handle perturbations and varying parameters, and can thus be applied to distributed synthesis.

4.1 Linear systems and disturbances

Let us consider an affine system satisfying

$$\dot{x} = Ax + b \quad (4.1)$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, and $b \in \mathbb{R}^n$. As seen in the previous chapter, one can compute the solution at time $t > 0$ of (4.1) using equation (3.1). Being given a sampling time τ (taken equal to 1 for the sake of simplicity), system (4.1) can be turned into a discrete time system

$$x(t+1) = Cx(t) + d \quad (4.2)$$

with $C = e^A$ and $d = \int_0^1 e^{A(t-1)} dt$. System (4.2) can be decomposed in blocks as follows:

$$\begin{pmatrix} \dot{x}_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}. \quad (4.3)$$

where $x_1, d_1 \in \mathbb{R}^{n_1}$ and $x_2, d_2 \in \mathbb{R}^{n_2}$ with $n = n_1 + n_2$, and $C_{11}, C_{12}, C_{21}, C_{22}$ matrices of appropriate dimensions. Let us now consider an initial set given as a zonotope

$$Z = \left\langle \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}, \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \right\rangle,$$

with $c_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$, $G_1 \in \mathbb{R}^{n_1 \times n}$ and $G_2 \in \mathbb{R}^{n_2 \times n}$. We know that the image at time $t + 1$ of Z is the zonotope

$$Z' = \langle \begin{pmatrix} C_{11}c_1 + C_{12}c_2 + d_1 \\ C_{21}c_1 + C_{22}c_2 + d_2 \end{pmatrix}, \begin{pmatrix} C_{11}G_1 + C_{12}G_2 \\ C_{21}G_1 + C_{22}G_2 \end{pmatrix} \rangle.$$

We thus have $x_1(t + 1) \in Z'_1 = \langle C_{11}c_1 + C_{12}c_2 + d_1, C_{11}G_1 + C_{12}G_2 \rangle$. Now, assume that x_2 stays in a safety zone S_2 given as a zonotope $\langle s_2, F_2 \rangle$, we have

$$x_1(t + 1) \in Z'_1 = \langle C_{11}c_1 + C_{12}s_2 + d_1, C_{11}G_1 + C_{12}F_2 \rangle. \quad (4.4)$$

We can then compute a bounding box of the latter, such as in [68], given as a zonotope $Z_1^+ = \square(Z'_1)$ of the form $\langle c'_1, G'_1 \rangle$ with $G'_1 \in \mathbb{R}^{n_1 \times n_1}$. The same can be done for component two, a bounding zonotope $Z_2^+ = \square(Z'_2)$ of Z'_2 of the form $\langle c'_2, G'_2 \rangle$ with $G'_2 \in \mathbb{R}^{n_2 \times n_2}$ can be inferred, assuming that component 1 stays in a safety zone S_1 . This now gives an overapproximation $Z_1^+ \times Z_2^+$ of zonotope Z' .

We can then iterate this, by computing $Z_1^{++} = \square((Z_1^+)')$ as an overapproximation of the image of Z_1^+ , assuming that component 2 stays in the safety zone S_2 , and reciprocally for component 2, we obtain $Z_2^{++} = \square((Z_2^+)')$. We thus have $Z_1^{++} \times Z_2^{++}$ as an overapproximation of Z'' , and we now see the main interest: each component only has to know its state. When computing images Z_1^+ , Z_1^{++} , the state of component 2 is overapproximated by S_2 , and reciprocally. Assuming that x_1 forever stays in S_1 , and x_2 forever stays in S_2 , the successive images can be computed separately for each component.

Assuming that x_1 and x_2 forever stay in their respective safety zones S_1 and S_2 , this actually gives a way to successively compute over-approximations $Z_1^+ \times Z_2^+$, $Z_1^{++} \times Z_2^{++}$, ... of the images Z' , Z'' , ..., of the zonotope Z , by only looking at component 1 and component 2 separately.

If we now take a switched version of (4.2) (by adding an index $j \in U$ to matrix A and vector b), the previous approach allows to separately compute two controllers for both components. This however requires that both components stay in a given safety zone. In other words, one has to successfully compute two safety controllers, for both components, for this method to work. Actually, safety properties are mandatory to apply such distributed methods, we find them in several compositional or assume-guarantee based methods [53, 101, 135].

Using this distributed method, component 2 can actually be seen as a bounded perturbation for component 1, where the perturbation is bounded in S_2 . We could in fact extend this method to more general perturbations, for systems of the form

$$\dot{x} = Ax + Bw + b \quad (4.5)$$

where w is the bounded perturbation (varying in a given set within time). Note that [109] proposes a subtle approach to extend this type of calculations to a wider range of perturbations, notably including varying parameters.

In the following, we apply this method in an iterated manner, first in a discrete-time framework, before applying it to continuous-time systems.

4.2 Distributed control using zonotopes

In this section, we first focus on discrete-time systems and present an approach mainly aimed at controlling building heating applications. We introduce an extension of the algorithm of Chapter 2 allowing to perform iterated (backward) reachability. We then extend it to distributed synthesis, by introducing a state over-approximation technique which avoids the use of non-local information by the subsystem controllers. This procedure allowed to synthesize a controller for a real case study of temperature control in a building with 11 rooms and $2^{11} = 2048$ switching modes of control. This approach is then extended to continuous-time systems using Runge-Kutta schemes and the DynIBEX library.

4.2.1 State-dependent Switching Control

We first consider the *discrete-time* setting. The time t then takes its values in \mathbb{N} .

Control modes

Consider the following discrete-time system with *finite control*:

$$x_1(t+1) = f_1(x_1(t), x_2(t), u_1) \quad x_2(t+1) = f_2(x_1(t), x_2(t), u_2)$$

where x_1 (resp. x_2) is the first (resp. second) component of the state vector, and takes its values in \mathbb{R}^{n_1} (resp. \mathbb{R}^{n_2}), and where u_1 (resp. u_2) is the first (resp. second) component of the control *mode*, and takes its values in the *finite* set U_1 (resp. U_2). We will often write x for (x_1, x_2) , u for (u_1, u_2) , and n for $n_1 + n_2$. We will also abbreviate the set $U_1 \times U_2$ as U . Let N_1 (resp. N_2) by the cardinality of U_1 (resp. U_2), and $N = N_1 \cdot N_2$ be the cardinality of U .

More generally, we abbreviate the discrete-time system under the form:

$$x(t+1) = f(x(t), u)$$

where x is a vector state variable, taking its values in $\mathbb{R}^n = \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, and where u is of the form (u_1, u_2) , where u_1 takes its values in U_1 and u_2 in U_2 .

In this context, we are interested by the following *centralized* control-synthesis problem: at each discrete-time t , select some appropriate mode $u \in U$ in order to satisfy a given property. In a *distributed* setting, the control-synthesis problem consists in selecting the value of u_1 in U_1 according to the value of $x_1(t)$ *only*, and the value of u_2 in U_2 according to the value of $x_2(t)$ *only*.

The properties that we consider are *reachability* properties: given a set S and a set R , we look for a control which steers any element of S into R in a bounded

number of steps. We also consider *stability* properties, requiring that once the state x of the system is in R at time t , the control will maintain it in R indefinitely. Actually, given a state set R , we will present a method that does not start from a given set S , but constructs it, together with a control that steers all the elements of S to R within a bounded number of steps (S can be seen as a “capture set” of R).

In this section, we consider that R and S are “rectangles” of the state space. More precisely, $R = R_1 \times R_2$ is a rectangle of reals, i.e., R is a product of n closed intervals of reals, and R_1 (resp. R_2) is a product of n_1 (resp. n_2) closed intervals of reals. Likewise, we assume that $S = S_1 \times S_2$ is a rectangular sub-area of the state space.

Example 1. *The centralized and distributed approaches will be illustrated by the example of a two-room apartment, heated by one heater in each room (adapted from [76]). In this example, the objective is to control the temperature of both rooms. There is heat exchange between the two rooms and with the environment. The continuous dynamics of the system is given by the equation:*

$$\begin{pmatrix} \dot{T}_1 \\ \dot{T}_2 \end{pmatrix} = \begin{pmatrix} -\alpha_{21} - \alpha_{e1} - \alpha_f u_1 & \alpha_{21} \\ \alpha_{12} & -\alpha_{12} - \alpha_{e2} - \alpha_f u_2 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} + \begin{pmatrix} \alpha_{e1} T_e + \alpha_f T_f u_1 \\ \alpha_{e2} T_e + \alpha_f T_f u_2 \end{pmatrix}.$$

Here T_1 and T_2 are the temperatures of the two rooms, and the state of the system corresponds to $T = (T_1, T_2)$. The control mode variable u_1 (respectively u_2) can take the values 0 or 1, depending on whether the heater in room 1 (respectively room 2) is switched off or on (hence $U_1 = U_2 = \{0, 1\}$). Hence, here $n_1 = n_2 = 1$, $N_1 = N_2 = 2$, and $n = 2$ and $N = 4$.

Temperature T_e corresponds to the temperature of the environment, and T_f to the temperature of the heaters. The values of the different parameters are as follows: $\alpha_{12} = 5 \times 10^{-2}$, $\alpha_{21} = 5 \times 10^{-2}$, $\alpha_{e1} = 5 \times 10^{-3}$, $\alpha_{e2} = 5 \times 10^{-3}$, $\alpha_f = 8.3 \times 10^{-3}$, $T_e = 10$ and $T_f = 35$. The dynamics of the system is recalled in Appendix A.2.

We suppose that the heaters can be switched periodically at sampling instants τ , $2\tau, \dots$ (here, $\tau = 5s$). By integration of the continuous dynamics between t and $t+\tau$, the system can be easily put under the desired discrete-time form:

$$T_1(t+1) = f_1(T_1(t), T_2(t), u_1) \quad T_2(t+1) = f_2(T_1(t), T_2(t), u_2)$$

where f_1 and f_2 are affine functions.

Given an objective rectangle for $T = (T_1, T_2)$ of the form $R = [18.5, 22] \times [18.5, 22]$, the control synthesis problem is to find a rectangular capture set S (as large as possible) from which one can steer the state T to R (“reachability”), and then maintain T within R for ever (“stability”).

Control patterns

It is often easier to design a control of the system using several applications of f in a row rather than using just a single application of f at each time. We are

thus led to the notion of “macro-step”, and “control pattern”. A (*control*) *pattern* $\pi = (\pi_1, \pi_2)$ of length k is a sequence of modes defined recursively by:

1. π is of the form $(u_1, u_2) \in U_1 \times U_2$ if $k = 1$,
2. π is of the form $(u_1 \cdot \pi'_1, u_2 \cdot \pi'_2)$, where u_1 (resp. u_2) is in U_1 (resp. U_2), and (π'_1, π'_2) is a (control) pattern of length $k - 1$ if $k \geq 2$.

The set of patterns of length k is denoted by Π^k (for length $k = 1$, we have $\Pi^1 = U$). Likewise, for $k \geq 1$, we denote by Π_1^k (resp. Π_2^k) the set of sequences of k elements of U_1 (resp. U_2).

For a system defined by $x(t+1) = f(x(t), (u_1, u_2))$ and a pattern $\pi = (\pi_1, \pi_2)$ of length k , one can recursively define $x(t+k) = f(x(t), (\pi_1, \pi_2))$ with $(\pi_1, \pi_2) \in \Pi^k$, by:

1. $f(x(t), (\pi_1, \pi_2)) = f(x(t), (u_1, u_2))$, if (π_1, π_2) is a pattern of length $k = 1$ of the form $(u_1, u_2) \in U$,
2. $f(x(t), (\pi_1, \pi_2)) = f(f(x(t), (\pi'_1, \pi'_2)), (u_1, u_2))$, if (π_1, π_2) is a pattern of length $k \geq 2$ of the form $(u_1 \cdot \pi'_1, u_2 \cdot \pi'_2)$ with $(u_1, u_2) \in U$ and $(\pi'_1, \pi'_2) \in \Pi^{k-1}$.

One defines $(f(x, \pi))_1 \in \mathbb{R}^{n_1}$ and $(f(x, \pi))_2 \in \mathbb{R}^{n_2}$ to be the first and second components of $f(x, \pi) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} = \mathbb{R}^n$, i.e: $f(x, \pi) = ((f(x, \pi))_1, f(x, \pi)_2)$.

In the following, we fix an upper bound $K \in \mathbb{N}$ on the length of patterns. The value of K can be seen as a maximum number of time steps, for which we compute the future behaviour of the system (“horizon”). We denote by $\Pi_1^{\leq K}$ (resp. $\Pi_2^{\leq K}$) the expression $\bigcup_{1 \leq k \leq K} \Pi_1^k$ (resp. $\bigcup_{1 \leq k \leq K} \Pi_2^k$). Likewise, we denote by $\Pi^{\leq K}$ the expression $\bigcup_{1 \leq k \leq K} \Pi^k$.

4.2.2 Control synthesis using tiling

Tiling

Let $R = R_1 \times R_2$ be a rectangle. We say that \mathcal{R} is a (*finite rectangular*) *tiling* of R if \mathcal{R} is of the form $\{r_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$, where I_1 and I_2 are given finite sets of positive integers, each r_{i_1, i_2} is a sub-rectangle of R of the form $r_{i_1} \times r_{i_2}$, and r_{i_1}, r_{i_2} are closed sub-intervals of R_1 and R_2 respectively. Besides, we have $\bigcup_{i_1 \in I_1} r_{i_1} = R_1$ and $\bigcup_{i_2 \in I_2} r_{i_2} = R_2$ (Hence $R = \bigcup_{i_1 \in I_1, i_2 \in I_2} r_{i_1, i_2}$).

We will refer to r_{i_1}, r_{i_2} and r_{i_1, i_2} as “tiles” of R_1 , R_2 and R respectively. The same notions hold for rectangle S .

In the centralized context, given a rectangle R , the *macro-step (backward reachability) control synthesis problem with horizon K* consists in finding a rectangle S and a tiling $\mathcal{S} = \{s_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$ of S such that, for each $(i_1, i_2) \in I_1 \times I_2$, there exists $\pi \in \Pi^{\leq K}$ such that:

$$f(s_{i_1, i_2}, \pi) \subseteq R$$

(i.e., for all $x \in s_{i_1, i_2}$: $f(x, \pi) \in R$). This is illustrated in Figure 4.1.

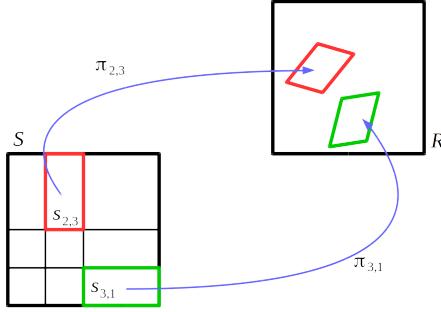


Figure 4.1: Mapping of tile $s_{2,3}$ to R via pattern $\pi_{2,3}$, and mapping of tile $s_{3,1}$ via $\pi_{3,1}$.

Parametric extension of tiling

In the following, we assume that the set S we are looking for is a *parametric extension* of R , denoted by $R + (a, a)$, which is defined in the following.

Suppose that $R = R_1 \times R_2$ is given as well as a tiling $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2 = \{r_{i_1} \times r_{i_2}\}_{i_1 \in I_1, i_2 \in I_2} = \{r_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$. Then R_1 can be seen as a product of n_1 closed intervals of the form $[\ell, m]$. Consider a nonnegative real parameter a . Let $(R_1 + a)$ denote the corresponding product of n_1 intervals of the form $[\ell - a, m + a]$.¹ We define $(R_2 + a)$ similarly. Finally, we define $R + (a, a)$ as $(R_1 + a) \times (R_2 + a)$.

We now consider that S is a (parametric) superset of R of the form $R + (a, a)$. We define a tiling $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2$ of S of the form $\{s_{i_1} \times s_{i_2}\}_{i_1 \in I_1, i_2 \in I_2}$, which is obtained from $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2 = \{r_{i_1} \times r_{i_2}\}_{i_1 \in I_1, i_2 \in I_2}$ by a simple extension, as follows: A tile r_{i_1} (resp. r_{i_2}) of \mathcal{R}_1 (resp. \mathcal{R}_2) in “contact” with ∂R_1 (resp. ∂R_2) is extended as a tile s_{i_1} (resp. s_{i_2}) in order to be in contact with $\partial(R_1 + a)$ (resp. $\partial(R_2 + a)$); a tile “interior” to R_1 (i.e., with no contact with ∂R_1) is kept unchanged, and coincides with s_{i_1} , and similarly for R_2 .

We denote the resulting tiling \mathcal{S} by $\mathcal{R} + (a, a)$. We also denote s_{i_1} (resp. s_{i_2}) by $r_{i_1} + a$ (resp. $r_{i_2} + a$), even if r_{i_1} (resp. r_{i_2}) is “interior” to R_1 (resp. R_2). Likewise, we denote $s_{i,j}$ by $r_{i,j} + (a, a)$. Note that a tiling of R of index set $I_1 \times I_2$ induces a tiling of $R + (a, a)$ with the same index set $I_1 \times I_2$, hence the same number of tiles as R , for any $a \geq 0$. This is illustrated in Figure 4.2, where the tiling of R is represented with black continuous lines, and the extended tiling of $R + (a, a)$ with red dashed lines.

Generate-and-test tilings

By replacing S with $R + (a, a)$ in the notions defined in Section 4.2.2 the problem of macro-step control synthesis can now be reformulated as: “find a tiling \mathcal{R} of R that induces a macro-step control of $R + (a, a)$ towards R , for some $a \geq 0$ (as large

1. Actually, we will consider in the examples that $(R_1 + a)$ is a product of intervals of the form $[\ell - a, m]$ where the interval is extended only at its *lower* end, but the method is strictly identical.

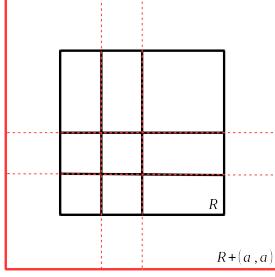


Figure 4.2: Tiling of $R + (a, a)$ induced by tiling \mathcal{R} of R .

as possible”.

This problem can be solved by a simple “generate-and-test” procedure: we *generate* a candidate tiling, and then *test* if it satisfies the control property (the control test procedure is explained in Section 4.2.3); if the test fails, we generate another candidate, and so on iteratively.

In practice, the generation of a candidate \mathcal{R} is performed by starting from the trivial tiling (made of one tile equal to R), and using successive *bisections* of R until, either the control test succeeds (“success”), or the depth of bisection of the new candidate is greater than a given upper bound D (“failure”). See more details in [67].

Tiling refinement

Let us now explain how we find a tiling \mathcal{R} of R such that $\Pi_{i_1, i_2} \neq \emptyset$. We focus on the centralized case, but the distributed case is similar. We start from the trivial tiling $\mathcal{R}^0 = \{R\}$, which only contains tile R . If $f(R, \pi) \subseteq R$ for some $\pi \in \Pi^{\leq K}$, then \mathcal{R}^0 is the desired tiling. Otherwise, we refine \mathcal{R}^0 by *bisection*, which gives a tiling \mathcal{R}^1 of the form $\{r_{(i,1),(j,2)}\}_{1 \leq i,j \leq n}$. If, for all $1 \leq i, j \leq n$ there exists some $\pi \in \Pi^{\leq K}$ such that $f(r_{(i,1),(j,2)}, \pi) \subseteq R$, then \mathcal{R}^1 is the desired tiling. Otherwise, there exist some “bad” tiles of the form $r_{(i,1),(j,2)}$ with $1 \leq i, j \leq n$ such that $\forall \pi \in \Pi^{\leq K} f(r_{(i,1),(j,2)}, \pi) \not\subseteq R$; we then transform \mathcal{R}^1 into \mathcal{R}^2 by bisecting all those bad tiles. By iterating this procedure, we produce tilings $\mathcal{R}^1, \mathcal{R}^2, \dots, \mathcal{R}^d$, until either no bad tiles remain in \mathcal{R}^d (*success*), or the bisection depth d is greater than the given upper bound D (*failure*).

Iterated macro-step control synthesis

Suppose that we are given an objective rectangle $R = R_1 \times R_2$. If the one-step control synthesis described in Section 4.2.2 succeeds, then there is a nonnegative real $a^{(1)} = A$ and a tiling \mathcal{R} of R that induces a control steering all the points of $R^{(1)} = R + (a^{(1)}, a^{(1)})$ to R in one step. Now the macro-step control synthesis can be reapplied to $R^{(1)}$. If it succeeds again, then it produces a tiling $\mathcal{R}^{(1)}$ of $R^{(1)}$ which induces a control that steers $R^{(2)} = R^{(1)} + (a^{(2)}, a^{(2)})$ to $R^{(1)}$ for some $a^{(2)} \geq 0$. The

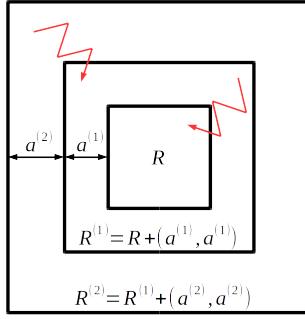


Figure 4.3: Iterated control of $R^{(1)} = R + (a^{(1)}, a^{(1)})$ towards R , and $R^{(2)} = R^{(1)} + (a^{(2)}, a^{(2)})$ towards $R^{(1)}$.

iterated application of macro-step control synthesis outputs a sequence of tilings $\mathcal{R}^{(i)}$, each of which induces a control that steers $R^{(i+1)} = R + (\sum_{j=1}^{i+1} a^{(j)}, \sum_{j=1}^{i+1} a^{(j)})$ to $R^{(i)}$. In the end, this synthesizes a control that steers $R^{(i+1)}$ to R in at most $i + 1$ macro-steps ($i \geq 0$), using an increasing sequence of nested rectangles around R . This is illustrated in Figure 4.3, for $i = 1$.

The iteration process halts at some step, say m , when the last macro-step control synthesis fails because the maximum bisection depth D is reached while “bad” tiles still remain (see Section 4.2.2). We also stop the process when the last macro-step control synthesis outputs a real $a^{(m)}$ which is smaller than a given bound: this is because the sequence of controllable rectangles around R seems to approach a limit.

Remark 5. Note that, if the generate-and-test process stops with “success” for a tiling \mathcal{R} , then the tiling $\mathcal{R}_{D,\text{uniform}}$ also solves the problem, where $\mathcal{R}_{D,\text{uniform}}$ is the “finest” tiling obtained by bisecting D times all the n components of R . Since $\mathcal{R}_{D,\text{uniform}}$ has exactly 2^{nD} tiles, it is in general impractical to perform directly the control test on it. From a theoretical point of view however, it is convenient to suppose that $\mathcal{R} = \mathcal{R}_{D,\text{uniform}}$ for reducing the worst case time complexity of the control synthesis procedure to the complexity of the control test part only (see Section 4.2.3).

4.2.3 Centralized control

Tiling test procedure

As seen in Section 4.2.2, the (macro-step) control synthesis problem with horizon K consists in finding $a \geq 0$ (as big as possible), and a tiling $\mathcal{R} = \{r_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$ of R such that, for each $(i_1, i_2) \in I_1 \times I_2$, there exists some $\pi \in \Pi^{\leq K}$ with

$$f(r_{i_1, i_2} + (a, a), \pi) \subseteq R. \quad (4.6)$$

It is easy to see that if (4.6) holds for some $a \geq 0$, then it also holds for all $a' \leq a$. In order to *test* if a tiling candidate $\mathcal{R} = \{r_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$ of R satisfies the desired

property, we define, for each $(i_1, i_2) \in I_1 \times I_2$:

$$\Pi_{i_1, i_2}^{\leq K} = \{\pi \in \Pi^{\leq K} \mid f(r_{i_1, i_2}, \pi) \subseteq R\}. \quad (4.7)$$

Suppose that $\Pi_{i_1, i_2}^{\leq K} \neq \emptyset$. Then we know that Formula (4.6) is satisfied for $a = 0$. In order to find a “as large as possible”, we look for the existence of a pattern π such that Formula (4.6) holds also for $a = \frac{|R|}{100}$ and $a = \frac{|R|}{10}$, where $|R|$ denotes the length of the smallest side of rectangle R . Numerous variants of such tests are of course possible, but such a simple test works well in practice, and we keep it here for the sake of simplicity. When $\Pi_{i_1, i_2}^{\leq K} \neq \emptyset$, we thus define:

$$a_{i_1, i_2} = \max\{a \in \{0, \frac{|R|}{100}, \frac{|R|}{10}\} \mid \exists \pi \in \Pi^{\leq K} \ f(r_{i_1, i_2} + (a, a), \pi) \subseteq R\}.$$

Suppose that, for all $(i_1, i_2) \in I_1 \times I_2$: $\Pi_{i_1, i_2}^{\leq K} \neq \emptyset$, and let $A = \min_{(i_1, i_2) \in I_1 \times I_2} \{a_{i_1, i_2}\}$. It is easy to see that, for all $(i_1, i_2) \in I_1 \times I_2$, there exists a pattern, denoted by π_{i_1, i_2} , such that: $f(r_{i_1, i_2} + (A, A), \pi_{i_1, i_2}) \subseteq R$.

Proposition 3. *Suppose that there exists a tiling $\mathcal{R} = \{r_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$ of R such that:*

$$\forall (i_1, i_2) \in I_1 \times I_2 \quad \Pi_{i_1, i_2}^{\leq K} \neq \emptyset.$$

Then \mathcal{R} induces a macro-step control of horizon K of $R + (A, A)$ towards R with:

$$\forall (i_1, i_2) \in I_1 \times I_2 : \quad f(r_{i_1, i_2} + (A, A), \pi_{i_1, i_2}) \subseteq R$$

where A and π_{i_1, i_2} are defined as above.

For each tile r_{i_1, i_2} of R and each $\pi \in \Pi^{\leq K}$, the test of inclusion $f(r_{i_1, i_2}, \pi) \subseteq R$ can be achieved in time polynomial in n when f is affine. Hence the test $\Pi_{i_1, i_2}^{\leq K} \neq \emptyset$ can be done in $O(N^K \cdot n^\alpha)$ since $\Pi^{\leq K}$ contains $O(N^K)$ elements. The computation time of $\{a_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$, π_{i_1, i_2} , and A is thus in $O(N^K \cdot 2^{nD})$, where D is the maximal bisection depth. Hence the complexity of testing a candidate tiling \mathcal{R} is in $O(N^K \cdot 2^{nD})$. By Remark 5 above, the running time of the control synthesis by the generate-and-test procedure is also in $O(N^K \cdot 2^{nD})$.

Once a candidate tiling \mathcal{R} satisfying the control test property is found, the generate-and-test procedure ends with *success* (see Section 4.2.2), and a set $S = R + (a^{(1)}, a^{(1)})$ with $a^{(1)} = A$ has been found. One can then *iterate* the “generate-and-test” procedure in order to construct an increasing sequence of nested rectangles of the form $R + (a^{(1)}, a^{(1)})$, $R + (a^{(1)} + a^{(2)}, a^{(1)} + a^{(2)})$, \dots , which can all be driven to R . The process ends at the first step $i \geq 1$ for which $a^{(i)} = 0$ (no proper extension of the current rectangle has been found).

Example 2. Consider the specification of a two-room apartment given in Example 1 and Appendix A.2. Set $R = [18.5, 22] \times [18.5, 22]$. Let $D = 1$ (the depth of bisection is at most 1), and $K = 4$ (the maximum length of patterns is 4). We

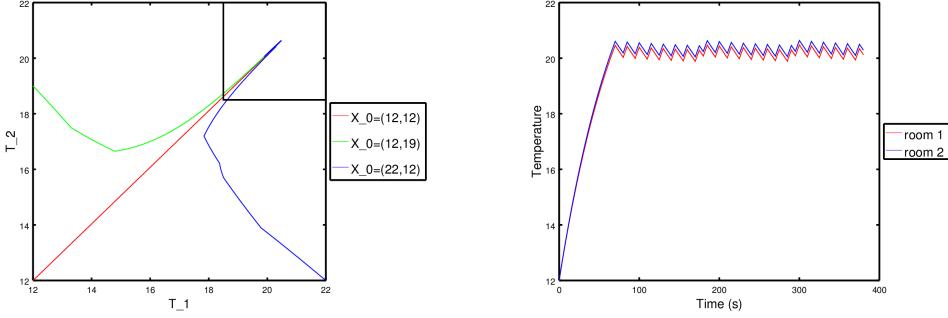


Figure 4.4: Simulations of the centralized reachability controller for three different initial conditions plotted in the state space plane (left); simulation of the centralized reachability controller for the initial condition $(12, 12)$ plotted within time (right).

look for a centralized controller which will steer the rectangle $S = [18.5 - a, 22] \times [18.5 - a, 22]$ to R with a as large as possible, and stay in R indefinitely. Using our implementation, the computation of the control synthesis takes 4.14s of CPU time.

The method iterates successfully 15 times the macro-step control synthesis procedure. We find $S = R + (a, a)$ with $a = 53.5$, i.e. $S = [-35, 22] \times [-35, 22]$. This means that any element of S can be driven to R within 15 macro-steps of length (at most) 4, i.e., within $15 \times 4 = 60$ units of time. Since each unit of time is of duration $\tau = 5$ s, any trajectory starting from S reaches R within $60 \times 5 = 300$ s. Once the trajectory $x(t)$ is in R , it returns in R every macro-step of length (at most) 4, i.e., every $4 \times 5 = 20$ s.

These results are consistent with the simulation given in Figure 4.4 for the time evolution of (T_1, T_2) starting from $(12, 12)$. Simulations of the control, starting from $(T_1, T_2) = (12, 12)$, $(T_1, T_2) = (12, 19)$ and $(T_1, T_2) = (22, 12)$ are also given in the state space plane in Figure 4.4.

Stability as a special case of reachability

Instead of looking for a set of the form $S = R + (a, a)$ from which R is reachable via a macro-step, let us consider the particular case where $S = R$ (i.e., $a = 0$).

The problem now consists in constructing a tiling $\mathcal{R} = \{r_{i_1, i_2}\}_{i_1 \in I_1, i_2 \in I_2}$ of R such that, for all $(i_1, i_2) \in I_1 \times I_2$, there exists a pattern $\pi_{i_1, i_2} \in \Pi^{\leq K}$ ensuring $f(r_{i_1, i_2}, \pi_{i_1, i_2}) \subseteq R$. If such a tiling \mathcal{R} exists, then² $x(t) \in R$ implies $x(t+k) \in R$ for some $k \leq K$. Actually, we can slightly modify the procedure in order to additionally impose that for some $\varepsilon > 0$, it holds $x(t+k') \in R + (\varepsilon, \varepsilon)$ for any $k' = 1, \dots, k-1$ (see Section 4.2.4). It follows that R is “stable” (with tolerance ε) under the control induced by \mathcal{R} . We can thus treat the stability control of R as a special case of reachability control.

2. If $x(t) \in R$, then $x(t) \in r_{i,j}$ for some $(i, j) \in I_1 \times I_2$, hence $x(t+k) = f(x, \pi_{i,j}) \in R$ for some $k \leq K$.

4.2.4 Distributed control

Background

In the distributed context, given a set $R = R_1 \times R_2$, the (*macro-step*) *distributed control synthesis problem with horizon K* consists in finding $a \geq 0$, and a tiling $\mathcal{R}_1 = \{r_{i_1}\}_{i_1 \in I_1}$ of R_1 which induces a (macro-step) control on $R_1 + a$, a tiling $\mathcal{R}_2 = \{r_{i_2}\}_{i_2 \in I_2}$ which induces a (macro-step) control on $R_2 + a$.

More precisely, we seek tilings \mathcal{R}_1 and \mathcal{R}_2 such that: there exists $\ell \in \mathbb{N}$ such that, for each $i_1 \in I_1$ there exists a pattern π_1 of ℓ modes in U_1 , and for each $i_2 \in I_2$, a pattern π_2 of ℓ modes in U_2 such that:

$$f((r_{i_1} + a) \times (R_2 + a), (\pi_1, \pi_2))|_1 \subseteq R_1 \wedge f((R_1 + a) \times (r_{i_2} + a), (\pi_1, \pi_2))|_2 \subseteq R_2.$$

In order to synthesize a *distributed* strategy where the control pattern π_1 is determined only by i_1 (regardless of the value of i_2), and the control pattern π_2 only by i_2 (regardless of the value of i_1), we now define an *over-approximation* $X_{i_1}(a, \pi_1)$ for $f((r_{i_1} + a) \times (R_2 + a), (\pi_1, \pi_2))|_1$, and an *over-approximation* $X_{i_2}(a, \pi_2)$ for $f((R_1 + a) \times (r_{i_2} + a), (\pi_1, \pi_2))|_2$. The correctness of these over-approximations relies on the existence of a fixed positive value for parameter ε . Intuitively, ε represents the width of the additional margin (around $R + (a, a)$) within which all the intermediate states lie when a macro-step is applied to a point of $R + (a, a)$.

Tiling test procedure

Let π_1^k (resp. π_2^k) denote the prefix of length k of π_1 (resp. π_2), and $\pi_1(k)$ (resp. $\pi_2(k)$) the k -th element of pattern π_1 (resp. π_2).

Definition 8. Consider an element r_{i_1} (resp. r_{i_2}) of a tiling \mathcal{R}_1 (resp. \mathcal{R}_2) of R_1 (resp. R_2), and a pattern $\pi_1 \in \Pi_1^{\leq K}$ (resp. $\pi_2 \in \Pi_2^{\leq K}$) of length ℓ_1 (resp. ℓ_2). The approximate first-component (resp. second-component) sequence $\{X_{i_1}^k(a, \pi_1)\}_{0 \leq k \leq \ell_1}$ (resp. $\{X_{i_2}^k(a, \pi_2)\}_{0 \leq k \leq \ell_2}$) is defined as follows:

- $X_{i_1}^0(a, \pi_1) = r_{i_1} + a$ (resp. $X_{i_2}^0(a, \pi_2) = r_{i_2} + a$) and
- $X_{i_1}^k(a, \pi_1) = f_1(X_{i_1}^{k-1}(a, \pi_1), R_2 + a + \varepsilon, \pi_1(k))$ for $1 \leq k \leq \ell_1$ (respectively $X_{i_2}^k(a, \pi_2) = f_2(R_1 + a + \varepsilon, X_{i_2}^{k-1}(a, \pi_2), \pi_2(k))$ for $1 \leq k \leq \ell_2$).

We define the property $Prop_1(a, i_1, \pi_1)$ of $\{X_{i_1}^k(a, \pi_1)\}_{0 \leq k \leq \ell_1}$ by:

$$X_{i_1}^k(a, \pi_1) \subseteq R_1 + a + \varepsilon \text{ for } 1 \leq k \leq \ell_1 - 1, \text{ and } X_{i_1}^{\ell_1}(a, \pi_1) \subseteq R_1.$$

Likewise, we define the property $Prop_2(a, i_2, \pi_2)$ of $\{X_{i_2}^k(a, \pi_2)\}_{0 \leq k \leq \ell_2}$ by:

$$X_{i_2}^k(a, \pi_2) \subseteq R_2 + a + \varepsilon \text{ for } 1 \leq k \leq \ell_2 - 1, \text{ and } X_{i_2}^{\ell_2}(a, \pi_2) \subseteq R_2.$$

Figure 4.5 illustrates property $Prop_1(a, i_1, \pi_1)$ for $\pi_1 = (u_1 \cdot v_1)$, $\ell_1 = 2$ and a given tile r_{i_1} with $i_1 \in I_1$: $Prop_1(a, i_1, \pi_1)$ is satisfied because $X_1^1(a, \pi_1) \subseteq R_1 + a + \varepsilon$ and $X_1^2(a, \pi_1) \subseteq R_1$ are true.

Suppose now that there exist ℓ_1 and ℓ_2 ($1 \leq \ell_1, \ell_2 \leq K$) such that:

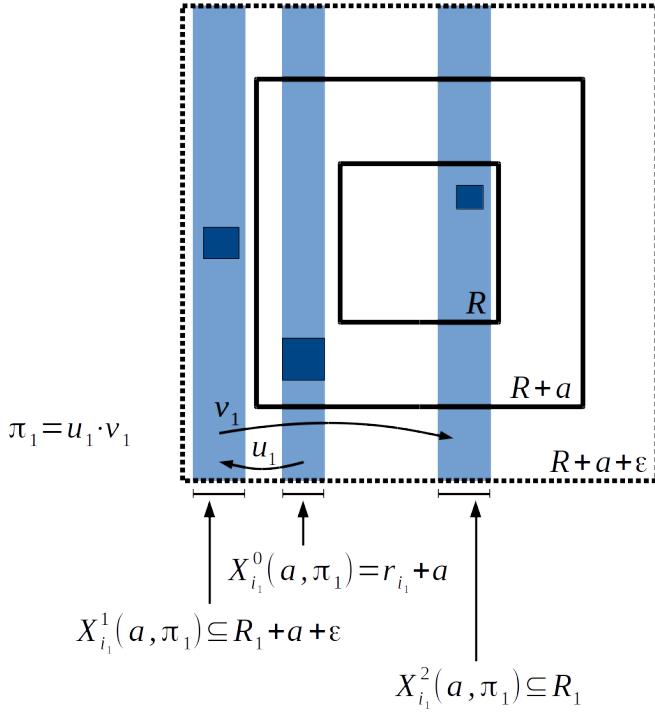


Figure 4.5: Illustration of $Prop_1(a, i_1, \pi_1)$ with $i_1 \in I_1$, $|\pi_1| = \ell_1 = 2$. The dark blue squares represent the centralized case, where both dimensions are controlled. The pale blue ribbons represent the distributed case, where we control only the first dimension, and over-approximate the behavior of the centralized case.

$H1(\ell_1)$: $\forall i_1 \in I_1 \exists \pi_1 \in \Pi_1^{\ell_1} Prop_1(0, i_1, \pi_1)$.

$H2(\ell_2)$: $\forall i_2 \in I_2 \exists \pi_2 \in \Pi_2^{\ell_2} Prop_1(0, i_1, \pi_2)$.

Then we define:

$$a(\ell_1) = \max\{a \in \{0, \frac{|R|}{100}, \frac{|R|}{10}\} \mid \forall i_1 \in I_1 \exists \pi_1 \in \Pi_1^{\ell_1} Prop_1(a, i_1, \pi_1)\}.$$

$$a(\ell_2) = \max\{a \in \{0, \frac{|R|}{100}, \frac{|R|}{10}\} \mid \forall i_2 \in I_2 \exists \pi_2 \in \Pi_2^{\ell_2} Prop_2(a, i_2, \pi_2)\}.$$

Let $A = \min\{a(\ell_1), a(\ell_2)\}$. From $H1(\ell_1)$ - $H2(\ell_2)$, it follows that, for all $i_1 \in I_1$ there exists a pattern of $\Pi_1^{\ell_1}$, denoted by π_{i_1} , such that $Prop_1(A, i_1, \pi_{i_1})$, and there exists a pattern of $\Pi_2^{\ell_2}$, denoted by π_{i_2} such that $Prop_2(A, i_2, \pi_{i_2})$.

Remark 6. Given a tiling $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$, $H1(\ell_1)$ means that the points of $R_1 + A$ can be (macro-step) controlled to R_1 using patterns which all have the same length ℓ_1 ; in other terms, all the macro-steps controlling $R_1 + A$ contain the same number ℓ_1 of elementary steps, and symmetrically for $H2(\ell_2)$.

Remark 7. The selection of an appropriate value for ϵ is for the moment performed by hand, and is the result of a compromise: if ϵ is too small, then $f_1(r_{i_1}, R_2, \pi_1(1)) \subseteq R_1 + \epsilon$ for no $\pi_1 \in \Pi_1^{\ell_1}$; if ϵ is too large, then $f_1(X_{i_1}^{\ell_1}, R_2 + \epsilon, \pi_1(\ell_1)) \subseteq R_1$ for no $\pi_1 \in \Pi_1^{\ell_1}$.

Using the same kinds of calculation as in the centralized case (see Section 4.2.3), one can see that finding ℓ_1, ℓ_2 such that $\Pi_{i_1}^{\ell_1} \neq \emptyset$ and $\Pi_{i_2}^{\ell_2} \neq \emptyset$, generating A and $\{\pi_{i_1}\}_{i_1 \in I_1}$, and $\{\pi_{i_2}\}_{i_2 \in I_2}$, can be performed in time $O((\max(N_1, N_2))^K \cdot 2^{\max(n_1, n_2)D})$. Hence the running time of the control test procedure is also in $O((\max(N_1, N_2))^K \cdot 2^{\max(n_1, n_2)D})$.

Lemma 1. *Consider a tiling $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ of the form $\{r_{i_1} \times r_{i_2}\}_{(i_1, i_2) \in I_1 \times I_2}$. Suppose that $H1(\ell_1)$ and $H2(\ell_2)$ hold for some $\ell_1, \ell_2 \leq K$. Then we have:*

— in case $\ell_1 \leq \ell_2$: for all $1 \leq k \leq \ell_1$ and all $i_1 \in I_1$,

$$\begin{aligned} f((r_{i_1} + A) \times (R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_1 &\subseteq X_{i_1}^k(A, \pi_{i_1}) \subseteq R_1 + A + \varepsilon \\ f((R_1 + A) \times (r_{i_2} + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_2 &\subseteq X_{i_2}^k(A, \pi_{i_2}) \subseteq R_2 + A + \varepsilon \\ f((r_{i_1} + A) \times (R_2 + A), (\pi_{i_1}^{\ell_1}, \pi_{i_2}^{\ell_1}))|_1 &\subseteq X_{i_1}^{\ell_1}(A, \pi_{i_1}) \subseteq R_1 \end{aligned}$$

— in case $\ell_2 \leq \ell_1$: for all $1 \leq k \leq \ell_2$ and all $i_2 \in I_2$,

$$\begin{aligned} f((r_{i_1} + A) \times (R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_1 &\subseteq X_{i_1}^k(A, \pi_{i_1}) \subseteq R_1 + A + \varepsilon \\ f((R_1 + A) \times (r_{i_2} + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_2 &\subseteq X_{i_2}^k(A, \pi_{i_2}) \subseteq R_2 + A + \varepsilon \\ f((R_1 + A) \times (r_{i_2} + A), (\pi_{i_1}^{\ell_2}, \pi_{i_2}^{\ell_2}))|_2 &\subseteq X_{i_2}^{\ell_2}(A, \pi_{i_2}) \subseteq R_2. \end{aligned}$$

The proof is given in Appendix B.

At $t = 0$, consider a point $x(0) = (x_1(0), x_2(0))$ of $R + (A, A)$, and let us apply concurrently the strategy induced by \mathcal{R}_1 on x_1 , and \mathcal{R}_2 on x_2 . After ℓ_1 steps, by Lemma 1, we obtain a point $x(\ell_1) = (x_1(\ell_1), x_2(\ell_1)) \in R_1 \times (R_2 + A + \varepsilon)$. Then, after ℓ_1 steps, we obtain again a point $x(2\ell_1) \in R_1 \times (R_2 + A + \varepsilon)$, and so on iteratively. Likewise, we obtain points $x(\ell_2), x(2\ell_2), \dots$ which all belong to $(R_1 + A + \varepsilon) \times R_2$. It follows that, after $\ell = \text{lcm}(\ell_1, \ell_2)$ steps, we obtain a point $x(\ell)$ which belongs to $R_1 \times R_2 = R$, where $\text{lcm}(\ell_1, \ell_2)$ denotes the least common multiple of ℓ_1 and ℓ_2 .

Theorem 5. *Suppose that there is a tiling $\mathcal{R}_1 = \{r_{i_1}\}_{i_1 \in I_1}$ of R_1 , a tiling $\mathcal{R}_2 = \{r_{i_2}\}_{i_2 \in I_2}$ of R_2 , a positive real ε , and two positive integers $\ell_1, \ell_2 \leq K$ such that $H1(\ell_1)$ and $H2(\ell_2)$ hold. Let $\ell = \text{lcm}(\ell_1, \ell_2)$ with $\ell = \alpha_1 \ell_1 = \alpha_2 \ell_2$ for some $\alpha_1, \alpha_2 \in \mathbb{N}$.*

Then \mathcal{R}_1 induces a sequence of α_1 macro-steps on $R_1 + A$, and \mathcal{R}_2 a sequence of α_2 macro-steps on $R_2 + A$, such that, applied concurrently, we have, for all $i_1 \in I_1$ and $i_2 \in I_2$:

$$f((r_{i_1} + A) \times (R_2 + A), \pi)|_1 \subseteq R_1 \quad \wedge \quad f((R_1 + A) \times (r_{i_2} + A), \pi)|_2 \subseteq R_2,$$

for some $\pi = (\pi_1, \pi_2) \in \Pi^\ell$ where π_1 (resp. π_2) is of the form $\pi_1^1 \cdots \pi_1^{\alpha_1}$ (resp. $\pi_2^1 \cdots \pi_2^{\alpha_2}$) with $\pi_1^i \in \Pi_1^{\ell_1}$ for all $1 \leq i \leq \alpha_1$ (resp. $\pi_2^i \in \Pi_2^{\ell_2}$ for all $1 \leq i \leq \alpha_2$).

Hence:

$$f(r_{i_1, i_2} + (A, A), \pi) \subseteq R.$$

Besides, for all prefix π' of π , we have:

$$f((r_{i_1} + A) \times (R_2 + A), \pi')|_1 \subseteq R_1 + A + \varepsilon \quad \wedge \quad f((R_1 + A) \times (r_{i_2} + A), \pi')|_2 \subseteq R_2 + A + \varepsilon.$$

Hence:

$$f(r_{i_1, i_2} + (A, A), \pi') \subseteq R + (A + \varepsilon, A + \varepsilon).$$

If $H1(\ell_1)$ - $H2(\ell_2)$ hold, there exists a control that steers $R + (A, A)$ to R in ℓ steps. Letting $R^{(1)} = R + (A, A)$, it is then possible to iterate the process on $R^{(1)}$ and, in case of success, to generate a rectangle $R^{(2)} = R^{(1)} + (A^{(1)}, A^{(1)})$ from which $R^{(1)}$ would be reachable in ℓ' steps, for some $A^{(1)} \geq 0$ and $\ell' \in \mathbb{N}$. And so on, iteratively, one generates an increasing sequence of nested control rectangles, as in Section 4.2.3, until a step i for which $A^{(i)} = 0$.

Theorem 5 allows us to implement the method as far as we are able to compute the results of applying mappings f_1 and f_2 to symbolic states represented by rectangles. When f_1 and f_2 are affine, the results can be easily computed using the data structure of “zonotopes” [73]. The method has been implemented in the case of affine mappings, using the system MINIMATOR [67, 106].

Example 3. Consider again the specification of a two-room apartment given in Example 1 and Appendix A.2. We consider the distributed control synthesis problem where the first (resp. second) state component corresponds to the temperature of the first (resp. second) room T_1 (resp. T_2), and the first (resp. second) control mode component corresponds to the heater u_1 (resp. u_2) of the first (resp. second) room.

Set $R = R_1 \times R_2 = [18.5, 22] \times [18.5, 22]$. Let $D = 3$ (the depth of bisection is at most 3), and $K = 10$ (the maximum length of patterns is 10). The parameter ε is set to value 1.5°C . We look for a distributed controller which steers any temperature state in $S = S_1 \times S_2 = [18.5 - a, 22] \times [18.5 - a, 22]$ to R with a as large as possible, then maintain it in R indefinitely.

Using our implementation, the computation of the control synthesis takes 220s of CPU time. The method iterates 8 times the macro-step control synthesis procedure. We find $S = [18.5 - a, 22] \times [18.5 - a, 22]$ with $a = 6.5$, i.e. $S = [12, 22] \times [12, 22]$. This means that any element of S can be driven to R within 8 macro-steps of length (at most) 10, i.e., within $8 \times 10 = 80$ units of time. Since each unit of time is of duration $\tau = 5\text{s}$, any trajectory starting from S reaches R within $80 \times 5 = 400\text{s}$. The trajectory is then guaranteed to always stay (at each discrete time t) in $R + (\varepsilon, \varepsilon) = [17, 23.5] \times [17, 23.5]$.

These results are consistent with the simulation given in Figure 4.6 showing the time evolution of (T_1, T_2) starting from $(12, 12)$. Simulations of the control are also given in the state space plane, in Figure 4.6, for initial states $(T_1, T_2) = (12, 12)$, $(T_1, T_2) = (12, 19)$ and $(T_1, T_2) = (22, 12)$.

Not surprisingly, the performance guaranteed by the distributed approach ($a = 6.5$, reachability of R in 400s) are worse than those guaranteed by the centralized approach of Example 2 ($a = 53.5$, reachability of R in 300s). However, unexpectedly, the CPU computation time in the distributed approach (220s) is here worse than the

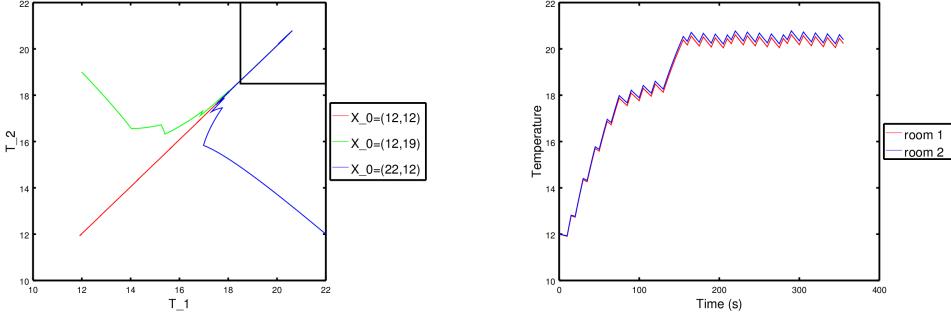


Figure 4.6: Simulations of the distributed reachability controller for three different initial conditions plotted in the state space plane (left); simulation of the distributed reachability controller for the initial condition $(12, 12)$ plotted within time (right).

CPU time of the centralized approach (4.14s). This relative inefficiency is due to the small size of the example.

4.2.5 Case study

This case study, proposed by the Danish company Seluxit, aims at controlling the temperature of an eleven rooms house, heated by geothermal energy. The *continuous* dynamics of the system is the following:

$$\frac{d}{dt}T_i(t) = \sum_{j=1}^n A_{i,j}^d(T_j(t) - T_i(t)) + B_i(T_{env}(t) - T_i(t)) + H_{i,j}^v \cdot v_j \quad (4.8)$$

The temperatures of the rooms are the T_i . The matrix A^d contains the heat transfer coefficients between the rooms, matrix B contains the heat transfer coefficients between the rooms and the external temperature, set to $T_{env} = 10^\circ C$ for the computations. The control matrix H^v contains the effects of the control on the room temperatures, and the control variable is here denoted by v_j . We have $v_j = 1$ (resp. $v_j = 0$) if the heater in room j is turned on (resp. turned off). We thus have $n = 11$ and $N = 2^{11} = 2048$ switching modes. The dynamics of the system is recalled in Appendix A.9.

Note that the matrix A^d is parametrized by the open or closed state of the doors in the house. In our case, the average between closed and open matrices was taken for the computations. The exact values of the coefficients are given in [112]. The controller has to select which heater to turn on in the eleven rooms. Due to a limitation of the capacity supplied by the geothermal device, the 11 heaters cannot be turned on at the same time. In our case, we limit to 4 the number of heaters that can be on at the same time.

We consider the distributed control synthesis problem where the first (resp. second) state component corresponds to the temperatures of rooms 1 to 5 (resp. 6 to 11), and the first (resp. second) control mode component corresponds to the heaters of rooms 1 to 5 (resp. 6 to 11). Hence $n_1 = 5, n_2 = 6, N_1 = 2^5, N_2 = 2^6$.

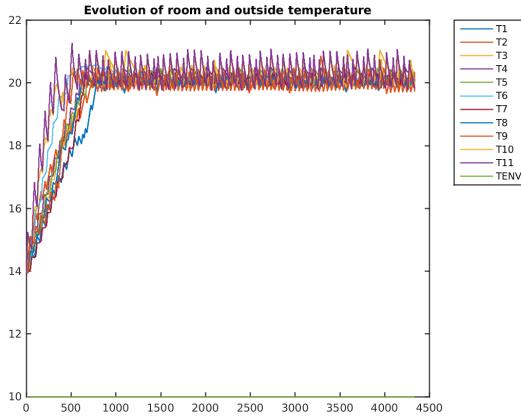


Figure 4.7: Simulation of the Seluxit case study plotted with time (in min) for $T_{env} = 10^\circ C$.

We impose that at most two heaters are switched on at the same time in the first sub-system, and at most two in the second sub-system.

Let $D = 1$ (the bisection depth is at most 1), and $K = 4$ (the maximum length of patterns is 4). The parameter ε is set to value $0.5^\circ C$. The sampling time is $\tau = 15$ minutes.

We look for a distributed controller which steers any temperature state in the rectangle $S = [18 - a, 22]^{11}$ to $R = [18, 22]^{11}$ with a as large as possible, then maintain the temperatures in R indefinitely. Using our implementation, the computation of the control synthesis takes around 20 hours of CPU time. The method iterates the macro-step control synthesis procedure 15 times. We find $S = [18 - a, 22]^{11}$ with $a = 4.2$, i.e. $S = [13.8, 22]^{11}$. This means that any element of S can be driven into R within 15 macro-steps of length (at most) 4, i.e., within $15 \times 4 = 60$ units of time. Since each time unit is of duration $\tau = 15$ min, any trajectory starting from S reaches R within $60 \times 15 = 900$ min. The trajectory is then guaranteed to stay in $R + (\varepsilon, \varepsilon) = [17.5, 22.5]^{11}$. These results are consistent with the simulation given in Figure 4.7 showing the time evolution of the temperature of the rooms, starting from 14¹¹.

Robustness Experiments

We now perform the same simulations as in Figure 4.7, except that the environment temperature is not fixed at $10^\circ C$ but follows scenarios of soft winter (Figure 4.8) and spring (Figure 4.9). The environment temperature is plotted in green in the figures. The spring scenario is taken from [112], and the soft winter scenario is the winter scenario of [112] with 5 additional degrees. We see that our controller, which is designed for $T_{env} = 10^\circ C$ still satisfies the properties of reachability and stability. These simulations are very close to those obtained in [112].

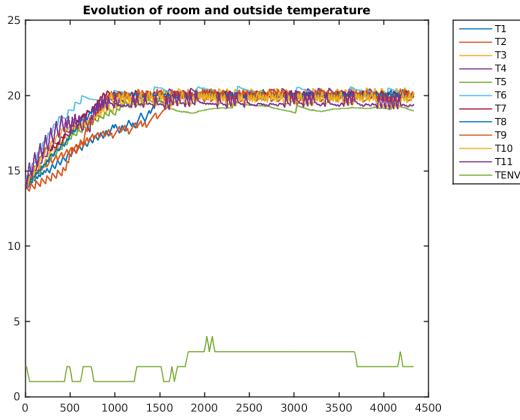


Figure 4.8: Simulation of the Seluxit case study in the soft winter scenario.

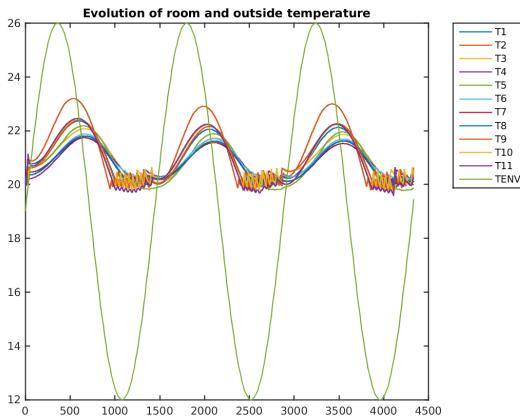


Figure 4.9: Simulation of the Seluxit case study in the spring scenario.

4.2.6 Continuous-time case

In this section, we consider the case of continuous-time differential equations. The time t now takes its values in $\mathbb{R}_{\geq 0}$.

4.2.7 Reachability in continuous time

Consider the continuous-time system with *finite control*:

$$\dot{x}_1(t) = f_1(x_1(t), x_2(t), u_1) \quad (4.9)$$

$$\dot{x}_2(t) = f_2(x_1(t), x_2(t), u_2) \quad (4.10)$$

where x_1 (resp. x_2) is the first (resp. second) component of the state vector variable, taking its values in \mathbb{R}^{n_1} (resp. \mathbb{R}^{n_2}), and where u_1 (resp. u_2) is the first (resp. second) component of the control *mode*, taking its values in the *finite set* U_1 (resp. U_2). We will often write x for (x_1, x_2) , u for (u_1, u_2) , and n for $n_1 + n_2$. We will also abbreviate the set $U_1 \times U_2$ as U . We abbreviate the continuous-time system under the form:

$$\dot{x}(t) = f(x(t), u) \quad (4.11)$$

where x is a vector state variable taking its values in $\mathbb{R}^n = \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, and where u is of the form (u_1, u_2) , with u_1 taking its values in U_1 and u_2 in U_2 . We assume that, given an initial value x_0 , Equation (4.11) has a solution (e.g., assuming that the vector field f (resp. f_1, f_2) is Lipschitz).

We define the reachable set of (4.11) from a set of initial states X_0 , at time t ($0 \leq t \leq \tau$) under control mode u :

$$Reach_f(t, X_0, u) = \{\Phi(t, x_0, u) \mid x_0 \in X_0\}.$$

where $\Phi(t, x, u)$ denotes the state $x(t)$ reached at time t ($0 \leq t \leq \tau$) starting from the initial state x , under control mode $u \in U$.

We define the reachable set of (4.9) from a set of initial states $X_1 \subset \mathbb{R}^{n_1}$, at time t ($0 \leq t \leq \tau$) under control mode $u_1 \in U_1$ and perturbation $X_2 \subset \mathbb{R}^{n_2}$:

$$Reach_{f_1}(t, X_1, X_2, u_1) = \{\Phi_1(t, x_1, X_2, u_1) \mid x_1 \in X_1\}.$$

where $\Phi_1(t, x_1, X_2, u_1)$ is the set of states $x_1(t)$ reached at time t ($t \geq 0$) from the initial state x_1 , under control mode u_1 and perturbation X_2 .

Symmetrically, we define the reachable set of (4.10) from a set of initial states $X_2 \subset \mathbb{R}^{n_2}$, at time t ($0 \leq t \leq \tau$) under control mode $u_2 \in U_2$ and perturbation $X_1 \subset \mathbb{R}^{n_1}$:

$$Reach_{f_2}(t, X_1, X_2, u_2) = \{\Phi_2(t, X_1, x_2, u_2) \mid x_2 \in X_2\}.$$

where $\Phi_2(t, X_1, x_2, u_2)$ is the set of states $x_2(t)$ reached at time $t \geq 0$ from the initial state x_2 , under control mode u_2 and perturbation X_1 .

All the notions of reachable sets for modes are extended in the natural manner to the notions of reachable sets for *patterns*. For example, for the pattern $\pi = u \cdot v$ of length 2, and for $0 \leq t \leq \tau$, we define:

$$\begin{aligned} Reach_f(t, X_0, \pi) &= Reach_f(t, X_0, u) \\ Reach_f(\tau + t, X_0, \pi) &= Reach_f(t, X_1, v) \quad \text{with } X_1 = Reach_f(\tau, X_0, u). \end{aligned}$$

Distributed control

Recall that π_1^k (resp. π_2^k) denotes the prefix of length k of π_1 (resp. π_2), and $\pi_1(k)$ (resp. $\pi_2(k)$) the k -th element of sequence π_1 (resp. π_2). We now give the counterpart of Definition 8.

Definition 9. Consider an element r_{i_1} (resp. r_{i_2}) of a tiling \mathcal{R}_1 (resp. \mathcal{R}_2) of R_1 (resp. R_2), and a sequence $\pi_1 \in \Pi_1^{\leq K}$ (resp. $\pi_2 \in \Pi_2^{\leq K}$) of length ℓ_1 (resp. ℓ_2). The approximate first-component sequence $\{Y_{i_1}^k(a, \pi_1)\}_{0 \leq k \leq \ell_1}$ is defined as follows:

- $Y_{i_1}^0(a, \pi_1) = r_{i_1} + a$ and
- $Y_{i_1}^k(a, \pi_1) = \bigcup_{0 \leq t \leq \tau} Reach_{f_1}(t, Y_{i_1}^{k-1}(a, \pi_1), R_2 + a + \varepsilon, \pi_1(k))$ for $1 \leq k \leq \ell_1$.

Similarly, the approximate second-component sequence $\{Y_{i_2}^k(a, \pi_2)\}_{0 \leq k \leq \ell_2}$ is defined by

- $Y_{i_2}^0(a, \pi_2) = r_{i_2} + a$ and
- $Y_{i_2}^k(a, \pi_2) = \bigcup_{0 \leq t \leq \tau} \text{Reach}_{f_2}(t, R_1 + a + \varepsilon, Y_{i_2}^{k-1}(a, \pi_2), \pi_2(k))$ for $1 \leq k \leq \ell_2$.

We define the property $\text{Prop}_1(a, i_1, \pi_1)$ by:

$$\begin{aligned} Y_{i_1}^k(a, \pi_1) &\subseteq R_1 + a + \varepsilon \text{ for } 1 \leq k \leq \ell_1 \\ \text{and } \text{Reach}_{f_1}(\ell_1\tau, r_{i_1} + a, R_2 + a + \varepsilon, \pi_1) &\subseteq R_1. \end{aligned}$$

Likewise, we define the property $\text{Prop}_2(a, i_2, \pi_2)$ by:

$$\begin{aligned} Y_{i_2}^k(a, \pi_2) &\subseteq R_2 + a + \varepsilon \text{ for } 1 \leq k \leq \ell_2 \\ \text{and } \text{Reach}_{f_2}(\ell_2\tau, R_1 + a + \varepsilon, r_{i_2} + a, \pi_2) &\subseteq R_2. \end{aligned}$$

Assumptions $H1(\ell_1)$, $H2(\ell_2)$ and expressions A , π_{i_1} , π_{i_2} are defined exactly as in Section 4.2.4. We now give the counterpart of Lemma 1 (the proof is similar).

Lemma 2. *Consider a tiling $\mathcal{R} = \mathcal{R}_1 \times \mathcal{R}_2$ of the form $\{r_{i_1} \times r_{i_2}\}_{(i_1, i_2) \in I_1 \times I_2}$. Suppose that $H1(\ell_1)$ and $H2(\ell_2)$ hold, for some positive real ε , and some positive integers ℓ_1, ℓ_2 . Then we have*

- in case $\ell_1 \leq \ell_2$, for all $t \in [(k-1)\tau, k\tau]$ ($1 \leq k \leq \ell_1$):

$$\begin{aligned} \text{Reach}_f(t, (r_{i_1} + A) \times (R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_1 &\subseteq Y_{i_1}^k(a, \pi_{i_1}) \subseteq R_1 + A + \varepsilon \\ \text{Reach}_f(t, (R_1 + A) \times (r_{i_2} + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_2 &\subseteq Y_{i_2}^k(a, \pi_{i_2}) \subseteq R_2 + A + \varepsilon \\ \text{Reach}_f(\ell_1\tau, (r_{i_1} + A) \times (R_2 + A), (\pi_{i_1}^{\ell_1}, \pi_{i_2}^{\ell_1}))|_1 &\subseteq R_1. \end{aligned}$$

- in case $\ell_2 \leq \ell_1$, for all $t \in [(k-1)\tau, k\tau]$ ($1 \leq k \leq \ell_2$):

$$\begin{aligned} \text{Reach}_f(t, (r_{i_1} + A) \times (R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_1 &\subseteq Y_{i_1}^k(a, \pi_{i_1}) \subseteq R_1 + A + \varepsilon \\ \text{Reach}_f(t, (R_1 + A) \times (r_{i_2} + A), (\pi_{i_1}^k, \pi_{i_2}^k))|_2 &\subseteq Y_{i_2}^k(a, \pi_{i_2}) \subseteq R_2 + A + \varepsilon \\ \text{Reach}_f(\ell_2\tau, (R_1 + A) \times (r_{i_2} + A), (\pi_{i_1}^{\ell_2}, \pi_{i_2}^{\ell_2}))|_2 &\subseteq R_2. \end{aligned}$$

We now give the counterpart of Theorem 5 (the proof is similar).

Theorem 6. *Suppose that there is a tiling $\mathcal{R}_1 = \{r_{i_1}\}_{i_1 \in I_1}$ of R_1 and a tiling $\mathcal{R}_2 = \{r_{i_2}\}_{i_2 \in I_2}$ of R_2 , such that $H1(\ell_1)$ and $H2(\ell_2)$ hold for some $\ell_1, \ell_2 \leq K$. Let $\ell = \text{lcm}(\ell_1, \ell_2)$ with $\ell = \alpha_1\ell_1 = \alpha_2\ell_2$ for some $\alpha_1, \alpha_2 \in \mathbb{N}$.*

Then \mathcal{R}_1 induces a sequence of α_1 macro-steps on $R_1 + A$, and \mathcal{R}_2 a sequence of α_2 macro-steps on $R_2 + A$, such that, when applied concurrently, we have for all $i_1 \in I_1$ and $i_2 \in I_2$:

$$\begin{aligned} \text{Reach}_f(\ell\tau, (r_{i_1} + A) \times (R_2 + A), \pi)|_1 &\subseteq R_1 \wedge \\ \text{Reach}_f(\ell\tau, (R_1 + A) \times (r_{i_2} + A), \pi)|_2 &\subseteq R_2, \end{aligned}$$

for some $\pi = (\pi_1, \pi_2) \in \Pi^\ell$ where π_1 (resp. π_2) is of the form $\pi_1^1 \cdots \pi_1^{\alpha_1}$ (resp. $\pi_2^1 \cdots \pi_2^{\alpha_2}$) with $\pi_1^i \in \Pi_1^{\ell_1}$ for all $1 \leq i \leq \alpha_1$ (resp. $\pi_2^i \in \Pi_2^{\ell_2}$ for all $1 \leq i \leq \alpha_2$).

Hence:

$$\text{Reach}_f(\ell\tau, r_{i_1, i_2} + (A, A), \pi) \subseteq R.$$

Besides, for all $0 \leq t \leq \ell\tau$, we have:

$$\begin{aligned} Reach_f(t, (r_{i_1} + A) \times (R_2 + A), \pi)_{|1} &\subseteq R_1 + A + \varepsilon \\ \wedge \quad Reach_f(t, (R_1 + A) \times (r_{i_2} + A), \pi)_{|2} &\subseteq R_2 + A + \varepsilon. \end{aligned}$$

Hence, for all $0 \leq t \leq \ell\tau$:

$$Reach_f(t, r_{i_1, i_2} + (A, A), \pi) \subseteq R + (A + \varepsilon, A + \varepsilon).$$

Theorem 6 allows us to implement the method along the same lines as in the discrete-time case, except that we apply the operator $Reach_{f_1}$ and $Reach_{f_2}$ on continuous time intervals of the form $[k, (k + 1)\tau]$ instead of the mappings f_1 and f_2 at times $k\tau$. We have implemented the method using the system *DynIBEX* [5, 56] which makes use of interval arithmetic [141] and Runge-Kutta methods to compute (an overapproximation of) the application results of $Reach_{f_1}$ and $Reach_{f_2}$.

Application

We demonstrate the feasibility of our approach on the 4-room building ventilation application adapted from [134], and recalled in Appendix A.4. The centralized controller was obtained with 704 tiles in 29 minutes, the distributed controller was obtained with $16 + 16$ tiles in 20 seconds. In both cases, patterns of length 1 are used. The perturbation due to human beings has been taken into account by setting the parameters δ_{s_i} equal to the whole interval $[0, 1]$ for the decomposition, and the imposed perturbation for the simulation is given Figure 4.10. The temperatures T_o and T_c have been set to the interval $[27, 30]$ for the decomposition, and are set to $30^\circ C$ for the simulation. A simulation of the controller obtained with the state-space bisection procedure is given in Figure 4.11, where the control objective is to stabilize the temperature in $[20, 22]^2 \times [22, 24]^2$ while never going out of $[19, 23]^4 \times [21, 25]^4$.

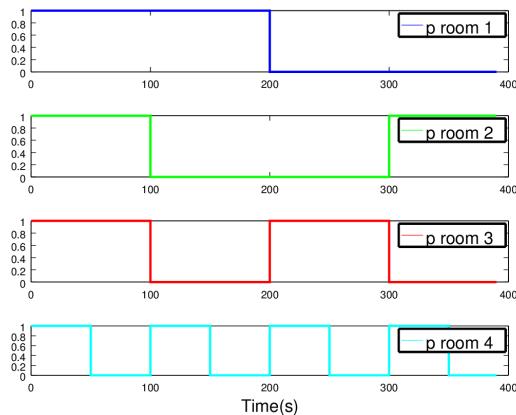


Figure 4.10: Perturbation (presence of humans) imposed within time in the different rooms.

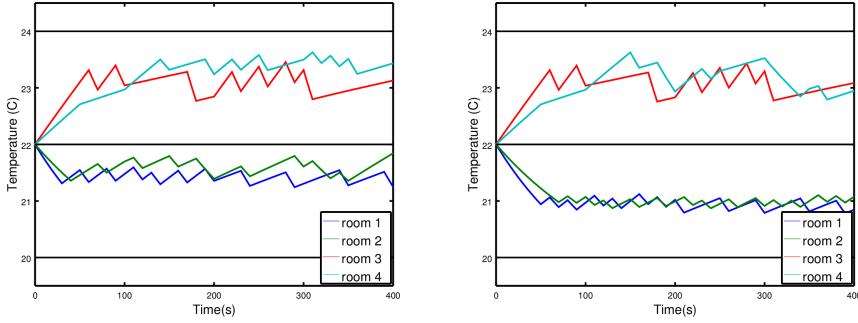


Figure 4.11: Simulation of the centralized (left) and distributed (right) controllers from the initial condition (22, 22, 22, 22).

4.2.8 Final remarks

In this chapter, we have proposed a distributed approach for control synthesis of sampled switching systems in the discrete-time framework and applied it to a real floor heating system. To our knowledge, this is the first time that reachability and stability properties are guaranteed for a case study of this size. We have also explained how the method extends to the continuous-time framework. The method can be extended to take into account obstacles and safety constraints.

Note that it is essential in our method that the components are *sampled* with the *same* sampling period τ , and that their clocks are synchronized. It would be interesting to investigate how the approach behaves when clocks are badly synchronized or when they have different periods (see, e.g., [99]).

4.3 Perturbed and distributed Euler scheme

We consider the perturbed control system

$$\dot{x} = f_j(x, d), \quad (4.12)$$

where d is assumed to belong to a given set D . In the following, we denote by d^m the center (centroid or center of gravity) of set D . In practice, the set D is given as a box, a we thus take d^m the center of the box.

In the same manner as the previous chapter, we introduce some additional hypotheses allowing us to use an Euler's scheme with precise error bounds. We suppose that the system is Lipschitz in the following sense:

For all $j \in U$, there exists a constant $L_j > 0$ such that:

$$\|f_j(x, d) - f_j(y, e)\| \leq L_j \left\| \begin{pmatrix} x \\ d \end{pmatrix} - \begin{pmatrix} y \\ e \end{pmatrix} \right\|, \quad \forall x, y \in S, \forall d, e \in D$$

We then introduce the constant:

$$C_j = \sup_{x \in S} L_j \|f_j(x, d^m)\|$$

where d^m denotes the center of box D .

We now introduce a hypothesis similar to (H1) made in Chapter 4 (2), with additional disturbance.

($H_{U,D}$) For every mode $j \in U$, there exists constants $\lambda_j \in \mathbb{R}$ and $\gamma_j \in \mathbb{R}_{>0}$ such that $\forall x, x' \in S$ and $\forall y, y' \in D$, the following expression holds

$$\langle f_j(x, y) - f_j(x', y'), x - x' \rangle \leq \lambda_j \|x - x'\|^2 + \gamma_j \|x - x'\| \|y - y'\|.$$

While the OSL condition is related to incremental stability, hypothesis ($H_{U,D}$) seems related to the notion of incremental input-to-state stability [13, 14, 138] (sometimes denoted δ -ISS in the literature). Indeed, an incrementally input-to-state system verifies a relation close to ($H_{U,D}$), with a positive constant λ_j (or more generally a κ function). Here, we thus generalize this notion with negative constants λ_j , making the hypothesis much weaker. Because the system lies in a compact set (provided that a controller is found), constants λ_j and γ_j can always be found.

Computation of constants λ_j and γ_j , L_j and C_j The computation of constants L_j , C_j , λ_j ($j \in U$) are realized with a constrained optimization algorithm. They are performed using the “sqp” function of Octave, applied on the following optimization problems:

— Constant L_j is computed exactly as in the unperturbed case:

$$L_j = \max_{(x,d),(y,e) \in S \times D, (x,d) \neq (y,e)} \frac{\|f_j(x,d) - f_j(y,e)\|}{\left\| \begin{pmatrix} x \\ d \end{pmatrix} - \begin{pmatrix} y \\ e \end{pmatrix} \right\|}$$

— Constant C_j is computed with the following optimization problem:

$$C_j = \max_{x \in S} L_j \|f_j(x, d^m)\|$$

Knowing that:

$$\begin{aligned} \langle f_j(x, y) - f_j(x', y'), x - x' \rangle &= \\ \langle f_j(x, y) - f_j(x', y), x - x' \rangle + \langle f_j(x', y) - f_j(x', y'), x - x' \rangle & \end{aligned}$$

— Constant λ_j is first computed as follows:

$$\lambda_j = \max_{x, x' \in T, y \in D, x \neq x'} \frac{\langle f_j(x, y) - f_j(x', y), x - x' \rangle}{\|x - x'\|^2}$$

— Constant γ_j is then computed:

$$\gamma_j = \max_{x, x' \in T, y, y' \in D, x \neq x', y \neq y'} \frac{\langle f_j(x, y) - f_j(x', y'), x - x' \rangle - \lambda_j \|x - x'\|^2}{\|x - x'\| \|y - y'\|}$$

Perturbed Euler's scheme We now define a perturbed Euler's scheme as follows:

$$\tilde{x}(\tau) = \tilde{x}(0) + \tau f_j(\tilde{x}(0), d^m) \quad (4.13)$$

We define the approximate trajectory computed with the distributed Euler's scheme by $\tilde{\phi}_j(t; \tilde{x}^0) = \tilde{x}^0 + t f_j(\tilde{x}^0, d^m)$ for $t \in [0, \tau]$, when the system is in mode j and with an initial condition \tilde{x}^0 .

We now give a perturbed version of Theorem 3.

Theorem 7. *Given a distributed sampled switched system, suppose that the system satisfies $(H_{U,D})$, and consider a point \tilde{x}^0 and a positive real δ . We have, for all $x^0 \in B(\tilde{x}^0, \delta)$, $w : \mathbb{R}^+ \rightarrow D$, $t \in [0, \tau]$, $j \in U$:*

$$\phi_j(t; x^0, w) \in B(\tilde{\phi}_j(t; \tilde{x}^0), \delta_j(t)).$$

with, denoting by $|D|$ the diameter of D :

— if $\lambda_j < 0$,

$$\begin{aligned} \delta_j(t) = & \left(\frac{(C_j)^2}{-(\lambda_j)^4} \left(-(\lambda_j)^2 t^2 - 2\lambda_j t + 2e^{\lambda_j t} - 2 \right) \right. \\ & + \frac{1}{(\lambda_j)^2} \left(\frac{C_j \gamma_j |D|}{-\lambda_j} (-\lambda_j t + e^{\lambda_j t} - 1) \right. \\ & \left. \left. + \lambda_j \left(\frac{(\gamma_j)^2 (|D|/2)^2}{-\lambda_j} (e^{\lambda_j t} - 1) + \lambda_j \delta^2 e^{\lambda_j t} \right) \right) \right)^{1/2} \end{aligned} \quad (4.14)$$

— if $\lambda_j > 0$,

$$\begin{aligned} \delta_j(t) = & \frac{1}{(3\lambda_j)^{3/2}} \left(\frac{C^2}{\lambda_j} \left(-9(\lambda_j)^2 t^2 - 6\lambda_j t + 2e^{3\lambda_j t} - 2 \right) \right. \\ & + 3\lambda_j \left(\frac{C \gamma_j |D|}{\lambda_j} (-3\lambda_j t + e^{3\lambda_j t} - 1) \right. \\ & \left. \left. + 3\lambda_j \left(\frac{(\gamma_j)^2 (|D|/2)^2}{\lambda_j} (e^{3\lambda_j t} - 1) + 3\lambda_j \delta^2 e^{3\lambda_j t} \right) \right) \right)^{1/2} \end{aligned} \quad (4.15)$$

— if $\lambda_j = 0$,

$$\begin{aligned} \delta_j(t) = & ((C_j)^2 (-t^2 - 2t + 2e^t - 2) \\ & + (C_j \gamma_j |D| (-t + e^t - 1) \\ & + ((\gamma_j)^2 (|D|/2)^2 (e^t - 1) + \delta^2 e^t)))^{1/2} \end{aligned} \quad (4.16)$$

A similar result can be established for sub-system 2, permitting to perform a distributed control synthesis.

Proof. We have, for all $x, \tilde{x} \in S^2$:

$$\begin{aligned}
\frac{1}{2} \frac{d(\|x - \tilde{x}\|^2)}{dt} &= \langle f_j(x, w) - f_j(\tilde{x}(0), d^m), x - \tilde{x} \rangle \\
&= \langle f_j(x, w) - f_j(\tilde{x}, d^m) + f_j(\tilde{x}, d^m) - f_j(\tilde{x}(0), d^m), x - \tilde{x} \rangle \\
&\leq \langle f_j(x, w) - f_j(\tilde{x}, d^m), x - \tilde{x} \rangle + \langle f_j(\tilde{x}, d^m) - f_j(\tilde{x}(0), d^m), x - \tilde{x} \rangle \\
&\leq \langle f_j(x, w) - f_j(\tilde{x}, d^m), x - \tilde{x} \rangle + \|f_j(\tilde{x}, d^m) - f_j(\tilde{x}(0), d^m)\| \|x - \tilde{x}\| \\
&\leq \langle f_j(x, w) - f_j(\tilde{x}, d^m), x - \tilde{x} \rangle + L \left\| \begin{pmatrix} \tilde{x} \\ d^m \end{pmatrix} - \begin{pmatrix} \tilde{x}(0) \\ d^m \end{pmatrix} \right\| \|x - \tilde{x}\| \\
&\leq \lambda \|x - \tilde{x}\|^2 + \gamma \|w - d^m\| \|x - \tilde{x}\| + Lt \|f(\tilde{x}(0), d^m)\| \|x - \tilde{x}\| \\
&\leq \lambda_j \|x - \tilde{x}\|^2 + \left(\gamma_j \frac{|D|}{2} + C_j t \right) \|x - \tilde{x}\|
\end{aligned}$$

where $|D|$ denotes the diameter of D . Using the fact that $\|x - \tilde{x}\| \leq \frac{1}{2}(\alpha \|x - \tilde{x}\|^2 + \frac{1}{\alpha})$ for any $\alpha > 0$, we can write three formulas following the sign of λ_j .

— if $\lambda_j < 0$, we can choose $\alpha = \frac{-\lambda_j}{C_j t + \gamma_j |D|/2}$, and we get the differential inequality:

$$\frac{d(\|x - \tilde{x}\|^2)}{dt} \leq \lambda_j \|x - \tilde{x}\|^2 + \frac{C_j^2}{-\lambda_j} t^2 + \frac{C_j \gamma_j |D|}{-\lambda_j} t + \frac{\gamma_j^2 (|D|/2)^2}{-\lambda_j}$$

— if $\lambda_j > 0$, we can choose $\alpha = \frac{\lambda_j}{C_j t + \gamma_j |D|/2}$, and we get the differential inequality:

$$\frac{d(\|x - \tilde{x}\|^2)}{dt} \leq 3\lambda_j \|x - \tilde{x}\|^2 + \frac{C_j^2}{\lambda_j} t^2 + \frac{C_j \gamma_j |D|}{\lambda_j} t + \frac{\gamma_j^2 (|D|/2)^2}{\lambda_j}$$

— if $\lambda_1 = 0$, we can choose $\alpha = \frac{1}{C_j t + \gamma_j |D|/2}$, and we get the differential inequality:

$$\frac{d(\|x - \tilde{x}\|^2)}{dt} \leq \|x - \tilde{x}\|^2 + C_j^2 t^2 + C_j \gamma_j |D| t + \gamma_j^2 (|D|/2)^2$$

In every case, the differential inequalities can be integrated to obtain the formulas of the theorem. □

Remark 8. One can note that for linear systems of the form

$$\dot{x} = A_j x + B_j w + C_j,$$

constants λ_j and γ_j can be replaced in the proof of Theorem 7 by the largest eigenvalue of $\frac{A_j + A_j^\top}{2}$ and $\|B_j\|$ respectively, and are thus not needed to be pre-computed with optimization algorithms.

We then establish a perturbed version of Corollary 2, using the same notations for the sequences δ_π^k .

Corollary 3. Given a switched system satisfying $(H_{U,D})$, consider a positive real δ and a set of points $\tilde{x}_1, \dots, \tilde{x}_m$ such that all the balls $B(\tilde{x}_i, \delta)$ for $1 \leq i \leq m$ cover R . Suppose that there exists patterns π_i of length k_i such that :

1. $B((\tilde{x}_i)_{\pi_i}^{k'}, \delta_{\pi_i}^{k'}) \subseteq S$, for all $k' = 1, \dots, k_i - 1$
2. $B((\tilde{x}_i)_{\pi_i}^{k_i}, \delta_{\pi_i}^{k_i}) \subseteq R$.
3. $\frac{d^2(\delta_j'(t))}{dt^2} > 0$ with $j = \pi_i(k')$ and $\delta' = \delta_{\pi_i}^{k'-1}$, for all $k' \in \{1, \dots, k_i\}$ and $t \in [0, \tau]$.

The above properties induce a control guaranteeing recurrence in R and safety in S , thus solving Problem 1. I.e., for any perturbation $w : \mathbb{R}^+ \rightarrow D$: if $x \in R$, then $\phi_\sigma(t; x, w) \in S$ for all $t \geq 0$, and any trajectory starting from R returns infinitely often in R .

The above corollary actually solves Problem 1 in presence of perturbations. Let us now explain how a system can be split in two sub-systems, and considering the state of the other sub-system as a disturbance allows us to build a compositional synthesis, drastically lowering the computational cost of the method.

4.3.1 Distributed synthesis

The goal is to split the system into two (or more) sub-systems and synthesize controllers for the sub-systems independently.

We consider the distributed control system

$$\dot{x}_1 = f_{\sigma_1}^1(x_1, x_2) \quad (4.17)$$

$$\dot{x}_2 = f_{\sigma_2}^2(x_1, x_2) \quad (4.18)$$

where $x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$, with $n_1 + n_2 = n$. Furthermore, $\sigma_1 \in U_1$ and $\sigma_2 \in U_2$ and $U = U_1 \times U_2$.

Note that the system (4.17-4.18) can be seen as the *interconnection* of sub-system (4.17) where x_2 plays the role of an “input” given by (4.18), with sub-system (4.18) where x_1 is an “input” given by (4.17).

Let $R = R_1 \times R_2$, $S = S_1 \times S_2$, $T = T_1 \times T_2$ and x_1^m (resp. x_2^m) be the center of R_1 (resp. R_2). We denote by $L_{\sigma_1}^1$ the Lipschitz constant for sub-system 1 under mode σ_1 :

$$\|f_{\sigma_1}^1(x_1, x_2) - f_{\sigma_1}^1(y_1, y_2)\| \leq L_{\sigma_1}^1 \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right\|$$

We then introduce the constant:

$$C_{\sigma_1}^1 = \sup_{x_1 \in S_1} L_{\sigma_1}^1 \|f_{\sigma_1}^1(x_1, x_2^m)\|$$

Similarly, we define the constants for sub-system 2:

$$\|f_{\sigma_2}^2(x_1, x_2) - f_{\sigma_2}^2(y_1, y_2)\| \leq L_{\sigma_2}^2 \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right\|$$

and

$$C_{\sigma_2}^2 = \sup_{x_2 \in S_2} L_{\sigma_2}^2 \|f_{\sigma_2}^2(x_1^m, x_2)\|$$

Let us now make additional assumptions on the coupled sub-systems, closely related to the notion of (incremental) input-to-state stability.

(H_{U₁,T₂}) For every mode $\sigma_1 \in U_1$, there exists constants $\lambda_{\sigma_1}^1 \in \mathbb{R}$ and $\gamma_{\sigma_1}^1 \in \mathbb{R}_{>0}$ such that $\forall x, x' \in S_1^2$ and $\forall y, y' \in T_2^2$, the following expression holds

$$\langle f_{\sigma_1}^1(x, y) - f_{\sigma_1}^1(x', y'), x - x' \rangle \leq \lambda_{\sigma_1}^1 \|x - x'\|^2 + \gamma_{\sigma_1}^1 \|x - x'\| \|y - y'\|.$$

(H_{U₂,T₁}) For every mode $\sigma_2 \in U_2$, there exists constants $\lambda_{\sigma_2}^2 \in \mathbb{R}$ and $\gamma_{\sigma_2}^2 \in \mathbb{R}_{>0}$ such that $\forall x, x' \in T_1^2$ and $\forall y, y' \in S_2^2$, the following expression holds

$$\langle f_{\sigma_2}^2(x, y) - f_{\sigma_2}^2(x', y'), y - y' \rangle \leq \lambda_{\sigma_2}^2 \|y - y'\|^2 + \gamma_{\sigma_2}^2 \|x - x'\| \|y - y'\|.$$

These assumptions express (a variant of) the fact that the function $V(x, x') = \|x - x'\|^2$ is an *ISS-Lyapunov function* (see, e.g., [13, 88]). Note that all the constants defined above can be numerically computed using constrained optimization algorithms.

Let us define the distributed Euler scheme:

$$\tilde{x}_1(\tau) = \tilde{x}_1(0) + \tau f_{\sigma_1}^1(\tilde{x}_1(0), x_2^m) \quad (4.19)$$

$$\tilde{x}_2(\tau) = \tilde{x}_2(0) + \tau f_{\sigma_2}^2(x_1^m, \tilde{x}_2(0)) \quad (4.20)$$

The exact trajectory is now denoted, for all $t \in [0, \tau]$, by $\phi_{(j_1, j_2)}(t; x^0)$ for an initial condition $x^0 = \begin{pmatrix} x_1^0 & x_2^0 \end{pmatrix}^T$, and when sub-system 1 is in mode $j_1 \in U_1$, and sub-system 2 is in mode $j_2 \in U_2$.

We define the approximate trajectory computed with the distributed Euler's scheme by $\tilde{\phi}_{j_1}^1(t; \tilde{x}_1^0) = \tilde{x}_1^0 + t f_{\sigma_1}^1(\tilde{x}_1^0, x_2^m)$ for $t \in [0, \tau]$, when sub-system 1 is in mode j_1 and with an initial condition \tilde{x}_1^0 . Similarly, for sub-system 2, $\tilde{\phi}_{j_2}^2(t; \tilde{x}_2^0) = \tilde{x}_2^0 + t f_{\sigma_2}^2(x_1^m, \tilde{x}_2^0)$ when sub-system 2 is in mode j_2 and with an initial condition \tilde{x}_2^0 .

We now give a distributed version of Theorem 3.

Theorem 8. *Given a distributed sampled switched system, suppose that sub-system 1 satisfies (H2), and consider a point \tilde{x}_1^0 and a positive real δ . We have, for all $x_1^0 \in B(\tilde{x}_1^0, \delta)$, $x_2^0 \in S_2$, $t \in [0, \tau]$, $j_1 \in U_1$ and any $\sigma_2 \in U_2$:*

$$\phi_{(j_1, \sigma_2)}(t; x^0)|_1 \in B(\tilde{\phi}_{j_1}^1(t; \tilde{x}_1^0), \delta_{j_1}(t)).$$

with $x^0 = \begin{pmatrix} x_1^0 & x_2^0 \end{pmatrix}^T$ and

— if $\lambda_{j_1}^1 < 0$,

$$\begin{aligned} \delta_{j_1}(t) = & \left(\frac{(C_{j_1}^1)^2}{-(\lambda_{j_1}^1)^4} \left(-(\lambda_{j_1}^1)^2 t^2 - 2\lambda_{j_1}^1 t + 2e^{\lambda_{j_1}^1 t} - 2 \right) \right. \\ & + \frac{1}{(\lambda_{j_1}^1)^2} \left(\frac{C_{j_1}^1 \gamma_{j_1}^1 |T_2|}{-\lambda_{j_1}^1} \left(-\lambda_{j_1}^1 t + e^{\lambda_{j_1}^1 t} - 1 \right) \right. \\ & \left. \left. + \lambda_{j_1}^1 \left(\frac{(\gamma_{j_1}^1)^2 (|T_2|/2)^2}{-\lambda_{j_1}^1} (e^{\lambda_{j_1}^1 t} - 1) + \lambda_{j_1}^1 \delta^2 e^{\lambda_{j_1}^1 t} \right) \right) \right)^{1/2} \end{aligned} \quad (4.21)$$

— if $\lambda_{j_1}^1 > 0$,

$$\begin{aligned}\delta_{j_1}(t) = & \frac{1}{(3\lambda_{j_1}^1)^{3/2}} \left(\frac{C_1^2}{\lambda_{j_1}^1} \left(-9(\lambda_{j_1}^1)^2 t^2 - 6\lambda_{j_1}^1 t + 2e^{3\lambda_{j_1}^1 t} - 2 \right) \right. \\ & + 3\lambda_{j_1}^1 \left(\frac{C_1 \gamma_{j_1}^1 |T_2|}{\lambda_{j_1}^1} \left(-3\lambda_{j_1}^1 t + e^{3\lambda_{j_1}^1 t} - 1 \right) \right. \\ & \left. \left. + 3\lambda_{j_1}^1 \left(\frac{(\gamma_{j_1}^1)^2 (|T_2|/2)^2}{\lambda_{j_1}^1} (e^{3\lambda_{j_1}^1 t} - 1) + 3\lambda_{j_1}^1 \delta^2 e^{3\lambda_{j_1}^1 t} \right) \right) \right)^{1/2} \quad (4.22)\end{aligned}$$

— if $\lambda_{j_1}^1 = 0$,

$$\begin{aligned}\delta_{j_1}(t) = & ((C_{j_1}^1)^2 (-t^2 - 2t + 2e^t - 2) \\ & + (C_{j_1}^1 \gamma_{j_1}^1 |T_2| (-t + e^t - 1) \\ & + ((\gamma_{j_1}^1)^2 (|T_2|/2)^2 (e^t - 1) + \delta^2 e^t)))^{1/2} \quad (4.23)\end{aligned}$$

A similar result can be established for sub-system 2, permitting to perform a distributed control synthesis.

Proof. In order to simplify the reading, we omit the mode j_1 (which does not intervene in the proof as long as $t \in [0, \tau]$) and write the proof for $f_{j_1}^1 = f_1$, $L_{j_1}^1 = L_1$, $C_{j_1}^1 = C_1$, $\lambda_{j_1}^1 = \lambda_1$. We have, for all $x_1, \tilde{x}_1 \in S_1^2$:

$$\begin{aligned}\frac{1}{2} \frac{d(\|x_1 - \tilde{x}_1\|^2)}{dt} &= \langle f_1(x_1, x_2) - f_1(\tilde{x}_1(0), x_2^m), x_1 - \tilde{x}_1 \rangle \\ &= \langle f_1(x_1, x_2) - f_1(\tilde{x}_1, x_2^m) + f_1(\tilde{x}_1, x_2^m) - f_1(\tilde{x}_1(0), x_2^m), x_1 - \tilde{x}_1 \rangle \\ &\leq \langle f_1(x_1, x_2) - f_1(\tilde{x}_1, x_2^m), x_1 - \tilde{x}_1 \rangle + \langle f_1(\tilde{x}_1, x_2^m) - f_1(\tilde{x}_1(0), x_2^m), x_1 - \tilde{x}_1 \rangle \\ &\leq \langle f_1(x_1, x_2) - f_1(\tilde{x}_1, x_2^m), x_1 - \tilde{x}_1 \rangle + \|f_1(\tilde{x}_1, x_2^m) - f_1(\tilde{x}_1(0), x_2^m)\| \|x_1 - \tilde{x}_1\| \\ &\leq \langle f_1(x_1, x_2) - f_1(\tilde{x}_1, x_2^m), x_1 - \tilde{x}_1 \rangle + L_1 \left\| \begin{pmatrix} \tilde{x}_1 \\ x_2^m \end{pmatrix} - \begin{pmatrix} \tilde{x}_1(0) \\ x_2^m \end{pmatrix} \right\| \|x_1 - \tilde{x}_1\| \\ &\leq \lambda_1 \|x_1 - \tilde{x}_1\|^2 + \gamma_1 \|x_2 - x_2^m\| \|x_1 - \tilde{x}_1\| + L_1 t \|f_1(\tilde{x}_1(0), x_2^m)\| \|x_1 - \tilde{x}_1\| \\ &\leq \lambda_1 \|x_1 - \tilde{x}_1\|^2 + \left(\gamma_1 \frac{|T_2|}{2} + C_1 t \right) \|x_1 - \tilde{x}_1\|\end{aligned}$$

where $|T_2|$ denotes the diameter of T_2 . Using the fact that $\|x_1 - \tilde{x}_1\| \leq \frac{1}{2}(\alpha \|x_1 - \tilde{x}_1\|^2 + \frac{1}{\alpha})$ for any $\alpha > 0$, we can write three formulas following the sign of λ_1 .

— if $\lambda_1 < 0$, we can choose $\alpha = \frac{-\lambda_1}{C_1 t + \gamma_1 |T_2|/2}$, and we get the differential inequality:

$$\frac{d(\|x_1 - \tilde{x}_1\|^2)}{dt} \leq \lambda_1 \|x_1 - \tilde{x}_1\|^2 + \frac{C_1^2}{-\lambda_1} t^2 + \frac{C_1 \gamma_1 |T_2|}{-\lambda_1} t + \frac{\gamma_1^2 (|T_2|/2)^2}{-\lambda_1}$$

— if $\lambda_1 > 0$, we can choose $\alpha = \frac{\lambda_1}{C_1 t + \gamma_1 |T_2|/2}$, and we get the differential inequality:

$$\frac{d(\|x_1 - \tilde{x}_1\|^2)}{dt} \leq 3\lambda_1 \|x_1 - \tilde{x}_1\|^2 + \frac{C_1^2}{\lambda_1} t^2 + \frac{C_1 \gamma_1 |T_2|}{\lambda_1} t + \frac{\gamma_1^2 (|T_2|/2)^2}{\lambda_1}$$

— if $\lambda_1 = 0$, we can choose $\alpha = \frac{1}{C_1 t + \gamma_1 |T_2|/2}$, and we get the differential inequality:

$$\frac{d(\|x_1 - \tilde{x}_1\|^2)}{dt} \leq \|x_1 - \tilde{x}_1\|^2 + C_1^2 t^2 + C_1 \gamma_1 |T_2| t + \gamma_1^2 (|T_2|/2)^2$$

In every case, the differential inequalities can be integrated to obtain the formulas of the theorem.

□

It then follows a distributed version of Corollary 2.

Corollary 4. *Given a positive real δ , consider two sets of points $\tilde{x}_1^1, \dots, \tilde{x}_{m_1}^1$ and $\tilde{x}_1^2, \dots, \tilde{x}_{m_2}^2$ such that all the balls $B(\tilde{x}_{i_1}^1, \delta)$ and $B(\tilde{x}_{i_2}^2, \delta)$, for $1 \leq i_1 \leq m_1$ and $1 \leq i_2 \leq m_2$, cover R_1 and R_2 . Suppose that there exists patterns $\pi_{i_1}^1$ and $\pi_{i_2}^2$ of length k_{i_1} and k_{i_2} such that :*

1. $B((\tilde{x}_{i_1}^1)_{\pi_{i_1}^1}^{k'}, \delta_{\pi_{i_1}^1}^{k'}) \subseteq S_1$, for all $k' = 1, \dots, k_{i_1} - 1$
2. $B((\tilde{x}_{i_1}^1)_{\pi_{i_1}^1}^{k_{i_1}}, \delta_{\pi_{i_1}^1}^{k_{i_1}}) \subseteq R_1$.
3. $\frac{d^2(\delta'_{j_1}(t))}{dt^2} > 0$ with $j_1 = \pi_{i_1}^1(k')$ and $\delta' = \delta_{\pi_{i_1}^1}^{k'-1}$, for all $k' \in \{1, \dots, k_{i_1}\}$ and $t \in [0, \tau]$.
1. $B((\tilde{x}_{i_2}^2)_{\pi_{i_2}^2}^{k'}, \delta_{\pi_{i_2}^2}^{k'}) \subseteq S_2$, for all $k' = 1, \dots, k_{i_2} - 1$
2. $B((\tilde{x}_{i_2}^2)_{\pi_{i_2}^2}^{k_{i_2}}, \delta_{\pi_{i_2}^2}^{k_{i_2}}) \subseteq R_2$.
3. $\frac{d^2(\delta'_{j_2}(t))}{dt^2} > 0$ with $j_2 = \pi_{i_2}^2(k')$ and $\delta' = \delta_{\pi_{i_2}^2}^{k'-1}$, for all $k' \in \{1, \dots, k_{i_2}\}$ and $t \in [0, \tau]$.

The above properties induce a distributed control $\sigma = (\sigma_1, \sigma_2)$ guaranteeing (non simultaneous) recurrence in R and safety in S . I.e.

- if $x \in R$, then $\phi_\sigma(t; x) \in S$ for all $t \geq 0$
- if $x \in R$, then $\phi_\sigma(k_1 \tau; x)|_1 \in R_1$ for some $k_1 \in \{k_{i_1}, \dots, k_{i_{m_1}}\}$, and symmetrically $\phi_\sigma(k_2 \tau; x)|_2 \in R_2$ for some $k_2 \in \{k_{i_2}, \dots, k_{i_{m_2}}\}$

4.3.2 Application

We demonstrate the feasibility of our approach on the (linearized) building ventilation application adapted from [134], given in Appendix A.5, with constant parameters $T_o = 30$, $T_c = 30$, $T_u = 17$, $\delta_{s_i} = 1$ for $i \in \mathcal{N}$. The centralized controller was obtained with 256 balls in 48 seconds, the distributed controller was obtained with $16 + 16$ balls in less than a second. In both cases, patterns of length 2 are used. A sub-sampling of $h = \tau/20$ is required to obtain a controller with the centralized approach. For the distributed approach, no sub-sampling is required for the first sub-system, while the second one requires a sub-sampling of $h = \tau/10$. Simulations of the centralized and distributed controllers are given in Figure 4.12, where the control objective is to stabilize the temperature in $[20, 22]^4$ while never going out of $[19, 23]^4$.

Table 4.1: Numerical results for centralized four-room example.

	Centralized
R	$[20, 22]^4$
S	$[19, 23]^4$
τ	30
Time subsampling	$\tau/20$
Complete control	Yes
Error parameters	$\max_{j=1,\dots,16} \lambda_j = -6.30 \times 10^{-3}$ $\max_{j=1,\dots,16} C_j = 4.18 \times 10^{-6}$
Number of balls/tiles	256
Pattern length	2
CPU time	48 seconds

Table 4.2: Numerical results for the distributed four-room example.

	Sub-system 1	Sub-system 2
R	$[20, 22]^2 \times [20, 22]^2$	
S	$[19, 23]^2 \times [19, 23]^2$	
τ	30	
Time subsampling	No	$\tau/10$
Complete control	Yes	Yes
Error parameters	$\max_{j_1=1,\dots,4} \lambda_{j_1}^1 = -1.39 \times 10^{-3}$ $\max_{j_1=1,\dots,4} \gamma_{j_1}^1 = 1.79 \times 10^{-4}$ $\max_{j_1=1,\dots,4} C_{j_1}^1 = 4.15 \times 10^{-4}$	$\max_{j_2=1,\dots,4} \lambda_{j_2}^2 = -1.42 \times 10^{-3}$ $\max_{j_2=1,\dots,4} \gamma_{j_2}^2 = 2.47 \times 10^{-4}$ $\max_{j_2=1,\dots,4} C_{j_2}^2 = 5.75 \times 10^{-4}$
Number of balls/tiles	16	16
Pattern length	2	2
CPU time	< 1 second	< 1 second

4.3.3 Final remarks and future work

We have given a new distributed control synthesis method based on Euler's method. The method makes use of the notions of δ -ISS-stability and ISS Lyapunov functions. From a certain point of view, this method is along the lines of [53] and [101] which are inspired by small-gain theorems of control theory (see, *e.g.*, [97]). In the future, we plan to apply our distributed Euler-based method to significant examples such as the 11-room example of Appendix A.9.

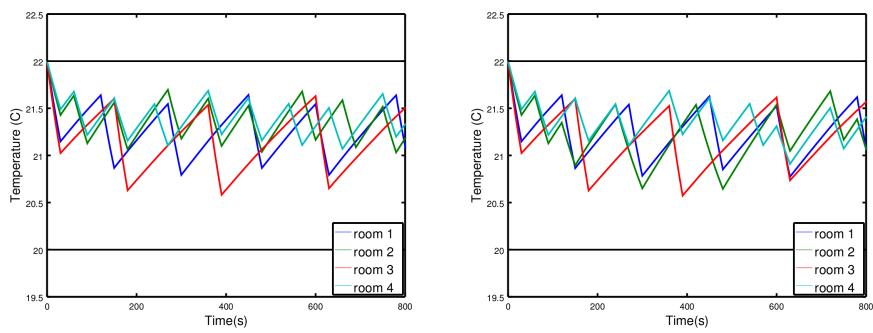


Figure 4.12: Simulation of the centralized (left) and distributed (right) controllers from the initial condition (22, 22, 22, 22).

Chapter 5

Control of high dimensional ODEs

In this chapter, we aim at extending the previous works to the control synthesis of partial differential equations, mainly used to model mechanical systems. While the models of switched systems are usually used for (low dimensional) ordinary differential equations controlled with a piecewise constant function, it is also possible to use these models for the control of mechanical systems. Indeed, the dynamics of most mechanical systems can be modeled by partial differential equations, and the spacial discretization of such systems leads to high dimensional ODEs. Controlled with a piecewise constant function on the boundary, and written in a proper way (the state space representation), one obtains high dimensional switched control systems. As stated in Chapter 3, the computational cost of the synthesis algorithms is exponential in the dimension of the system. Whether a finite element, a finite difference, or any discretization method is used, an accurate discretized model of a mechanical system leads to ODEs of dimension larger than 1000. The dimension of real case studies used in industry often exceeds 10^6 . It is thus irrelevant to directly use the algorithms of Chapter 3 to discretized PDEs. A model order reduction is required in order to synthesize a controller at the reduced-order level. In this chapter, linear systems are considered, and we use the reachability computations of Chapter 3.1 since they provide the most accurate results. Two methods are proposed: a fully offline procedure, and a semi-online procedure requiring online state estimation. The state is first supposed known at each time point, we then provide a first step to the use of state observers (i.e. partial observation). Note that the synthesis is always performed offline, we refer to semi-online because the application of the induced controller requires online state estimation.

Comparison with related work.

Model order reduction techniques for hybrid or switched systems are classically used in *numerical simulation* in order to construct, at the reduced level, trajectories which cannot be computed directly at the original level due to complexity and large size dimension [16, 46]. Model reduction is used in order to perform *set-based reachability analysis* in [85]. Isolated trajectories issued from isolated points are not constructed, instead, (an over-approximation of) the infinite set of trajectories is derived from a dense set of initial points. This allows to perform formal verification of properties such as *safety*. In both approaches, the control is *given* as an input of the problem. In contrast here, the control is *synthesized* using set-based methods in order to achieve by construction properties such as *convergence* and *stability*.

While symbolic approaches are mostly used for the control of low order ODEs, the control of mechanical systems can be realized using the control theory approach, where a continuous control law is guessed and proved to be efficient on the continuous PDE model [22, 111, 164]. The damping of vibrations with piezoelectric devices is in particular a widely developed branch of the control of mechanical systems. The shunting of piezoelectric devices with electric circuits permits to convert the

vibration energy into electrical energy, which is then dissipated in the electric circuits [83]. Note that this approach can be active or passive, depending on the electric energy furnished to the electric circuit. A switched control approach is developed in [47, 152], the piezoelectric device is shunted on several electric circuits, but only one is selected at a time depending on the state of the mechanical system. This approach is called semi-active since the electric circuits are passive but the switching requires energy. In the present chapter, the approach is fully active.

Plan.

In Section 5.1, we give some preliminaries on switched control systems and their link with PDEs and mechanical systems. In Section 5.2, we introduce some elements of control theory and the state-space bisection method. In Section 5.3, we explain how to construct a reduced model, apply the state-space bisection method at this level, and compute upper bounds to the error induced at the original level. In Section 5.4, we propose two methods of control synthesis allowing to synthesize (either offline or online) a controller at the reduced-order level and apply it to the full-order system. In Section 5.5, we apply our approach to several examples of the literature. In section 5.6, we extend our method to the use of observers. We conclude in Section 5.7.

5.1 Background

We consider systems governed by Partial Differential Equations (PDEs) having actuators allowing to impose forces on the boundary; these systems can represent transient thermal problems, vibration problems... By applying the right external force at the right time, one can drive the system to a desired operating mode. Our goal here is to synthesize a law which, given the state of the system, computes the boundary force to apply.

In order to illustrate our approach, we use the example of the heat equation:

$$\begin{cases} \frac{\partial T}{\partial t}(x, t) - \alpha \Delta T(x, t) = 0 & \forall (t, x) \in [0, T] \times \Omega \\ T(x, \cdot) = T^d(x, \cdot) & \forall x \in \partial \Omega^T \\ \frac{\partial T}{\partial x}(x, \cdot).n = \varphi^d(x, \cdot) & \forall x \in \partial \Omega^\varphi \\ T(x, 0) = T_0(x) \end{cases} \quad (5.1)$$

Discretized by finite elements, the nodal temperatures $\{T\}$ are computed with respect to time, and the system becomes:

$$\begin{cases} C_{FE}\{\dot{T}\} + K_{FE}\{T\} = \{F^d\} \\ \{T(0)\} = \{T_0\} \end{cases} \quad (5.2)$$

The purpose is then to compute the forces $\{F^d\}$ with respect to time such that the temperature field verifies some desired properties.

For example, one may want to impose that the temperature in a particular node remains within a given temperature range. Usually, the quantities of interest one wants to control are given in discrete points, which are for example sensor measurements, or they are given as local averaging. Here, we consider the case where the quantities of interest can be directly extracted from the nodal values with a matrix called *output matrix* (see equation (5.3)).

We consider a particular kind of actuators; the force applied only takes a finite number N of values. For example, in (5.1) for the case of a room heated with a heater, the flux φ^d is equal to 0 when the heater is turned off and equal to a positive value when it is turned on. The control systems associated to such behaviors are naturally written under the form of switched systems (2.1). Focusing on linear PDEs, the addition of an output leads to a system of the form:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \quad (5.3)$$

The n -vector x is called the state of the system, the p -vector u is the control input, the m -vector y is the output of the system, A is an $n \times n$ -matrix, B an $n \times p$ -matrix, and C an $m \times n$ matrix. Writing the discretized equation (5.2) under this form is straightforward by multiplying the first line by C_{FE}^{-1} (which is invertible), and the state vector is then $\{T\}$. In the case of higher order PDEs (for example in the case of the wave equation), we merely need to enlarge the state vector to take the first derivative of the nodal values in it.

5.2 Problem setting

We will synthesize controllers using adaptations of Algorithms 1 and 2 by adding constraints on the outputs of the system.

The entries of the problem are the following:

1. a subset $R_x \subset \mathbb{R}^n$ of the state space, called *interest set*,
2. a subset $R_y \subset \mathbb{R}^m$ of the output space, called *objective set*.

The objective is to find a law $u(\cdot)$ which, for any initial state $x_0 \in R_x$, stabilizes the output y in the set R_y . The set R_x is in fact the set of all initial conditions considered, and the set R_y is a target set, where we want the output to stabilize. The sets R_x and R_y are given under the form of boxes, i.e. interval products of \mathbb{R}^n and \mathbb{R}^m respectively.

In the remainder of this chapter, we will denote control patterns by $Pat \in U^k$ for some $k \geq 1$ in order to avoid confusion with projectors, classically denoted by π . We extend the definition of the Post operator for outputs as follows: the *output successor set* of a set $X \subset \mathbb{R}^n$ of states under switching mode u is:

$$Post_{u,C}(X) = \bigcup_{x_0 \in X} C\phi_u(t; t_0, x_0).$$

We similarly extend this definition for sequences of inputs (patterns) $Pat \in U^k$ for some $k \geq 1$:

$$Post_{Pat,C}(X) = \bigcup_{x_0 \in X} C\phi_{Pat}(t; t_0, x_0).$$

With these definitions and notations, we are now able to present the adaptations of the algorithm presented in Chapter 2. It relies on the decomposition of the set R_x . Given the sets R_x and R_y , and a maximum length of input pattern K , it returns a set Δ of the form $\{(V_i, Pat_i)\}_{i \in I}$ where I is a finite set of indices. Each V_i is a subset of R_x and each Pat_i is a pattern of length at most K , such that:

- (a) $\bigcup_{i \in I} V_i = R_x$,
- (b) for all $i \in I$: $Post_{Pat_i}(V_i) \subseteq R_x$,
- (c) for all $i \in I$: $Post_{Pat_i,C}(V_i) \subseteq R_y$.

The algorithm thus returns several sets V_i that cover R_x , and each V_i is associated to a pattern Pat_i that sends V_i in R_x , and the output in R_y . The set R_x is thus decomposed in several sets, and for each one, we have one control law: $\forall x \in V_i, u(x) = Pat_i$. Therefore, for two initial conditions in a set V_i , we apply the same input pattern. The fact that we use set based operations has a key role which allows us to consider sets of initial conditions, and this is how we manage to obtain a law $u(x)$. In the following, when a decomposition Δ is successfully obtained, we denote by u_Δ the induced control law.

Algorithms 4 and 5 show the main functions used by the state-space decomposition algorithm. Note that function “Decomposition” now takes an additional input R_y . When looking for stabilizing patterns, we add the more restrictive constraint that the output of the system is sent in R_y .

At the beginning, the function “Decomposition” calls sub-function “Find_Pattern” in order to get a k -pattern (a pattern of length up to k) Pat such that $Post_{Pat}(R_x) \subseteq R_x$ and $Post_{Pat,C}(R_x) \subseteq R_y$. If it succeeds, then it is done. Otherwise, it divides R_x into 2^n sub-boxes V_1, \dots, V_{2^n} of equal size. If for each V_i , Find_Pattern gets a k -pattern Pat_i such that $Post_{Pat_i}(V_i) \subseteq R_x$ and $Post_{Pat_i,C}(V_i) \subseteq R_y$, it is done. If, for some V_j , no such input pattern exists, the function is recursively applied to V_j . It ends with success when a successful decomposition of (R_x, R_y, k) is found, or failure when the maximal degree d of bisection is reached. The main function Bisection(W, R_x, R_y, D, K) is called with R_x as input value for W , d for input value for D , and k as input value for K ; it returns either $\langle \{(V_i, Pat_i)\}_i, True \rangle$ with

$$\begin{aligned} \bigcup_i V_i &= W, \\ \bigcup_i Post_{Pat_i}(V_i) &\subseteq R_x, \\ \bigcup_i Post_{Pat_i,C}(V_i) &\subseteq R_y \end{aligned}$$

when it succeeds, or $\langle _, False \rangle$ when it fails. Function Find_Pattern(W, R_x, R_y, K)

looks for a K -pattern Pat for which $Post_{Pat}(W) \subseteq R_x$ and $Post_{Pat,C}(W) \subseteq R_y$: it selects all the K -patterns by increasing length order until either it finds such an input pattern Pat (output: $\langle Pat, True \rangle$), or none exists (output: $\langle _, False \rangle$).

Algorithm 4 Decomposition(W, R_x, R_y, D, K)

Input: A box W , a box R_x , a box R_y , a degree D of bisection, a length K of input pattern

Output: $\langle \{(V_i, Pat_i)\}_i, True \rangle$ with $\bigcup_i V_i = W$, $\bigcup_i Post_{Pat_i}(V_i) \subseteq R_x$ and $\bigcup_i Post_{Pat_i,C}(V_i) \subseteq R_y$, or $\langle _, False \rangle$

$$(Pat, b) := Find_Pattern(W, R_x, R_y, K)$$

if $b = True$ **then**

return $\langle \{(W, Pat)\}, True \rangle$

else

if $D = 0$ **then**

return $\langle _, False \rangle$

else

Divide equally W into (W_1, \dots, W_{2^n})

for $i = 1 \dots 2^n$ **do**

$(\Delta_i, b_i) :=$ Decomposition($W_i, R_x, R_y, D - 1, K$)

end for

return $(\bigcup_{i=1 \dots 2^n} \Delta_i, \bigwedge_{i=1 \dots 2^n} b_i)$

end if

end if

5.3 Model order reduction

As seen in Chapter 2, the main drawback of the previous state-space decomposition algorithm is the computational cost, with a complexity in $O(2^{nd}N^k)$, with n the state-space dimension, d the maximum degree of decomposition, N the number of modes and k the maximum length of researched patterns. It is thus subject to the *curse of dimensionality*. In practice, the dimension n must be lower than 10 for acceptable computation times. Thus, by directly applying the bisection algorithm to a discretized PDE, the number of degrees of freedom is limited to 10 for a first order PDE, and even less for a higher order PDE written in state-space representation. The use of a Model Order Reduction (MOR) is thus unavoidable.

We choose here to use *projection-based* model order reduction methods [16]. Given a full-order system Σ , an interest set $R_x \subset \mathbb{R}^n$ and an objective set $R_y \subset \mathbb{R}^m$, we construct a reduced-order system $\hat{\Sigma}$ using a projection π of \mathbb{R}^n to \mathbb{R}^{n_r} . If $\pi \in \mathbb{R}^{n \times n}$ is a projection, it verifies $\pi^2 = \pi$, and π can be written as $\pi = \pi_L \pi_R$, where $\pi_L \in \mathbb{R}^{n \times n_r}$, $\pi_R \in \mathbb{R}^{n_r \times n}$ and $n_r = \text{rank}(\pi)$. The reduced-order system $\hat{\sigma}$ is then

Algorithm 5 Find_Pattern(W, R_x, R_y, K)

Input: A box W , a box R_x , a box R_y , a length K of input pattern

Output: $\langle Pat, True \rangle$ with $Post_{Pat}(W) \subseteq R_x, Post_{Pat,C}(W) \subseteq R_y$ and $Unf_{Pat}(W) \subseteq S$, or $\langle -, False \rangle$ when no input pattern maps W into R_x and CW into R_y

for $i = 1 \dots K$ **do**

$\Pi :=$ set of input patterns of length i

while Π is non empty **do**

Select Pat in Π

$\Pi := \Pi \setminus \{Pat\}$

if $Post_{Pat}(W) \subseteq R_x$ and $Post_{Pat,C}(W) \subseteq R_y$ **then**

return $\langle Pat, True \rangle$

end if

end while

end for

return $\langle -, False \rangle$

obtained by the change of variable $\hat{x} = \pi_R x$:

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), \\ y_r(t) &= \hat{C}\hat{x}(t), \end{cases}$$

with

$$\hat{A} = \pi_R A \pi_L, \quad \hat{B} = \pi_R B, \quad \hat{C} = C \pi_L.$$

The projection π can be constructed by multiple methods: Proper Orthogonal Decomposition [48, 98], balanced truncation [15, 29, 30, 140], balanced POD [172]... We use here the balanced truncation method, widely used in the control community and particularly adapted to the models used here, written under state-space representation.

The objective is now to compute a decomposition at the low order level, and apply the induced reduced control to the full order system. In order to ensure that the reduced control is effective, we introduce the following notations, simplifying the reading of the remainder of this chapter:

- $\mathbf{x}(t, x, u)$ denotes the point reached by Σ at time t under mode $u \in U$ from the initial condition x .
- $\hat{\mathbf{x}}(t, \hat{x}, u)$ denotes the point reached by $\hat{\Sigma}$ at time t under mode $u \in U$ from the initial condition \hat{x} .
- $\mathbf{y}(t, x, u)$ denotes the output point reached by Σ at time t under mode $u \in U$ from the initial condition x .
- $\mathbf{y}_r(t, \hat{x}, u)$ denotes the output point reached by $\hat{\Sigma}$ at time t under mode $u \in U$ from the initial condition \hat{x} .

When a control u is applied to both full-order and reduced-order systems, an error between the output trajectories $\mathbf{y}(t, x, u)$ and $\mathbf{y}_r(t, \pi_R x, u)$ is unavoidable, and we denote it by $e_y(t, x, u)$. A first tool to ensure the effectiveness of the reduced-order control is to compute a bound on $\|e_y(t, x, u)\|$. A second source of error is the deviation between $\pi_R \mathbf{x}(t, x, u)$ and $\hat{\mathbf{x}}(t, \pi_R x, u)$, which we denote by $e_x(t, x, u)$. Computing a bound on $\|e_x(t, x, u)\|$ will also be necessary. Before establishing these error bounds, we first briefly describe the balanced truncation method. We then present how we compute a reduced-order control and apply it to the full-order system.

5.3.1 The balanced truncation

Applying the balanced truncation consists in balancing then truncating the system. Balancing the system requires finding balancing transformations which diagonalize the controllability and observability gramians of the system in the same basis.

The controllability and observability gramians W_c and W_o of the system Σ are respectively the solutions of the dual (infinite-time horizon) Lyapunov equations

$$AW_c + W_c A^\top + BB^\top = 0 \quad (5.4)$$

and

$$A^\top W_o + W_o A + C^\top C = 0 \quad (5.5)$$

The balancing transformations π_R and π_L are then computed as follows [30]:

1. Compute the Cholesky factorization $W_c = UU^\top$
2. Compute the eigenvalue decomposition of $U^\top W_o U$

$$U^\top W_o U = K\sigma^2 K^\top$$

where the entries in σ are ordered by decreasing order

3. Compute the transformations

$$\pi_R = \sigma^{-\frac{1}{2}} K^\top U^{-1}$$

$$\pi_L = UK\sigma^{-\frac{1}{2}}$$

One can then verify that

$$\pi_R W_c \pi_R^\top = \pi_L^\top W_o \pi_L = \sigma$$

and σ contains the Hankel singular values of the system.

Computing the balancing transformations for large scale systems derived for example from discretized partial differential equations is usually very expensive - even sometimes irrelevant - and many advances have been carried out in order to solve the Lyapunov equations and compute the transformations with approximate methods, often based on Krylov subspace methods (see for example [15, 29, 146]).

5.3.2 Error bounding

Error bounding for the output trajectory

Here, a scalar *a posteriori* error bound for e_y is given (mainly inspired from [85]). The error bound ε_y can be computed from simulations of the full and reduced-order systems. The computation time for simulations is negligible compared with that of the bisection method to generate the decompositions.

Computing an upper bound of $\|e_y(t, x, u)\|$ is equivalent to seeking the solution of the following (optimal control) problem:

$$\begin{aligned}\varepsilon_y(t) &= \sup_{u \in U, x_0 \in R_x} \|e(t, x_0, u)\| \\ &= \sup_{u \in U, x_0 \in R_x} \|\mathbf{y}(t, x_0, u) - \mathbf{y}_r(t, \pi_R x_0, u)\|.\end{aligned}$$

Since the full-order and reduced-order systems are linear, one can use a superposition principle and the error bound can be estimated as $\varepsilon_y(t) \leq \varepsilon_y^{x_0=0}(t) + \varepsilon_y^{u=0}(t)$ where $\varepsilon_y^{x_0=0}$ is the error of the zero-state response, given by (see [85])

$$\begin{aligned}\varepsilon_y^{x_0=0}(t) &= \max_{u \in U} \|u\| \cdot \|e_y(t, x_0 = 0, u)\| \\ &= \max_{u \in U} \|u\| \cdot \|\mathbf{y}(t, 0, u) - \mathbf{y}_r(t, 0, u)\|,\end{aligned}$$

and $\varepsilon_y^{u=0}$ is the error of the zero-input response, given by

$$\begin{aligned}\varepsilon_y^{u=0}(t) &= \sup_{x_0 \in R_x} \|e_y(t, x_0, u = 0)\| \\ &= \sup_{x \in R_x} \|\mathbf{y}(t, x_0, 0) - \mathbf{y}_r(t, \pi_R x_0, 0)\|.\end{aligned}$$

Using some algebraic manipulations (see [85]), one can find a precise bound for $\varepsilon_y^{x_0=0}$ and $\varepsilon_y^{u=0}$:

$$\varepsilon_y^{x_0=0}(t) \leq \|u(\cdot)\|_\infty^{[0,t]} \int_0^t \left\| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & B \\ e^{t\hat{A}} & \hat{B} \end{bmatrix} \right\| dt, \quad (5.6)$$

$$\varepsilon_y^{u=0}(t) \leq \sup_{x_0 \in R_x} \left\| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & B \\ e^{t\hat{A}} & \hat{B} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \right\|. \quad (5.7)$$

The first error bound (5.6) always increases with time whereas the second bound (5.7) can either increase or decrease. These properties are used to compute a guaranteed bound. For all $j \in \mathbb{N}$ (j corresponds to the length of the pattern applied), we have:

$$\varepsilon_y(j\tau) \leq \varepsilon_y^j$$

with

$$\begin{aligned}\varepsilon_y^j &= \|u(\cdot)\|_\infty^{[0,j\tau]} \int_0^{j\tau} \left\| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{tA} & B \\ e^{t\hat{A}} & \hat{B} \end{bmatrix} \right\| dt \\ &\quad + \sup_{x_0 \in R_x} \left\| \begin{bmatrix} C & -\hat{C} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & B \\ e^{j\tau \hat{A}} & \hat{B} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \right\|. \quad (5.8)\end{aligned}$$

Furthermore, we have:

$$\forall t \geq 0, \quad \varepsilon_y(t) \leq \varepsilon_y^\infty$$

with

$$\varepsilon_y^\infty = \sup_{t \geq 0} \varepsilon_y(t). \quad (5.9)$$

This bound exists when the modulus of the eigenvalues of $e^{\tau A}$ and $e^{\tau \hat{A}}$ is strictly inferior to one, which we suppose here.

Error bounding for the state trajectory

Denoting by $j \in \mathbb{N}$ the length of the pattern applied, the following results holds:

$$\mathbf{x}(t = j\tau, x, u) = e^{j\tau A}x + \int_0^{j\tau} e^{A(j\tau-t)}Bu(t)dt,$$

$$\hat{\mathbf{x}}(t = j\tau, \pi_R x, u) = e^{j\tau \hat{A}}\pi_R x + \int_0^{j\tau} e^{\hat{A}(j\tau-t)}\hat{B}u(t)dt,$$

Using an approach similar to the construction of the bounds (5.6) and (5.7), we obtain the following bound, which depends on the length j of the pattern applied:

$$\|\pi_R \mathbf{x}(t = j\tau, x, u) - \hat{\mathbf{x}}(t = j\tau, \pi_R x, u)\| \leq \varepsilon_x^j, \quad (5.10)$$

with

$$\begin{aligned} \varepsilon_x^j &= \|u(\cdot)\|_\infty^{[0, j\tau]} \int_0^{j\tau} \left\| \begin{bmatrix} \pi_R & -I_{n_r} \end{bmatrix} \begin{bmatrix} e^{tA} & \\ & e^{t\hat{A}} \end{bmatrix} \begin{bmatrix} B \\ \hat{B} \end{bmatrix} \right\| dt \\ &\quad + \sup_{x_0 \in R_x} \left\| \begin{bmatrix} \pi_R & -I_{n_r} \end{bmatrix} \begin{bmatrix} e^{j\tau A} & \\ & e^{j\tau \hat{A}} \end{bmatrix} \begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix} \right\|. \end{aligned} \quad (5.11)$$

Remark: in order to simplify the reading, the notation $|Pat|$ will often be used in the following to denote the length of the pattern Pat .

5.4 Reduced order control

Two procedures are proposed for synthesizing reduced-order controllers: (i) an offline procedure, consisting in computing a complete sequence of control inputs for a given initial condition; (ii) a semi-online procedure, where the patterns are computed through online projection of the full-order state. We describe these approaches in the following subsections.

5.4.1 Offline procedure

Suppose that we are given a system Σ , an interest set R_x , and an objective set R_y . The reduced-order system $\hat{\Sigma}$ of order n_r , obtained by balanced truncation, is written under the form of equation (5.3):

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t), \\ y_r(t) = \hat{C}\hat{x}(t), \end{cases}$$

where $\hat{A} = \pi_R A \pi_L \in \mathbb{R}^{n_r \times n_r}$, $\hat{B} = \pi_R B \in \mathbb{R}^{n_r \times p}$, $\hat{C} = C \pi_L \in \mathbb{R}^{m \times n_r}$.

We denote by \hat{R}_x the projection of R_x . Given the interest set \hat{R}_x , the objective set R_y and a maximal length of researched pattern K , the application of the state-space decomposition algorithm to the reduced system returns, when it succeeds, a decomposition $\hat{\Delta}$ of the form $\{\hat{V}_i, Pat_i\}_{i \in I}$, with I a finite set of indices, such that:

1. $\bigcup_{i \in I} \hat{V}_i = \hat{R}_x$,
2. for all $i \in I$: $Post_{Pat_i}(\hat{V}_i) \subseteq \hat{R}_x$,
3. for all $i \in I$: $Post_{Pat_i, \hat{C}}(\hat{V}_i) \subseteq R_y$.

The decomposition $\hat{\Delta}$ induces a control $u_{\hat{\Delta}}$ on \hat{R}_x . Applied on the reduced-order system $\hat{\Sigma}$, the control $u_{\hat{\Delta}}$ keeps \hat{x} in \hat{R}_x and sends y_r in R_y . This control can be applied to the full-order system in two steps: a sequence of patterns is computed on the reduced-order system, and it is then applied to the full order system:

- (a) Let x_0 be an initial condition in R_x . Let $\hat{x}_0 = \pi_R x_0$ be its projection belonging to \hat{R}_x , $\hat{x}_0 = \pi_R x_0$ is the initial condition for the reduced system $\hat{\Sigma}$: \hat{x}_0 belongs to \hat{V}_{i_0} for some $i_0 \in I$; thus, after applying Pat_{i_0} , the system is led to a state \hat{x}_1 ; \hat{x}_1 belongs to \hat{V}_{i_1} for some $i_1 \in I$; and iteratively, we build, from an initial state \hat{x}_0 , a sequence of states $\hat{x}_1, \hat{x}_2, \dots$ obtained by application of the sequence of k -patterns $Pat_{i_0}, Pat_{i_1}, \dots$ (steps (1), (2) and (3) of Figure 5.1).
- (b) The sequence of k -patterns is computed for the reduced system $\hat{\Sigma}$, but it can be applied to the full-order system Σ : we build, from an initial point x_0 , a sequence of points x_1, x_2, \dots by application of the k -patterns $Pat_{i_0}, Pat_{i_1}, \dots$ (steps (4), (5) and (6) of Figure 5.1). Moreover, for all $x_0 \in R_x$ and for all $t \geq 0$, the error $\|\mathbf{y}(t, x_0, u) - \mathbf{y}_r(t, \pi_R x_0, u)\|$ is bounded by ε_y^∞ , as defined in equation (5.9).

This procedure thus allows, for any system Σ of the form (5.3), and given an interest set R_x and an objective set R_y , to send the output of the full-order system in the set $R_y + \varepsilon_y^\infty$. More precisely, if $\hat{\Sigma}$ is the projection by balanced truncation of Σ , let $\hat{\Delta}$ be a decomposition for (\hat{R}_x, R_y, k) w.r.t. $\hat{\Sigma}$. Then, for all $x_0 \in R_x$, the induced control $u_{\hat{\Delta}}$ applied to the full-order system Σ in x_0 is such that for all $j > 0$, the output of the full-order system $y(t)$ returns to $R_y + \varepsilon_y^\infty$ after at most $k \tau$ -steps.

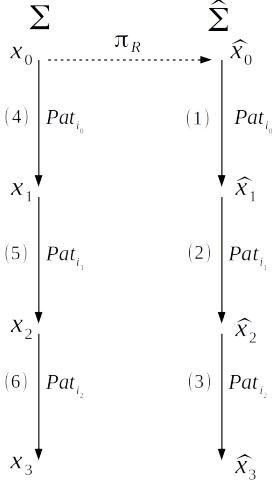


Figure 5.1: Diagram of the offline procedure for a simulation of length 3.

Here, $R_y + \varepsilon_y^\infty$ denotes the set containing R_y with a margin of ε_y^∞ . If R_y is an interval product of the form $[a_1, b_1] \times \cdots \times [a_m, b_m]$, then $R_y + \varepsilon_y^\infty$ is defined by $[a_1 - \varepsilon_y^\infty, b_1 + \varepsilon_y^\infty] \times \cdots \times [a_m - \varepsilon_y^\infty, b_m + \varepsilon_y^\infty]$.

Remark: Here, we ensure that $\mathbf{y}(t, x_0, u)$ is in $R_y + \varepsilon_y^\infty$ at the end of each pattern, but an easy improvement is to ensure that $\mathbf{y}(t, x_0, u)$ stays in a safety set $S_y \supset R_y$ at each step of time $k\tau$. Indeed, as explained in [67], we can ensure that the unfolding of the output trajectory stays in a given safety set S_y . The unfolding of the output of a set is defined as follows: given a pattern Pat of the form $(u_1 \cdots u_m)$, and a set $X \subset \mathbb{R}^n$, the *unfolding of the output of X via Pat* , denoted by $Unf_{Pat,C}(X)$, is the set $\bigcup_{i=0}^m X_i$ with:

- $X_0 = \{Cx | x \in X\}$,
- $X_{i+1} = Post_{u_{i+1},C}(X_i)$, for all $0 \leq i \leq m-1$.

The unfolding thus corresponds to the set of all the intermediate outputs produced when applying pattern Pat to the states of X . In order to guarantee that $\mathbf{y}(t, x_0, u)$ stays in S_y , we just have to make sure that $\mathbf{y}_r(t, \pi_R x_0, u)$ stays in the reduced safety set $S_y - \varepsilon_y^\infty$. We thus have to add, in the line 6 of Algorithm 5, the condition: “and $Unf_{Pat,C}(W) \subset S_y - \varepsilon_y^\infty$ ”.

5.4.2 Semi-online procedure

Up to this point, the procedure of control synthesis consists in computing a complete sequence of patterns on the reduced order model $\hat{\Sigma}$ for a given initial state x_0 , and applying the pattern sequence to the full-order model Σ . The entire control law is thus computed offline. While the decomposition is always performed offline, one can however use the decomposition $\hat{\Delta}$ online as follows: let x_0 be the initial state in R_x and $\hat{x}_0 = \pi_R x_0$ (step (1) of Figure 5.2) its projection belonging to \hat{R}_x , \hat{x}_0 belongs to \hat{V}_{i_0} for some $i_0 \in I$; we can thus apply the associated pattern Pat_{i_0} to the full-order

system Σ , which yields a state $x_1 = \mathbf{x}(|Pat_{i_0}| \tau, x_0, Pat_{i_0})$ (step (2) of Figure 5.2), the corresponding output is sent to $y_1 = \mathbf{y}(|Pat_{i_0}| \tau, x_0, Pat_{i_0}) \in R_y + \varepsilon_y^{|Pat_{i_0}|}$; in order to continue to step (3), we have to guarantee that $\pi_R \mathbf{x}(|Pat_i| \tau, x, Pat_i)$ belongs to \hat{R}_x for all $x \in R_x$ and for all $i \in I$. As explained below, this is possible using the computation of an upper bound to the error $\|\pi_R \mathbf{x}(|Pat_i| \tau, x, Pat_i) - \hat{\mathbf{x}}(|Pat_i| \tau, \pi_R x, Pat_i)\|$ and a reinforcement of the procedure for taking into account this error.

Let $\varepsilon_x^{|Pat|}$ be the upper bound to

$$\|\pi_R \mathbf{x}(|Pat| \tau, x, Pat) - \hat{\mathbf{x}}(|Pat| \tau, \pi_R x, Pat)\|,$$

as defined in equation (5.11). We modify the Algorithms 4 and 5, which become “Bisection_Dyn” and “Find_Pattern_Dyn” (Algorithms 6 and 7), they are computed with an additional input $\varepsilon_x = (\varepsilon_x^1, \dots, \varepsilon_x^k)$, k being the maximal length of the patterns. With such an additional input, we perform an ε -decomposition. Given a system Σ , two sets R_x and R_y respectively subsets of \mathbb{R}^n and \mathbb{R}^m , a positive integer k , and a vector of errors $\varepsilon_x = (\varepsilon_x^1, \dots, \varepsilon_x^k)$, application of the ε -decomposition returns a set Δ of the form $\{V_i, Pat_i\}_{i \in I}$, where I is a finite set of indexes, every V_i is a subset of R_x , and every Pat_i is a k -pattern such that:

- (a') $\bigcup_{i \in I} V_i = R_x$,
- (b') for all $i \in I$: $Post_{Pat_i}(V_i) \subseteq R_x - \varepsilon_x^{|Pat_i|}$,
- (c') for all $i \in I$: $Post_{Pat_i,C}(V_i) \subseteq R_y$.

Note that condition (b') is a strengthening of condition (b) in subsection 5.2.

Accordingly, line 6 of Algorithm 5 becomes in Algorithm 7:

6 if $Post_{Pat}(W) \subseteq R_x - \varepsilon_x^i$ **and** $Post_{Pat,C}(W) \subseteq R_y$ **then**

The new algorithms enable to guarantee that the projection of the full-order system state $\pi_R x$ always stays in \hat{R}_x , we can thus perform the online control as follows:

Since $Post_{Pat_{i_0}}(\hat{V}_{i_0}) \subseteq \hat{R}_x - \varepsilon_x^{|Pat_{i_0}|}$ and $\pi_R x_0 \in \hat{V}_{i_0}$, we have $Post_{Pat_{i_0}}(\pi_R x_0) \in \hat{R}_x - \varepsilon_x^{|Pat_{i_0}|}$; thus $\pi_R x_1 = \pi_R \mathbf{x}(|Pat_{i_0}| \tau, x_0, Pat_{i_0})$ belongs to \hat{R}_x , because $\varepsilon_x^{|Pat_{i_0}|}$ is a bound of the maximal distance between the trajectories $\hat{\mathbf{x}}(|Pat_{i_0}| \tau, \pi_R x_0, Pat_{i_0})$ and $\pi_R \mathbf{x}(|Pat_{i_0}| \tau, x_0, Pat_{i_0})$;

since $\pi_R x_1$ belongs to \hat{R}_x , it belongs to V_{i_1} for some $i_1 \in I$; we can thus compute the input pattern Pat_{i_1} , and therefore, we can reapply the procedure and compute an input pattern sequence $Pat_{i_0}, Pat_{i_1}, \dots$. As for the output, the yielded points $y_1 = \mathbf{y}(|Pat_{i_0}| \tau, x_0, Pat_{i_0})$, $y_2 = \mathbf{y}(|Pat_{i_1}| \tau, x_1, Pat_{i_1})$, \dots belong respectively to the sets $R_y + \varepsilon_y^{|Pat_{i_0}|}, R_y + \varepsilon_y^{|Pat_{i_1}|}, \dots$

The main advantage of such an online control is that the estimated errors $\varepsilon_y^{|Pat_{i_0}|}, \varepsilon_y^{|Pat_{i_1}|}, \dots$ are dynamically computed, and are smaller than the static bound ε_y^∞ used in the offline control. The price to be paid is the strengthening of condition (b'). In the best case, i.e. if the errors are low and the system is very contractive, this can result in the same decomposition and computation time as in the offline

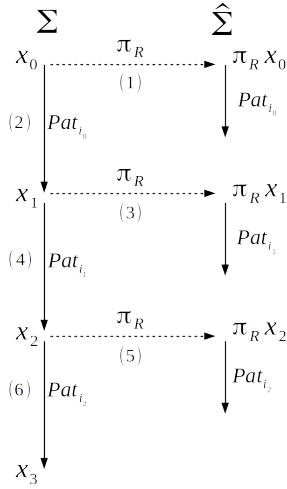


Figure 5.2: Diagram of the online procedure for a simulation of length 3.

procedure. But if the system is not contractive enough or if the errors are too large, this can lead to a more complicated decomposition, and thus higher computation times, and in the worst case, no successful decomposition at all.

5.5 Numerical results

5.5.1 Thermal problem on a metal plate

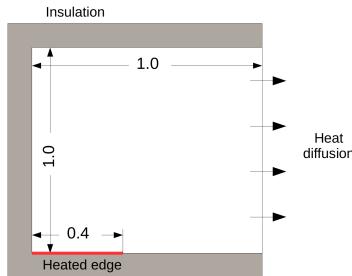


Figure 5.3: Geometry of the square plate.

We consider here the problem of controlling the central node temperature of a square metal plate, discretized by finite elements; this example is taken from [86]. The square plate is subject to the heat equation: $\frac{\partial T}{\partial t}(x, t) - \alpha \Delta T(x, t) = 0$. After discretization, the system is written under its state-space representation (5.3). The plate is insulated along three edges, while the right edge is open. The left half of the bottom edge is connected to a heat source. The exterior temperature is set to 0°C, the temperature of the heat source is either 0°C (mode 0) or 1°C (mode 1). The heat transfers with the exterior and the heat source are modeled by a convective transfer. The full-order system state corresponds to the nodal temperatures. The output is the temperature of the central node. The system is reduced from $n = 897$

Algorithm 6 Decomposition_Dyn($W, R_x, R_y, D, K, \varepsilon_x$)

Input: A box W , a box R_x , a box R_y , a length K of pattern, a vector of errors ε_x , a degree D of bisection

Output: $\langle \{(V_i, Pat_i)\}_i, True \rangle$ with $\bigcup_i V_i = W$, $\bigcup_i Post_{Pat_i}(V_i) \subseteq R_x$ and $\bigcup_i Post_{Pat_i,C}(V_i) \subseteq R_y$, or $\langle _, False \rangle$

$$(Pat, b) := \text{Find_Pattern_Dyn}(W, R_x, R_y, K, \varepsilon_x)$$

```

if  $b = True$  then
    return  $\langle \{(W, Pat)\}, True \rangle$ 
else
    if  $D = 0$  then
        return  $\langle \_, False \rangle$ 
    else
        Divide equally  $W$  into  $(W_1, \dots, W_{2^n})$ 
        for  $i = 1 \dots 2^n$  do
             $(\Delta_i, b_i) := \text{Decomposition\_Dyn}(W_i, R_x, R_y, K, \varepsilon_x, D - 1)$ 
        end for
        return  $(\bigcup_{i=1 \dots 2^n} \Delta_i, \bigwedge_{i=1 \dots 2^n} b_i)$ 
    end if
end if

```

to $n_r = 2$ (Figure 5.5) and $n_r = 3$ (Figure 5.6). The interest set is $R_x = [0, 0.15]$ ⁸⁹⁷ and the objective set $R_y = [0.06, 0.09]$. The sampling time is set to $\tau = 8$ s. The geometry of the system is given in Figure 5.3. The decomposition obtained with the offline procedure is given in Figure 5.4.

The decompositions and simulations have been performed with MINIMATOR (an Octave code available at https://bitbucket.org/alecoent/minimator_red) on a 2.80 GHz Intel Core i7-4810MQ CPU with 8 GB of memory. The decompositions were obtained in 5 seconds for the case $n_r = 2$ and in 2 minutes for the case $n_r = 3$.

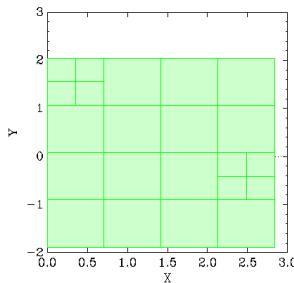


Figure 5.4: Decomposition of $\hat{R}_x = \pi_R R_x$ in the plane (\hat{x}_1, \hat{x}_2) (for $n_r = 2$) with the offline procedure.

Simulations of the offline and online methods are given in Figures 5.5 and 5.6. We notice in Figure 5.5 that the trajectory y (resp. y_r) exceeds the objective set R_y (resp. $R_y + \varepsilon_y^{|Pat_i|}$) during the application of the second pattern, yet the markers

Algorithm 7 Find_Pattern_Dyn($W, R_x, R_y, K, \varepsilon_x$)

Input: A box W , a box R_x , a box R_y , a length K of pattern, a vector of errors ε_x

Output: $\langle Pat, True \rangle$ with $Post_{Pat}(W) \subseteq R_x, Post_{Pat,C}(W) \subseteq R_y$ and $Unf_{Pat}(W) \subseteq S$, or $\langle _, False \rangle$ when no pattern maps W into R_x and CW into R_y

for $i = 1 \dots K$ **do**

$\Pi :=$ set of patterns of length i

while Π is non empty **do**

Select Pat in Π

$\Pi := \Pi \setminus \{Pat\}$

if $Post_{Pat}(W) \subseteq R_x - \varepsilon_x^i$ and $Post_{Pat,C}(W) \subseteq R_y$ **then**

return $\langle Pat, True \rangle$

end if

end while

end for

return $\langle _, False \rangle$

corresponding to the end of input patterns do belong to objective sets. Comparing the cases $n_r = 2$ and $n_r = 3$, we finally observe that a less reduced model causes lower error bounds, and thus a more precise control, at the expense of a higher computation time.

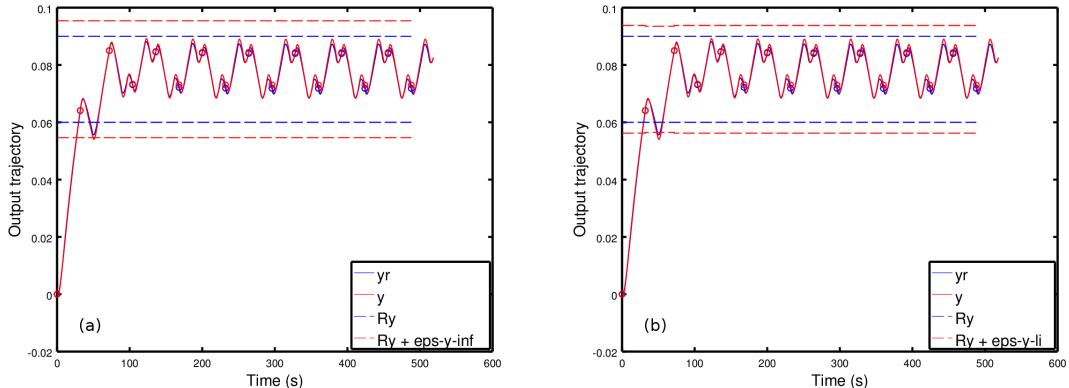


Figure 5.5: For $n_r = 2$, simulation of $y(t) = Cx(t)$ and $y_r(t) = \hat{C}\hat{x}(t)$ from the initial condition $x_0 = (0)^{897}$. (a): guaranteed offline control; (b): guaranteed online control.

5.5.2 Vibrating beam

In this case study, which comes from a practical work designed by Fabien Formosa [63], we apply our method to vibration control of a cantilever beam. The objective

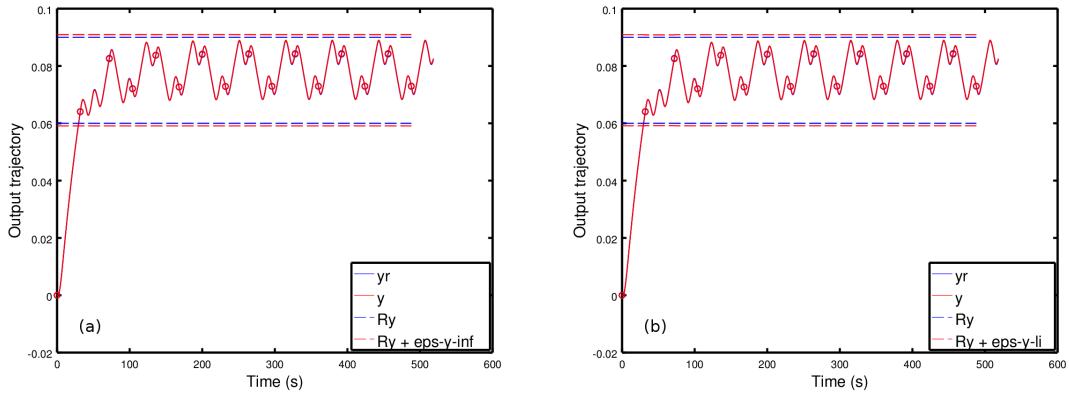


Figure 5.6: For $n_r = 3$, simulation of $y(t) = Cx(t)$ and $y_r(t) = \hat{C}\hat{x}(t)$ from the initial condition $x_0 = (0)^{897}$. (a): guaranteed offline control; (b): guaranteed online control.

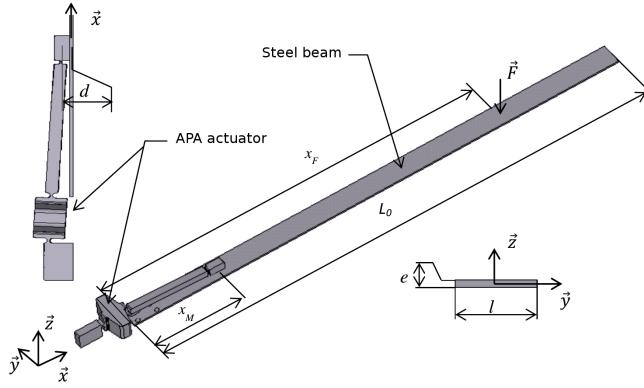


Figure 5.7: Scheme of the vibrating beam.

is to keep the tip displacement of the beam as close as possible to zero. To stabilize the beam, a piezoelectric patch applies a torque with the mechanism schemed in Figure 5.7 at a distance x_M from the blocked side of the beam. The model retained is a finite element model with classical beam elements. The beam equation is the following:

$$m\ddot{w}(x, t) + EI \frac{\partial^4 w(x, t)}{\partial x^4} = \frac{\partial M_u}{\partial x} \delta(x - x_M) \quad (5.12)$$

The torque M_u is chosen with the control variable u . By applying the right torque at the right time, we hope to stabilize the beam. In its finite element writing, the system is:

$$M\ddot{W} + KW = F_u \quad (5.13)$$

Using a modal decomposition

$$W(x, t) = \sum_{i \leq n_{modes}} a_i(t) \varphi_i(x),$$

we can write a reduced system of the form:

$$M_r \ddot{a}_i(t) + 2\zeta_i \dot{a}_i(t) + K_r a_i(t) = F_{r,u}. \quad (5.14)$$

Note that a modal damping is added in this step, it permits to have a realistic behaviour of the beam since it is subject to loss of energy. By rearranging the terms of equation (5.14) into a first order ODE, we can write the system under a state-space representation:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \quad (5.15)$$

where the output y is the tip displacement of the beam. Henceforth, the state variable contains the variables a_i and \dot{a}_i . The dimension of the state-space is thus twice the number of retained modes. In this way, the system can be treated with the method developed here, applying a balanced truncation to the system (5.15) and building a reduced-order control.

Note that the intermediate model order reduction by modal decomposition cannot actually be avoided, because the direct rearrangement of system (5.13) into its state-space representation leads to a matrix A possessing some positive eigenvalues (instead of only negative ones), and the calculation of balancing transformations is then much more complicated, or even impossible.

The finite element model is composed of 60 elements (thus 120 degrees of freedom to take the rotation into account), we retain 20 modes for the modal decomposition, and the system is reduced to $n_r = 4$. Nine control modes are chosen to control the beam, including the mode corresponding to a null torque. Two simulations for different initial conditions and objective sets are given in Figure 5.8. In the first one, several modes are initially excited, whereas only the first mode is excited in the second one. In both cases, the online procedure is applied, and we manage to stabilize the tip displacement relatively fast. The output of the full-order system is stabilized in $R_y + \varepsilon_y^{|Pat_i|}$ with $\varepsilon_y^{|Pat_i|} \simeq 0.2$. The errors $\varepsilon_y^{|Pat_i|}$ can seem quite high compared to the tip displacement, this comes from the hyperbolic nature of the equations which rule this example. However, in a practical point of view, this is clear that the reduced-order output fits well the behavior of the full-order system.

5.5.3 Vibrating aircraft panel

In order to verify the handling of higher dimensional systems, we apply our method to the vibration control of an aircraft panel. This example, taken from [95], consists in stabilizing the panel as close as possible to the equilibrium, which corresponds to a null displacement inside the whole panel. In this purpose, seven piezoelectric patches are glued on the panel, one is used for exciting the panel (patch 1 of Figure 5.9), one is used as a sensor to evaluate the performance of the control (patch 2), one is used for the observation of modal states (patch 6), and three are used for vibration control (patches 3 to 5), the last patch being used to validate the reconstruction (patch 7). For the numerical simulations, we choose the measurements of the sensor patch as the output of the system.

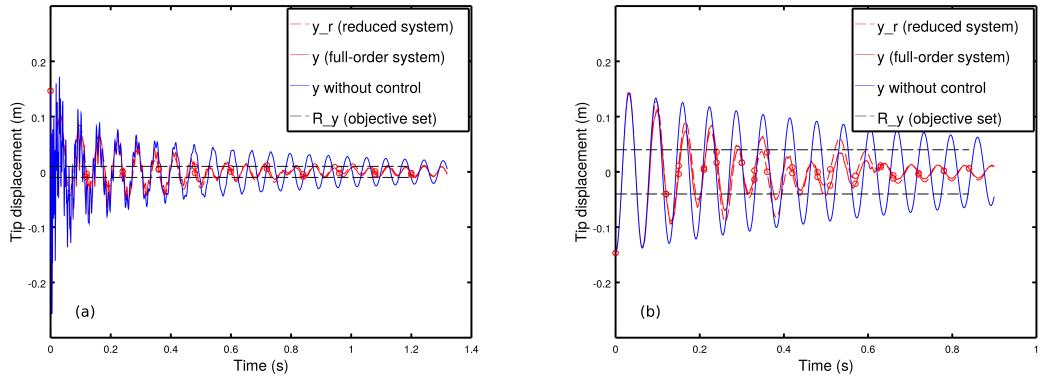


Figure 5.8: Simulations of vibration control of the cantilever beam for two different initial conditions and objective boxes. (a): several modes excited; (b): first mode excited.

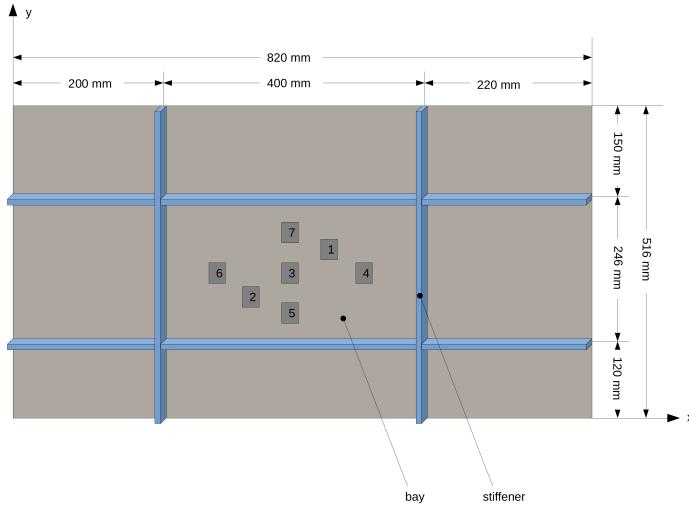


Figure 5.9: Scheme of the vibrating aircraft panel.

Just as the cantilever beam, we use a finite element model reduced by modal decomposition then balanced truncation. The system is written exactly in the same way, but with shell elements, and thus six degrees of freedom per node. The finite shell element model consists of 57000 degrees of freedom. We retain 50 modes for the modal decomposition, and the model is reduced down to $n_r = 5$ by balanced truncation. Seven control modes are used for vibration control, it corresponds to a null voltage applied on all the control patches, a positive constant voltage applied on each control patch (one patch is subject to a voltage at a time), and a negative constant voltage applied on each control patch. The reader is referred to [95] for more information on the exact functioning of the piezoelectric patches used in this case study, and see for example [83, 139] for more general information on piezoelectric patches and their use for structural damping. With the same hardware configuration as in the previous example, the computation of a decomposition took nearly a week. A simulation of the online procedure is given in Figure 5.10 and 5.11.

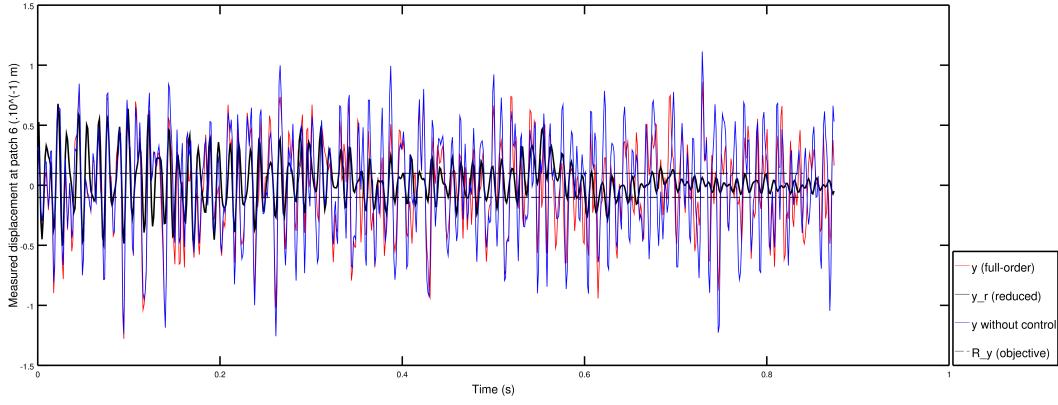


Figure 5.10: Simulation of vibration control of the aircraft panel.

We observe that the response of the controlled full-order system is better than the non-controlled one, the main peaks observed in the non-controlled response are avoided. Nevertheless, the stabilization is not as efficient as one may expect. One can see that the reduced-order system is however well stabilized. This points out that the model reduction does not catch, in this case, all the information needed for control purposes. While we are currently investigating new model reduction techniques, adapted to hyperbolic and non-linear systems, we also think that in practice, the stabilization would be better because of the smoothness appearing in the applied torques in a real application.

5.6 Extension to output feedback control

So far, we designed reduced state-dependent controllers for switched control systems, permitting to stabilize the output of the system in a given objective set R_y . During a real online use, one is only supposed to know a part of the state of the system, such as measurements of sensors. We now want to take these partial measurements into account, by adding an intermediate step in the online use, namely, observation. We suppose that only the output of the system is known online. In the next sub-section, we introduce the principle of observation and give some preliminary results justifying the use of observers for switched control systems, allowing us to adapt our algorithms to the use of observers. We then present some numerical results of the use of observers with model order reduction. The whole approach with model order reduction is schemed in Figure 5.12, but as we do not have any proof for the efficiency of the use of observers with model order reduction, we only provide some numerical simulations. We are currently working on the establishment an error bound taking into account the projection error and the observation error, that will permit to construct a guaranteed reduced observer based control.

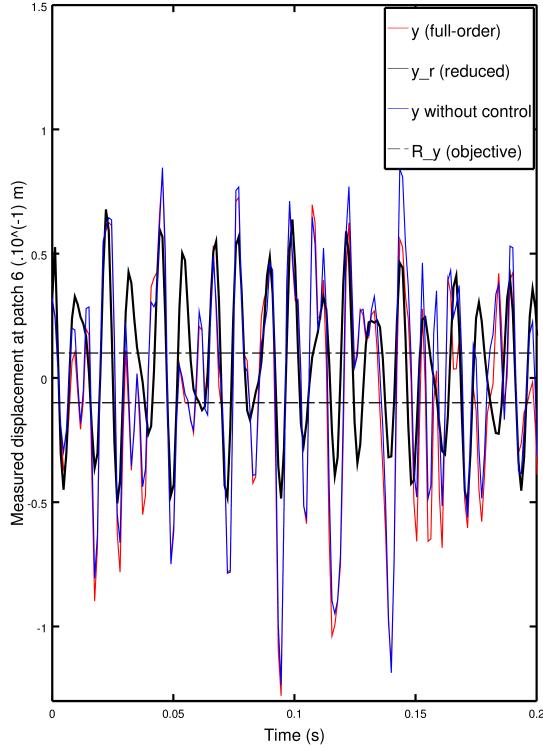


Figure 5.11: Enlargement of Figure 5.11 on the time interval $[0, 0.2]$.

5.6.1 Partial observation

Having defined the state-space bisection algorithm for switched control systems with output, we now add the constraint that the system is partially observed. The objective is to design an *output feedback* controller using the state-space bisection algorithm introduced above.

We recall that the switched system Σ is written under the following form:

$$\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t). \end{cases}$$

We suppose that during an online use, one is only supposed to know $y(t)$ (we suppose that y can be measured in real time, that is at every time t). If just this partial information of the state is known, we cannot directly apply our state-dependent controller synthesis method. An intermediate step must be introduced: the reconstruction of the state. The reconstruction is made with the help of an observer: it is an intermediate system that provides an estimate of the state of the system Σ from the measurements of the output y and the input u of the system Σ . In fact, this means that we want to design an output feedback law for the system Σ with the help of an observer. In this chapter, we retain the Luenberger observer [3, 4, 176]

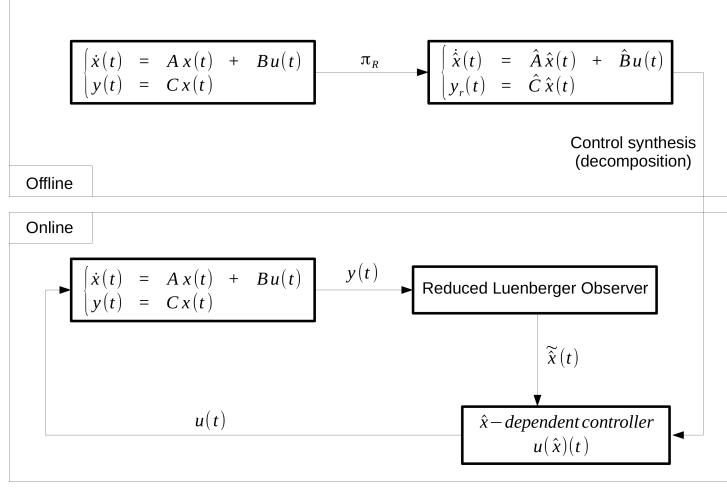


Figure 5.12: Principle of the output feedback control

to reconstruct the state of Σ , it is subject to the following equation:

$$\dot{\tilde{x}} = A\tilde{x} - L(u)(C\tilde{x} - y) + Bu, \quad L(u) \in \mathbb{R}^{n \times m} \quad (5.16)$$

Obviously, the observer does not reconstruct exactly the state x of the system Σ , we thus introduce the reconstruction error $\eta(t) = \|x(t) - \tilde{x}(t)\|$. Our goal is to control the system Σ with this estimate \tilde{x} : we apply a law $u(\tilde{x})$. One can note that the method relies on the convergence of the observer \tilde{x} to the state x , this aspect is developed in the following section.

The entries of the control problem we retain are then the following:

- an interest set $R_x \subset \mathbb{R}^n$,
- an objective set $R_y \subset \mathbb{R}^m$,
- an initial, a priori known, reconstruction error η_0 .

With the method given below, the outputs of the problem are the following:

- a decomposition of R_x w.r.t. η_0 and the dynamics of Σ ,
- a procedure to choose u knowing \tilde{x} ,
- and the guarantee that, for any pattern Pat , if $x_0 \in R_x$ and $\eta(0) \leq \eta_0$, then $\mathbf{x}(|Pat|\tau, x_0, Pat) \in R_x$ and $\mathbf{y}(|Pat|\tau, x_0, Pat) \in R_y$.

Let us now introduce some hypotheses and important results to ensure the efficiency of the method.

5.6.2 Convergence of the observer

The properties of the Luenberger observer depend on the choice of the matrices $L(u)$ appearing in (5.16). A crucial assumption in what follows is that it is possible to choose $L(\cdot)$ in such a way that the modes of the Luenberger observer share a common non-strict quadratic Lyapunov functions, i.e., there exists a positive definite matrix

P such that:

$$\forall u, \quad P(A + L(u)C) + (A + L(u)C)^\top P \leq 0. \quad (5.17)$$

The dynamics of the original switched system and of the Luenberger switch observer can be grouped in the augmented system

$$\begin{pmatrix} \dot{\tilde{x}} \\ \dot{x} \end{pmatrix} = \begin{pmatrix} A - L(u)C & L(u)C \\ 0 & A \end{pmatrix} \begin{pmatrix} \tilde{x} \\ x \end{pmatrix} + \begin{pmatrix} Bu \\ Bu \end{pmatrix}.$$

Define $e(t) = x(t) - \tilde{x}(t)$ and $\eta(t) = e(t)^T Pe(t)$. By definition $e(\cdot)$ satisfies

$$\dot{e} = (A - L(u)C)e \quad (5.18)$$

and assumption (5.17) implies that η is non-increasing along all trajectories. The patterns in $u(\cdot)$ will be chosen in order to guarantee that not only η decreases, but actually converges to zero.

An assumption which may be motivated by the technical constraints of the system under consideration is the existence of a *dwell-time*, that is, a positive constant τ such that two subsequent discontinuities of $u(\cdot)$ have a distance of at least τ (recall that $u(\cdot)$ is assumed to be piecewise constant). The dwell-time condition not only reflects technological constraints, but is also useful in the asymptotic analysis of the switched system (5.3). The basic result that we will use is a simplified version of [163, Theorem II.5], which states that under the dwell-time hypothesis, and by choosing properly the patterns, one can manage to make $\eta(t)$ converge to 0. (For further asymptotic results of linear switched systems with a common non-strict quadratic Lyapunov function, see [24, 155].)

The strategy suggested by the previous theorem is the following:

- identify $u_{*,1}, \dots, u_{*,m}$ such that

$$\cap_{j=1}^m \text{Ker}(A - L(u_{*,j})C) = (0);$$

- impose that each pattern takes all values $u_{*,1}, \dots, u_{*,m}$.

Under these constraints the solution e of (5.18) is guaranteed to converge to the origin (monotonically with respect to the norm induced by the positive matrix P).

In the case of the metal plate we will see that it is sufficient to take $m = 2$ and that the constraint that each pattern passes through the two values $u_{*,1}, u_{*,2}$ is not a heavy obstacle in the implementation of the proposed algorithm. As a result, we will obtain a strategy $u(\tilde{x})$ that, under the assumption that the initial state $x(0)$ and the initial estimation $\tilde{x}(0)$ are in R_x and satisfy $\eta(0) < \eta_0$, the trajectory $\mathbf{x}(t, x(0), u)$ and the estimated trajectory, denoted by $\tilde{\mathbf{x}}(t, \tilde{x}(0), u)$, are such that the evaluation of $\mathbf{x}(\cdot)$ after each pattern is again in R_x and $\mathbf{x}(t, x(0), u) - \tilde{\mathbf{x}}(t, x(0), u) \rightarrow 0$ as $t \rightarrow +\infty$.

5.6.3 Observer based decomposition

We present here the adaptations of the algorithms taking the observation into account. The *observer based decomposition* algorithm takes η_0 as a new input. Given a system Σ , two sets $R_x \subset \mathbb{R}^n$ and $R_y \subset \mathbb{R}^m$, a positive integer k , and an initial reconstruction error η_0 , a successful observer based decomposition returns a set $\tilde{\Delta}$ of the form $\{V_i, Pat_i\}_{i \in I}$, where I is a finite set of indices, every V_i is a subset of R_x , and every Pat_i is a k -pattern such that:

- (a) $\bigcup_{i \in I} V_i = R_x$,
- (b) for all $i \in I$: $Post_{Pat_i}(V_i + \eta_0) \subseteq R_x - \eta_0$,
- (c) for all $i \in I$: $Post_{Pat_i,C}(V_i + \eta_0) \subseteq R_y$.

Such a decomposition allows to perform an output feedback control on Σ as stated in the following. The algorithm relies on two functions given in Algorithms 8 and 9. If a successful observer based decomposition is obtained, it naturally induces an estimate-dependent control, which we denote by $\mathbf{u}_{\tilde{\Delta}}$. By looking for patterns mapping $R_x + \eta_0$ into R_x , we guarantee that $\mathbf{x}(t, x, u)$ is stabilized in R_x . Indeed, if $x(0)$ is the initial state, and $\tilde{x}(0)$ the initial estimation (supposed belonging to R_x), we know that $\tilde{x}(0)$ belongs to V_{i_0} for some $i_0 \in I$, and that $x(0)$ belongs to $V_{i_0} + \eta_0$, so the application of the pattern Pat_{i_0} yields $\mathbf{x}(|Pat_{i_0}| \tau, x(0), Pat_{i_0}) \in R_x - \eta_0$ (because $Post_{Pat_{i_0}}(V_{i_0} + \eta_0) \subseteq R_x - \eta_0$) and $\tilde{\mathbf{x}}(|Pat_{i_0}| \tau, \tilde{x}(0), Pat_{i_0}) \in R_x$ because

$$\begin{aligned} & \|\mathbf{x}(|Pat_{i_0}| \tau, x(0), Pat_{i_0}) - \tilde{\mathbf{x}}(|Pat_{i_0}| \tau, \tilde{x}(0), Pat_{i_0})\| \\ & < \eta_0. \end{aligned}$$

Note that we plan to improve these algorithms by taking the decrease of $\eta(t)$ into account, so that the decomposition is less restrictive when $\eta(t)$ is small.

5.6.4 Reduced output feedback control

Algorithms 8 and 9 allow to synthesize guaranteed output feedback controllers for switched control systems without model order reduction. However, the use of model order reduction and observation for the thermal problem of section 5.5.1 is indeed possible, this is partly enabled thanks to the elliptic nature and highly contractive behavior of the system.

The online simulations are performed just as sated in Figure 5.12. From the full-order system Σ , we build a reduced-order system $\hat{\Sigma}$ by balanced truncation. An ε -decomposition is then performed on $\hat{\Sigma}$, yielding a \hat{x} -dependent controller (the decomposition was obtained in about two minutes). The control $u(\tilde{x})$ is then computed online with the reconstructed variable \tilde{x} , which dynamics is the following:

$$\dot{\tilde{x}} = \hat{A}\tilde{x} - L(u)(\hat{C}\tilde{x} - Cx) + \hat{B}u, \quad L(u) \in \mathbb{R}^{n_r \times m} \quad (5.19)$$

As the ε -decomposition is already quite restrictive (i.e. the error bound overestimates the real projection error) and because the Luenberger observer converges

Algorithm 8 Decomposition_Obs($W, R_x, R_y, D, K, \eta_0$)

Input: A box W , a box R_x , a box R_y , a degree D of bisection, a length K of input pattern, an initial reconstruction error η_0

Output: $\langle\{(V_i, Pat_i)\}_i, True\rangle$ with $\bigcup_i V_i = W$, $\bigcup_i Post_{Pat_i}(V_i + \eta_0) \subseteq R_x$ and $\bigcup_i Post_{Pat_i,C}(V_i + \eta_0) \subseteq R_y$, or $\langle_, False\rangle$

$(Pat, b) := Find_Pattern(W, R_x, R_y, K, \eta_0)$

if $b = True$ **then**

return $\langle\{(W, Pat)\}, True\rangle$

else

if $D = 0$ **then**

return $\langle_, False\rangle$

else

Divide equally W into (W_1, \dots, W_{2^n})

for $i = 1 \dots 2^n$ **do**

$(\Delta_i, b_i) := Decomposition_Obs(W_i, R_x, R_y, D - 1, K, \eta_0)$

end for

return $(\bigcup_{i=1 \dots 2^n} \Delta_i, \bigwedge_{i=1 \dots 2^n} b_i)$

end if

end if

Algorithm 9 Find_Pattern_Obs(W, R_x, R_y, K, η_0)

Input: A box W , a box R_x , a box R_y , a length K of input pattern, an initial reconstruction error η_0

Output: $\langle Pat, True\rangle$ with $Post_{Pat}(W + \eta_0) \subseteq R_x, Post_{Pat,C}(W + \eta_0) \subseteq R_y$, or $\langle_, False\rangle$ when no input pattern maps $W + \eta_0$ into R_x

for $i = 1 \dots K$ **do**

$\Pi :=$ set of input patterns of length i

while Π is non empty **do**

Select Pat in Π

$\Pi := \Pi \setminus \{Pat\}$

if $Post_{Pat}(W + \eta_0) \subseteq R_x - \eta_0$ and $Post_{Pat,C}(W + \eta_0) \subseteq R_y$ **then**

return $\langle Pat, True\rangle$

end if

end while

end for

return $\langle_, False\rangle$

fast, we observe that the induced control already works, even if we do not have any justification of the efficiency yet. The proof should be established by evaluating, for any pattern Pat , a bound of the following error:

$$\|\pi_R \mathbf{x}(|Pat| \tau, x(0), Pat) - \tilde{\mathbf{x}}(|Pat| \tau, \tilde{x}(0), Pat)\| \quad (5.20)$$

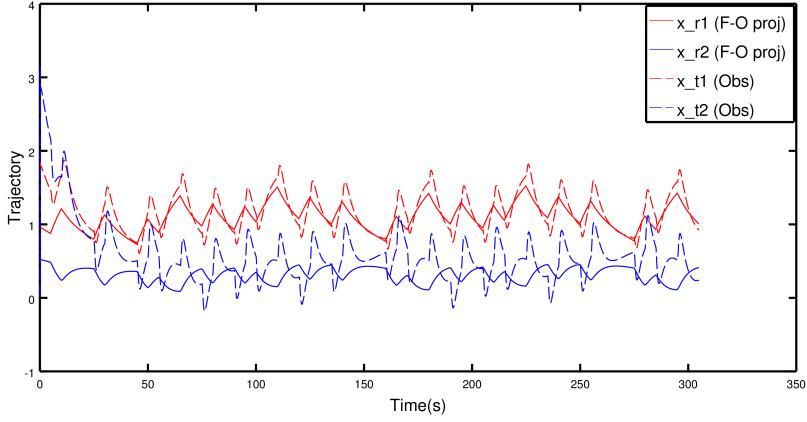


Figure 5.13: Simulation of the thermal problem with observation: projected variables. x_{r1} and x_{r2} are the two variables $\pi_R x$ plotted within time (plain lines), it corresponds to the projection of the full-order system state. x_{t1} and x_{t2} are the two variables \tilde{x} plotted within time (dotted lines), it corresponds to the state of the reduced observer.

In the simulations Figures 5.13 and 5.14, the full-order system is of order $n = 897$, the reduced order system of order $n_r = 2$. The full-order system is initialized with a uniform temperature field of $x(0) = 0.06^n$. The reduced observer is initialized at $\tilde{x}(0) = 0^2$. The two projected variables $\pi_R x$ cannot be reconstructed exactly because of (at least) the projection error, but the output is still very well reconstructed. Both the observer and the full-order outputs are sent in the objective set R_y , which means that we should manage to control a thermal problem just with the information obtained with few sensors.

5.7 Final remarks

Two methods have been proposed to synthesize controllers for switched control systems using model order reduction and the state-space bisection procedure. An offline and an online use are enabled, both uses are efficient but they present different advantages. The offline method allows to obtain the same behavior as the reduced-order model, but the associated bound is more pessimistic, and the controller has to be computed before the use of the real system. The online method leads to less pessimistic bounds but implies a behavior slightly different from the reduced-order model, and the limit cycles may be different from those computed on the reduced system. The behavior of the full-order system is thus less known, but its use can be performed in real time.

A first step to the online reconstruction of the state of the system has been done with the help of Luenberger observers. Numerical simulations seem to show a good behavior with reconstruction and model reduction but the efficiency must still be

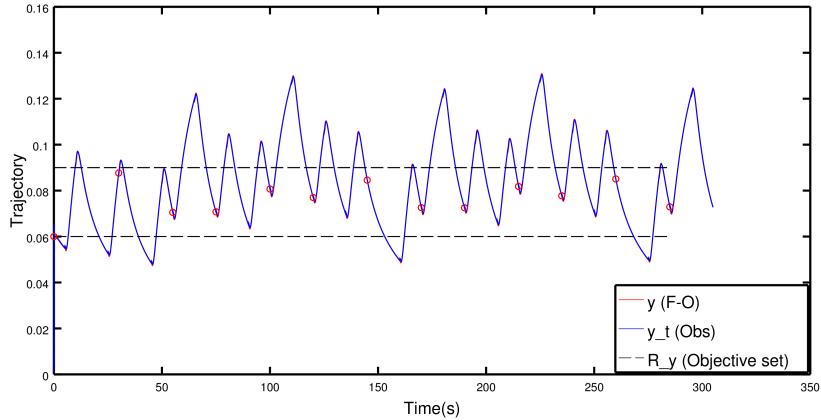


Figure 5.14: Simulation of the thermal problem with observation: output variables. The output of the full-order system (plain red) coincides with the output reconstructed by the observer (plain blue), both are sent in the objective set at the end of patterns (red circles).

proved. The use of Kalman filters is however not dismissed.

We are still investigating new model order reductions, more adapted to hyperbolic systems, and with the aim of controlling non linear PDEs. A recent trail which we also want to develop is the dimensionality reduction [82, 156, 160]. Less restrictive than model order reduction, it should permit to use a fine solver and post-processing techniques to use bisection on a reduced space more representative of the system behavior.

Chapter 6

Control of PDEs

Terminology

C_Ω	Poincaré's constant (depending on Ω)
f	Source heat term of the heat equation
$g = -\frac{\partial u_q}{\partial t}(:, \boldsymbol{\xi}(t))$	Source term of the equation in ψ
K	Reduced-order truncation rank (low-order dimension)
$\kappa(\cdot)$	(space-varying) conductivity coefficient
κ_m	Minimal conductivity coefficient
K	Truncation rank for the reduced-order space
L	Length of the spatial interval
M	Number of control modes
$\psi = \psi(., t)$	such that $u(., t) = u^\infty(.) + u_q(., t) + \psi(., t)$
$\tilde{\psi}$	reduced-order model for ψ
$r_\xi(v)$	Residual of the approximate solution $\tilde{\psi}$ against v
$\Omega = (0, L)$	Spatial domain
ρ	Tolerance radius for the distance between u and u^∞
R_ξ	Recurrence set for the $\boldsymbol{\xi}$ variable
Σ^τ	Space of admissible switch control sequences
U	Set of switched modes
τ	Switching sampling time
t	Time variable
$u = u(x, t)$	Solution of the controlled heat problem
$\tilde{u} = \tilde{u}(x, t)$	Reduced-order solution of the heat problem
$u^\infty(.)$	"Objective" heat function
$u_q(., t)$	Solution of the quasistatic heat problem at time t
$V = H_0^1(\Omega)$	Sobolev space
x	Space variable
$\boldsymbol{\xi}(t) = (\xi_1(t), \xi_2(t))^T$	Vector of boundary control values
ξ_1^∞	$= u^\infty(0)$
ξ_2^∞	$= u^\infty(L)$
$\boldsymbol{\xi}^\infty$	$= (\xi_1^\infty, \xi_2^\infty)^T$
$W^K = \text{span}(\varphi^1, \dots, \varphi^K)$	Reduced-order linear space, $W^K \subset V$

6.1 Introduction

In the previous chapter, we managed to synthesize reduced order controllers for high dimensional ODEs, obtained from the discretization of PDEs. We now want to use this kind of techniques for results on the PDE problem. A first possibility would have been to use error estimations of the discretization techniques employed, such as the ZZ estimators [178] for finite element methods. However, such estimators are quite pessimistic and imply large errors, preventing us from synthesizing

guaranteed controllers in practice. In this chapter, we aim at keeping a PDE formulation undiscretized, and by properly transforming the problem, synthesizing low order controllers. We first provide some of the developments made to obtain such results, and show the underlying difficulties. We first tried to use simple projection methods, such as spectral methods, associated to the Empirical Interpolation Method (EIM) [129]. The EIM is a recent algorithm which provides the best sets of points for Lagrangian interpolation, which permits to efficiently represent complex functions with few generating functions. It has been derived for many efficient reduced basis methods. The EIM was one of our first choices for guaranteed control of PDEs since it comes with an L^∞ error bound, and it seemed to be a natural way of obtaining continuous equivalents of Chapter 5. It revealed more complicated than expected to derive an L^∞ guaranteed control, but we hope that these results might be of interest for future works. After a long time struggling on L^∞ bounds, we finally came to a change of topology for our reduced models, in order to develop L^2 guaranteed controls. As a matter of fact, L^2 error bounds are actually much more classical in the field of structural mechanics, particularly when it comes to reduced order modeling. We thus present a second approach, aimed at synthesizing L^2 guaranteed controls. The goal is now to use Galerkin methods for model order reduction, which is much more general than the balanced truncation or spectral methods, and allows to adapt the reduction technique to PDE problem. A second objective is to get an L^2 error estimation directly for the PDE problem, and not a discretized version. In the following, we present our approaches on a given coupled ODE-PDE problem, for which the ODE is controlled.

6.2 Setting of the problem

Let $L > 0$, let $\Omega = (0, L)$ be the domain of definition of the PDE. Let $\kappa \in L^\infty(0, L)$, and suppose there exist two constants κ_m and κ_M , $0 < \kappa_m \leq \kappa_M$ such that

$$\kappa_m \leq \kappa(x) \leq \kappa_M \text{ for a.e. } x \text{ in } [0, L].$$

The space of admissible switch control sequences is

$$\Sigma^\tau = \left\{ \sigma : [0, +\infty[\rightarrow \{1, \dots, M\}, \sigma|_{[q\tau, (q+1)\tau[}(t) \in U \ \forall q \in \mathbb{N} \right\}. \quad (6.1)$$

In this chapter, we consider the one-dimensional boundary switched control heat problem: find a piecewise constant sequence $\sigma(\cdot) \in \Sigma^\tau$, such that the vector-valued state $\xi(\cdot) \in [\mathcal{C}_b^0(0, \infty)]^2$ and the function $u \in L^2(0, \infty; H^1(\Omega))$ solutions of the prob-

lem

$$\frac{d\boldsymbol{\xi}}{dt} = A_\sigma \boldsymbol{\xi} + \mathbf{b}_\sigma, \quad t > 0, \quad (6.2)$$

$$\boldsymbol{\xi}(0) = \boldsymbol{\xi}^0, \quad (6.3)$$

$$\frac{\partial u}{\partial t} - \nabla \cdot (\kappa(\cdot) \nabla u) = f \quad \text{in } \Omega \times (0, +\infty), \quad (6.4)$$

$$u(0, t) = \xi_1(t), \quad \text{for all } t > 0, \quad (6.5)$$

$$u(L, t) = \xi_2(t), \quad \text{for all } t > 0, \quad (6.6)$$

$$u(., t = 0) = u^0 \quad (6.7)$$

verify, for any initial conditions $\boldsymbol{\xi}_0$ and u_0 , the stability constraints

$$\begin{cases} \boldsymbol{\xi}(t) \in R_\xi & \text{for all } t > 0, \\ \|u(., t) - u^\infty(\cdot)\|_{L^2(\Omega)} \leq \rho & \text{for all } t > 0. \end{cases} \quad (6.8)$$

Thus the expected recurrence set for the global state $(\boldsymbol{\xi}(t), u(., t))$ is the product set $R_\xi \times B(u^\infty, \rho; L^2(\Omega)) \subset \mathbb{R}^2 \times L^2(\Omega)$. The sequence $\sigma(\cdot)$ will depend on the state of the system itself in order to enforce stability in the product recurrence set. The control problem is formalized as follows:

Problem 4 (ODE-PDE stability control problem). *Let us consider the equation system (6.2)-(6.7). Given a set R_ξ , a tolerance ρ and an objective state $u^\infty(\cdot)$, find a rule $\sigma((\boldsymbol{\xi}, u)) \in \Sigma^\tau$ such that, for all $t > 0$ and for all $(\boldsymbol{\xi}(0), v(x, 0)) \in R_\xi \times B(u^\infty, \rho; L^2(\Omega))$, we have $(\boldsymbol{\xi}(t), u(., t)) \in R_\xi \times B(u^\infty, \rho; L^2(\Omega))$.*

We can also consider the reachability problem:

Problem 5 (ODE-PDE reachability control problem). *Let us consider the equation system (6.2)-(6.7). Given two sets R_ξ and R'_ξ with $R'_\xi \subset R_\xi$, two tolerances ρ and ρ' with $\rho' < \rho$, and an objective state $u^\infty(\cdot)$, find a rule $\sigma((\boldsymbol{\xi}, u)) \in \Sigma^\tau$ such that, for all $(\boldsymbol{\xi}(0), v(x, 0)) \in R_\xi \times B(u^\infty, \rho; L^2(\Omega))$, there exists a time $t' > 0$ such that for all $t > t'$ we have $(\boldsymbol{\xi}(t), u(., t)) \in R'_\xi \times B(u^\infty, \rho'; L^2(\Omega))$.*

6.3 Spectral decomposition and EIM

We now present our first approach, based on a spectral decomposition associated to the EIM [129].

6.3.1 Problem statement

Let us first consider a slightly simpler (linear) problem, on which we already see the complexity of the problem.

We wish to consider the equation system (6.9)-(6.12) given by:

$$\frac{d\xi}{dt} = A_\sigma \xi + b_\sigma, \quad t > 0, \quad (6.9)$$

$$\frac{\partial u}{\partial t} - \frac{1}{\alpha} \nabla \cdot (\nabla u) = f \quad \text{in } \Omega \times (0, +\infty), \quad (6.10)$$

$$u(0, t) = \xi_1(t), \quad \text{for all } t > 0, \quad (6.11)$$

$$u(L, t) = \xi_2(t), \quad \text{for all } t > 0, \quad (6.12)$$

We suppose that we have four switched modes:

$$b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad b_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad b_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad b_4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

In order to apply a symbolic (guaranteed) control synthesis method, we need to rewrite the system under the form of an ODE of lowest possible dimension m :

$$\dot{y} = Ay + d_\sigma \quad (6.13)$$

where $y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times m}$, $d_\sigma \in \mathbb{R}^m$.

For this purpose, we will first write a low dimensional equation with a spectral model reduction.

6.3.2 Spectral Model Reduction

We wish to approximate the state $u(x, t)$ of the PDE by a state $\tilde{u}(x, t)$ as close as possible to $u(x, t)$, but which can be computed much more easily than by solving the PDE (e.g. with a finite element method). A natural way of computing an approximate solution of (6.10) is using a modal (spectral) decomposition [40]. An accurate approximate solution of (6.10) can be obtained with few eigen modes when the boundary conditions are homogeneous. This is why we use here a reduced model made of a modal decomposition with a lifting:

$$\tilde{u}(x, t) = \xi_1(t)(1 - x) + \xi_2(t)x + \sum_{i=1}^N \beta_i(t)\varphi_i(x) \quad (6.14)$$

where the β_i are the time coefficients associated to the space functions φ_i , which are precomputed (the computation of the φ_i is detailed in the following).

Let us explain why the lifting is interesting. If we write $\tilde{u}(x, t) = \xi_1(t)(1 - x) + \xi_2(t)x + w(x, t)$ and inject it in (6.10, 6.11, 6.12), we have:

$$\begin{aligned} \alpha \frac{\partial \tilde{u}}{\partial t} - \frac{\partial^2 \tilde{u}}{\partial x^2} &= 0 \quad \text{in } \Omega \\ \tilde{u}(0, t) &= \xi_1(t) \\ \tilde{u}(1, t) &= \xi_2(t) \end{aligned}$$

$$\begin{aligned} \alpha \left(\dot{\xi}_1(t)(1-x) + \dot{\xi}_2(t)x + \frac{\partial w}{\partial t} \right) - \frac{\partial^2 w}{\partial x^2} &= 0 \quad \text{in } \Omega \\ w(0, t) + \xi_1(t) &= \xi_1(t) \\ w(1, t) + \xi_2(t) &= \xi_2(t) \end{aligned}$$

$$\begin{aligned} \alpha \frac{\partial w}{\partial t} - \frac{\partial^2 w}{\partial x^2} &= -\alpha(\dot{\xi}_1(t)(1-x) + \dot{\xi}_2(t)x) \quad \text{in } \Omega \\ w(0, t) &= 0 \\ w(1, t) &= 0 \end{aligned}$$

The lifting $\xi_1(t)(1-x) + \xi_2(t)x$ permits to obtain homogeneous boundary conditions for w . The associated eigenvalue problem $\phi'' = \mu\phi$ with homogeneous boundary conditions leads to eigenmodes (see [40]):

$$\varphi_i(x) = \sqrt{2} \sin(i\pi x) \quad (6.15)$$

Note that the eigenmodes φ_i have been normalized w.r.t. the scalar product $\langle \cdot, \cdot \rangle_\Omega$. A solution for w can then be decomposed on the basis of the eigenmodes $w(x, t) = \sum_{i=1}^{\infty} \beta_i(t)\varphi_i(x)$. Having written w under this last form, an exact solution for equations (6.10,6.11,6.12) can be found as

$$\alpha \frac{\partial w}{\partial t} - \frac{\partial^2 w}{\partial x^2} = \sum_{i=0}^{\infty} \langle -\alpha(\dot{\xi}_1(t)(1-x) + \dot{\xi}_2(t)x), \varphi_i \rangle_\Omega \varphi_i \quad (6.16)$$

Instead, we will look for an approximate solution by truncating the sum at an order N . Let us now find $\tilde{u}(x, t)$ of the form (6.14), solution of the equation system (6.10) with boundary conditions (6.11-6.12). We have:

$$\begin{aligned} \alpha \frac{\partial \tilde{u}}{\partial t} - \frac{\partial^2 \tilde{u}}{\partial x^2} &= 0 \quad \text{in } \Omega \\ \alpha \frac{\partial \tilde{u}}{\partial t} w - \frac{\partial^2 \tilde{u}}{\partial x^2} w &= 0 \quad \text{in } \Omega \quad \forall w \in H_0^1(\Omega) \end{aligned}$$

Writing the weak form formulation and using an integration by parts, we obtain:

$$\alpha \frac{d}{dt} \int_{\Omega} \tilde{u} w dx + \int_{\Omega} \frac{\partial \tilde{u}}{\partial x} \frac{\partial w}{\partial x} dx = 0 \quad \forall w \in H_0^1(\Omega)$$

This is true for all $w \in H_0^1(\Omega)$, we can thus write:

$$\alpha \frac{d}{dt} \int_{\Omega} \tilde{u} w dx + \int_{\Omega} \frac{\partial \tilde{u}}{\partial x} \frac{\partial w}{\partial x} dx = 0, \quad \forall w \in W^k = \text{Vect}(\varphi_k)$$

This leads to:

$$\begin{aligned} \alpha \int_{\Omega} ((1-x)\dot{\xi}_1 + x\dot{\xi}_2) \varphi_k dx + \int_{\Omega} ((1-x)\xi_1 + x\xi_2) \frac{\partial \varphi_k}{\partial x} dx \\ + \alpha \sum_{i=1}^N \dot{\beta}_i \int_{\Omega} \varphi_i \varphi_k dx + \sum_{i=1}^N \beta_i \int_{\Omega} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_k}{\partial x} dx = 0, \quad \forall k = 1, \dots, N \end{aligned}$$

The second term being equal to zero, we then have a low dimensional equation:

$$\alpha C_r \dot{\boldsymbol{\beta}} + K_r \boldsymbol{\beta} = -\alpha \mathbf{F}_r(\dot{\boldsymbol{\xi}}, t) \quad (6.17)$$

with $\boldsymbol{\beta}$ the vector composed of the β_i , which we call the reduced state, $C_{r,ij} = \int_{\Omega} \varphi_i \varphi_j dx$, $K_{r,ij} = \int_{\Omega} \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} dx$ and $F_{r,i}(\dot{\boldsymbol{\xi}}, t) = \int_{\Omega} ((1-x)\dot{\xi}_1 + x\dot{\xi}_2) \varphi_i dx$. Note here that matrices C_r and K_r are diagonal, because functions φ_i are orthogonal. This is one of the main advantages in using such a modal decomposition: an accurate approximate solution can be computed in a very cheap way.

Solving the equation system (6.9-6.10-6.11-6.12) with the reduced order solution (6.14) then leads to solving the reduced system:

$$\begin{pmatrix} \dot{\boldsymbol{\xi}}(t) \\ \dot{\boldsymbol{\beta}}(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1/\alpha C_r^{-1} K_r \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}(t) \\ \boldsymbol{\beta}(t) \end{pmatrix} + \begin{pmatrix} \mathbf{b}_u(t) \\ -C_r^{-1} \mathbf{F}_r(\mathbf{b}_u(t), t) \end{pmatrix} \quad (6.18)$$

However, although the lifting $\xi_1(t)(1-x) + \xi_2(t)x$ permits to construct an accurate reduced model with few functions φ_i , it raises a new problem: the coefficients β_i have no physical meaning. It is thus not trivial to infer a reduced objective (a box, or an objective set) for the reduced state $\boldsymbol{\beta}$. In other words, we do not know where the β_i should stabilize to obtain a PDE state as close to zero as we want.

In order to give a physical meaning to the reduced state, and infer an initial and objective box the reduced state variable, we build a reduced model with slightly different basis functions:

$$\tilde{u}(x, t) = \xi_1(t)(1-x) + \xi_2(t)x + \sum_{i=1}^N \gamma_i(t) \psi_i(x) \quad (6.19)$$

where functions ψ_i interpolate N points x_1, \dots, x_N of the PDE domain, i.e.:

$$\psi_i(x_j) = \delta_{ij} \quad \forall i \in \{1, \dots, N\}. \quad (6.20)$$

Here, δ_{ij} denotes the Kronecker symbol. The functions ψ_i , as well as the interpolated points x_i , are computed with the EIM [129]. The use of the EIM is particularly opportune since it permits to establish an L^∞ error bound which allows to compute a guaranteed control (see Section 6.3.3). Furthermore, the interpolated points are optimal and lead to the lowest possible error bound.

The algorithm for computing the interpolation points is the following:

Let $x_1 = \arg \max_{x \in \Omega} |\varphi_1(x)|$.

Interpolation points $\{x_1, \dots, x_N\}$ are then constructed by induction on $M \leq N$ as follows. For all i , $1 \leq i \leq M-1$, look for h_{ij}^{M-1} such that $\varphi_M(x_i) = \sum_{j=1}^{M-1} h_{ij}^{M-1} \varphi_j(x_i)$, and set $x_M = \arg \max_{x \in \Omega} |\varphi_M(x) - \sum_{j=1}^{M-1} h_{ij}^{M-1} \varphi_j(x)|$. In the EIM terminology, $\sum_{j=1}^{M-1} h_{ij}^{M-1} \varphi_j(\cdot)$ is denoted as the interpolant $\mathcal{I}_{M-1}[\varphi_M(\cdot)]$ since it interpolates exactly $\varphi_M(\cdot)$ in x_1, \dots, x_{M-1} .

Functions ψ_i are then computed as linear combinations of the functions φ_i as follows. For all $1 \leq i \leq N$, solve $\sum_{j=1}^N \varphi_j(x_i) h_{ij}^N = \delta_{ij}$ for h_{ij}^N . Then set $\psi_i = \sum_{j=1}^N h_{ij}^N \varphi_j$ so that functions ψ_i do verify (6.20). In the following, for any $u \in H^1(\Omega)$, we will denote by $\mathcal{I}_N[u(\cdot)]$ the interpolation of order N of $u(\cdot)$, i.e. $\mathcal{I}_N[u(\cdot)] = \sum_{i=1}^N u(x_i) \sum_{j=1}^N h_{ij}^N \varphi_j(\cdot)$.

The reduced system is then computed just as system (6.18) but with functions ψ_i instead of φ_i , this leads to:

$$\begin{pmatrix} \dot{\xi}(t) \\ \dot{\gamma}(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1/\alpha C_r'^{-1} K_r' \end{pmatrix} \begin{pmatrix} \xi(t) \\ \gamma(t) \end{pmatrix} + \begin{pmatrix} \mathbf{b}_u(t) \\ -C_r^{-1} \mathbf{F}'_r(\mathbf{b}_u(t), t) \end{pmatrix} \quad (6.21)$$

with γ the vector composed of the γ_i , which we call the reduced state, $C_{r,ij}' = \int_{\Omega} \psi_i \psi_j dx$, $K_{r,ij}' = \int_{\Omega} \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} dx$ and $F'_{r,i}(\dot{\xi}, t) = \int_{\Omega} ((1-x)\dot{\xi}_1 + x\dot{\xi}_2) \psi_i dx$. Note that here, matrices C'_r and K'_r are no longer diagonal, which results in slightly higher computation costs, but since the dimension of those matrices must be low, this is not prohibitive.

The main interest in using interpolating functions is that the variables γ_i have now a physical meaning: $\gamma_i(t)$ is equal to the value the temperature field (without lifting) in x_i at time t .

We have:

$$\tilde{u}(x_i, t) = \xi_1(t)(1-x_i) + \xi_2(t)x_i + \gamma_i(t), \quad \forall i \in \{1, \dots, N\} \quad (6.22)$$

If we want $u(x_i, t)$ to stay in a box $[u_i^{\min}, u_i^{\max}]$, then we have to ensure that $\xi_1, \xi_2 \in [u_i^{\min}/2, u_i^{\max}/2]$, and $\gamma_i \in [u_i^{\min}/2, u_i^{\max}/2]$ (note that other combinations are possible).

6.3.3 Error bounding

With the above developments, we can ensure that $\tilde{u}(x_i, t)$ reaches infinitely often the box $[u_{\min}, u_{\max}]$ with symbolic methods thanks to equation (6.21). In order to provide a guaranteed controller, we still need to bound the error between the reduced order and the full order system. The minimal result required to ensure recurrence is to bound: $|\tilde{u}(x_i, t) - u(x_i, t)|$ for all $t > 0$. Or, more precisely, for a pattern of length k , compute a bound $\varepsilon_1(k)$ such that:

$$|u(x_i, t_0 + k\tau) - \gamma_i(t_0 + k\tau)| \leq \varepsilon_1(k) \quad \forall i = 1, \dots, N \quad (6.23)$$

But in order to ensure that the whole state $u(x, t)$ stays in $[u_{min}, u_{max}]$, we also need to bound $|\tilde{u}(x, t) - u(x, t)|$ for all $x \in \Omega$ and $t \geq 0$. We thus need to obtain an L^∞ bound. I.e., for all $k \geq 0$, compute a bound $\varepsilon_2(k)$ such that:

$$\|u(\cdot, t_0 + k\tau) - \tilde{u}(\cdot, t_0 + k\tau)\|_{L^\infty(\Omega)} \leq \varepsilon_2(k) \quad (6.24)$$

As mentioned above, the EIM provides an L^∞ error bound. For all $M \geq 0$ and for all $v \in H^1(\Omega)$, let $\{\phi_k\}_{k=1,\dots,M+1}$ be the first $M + 1$ basis functions returned by the EIM for v , we have the following error bound for the EIM interpolant of v :

$$\|v(\cdot) - \mathcal{I}_M[v(\cdot)]\|_{L^\infty(\Omega)} \leq \|\phi_{M+1}(\cdot) - \mathcal{I}_M[\phi_{M+1}(\cdot)]\|_{L^\infty(\Omega)} \quad (6.25)$$

Let us suppose \mathcal{I}_N has been computed as in Section 6.3.2. We have, for all $x \in \Omega$ and for all $t > 0$:

$$|v(x, t) - v_N(x, t)| \leq |v(x, t) - \mathcal{I}_N[v(\cdot, t)](x, t)| + |\mathcal{I}_N[v(\cdot, t)](x, t) - v_N(x, t)| \quad (6.26)$$

The first right-hand term $|v(x, t) - \mathcal{I}_N[v(\cdot, t)](x, t)|$ can be bounded by the EIM bound (6.25). The second right-hand term $|\mathcal{I}_N[v(\cdot, t)](x, t) - v_N(x, t)|$ being constructed with functions $\varphi_1, \dots, \varphi_N$, it is equal, for all $t \geq 0$ and $x \in \Omega$, to the analytical solution of the truncated projected solution:

$$\begin{aligned} \mathcal{I}_N[v(\cdot, t)](x, t) - v_N(x, t) &= \mathcal{I}_N[v(\cdot, t)](x, t) - \sum_{i=1}^N \beta_i(t) \varphi_i(x) \\ \mathcal{I}_N[v(\cdot, t)](x, t) - v_N(x, t) &= \mathcal{I}_N[v(\cdot, t)](x, t) - \sum_{i=1}^N \gamma_i(t) \psi_i(x) \end{aligned}$$

We hoped to bound this term in the same fashion as [60], but it revealed more difficult than expected. The interpolation $\mathcal{I}_N[v(\cdot, t)](x, t)$ should in fact be computed for every time t , and bounding this for every time would be numerically irrelevant. As explained in [60], it is possible to bound such a term when the state v depends explicitly on a parameter, and for which the derivatives w.r.t the parameter can be computed. We hoped to evaluate this term by taking time as a parameter, but this is actually not possible straightforwardly. We however think that this term can be evaluated with further developments, using for example an EIM coupled with another model reduction such as the Proper Generalized Decomposition [45, 46].

6.4 L^2 guaranteed control

Having introduced our attempt of L^∞ guaranteed control, we now present an L^2 approach closer to classical techniques used in the field of structural mechanics. The reduced state we build will now be associated with an L^2 distance instead of an Euclidean one, so that the sets (balls) defined on the reduced space have a meaning directly on the PDE state. We now consider the original problem (6.2)-(6.7).

6.4.1 Transformation of the problem

Denoting by $u_q = u_q(., t)$ the solution of the quasi-static problem at each time t :

$$-\nabla \cdot (\kappa(.) \nabla u_q) = f + \nabla \cdot (\kappa(.) \nabla u^\infty) \text{ in } \Omega, \quad (6.27)$$

$$u_q(0, t) = \xi_1(t) - \xi_1^\infty, \quad (6.28)$$

$$u_q(L, t) = \xi_2(t) - \xi_2^\infty, \quad (6.29)$$

one can express the solution u as the sum of u^∞ , u_q and a function ψ , i.e.

$$u(., t) = u^\infty(.) + u_q(., t) + \psi(., t) \quad (6.30)$$

where $\psi(., t)$ is solution of the heat problem with homogeneous Dirichlet boundary conditions

$$\frac{\partial \psi}{\partial t} - \nabla \cdot (\kappa(.) \nabla \psi) = g(.; \boldsymbol{\xi}(t)) \quad \text{in } \Omega \times (0, +\infty) \quad (6.31)$$

$$\psi(0, t) = \psi(L, t) = 0, \quad t > 0, \quad (6.32)$$

$$\psi(., t = 0) = \psi^0, \quad (6.33)$$

with

$$g(.; \boldsymbol{\xi}(t)) = -\frac{\partial u_q}{\partial t}(.; \boldsymbol{\xi}(t)), \quad \psi^0 = u^0 - u^\infty - u_q(., 0).$$

We thus consider the functional Sobolev space $V = H_0^1(\Omega)$. The weak variational formulation of the problem (6.31)-(6.33) is to find $\psi \in L^2(0, \infty; V)$, $\psi(., t = 0) = \psi^0$, solution of

$$\left(\frac{\partial \psi}{\partial t}, v \right) + (\kappa(.) \nabla \psi, \nabla v) = (g(.; \boldsymbol{\xi}(t)), v) \quad \forall v \in V. \quad (6.34)$$

The decomposition (6.30) actually lets us study the different behaviors we observe in the equation: the quasi-static behavior, which is attained when the time step gets large; and the dynamic behavior, being observed mainly at the beginning of a switch. We also exhibit the objective state, and it will reveal the possible (attainable) target states.

6.4.2 Stability requirements

Let us first show the following proposition:

Proposition 4. *There exist constants $C > 0$ and $L > 0$, such that a sufficient condition to satisfy the stability constraint*

$$\|u(., t) - u^\infty(.)\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t > 0 \quad (6.35)$$

is to fulfill

$$C \|f + \nabla \cdot (\kappa(.) \nabla u^\infty)\|_{L^2(\Omega)} + L \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty + \|\psi(., t)\|_{L^2(\Omega)} \leq \rho. \quad (6.36)$$

where $\psi(., t)$ is solution of (6.31)-(6.33).

Proof. Because of (6.30), the stability requirement

$$\|u(., t) - u^\infty(.)\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t > 0$$

in (6.8) can be equivalently expressed as

$$\|u_q(., t) + \psi(., t)\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t > 0.$$

The solution u_q itself can be decomposed as

$$u_q(., t) = \bar{u}(.) + w_q(., t),$$

where \bar{u} is solution of the steady elliptic problem with homogeneous Dirichlet boundary conditions

$$-\nabla \cdot (\kappa(.) \bar{u}) = f + \nabla \cdot (\kappa(.) \nabla u^\infty) \quad \text{in } \Omega, \quad (6.37)$$

$$\bar{u}(0) = \bar{u}(L) = 0, \quad (6.38)$$

and w_q is solution of the quasi-static problem at each time t :

$$-\nabla \cdot (\kappa(.) \nabla w_q) = 0 \text{ in } \Omega, \quad (6.39)$$

$$w_q(0, t) = \xi_1(t) - \xi_1^\infty, \quad \text{for all } t > 0, \quad (6.40)$$

$$w_q(L, t) = \xi_2(t) - \xi_2^\infty, \quad \text{for all } t > 0. \quad (6.41)$$

The solution \bar{u} is continuous with respect to the source term in (6.37) [70], i.e. there exists $C > 0$ such that:

$$\|\bar{u}\|_V \leq C \|f + \nabla \cdot (\kappa(.) \nabla u^\infty)\|_{L^2(\Omega)}. \quad (6.42)$$

For the solution w_q of (6.39)-(6.41), because of the maximum principle [133], we have

$$\|w_q(., t)\|_{L^\infty(\Omega)} = \max(|\xi_1(t) - \xi_1^\infty|, |\xi_2(t) - \xi_2^\infty|) = \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty. \quad (6.43)$$

Thus,

$$\begin{aligned} \|u_q(., t) + \psi(., t)\|_{L^2(\Omega)} &\leq \|\bar{u}\|_{L^2(\Omega)} + \|w_q\|_{L^2(\Omega)} + \|\psi(., t)\|_{L^2(\Omega)} \\ &\leq \|\bar{u}\|_{L^2(\Omega)} + L \|w_q\|_{L^\infty} + \|\psi(., t)\|_{L^2(\Omega)}, \end{aligned}$$

and finally

$$\begin{aligned} \|u_q(., t) + \psi(., t)\|_{L^2(\Omega)} &\leq C \|f + \nabla \cdot (\kappa(.) \nabla u^\infty)\|_{L^2(\Omega)} \\ &\quad + L \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty + \|\psi(., t)\|_{L^2(\Omega)} \end{aligned}$$

A sufficient condition to satisfy the stability constraint (6.35) is then to fulfill (6.36). \square

The solution ψ lives in an infinite-dimensional space, so that it is hard or impossible to build a control synthesis based on a state-space decomposition. In the sequel of the chapter, we will rather use a low-dimensional approximation $\tilde{\psi}$ (the reduced-order model of ψ) in the form

$$\tilde{\psi}(x, t) = \sum_{k=1}^K \tilde{\beta}_k(t) \varphi^k(x) \quad (6.44)$$

with a reduced basis $\{\varphi^k\}_{k=1,\dots,K}$ assumed to be orthonormal in $L^2(\Omega)$. In the sequel we will denote by W^K the linear vector space of dimension K spanned by the reduced basis $\{\varphi^k\}_k$:

$$W^K = \text{span}(\varphi^1, \dots, \varphi^K).$$

Denoting by $\tilde{\boldsymbol{\beta}}(t) = (\tilde{\beta}_1(t), \dots, \tilde{\beta}_K(t))^T$ the vector of coefficients, we then have

$$\|\tilde{\psi}(\cdot, t)\|_{L^2(\Omega)} = \|\tilde{\boldsymbol{\beta}}(t)\|_{2, \mathbb{R}^K}.$$

By the triangular inequality we can write

$$\|\psi(\cdot, t)\|_{L^2(\Omega)} \leq \|\psi(\cdot, t) - \tilde{\psi}(\cdot, t)\|_{L^2(\Omega)} + \|\tilde{\psi}(\cdot, t)\|_{L^2(\Omega)} \quad (6.45)$$

$$\leq \|\psi(\cdot, t) - \tilde{\psi}(\cdot, t)\|_{L^2(\Omega)} + \|\tilde{\boldsymbol{\beta}}(t)\|_2. \quad (6.46)$$

Let us assume that we have the stability estimate for the reduced-order approximation: there exists a constant $\mu > 0$ such that

$$\|\psi(\cdot, t) - \tilde{\psi}(\cdot, t)\|_{L^2(\Omega)} \leq \mu \|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \quad \forall t \in [0, \tau] \quad (6.47)$$

for any constant control mode $\sigma \in \{1, \dots, M\}$ (uniform stability with respect to the controls). This hypothesis can actually be verified with a proper construction of the reduced basis. Then, a more restrictive sufficient condition to fulfill the stability constraint (6.35) is to verify

$$\begin{aligned} C \|f + \nabla \cdot (\kappa(\cdot) \nabla u^\sigma)\|_{L^2(\Omega)} + L \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\sigma\|_\infty \\ + \|\tilde{\boldsymbol{\beta}}(t)\|_2 + \mu \|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \leq \rho. \end{aligned} \quad (6.48)$$

This equation is interesting since it enlightens the different controllable and uncontrollable terms.

Let us denote by $\pi^K : V \rightarrow W^K$ the continuous linear orthogonal projection operator over the low-order space W^K . Still by a triangular inequality, we have

$$\|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \leq \|\psi^0 - \pi^K \psi^0\|_{L^2(\Omega)} + \|\pi^K \psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)},$$

The projection $\pi^K \psi^0$ is given by

$$\pi^K \psi^0 = \sum_{k=1}^K \beta_k^0 \varphi^k,$$

with $\beta_k^0 = (\psi^0, \varphi^k)_{L^2(\Omega)}$, $k = 1, \dots, K$. By denoting $\boldsymbol{\beta}^0 = (\beta_1^0, \dots, \beta_K^0)$, we then have

$$\|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \leq \|\psi^0 - \pi^K \psi^0\|_{L^2(\Omega)} + \|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^0\|_2,$$

We thus have a reduced-order version of Proposition 4:

Proposition 5. *Under the above-mentioned notations, let us suppose that there exists $\mu > 0$ such that (6.47) holds. There exist constants $C > 0$ and $L > 0$ such that a sufficient condition to satisfy the stability constraint (6.35) is to fulfill*

$$\begin{aligned} C \|f + \nabla \cdot (\kappa(\cdot) \nabla u^\infty)\|_{L^2(\Omega)} + L \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty + \|\tilde{\boldsymbol{\beta}}(t)\|_2 \\ + \mu \|\psi^0 - \pi^K \psi^0\|_{L^2(\Omega)} + \mu \|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^0\|_2 \leq \rho. \end{aligned} \quad (6.49)$$

Let us interpret equation (6.49). If we want to fulfill the inequality (6.49), all the terms in the left-hand side have to be “small enough”. In particular, this means that u^∞ should be compatible with the source term in the sense that

$$-\nabla \cdot (\kappa(\cdot) \nabla u^\infty) \approx f \quad \text{in } \Omega.$$

Moreover, the vector state $\boldsymbol{\xi}(t)$ should stay close to $\boldsymbol{\xi}^\infty$ for any time, the coefficient vector $\tilde{\boldsymbol{\beta}}(t)$ in the reduced-space has to stay rather small in norm. The terms $L \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty$ and $\|\tilde{\boldsymbol{\beta}}(t)\|_2$ are actually controlled terms, these are the ones we have to synthesize a controller with our symbolic approach. Note that $L \|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty$ actually justifies that we stabilize $\boldsymbol{\xi}$ in a box. We should also have $\|\boldsymbol{\beta}^0 - \tilde{\boldsymbol{\beta}}^0\|$ small enough for any initial data subject to any admissible control, as well as $\|\psi^0 - \pi^K \psi^0\|_{L^2(\Omega)}$, meaning that the reduced basis is able to correctly reproduce any admissible initial data. In a nutshell, we have to synthesize a controller for the reduced state $(\boldsymbol{\xi}, \tilde{\boldsymbol{\beta}})$ using symbolic methods, and the other terms are fulfilled as long as the objective state is compatible with the source term, and the reduced basis represents accurately the initial conditions.

6.4.3 Strategy for stability control

At a switch time (reset to time zero for the sake of simplicity), consider the approximate heat solution

$$\tilde{u}^0 = u^\infty + u_q(\cdot; \boldsymbol{\xi}^0) + \tilde{\psi}^0$$

and the exact solution written as

$$u^0 = u^\infty + u_q(\cdot; \boldsymbol{\xi}^0) + \psi^0.$$

Considering Problem 4, we assume the following initial properties: there exist constants $\delta_\xi, \rho_\beta, \delta > 0$ such that

$$L\|\boldsymbol{\xi}^0 - \boldsymbol{\xi}^\infty\|_\infty \leq \delta_\xi, \quad (6.50)$$

$$\|\tilde{\boldsymbol{\beta}}^0\|_2 \leq \rho_\beta, \quad (6.51)$$

$$\|\psi^0 - \tilde{\psi}^0\|_{L^2(\Omega)} \leq \delta. \quad (6.52)$$

It will be assumed that, δ_ξ, ρ_β and δ are such that

$$c_1 + \delta_\xi + \rho_\beta + \delta \leq \rho \quad (6.53)$$

where $c_1 = C \|f + \nabla \cdot (\kappa(\cdot) \nabla u^\infty)\|_{L^2(\Omega)}$. We look for controls that preserve these properties (ans solve Problem 4). I.e., we look for control modes such that, for all time $t \in [0, \tau]$ (before the next switch), we have:

$$L\|\boldsymbol{\xi}(t) - \boldsymbol{\xi}^\infty\|_\infty \leq \delta_\xi, \quad (6.54)$$

$$\|\tilde{\boldsymbol{\beta}}(t)\|_2 \leq \rho_\beta, \quad (6.55)$$

$$\|\psi(t) - \tilde{\psi}(\tau)\|_{L^2(\Omega)} \leq \delta. \quad (6.56)$$

Then by construction we will automatically fulfill the stability requirement (6.35) on the heat solution for a given control mode σ , i.e.

$$\|u(\cdot, t) - u^\infty\|_{L^2(\Omega)} \leq \rho \quad \text{for all } t \in (0, \tau]. \quad (6.57)$$

These properties can also be ensured for control sequences $\pi = (\sigma_1, \dots, \sigma_k)$, and have to be verified for all $t \in [0, k\tau]$.

Remark 9. From (6.50) and (6.54), it is appropriate to choose the recurrence set R_ξ for the $\boldsymbol{\xi}(\cdot)$ variable as the ball of center $\boldsymbol{\xi}^\infty$ and radius δ_ξ for the topology induced by the norm $\|\cdot\|_\infty$, i.e. a box centered around $\boldsymbol{\xi}^\infty$.

The synthesis can now be performed, provided that the reduced basis ensures for all $t \in [0, k\tau]$, $\|\psi(t) - \tilde{\psi}(\tau)\|_{L^2(\Omega)} \leq \delta$ (this point is addressed in the following). The state $\boldsymbol{\xi}$ is subject to an ODE (of dimension 2 in our case), and it can thus be controlled easily with the methods described in the previous chapters. Besides, the reduced state $\tilde{\boldsymbol{\beta}}$ verifies a nonlinear ODE. Indeed, the reduced-order approximation $\tilde{\psi} \in W^K$ is chosen in such a way that it verifies the equation:

$$\left(\frac{\partial \tilde{\psi}}{\partial t}, w \right) + (\kappa(\cdot) \nabla \tilde{\psi}, \nabla w) = (g(\cdot; \boldsymbol{\xi}(t)), w) \quad \forall w \in W^K, \quad (6.58)$$

$$\tilde{\psi}(\cdot, t=0) = \tilde{\psi}^0. \quad (6.59)$$

The basis functions $(\varphi^1, \dots, \varphi^K)$ being chosen orthonormal in $L^2(\Omega)$, it leads to a system of differential equations, for all $1 \leq i \leq K$:

$$\dot{\tilde{\beta}}_i + \beta_i(\kappa(\cdot) \nabla \varphi_i, \nabla \varphi_j) = (g(\cdot; \boldsymbol{\xi}(t)), \varphi_j) \quad 1 \leq j \leq K, \quad (6.60)$$

which is a system of nonlinear differential equations, that can be handled by the synthesis algorithm presented in Chapter 3.3. This algorithm is particularly adapted to this purpose since $\|\tilde{\psi}(., t)\|_{L^2(\Omega)} = \|\tilde{\beta}(t)\|_{2, \mathbb{R}^K}$. By covering the ball $B(0, \rho_\beta; L^2(\Omega))$ with smaller balls, we ensure (6.55). Exactly as in Chapter 3.3, we just have to verify that the images of the ball after one (or several) time steps are included in the objective (the objective being convex, we do not need to verify the property for the whole tube). Furthermore, verifying the inclusion of a ball in a ball is numerically very cheap.

6.4.4 Certified reduced basis for control

Let us now present the construction of a proper reduced basis, allowing to verify (6.49). Considering the space of all possible sequences of switched controls of lengths less than M , we have to derive a reduced-order model which guarantees a prescribed accuracy for any switched control sequence.

For that purpose, it seems appropriate to build a reduced-order model using a posteriori error estimates within an iterative greedy approach.

Let us consider a low-dimensional vector space $W \subset V$ and a Galerkin approach with a reduced-order approximation $\tilde{\psi}$ solution of the finite dimensional variational problem

$$\left(\frac{\partial \tilde{\psi}}{\partial t}, w \right) + (\kappa(.) \nabla \tilde{\psi}, \nabla w) = (g(.; \xi(t)), w) \quad \forall w \in W, \quad (6.61)$$

$$\tilde{\psi}(., t = 0) = \tilde{\psi}^0. \quad (6.62)$$

A posteriori error estimation

From (6.34), one can directly derive a variational problem for the error function $e := \psi - \tilde{\psi}$: $\forall v \in V$,

$$\left(\frac{\partial e}{\partial t}, v \right) + (\kappa(.) \nabla e, \nabla v) = (g(.; \xi(t), v) - \left(\frac{\partial \tilde{\psi}}{\partial t}, v \right) - (\kappa(.) \nabla \tilde{\psi}, \nabla v), \quad (6.63)$$

$$e(., t = 0) = \psi^0 - \tilde{\psi}^0 := e^0. \quad (6.64)$$

The right hand side defines a residual linear form r_ξ depending on $\xi(t)$:

$$r_\xi(v) = (g(.; \xi(t), v) - \left(\frac{\partial \tilde{\psi}}{\partial t}, v \right) - (\kappa(.) \nabla \tilde{\psi}, \nabla v), \quad \forall v \in V. \quad (6.65)$$

By construction of the approximate solution $\tilde{\psi}$, from (6.58) we clearly have

$$r_\xi(w) = 0 \quad \forall w \in W.$$

One can define a norm for r_ξ in the dual space V' of V :

$$\|r_\xi\|_{V'} = \sup_{\|v\|_V \leq 1} |r_\xi(v)|.$$

Considering the particular test function $v = e$, we have

$$\frac{1}{2} \frac{d}{dt} \|e\|_{L^2}^2 + \|\kappa(\cdot) \nabla e\|_{L^2}^2 = r_\xi(e).$$

From Poincaré's inequality

$$\|v\|_{L^2} \leq C_\Omega \|\nabla v\| \quad \forall v \in V$$

and the lower bound κ_m of κ , we have also

$$\frac{1}{2} \frac{d}{dt} \|e\|_{L^2}^2 \leq -\frac{\kappa_m}{C_\Omega^2} \|e\|_{L^2}^2 + \|r_\xi\|_{V'}(t) \|e\|_{L^2}.$$

Let us denote the constant

$$\tilde{\eta} = \sup_{\xi(\cdot)} \sup_{t \geq 0} \|r_\xi\|_{V'}(t) \quad (6.66)$$

with $\sigma(\cdot) \in \Sigma^\tau$ such that $\xi(t) \in R_\xi$ for all $t \geq 0$, $\xi(\cdot)$ subject to

$$\dot{\xi} = A_\sigma \xi + B w_\sigma, \quad \xi(0) = \xi^0.$$

So we have the estimation

$$\frac{1}{2} \frac{d}{dt} \|e\|_{L^2}^2 \leq -\frac{\kappa_m}{C_\Omega^2} \|e\|_{L^2}^2 + \tilde{\eta} \|e\|_{L^2}. \quad (6.67)$$

By using the Young inequality

$$\tilde{\eta} \|e(t)\|_{L^2} \leq \frac{\kappa_m}{2C_\Omega^2} \|e(t)\|_{L^2}^2 + \frac{C_\Omega^2}{2\kappa_m} \tilde{\eta}^2$$

and Gronwall's lemma to the resulting estimate, we get the error estimate in L^2 -norm

$$\|e(t)\|_{L^2}^2 \leq \exp\left(-\frac{\kappa_m}{C_\Omega^2} t\right) \|e^0\|_{L^2}^2 + \frac{\tilde{\eta}^2 C_\Omega^4}{\kappa_m^2} \left(1 - \exp\left(-\frac{\kappa_m}{C_\Omega^2} t\right)\right). \quad (6.68)$$

From (6.68), we have the straightforward property:

Proposition 6. *A sufficient condition to guarantee*

$$\|e(t)\|_{L^2} \leq \|e(0)\|_{L^2} \quad \forall t > 0$$

is to fulfill the inequality

$$\frac{\tilde{\eta} C_\Omega^2}{\kappa_m} \leq \|e_0\|. \quad (6.69)$$

Remark 10. *Because the approximate problem is built from a Galerkin projection method, it is expected that the constant $\tilde{\eta}$ becomes small for a “good” finite discrete space W . So for an accuracy level $\|e_0\|_{L^2} \leq \delta$ on the initial state, the goal is to find a discrete reduced-order space W such that the inequality $\tilde{\eta} \leq \frac{\kappa_m \delta}{C_\Omega^2}$ holds. The constant $\tilde{\eta}$ defined in (6.66) is a uniform upper bound of the residual quantity, meaning that $\tilde{\eta}$ should be rather small for any switched control sequence $\sigma(\cdot)$ for practical use. This remark leads us to the following greedy algorithm for the construction of the reduced order basis (RB).*

Greedy algorithm and reduced bases

The greedy algorithm also to compute a reduced basis that spans the discrete space \tilde{W} in an iterative and greedy manner.

- First iterate $k = 1$. Define $\delta > 0$ and a residual threshold

$$r_M = \frac{\kappa_m \delta}{C_\Omega^2}.$$

Let us assume that $\psi \in V$ and $\psi^0 \neq 0$. Let us consider first

$$\varphi^1 = \frac{\psi^0}{\|\psi^0\|}$$

and $W^{(1)} = \text{span}(\varphi^1)$. Define a random sequence of control sequences $\sigma(\cdot) \in \Sigma^\tau$, i.e. control sequences of length less than K . As soon as

$$\|r_\xi\|_{V'}(t) < r_M,$$

solve the reduced-order model

$$\left(\frac{\partial \tilde{\psi}^{(1)}}{\partial t}, w \right) + (\kappa(\cdot) \nabla \tilde{\psi}^{(1)}, \nabla w) = (g(\cdot; \boldsymbol{\xi}(t), w) \quad \forall w \in W^{(1)}, \quad (6.70)$$

$$\tilde{\psi}^{(1)}(\cdot, t=0) = \tilde{\psi}^0. \quad (6.71)$$

- If there is a time $t^{(1)} > 0$ such that $\|r_\xi\|_{V'}(t^{(1)}) = r_M$, then compute

$$v^{(2)} = \arg \max_{\|v\|=1} |r_{\xi(t^{(1)})}(v)|$$

and define

$$\varphi^2 = \frac{v^{(2)}}{\|v^{(2)}\|}, \quad W^{(2)} = \text{span}(\varphi^1, \varphi^2).$$

- The reduced-order model at iterate (k) is

$$\left(\frac{\partial \tilde{\psi}^{(k)}}{\partial t}, w \right) + (\kappa(\cdot) \nabla \tilde{\psi}^{(k)}, \nabla w) = (g(\cdot; \boldsymbol{\xi}(t), w) \quad \forall w \in W^{(k)}, \quad (6.72)$$

$$\tilde{\psi}^{(k)}(\cdot, t=0) = \tilde{\psi}^0. \quad (6.73)$$

- Repeat until $\|r_\xi\|_{V'} < r_M$ for all time $t > 0$. Let us denote by K the final rank and $W^{(K)} = \text{span}(\varphi^1, \varphi^2, \dots, \varphi^K)$ the associated discrete space.

For performance and complexity aspects, the rank K is expected to be not too large. For that, the initial accuracy radius δ should be chosen not to small.

6.4.5 Numerical experiment for the L^2 guaranteed control synthesis by stability of error balls

As a proof of concept, we apply the strategy described in Section 6.4.3, on the case study (6.9-6.12) with a time step $\tau = 0.05$. The reduced basis used is a simple

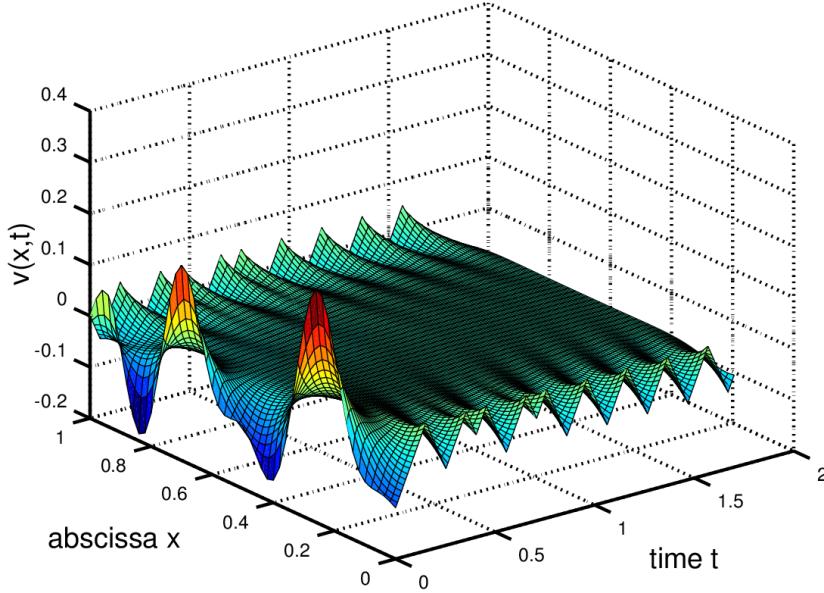


Figure 6.1: Simulation of the controller.

spectral decomposition, as constructed in Section 6.3. The spectral decomposition allows to fulfill (6.47) with $\mu = 1$, and thus to apply Proposition 5. The reduced basis is truncated at $K = 4$ eigenmodes. Associated to the ODE, we thus get a reduced system of dimension 6. Using control sequences of length 8, and a decomposition of the reduced state-space in $4^6 = 4096$ balls, we manage to synthesize a controller in approximately 20 minutes, with an objective state $(\xi^\infty, u^\infty) = (0_{\mathbb{R}^2}, 0_{L^2(\Omega)})$ and guaranteed L^2 error of $\rho = 0.5$. A simulation of the controller is given in Figure 6.1, where the initial condition is set as a random combination of the first ten eigenmodes and a lifting, such that (6.50-6.52) holds with $\delta_\xi = 0.2$, $\rho_\beta = 0.2$ and $\delta = 0.1$.

6.5 Reliable measurements, online control, and other applications

A first challenge for the future is to handle other types of PDEs (e.g. hyperbolic) with such methods, as well as different types of controls and coupling. A first application that could be interesting in the continuation of this work would be to apply such an approach to synthesizing a guaranteed controller for the SCOLE (Spacecraft COnrol Laboratory Experiment) model. It is described by the following

equations, for all $t > 0$:

$$\rho v_{tt}(x, t) + EIv_{xxxx}(x, t) + \rho Bv_t(x, t) = \rho\omega^2 v(x, t), \quad \forall x \in [0, L] \quad (6.74)$$

$$v(0, t) = v_x(0, t) = v_{xx}(L, t) = v_{xxx}(L, t) = 0, \quad (6.75)$$

$$\omega_t(t) = \frac{\Gamma(t) - 2\omega(t) \int_0^L \rho v(x, t)v_t(x, t)dx}{I_d + \int_0^L \rho v^2(x, t)dx}. \quad (6.76)$$

It actually models a metal beam fixed on a rotating rigid body, which rotation is controlled by the input torque Γ . We thus have a hyperbolic PDE coupled to an ODE, but in this case, the coupling goes through a Dirichlet boundary condition. Many theoretical approaches have been developed for this case study and its multiple variations: [22, 33, 50, 51]. We believe that a symbolic approach could be used to handle this case study.

While we gave some possible directions for the use of symbolic control applied to PDEs, some aspects are still not taken into account. One of which is partial observation, which was partly tackled in Chapter 5. In a general case, this should be taken into account by considering a system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t)) + \varepsilon(t; \mathbf{x}(t); \mu)$$

$$\mathbf{y}(t) = \mathcal{L}(\mathbf{x}(t)) + w(t).$$

for high dimensional ODEs. A general case is however more difficult to establish for PDEs since the observation can be performed locally (in a point) or in a distributed manner on a portion of the boundary, or on a portion of the domain of the PDE. Nevertheless, the possible objectives aimed by considering partial observation are numerous:

- taking state estimation errors into account
- evaluating the OSL/Lipschitz constants and parameters of the system
- use of Kalman filter-like state estimators
- partial observation and much more...

Chapter 7

Conclusions and perspectives

Summary

In this thesis, we proposed symbolic methods to synthesize “correct-by-design” state-dependent controllers for sampled switched systems, aimed at extending the field of application of former methods. A first step, introduced in Chapter 3, was the handling of nonlinear dynamics, made possible with appropriate reachability computation methods, using guaranteed numerical schemes. We presented an approach based on guaranteed Runge-Kutta schemes and interval analysis, accurate and fast enough to compete with state-of-the-art tools. We then presented a novel approach renewing the Euler method, thanks to the use of the OSL property, which is a much weaker hypothesis than those used in various symbolic tools such as incremental stability or monotonicity. The Euler approach led to impressive computation times compared to other symbolic tools, even if it failed on some systems presenting large positive OSL constants.

On account of the inherent exponential complexity of symbolic methods, we proposed in Chapter 4 compositional approaches for the synthesis of controllers, made possible with over-approximation techniques which allow us to synthesize local controllers, on sub-parts of the system. We provided three procedures:

- The first is available for linear systems and ensures discrete-time properties and relies on zonotopes, it is associated to an iterative backward reachability procedure extending the basic decomposition method.
- The second is available for nonlinear systems and ensures continuous-time properties thanks to the use of guaranteed Runge-Kutta schemes.
- The third one is available for nonlinear systems and relies on the Euler method introduced in Chapter 3. It can be used in a compositional way with the use of a weaker variant of the incremental input-to-state stability.

In Chapter 5, we laid out an approach allowing to control high dimensional ODEs obtained from the discretization of PDEs. We proposed to use approximate models obtained by balanced truncation in order to synthesize controllers at the reduced-order level, and by appropriately bounding the trajectory errors between the high and low dimensional systems, infer guaranteed controllers. We also gave initiating works to the use of state observers in the case of partial observation.

In Chapter 6, we gave two approaches relying on reduced-order modeling with the aim of obtaining guaranteed controllers for non discretized PDEs. Our first approach made use of the EIM and a spectral model reduction. These works are a first step to the synthesis of L^∞ guaranteed controllers, but the bounding of the reduction error revealed more complicated than expected, and we hope that further collaborations with researchers from the field of computational mechanics can complete this approach. We finally gave an operational procedure for obtaining L^2 guaranteed controllers, using Galerkin based reduced-order models, a proper decomposition of the terms of the solution, and an L^2 topology for the reduced-order level. This

allowed to synthesize a guaranteed controller for a coupled ODE-PDE system.

As a summary of this summary, the main contributions of this thesis are the following:

- improvements of the synthesis algorithms allowing better performances;
- innovative numerical schemes for the handling of nonlinear systems;
- compositional methods to break the complexity of the algorithms;
- reduced-order modeling for the handling of PDEs.

Perspectives and future research

The Euler method proposed in Chapter 3, even though very efficient on systems presenting negative OSL constants, can still be improved. A possible line of research for its development is the use of a posteriori error estimation, such as in [145], possibly improving the current results for negative OSL constants. The use of dual methods seems to be an appropriate way [80].

In the compositional reachability procedure proposed in Chapter 4.2, the choice of the safety parameter ε is left to the user. An interesting continuation of this work would be to automatically synthesize this parameter. This could be performed using approaches used in contract based design [26, 161]. More precisely, the use of parametric contracts allows to determine admissible parameters [102], in the same vein as [2, 99], and could be applied in our context. Furthermore, in this thesis, we do not discuss the choice of the decomposition in sub-systems. Certain automatic methods provide the best decompositions [143, 177]. This kind of techniques could be extended to our methods, with the objective of obtaining the least complex symbolic model.

The research of patterns is still one of the most cost consuming tasks in our algorithms. The recent development of learning algorithms might be a way of drastically lowering the number of tests performed when the length of patterns considered is long (such as in the path planning problem A.6). Furthermore, it could bring optimality in the method. The patterns we select here are the shortest ones, but optimizing the energy consumption of a system is a very topical issue, and learning algorithms can steer us to this objective.

As for PDEs, we would like to point out that compositional approaches can actually be compared to domain decomposition methods used in computational mechanics [151]. It could be interesting to study the compatibility of both methods. In the case where multiple actuators are applied, for example, on a flexible beam, a domain decomposition method can be used to compute a solution for the displacement in the beam. If a compositional synthesis sharing this domain decomposition were possible, we could contemplate applying our methods on much more complex and realistic case studies. However, this kind of models being usually used in private companies, further collaborations with the latter might be needed to see the

applicability of such methods.

More generally, regarding PDEs and Chapter 6, we only applied our method to a single case study. It seems mandatory to test our method on other types of equations and case studies, and the SCOLE model might be a start. All in all, it is only by continuing this line of research that we may see if a generic symbolic method can be inferred for PDEs with our approach.

Appendices

Appendix A

Case studies modeled by ODEs

A.1 Boost DC-DC converter

This linear example is taken from [27]. The system is a boost DC-DC converter with one switching cell. There are two switching modes depending on the position of the switching cell. The dynamics is given by the equation $\dot{x}(t) = A_{\sigma(t)}x(t) + B_{\sigma(t)}$ with $\sigma(t) \in U = \{1, 2\}$. The two modes are given by the matrices:

$$A_1 = \begin{pmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c} \frac{1}{r_0+r_c} \end{pmatrix} \quad B_1 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} -\frac{1}{x_l} (r_l + \frac{r_0 \cdot r_c}{r_0+r_c}) & -\frac{1}{x_l} \frac{r_0}{r_0+r_c} \\ \frac{1}{x_c} \frac{r_0}{r_0+r_c} & -\frac{1}{x_c} \frac{r_0}{r_0+r_c} \end{pmatrix} \quad B_2 = \begin{pmatrix} \frac{v_s}{x_l} \\ 0 \end{pmatrix}$$

with $x_c = 70$, $x_l = 3$, $r_c = 0.005$, $r_l = 0.05$, $r_0 = 1$, $v_s = 1$. The sampling period is $\tau = 0.5$. The parameters are exact and there is no perturbation.

A.2 Two-room apartment

This case study is based a simple model of a two-room apartment, heated by one heater in each room (adapted from [76]). In this example, the objective is to control the temperature of both rooms. There is heat exchange between the two rooms and with the environment. The *continuous* dynamics of the system is given by the equation:

$$\begin{pmatrix} \dot{T}_1 \\ \dot{T}_2 \end{pmatrix} = \begin{pmatrix} -\alpha_{21} - \alpha_{e1} - \alpha_f u_1 & \alpha_{21} \\ \alpha_{12} & -\alpha_{12} - \alpha_{e2} - \alpha_f u_2 \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} + \begin{pmatrix} \alpha_{e1} T_e + \alpha_f T_f u_1 \\ \alpha_{e2} T_e + \alpha_f T_f u_2 \end{pmatrix}.$$

Here T_1 and T_2 are the temperatures of the two rooms, and the state of the system corresponds to $T = (T_1, T_2)$. The control mode variable u_1 (respectively u_2) can take the values 0 or 1, depending on whether the heater in room 1 (respectively room 2) is switched off or on (hence $U_1 = U_2 = \{0, 1\}$). Hence, here $n_1 = n_2 = 1$, $N_1 = N_2 = 2$, and $n = 2$ and $N = 4$.

Temperature T_e corresponds to the temperature of the environment, and T_f to the temperature of the heaters. The values of the different parameters are as follows: $\alpha_{12} = 5 \times 10^{-2}$, $\alpha_{21} = 5 \times 10^{-2}$, $\alpha_{e1} = 5 \times 10^{-3}$, $\alpha_{e2} = 5 \times 10^{-3}$, $\alpha_f = 8.3 \times 10^{-3}$, $T_e = 10$ and $T_f = 35$.

A.3 A polynomial example

In this case study, we consider the polynomial system taken from [126], presented as a difficult example:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 - 1.5x_1 - 0.5x_1^3 + u_1 + d_1 \\ x_1 + u_2 + d_2 \end{bmatrix}. \quad (\text{A.1})$$

The control inputs are given by $u = (u_1, u_2) = K_{\sigma(t)}(x_1, x_2)$, $\sigma(t) \in U = \{1, 2, 3, 4\}$, which correspond to four different state feedback controllers $K_1(x) = (0, -x_2^2 + 2)$, $K_2(x) = (0, -x_2)$, $K_3(x) = (2, 10)$, $K_4(x) = (-1.5, 10)$. We thus have four switching modes. The disturbance $d = (d_1, d_2)$ lies in $[-0.005, 0.005] \times [-0.005, 0.005]$. The objective is to visit infinitely often two zones R_1 and R_2 , without going out of a safety zone S , and while never crossing a forbidden zone B . The sampling period is set to $\tau = 0.15$.

A.4 Four room apartment

We consider a building ventilation application adapted from [134]. The system is a four room apartment subject to heat transfer between the rooms, with the external environment, with the underfloor, and with human beings. The dynamics of the system is given by the following equation:

$$\frac{dT_i}{dt} = \sum_{j \in \mathcal{N}^* \setminus \{i\}} a_{ij}(T_j - T_i) + \delta_{s_i} b_i(T_{s_i}^4 - T_i^4) + c_i \max \left(0, \frac{V_i - V_i^*}{\bar{V}_i - V_i^*} \right) (T_u - T_i). \quad (\text{A.2})$$

The state of the system is given by the temperatures in the rooms T_i , for $i \in \mathcal{N} = \{1, \dots, 4\}$. Room i is subject to heat exchange with different entities stated by the indexes $\mathcal{N}^* = \{1, 2, 3, 4, u, o, c\}$.

The heat transfer between the rooms is given by the coefficients a_{ij} for $i, j \in \mathcal{N}^2$, and the different perturbations are the following:

- The convective heat transfer with the external environment: it has an effect on room i with the coefficient a_{io} and the outside temperature T_o , varying between $27^\circ C$ and $30^\circ C$.
- The convective heat transfer through the ceiling: it has an effect on room i with the coefficient a_{ic} and the ceiling temperature T_c , varying between $27^\circ C$ and $30^\circ C$.

Room i	1	2	3	4
$a_{i,1}$		7.60×10^{-5}		1.09×10^{-4}
$a_{i,2}$	2.85×10^{-4}		1.79×10^{-4}	
$a_{i,3}$		1.89×10^{-4}		1.07×10^{-4}
$a_{i,4}$	2.47×10^{-4}		3.81×10^{-4}	
$a_{i,u}$	7.36×10^{-5}	7.02×10^{-5}	3.45×10^{-5}	3.26×10^{-5}
$a_{i,o}$	9.27×10^{-5}	2.42×10^{-4}	3.21×10^{-8}	1.73×10^{-4}
$a_{i,c}$	5.78×10^{-4}	6.21×10^{-4}	5.64×10^{-4}	5.99×10^{-4}
b_i	3.12×10^{-17}	2.55×10^{-16}	8.57×10^{-13}	3.57×10^{-17}
T_{s_i}	3.73×10^3	1.78×10^3	3.80×10^2	3.93×10^3
c_i	2.12×10^{-3}	1.88×10^{-3}	3.05×10^{-3}	1.40×10^{-3}

Table A.1: Identified parameters for the four room apartment model (A.2).

- The convective heat transfer with the underfloor: it is given by the coefficient a_{iu} and the underfloor temperature T_u , set to $17^\circ C$ (T_u is constant, regulated by a PID controller).
- The perturbation induced by the presence of humans, modeled by a radiation term: it is given in room i by the term $\delta_{s_i} b_i (T_{s_i}^4 - T_i^4)$, the parameter δ_{s_i} is equal to 1 when someone is present in room i , 0 otherwise, and T_{s_i} is a given identified parameter.

The control V_i , $i \in \mathcal{N}$, is applied through the term $c_i \max(0, \frac{V_i - V_i^*}{V_i^* - V_i})(T_u - T_i)$. A voltage V_i is applied to force ventilation from the underfloor to room i , and the command of an underfloor fan is subject to a dry friction. Because we work in a switched control framework, V_i can take only discrete values, which removes the problem of dealing with a “max” function in interval analysis. In the experiment, V_1 and V_4 can take the values 0V or 3.5V, and V_2 and V_3 can take the values 0V or 3V. This leads to a system of the form of Equation (2.1) with $\sigma(t) \in U = \{1, \dots, 16\}$, the 16 switching modes corresponding to the different possible combinations of voltages V_i . The sampling period is $\tau = 10s$.

The parameters T_{s_i} , V_i^* , \bar{V}_i , a_{ij} , b_i , c_i are given in Table A.1 and have been identified with a proper identification procedure detailed in [137]. Note that here we have neglected the term $\sum_{j \in \mathcal{N}} \delta_{d_{ij}} c_{i,j} * h(T_j - T_i)$ of [134], representing the perturbation induced by the open or closed state of the doors between the rooms. Taking a “max” function into account with set based methods is actually still a difficult task. However, this term could have been taken into account with a proper regularization (smoothing).

A.5 Linearized four room apartment

This case study is a linearized version of A.4. The dynamics of the system is given by the same equation, except that the nonlinear term $\delta_{s_i} b_i (T_{s_i}^4 - T_i^4)$ is neglected. The system is thus ruled by the equation:

$$\frac{dT_i}{dt} = \sum_{j \in \mathcal{N}^* \setminus \{i\}} a_{ij} (T_j - T_i) + c_i \max \left(0, \frac{V_i - V_i^*}{\bar{V}_i - V_i^*} \right) (T_u - T_i). \quad (\text{A.3})$$

The behavior of the system is exactly the same as case study A.4, except that the perturbation induced by the presence of humans is neglected. The parameters of the model are the same and are given in Table A.1.

A.6 A path planning problem

This case study is based on a model of a vehicle initially introduced in [19] and successfully controlled in [154, 175] with the tools PESSOA and SCOTS. In this model, the motion of the front and rear pairs of wheels are approximated by a single front wheel and a single rear wheel. The dynamics of the vehicle is given by:

$$\begin{aligned} \dot{x} &= v_0 \frac{\cos(\alpha+\theta)}{\cos(\alpha)} \\ \dot{y} &= v_0 \frac{\sin(\alpha+\theta)}{\cos(\alpha)} \\ \dot{\theta} &= \frac{v_0}{b} \tan(\delta) \end{aligned} \quad (\text{A.4})$$

where $\alpha = \arctan(a \tan(\delta)/b)$. The system is thus of dimension 3, (x, y) is the position of the vehicle, while θ is the orientation of the vehicle. The control inputs are v_0 , an input velocity, and δ , the steering angle of the rear wheel. The parameters are: $a = 0.5$, $b = 1$. Just as in [154, 175], we suppose that the control inputs are piecewise constant, which leads to a switched system of the form of Equation (2.1) with no perturbation. The objective is to send the vehicle into an objective region $R_2 = [9, 9.5] \times [0, 0.5] \times]-\infty, +\infty[$ from an initial region $R_1 = [0, 0.5] \times [0, 0.5] \times [0, 0]$. The safety set is $S = [0, 10] \times [0, 10] \times]-\infty, +\infty[$. There is in fact no particular constraint on the orientation of the vehicle, but multiple obstacles are imposed for the two first dimensions, they are represented in Figure 3.6 of Chapter 3. The input velocity v_0 can take the values in $\{-0.5, 0.5, 1.0\}$. The rear wheel orientation δ can take the values in $\{0.9, 0.6, 0.5, 0.3, 0.0, -0.3, -0.5, -0.6, -0.9\}$. The sampling period is $\tau = 0.3$.

A.7 Two-tank system

The two-tank system is a linear example taken from [89]. The system consists of two tanks and two valves. The first valve adds to the inflow of tank 1 and the second valve is a drain valve for tank 2. There is also a constant outflow from

tank 2 caused by a pump. The system is linearized at a desired operating point. The objective is to keep the water level in both tanks within limits using a discrete open/close switching strategy for the valves. Let the water level of tanks 1 and 2 be given by x_1 and x_2 respectively. The behavior of x_1 is given by $\dot{x}_1 = -x_1 - 2$ when the tank 1 valve is closed, and $\dot{x}_1 = -x_1 + 3$ when it is open. Likewise, x_2 is driven by $\dot{x}_2 = x_1$ when the tank 2 valve is closed and $\dot{x}_2 = x_1 - x_2 - 5$ when it is open. On this example, the Euler-based method works better than *DynIBEX* in terms of CPU time.

A.8 Helicopter

The helicopter is a linear example taken from [55]. The problem is to control a quadrotor helicopter toward a particular position on top of a stationary ground vehicle, while satisfying constraints on the relative velocity. Let g be the gravitational constant, x (reps. y) the position according to x -axis (resp. y -axis), \dot{x} (resp. \dot{y}) the velocity according to x -axis (resp. y -axis), ϕ the pitch command and ψ the roll command. The possible commands for the pitch and the roll are the following: $\phi, \psi \in \{-10, 0, 10\}$. Since each mode corresponds to a pair (ϕ, ψ) , there are nine switched modes. The dynamics of the system is given by the equation:

$$\dot{X} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} X + \begin{pmatrix} 0 \\ g \sin(-\phi) \\ 0 \\ g \sin(\psi) \end{pmatrix}$$

where $X = (x \ \dot{x} \ y \ \dot{y})^\top$. Since the variables x and y are decoupled in the equations and follow the same equations (up to the sign of the command), it suffices to study the control for x (the control for y is the opposite).

A.9 Eleven room house

This case study, proposed by the Danish company Seluxit, aims at controlling the temperature of an eleven rooms house, heated by geothermal energy. The *continuous* dynamics of the system is the following:

$$\frac{d}{dt} T_i(t) = \sum_{j=1}^n A_{i,j}^d (T_j(t) - T_i(t)) + B_i (T_{env}(t) - T_i(t)) + H_{i,j}^v \cdot v_j \quad (\text{A.5})$$

The temperatures of the rooms are the T_i . The matrix A^d contains the heat transfer coefficients between the rooms, matrix B contains the heat transfer coefficients between the rooms and the external temperature, set to $T_{env} = 10^\circ C$ for the computations. The control matrix H^v contains the effects of the control on the

room temperatures, and the control variable is here denoted by v_j . We have $v_j = 1$ (resp. $v_j = 0$) if the heater in room j is turned on (resp. turned off). We thus have $n = 11$ and $N = 2^{11} = 2048$ switching modes.

Note that the matrix A^d is parametrized by the open or closed state of the doors in the house. In our case, the average between closed and open matrices was taken for the computations. The exact values of the coefficients are given in [112]. The controller has to select which heater to turn on in the eleven rooms. Due to a limitation of the capacity supplied by the geothermal device, the 11 heaters cannot be turned on at the same time. In our case, we limit to 4 the number of heaters that can be on at the same time.

Appendix B

Proof of Lemma 1

Proof. Suppose $\ell_1 \leq \ell_2$, and denote by $P_{i_1}^1(k)$ the property

$$(f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k)))_1 \subseteq X_{i_1}^k$$

and by $P_{i_1}^2(k)$

$$X_{i_1}^k \subseteq R_1 + A + \varepsilon$$

and similarly for $P_{i_2}^1(k)$ and $P_{i_2}^2(k)$.

We show by induction on k the following property $P(k)$:

$$\forall i_1 \in I_1, P_{i_1}^1(k) \wedge P_{i_1}^2(k) \quad \text{and} \quad \forall i_2 \in I_2, P_{i_2}^1(k) \wedge P_{i_2}^2(k).$$

Let us first consider the case $k = 1$. Let us prove $\forall i_1 \in I_1, P_{i_1}^1(k) \wedge P_{i_1}^2(k)$ (the proof is similar for $\forall i_2 \in I_2, P_{i_2}^1(k) \wedge P_{i_2}^2(k)$). Let us show that $(f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k)))_1 \subseteq X_{i_1}^k$ and $X_{i_1}^k \subseteq R_1 + A + \varepsilon$.

For $k = 1$, $\pi_{i_1}^k$ and $\pi_{i_2}^k$ are of the form u_1 and u_2 . We have:

1. $(f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k)))_1 = f_1(r_{i_1} + a, R_2 + a, u_1)$
2. $X_{i_1}^1 = f_1(X_{i_1}^0, R_2 + A + \varepsilon, u_1) = f_1(r_{i_1} + a, R_2 + A + \varepsilon, u_1)$

Hence $(f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^k, \pi_{i_2}^k)))_1 \subseteq X_{i_1}^k$ holds for $k = 1$. And $X_{i_1}^k \subseteq R_1 + A + \varepsilon$ because of $Prop_1(A, i_1, \pi_{i_1})$.

Let us now suppose that $k > 1$ and that $P(k - 1)$ holds. We prove $P(k)$. Properties $P_{i_1}^2(k)$ and $P_{i_2}^2(k)$ are true for all i_1, i_2 because, by construction, the sequence $X_{i_1}^k$ (resp. $X_{i_2}^k$) satisfies $Prop_1(a, i_1, \pi_{i_1})$ (resp. $Prop_2(a, i_2, \pi_{i_2})$). Let us prove $P_{i_1}^1(k)$ and $P_{i_2}^1(k)$:

$$\begin{aligned} (f(r_{i_1} + A, R_2 + A, (\pi_{i_1}^k, \pi_{i_2}^k)))_1 &= (f(f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))), \\ &\quad (\pi_{i_1}(k), \pi_{i_2}(k)))_1 \\ &= f_1([f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))], \\ &\quad [f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))], \pi_{i_1}(k)). \end{aligned}$$

Note that the first argument of f_1 in the last expression satisfies $[f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))] \subseteq X_{i_1}^k$ by $P_{i_1}^1(k-1)$. Besides, the second argument satisfies $[f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))] \subseteq \bigcup_{j_2 \in I_2} X_{j_2}^{k-1} \subseteq R_2 + A + \varepsilon$, because

1. $r_{i_1} + A \subseteq R_1 + A$
2. $\bigcup_{j_2 \in I_2} X_{j_2}^{k-1} \subseteq R_2 + A + \varepsilon$ since $X_{j_2}^{k-1} \subseteq R_2 + A + \varepsilon$ (by $P_{j_2}^2(k-1)$ which holds for all j_2)
3. $[f((R_1 + A, r_{j_2} + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))] \subseteq X_{j_2}^{k-1}$ (by $P_{j_2}^1(k-1)$).

Hence

$$\begin{aligned} f_1([f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))], [f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{k-1}, \pi_{i_2}^{k-1}))], \pi_{i_1}^{(k)}) \\ \subseteq f_1(X_{i_1}^{k-1}, R_2 + A + \varepsilon, \pi_{i_1}(k)) = X_{i_1}^k \end{aligned}$$

We have thus proved $P_{i_1}^1(k)$:

$$(f(r_{i_1} + A, R_2 + A, (\pi_{i_1}^k, \pi_{i_2}^k)))_1 \subseteq X_{i_1}^k$$

This completes the proof of $\forall i_1 \in I_1, P_{i_1}^1(k) \wedge P_{i_1}^2(k)$. We prove $P_{i_2}^1(k) \wedge P_{i_2}^2(k)$ for all $i_2 \in I_2$ similarly, which concludes the proof of $P(k)$. The proof of $(f((r_{i_1} + A, R_2 + A), (\pi_{i_1}^{\ell_1}, \pi_{i_2}^{\ell_1})))_1 \subseteq X_{i_1}^{\ell_1}(a, \pi_{i_1}) \subseteq R_1$ is similar. \square

Bibliography

- [1] M. Abbaszadeh and H. Marquez. Nonlinear observer design for one-sided lipschitz systems. In *Proceedings of the American Control Conference (ACC)*, pages 799–806. IEEE, 2010.
- [2] M. Al Khatib, A. Girard, and T. Dang. Scheduling of embedded controllers under timing contracts. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 131–140. ACM, 2017.
- [3] A. Alessandri, M. Baglietto, and G. Battistelli. Luenberger observers for switching discrete-time linear systems. *International Journal of Control*, 80(12):1931–1943, 2007.
- [4] A. Alessandri and P. Coletta. Design of luenberger observers for a class of hybrid linear systems. In *Hybrid systems: computation and control*, pages 7–18. Springer, 2001.
- [5] J. Alexandre dit Sandretto and A. Chapoutot. Dynibex library. <http://perso.ensta-paristech.fr/chapoutot/dynibex/>, 2015.
- [6] J. Alexandre dit Sandretto and A. Chapoutot. Validated Solution of Initial Value Problem for Ordinary Differential Equations based on Explicit and Implicit Runge-Kutta Schemes. Research report, ENSTA ParisTech, 2015.
- [7] J. Alexandre dit Sandretto and A. Chapoutot. Validated explicit and implicit runge-kutta methods. *Reliable Computing*, 22:79–103, 2016.
- [8] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Hybrid Systems: Computation and Control*, pages 173–182, 2013.
- [9] M. Althoff. Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 173–182. ACM, 2013.
- [10] M. Althoff, O. Stursberg, and M. Buss. Verification of uncertain embedded systems by computing reachable sets based on zonotopes.
- [11] R. Alur and T. A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

- [12] R. Alur, S. Moarref, and U. Topcu. Pattern-based refinement of assume-guarantee specifications in reactive synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 501–516. Springer, 2015.
- [13] D. Angeli. A Lyapunov approach to incremental stability. In *Proc. of IEEE Conference on Decision and Control*, volume 3, pages 2947–2952, 2000.
- [14] D. Angeli. Further results on incremental input-to-state stability. *IEEE Transactions on Automatic Control*, 54(6):1386–1391, 2009.
- [15] A. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: an overview. *International Journal of Applied Mathematics and Computer Science*, 11(5):1093–1121, 2001.
- [16] A. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2000.
- [17] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88(7):1011–1025, 2000.
- [18] E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of nonlinear systems. *Acta Informatica*, 43(7):451–476, 2007.
- [19] K. J. Aström and R. M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.
- [20] K. E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2008.
- [21] J. A. Atwell and B. B. King. Reduced order controllers for spatially distributed systems via proper orthogonal decomposition. *SIAM Journal on Scientific Computing*, 26(1):128–151, 2004.
- [22] M. Azam, S. N. Singh, A. Iyer, and Y. Kakad. Nonlinear rotational maneuver and vibration damping of nasa scole system. *Acta Astronautica*, 32(3):211–220, 1994.
- [23] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied numerical mathematics*, 43(1-2):9–44, 2002.
- [24] M. Balde and P. Jouan. Geometry of the limit sets of linear switched systems. *SIAM J. Control Optim.*, 49(3):1048–1063, 2011.
- [25] G. Bastin and J.-M. Coron. *Stability and boundary stabilization of 1-d hyperbolic systems*, volume 88. Springer.
- [26] S. S. Bauer, A. David, R. Hennicker, K. G. Larsen, A. Legay, U. Nyman, and A. Wasowski. Moving from specifications to contracts in component-based design. In *Fase*, volume 7212, pages 43–58. Springer, 2012.

- [27] A. G. Beccuti, G. Papafotiou, and M. Morari. Optimal control of the boost DC-DC converter. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 4457–4462. IEEE, 2005.
- [28] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free galerkin methods. *International journal for numerical methods in engineering*, 37(2):229–256, 1994.
- [29] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale lyapunov equations, riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.
- [30] P. Benner and A. Schneider. Balanced truncation model order reduction for lti systems with many inputs or outputs. In *Proceedings of the 19th international symposium on mathematical theory of networks and systems–MTNS*, volume 5, 2010.
- [31] B. Besselink, N. van de Wouw, J. M. Scherpen, and H. Nijmeijer. Model reduction for nonlinear systems by incremental balanced truncation. *IEEE Transactions on Automatic Control*, 59(10):2739–2753, 2014.
- [32] W.-J. Beyn and J. Rieger. The implicit euler scheme for one-sided lipschitz differential inclusions. *Discr. and Cont. Dynamical Systems*, B(14):409–428, 1998.
- [33] S. K. Biswas and N. Ahmed. Optimal control of large space structures governed by a coupled system of ordinary and partial differential equations. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(1):1–18, 1989.
- [34] S. Bogomolov, G. Frehse, M. Greitschus, R. Grosu, C. Pasareanu, A. Podelski, and T. Strump. Assume-guarantee abstraction refinement meets hybrid systems. In *Haifa verification conference*, pages 116–131. Springer, 2014.
- [35] O. Bouissou, A. Chapoutot, and A. Djoudi. Enclosing temporal evolution of dynamical systems using numerical methods. In *NASA Formal Methods*, number 7871 in LNCS, pages 108–123. Springer, 2013.
- [36] O. Bouissou and M. Martel. GRKLib: a Guaranteed Runge Kutta Library. In *Scientific Computing, Computer Arithmetic and Validated Numerics*, 2006.
- [37] O. Bouissou, S. Mimram, and A. Chapoutot. HySon: Set-based simulation of hybrid systems. In *Rapid System Prototyping*. IEEE, 2012.
- [38] R. W. Brockett et al. Asymptotic stability and feedback stabilization. *Differential geometric control theory*, 27(1):181–191, 1983.
- [39] X. Cai, Z. Wang, and L. Liu. Control design for one-side lipschitz nonlinear differential inclusion systems with time-delay. *Neurocomput.*, 165(C):182–189, Oct. 2015.
- [40] G. Cain and G. H. Meyer. *Separation of variables for partial differential equations: an eigenfunction approach*. CRC Press, 2005.

- [41] K. Chatterjee and T. Henzinger. Assume-guarantee synthesis. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 261–275, 2007.
- [42] X. Chen, E. Abraham, and S. Sankaranarayanan. Taylor model flowpipe construction for non-linear hybrid systems. In *IEEE 33rd Real-Time Systems Symposium*, pages 183–192. IEEE Computer Society, 2012.
- [43] X. Chen, E. Ábrahám, and S. Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Computer Aided Verification*, pages 258–263. Springer, 2013.
- [44] X. Chen and S. Sankaranarayanan. Decomposed reachability analysis for non-linear systems. In *Proc. of IEEE Real-Time Systems Symposium*, pages 13–24, 2016.
- [45] F. Chinesta, A. Ammar, and E. Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational methods in Engineering*, 17(4):327–350, 2010.
- [46] F. Chinesta, P. Ladeveze, and E. Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, 2011.
- [47] W. Clark. Vibration control with state-switched piezoelectric materials. *Journal of intelligent material systems and structures*, 11(4):263–271, 2000.
- [48] L. Cordier and M. Bergmann. Proper Orthogonal Decomposition: an overview. In *Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Karman Institute for Fluid Dynamics, 2008.*, page 46 pages. VKI, 2008.
- [49] J.-M. Coron. *Control and nonlinearity*. Number 136. American Mathematical Soc., 2007.
- [50] J.-M. Coron and B. d’Andrea Novel. Stabilization of a rotating body beam without damping. *IEEE transactions on automatic control*, 43(5):608–618, 1998.
- [51] R. Curtain and H. Zwart. Stabilization of collocated systems by nonlinear boundary control. *Systems & control letters*, 96:11–14, 2016.
- [52] G. Dahlquist. Error analysis for a class of methods for stiff non-linear initial value problems. *Numerical analysis*, pages 60–72, 1976.
- [53] E. Dallal and P. Tabuada. On compositional symbolic controller synthesis inspired by small-gain theorems. In *Proc. of IEEE Conference on Decision and Control*, pages 6133–6138, 2015.
- [54] L. H. de Figueiredo and J. Stolfi. *Self-Validated Numerical Methods and Applications*. Brazilian Mathematics Colloquium monographs. IMPA/CNPq, 1997.

- [55] J. Ding, E. Li, H. Huang, and C. J. Tomlin. Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2160–2165. IEEE, 2011.
- [56] J. A. dit Sandretto and A. Chapoutot. Validated explicit and implicit runge-kutta methods. *Reliable Computing*, 22:79–103, 2016.
- [57] T. Donchev and E. Farkhi. Stability and euler approximation of one-sided lipschitz differential inclusions. *SIAM J. Control Optim.*, 36(2):780–796, 1998.
- [58] T. Donchev and E. Farkhi. Stability and euler approximation of one-sided lipschitz differential inclusions. *SIAM journal on control and optimization*, 36(2):780–796, 1998.
- [59] T. Dzektulić. Rigorous integration of non-linear ordinary differential equations in Chebyshev basis. *Numerical Algorithms*, 69(1):183–205, 2015.
- [60] J. L. Eftang, M. A. Grepl, and A. T. Patera. A posteriori error bounds for the empirical interpolation method. *Comptes Rendus Mathematique*, 348(9-10):575–579, 2010.
- [61] A. Eggers, M. Fränzle, and C. Herde. SAT modulo ODE: A direct SAT approach to hybrid systems. In *Automated Technology for Verification and Analysis*, volume 5311 of *LNCS*, pages 171–185. Springer, 2008.
- [62] H. O. Fattorini. *Infinite dimensional optimization and control theory*, volume 54. Cambridge University Press, 1999.
- [63] F. Formosa. Contrôle actif des vibrations des structures type poutre. *Private report*.
- [64] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *Computer Aided Verification*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.
- [65] L. Fribourg, U. Kühne, and N. Markey. Game-based Synthesis of Distributed Controllers for Sampled Switched Systems. In *2nd International Workshop on Synthesis of Complex Parameters (SynCoP’15)*, volume 44 of *OpenAccess Series in Informatics (OASIcs)*, pages 48–62, Dagstuhl, Germany, 2015.
- [66] L. Fribourg, U. Kühne, and R. Soulat. Finite controlled invariants for sampled switched systems. *Formal Methods in System Design*, 45(3):303–329, Dec. 2014.
- [67] L. Fribourg, U. Kühne, and R. Soulat. Finite controlled invariants for sampled switched systems. *Formal Methods in System Design*, 45(3):303–329, 2014.
- [68] L. Fribourg and R. Soulat. *Control of Switching Systems by Invariance Analysis: Application to Power Electronics*. Wiley-ISTE, July 2013. 144 pages.

- [69] L. Fribourg and R. Soulat. Stability controllers for sampled switched systems. In P. A. Abdulla and I. Potapov, editors, *Proceedings of the 7th Workshop on Reachability Problems in Computational Models (RP'13)*, volume 8169 of *Lecture Notes in Computer Science*, pages 135–145, Uppsala, Sweden, Sept. 2013. Springer.
- [70] P. K. Friz. Heat kernels, parabolic pdes and diffusion processes.
- [71] K. Gajda, M. Jankowska, A. Marciniak, and B. Szyszka. A survey of interval Runge–Kutta and multistep methods for solving the initial value problem. In *Parallel Processing and Applied Mathematics*, volume 4967 of *LNCS*, pages 1361–1371. Springer Berlin Heidelberg, 2008.
- [72] J. H. Gillula, G. M. Hoffmann, H. Huang, M. P. Vitus, and C. Tomlin. Applications of hybrid reachability analysis to robotic aerial vehicles. *The International Journal of Robotics Research*, page 0278364910387173, 2011.
- [73] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, pages 291–305, 2005.
- [74] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, pages 291–305. Springer, 2005.
- [75] A. Girard. Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 111–120. ACM, 2010.
- [76] A. Girard. Low-complexity switching controllers for safety using symbolic models. *IFAC Proceedings Volumes*, 45(9):82–87, 2012.
- [77] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *Automatic Control, IEEE Transactions on*, 55(1):116–126, Jan 2010.
- [78] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, 55(1):116–126, 2010.
- [79] R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems: modeling, stability, and robustness, 2012.
- [80] T. Grätsch and K.-J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & structures*, 83(4):235–265, 2005.
- [81] C. Gu. *Model order reduction of nonlinear dynamical systems*. PhD thesis, University of California, Berkeley, 2011.
- [82] R. Gunawan, E. L. Russell, and R. D. Braatz. Comparison of theoretical and computational characteristics of dimensionality reduction methods for large-scale uncertain systems. *Journal of Process Control*, 11(5):543–552, 2001.

- [83] N. W. Hagood and A. von Flotow. Damping of structural vibrations with piezoelectric materials and passive electrical networks. *Journal of Sound Vibration*, 146:243–268, Apr. 1991.
- [84] E. Hairer, S. P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, 2nd edition, 2009.
- [85] Z. Han and B. H. Krogh. Reachability analysis of hybrid systems using reduced-order models. In *American Control Conference*, pages 1183–1189. IEEE, 2004.
- [86] Z. Han and B. H. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 287–301. Springer Berlin Heidelberg, 2006.
- [87] F. M. Hante and M. Sigalotti. Converse lyapunov theorems for switched systems in banach and hilbert spaces. *SIAM Journal on Control and Optimization*, 49(2):752–770, 2011.
- [88] J. P. Hespanha, D. Liberzon, and A. R. Teel. Lyapunov conditions for input-to-state stability of impulsive systems. *Automatica*, 44(11):2735–2744, 2008.
- [89] I. A. Hiskens. Stability of limit cycles in hybrid systems. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. IEEE, 2001.
- [90] T. Horsin and P. I. Kogut. Optimal l₂-control problem in coefficients for a linear elliptic equation. *arXiv preprint arXiv:1306.2513*, 2013.
- [91] T. Horsin and P. I. Kogut. On unbounded optimal controls in coefficients for ill-posed elliptic dirichlet boundary value problems. *Asymptotic Analysis*, 98(1-2):155–188, 2016.
- [92] F. Immler. Verified reachability analysis of continuous systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 9035 of *LNCS*, pages 37–51. Springer, 2015.
- [93] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001.
- [94] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, 2001.
- [95] H. Ji, J. Qiu, H. Nie, and L. Cheng. Semi-active vibration control of an aircraft panel using synchronized switch damping method. *International Journal of Applied Electromagnetics and Mechanics*, 46(4), 2014.
- [96] Z.-P. Jiang, I. M. Mareels, and Y. Wang. A Lyapunov formulation of the nonlinear small-gain theorem for interconnected iss systems. *Automatica*, 32(8):1211–1215, 1996.

- [97] Z.-P. Jiang, A. R. Teel, and L. Praly. Small-gain theorem for ISS systems and applications. *Mathematics of Control, Signals, and Systems*, 7(2):95–120, 1994.
- [98] G. Kerschen, J.-C. Golinval, A. Vakakis, and L. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. *Nonlinear Dynamics*, 41(1-3):147–169, 2005.
- [99] M. A. Khatib, A. Girard, and T. Dang. Verification and synthesis of timing contracts for embedded controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC 2016, Vienna, Austria, April 12-14, 2016*, pages 115–124, 2016.
- [100] E. S. Kim, M. Arcak, and S. A. Seshia. Compositional controller synthesis for vehicular traffic networks. In *54th IEEE Conference on Decision and Control, CDC 2015, Osaka, Japan, December 15-18, 2015*, pages 6165–6171, 2015.
- [101] E. S. Kim, M. Arcak, and S. A. Seshia. Compositional controller synthesis for vehicular traffic networks. In *Proc. of IEEE Annual Conference on Decision and Control*, pages 6165–6171, 2015.
- [102] E. S. Kim, M. Arcak, and S. A. Seshia. A small gain theorem for parametric assume-guarantee contracts. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, pages 207–216. ACM, 2017.
- [103] E. S. Kim, M. Arcak, and S. A. Seshia. Symbolic control design for monotone systems with directed specifications. *Automatica*, 83:10 – 19, 2017.
- [104] M. Krstic, I. Kanellakopoulos, and P. Kokotovic. Nonlinear and adaptive control design, 1995. *John Wiley, New York*.
- [105] W. Kühn. Zonotope dynamics in numerical quality control. In *Mathematical Visualization*, pages 125–134. Springer, 1998.
- [106] U. Kühne and R. Soulat. Minimator 1.0. <https://bitbucket.org/ukuehne/minimator/overview>, 2015.
- [107] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische mathematik*, 90(1):117–148, 2001.
- [108] M. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Assume-guarantee verification for probabilistic systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 23–37. Springer, 2010.
- [109] A. Lalami. *Diagnostic et approches ensemblistes à base de zonotopes*. PhD thesis, Université de Cergy-Pontoise, 2008.
- [110] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International journal of robust and nonlinear control*, 12(6):519–535, 2002.

- [111] P.-O. Lamare, A. Girard, and C. Prieur. Switching rules for stabilization of linear systems of conservation laws. *SIAM Journal on Control and Optimization*, 53(3):1599–1624, 2015.
- [112] K. G. Larsen, M. Mikučionis, M. Muniz, J. Srba, and J. H. Taankvist. Online and compositional learning of controllers with application to floor heating. In *Tools and Algorithms for Construction and Analysis of Systems (TACAS), 22nd International Conference*, 2016.
- [113] J.-B. Lasserre, D. Henrion, C. Prieur, and E. Trélat. Nonlinear optimal control via occupation measures and lmi-relaxations. *SIAM Journal on Control and Optimization*, 47(4):1643–1666, 2008.
- [114] A. Le Coënt, J. Alexandre dit Sandretto, A. Chapoutot, F. De Vuyst, L. Chamoin, and L. Fribourg. Distributed control synthesis using euler’s method. In *Accepted for publication in International Workshop on Reachability Problems*, 2017.
- [115] A. Le Coënt, J. Alexandre dit Sandretto, A. Chapoutot, and L. Fribourg. Control of nonlinear switched systems based on validated simulation. In *Workshop on Symbolic and Numerical Methods for Reachability Analysis*, pages 1–6. IEEE, 2016.
- [116] A. Le Coënt, J. Alexandre dit Sandretto, A. Chapoutot, and L. Fribourg. An improved algorithm for the control synthesis of nonlinear sampled switched systems. In *To appear in Formal Methods in System Design*, 2017.
- [117] A. Le Coënt, F. de Vusyt, C. Rey, L. Chamoin, and L. Fribourg. Guaranteed control of switched control systems using model order reduction and state-space bisection. *Open Acces Series in Informatics*, 2015.
- [118] A. Le Coënt, F. De Vuyst, L. Chamoin, and L. Fribourg. Control synthesis of nonlinear sampled switched systems using Euler’s method. In *Proc. of International Workshop on Symbolic and Numerical Methods for Reachability Analysis*, volume 247 of *EPTCS*, pages 18–33. Open Publishing Association, 2017.
- [119] A. Le Coënt, F. de Vuyst, C. Rey, L. Chamoin, and L. Fribourg. Control of mechanical systems using set based methods. *International Journal of Dynamics and Control*, pages 1–17, 2016.
- [120] A. Le Coënt, L. Fribourg, N. Markey, F. de Vuyst, and L. Chamoin. Distributed synthesis of state-dependent switching control. In *Workshop on Reachability Problems*, volume 9899 of *LNCS*, pages 119–133. Springer, 2016.
- [121] A. Le Coënt, L. Fribourg, N. Markey, F. De Vuyst, and L. Chamoin. Distributed synthesis of state-dependent switching control. In *Submitted to Theoretical Computer Science*, 2017.

- [122] E. Le Corronc, A. Girard, and G. Goessler. Mode sequences as symbolic states in abstractions of incrementally stable switched systems. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3225–3230. IEEE, 2013.
- [123] F. Lempio. Set-valued interpolation, differential inclusions, and sensitivity in optimization. In *Recent Developments in Well-Posed Variational Problems*, pages 137–169. Kluwer Academic Publishers, 1995.
- [124] D. Liberzon. *Switching in systems and control*. Springer, 2012.
- [125] Y. Lin and M. A. Stadtherr. Validated solutions of initial value problems for parametric odes. *Appl. Numer. Math.*, 57(10):1145–1162, 2007.
- [126] J. Liu, N. Ozay, U. Topcu, and R. M. Murray. Synthesis of reactive switching protocols from temporal logic specifications. *Automatic Control, IEEE Transactions on*, 58(7):1771–1785, 2013.
- [127] R. J. Lohner. Enclosing the solutions of ordinary initial and boundary value problems. *Computer Arithmetic*, pages 255–286, 1987.
- [128] Q. Lu and E. Zuazua. Robust null controllability for heat equations with unknown switching control mode. 2014.
- [129] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. Pau. A general, multipurpose interpolation procedure: the magic points. 2007.
- [130] K. Makino and M. Berz. Rigorous integration of flows and odes using taylor models. In *Proceedings of the 2009 Conference on Symbolic Numeric Computation*, SNC ’09, pages 79–84, New York, USA, 2009. ACM.
- [131] S. Marx and E. Cerpa. Output feedback stabilization of the korteweg-de vries equation. *arXiv preprint arXiv:1609.06096*, 2016.
- [132] M. Mazo, A. Davitian, and P. Tabuada. *PESSOA: A Tool for Embedded Controller Synthesis*, volume 6174 of *LNCS*, pages 566–569. Springer, 2010.
- [133] G. Menon. Lectures on partial differential equations. 2005.
- [134] P.-J. Meyer. *Invariance and symbolic control of cooperative systems for temperature regulation in intelligent buildings*. Theses, Université Grenoble Alpes, Sept. 2015.
- [135] P.-J. Meyer, A. Girard, and E. Witrant. Safety control with performance guarantees of cooperative systems using compositional abstractions. *IFAC-PapersOnLine*, 48(27):317–322, 2015.
- [136] P.-J. Meyer, A. Girard, and E. Witrant. Robust controlled invariance for monotone systems: application to ventilation regulation in buildings. *Automatica*, 70:14–20, 2016.
- [137] P.-J. Meyer, H. Nazarpour, A. Girard, and E. Witrant. Experimental implementation of UFAD regulation based on robust controlled invariance. In *European Control Conference*, pages 1468–1473, 2014.

- [138] M. Mohamed and S. Swaroop. Incremental input to state stability of under-water vehicle. *IFAC-PapersOnLine*, 49(1):41–46, 2016.
- [139] S. R. Moheimani and A. Fleming. *Piezoelectric transducers for vibration control and damping*. Springer Science & Business Media, 2006.
- [140] B. C. Moore. Principal component analysis in linear systems: Controllability, observability and model reduction. *IEEE Transaction on Automatic Control*, 26(1), 1981.
- [141] R. Moore. *Interval Analysis*. Prentice Hall, 1966.
- [142] S. Mouelhi, A. Girard, and G. Gössler. Cosyma: a tool for controller synthesis using multi-scale abstractions. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 83–88. ACM, 2013.
- [143] W. Nam and R. Alur. Learning-based symbolic assume-guarantee reasoning with automatic decomposition. In *International Symposium on Automated Technology for Verification and Analysis*, pages 170–185. Springer, 2006.
- [144] N. S. Nedialkov, K. Jackson R., and G. F. Corliss. Validated solutions of initial value problems for ordinary differential equations. *Appl. Math. and Comp.*, 105(1):21 – 68, 1999.
- [145] R. H. Nochetto, G. Savaré, and C. Verdi. A posteriori error estimates for variable time-step discretizations of nonlinear evolution equations. *Communications on Pure and Applied Mathematics*, 53(5):525–589, 2000.
- [146] R. Nong and D. C. Sorensen. A parameter free adi-like method for the numerical solution of large scale lyapunov equations. 2009.
- [147] B. N. Parlett. The rayleigh quotient iteration and some generalizations for nonnormal matrices. *Mathematics of Computation*, 28(127):679–693, 1974.
- [148] J. D. Pintér. *Global optimization in action: continuous and Lipschitz optimization: algorithms, implementations and applications*, volume 6. Springer Science & Business Media, 2013.
- [149] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.
- [150] C. Prieur, A. Girard, and E. Witrant. Stability of switched linear hyperbolic systems by lyapunov techniques. *IEEE Transactions on Automatic control*, 59(8):2196–2202, 2014.
- [151] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Number CMCS-BOOK-2009-019. Oxford University Press, 1999.
- [152] A. Ramaratnam, N. Jalili, and D. M. Dawson. Semi-active vibration control using piezoelectric-based switched stiffness. In *American Control Conference, 2004. Proceedings of the 2004*, volume 6, pages 5461–5466. IEEE, 2004.

- [153] H. Ravanbakhsh and S. Sankaranarayanan. Robust controller synthesis of switched systems using counterexample guided framework. In *Embedded Software (EMSOFT), 2016 International Conference on*, pages 1–10. IEEE, 2016.
- [154] G. Reissig, A. Weber, and M. Runger. Feedback refinement relations for the synthesis of symbolic controllers. *arXiv preprint arXiv:1503.03715*, 2015.
- [155] P. Riedinger, M. Sigalotti, and J. Daafouz. On the algebraic characterization of invariant sets of switched linear systems. *Automatica J. IFAC*, 46(6):1047–1052, 2010.
- [156] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [157] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using pod and galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1):115–129, 2004.
- [158] J. Rubio. *Control and Optimization: The Linear Treatment of Nonlinear Problems*. Nonlinear science : theory and applications. Manchester University Press, 1986.
- [159] M. Runger and M. Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pages 99–104. ACM, 2016.
- [160] A. A. H. Salah, T. Garna, J. Ragot, and H. Messaoud. Synthesis of a robust controller with reduced dimension by the loop shaping design procedure and decomposition based on laguerre functions. *Transactions of the Institute of Measurement and Control*, page 0142331215583101, 2015.
- [161] A. L. Sangiovanni-Vincentelli, W. Damm, and R. Passerone. Taming dr. frankenstein: Contract-based design for cyber-physical systems. *Eur. J. Control*, 18(3):217–238, 2012.
- [162] S. Sargsyan, S. L. Brunton, and J. N. Kutz. Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries. *Physical Review E*, 92(3):033304, 2015.
- [163] U. Serres, J.-C. Vivalda, and P. Riedinger. On the convergence of linear switched systems. *IEEE Transactions on Automatic Control*, 56(2):320–332, 2011.
- [164] S. M. Shahruz. Boundary control of the axially moving kirchhoff string. *Automatica*, 34(10):1273–1277, 1998.
- [165] H. R. Shaker and R. Wisniewski. Model reduction of switched systems based on switching generalized gramians. *International Journal of Innovative Computing, Information and Control*, 8(7):5025–5044, 2012.
- [166] R. E. Showalter. *Monotone operators in Banach space and nonlinear partial differential equations*, volume 49. American Mathematical Soc., 2013.

- [167] S. W. Smith, P. Nilsson, and N. Ozay. Interdependence quantification for compositional control synthesis with an application in vehicle safety systems. In *55th IEEE Conference on Decision and Control, CDC 2016, Las Vegas, NV, USA, December 12-14, 2016*, pages 5700–5707, 2016.
- [168] G. Söderlind. On nonlinear difference and differential equations. *BIT Numerical Mathematics*, 24(4):667–680, 1984.
- [169] R. Soulat. *Synthesis of correct-by-design schedulers for hybrid systems*. PhD thesis, 2014. Thèse de doctorat en Informatique, dirigée par Laurent Fribourg, Ecole normale supérieure de Cachan 2014.
- [170] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.
- [171] E. Trélat. Optimal control and applications to aerospace: some results and challenges. *Journal of Optimization Theory and Applications*, 154(3):713–758, 2012.
- [172] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, pages 2323–2330, 2002.
- [173] G. Yang and D. Liberzon. A lyapunov-based small-gain theorem for interconnected switched systems. *Systems & Control Letters*, 78:47–54, 2015.
- [174] H. Yang, J. Bin, and V. Cocquempot. *Stabilization of switched nonlinear systems with unstable modes*, volume 9. Springer, 2014.
- [175] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.
- [176] M. Zeitz. The extended luenberger observer for nonlinear systems. *Systems & Control Letters*, 9(2):149–156, 1987.
- [177] Y. Zhao, J. Kim, and M. Filippone. Aggregation algorithm towards large-scale boolean network analysis. *Automatic Control, IEEE Transactions on*, 58(8):1976–1985, 2013.
- [178] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *International journal for numerical methods in engineering*, 24(2):337–357, 1987.