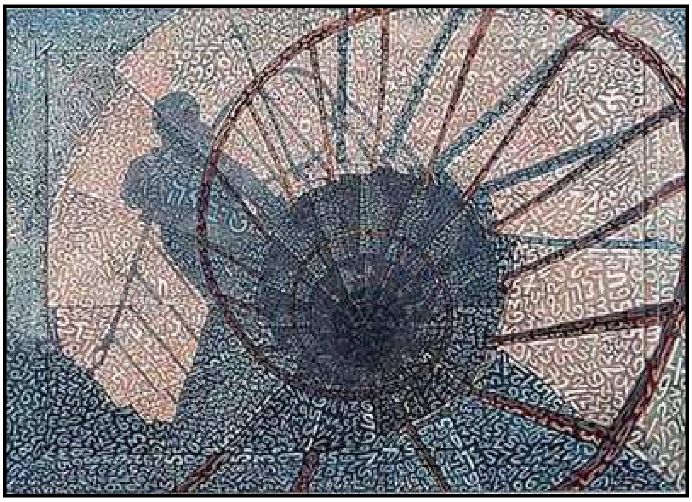
UNIVERSITÀ DEGLI STUDI DI CAGLIARI

DIPARTIMENTO DI INGEGNERIA ELETTRICA ED ELETTRONICA

Tesi Dottorato di Ricerca – Ph.D. Thesis

OPTIMAL CONTROL OF LINEAR AFFINE HYBRID AUTOMATA

Daniele Corona



Responsabile scientifico – Scientific Advisor
Prof. Alessandro Giua
Dipartimento di Ingegneria Elettrica ed Elettronica
Facoltà di Ingegneria
(Cagliari, Italia)

Febbraio 2005

Image: Verso dentro, Tobia Ravà

To all nature

Contents

Abstract

A recent research area in the field of Computer Science and Control Technology deals extensively with the study of optimal control and stability of a new class of systems, called *hybrid systems*. These systems are characterized by the co-existence of continuous time dynamics, and discrete events dynamics. This new class of system, obtained by the combination of the two types of dynamics, requires novel methodologies and approaches, only partially related to those developed for the original classes. We consider a particular subclass of hybrid systems, the *linear affine hybrid automata*, where the *continuous* behavior is governed by *linear affine differential equations*, and the *discrete* behavior is determined by the *firing* of arcs in an oriented graph. Constraints on the state space, *guards* and *invariants*, are also considered. For this class of systems we aim to solve an infinite time horizon optimal control problem, that quadratically weights the continuous state and associates a cost to the occurrence of every switch. The decision variables are the switching instants and the sequence of operating modes. We *initially* assume that the switching sequence has a *finite* length. We propose a numerical procedure, namely the *switching table procedure*, inspired by dynamic programming arguments, that identifies the regions of the state space where an optimal switch should occur. The main advantages of this procedure are the following: it provides the global minimum of the optimization problem, it performs calculations off line and it provides a feedback solution. Its main disadvantage is the requirement of the discretization of the state space, that turns into long computational times and memory occupancy. The method is then extended to the case of an *infinite* number of switches. We prove that the switching tables converge to the same one when the number of allowed switches increases. The methodology is successfully applied to the design of a semiactive suspension system of a quarter-car, where each linear dynamics corresponds to a given value of the damping coefficient f. Finally we show how the *switching table procedure* can also be used to design a stabilizing switching law for a particular hybrid automaton, the *switched system*. In this case we consider switched systems composed of *linear time invariant non Hurwitz* dynamics and we apply the procedure to a system *augmented* with a stable dynamics. If the system with unstable modes is globally exponentially stabilizable, then the method is guaranteed to provide the stabilizing feedback control law that in addition minimizes the chosen quadratic performance index. Specific examples are offered throughout the dissertation.

Keywords: control theory, hybrid systems, switched systems, optimal control, stability and stabilizability.

Acknowledgments

Primarily I would like to express my deepest sense of gratitude to Professor Alessandro Giua, my Ph. D. advisor, for having dedicated to me his constant guidance, profound knowledge and experience throughout my doctoral studies. I surely recognize that his patient assistance and his brilliant approaches in all scientific problems have been indispensable for my education as a science researcher. I consider myself privileged for working with him. In addition I avail myself of this opportunity to thank him for his wise advice in facing personal difficulties that occurred during this term.

Other special thanks are devoted to my colleague, but firstly friend, Research Assistant Carla Seatzu, who contributed in my educational growth with her stimulating ideas and perspectives in many theoretical and applicative problems. With Carla I shared the constant pleasure of scientific as well as leisure time discussions.

I would also like to thank Professor Jörg Raisch, of the Lehrstuhl für Systemtheorie technischer Prozesse, University of Magdeburg, Germany, where I have been visiting for a common research project, for his valuable guidance and assistance in the challenging purpose of integrating our different methodologies. On this occasion I also met my colleagues Dmitri Gromov and Eckart Mayer, now good friends, who shared with me various moments of joy and pain in the tortuous trail of the research activity. I expect you all in Sardinia!

Moreover I am grateful to Professor Alberto Bemporad, University of Siena, Italy, with whom I have the honor of being co-author in two publications, for his many useful suggestions and for accepting to be a committee member for my Ph. D. dissertation.

Another important acknowledgement is for Dr. John Wade, Education Department, University of Cagliari, Italy, for his aid in the linguistic aspect of this thesis. I also express to John my personal congratulations for his recent successes.

The most important gratitude is to my family, my parents Luisa and Lorenzo and my beloved sister Gabriella, who supported and encouraged my work and constantly provided me with their love and their suggestions. A warm hug is for Carolina, the *eye-master*, who was so patient in recomposing the jigsaw puzzle, fallen to pieces after a storm. Special thanks are to my friends Roberto, *Jazz*, whose *Weltanschauung* was really fruitful for my professional and private growth, to Francesca, *Selva*, for her precious aid in my darkest moments, to Alessandra, *Zia*, for giving me pearls of sincere hilarity and to Maria, for bringing me continuous inspiration. I will never forget the magic hours of conversation, between dream and reality, with all of you.

Cagliari, February 2005.

List of publications

The contents of this thesis are based on the following references, submitted and published in international journals or in conference proceedings.

Journals

1. D. Corona, A. Giua, and C. Seatzu, *Optimal control of hybrid automata: an application to the design of a semiactive suspension*, Control Engineering Practice **12** (2004), 1305-1318.
2. A. Bemporad, D. Corona, A. Giua, and C. Seatzu, *Optimal control of continuous-time switched affine systems*, IEEE Transaction Automatic Control (2004), Conditionally accepted as a regular paper.
3. D. Corona, A. Giua, and C. Seatzu, *Stabilization of switched systems via optimal control*, Automatica (2005), Submitted as a brief paper.

Conference proceedings

1. A. Bemporad, D. Corona, A. Giua, and C. Seatzu, *Optimal state-feedback quadratic regulation of linear hybrid automata*, Proceedings IFAC Conference on Analysis and Design of Hybrid Systems (St. Malò, France), 2003, pp. 407-412.
2. D. Corona, A. Giua, and C. Seatzu, *Optimal feedback switching laws for autonomous hybrid automata*, Proceedings IEEE International Symposium on Intelligent Control (Taipei, Taiwan), September 2004, pp. 31-36.
3. D. Corona, A. Giua, C. Seatzu, D. Gromov, E. Mayer, and J. Raisch, *Optimal hybrid control for switched linear systems under safety and liveness constraints*, Proceedings IEEE Symposium on Computer Aided Control System Design (Taipei, Taiwan), September 2004, pp. 35-40.
4. D. Corona, A. Giua, and C. Seatzu, *Stabilization of switched systems via optimal control*, Proceedings IFAC World Congress (Prague, The Czech Republic), 2005, To appear.
5. D. Corona, A. Giua, C. Seatzu, D. Gromov, E. Mayer, and J. Raisch, *Optimal control of discrete time hybrid automata under safety and liveness constraints*, Proceedings IEEE International Symposium on Intelligent Control (Limassol, Cyprus), June 2005, Submitted.

1

Introduction

A *Hybrid System* is a *macro-system* generally characterized by the coexistence of two different kinds of dynamics, namely *event driven* dynamics and *time driven* dynamics [47]. The global behavior is hence determined by the *occurrence of events*, that are particularly suited to model, as an example, logical changes or inputs of a physical system, interlaced with *continuous functions of time*, that, in the majority of the cases of interest, are expressed by continuous/discrete time differential equations.

We may observe that hybrid systems are the mathematical expressions of the empirical phenomenon that is usually called *hybrid process*. Notably our lives are surrounded by hybrid processes, e.g., the controlled heating of an oven, the gear shift in a car [90], a hard drives motor, to cite a few. What is more, we may not stand indifferent to the fact that several biological processes can be classified as *hybrid*, in the sense described above. Think, as a trivial example, of the *eye blinking* that *resets* the humidity of the *cornea* (discrete event), subject to the continuous dehydration in contact with the atmosphere (time driven dynamics), or to the *cardiovascular* system and so on.

1.1 Motivation and general description

In the last decades hybrid systems have received significant amounts of intellectual efforts from scientists and researchers in the Computer Science and Control Technology fields. The reason for this interest from the scientific and industrial community may be related to the remarkable modelling power of this new class of systems. The fee of this advantage is normally paid in terms of complexity of the algorithms that attempt to govern, synthesize or to analyze these models. Nevertheless the latter, apart from its challenging aspect, is partially counterbalanced by the increasing computational velocity of modern calculators and the capacity of data storing.

The great variety of real and theoretical situations brought the researchers to tailor and solve a series of specific problems [118]. As a result the literature on hybrid systems appears various and complex, both in the modelling and designing aspects.

Nowadays one of the most considered *subclass* of hybrid systems is the *hybrid automaton*, and in particular one specific case, the *switched system*.

These models have been considered in this thesis, and for this particular class of hybrid systems we want to solve an optimal control problem with piecewise linear quadratic performance index, as subsequently described.

As an important extension we observe and formally prove that the solution of the optimal control problem considered in this research is *also* a solution for the stabilization of a switched system. This will be the main content of Chapter 7.

The problems of optimal control and stability of switched system are one of the most studied issues in the field of hybrid systems. With this thesis we would like to present our contribution by taking into account a model, consistent with the majority of the current literature, restricted to the linear time invariant particular case.

The work is organized as follows: Chapter 2 contains a brief updated literature review on the optimal control and stability of hybrid systems, and in particular of switched systems. We tried therein to analyze the work of the most active schools in the field of optimal control and the most currently used methodologies in the field of stability and stabilizability.

Chapter 3 is dedicated to the model description and problem formulation. It is divided into 2 parts:

1. The model *switched system* and its annexed optimal control problem. This is the model description and problem formulation for the synthesis of the control law that will be derived in Chapters 4, 6 and 7.
2. The model *hybrid automaton* and the annexed optimal control problem, whose solution will be provided in Chapter 5.

Furthermore the first part of Chapter 3 is divided into two parts:

1. The problem description with a *finite* number of switches, that will be tackled and solved in Chapter 4 for a switched system and Chapter 5 for a hybrid automaton.
2. The problem description with *infinite* number of switches that will be solved in Chapter 6 for switched systems and applied to the design of optimally stabilizing switching signals for switched systems in Chapter 7.

The model *hybrid automaton*, and the subclass *switched system*, are particular hybrid systems where the continuous evolution is governed by first-order vectorial differential equation of the form $\dot{\boldsymbol{x}}(t) = f_{i(t)}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$, where $\boldsymbol{x} \in \mathbb{R}^n$ represents the *continuous* part of the system state, and $\boldsymbol{u}(t)$ is an external *continuous* control input.

The subscript $i(t)$ is a function that indicates the current active mode at time t and its value represents the discrete event part of the system state. In the hybrid automaton framework $i(t)$ is a *piecewise constant* function that takes values from a finite and countable set of indexes of *locations*. In other words when it holds $i(t) = i$ the continuous part of the hybrid state $\boldsymbol{x}$ evolves according to the current dynamics f_i associated with location i.

The piecewise constant function $i(t)$ has (under special conditions) a countable number of discontinuities in time instants τ, namely $\mathcal{T} \equiv \{\tau_1, \tau_2, \ldots, \tau_k, \ldots\}$ that are called *switching instants*. When $t = \tau_k$ a *switch* occurs and the time driven evolution continues with dynamics f_{i_k} associated with the new location. We also call the *switching sequence*, henceforth $\mathcal{I} \equiv \{i_0, i_1, \ldots, i_k, \ldots\}$, the list of values taken by the function $i(t)$ in the time intervals defined by the switching instants.

In the considered model the occurrence of a switch may provoke a *state space resetting*, thus whenever there occurs a switch the evolution continues from a *new* initial state, that may be related to the state reached during the previous time driven evolution.

The hybrid automata allow also the modelling of some constraints that have a relevant practical interest.

Constraint 1. The switching sequence is subject to logical constraints. This means that from the current mode i not all other modes can be reached with a single switch. This may be described by an oriented graph where to each node (*location*) is associated a *dynamics*, and to each arc (*edge*) a *switching path*. This is very common in many physical applications where the switching path is constrained by constructive or safety specifications.

Constraint 2. Once entered in a location i we cannot leave it before a time $\delta_{\min}(i)$ has elapsed. This is a common constraint in many real applications: $\delta_{\min}$ may be the time necessary to control an actuator, or it may be the scan time of a PLC that triggers the switches.

Constraint 3. The value of the current continuous state space may influence the switching action. In other words, in a hybrid automaton, it is possible to define subsets of the domain of $\boldsymbol{x}$ where some switches are allowed or forbidden or restricted or forced. This is done by the introduction of the notion of *guards* and *invariants*. Broadly speaking these are continuous subsets of $\mathbb{R}^n$ where the switching strategy is conditioned. This notion is useful in the modelling of *safety and specification* constraints and in general those situations where the crossing of specific thresholds provokes changes in the dynamics[1]. This particular class of problems will be considered in Chapter 5.

If the *Constraint 3* is applied we refer explicitly to *hybrid automaton*, otherwise we will simply refer to the model as *switched system*. In this sense the switched system is a particular hybrid automaton without state space constraints that may condition the switching behavior. Chapters 4, 6 and 7 are devoted to switched systems, while in Chapter 5 we consider the more general hybrid automaton.

In this thesis we focus the attention on the hybrid automaton characterized by the following extra restrictions.

1. We consider these dynamics *autonomous*, meaning that there is no control input $\boldsymbol{u}(t)$. The only control action that we can design is the function $i(t)$, that includes the design of the *switching instants* and the *switching sequence*.
2. The dynamics of the system are all *linear and time invariants*, LTI, i.e., $\dot{\boldsymbol{x}} = \boldsymbol{A}_i\boldsymbol{x}$. This class also includes the case of *affine* systems, $\dot{\boldsymbol{x}} = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{f}_i$, that can be reduced via an appropriate technique, described in Section 4.2.4, to a LTI of the form $\dot{\boldsymbol{x}} = \tilde{\boldsymbol{A}}_i\boldsymbol{x}$. The LTI hypothesis significantly reduces the complexity of the model because many results on the traditional system theory may be used. Furthermore in many cases of practical relevance the linear model provides a satisfactory approximation of reality.
3. We assume that the state resetting, namely *state jumps*, is linear, i.e., at the occurrence of a switch the new value of the state space $\boldsymbol{x}^+ = \boldsymbol{M}\boldsymbol{x}^-$, where $\boldsymbol{M}$ is a constant matrix associated to the edge of the automaton, and the superscript $+$ and $-$ denote respectively the value after and before the switch.

For the hybrid automaton described above we want to design the function $i(t)$ that minimizes a linear quadratic performance index in infinite time horizon. The analyzed optimal control problem will be described in detail in Chapter 3.

We associate to each location i a semi-definite positive matrix $\boldsymbol{Q}_i$ that *weights* the continuous state space $\boldsymbol{x}(t)$ quadratically, and to each edge a positive constant H that *weights* the event driven evolution, i.e., a cost is associated to every switch. More specifically, yet not as rigorously as in Chapter 3, for any given initial point $(\boldsymbol{x}_0, i_0)$ we would like to design the function $i(t)$ in order that a performance index of the form

$$J_N(\boldsymbol{x}_0, i_0) = \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k=1}^{N} H_k$$

is minimized under the given constraints imposed by the hybrid automaton.

[1] As a trivial example consider a circuit containing a diode where the voltage threshold $x_1(t) < 0$ denotes the condition where the diode behaves as an open circuit.

Note that the equation above includes two terms: an integral, modelling the cost of the time driven evolution, and a sum modelling the cost of the event driven evolution.

The number N is crucial in this research. It represents the total number of available switches, i.e., $N + 1$ is the *maximum* number of branches of the piecewise function $i(t)$. In Chapters 4 and 5 we propose a solution of the described problem with finite N for switched systems and hybrid automata respectively. In this case a sufficient condition that permits us to obtain a finite cost is that at least one dynamics of the automaton is Hurwitz, i.e., all its eigenvalues are in the negative complex plane.

In Chapters 6 and 7 we relax this additional constraint and we allow N to increase indefinitely. This extension is extremely significant, especially from the viewpoint of stabilizability of switched systems. In fact it is well known that there exist stabilizing switching signals even for switched systems whose dynamics are unstable. Therefore the structural assumption of at least one Hurwitz dynamics can be relaxed. Furthermore this possibility is relevant in many applications, like those where the *steady state* of a set of variables is reached and maintained via an indefinite number of commutations. In some of these cases (in $\mathbb{R}^2$) the function $i(t)$ may become periodical under given conditions on the vector fields.

The procedure that solves the problems defined in Chapter 3, is the main contribution of this thesis. This is built on the results presented in [49, 9] extended to the classes of constrained systems described above, and to an infinite number of switches.

In particular we propose an *off line* procedure, namely the *switching tables procedure*, STP, that allows a *state feedback* control technique based on the construction of an appropriate set of *switching tables*. Here by *off line* we mean that the procedure is not developed *real time*. Once all tables are constructed off line, the real time implementation is achievable because no further calculations are needed.

Moreover we use the term *state feedback* to stress the fact that our procedure generates a *closed loop* control law, in opposition to most current results on optimal control of hybrid systems that only provide *open loop* solutions, i.e., dependent on the initial conditions. This is notable, because a closed loop control law has several advantages over an open loop one, including the fact that it is *robust* against external or measurements disturbance. In addition, as remarked above, it does not require any on line calculations, hence it is faster and implementable on *real time systems*.

A switching table is a *partition* of the state space into different regions where a specific mode must be *active*. If these tables are appropriately used, i.e., if the switches are performed consistently with the partitions, the evolution of the hybrid automaton, for any given initial state, is *the* one that minimizes the performance index described above.

For each location i of the hybrid automaton we provide a specific set of tables, meaning that whenever the continuous state is evolving in location i the controller must use the tables constructed for location i.

The switching table procedure is developed in Chapter 4 for a finite number of switches and for a switched system characterized by the *Constraints* 1 and 2. In Chapter 5 we extend the procedure by the addition of *Constraint* 3.

The tables are obtained by induction on the number $k \leq N$ of remaining switches. That is to say that, for each mode $i = 1, \ldots, s$ of the hybrid automaton, we first construct the table when only one switch is available, $\mathcal{C}_1^i$, then, using the data stored in $\mathcal{C}_1^i$, we proceed backwards and build the table $\mathcal{C}_2^i$, when 2 switches are available. Generalizing, given the table $\mathcal{C}_{k-1}^i$, we can construct the table $\mathcal{C}_k^i$. This is repeated until $k = N$.

Finally we obtain a battery of $N \times s$ tables $\mathcal{C}_k^i$, $k = 1, \ldots, N$, $i = 1, \ldots, s$ that allows us to perform a state feedback control law that minimizes the given performance index.

The tables are then used as follows. During an arbitrary evolution, the controller observes the current *hybrid state* $(\boldsymbol{x}, i)$ at each instant $t > 0$, and the number of remaining switches k (initially $k = N$). With this information the controller checks the table $\mathcal{C}_k^i$ in the point $\boldsymbol{x}$ to know wether a switch should occur and eventually to what location. If a switch *occurs* to location j the controller will now pass to check the table $\mathcal{C}_{k-1}^j$ during the evolution in mode j. If a switch *does not occur* the evolution continues in location i until $\boldsymbol{x}(t)$ crosses a switching manifold or it reaches the origin. This is repeated until $k = 0$ *or* $\boldsymbol{x} = \mathbf{0}$.

As pointed out above the procedure is an inductive methodology based on dynamic programming arguments. The main idea is that for every point $\boldsymbol{x}$ of the state space and for every location i of the hybrid automaton, we calculate a function $T_k^*(\boldsymbol{x}, i)$, that represents the *optimal residual cost* of an evolution starting from point $(\boldsymbol{x}, i)$ and performing k switches.

Knowing this, when $k + 1$ switches are available, the function $T_{k+1}^*(\cdot)$ can be obtained by minimizing a function that reaches a certain point $(\boldsymbol{x}, i)$ and henceforth exploits an already optimal solution, given by the previously calculated $T_k^*(\boldsymbol{x}, i)$.

This strategy, universally accepted as the *dynamic programming principle*, introduced by Bellman in [3], as several advantages that have been briefly described before.

To our concern we point out that it allows the computational feasibility of the procedure. In fact, as will be proved in Chapter 4, it permits us to convert the solution of *one* MIQP (mixed integer quadratic programming) problem of $N + s$ variables, i.e., the switching instants and the possible modes, into Ns problems of 1 variable. This is in general a significant aspect.

The main *advantages* of the proposed procedure may be briefly summarized as follows:

- it is guaranteed to find the optimal solution under the given constraints;
- it has an affordable computational complexity of order that grows linearly with N, the number of available switches, and quadratically with s, the number of possible modes of the hybrid automaton;
- it provides a *global* closed-loop solution, i.e., the tables may be used to determine the optimal state feedback law for all initial states.
- it performs calculations off line.

The main *disadvantage* of the procedure is that it requires a state space discretization. This problem is partially avoided for low dimensions of the state space $n = 2, 3, 4$, because under certain conditions we may limit to discretize the unitary semisphere. As a consequence this procedure, although theoretically efficient, is practically unaffordable for higher than $n = 4$ dimensions. In fact the number of points of the discretization grows exponentially with n and consequently the computational time *and* the memory occupancy.

1.2 Structure of the work

After the introduction we provide, in Chapter 2, a bibliographic survey. We will try therein to describe the state of art and the recent results on the topic of optimal control and stability of hybrid systems.

In Chapter 3 we formally define the considered class of system, i.e., hybrid automata, and the particular subclasses described previously, i.e., switched system, constrained hybrid automata and autonomous hybrid automata. We define also the dynamical behavior, and we describe in detail all the elements that characterize the model. In parallel we give the problem formulation, i.e., an optimal control problem characterized by a linear quadratic performance index, studied for the considered subclass, that is tackled and solved in the successive chapters.

In Chapter 4 we describe the switching table procedure, and we formally prove, by means of dynamic programming arguments that it allows to solve the optimal control problem for a switched system in feedback form. In this chapter we consider a problem that limits the number of switches to be finite. We also introduce the notion of *lexicographic ordering* that allows the uniqueness of the switching tables and we show that under particular conditions the switching regions are *homogeneous*. Hints on the computational complexity are provided for the fundamental algorithm of the STP presented therein. Lastly, specific numerical and physical examples are presented.

In Chapter 5 we extend the STP to the hybrid automaton, namely we show that the STP is still valid whenever two types of constraints are considered:

1. The system may perform autonomous switches (also called *internally forced* in [122]), i.e., not all switches are controllable but some of them may occur autonomously if $\boldsymbol{x}$ enters given regions of the state space. We call this automaton AHA (*Autonomous Hybrid Automaton*).
2. The degree of freedom (DOF) of the controller is restricted according to the value of the state space. We call this automaton CHA (*Constrained Hybrid Automaton*). In this second case we also provide a very common example in the literature on hybrid systems, inspired by [25]. The study of this class of system has been done in collaboration with a group of the University of Magdeburg (headed by Professor Jörg Raisch). The framework was motivated by the introduction of *safety* specifications that may be converted into state space constraints. The procedure that converts safety specifications into constraints on the state space is in Appendix D, and it is a result taken from Raisch *et al.* [95, 85] and integrated with the STP.

In Chapter 6 we consider the results obtained in Chapter 4 and we extend to the case where an infinite number of switches are allowed. We firstly conjecture that the cost of an evolution, J_N^* described above, must be a decreasing function of N for every initial point. With similar arguments we assume that the switching tables $\mathcal{C}_N^i$ must converge to the same one, that we call $\mathcal{C}_\infty^i$, with obvious notation; hence the controller may use them *indefinitely* until the state $\boldsymbol{x}$ has reached the steady state.

After formally proving these results, a case study is analyzed in detail: it is the design of a *semiactive* suspension, with LTI model, whose damper coefficient may take values from a finite set[2]. We consider the cases with one DOF, i.e., neglecting the deformation of the tire, in $\mathbb{R}^2$, and with 2 DOF, that considers the deformation of the tire, that is modelled in $\mathbb{R}^4$.

The application of the STP in the fourth dimensional case has been particularly challenging. The numerical strategy, concerning the *polar discretization* and the *interpolation* of the cost are detailed in Appendices C.1 and C.2.

Chapter 7 is dedicated to stability. Therein we consider a *completely connected* switched system, i.e., every location is connected to all the others by an oriented arc, and we apply the STP with infinite number of switches. In this context we also prove that *all* tables converge to $\mathcal{C}_\infty$, that is independent from the particular location.

[2]Some fluids may vary their *viscosity* when subjected to an appropriate magnetic field, [48].

The main idea contained in this chapter is briefly summarized as follows: if the STP is able to design a table $\mathcal{C}_\infty$ that drives the system to the origin then the optimal signal $i(t)$, is also a *stabilizing* signal for the switched system. This is remarkable because it relates the notion of stability with the notion of optimal control. As a result this is an alternative methodology of designing a stabilizing feedback control law for a switched system, which is one of the major efforts in the current literature.

Moreover we prove another important result: if the table $\mathcal{C}_\infty$ *does not* contain the region of a specific mode j of the switched system, then the *reduced* switched system, obtained by refining the original system of mode j, must have the same table obtained for the original system.

This is relevant and we present significant examples, many of them taken from literature as benchmarks, where we design a stabilizing switching law for systems composed of only unstable modes.

As a final example we propose a comparison between the table obtained with the STP and the table obtained analytically by analytical minimization of the cost function over the parameters of an expected switching surface.

Chapter 8 draws the conclusion of the work and it glances over open perspectives and developments.

A certain number of Appendices was considered necessary to complete the work. We will briefly summarize their content in the following.

The continuous references to LQR problems required an appendix to the classical results in system theory on LQR problems for LTI systems. Theoretical notions are recalled in Appendix A, and their numerical implementations in Appendix B.

Appendix C contains issues on the state space discretization. In particular we provide a convenient method to discretize the unitary semisphere in $\mathbb{R}^n$. This serves mainly in Chapter 6 where a problem in $\mathbb{R}^4$ is considered.

Appendix D describes the methodology, developed by Gromov *et al.* [32] on the basis of the $l-$complete approximation, that converts specifications on the output signals of a hybrid system into constraints on the state space.

In Appendix E we provide a user guide of the software developed to obtain part of the numerical results presented in Chapters 4, 6, 7. This software constructs the switching tables and uses them in an on line simulation for a switched system. It is available in the $\mathbb{R}^2$ and $\mathbb{R}^4$ case at the site

http://www.diee.unica.it/∼dcorona/thesis.html.

Finally we refer the reader to Appendix F, where all the acronyms, symbols, notation and units of measurements are collected.

2

Optimal control and stability of hybrid systems: literature review

In this chapter we propose a literature review on two major fields in the context of *hybrid systems*, namely the *optimal control* and the *stability* of hybrid systems.

Both of these topics are of relevant interest in the control and computer science community and many theoretical results and algorithms are available.

Clearly the research on the hybrid systems and models is not only restricted to the mentioned fields, but it also involves other issues, such as reachability, controllability and observability, to cite few.

Nonetheless we decided to review only the literature concerned with the optimal control and the stability, from where we derived many suggestions, properties and ideas that have been crucial in our research.

This survey has synthetic form, and sometimes, for the sake of brevity we could not analyze in appropriate detail the presented topics. However we hope that the references therein may be useful.

2.1 The optimal control problem for hybrid systems

The problem of determining optimal control laws for *hybrid systems* has been widely investigated in the last years and many results can be found in the control and computer science literature. The increasing interest in this new class of *synthesis design problems* is probably due to the reasonable trade off between the *modelling power* of these models and the *feasibility of the solution*. The vaste spectra of physical systems that can be modelled by a hybrid system and the different targets of a control strategy have led to an extremely various literature.

Therefore we considered useful for the reader to present in this thesis the most significant result in the problem of designing optimal control laws, in general hybrid themselves, for hybrid systems. This collection, far from being considered exhaustive, may also orient the reader in the intricate variety of publications on this topic.

This section gives particular attention to results that have been of relevant interest in the field and we decided to collect and describe separately the work of the authors that mostly influenced the development of our work.

2.1.1 Antsaklis and Xu

A relevant contribution to the study of the optimal control problem of hybrid system is certainly due to the extensive work of Xu and Antsaklis. In [120] they approach the finite time horizon general problem under the condition of pre-assigned finite length switching sequence. In this framework the control variables are the *switching instants*

σ and the continuous control in each branch $\boldsymbol{u}(t)$. The problem is to minimize a cost functional J that weights the state $\boldsymbol{x}(t)$, the continuous control $\boldsymbol{u}(t)$ with fixed finite final time and free terminal state, for a given initial point.

In general the authors propose a two stage algorithm obtained by decoupling the minimization procedure over the two control variables: first a minimization over the *continuous control* is performed, then the *switching instants* are tuned to obtain a global minimization of the objective function.

They suggest therein an approach based on dynamic programming arguments that limits considerably the explosion of the computational effort, due to the combinatoric nature of the problem. Moreover it permits them to impose continuity conditions at the switching instants.

They also study in [121] how the method admits an analytical formulation in the particular GSLQ (General Switched Linear Quadratic) problem, because the boundary conditions given by the Riccati equation can be efficiently exploited.

This technique led interestingly to a reformulation of the general nonlinear problem into an equivalent problem where the switching instants are *parameterized*, i.e, expressed in terms of the derivatives of the cost functional in the surrounding of the switching instants, as presented in [123] and in [129]. In these works the boundary conditions at the switching instants are obtained from the solution of a *two points boundary value DE*, composed of the state and costate defined in the Hamiltonian function, as described in [93].

Once this has been done a direct differentiation of the cost can be performed, as described in [125]. Note that the methods described are of difficult solution when the number of switches N grows, because a constrained N dimensional minimization of a function must be performed. Furthermore the function is, in the general case, given only numerically. Finally these methods do not provide a feedback solution, hence the optimal strategy is valid only for the given initial point.

Nonetheless, in [118], the author indicates several classes of theoretical and applicative relevancy where these ill-conditioned tasks may be avoided or simplified.

The general methods have also been applied to two interesting particular cases:

- Optimal control of switched *autonomous* systems [124], $\dot{\boldsymbol{x}} = f_{i(t)}(\boldsymbol{x}, t)$, i.e., the minimization is performed only under the switching instants, with $\boldsymbol{u} = \boldsymbol{0}$. This case is relatively close to the one considered in [49] and the general problem considered in this thesis. The remarkable difference is that a finite time horizon with nonlinear functions are considered. Conversely the solution is not state feedback, hence the necessity of an on line computational effort. This framework has been also explored in presence of state jumps in [127], allowing a more general model that also takes into account the switching costs as in [50, 28].
- Optimal control of switched systems with *internally forced* switchings [122], where the unique control variable is the continuous input $\boldsymbol{u}$, while the while the system is subject the occurrence of *state* or *time* dependent switches. This idea has been considered in this thesis in Chapter 5, based on the results provided in [30].

A detailed survey on the optimal open loop control methodology developed by Xu and Antsaklis can be found in [128], where also comparisons with different approaches are provided.

Another problem presented in this thesis is somehow related to the work of Xu and Antsaklis. The CHA problem considered in Chapter 5 and developed in [32] is in fact similar to the problem presented in [126]. In this work the authors consider an *integrator switched system*, i.e., a linear affine switched system where all dynamics $\boldsymbol{A}_i$ are null, and a *time* optimal control problem. The objective is to design the switching scheduling in order to drive the system state at a given destination in *minimal time*, in presence of constraints on the state space. This is done by a conversion of the problem into a MILP (Mixed Integer Linear Programming) and solved with the available tools. We considered instead the problem of driving the system state into a given destination in minimal *energy*, i.e, by minimizing an LQR-like cost function.

2.1.2 Rantzer, Hedlund and Lincoln

A different approach than the one presented by Xu and Antsaklis is suggested by *Rantzer et al.*. In [57] the authors consider a model of switched system and an annexed optimal control problem in general form, i.e., non linear piecewise vector fields and general cost functional. As considered in Chapter 4 and in [29] the authors approach the problem of minimizing a cost function defined for a *finite length*, yet not fixed, switching sequence. Moreover they consider the case of a cost associated to the event driven evolution, by associating a switching cost to each switch, so preventing the possibility of any Zeno behavior executions, as defined in [66].

As a difference they limit the investigation to a *finite time* horizon, hence they aim to drive the system to a fixed terminal *hybrid state* while they let free the terminal time. In this work the notion of guard is introduced, although not explicitly defined, i.e., the system is allowed to perform a switch from a location i to a location j whenever the continuous state $\boldsymbol{x}$ enters a region $S_{i,j} \subseteq \mathbb{R}^n$.

The optimization problem is tackled by the introduction of a set of inequalities of particular functions V_i in the *hamiltonian*[1] form. Boundary conditions are imposed as equality and inequality constraints on this functions. The authors prove that the minimization problem is lower bounded by this set of functions in the state variables. Hence it is sufficient to maximize the given functions. This method requires a state space discretization, and it provides a switching strategy in feedback form. Nevertheless it encounters major difficulties due to *dimensionality*.

As an extension to the described research an algorithm to optimize switching sequences that has an arbitrary degree of *suboptimality* was presented by Lincoln and Rantzer in [77] and in [79]. Therein the authors consider a quadratic optimization problem whose solution is *suboptimal*, but with known error bounds. This is achieved via a *relaxed dynamic programming*, obtained by relaxing the Bellman principle [14] to a non strict inequality. The idea is described in [78].

As a difference with the work previously described [57] the authors consider discrete time systems. In particular in [79] the discrete time switched system is composed of only two vector fields, whose current mode is active according to a given partition of the state space, hence the control variable is restricted to the continuous input $\boldsymbol{u}(t_k)$. Briefly the method consists in locating the solution of the optimal control problem, which is *non convex* for switched systems, between two $\alpha-$*stretched* values. The interval may be restricted, at the cost of a higher computational effort.

This idea is extensively described by Rantzer and Hedlund in [58] where they use *convex dynamic programming* to approximate hybrid optimal control laws and to compute lower and upper bounds of the optimal cost. For determining the optimal feedback control law these techniques require the discretization of the state space in order to solve the corresponding HJB equations.

In [76] Lincoln and Bernhardsonn propose a method for efficient pruning of the search tree in order to avoid combinatoric explosions amongst all the possible paths of a hybrid execution. In such a way they obtain a numerically viable procedure that permits them to solve a finite discrete time LQR problem for a switched system.

[1] We refer to the classical definition of hamiltonian functions as, for example, in [69], i.e., a function that weights the vector field by the co-state and the cost functional.

2.1.3 Shaikh and Caines

The work of Shaikh and Caines in the field of optimal control of hybrid systems is based on the construction of a set of necessary conditions that represent, to an extent, a generalization of the maximum principle, in analogy with the works of [93, 110].

In [102] the authors propose an algorithm that performs a minimization search for a finite continuous time cost function and a controlled set of vector fields. They initially assume that the switching schedule is fixed, hence the minimization is performed under the control variables $\boldsymbol{u}$ and the switching instants.

In [105] the results are formally presented for a finite-time hybrid optimal control problem and necessary *optimality* conditions for a fixed sequence of modes using the maximum principle are provided.

In [104] these results are extended to *non-fixed* sequences by using a suboptimal result based on the *Hamming* distance permutations of an initial given sequence. In this framework the approach of Shaikh and Caines appears similar to the idea contained in the *master-slave* procedure described in [29].

Besides, in [103], the authors derive a feedback law (similar to that one considered in [28]) but for a finite time LQR problem whose solutions are strongly dependent on the initial conditions, thus providing open-loop solutions.

Consistently with most of the literature on the hybrid systems the authors conclude in [103] that any feedback control law that minimizes a given performance index has to be represented as a partition of the state space. In fact they present an algorithm, namely HMP[Z] based on *optimal zones* in time-space, which is an extension of the HMP[MCS].

These regions are necessary to identify the optimal switching instants and sequences, while the continuous control in each mode of the hybrid system can be determined, in the LQR case, by using the boundary conditions given by the finite time Riccati equation.

The optimal sequence of modes in the hybrid trajectory is obtained via dynamic programming arguments, for a finite number of switches. This is to avoid the combinatoric explosion of all possible switching sequences. A detailed description of these algorithms and their possible applications and developments can be read in [101].

2.1.4 Cassandras and Wardi

The work of Cassandras and Wardi took basically two research areas in the vast field of the optimal control of hybrid systems. The first one, introduced by Cassandras *et al.*, consists in developing algorithms to the aim of optimally controlling a *manufacturing* system modelled by a hybrid system. The latter, further developed by Wardi *et al.*, concerns the optimal control of the commonly defined *switched systems*.

In [21] the authors consider a manufacturing system modelled via a hybrid system. The system is composed of a *continuous* variable $\dot{z}_i = u_i$ in each job i, related to the quality of the jobs, and a time variable, subject to discrete events, such as start-time, end-time, duration thresholds and so on.

In this framework an optimal control technique is necessary to balance the trade off between the quality of the product, that increases with duration, and time consuming. A quadratic cost that weights both the processing time and the continuous control input for a system of N jobs and two servers is considered.

The control law must be tuned in order to guarantee that the jobs are satisfactorily processed in a relatively short term. The suggested approach is the maximum principle, subtly applied with the aid of the *Bezier* approximation of the co-state at the intermediate condition.

An extension that concerns the continuous dynamics of the model is studied in [52], governed by a first order DE of the form $\dot{z}_i = g_i(z_i, u_i, t)$. In the mentioned work the authors use a hierarchical decomposition approach [89, 20, 94] to break down the overall optimal control problem into smaller ones. In particular the decomposition of the global system into a *low level*, governed by continuous dynamics, and a *high level*, governed by event driven dynamics, allows the design of a hybrid controller that solves a mixed optimal control problem. In so doing, discretization is not involved and the main computational complexity arises from a higher-level nonlinear programming problem. In Chapter 5 we adopted, in some sense, this methodology for a plant that must respond to *safety* constraints (low level) and optimality (high level).

A more general first-order optimality conditions and several properties of optimal trajectories, that significantly simplify the task of the explicitly design the control law, are proved in [23]. Wardi *et al.*, in [114] propose an algorithm that designs the optimal control law by proceeding backward in time, i.e., from the last job to the first, by means of a similar methodology of the dynamic programming used in the algorithm developed in this thesis.

In a different context Wardi *et al.* in [41] analyze the *autonomous* switched system model as defined in [124], which is basically the model considered also in this thesis. For this class of system they propose a *finite time* optimal control problem, where the control variables are the *switching instants*.

The considered approach is based on the parameterization of the cost function with the switching instants $\{\tau_1, \tau_2, \ldots, \tau_N\}$ and perform a descent *gradient* method to obtain the minimum under the N variables. Note that this approach becomes unfeasible when N grows. Furthermore it is in general dependent on the initial conditions, hence it does not provide a state feedback control law.

In a recent paper [113] the same problem is tackled with the aid of *parameterized switching surfaces*. Conceptually the authors consider a given family of switching manifolds in $\mathbb{R}^n$, parameterized by $k < n$ parameters, and express the cost functional in terms of these parameters. The main goal of this idea is to allow an iterative procedure.

Note that the example described in Section 7.6.3 follows the same idea. In fact therein we consider a *conic* switching law in $\mathbb{R}^2$ parameterized by the slopes m_1 and m_2. Then we attempt to express the cost as a function of m_1, m_2 and minimize over them. The arguments m_1^* and m_2^* that minimize the cost must be the optimal switching surfaces. In fact it will be proved in the following chapter that the optimal switching surfaces of the considered class of *autonomous* (in the sense of Xu and Antsaklis) switched linear systems must be *conic*.

2.1.5 Bemporad and Morari

The hybrid optimal control problem becomes less complex when the dynamics is expressed in discrete time or as discrete events. For discrete time linear hybrid systems, Bemporad and Morari [11] introduce a hybrid modelling *unified framework* MLD that is focussed on linear systems described by continuous and logic rules.

This framework handles in particular the hybrid systems with both internally forced switches, i.e., caused by the state reaching a particular boundary, and controllable switches (i.e., a switch to another operating mode can be directly imposed). In addition the authors show how *mixed-integer quadratic programming* (MIQP)[10] can be efficiently used to determine optimal control sequences.

They also show that when the optimal control action is implemented in a receding horizon fashion by repeatedly solving MIQP's on-line, an asymptotically stabilizing control law is obtained.

It is relevant to remark that most of the hybrid models, i.e., models that integrate logics and dynamics, can be described in a unified framework, hence they might be approached via MIQP algorithms. A notable work that shows the theoretical *equivalence*[2] of 5 classes of discrete time hybrid systems, *piecewise affine* PWA, *linear complementarity* LC, (De Schutter, [38]), *extended linear complementarity* ELC, *max-min-plus-scaling* MMPS, (De Schutter and Van Den Boom, [39]) is provided in [12].

Bemporad, in [4], proposes two algorithms for an efficient conversion of a MLD into a PWA system.

For those cases where on line optimization is not viable, Bemporad *et al.* [5, 6] and Borrelli *et al.* [15] propose multi-parametric programming as an effective means for solving in *state feedback* form the finite time hybrid optimal control problem with performance criteria based on 1-, ∞-, and 2-norms, by also showing that the resulting optimal control law is piecewise affine.

In the discrete time case, the main source of complexity is the combinatorial number of possible switching sequences. By combining reachability analysis and quadratic optimization, Bemporad *et al.* [7] propose a technique that rules out switching sequences that are either not optimal or simply not compatible with the evolution of the dynamical system.

In many cases the optimization of hybrid processes is achieved by decoupling the logic optimization from the continuous optimization, as for instance with *hierarchical* approaches (see, among many, [52, 94]) or *two stage* optimization (see, for example, [120]). This task can be also viewed as a combination of mixed integer linear programming (MILP) with continuous dynamic simulations, to obtain a potentially optimal switching sequence, as it is proposed in [88].

Another approach that merges the techniques developed on discrete events dynamics and continuous time switched systems is called *master-slave procedure* (MSP) [8, 29].

The procedure *alternates* between two different procedures, to the aim of optimizing hybrid processes. In other words the procedure *iterates* between a *master* procedure that finds an optimal switching sequence of modes, and a *slave* procedure that finds the optimal switching instants.

- The *master* procedure is based on mixed-integer quadratic programming (MIQP) and finds an optimal switching sequence for a given initial state, assuming the switching instants are known.
- The *slave* procedure, based on the construction of the switching regions [49] by means of dynamic programming arguments, solves an infinite time horizon with finite number of switches for a fixed sequence linear switched system composed of autonomous dynamics. Here the control variables are the switching instants, that must be tuned in order to minimize a performance index of piecewise LQR class.

It can be proved that this algorithm converges with finite number of steps, but it is not guaranteed to detect the global minimum. A few simple heuristics, that explores small perturbations on the sequence of the switching indexes, can be added to the algorithm to improve its performance.

[2] Equivalency *means that for the same initial conditions and input sequences the trajectories of the systems are identical* [4].

2.2 Miscellaneous work

For what concerns *switched affine systems*, that are a particular class of *hybrid systems* considered in the majority of the following chapters, the problem of optimal control synthesis has been investigated with accuracy during the last decade.

A significant portion of the current literature on optimal control of *switched systems* is focused on the study of necessary conditions for a trajectory to be optimal [93, 110, 45]. In particular *necessary optimality conditions* for a trajectory of a *switched system* are derived using the *maximum principle* by Sussmann [110] and Piccoli [93], who consider a fixed sequence of finite length, in finite time.

A similar approach is used by Riedinger *et al.* [97], who restrict the attention to linear quadratic cost functionals but considering both autonomous and controlled switches.

An important effort is devoted on the computation of *optimal/suboptimal solutions* by means of dynamic programming or the maximum principle [19, 18, 52, 57, 97, 123]. Optimal control of discrete-time hybrid systems is studied in [6].

For continuous-time hybrid systems, Branicky and Mitter [19] compare several algorithms for optimal control, while Branicky *et al.* [18] discuss general conditions for the existence of optimal control laws for hybrid systems.

For determining the optimal feedback control law some of these techniques require the discretization of the state space in order to solve the corresponding Hamilton-Jacobi-Bellman equations, see for instance [58].

Bengea and De Carlo [13] apply the maximum principle to an embedded system governed by a logic variable and a continuous control. The provided control law is open loop, nevertheless some necessary and sufficient conditions are introduced for optimality.

For a special class of discrete-event systems, De Schutter and Van Den Boom [39] proposed an optimal receding-horizon strategy that can be implemented via linear programming.

2.3 Stability and stabilizability of hybrid systems

The problem of analysis and control of hybrid systems has attracted the attention of many researchers. In particular most of the research literature is focused on defining the conditions [42, 40, 117, 130] of *stabilizability* of switched systems, and in particular of *linear affine* switched systems, the same class we considered in Chapter 7.

The theoretical effort in this sense is to express the *structural* conditions of a given switched system that guarantee the existence of a stabilizing switching signal $i(t)$, a piecewise constant function of infinite branches.

This problem, formulated in [17], is not trivial, and it is well known that the *Hurwitz* stability of at least one dynamics of the switched system is a sufficient, yet *not necessary* condition for the existence of such signal, provided that an infinite number of switches are allowed. Many examples can be found in literature and also in Chapter 7.

Conversely it is possible to design an appropriate switching law that may provoke *instability* of the global system even if all its modes are Hurwitz.

Nevertheless there does not exist yet a *general* result that provides necessary and sufficient conditions for the global asymptotic stabilizability of a switched system with unstable dynamics. Analogously, there does not exist yet a general procedure to compute, when it does exist, an asymptotically stabilizing switching law.

Necessary and sufficient conditions are given in [42, 115] in the case of two switched systems when the criterion under consideration is the *quadratic* stability of the switched systems. The main importance of this property is that it requires for uncertain systems a quadratic Lyapunov function which guarantees asymptotic stability for *all* uncertainties under consideration, and is thus a kind of robust stability with very good property, yet usually needs more restrictive conditions [131]. Iterative algorithms for constructing such common Lyapunov function can be found in [75].

Another interesting issue in the stabilizability field is the investigation of the convergence rate. In fact it is well known that traditional LTI systems are exponentially stable iff their dynamics are Hurwitz.

Sun, in a series of papers [109, 108, 107], provides important results that concern the convergence rates of a switched linear system subject to any switching signal. In particular he proved that for this class of system the stabilizability always implies an exponential stabilizability.

Far from pretending to be exhaustive we would like to present in the following a short description of the most commonly used methods to effectively design a stabilizing switching signal. We may always refer to switched LTI systems, in fact, apart from some very restrictive situations, there are only few results for more general classes.

2.3.1 Methods based on time approach

These methods aim to design a *time based* switching law. More precisely the switching signal is time scheduled and it is not presented in feedback form. Among the first researchers that proposed this method, namely the *dwell time* approach, there are Hespanha and Morse [60].

The main idea is based on the fact that the porter frequency of the switching signal is *slow-on-the-average*, when applied to switched systems composed of *only* stable dynamics. In particular, it is proved that exponential stability is achieved when the number of switches in any finite interval grows linearly with the length of the interval, and the growth rate is sufficiently small.

On the other hand the switching period of a switched system composed of only unstable modes must not be greater than a certain value in order to preserve that the state $\boldsymbol{x}$ is maintained in a desired neighborhood of the origin.

What we found interesting in these methods is the evident parallelism with the conditions of stability for slowly time varying systems [74]. Furthermore the results of Hespanha have inspired the introduction, in the methodology presented in this thesis, of the minimum permanence time in each location of the considered hybrid automaton, whose presence not only models a physical behaviors, but it also avoids instability and Zenoness.

The idea has been also reconsidered by Colaneri and Geromel in [26], where the minimum dwell time is determined by means of a family of quadratic Lyapunov function.

These methods however do not provide a closed loop control law, but merely time dependency.

2.3.2 Methods based on geometrical approaches: planar systems

A special attention is devoted to the design of stabilizing switching signals for *planar* switched linear systems[3]. In fact in this case a geometrical approach appears

[3] A planar system is a system whose vector fields is in $\mathbb{R}^2$.

dominant in all works. Another advantage of this class of systems is that the stabilizing switching signal, if any, may exhibit periodic behaviors, or a degeneracy into a sliding mode surface

To cite a few Antsaklis *et al.* in [62, 119] using a geometric approach, were able to obtain necessary and sufficient conditions for asymptotic stabilizability of switched systems with an arbitrarily large number of *second-order* LTI unstable systems. Moreover, when the switched system is asymptotically stabilizable, they also provide an approach to compute a stabilizing law.

Following the same idea Michel *et al.* prove in [63] necessary and sufficient conditions for planar systems that guarantee the existence of a common quadratic like Lyapunov function for the global switched system, and they also provide a synthesis method of conic switching regions based on the mutual directions of the vector fields.

For planar systems necessary and sufficient conditions for the existence of a stabilizing switching signal are given for nonlinear embedded *convexified* problem, i.e., $\dot{\boldsymbol{x}}(t) \triangleq i(t)F(\boldsymbol{x}(t)) + (1 - i(t))G(\boldsymbol{x}(t))$, as it can be seen in [16].

2.3.3 Methods based on multiple Lyapunov functions

Many works on the stability analysis of switched systems are based on the use of multiple Lyapunov functions (MLF's) [17, 74, 130, 82]. The general idea is to seek a piecewise multiple Lyapunov function, active in each mode of the switched system, that behaves as a global Lyapunov function for the global switched system.

Broadly speaking the main target of this methodology is to provide a solution of the matrix inequality $\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i < \boldsymbol{0}$ in order that the function $V(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{Z}_i\boldsymbol{x}$ is globally decreasing.

It is relevant to remark that in all these cases the proposed approaches only give sufficient conditions for the asymptotic stabilizability.

In some cases the switched system admits a *common* Lyapunov function. It is the case, for example, of *quadratically* stable systems, studied in detail by Feron in [42] and by Pettersson *et al.* in [40]. The quadratic stability involves the existence of a *common* Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{x}'\boldsymbol{Z}\boldsymbol{x}$, i.e., independent from the switching signal $i(t)$, such that it holds $\dot{V}(\boldsymbol{x}) \geq -\varepsilon\boldsymbol{x}'\boldsymbol{x}$, where ε is an arbitrarily small positive number.

The quadratic stability can be checked *a priori* by simply analyzing structural properties of the switched system. In fact it is proved in [40], Theorem 4.3, that a switched system is quadratically stable if there exists a stable convex combination of its modes. The condition becomes sufficient if only two dynamics are considered.

In the general case the problem of stabilizing a switched system with unstable dynamics $\boldsymbol{A}_i$'s is often translated into the problem of solving a set of quadratic inequalities. This task derives from the general idea of constructing a decreasing common or multiple Lyapunov function.

This is appealing, but it turns out to be a non convex problem (thus it only provides sufficient conditions) when the number of subsystems is greater then 2. Moreover many proposed solutions lean on *linear matrix inequalities* (LMI) or bilinear matrix inequalities (BMI) methods, which become computationally problematic as the number of modes grows [40, 90].

Recently, Ishii *et al.* in [64] present an alternative method for solving the search of a MLF, that is, as remarked above, a key issue in the synthesis of stabilizing a switched system. Their approach provides a probabilistic algorithm, that converges with a given probability that exploits a gradient descent method on energy and multi modal Lyapunov functions, as described also by Tempo *et al.* in [75]. The method presents no theoretical restrictions on the order of the LTI systems but it has exponential complexity, albeit it is guarantee to converge in a finite number of steps.

2.3.4 Methods that relate a quadratic cost with stabilizability

The idea of solving the Lyapunov matrix inequality extended to switched system has been considered in the previous section. A natural approach, described by [67], looks for a solution of an extended Lyapunov equation of the form $\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i + \boldsymbol{Q} < \boldsymbol{0}$ where $\boldsymbol{Q} \geq \boldsymbol{0}$.

The intuition behind this approach lies in the fact that this last inequality may be simpler to satisfy than $\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i < \boldsymbol{0}$. Furthermore this idea has been studied in detail in Pettersson's PhD thesis [90], where appropriate algorithms and extensions have also been developed. For a short resume of these techniques see also [91].

In a very recent paper, by Colaneri and Geromel [26], this technique has been related to the minimization of a linear quadratic performance index where the state valued is quadratically weighted with a $\boldsymbol{Q} \geq \boldsymbol{0}$ matrix. Hence the solution of the multiple Lyapunov equation $\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i + \boldsymbol{Q} < \boldsymbol{0}$ must be somehow connected to the minimization of $J = \int_0^\infty \boldsymbol{x}(t)'\boldsymbol{Q}\boldsymbol{x}(t)dt$. The authors propose a synthesis based on the solution of the Lyapunov Metzler equation, via a numerical approach based on LMI.

In [31] and in Chapter 7 we illustrate how the STP developed in this thesis may be also used for the *design* of a stabilizing control law for a general switched system by the minimization of a quadratic cost. The idea behind this methodology is related to the fact that if we achieve in finding a minimum *finite cost* in infinite time horizon, then, under appropriate conditions, we have also driven the system into the origin, hence we obtained a stabilization.

Sun, in a paper recently submitted to Automatica, [108], proposes some theoretical conditions for the existence of a switching signal $i(t)$ that not only exponentially stabilizes the switched system, but it also provides the minimization of a quadratic performance index.

3

Hybrid systems: models and optimization problems

3.1 Introduction

In this chapter we will describe in detail the models and the optimization problems considered in this thesis.

We will define formally the general notion of hybrid automaton GHA in Section 3.2, taken from [86] and [2], and then we will define the subclasses considered in this thesis.

In particular we initially define in Section 3.3 one of the *simplest* class of GHA, namely the linear affine switched system S, that switches between many operating modes, where each mode is governed by a linear affine dynamical law.

We define an annexed optimal control problem OP of the form *piecewise LQR* in Section 3.4.2 both with finite and infinite number of controllable switches.

The solution of the OP with finite number of switches is described in Chapter 4, while in Chapter 6 and 7 we deal with an infinite number of switches.

We will also define the linear affine hybrid automaton HA in Section 3.5, which is a generalization of a switched system characterized by the presence of constraints on the state space. This aspect influences its dynamical behavior, by the occurrence of autonomous switches (AHA), or it restricts the action of an external discrete controller (CHA), permitting to model safety or constructive specifications.

We will formally define the annexed optimal control problem in the two cases, whose solution is addressed in Chapter 5, and it only deals with finite number of switches.

3.2 Definition of general hybrid automaton GHA

A *hybrid automaton* consists of a classic automaton extended with a continuous state $x \in \mathbb{R}^n$ that may continuously evolve in time with arbitrary dynamics or have discontinuous jumps at the occurrence of a discrete event.

In this section we recall the general definition of the hybrid automaton. We denote in the following the general form of the hybrid automaton with the acronym GHA. A GHA is a structure $GHA = (\mathcal{L}, act, inv, \mathcal{E}, \mathcal{M})$ in consistency with the current literature definitions (see for instance [2] and [86]). Briefly

Definition 3.1 (Hybrid automaton) *A* hybrid automaton GHA *is a tuple* $GHA = (\mathcal{L}, act, inv, \mathcal{E}, \mathcal{M})$*, whose entries have the following meaning:*

— $\mathcal{L}$ *is a finite set of* locations *indexed by* $i = 1, \dots, s$.

— $act : \mathcal{L} \rightarrow$ Inclusions *is a function that associates to each location* i *a* differential inclusion.

— $inv : \mathcal{L} \rightarrow$ Invariants *is a function that associates to each location* i *an* invariant $inv_i \subseteq \mathbb{R}^n$ *such that* $\boldsymbol{x} \in inv_i$.

— $\mathcal{E} \subset \mathcal{L} \times Guards \times \mathcal{L}$ *is the set of edges. The edge* $e_{i,j}$ *is enabled when the current location is* i *and the current continuous state is* $\boldsymbol{x} \in g_{i,j}$*: it may fire reaching the new location* j.

— *A* jump *relation is* $\mathcal{M} \subset \mathbb{R}^n \times \mathbb{R}^n$ *associated to an edge* $e_{i,j}$*. When the edge fires,* $\boldsymbol{x}$ *is reset to* $\tilde{\boldsymbol{x}}$ *according to* $\mathcal{M}$.

■

The state of the GHA is the pair $(\boldsymbol{x}, i)$ where $\boldsymbol{x} \in \mathbb{R}^n$ is the continuous state, and the index i identifies the current discrete location.

From this general definition we will analyze the particular cases studied in this thesis.

We will only consider linear affine system, i.e., the function act_i coincides with a *linear time invariant* affine differential equation of the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{f}_i.$$

In the next section we define the *switched system*, whose model is considered in Chapters 4, 6 and 7.

In Section 3.5 we will define the model considered in Chapter 5.

3.3 Definition of switched systems S

A *switched system* is a particular class of hybrid automaton (GHA).

In Chapters 4, 6 and 7 we focus the attention on a particular class of GHA, that we call *switched linear affine systems* S. The S switches between many operating modes, where each mode is governed by its own characteristic dynamical law [1].

We provide a formal definition of a S that will be used in Chapter 4, 6 and 7.

Definition 3.2 (Switched system) *A* switched system *is a structure* $S = (\mathcal{L}, act, \mathcal{E}, \mathcal{M})$*, where*

— $\mathcal{L}$ *is a finite set of locations, indexed by* $i = 1, \ldots, s$.

— $act : \mathcal{L} \rightarrow (\mathbb{R}^n \times \mathbb{R}^n)$ *is a function that associates to each location* i *a LTI affine differential equation of the form* $\dot{\boldsymbol{x}} = \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{f}_i$.

— $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ *is the set of edges. An edge* $e_{i,j} = (i, j)$ *is an edge from location* i *to* j, $i \neq j$.

— $\mathcal{M} : \mathcal{E} \rightarrow \mathbb{R}^{n \times n}$ *associates to each edge* $e \in \mathcal{E}$ *a constant matrix in* $\mathbb{R}^{n \times n}$*. When the discrete state switches from location* i *to* j *at time* τ*, the continuous state* $\boldsymbol{x}$ *is reset to* $\boldsymbol{x}(\tau^+) = \boldsymbol{M}_{i,j} \boldsymbol{x}(\tau^-)$. ■

We denote by $\mathcal{S}$ the set of indexes associated to each location, and $s = |\mathcal{S}|$.

The S admits a time driven evolution governed by the law described in the activity and an event driven evolution described by the sequence of locations visited by the system during the time driven evolution.

The S starts from some initial state $(\boldsymbol{x}_0, i_0)$. The trajectory evolves with the location remaining constant and the continuous state x evolving according to the act function at that location. When at time τ a switch is made to location i_1 the continuous state is initialized to a new value $\boldsymbol{x}(\tau^+) = \boldsymbol{M}_{i_0,i_1} \boldsymbol{x}(\tau^-)$. The new state is the pair $(\boldsymbol{x}(\tau^+), i_1)$. The continuous state now moves with the new differential equation.

It is possible to associate to a S, or equivalently to a GHA, an oriented graph, according to the following definition.

Definition 3.3 (Oriented graph of a S) *An* oriented graph *of a S is obtained by associating to each location $i \in \mathcal{S}$ a node and to each edge $e_{i,j}$ an oriented arc from node i to node j.* ■

An example is depicted in Figure 3.1.

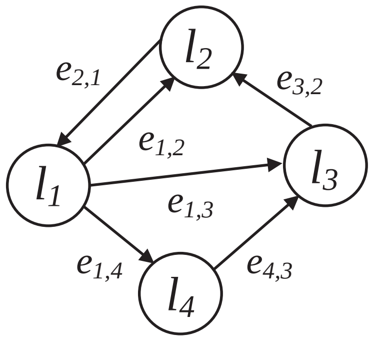


Fig. 3.1. *Oriented graph of a S composed of 4 locations and 6 edges.*

Before proceeding further it is important to provide the following definitions.

Definition 3.4 (Set of successors of i) *The* set of successors $succ(i) \subset \mathcal{S}$ *of location i in a S is the set of all $j \in \mathcal{S}$ such that $\exists\ e_{i,j} \in \mathcal{E}$.* ■

This set is composed of the location indexes that can be reached from location i by firing one and only one edge.

Note that the set $succ(i)$ does not include i itself. As an example we can consider the S whose graph depicted in Figure 3.1. It is clear that

$$succ(1) \equiv \{2,3,4\},\ succ(2) \equiv \{1\},\ succ(3) \equiv \{2\},\ succ(4) \equiv \{3\}.$$

The state of a S, and in general of a GHA, is completely identified by the continuous time variable $\boldsymbol{x} \in \mathbb{R}^n$ and the index of the current discrete location i.

Definition 3.5 (Hybrid state) *The* state *of the S is the couple $(\boldsymbol{x}, i)$ where i is the discrete location and $\boldsymbol{x} \in \mathbb{R}^n$ is the continuous state.* ■

Let us observe that the hybrid state is composed by a *continuous part* $\boldsymbol{x}$ and a *discrete part* i. Analogously we can define the *hybrid evolution*.

Definition 3.6 (Hybrid evolution) *The* hybrid evolution *in a time interval $[t_1, t_2]$ of the S is the couple $(\boldsymbol{x}(t), i(t))$ where i is the discrete location at time t and $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the continuous state at time t, $\forall\ t \in [t_1, t_2]$.* ■

Note that a hybrid evolution is a sequence of hybrid states.

The continuous part of the evolution is governed by the differential equation corresponding to the current location, given in the discrete part. The discrete part of the evolution is governed by the set functional $i(t) \in succ(i) \cup \{i\}$.

Let us consider a value of τ in the open interval (t_1, t_2) and the corresponding value of the hybrid state $(\boldsymbol{x}(\tau), i(\tau))$. From this point we would like to calculate the hybrid state at time $\tau + d\tau$, i.e., $(\boldsymbol{x}(\tau + d\tau), i(\tau + d\tau))$, where $d\tau \to 0$.

We can separately analyze the two cases.

Case 1: τ is a switching instant.

In this case an event driven evolution occurs. Suppose that $i(\tau) = i$ and $j \in succ(i)$, i.e., there exists an edge leading from location i to location j, and it fires at time τ.

The new value of the hybrid state is simply

$$\boldsymbol{x}(\tau + d\tau) = \boldsymbol{M}_{i,j}\boldsymbol{x}(\tau), \ \ i(\tau + d\tau) = j).$$

In the sequel we will often indicate for brevity $\boldsymbol{x}(\tau + d\tau) = \boldsymbol{x}(\tau^+)$ and $\boldsymbol{x}(\tau) = \boldsymbol{x}(\tau^-)$, denoting the right and the left part respectively of the

$$\lim_{t\to\tau}\boldsymbol{x}(t).$$

Note that the continuous part of the evolution is reset to the new value $\boldsymbol{M}_{i,j}\boldsymbol{x}(\tau)$. This linear dependency is a restriction of a more general model where the reset function is independent from $\boldsymbol{x}(\tau^-)$.

Nevertheless this framework is able to model several interesting cases: projection, stretching/contraction of the norm, change of coordinates and, obviously, state continuity, obtained by using $\boldsymbol{M}_{i,j} = \boldsymbol{I}_n$ (the identity matrix).

Another crucial reason why we referred to this model is that we will present a procedure (extensively described in Chapters 4,5,6,7) whose logics are based on the preservation of a linearity and quadratic property of appropriate functions during the hybrid evolution.

Case 2: τ is not a switching instant.

In this case no event driven evolution occurs. Suppose that $i(\tau) = i$, then, trivially,

$$i(\tau + d\tau) = i,$$

i.e., the evolution keeps evolving in the same location i.

The continuous part of the evolution is governed by the linear affine differential equation associated to location i, $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_i\boldsymbol{x}(t)+\boldsymbol{f}_i$, hence the new value $\boldsymbol{x}(\tau+d\tau)$ can be obtained numerically or by simple integration.

Note that in a more general framework it is possible to consider modes of the form $\dot{\boldsymbol{x}}(t) = f_i(t, \boldsymbol{x}, \boldsymbol{u})$, where $\boldsymbol{u}$ is a continuous control input.

However the LTI affine autonomous ($\boldsymbol{u} = \boldsymbol{0}$) case allows the development of a numerically viable procedure, described in the rest of this thesis, that permits to design the discrete part of the hybrid evolution $i(t)$ in feedback form.

To complete the description of the S model used in this thesis we finally describe an additional constraint, namely, the *minimum permanence time* in each location.

Definition 3.7 (Minimum permanence time) *Once entered in a location i we cannot leave it before a* minimum permanence time $\delta_{\min}(i) \geq 0$ *has elapsed.* ■

This is a common constraint in many real applications: $\delta_{\min}$ may be the time necessary to control an actuator, or it may be the scan time of a PLC that triggers the switches, or even the delay of a signal propagation in a distributed system or of the measuring instruments.

If a model admits a minimum permanence time, then undesirable behaviors, such as Zeno, that may arise when more than one switches in 0 time are permitted, are avoided.

In Chapters 4 and 6 we consider the S as in Definition 3.2. Therein we provide a method, based on dynamic programming arguments that enables one to design a state feedback control law of the discrete part of the hybrid evolution $i(t)$, by minimizing a performance index defined in the sequel.

In particular in Chapter 4 this law is obtained for a finite number N of admissible switches, while in Chapter 6 this hypothesis is relaxed, thus also an infinite number of switches is admitted.

3.3.1 Particular switched systems

We define in this section two special cases of the S defined above. One is the switched system that only admits a *fixed mode sequence*.

Definition 3.8 (S fixed mode sequence) *We denote by SF, S* with fixed mode sequence, *the particular class of S such that* $\forall\ i\ \mathcal{S}$ *it holds*

$$|succ(i)| \leq 1.$$

■

The importance of this special subclass of S, wrt the given optimization problem, is detailed in Section 4.6.1. In fact its simple structure implies that the DOF in every switch is 0.

In other words, given the initial location $i(t = 0)$ for this SF, the mode sequence is univocally determined, because each location admits at most one successor.

This implies that the design of $i(t)$ is drastically simplified by the fact that the controller can only choose the switching instant, while the next location is evidently constrained, in force of the fact that $succ(i)$ is a singleton or $\emptyset$.

Moreover this case is of historical relevancy, because it originally [49] gave birth to the switching table procedure (STP) extensively described in this thesis.

Figure 3.2 shows some possible SF.

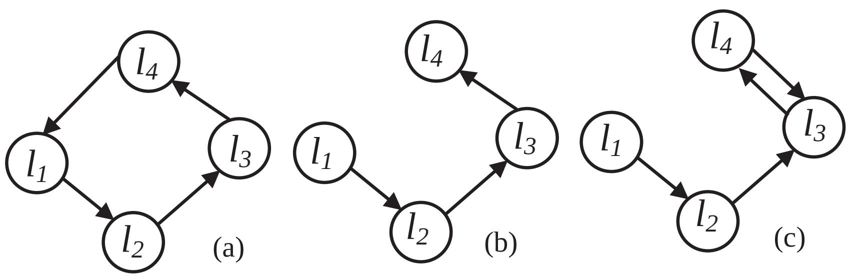


Fig. 3.2. *Oriented graphs of three possible SF.*

The other crucial case is the S that admits any arbitrary mode sequence. This model was originally studied in [9] and it is equivalent to a S where the switching sequence is completely unconstrained, i.e., from every location i it is possible to reach with one and only one switch every other location j of the S. Formally it is the particular S, namely SA, defined as follows.

Definition 3.9 (S arbitrary mode sequence) *We denote by SA, S* with arbitrary mode sequence, *the particular class of S such that* $\forall\ i\ \in\ \mathcal{S}$ *it holds*

$$succ(i) \equiv \mathcal{S} \setminus \{i\}.$$

■

This particular structure models all physical systems where it is possible (for safety or constructive point of view) to switch indifferently from one mode to another. Moreover we observe that the oriented graph of a SA is completely connected.

Figure 3.3 shows some possible SA's.

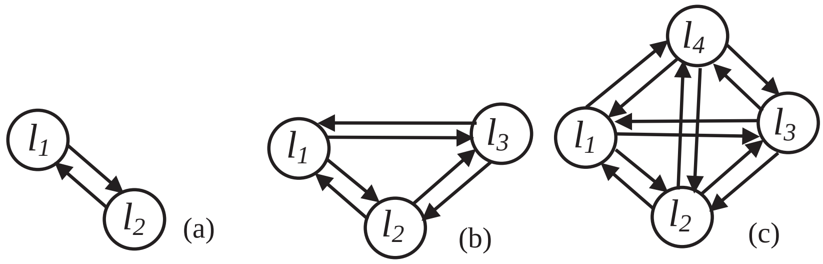


Fig. 3.3. *Oriented graphs of three possible SA's.*

We finally denote by $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ the particular SA such that the following restrictions are given: for all $i \in \mathcal{S}$

- $\delta_{\min}(i) = 0$;
- $\boldsymbol{f}_i = \mathbf{0}$.

This special class is considered in Chapter 7, where the notion of stability and stabilizability via optimal control is studied.

3.4 Optimal control problem for S

Before define the problem formulation we give some preliminary definitions.

3.4.1 Preliminary definitions

Definition 3.10 (Annexed weights) *Given a S as in Definition 3.2 we associate to each $i \in \mathcal{S}$ a matrix $\boldsymbol{Q}_i \geq \mathbf{0}$ and to each $e_{i,j},\ i \neq j \in \mathcal{E}$ a real constant $H_{i,j} \geq 0$ and $H_{i,i} = 0$.* ■

The matrices $\boldsymbol{Q}_i$ represent the quadratic weight of the continuous part $\boldsymbol{x}(t)$ of the hybrid evolution, i.e., whenever $i(t) = i$ the continuous state $\boldsymbol{x}(t)$ is weighted by the quadratic form $\boldsymbol{x}'(t)\boldsymbol{Q}_i\boldsymbol{x}(t)$.

The numbers $H_{i,j}$ weight the discrete part of the hybrid evolution, i.e., whenever a switch from location i to location j occurs, a cost $H_{i,j}$ is associated.

Property 3.1 *Consider a hybrid evolution $(\boldsymbol{x}(t), i(t))$, $t \in [0, +\infty)$, the function $i(t)$ is* piecewise constant. □

Proof. This is obvious, in fact the function $i(t)$ represents the values of the location indexes visited by the automaton during the evolution. ■

We now define two sets that are crucial in the development of this research.

Definition 3.11 (Sequence of switching instants) *We define the* sequence of switching instants *the set*

$$\mathcal{T} \equiv \{\tau_1, \tau_2, \ldots, \tau_k, \ldots\}$$

with $0 \leq \tau_1 \leq \tau_2 \leq \ldots \leq \tau_k \leq \ldots \leq +\infty$ the time instants at the occurrence of a switch. ■

Definition 3.12 (Sequence of indexes) *We define the* sequence of indexes *the set*

$$\mathcal{I} \equiv \{i_0, i_1, \ldots, i_k, i_{k+1} \ldots\}$$

with $i_{k+1} \in succ(i_k)$, $k \in \mathbb{N} \cup \{0\}$, *the list of values assumed by* $i(t)$ *when* $t \in [\tau_k, \tau_{k+1})$. ■

Figure 3.4 visualizes immediately the meaning of the definitions above. In this figure it is $\mathcal{I} \equiv \{1, 4, 3, 2, 3\}$ and $\mathcal{T} \equiv \{\tau_1, \tau_2, \tau_3, \tau_4\}$.

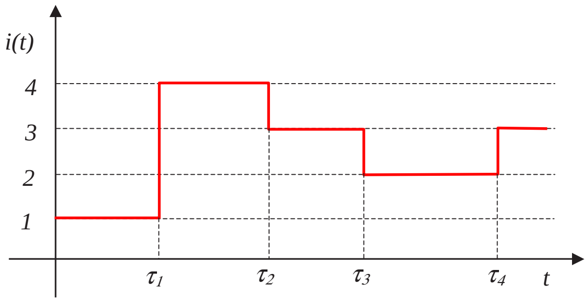


Fig. 3.4. *Example of the discrete part of evolution* $i(t)$. *The sets* $\mathcal{T}$, *Definition 3.11, and* $\mathcal{I}$, *Definition 3.12, can be extracted from the* piecewise constant *function* $i(t)$.

Definition 3.13 (Cost of the hybrid evolution) *Given a hybrid evolution* $(\boldsymbol{x}(t), i(t))$, $t \in [0, +\infty)$, *the cost of the evolution is given by*

$$F(\mathcal{I}, \mathcal{T}) = \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k \in \mathbb{N}} H_{i_{k-1}, i_k}. \quad (3.1)$$

■

Note that the semi positiveness of $\boldsymbol{Q}_i$ and $H_{i,j}$ make the cost function $F(\mathcal{I}, \mathcal{T})$ physically significant.

Remark 3.1 (A physical interpretation of the cost) *In specific models, for instance when the variable* $\boldsymbol{x}$ *represents the* position *and the* velocity *of a point of mass* m *and* $\boldsymbol{Q}$ *is diagonal, this functional is proportional to the total amount of kinetic and elastic energy spent during the motion.*

Generally speaking the LQR performance indexes are well suited to model energy consumptions. Nevertheless we do not consider here any further physical interpretations but rather we will care of the problem in a more abstract way. ■

The objective of this research is to provide a numerically viable procedure that allows the minimization of the functional (3.1), over the design variables $\mathcal{I}, \mathcal{T}$ derived from the switching signal $i(t)$.

This is equivalent to say that, given a switched system S, an annexed optimal control problem $\text{OP}_N(S)$, an initial hybrid state $(\boldsymbol{x}(0), i(0))$ we design the function $i(t)$ that minimizes the functional (3.1).

3.4.2 Finite number of switches $N < \infty$

We consider initially the minimization problem when $|\mathcal{I}| = N + 1 < \infty$. This problem is considered and solved in Chapter 4

In this case

- $\mathcal{T} \triangleq \{\tau_1, \ldots, \tau_N\}$ is a *finite* sequence of switching times;
- $\mathcal{I} \triangleq \{i_0, \ldots, i_N\}$ is a *finite* sequence of modes.

The functional (3.1) takes the form

$$F(\mathcal{I}, \mathcal{T}) = \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k=1}^{N} H_{i_{k-1},i_k}. \tag{3.2}$$

and the problem is defined as follows.

Definition 3.14 (Optimal control problem for a S) *We define the* optimal control problem for a switched system S *OP$_N$(S)as*

$$J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I}, \mathcal{T}) \triangleq \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k=1}^{N} H_{i_{k-1},i_k} \right\}$$

$$\begin{array}{lll} \text{s.t.} & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)} & \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 & \\ & i(t) = i_k \quad \textit{for } \tau_k \le t < \tau_{k+1} & k = 0, \ldots, N \\ & i_{k+1} \in succ(i_k) & k = 0, \ldots, N \\ & \tau_0 = 0, \ \tau_{N+1} = +\infty & \\ & \tau_{k+1} \ge \tau_k + \delta_{\min}(i_k) & k = 0, \ldots, N \\ & \boldsymbol{x}(\tau_k^+) = \boldsymbol{M}_{i_{k-1},i_k}\boldsymbol{x}(\tau_k^-) & k = 1, \ldots, N \end{array} \tag{3.3}$$

where the meaning of all terms has been extensively described in Section 3.4.1. ■

In a similar manner we may define OP$_N$(SA) and OP$_N$(SF), i.e., the optimal control problem of the particular switched systems SA and SF defined in Section 3.3.1.

The objective of the research described in Chapter 4 is to solve the above OP$_N$(S) for a given S.

Note that we considered

$$\tau_{k+1} \ge \tau_k + \delta_{\min}(i_k),$$

according to Definition 3.7.

The cost functional consists of two components: a quadratic cost that depends on the time evolution (the integral) and a cost that depends on the switches (the sum).

In Figure 3.5(a) a graphical meaning of the entries of Problem (3.3) are given for $N = 4$[1].

Note that in the last equation of (3.3), it is implicitly contained the following remark (see also Figure 3.5(b))

Remark 3.2 *Assume that the model of the system does not require a minimum permanence time in location i_k, i.e., for some $k = 0, \ldots, N$, $\delta_{\min}(i_k) = 0$. This allows*

[1] Clearly the space of the plot does not coincide with the state space.

the occurrence of simultaneous switches from locations i_{k-1} to i_{k+1} (or an immediate switch to i_1, whenever the case $k=0$ is considered), because the OP$_N(S)$ allows the solution $\tau_k = 0$, by virtue of $\tau_{k+1} \geq \tau_k + \delta_{\min}(i_k) = \tau_k$. Thus it may be possible that the optimal solution is to remain in location i_{k-1} and then switch immediately at time τ_k to location i_{k+1}. In such case it holds $\tau_k = \tau_{k+1}$, and consequently

$$\boldsymbol{x}(\tau_k^+) = \boldsymbol{x}(\tau_{k+1}^-)$$

(Figure 3.5(b)). Briefly the $k-th$ switch has no effect on the continuous time evolution, but it does on the discrete behavior of the automaton. In fact

$$\begin{array}{l} \boldsymbol{x}(\tau_k^+) = \boldsymbol{M}_{i_{k-1},i_k}\boldsymbol{x}(\tau_k^-) \\ \boldsymbol{x}(\tau_{k+1}^+) = \boldsymbol{M}_{i_k,i_{k+1}}\boldsymbol{x}(\tau_{k+1}^-) \end{array}$$

and consequently

$$\boldsymbol{x}(\tau_{k+1}^+) = \boldsymbol{M}_{i_k,i_{k+1}}\boldsymbol{M}_{i_{k-1},i_k}\boldsymbol{x}(\tau_k^-).$$

Similar considerations should be done for the switching costs.

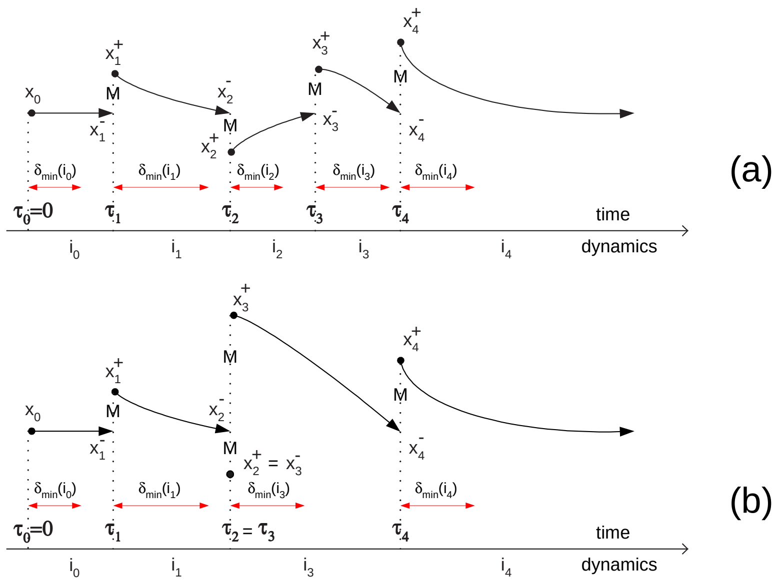


Fig. 3.5. *(a): sketch of a dummy evolution: explanation of the variables introduced in Problem (3.3); (b): the same evolution with the degeneracy of the second switch and its effect on the state jump. The same jump matrix $\boldsymbol{M}$ has been used for all switches.*

We might provide an equivalent form of the Problem 3.3 based on the time interval, rather than on absolute times. By letting

$$\varrho_k \triangleq \tau_k - \tau_{k-1} \geq \delta_{\min}(i_{k-1})$$

be the time interval elapsed between two consecutive switches, $k = 1, \ldots, N$, the Problem (3.3) can be rewritten as

$$J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ \sum_{k=0}^{N} \left[\boldsymbol{x}_k' \bar{\boldsymbol{Q}}_{i_k}(\varrho_{k+1}) \boldsymbol{x}_k + \bar{\boldsymbol{c}}_{i_k}(\varrho_{k+1}) \boldsymbol{x}_k + \bar{\alpha}_{i_k}(\varrho_{k+1}) \right] + \sum_{k=1}^{N} H_{i_{k-1},i_k} \right\}$$
$$\begin{aligned} \text{s.t. } & \boldsymbol{x}_{k+1} = \boldsymbol{M}_{i_k,i_{k+1}} \bar{\boldsymbol{A}}_{i_k}(\varrho_{k+1}) \boldsymbol{x}_k + \bar{\boldsymbol{f}}_{i_k}(\varrho_{k+1}),\ k = 0, \ldots, N-1 \\ & \boldsymbol{x}_0 = \boldsymbol{x}(0) \end{aligned} \tag{3.4}$$

where

$$\begin{aligned} \bar{\boldsymbol{A}}_i(\varrho) &\triangleq e^{\boldsymbol{A}_i \varrho}, \\ \bar{\boldsymbol{f}}_i(\varrho) &\triangleq \int_0^{\varrho} \bar{\boldsymbol{A}}_i(t) \boldsymbol{f}_i dt, \end{aligned} \tag{3.5}$$

and

$$\begin{aligned} &\bar{\boldsymbol{Q}}_i(\varrho) \triangleq \int_0^{\varrho} \bar{\boldsymbol{A}}'(t)\ \boldsymbol{Q}\ \bar{\boldsymbol{A}}(t) dt \\ &\bar{\boldsymbol{c}}_i(\varrho) \triangleq 2\boldsymbol{f}'\ \int_0^{\varrho} \left(\int_0^t \bar{\boldsymbol{A}}'(\tau) d\tau \right) \boldsymbol{Q}\ \bar{\boldsymbol{A}}(t) dt \\ &\bar{\alpha}_i(\varrho) \triangleq \boldsymbol{f}'\ \left[\int_0^{\varrho} \left(\int_0^t \bar{\boldsymbol{A}}'(\tau) d\tau \right) \boldsymbol{Q} \left(\int_0^t \bar{\boldsymbol{A}}(t) d\tau \right) dt \right]\ \boldsymbol{f} \end{aligned}$$

can be obtained by simple integration and linear algebra, as reported in Appendix B, or even resorting to numerical integration.

The approach of the solution, called *switching tables procedure* STP, is described in Chapter 4.

3.4.3 Infinite number of switches $N = \infty$

We now consider the case where the number of allowed switches can be infinite. In this case $|\mathcal{I}| = \infty$.

This problem is analyzed in Chapter 6 and 7[2].

Since we are dealing with infinite number of switches, we assume that all switching costs $H_{i,j}$ are null, i.e., we do not consider the cost of the event driven evolution.

Moreover we did not modelled the state jumps, hence $\forall\, i, j\ \in\ \mathcal{S}\ \boldsymbol{M}_{i,j} = \boldsymbol{I}_n$.

We can define the optimal control problem of a switched system with infinite number of switches as follows.

Definition 3.15 (Infinite OP$_\infty$(S)) *The* optimal control problem of a switched system with infinite number of switches*, OP$_\infty$(S), is*

$$\begin{aligned} & J_\infty^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t) \boldsymbol{Q}_{i(t)} \boldsymbol{x}(t) dt \right\} \\ & \text{s.t. } \ \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)} \boldsymbol{x}(t) + \boldsymbol{f}_{i(t)},\ \ \boldsymbol{x}(0) = \boldsymbol{x}_0,\ \ i(0) = i_0 \\ & \qquad i(t) = i_k\ \in\ succ(i_{k-1}) \text{ for } \tau_k \leq t < \tau_{k+1}, \\ & \qquad \tau_{k+1} \geq \tau_k + \delta_{\min}(i_k), \end{aligned} \tag{3.6}$$

$k\ \in \mathbb{N}$. *The initial state $\boldsymbol{x}_0$ and the initial location i_0 are given.* ■

The control variables are $\mathcal{T}$ and $\mathcal{I}$, where $\mathcal{T}$ is the set of switching times and $\mathcal{I}$ is the sequence of indices associated to the function $i(t)$. All the terms appearing in this section have been extensively described in Section 3.4.1 and 3.4.2.

[2] In Chapter 7 we studied the problem in the simpler case of a SA.

3.5 Definition of hybrid automata HA

In Definition 3.1 the notion of GHA is introduced, as it appears in the current literature [2], in a very general form.

Here we will define the HA as a particular case of the hybrid automaton defined in Definition 3.1 that is considered in Chapter 5. We indicate with the acronym HA the particular GHA that has been studied in this research.

Definition 3.16 (Hybrid automaton) *The* hybrid automaton HA *considered in Chapter 5 is the tuple* $HA = (\mathcal{L}, act, inv, \mathcal{E}, \mathcal{M})$ *whose entries have the following meaning:*

— $\mathcal{L}$ *is a finite set of locations indexed by* $i = 1, \ldots, s$.

— $act : \mathcal{L} \rightarrow$ Inclusions *is a function that associates to each location* i *a differential equation of the form*

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{f}_i.$$

— $inv : \mathcal{L} \rightarrow$ Invariants *is a function that associates to each location* i *an invariant* $inv_i \subseteq \mathbb{R}^n$ *such that* $\boldsymbol{x} \in inv_i$.

— $\mathcal{E} \subset \mathcal{L} \times Guards \times \mathcal{L}$ *is the set of edges. The edge* $e_{i,j}$ *is enabled when the current location is* i *and the current continuous state is* $\boldsymbol{x} \in g_{i,j}$*: it may fire reaching the new location* j.

— *A linear jump relation is* $\mathcal{M} \subset \mathbb{R}^n \times \mathbb{R}^n$ *associated to an edge* $e_{i,j}$*. When the edge fires,* $\boldsymbol{x}$ *is reset to* $\tilde{\boldsymbol{x}} = \boldsymbol{M}_{i,j}\boldsymbol{x}$*, where* $\boldsymbol{M} \in \mathbb{R}^{n \times n}$.

■

The classic definition of GHA [86] is more general than the one considered here because: the activity set may be a differential inclusion rather than a linear differential equation; the jump relation may be arbitrary and not necessarily defined by a matrix $\boldsymbol{M}$.

In consistency with the S we can define:

- the *state* of the HA as the pair $(\boldsymbol{x}, i)$ where $\boldsymbol{x} \in \mathbb{R}^n$ is the continuous state, and the index i identifies the discrete location i, as in Definition 3.5;
- the *evolution* of the HA $(\boldsymbol{x}(t), i(t))$ as in Definition 3.6;
- the minimum permanence time in each location, as in Definition 3.7.

We can also denote by

$$s = |\mathcal{L}|,$$

and

$$\mathcal{S} \equiv \{1, 2, \ldots, s\}$$

the set of indexes of the locations.

Note that the presence of guards and invariants in the model definition are crucial in the dynamical behavior of the discrete part of the hybrid state. In Definition 3.4 it was defined the set $succ(i)$ as a function of the current location i. The presence of guards and invariants leads us to reconsider the set $succ$ as function of both the continuous and the discrete part of the hybrid state.

Formally

$$succ : \mathbb{R}^n \times \mathcal{S} \;\rightarrow\; 2^{\mathcal{S}}. \tag{3.7}$$

We will separately analyze two cases.

The former is the case where the HA may exhibit *internally forced* switches, i.e., there is a subset of edges that, according to the value of the current hybrid state $(\boldsymbol{x}, i)$ may fire *autonomously*[3]. In this case *uncontrollable* switches may occur.

We will call this model as *autonomous HA*, AHA.

The latter is the case where all switches are controllable, but the set of possible successors of a current location i is dependent on the continuous state space. We will call this model as *constrained HA*, CHA.

Before proceeding further in the formal properties and restrictions of AHA and CHA it is important to recall some basic definitions.

Definition 3.17 (Invariant set) *An* invariant set inv_i, $i \in \mathcal{S}$, *is*

$$inv_i \subseteq \mathbb{R}^n,$$

such that if $\boldsymbol{x} \in inv_i$ *then the hybrid evolution* $(\boldsymbol{x}(t), i(t))$ *is allowed within location* i. ■

Definition 3.18 (Guard set) *A* guard set $g_{i,j}$, $i, j \in \mathcal{S}$, *is*

$$g_{i,j} \subseteq \mathbb{R}^n,$$

such that if $\boldsymbol{x} \in g_{i,j}$ *then the edge* $e_{i,j} \in \mathcal{E}$ *is enabled and it may fire.* ■

3.6 The considered cases of HA and annexed OP

As described in the introduction of this chapter, the presence of invariant sets and guards associated to edges influences the behavior of the HA, and consequently the problem formulation and its solution should be described consistently.

More precisely the presence of these sets have an effect on the switching scheduling, and thus on the designing of the control policy.

It is fundamental for the design of the control law, that the system is *deterministic*, i.e., the hybrid evolution $(\boldsymbol{x}(t), i(t))$ is exactly known for any given initial state.

Once this is guaranteed we may analyze two different interpretations of the switching constraints. In particular two classes of HA have been considered, and for each of them we applied the STP as described in Chapter 5.

In the course of this research we considered the annexed optimal control problem to the HA but, in difference with the switched system, we limited the study to the case where the total number of available controllable switches is limited to N.

Thus, a natural extension is to relax this restriction and consider, also for the HA the problem of infinite number of switches.

In fact the study of OP_∞(CHA), at least in restrictive conditions, is amongst the work in progress. Our recent results are briefly described in the Conclusions of this thesis.

3.6.1 Definition of autonomous hybrid automaton AHA

This case considers an *autonomous* HA, meaning that this system is subject to sequences of autonomous switches. In other words, not only the time driven evolution $\boldsymbol{x}(t)$ is uncontrolled (we only studied hybrid systems whose continuous control $\boldsymbol{u} = \mathbf{0}$), but also the discrete event evolution $i(t)$ is subject to autonomous behaviors according to subsets (named as *guards*) of the state space .

[3]Some authors [100] call the systems with this behavior as *switching systems*.

This class of systems, also denoted by *switching systems* [100], perform switches *autonomously* or so called *internally forced*[4], meaning that some switches may occur without any external control input, but merely according to the value of the continuous state $\boldsymbol{x}$.

Note that this is critical in the study of the hybrid systems. In fact there are several examples in literature, see for instance [17], where the presence of internally driven switches brings easily to instability.

An AHA is a particular $HA = (\mathcal{L}, act, inv, \mathcal{E}, \mathcal{M})$ whose set $\mathcal{E}$ satisfies the following Definitions 3.19 and 3.20, and whose guards satisfy Assumption 3.1.

Definition 3.19 (Set of controllable edges) *For each location* $i \in \mathcal{S}$*, the* set of controllable edges $\mathcal{E}_{i,c}$ *is the subset of all its output edges* $\mathcal{E}_i$ *such that*

$$\mathcal{E}_{i,c} = \{e \in \mathcal{E}_i \mid g_e = inv_i\}.$$

■

Definition 3.20 (Set of uncontrollable edges) *For each location* $i \in \mathcal{S}$*, the* set of uncontrollable edges $\mathcal{E}_{i,a}$ *is the subset of all its output edges* $\mathcal{E}_i$ *such that*

$$\mathcal{E}_{i,a} = \{e \in \mathcal{E}_i \mid g_e \cap inv_i = \emptyset\} \tag{3.8}$$

■

We clarify that

$$\mathcal{E}_i = \mathcal{E}_{i,c} \cup \mathcal{E}_{i,a}. \tag{3.9}$$

Finally we give Assumption 3.1

Assumption 3.1 (Guards and invariants of the AHA) *All guards associated to edges within the set* $\mathcal{E}_{i,a}$ *are disjoint sets. Formally:*

$$\forall\, e,\ \hat{e} \in \mathcal{E}_{i,a} \text{ with } e \neq \hat{e},\ \ g_e \cap g_{\hat{e}} = \emptyset. \tag{3.10}$$

Moreover, we assume:

$$inv_i \cup \left(\bigcup_{e \in \mathcal{E}_{i,a}} g_e \right) = \mathbb{R}^n. \tag{3.11}$$

■

We call this HA *autonomous* because there is no continuous control input and the autonomous edges are uncontrollable.

Note that the above definitions and assumption on the structure of the edges and guards of an AHA have several implications.

- Firstly, given an edge $e_{i,j} = (i, g_{i,j}, h) \in \mathcal{E}_{i,a}$ from location i if the continuous state is $\boldsymbol{x} \in g_{i,j}$, then a switch to location j should immediately occur.
  In fact, according to equation (3.8), $\boldsymbol{x} \notin inv_i$ and the system cannot remain in location i. We may call the edge $e_{i,j} \in \mathcal{E}_{i,a}$ *autonomous* (or equivalently *uncontrollable*).
- Whenever the continuous state reaches the guard $g_{i,j}$, thus *enabling* the edge $e_{i,j}$, the discrete autonomous behavior of the system is *deterministic*, because no other switch may occur. In fact, if there exist another output edge $e_{i,k}$ (be it controlled or autonomous), then by Assumption 3.1 it holds $g_{i,j} \cap g_{i,k} \equiv \emptyset$.

[4]This terminology was firstly introduced by Xu and Antsaklis in [122].

- If the continuous state $\boldsymbol{x}$ evolves within a given discrete location i and there exists an output edge $e_{i,j} = (i, g_{i,j}, q) \in \mathcal{E}_{i,c}$ then the system may either switch to location j or may keep evolving within location i. We assume that the choice is made by a discrete controller.

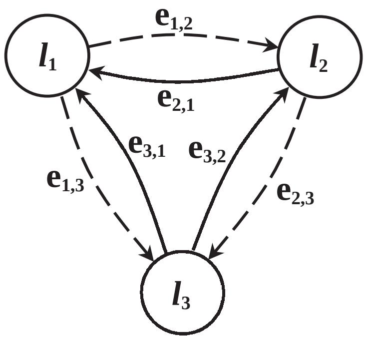


Fig. 3.6. *Oriented graph of the AHA considered in Example 3.1. The dashed arcs represent the autonomous edges, while the continuous arcs represent the controllable edges.*

Before providing some other useful definitions we provide a specific example, that shows the practical meaning of the described formalism of an AHA.

Example 3.1 *Let us consider the AHA whose* oriented graph, *Definition 3.3 is reported in Figure 3.6 where dashed arrows have been used to denote autonomous edges and continuous arrows have been used to denote controllable edges.*

The guards and invariant sets are depicted in Figure 3.7 In this particular $\mathbb{R}^2$ case, guards and invariants of the automaton are homogeneous. In such a case they may be easily described [90] as quadratic forms of $\boldsymbol{x}$. In particular, we assume that the guards associated to autonomous switches are

$$g_{1,2} = \{\boldsymbol{x} \in \mathbb{R}^2 |\, \boldsymbol{x}'\boldsymbol{G_{1,2}}\boldsymbol{x} \geq 0\},\ G_{1,2} = \begin{bmatrix} -0.2 & 0.6 \\ 0.6 & -1 \end{bmatrix}$$

$$g_{1,3} = \{\boldsymbol{x} \in \mathbb{R}^2 |\, \boldsymbol{x}'G_{1,3}\boldsymbol{x} \geq 0\},\ G_{1,3} = -\begin{bmatrix} 1 & 1.25 \\ 1.25 & 1 \end{bmatrix}$$

and

$$g_{2,3} = \{\boldsymbol{x} \in \mathbb{R}^2 |\, \boldsymbol{x}'G_{2,3}\boldsymbol{x} \geq 0\},\ G_{2,3} = \begin{bmatrix} -3 & 0.5 \\ 0.5 & 0 \end{bmatrix}$$

where $g_{1,2} \cap g_{1,3} = \emptyset$, thus verifying Assumption 3.1.

Consequently, by Assumption 3.1, the invariant sets may be defined as

$$\begin{array}{ll} inv_1 = \mathbb{R}^2 \setminus (g_{1,2} \cup g_{1,3}), & \\ inv_2 = \mathbb{R}^2 \setminus g_{2,3}, & inv_3 = \mathbb{R}^2, \end{array}$$

while the guards associated to controllable switches are

$$g_{2,1} = inv_2,\ g_{3,1} = g_{3,2} = inv_3.$$

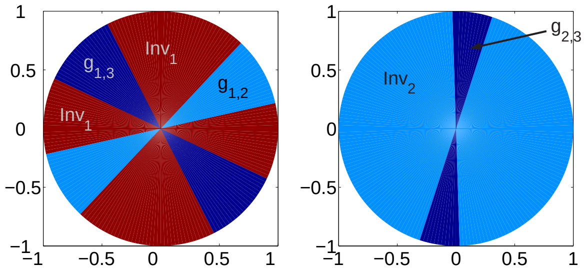


Fig. 3.7. *The guards and invariants of the AHA in Example 3.1.*

3.6.2 Dynamical behavior of an AHA

Let us assume that the current hybrid state is, at a given time t, $(\boldsymbol{x}, i)$. For this state there are two possible conditions:

1. $\boldsymbol{x} \in inv_i$.
2. $\boldsymbol{x} \notin inv_i$.

While the system is evolving in location i, the DOF of the external controller is limited to by $succ_c(i) \cup \{i\}$.

In other words the controller, while $\boldsymbol{x} \in inv_i$, can choose to switch amongst all controllable successors of location i or therein remain.

More precisely we give the following definition:

Definition 3.21 (Set of controllable successors) *The* set of controllable successors *of location i $succ_c(i)$ is defined as follows:*

$$succ_c(i) \equiv \{j \in \mathcal{S} \ : \ (i, g_{i,j}, j) \in \mathcal{E}_{i,c}\},$$

where the set $\mathcal{E}_{i,c}$ is taken as in Definition 3.19.

■

In case (2) the system must leave location i, in agreement with the definition of the invariant. Hence an autonomous switch will occur, and the systems falls *spontaneously* into another location, say j, which is univocally determined by the guard $g_{i,j}$.

Equivalently

$$\forall \ \boldsymbol{x} \ \in \mathbb{R}^n \setminus inv_i, \ \exists! \ j \ \in \ \mathcal{S}$$

such that $\boldsymbol{x} \in g_{i,j}$, where we indicate by $\exists!$ the "one and only one" exitance condition.

Now define the set

Definition 3.22 (Set of uncontrollable successors) *The* set of uncontrollable successors *of location i $succ_a(i)$ is defined as follows:*

$$succ_a(i) \equiv \{j \in \mathcal{S} \ : \ (i, g_{i,j}, j) \in \mathcal{E}_{i,a}\},$$

where the set $\mathcal{E}_{i,a}$ is taken as in Definition 3.20.

■

Consider for instance the location 2 in Example 3.1. Clearly the set $succ_c(2) \equiv \{1\}$ and $succ_c(2) \equiv \{3\}$.

We do not assume that the number of *autonomous* switches performed by an AHA is finite. Thus, according to the shape of the guards, the system may

- become unstable with no control;
- exhibit Zenoness.

In the sequel we provide sufficient structural conditions on the automaton graph that avoid these undesirable behaviors. Furthermore, in order to prevent non determinism, we assume that $succ_a(i)$ is a singleton. In the sequel we name this particular HA with the acronym AHA (Autonomous Hybrid Automaton).

3.6.3 Optimal control problem for AHA

We now define the optimal control problem annexed to the AHA. Before giving a formal definition of the problem it is helpful to introduce some additional notions.

Definition 3.23 (Sequence of autonomous switches) *Given a state* $(\boldsymbol{x}_0, i_0)$ *of an* AHA *we define the* sequence of autonomous switches

$$\sigma(\boldsymbol{x}_0, i_0) = \{(i_0, \theta_0), (i_1, \theta_1), ..., (i_h, \theta_h)\}$$

where i_k *is the index of the* $k-th$ *location visited from location* i_0 *and firing only autonomous edges of the* AHA*, while* $\theta_k \geq 0$ *is the time spent in location* i_k*. Formally the* θ_k*'s are time intervals such that for* $k = 0, \ldots, h$ *it holds:*

$$\begin{array}{ll} & \boldsymbol{x}_{k+1} = \boldsymbol{M}_{i_k, i_{k+1}} \bar{\boldsymbol{A}}_{i_k}(\theta_k) \boldsymbol{x}_k \\ \forall\, t \in [0, \theta_k) & \bar{\boldsymbol{A}}_{i_k}(t) \boldsymbol{x}_k \in inv_{i_k} \\ & \bar{\boldsymbol{A}}_{i_k}(\theta_k) \boldsymbol{x}_k \in g_{e_k} \\ & e_k = (i_k, g_{e_k}, i_{k+1}) \in \mathcal{E}_{i,a} \end{array} \tag{3.12}$$

with $\theta_h = +\infty$*.* ■

Note that the interval θ_k is the time it takes, once entered in location i_k, to reach the guard of the autonomous edge leading to location i_{k+1}. Therefore $\theta_k = 0$ implies that $\boldsymbol{x}_k \notin inv_{i_k}$.

Definition 3.24 (Bounded automata) *We say that an* AHA *is* bounded *if there exists an integer* $\hat{h} < +\infty$ *such that for all hybrid states* $(\boldsymbol{x}, i)$ *it holds*

$$|\sigma(\boldsymbol{x}, i)| \leq \hat{h}.$$

■

Note that this property implies that the automaton is not allowed to evolve autonomously for an infinite number of switches, thus avoiding undesired behaviors such as Zenoness [61] or instability [17].

Property 3.2 (Condition for bounded automata) *If the* oriented graph *of an* AHA *does not have cycles composed of only autonomous edges, then it is bounded.* □

Proof. The fact that no cycle composed of autonomous edges exists, is a sufficient (but not necessary) condition to imply that the bound $\hat{h}$ given in Definition 3.24 is less or equal to the length of the longest directed path containing only autonomous edges. ■

Note that the above property is structural of the oriented graph of the AHA, and it is very easy to verify.

As an example, we can immediately assert that the automaton in Figure 3.6 is bounded because it does not contain any cycle of autonomous (depicted with dashed arcs) edges.

We will only consider bounded AHA. Considering non bounded AHA can be meaningless, because of the potential instability of the system.

We shall now introduce a piecewise constant time function associated to the sequence $\sigma(\boldsymbol{x}_0, i_0)$.

Definition 3.25 (Indexes of autonomous trajectory) *The* indexes of autonomous trajectory *corresponding to a given sequence* $\sigma(\boldsymbol{x}_0, i_0) = \{(i_0, \theta_0), \ldots, (i_h, \theta_h)\}$ *is:*

$$\varphi_\sigma(t) = i_k, \textit{ if } t \in \left[\sum_{j=0}^{k-1} \theta_j, \sum_{j=0}^{k} \theta_j\right) \tag{3.13}$$

■

Example 3.1. Suppose that from a given AHA state $(\boldsymbol{x}, i)$ it has been computed the following sequence $\sigma(\boldsymbol{x}, i)$:

$$\sigma(\boldsymbol{x}, i) = \{(1, 2), (3, 1.5), (2, 2.5), (4, +\infty)\}$$

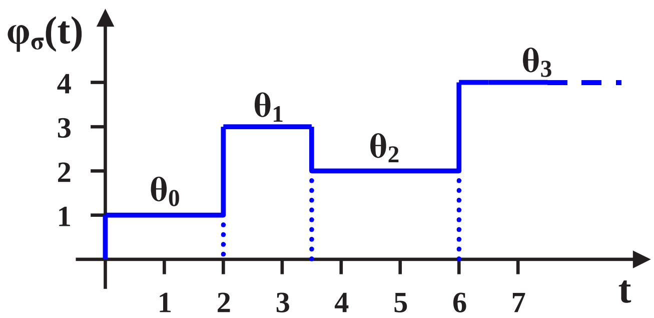


Fig. 3.8. *Function* $\varphi_\sigma(t)$ *of the autonomous sequence* $\sigma(\boldsymbol{x}, i) = \{(1, 2), (3, 1.5), (2, 2.5), (4, +\infty)\}$.

The associated function $\varphi_\sigma(t)$ is displayed in Figure 3.8.

The optimal control problem is based on the assumption that the discrete controller has at most N (fixed a priori) controllable switches available.

In analogy with the Definition 3.14 we define the optimal control problem associated to the AHA, and we indicate it with the acronym[5] OP(AHA).

For the explanation of the symbols given in the following definition concerning the optimal control refer to Section 3.4.1, and for the explanation of the symbols concerning the AHA refer to Section 3.6.1.

[5]Here the subscript N is useless because we do not deal with infinite number of switches.

Definition 3.26 (Optimal control problem for an AHA) *We define the* optimal control problem for an AHA *OP(AHA) in consistency with Definition 3.14, disregarding the switching costs,*

$$\begin{array}{rl} J_N^* \triangleq & \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt \right\} \\ & \text{s.t. } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)} \\ & \quad 0 = \tau_0 \leq \ldots \leq \tau_k \leq \ldots \leq \tau_{N+1} = +\infty \\ & \qquad \textit{(controlled switching times)} \\ & \quad i(0) = i_0 \quad \textit{(initial location)} \\ & \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \quad \textit{(initial state)} \\ & \quad i_k \in succ_c(i_{k-1}) \ \textit{(location reached after the } k-th \textit{ controlled switch)} \\ & \quad \boldsymbol{x}(\tau_k) = \boldsymbol{M}_{i_k,i_{k-1}}\boldsymbol{x}(\tau_k^-) \\ & \qquad \textit{(state reached after the } k-th \textit{ controlled switch)} \\ & \quad \sigma_k = \sigma(\boldsymbol{x}(\tau_k), i(\tau_k)) \quad \textit{(autonomous sequence)} \\ & \quad i(\tau_k+\theta) = \varphi_{\sigma_k}(\theta) \textit{ for } \theta \in [0, \tau_{k+1}-\tau_k) \\ & \qquad \textit{(autonomous index trajectory)} \end{array} \tag{3.14}$$

■

Here function $i(t)$ is composed of $N+1$ blocks delimited by the instants τ_k's where the controlled switches occur.

Each block is a *piecewise constant function*: steps internal to the interval $[\tau_k, \tau_{k+1})$ correspond to autonomous switches. More precisely, within the time interval $[\tau_k, \tau_{k+1})$ the function $i(t)$ is not constant but piecewise constant. In fact during the time elapsing $[\tau_k, \tau_{k+1})$ an autonomous evolution may occur.

The control variables in this problem are the sequence of *controlled switching times* $\mathcal{T} \triangleq \{\tau_1, \ldots, \tau_N\}$, and the *sequence of location indices* associated with controllable switches $\mathcal{I} \triangleq \{i(\tau_1), \ldots, i(\tau_N)\}$.

We want now to characterize some control problems such that the optimal cost is finite.

Definition 3.27 (Ultimate stability) *A location i of a bounded AHA is ultimately stable if $\forall \boldsymbol{x} \in inv_i$ the associated sequence $\sigma(i, \boldsymbol{x})$ reaches a final dynamics i_h (that may depend on $\boldsymbol{x}$) such that $\boldsymbol{A}_{i_h}$ is strictly Hurwitz.* ■

Proposition 3.1 *A bounded AHA can be stabilized by a switching control law if from every location i not ultimately stable there exists at least a controlled edge leading to an ultimately stable location.* □

Proof. We show that from any initial state $(\boldsymbol{x}_0, i_0)$ it is possible to steer the continuous state to the origin. In fact from the initial state we can wait until the last location i_h of the sequence $\sigma(\boldsymbol{x}_0, i_0)$ is reached. Obviously if $\boldsymbol{A}_{i_h}$ is not Hurwitz then i_h is not ultimately stable, hence by assumption there exists a controllable switch that leads to an ultimately stable location. ■

Note that this proposition is a sufficient (but not necessary) condition for the existence of a stabilizing control law. In order to make the problem (3.14) solvable with finite cost J_N^*, we assume that all considered AHA satisfy Proposition 3.1.

Finally, in order to express in a more compact way the following results, we recall that for a linear time invariant system of dynamics $\boldsymbol{A}$ an integral like

$$\mathcal{J} = \int_{\tau}^{\tau+\Delta\tau} \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt \tag{3.15}$$

with $\boldsymbol{Q} \geq \mathbf{0}$ is a quadratic form

$$\mathcal{J} = \boldsymbol{x}'(\tau)\bar{\boldsymbol{Q}}(\Delta\tau)\boldsymbol{x}(\tau) \tag{3.16}$$

that can be computed numerically or analytically as in Appendix B.

3.6.4 Definition of constrained hybrid automaton CHA

We analyze now another particular class of HA as in Definition 3.16, whose edges are *all* controllable but their firability depends on the value of the continuous state space $\boldsymbol{x}$.

We suddenly state that, in opposition to the AHA, in this case there are no autonomous sequences of switches, thus the instability issue and the Zenoness are avoided *a priori*.

This aspect of the HA simplifies the notation and we may directly describe the dynamical behavior of the CHA without providing any further definition.

The development of the CHA was motivated by a particular case study, described in Chapter 5, where the model of the plant is subject to safety constraints on the continuous state space[6].

The verification of these safety constraints can be guarantee if the sequences of discrete outputs of the plant obey to certain specification.

A procedure, developed by *Raisch et al.* [85], based on the $l-$complete approximation, and suited to the HA framework by the work of *Gromov et al.* as in [32], see Appendix D, allows the conversion of this specifications on the outputs into the definition of the invariant set.

We describe the dynamical behavior from a *high-level* point of view, meaning by high-level that we assume that the guards and invariants of the HA are given, with no concerns on how they are generated.

What makes this model a particular case of the HA is the definition of the guards. In this case we define:

Definition 3.28 (Guards of a CHA) *We define the* guard of the CHA $g_{i,j}$, *associated to the edge* $e_{i,j} = (i, g_{i,j}, j) \in \mathcal{E}$ *as* $g_{i,j} \equiv inv_j$. ■

The edge $e_{i,j}$ is thus enabled provided that the current continuous state $\boldsymbol{x} \in inv_j$.

3.6.5 Dynamical behavior of a CHA

The behavior is described as follows.

We may initially define the set of successors[7], as a function of the hybrid state $(\boldsymbol{x}, i)$

Definition 3.29 (Set of successors) *We define the* set of successors *of the hybrid state* $(\boldsymbol{x}, i)$

$$succ(\boldsymbol{x}, i) \equiv \{j \in \mathcal{S} : \boldsymbol{x} \in g_{i,j}\}.$$

■

Let us assume that the current hybrid state is, at a given time t, $(\boldsymbol{x}, i)$. For this state there are two possible conditions:

[6] With safety constraint we mean that the continuous variables of the plant should never take dangerous values, think, for instance to the pressure in a boiler.

[7] For this case there is no real need to specify *controllable* successors, in fact there is no autonomous evolution.

1. $\boldsymbol{x} \in inv_i$.
2. $\boldsymbol{x} \notin inv_i$.

In case (1) the controller can choose to switch to anyone of the locations enabled by the guards according to the current value of the continuous state $\boldsymbol{x}$, or it can decide to remain in the current location i, since the invariant condition is verified. The DOF of the controller is, in case (1):

$$succ(\boldsymbol{x}, i) \cup \{i\}.$$

In case (2), the system must, by Definition 3.17, leave location i. Hence the DOF of the controller is

$$succ(\boldsymbol{x}, i).$$

Remark 3.3 (Blocking CHA) *Case (2) offers a potentially blocking CHA. This can be obtained whenever the current location is i, $\boldsymbol{x} \notin inv_i$ and $succ(\boldsymbol{x}, i) \equiv \emptyset$.*

Nevertheless the procedure described in Appendix D provides the invariant sets that avoid this undesirable behavior. ■

Figure 3.9 better shows the significance of the $\boldsymbol{x}$-dependency on the set $succ(\boldsymbol{x}, i)$.

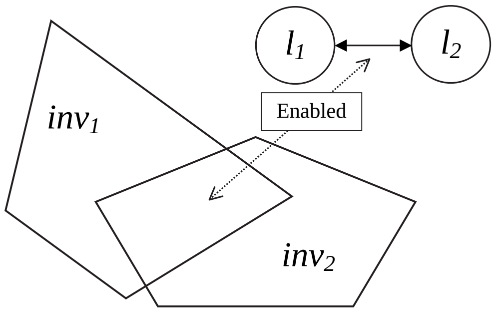


Fig. 3.9. *Meaning of state dependent successors. The edge between locations* 1 *and* 2 *and vice versa is enabled whenever* $\boldsymbol{x} \in inv_1 \cap inv_2$.

Note that the set $succ(\boldsymbol{x}, i)$ may be a singleton, thus the system may switch, to an extent, *autonomously*, because there is no other choice, but this is only an extreme case.

3.6.6 Optimal control for CHA

We now define the optimal control problem annexed to the CHA. As for the AHA we consider here only a finite number of available switches, namely N. We will call this problem with the acronym OP(CHA).

In consistency with the Definition 3.14 we define the optimal control problem associated to the CHA, and we indicate it with the acronym[8] OP(CHA).

For the explanation of the symbols given in the following definition concerning the optimal control refer to Section 3.4.1. For the explanation of the symbols concerning the CHA refer to Section 3.6.4.

[8]Here the subscript N is useless because we do not deal with infinite number of switches.

Definition 3.30 (Optimal control problem for a CHA) *We define the* optimal control problem for a CHA, *OP(CHA) in consistency with Definition 3.14, disregarding the switching costs,*

$$J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} F(\mathcal{I},\mathcal{T}) \triangleq \min_{\mathcal{I},\mathcal{T}} \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt$$

$$\text{s.t.} \qquad \begin{array}{l} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)} \\ i(0) = i_0 \quad \textit{(initial location)} \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \quad \textit{(initial state)} \\ i(t^+) \in \ succ(\boldsymbol{x}(t), i(t)) \cup \{i(t)\} \\ \boldsymbol{x}(t) \in inv_{i(t)} \subset \mathbb{R}^n, \ \ \forall\, t \geq 0 \end{array} \tag{3.17}$$

where $\mathcal{T}$ and $\mathcal{I}$ are defined in Definitions 3.11 and 3.12 with $\tau_k - \tau_{k-1} \geq \delta_{\min}(i_{k-1})$ $\forall\, k = 1, \ldots, N+1$, the minimum permanence time *imposed in each location.* ■

Note that this problem formulation is analogous to Problem (3.14), except for the presence of the invariant set, that restricts the set of actions of the controller in function of the current continuous state value $\boldsymbol{x}$.

3.7 Conclusions

In this chapter the models and the problems studied in this thesis have been formally defined. More precisely, starting from a general definition of hybrid automata, taken from [86] and [2], we restricted the attention to the linear affine particular cases.

We formally defined the linear affine switched system S, which is a system composed of several modes activated by a switching signal $i(t)$. We described two special cases of the S, namely the switched system that only admits a fixed mode sequence SF and the switched system that admits an arbitrary mode sequence SA.

We also defined the linear affine hybrid automaton HA, which is a switched system characterized by the presence of constraints on the state space, that may influence its dynamical behavior by the occurrence of autonomous switches (AHA), or they can restrict the action of an external discrete controller (CHA).

For this three classes of hybrid automata we described three different optimal control problems OP($\cdot$) of the form LQR, whose control variable is the switching signal $i(t)$, that is a piecewise constant function with *finite* segments. We point out that for the S we also defined the OP where $i(t)$ is a piecewise constant function with *infinite* segments.

4

Finite number of switches: switched systems

4.1 Introduction

The design of control laws for *hybrid systems* is a key issue in this research field. The peculiarities of these systems, merging a discrete event evolution with a continuous time evolution, may allow some particular behaviors, like chaotic trajectories [36, 25, 72] or the Zeno behaviors[1] as defined in [66, 59]. Several examples on Zenoness can be found in [61, 24]. Moreover some paradoxes, like the stabilizability properties described by [17], or the non uniqueness of an *execution* [65], make the object of this research particularly appealing.

In the previous chapter the type of systems considered in this thesis has been described in detail. We deal with hybrid systems composed of subsystems with linear time invariant and autonomous dynamics.

For this class of systems we consider the problem of finding an optimal switching strategy, i.e., an optimal control law, in feedback form. In short, we would like to find a procedure, that takes in input the continuous state space (in the sequel denoted by the variable $\boldsymbol{x}$) and the discrete state space (in the sequel indexed by i or j), and from this the switching strategy that minimizes a piecewise LQR performance index is suggested.

The procedure presented in this chapter represents the kernel of this research study. A formal presentation of the procedure will be given and supported by the help of specific examples.

However, for sake of clarity the procedure is not described here in its most general form. In particular we consider here the following restrictions:

– The system is allowed to switch to an adjacent location without constraints on the state space. In other words the continuous part of the hybrid state $(\boldsymbol{x}, i)$ has no effect on the switching strategy. This extension is considered in the next chapter.
– The total number of allowed switches is $N < +\infty$. This is to prevent undesirable behaviors, such as Zeno [66, 59]. Furthermore the procedure is developed time backward, thus it is based on the fact that there actually exists a "last switch". The extension to $N = +\infty$ is considered in Chapter 6.

The chapter is structured as follows. Initially the notions given in Chapter 3 are briefly summarized, focussing on the model and the problem under consideration. Then some theoretical results are presented. In particular these are fundamental to understand how the feedback control can law be constructed, why it is optimal, and

[1] A model of a hybrid system exhibits Zeno behavior when it performs an infinite number of transitions (equivalently, switches) in a finite time interval.

what its characteristics are. After a brief examination of the computational complexity, some examples follow.

In the last part of the chapter two special cases of the procedure are described. One is the *fixed mode sequence*, that was developed in [49]. The other is the *arbitrary mode sequence*, developed in [9]. What we find interesting in these cases is that they represent, to an extent, the extreme cases of the considered model, thus suggesting a simpler perspective. In the fixed mode sequence the sequence of locations visited during an evolution is pre-assigned, while in the arbitrary mode sequence the switching strategy has complete degree of freedom amongst all possible dynamics. The procedure presented here is a generalization of these cases, and some current research and results are still based on them, at least in an initial approach, thanks to their structural simplicity.

Finally we provide an example that introduces the case of infinite number of switches, more suitable for real life applications.

4.2 Linear affine switched system and optimal control

4.2.1 The linear affine switched system

We consider in this chapter the model defined in Section 3.3, namely *linear affine switched system*, $S = (\mathcal{L}, act, \mathcal{E}, \mathcal{M})$, in the sequel S, where, in consistency with Definition 3.2.

— $\mathcal{L}$ is a finite set of locations, indexed by $i = 1, \ldots, s$.

— $act : \mathcal{L} \to (\mathbb{R}^n \times \mathbb{R}^n)$ is a function that associates to each location i a linear affine differential equation, i.e., $\dot{\boldsymbol{x}} = \boldsymbol{A}_i \boldsymbol{x} + \boldsymbol{f}_i$.

— $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ is the set of edges. An edge $e_{i,j} = (i, j)$ is an edge from location i to location j, $i \neq j$.

— $\mathcal{M} : \mathcal{E} \to \mathbb{R}^{n \times n}$ associates to each edge $e \in \mathcal{E}$ a constant matrix in $\mathbb{R}^{n \times n}$. When the discrete state switches from location i to location j at time τ, the continuous state $\boldsymbol{x}$ is reset to $\boldsymbol{x}(\tau^+) = \boldsymbol{M}_{i,j} \boldsymbol{x}(\tau^-)$.

We denote by $\mathcal{S}$ the *set of indexes* associated to each location, and $s = |\mathcal{S}|$.

The considered system may be represented by an *oriented graph*, as in Definition 3.3.

The state of the S is the couple $(\boldsymbol{x}, i)$ where $i \in \mathcal{S}$ is the discrete location and $\boldsymbol{x} \in \mathbb{R}^n$ is the continuous state.

We assume that a *minimum permanence time* $\delta_{\min}(i) \geq 0$, as in Definition 3.7 is associated to each location i.

Moreover we recall the notion of the set $succ(i) \subset \mathcal{S}$, i.e., the set of location indexes reachable from location i by firing only one edge, formally defined in Definition 3.4.

4.2.2 Formulation of the optimal control problem

The objective is to solve the optimal control problem with an upper bound N on the number of the available switches, $\text{OP}_N(S)$, as in Definition 3.14 for the switched system S defined above.

We recall the problem formulation, as given in Definition 3.14:

$$
J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k=1}^{N} H_{i_{k-1},i_k} \right\}
$$

$$
\begin{array}{lll}
\text{s.t.} & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)} & \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 & \\
& i(t) = i_k \quad \text{for } \tau_k \le t < \tau_{k+1} & k = 0, \ldots, N \\
& i_{k+1} \in succ(i_k) & k = 0, \ldots, N \\
& \tau_0 = 0, \ \tau_{N+1} = +\infty & \\
& \tau_{k+1} \ge \tau_k + \delta_{\min}(i_k) & k = 0, \ldots, N \\
& \boldsymbol{x}(\tau_k^+) = \boldsymbol{M}_{i_{k-1},i_k}\boldsymbol{x}(\tau_k^-) & k = 1, \ldots, N
\end{array} \tag{4.1}
$$

where $\boldsymbol{Q}_i$ are positive semi-definite matrices and $\boldsymbol{x}_0$ is the initial state of the system.

In this optimization problem there are two types of decision variables, as in Definitions 3.11 and 3.12 respectively:

- $\mathcal{T} \triangleq \{\tau_1, \ldots, \tau_N\}$ is a finite sequence of switching times;
- $\mathcal{I} \triangleq \{i_0, \ldots, i_N\}$ is a finite sequence of modes.

Note that the cost (4.1) consists of two components: a quadratic cost that depends on the time evolution (the integral) and a cost that depends on the switches (the sum), where $H_{i,j} \ge 0$, $i,j \in \mathcal{S}$, is the cost for commuting from mode i to mode j, with $H_{i,i} = 0$, $\forall i \in \mathcal{S}$.

4.2.3 Fundamental assumption

The solution of problem (4.1) is finite, provided the following fundamental assumption:

Assumption 4.1

(i) there exists at least one location $i \in \mathcal{S}$ such that $\boldsymbol{A}_i$ is strictly Hurwitz, $\boldsymbol{f}_i = \boldsymbol{0}$;
(ii) if the initial location i_0 is imposed, than the number N of available switches is such that the location i must be reachable from i_0 in $k \le N$ steps.

■

In other words (i) states that there must exist at least one location in the automaton such that the corresponding differential equation has stability (in the Hurwitz sense) properties.

Moreover (ii) requires that this location i must be *reachable* within N switches, meaning that there exists at least an oriented path in the automaton graph, that brings from the initial location i_0 into i within N steps.

If i_0 is not assigned then (ii) can be relaxed.

In fact even in the worst case, i.e., $\forall\ j \ne i \ \in \ \mathcal{S}$

- the dynamics $\boldsymbol{A}_j$ isn't Hurwitz;
- there are no arcs entering location i, i.e., $\forall\ j \ \in \mathcal{S} \setminus \{i\},\ i \ \notin \ succ(j)$;

we can always choose $i_0 = i$.

In this worst case the problem OP$_N(S)$ admits, for any initial point $(\boldsymbol{x}_0, i_0)$, the unique trivial solution

$$
i(t) = i_0 = i, \ \ J_N^*(\boldsymbol{x}_0, i) = \boldsymbol{x}_0'\boldsymbol{Z}_i\boldsymbol{x}_0,
$$

where $\boldsymbol{Z}_i > \boldsymbol{0}$ is the unique solution of the Lyapunov equation

$$\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i = -\boldsymbol{Q}_i,$$

as described in Appendix A.3.

This solution chooses immediately the only stable dynamics and never switches from there.

4.2.4 Linear affine models

Before proceeding further, we observe that the original affine dynamics, modelled in Section 3.3,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)},\ i(t) \in \mathcal{S}$$

can be rewritten as a linear dynamics by simply augmenting the state space from $\mathbb{R}^n$ to $\mathbb{R}^{n+1}$:

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{x}(t) \\ \tilde{x}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{i(t)} & \boldsymbol{f}_{i(t)} \\ \boldsymbol{0}' & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \tilde{x}(t) \end{bmatrix} \tag{4.2}$$

with $\tilde{x}(0) = 1$. Note that the $(n+1)-th$ state variable $\tilde{x}(t)$ is a fictitious variable that does not influence the cost function, if the new weighting matrices are semi-definite positive matrices of the form

$$\begin{bmatrix} \boldsymbol{Q}_i & \boldsymbol{0} \\ \boldsymbol{0}' & 0 \end{bmatrix}$$

for all $i \in \mathcal{S}$.

Henceforth, wlg, the OP$_N(S)$ (4.1), is formally equivalent to an OP$_N(S')$, where all dynamics of S'have the form (4.2).

For this reason in the following we will talk only about linear systems in n dimensional space, meaning that it could also be an affine problem in $n-1$ dimensional space. Of course this advantage is significant only from a formal point of view; whenever the STP, later described, is implemented with affine modes it will be clearly specified. In fact, while the theoretical procedure is equivalent, some implementations *precautions* must be added, especially when discretizing the state space.

Note also that Assumption 4.1 is sufficient to ensure that the system is stabilizable on the origin (and hence that the OP we consider is solvable with a finite cost) but it is not strictly necessary. Consider in fact the following two particular cases.

4.2.5 Case 1

Assume that all dynamics $\boldsymbol{A}_i$ have a displacement term $\boldsymbol{f}_i \neq \boldsymbol{0}$ but at least one dynamics, say $\boldsymbol{A}_j$, is Hurwitz. One can make a state-coordinate transformation $\boldsymbol{x} \to \boldsymbol{z} - \boldsymbol{A}_j^{-1}\boldsymbol{f}_j$ and penalize — whenever mode i is active — the deviation from the target state through the quadratic term $(\boldsymbol{x} + \boldsymbol{A}_j^{-1}\boldsymbol{f}_j)'\boldsymbol{Q}_i(\boldsymbol{x} + \boldsymbol{A}_j^{-1}\boldsymbol{f}_j) = \boldsymbol{z}'\boldsymbol{Q}_i\boldsymbol{z}$.

Example 4.1 *Let us consider a model of a* boost converter*, inspired by [112], whose circuit is represented in Figure 4.1.*

The state of the system is

$$\boldsymbol{x} = \begin{bmatrix} u \\ i \end{bmatrix}.$$

This linear affine switched system has two possible modes, according to the position of the switch s. In particular, when the switch is open the DE of the system is:

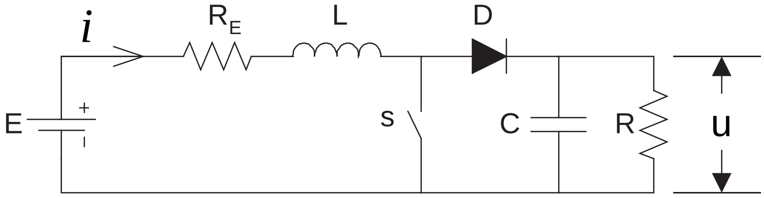


Fig. 4.1. *Boost converter.*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_1 \boldsymbol{x}(t) + \boldsymbol{f}_1 = \begin{bmatrix} -\dfrac{1}{RC} & \dfrac{1}{C} \\ \\ -\dfrac{1}{L} & -\dfrac{R_E}{L} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ \dfrac{E}{L} \end{bmatrix}, \tag{4.3}$$

and when the switch is closed

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_2 \boldsymbol{x}(t) + \boldsymbol{f}_2 = \begin{bmatrix} -\dfrac{1}{RC} & 0 \\ \\ 0 & -\dfrac{R_E}{L} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ \dfrac{E}{L} \end{bmatrix}. \tag{4.4}$$

This switched system does not apparently satisfy the Assumption 4.1, because both affine terms are non null.

Nevertheless, being both dynamics Hurwitz, it is possible to reformulate the problem with a state coordinate shift centered in one of the following equilibrium points:

$$\boldsymbol{x}_1 = \begin{bmatrix} \dfrac{R}{R_E + R} E \\ \\ \dfrac{1}{R_E + R} E \end{bmatrix}, \ \boldsymbol{x}_2 = \begin{bmatrix} 0 \\ \dfrac{E}{R_E} \end{bmatrix}.$$

More specifically we may consider the new variable

$$\boldsymbol{z} \triangleq \boldsymbol{x} + \boldsymbol{A}_2^{-1} \boldsymbol{f}_2 = \boldsymbol{x} + \begin{bmatrix} -RC & 0 \\ 0 & -\dfrac{L}{R_E} \end{bmatrix} \begin{bmatrix} 0 \\ \dfrac{E}{L} \end{bmatrix} = \boldsymbol{x} - \boldsymbol{x}_2.$$

Now the system becomes, by substitution,

$$\begin{cases} \dot{\boldsymbol{z}} = \boldsymbol{A}_1 \boldsymbol{z} + \begin{bmatrix} \frac{E}{R_E C} \\ 0 \end{bmatrix} \\ \dot{\boldsymbol{z}} = \boldsymbol{A}_2 \boldsymbol{z} \end{cases}$$

and it satisfies the Assumption 4.1, because $\boldsymbol{A}_2$ is Hurwitz and the new affine term of system 2 is null. ■

4.2.6 Case 2

Assume that $\boldsymbol{A}_j$ is a non Hurwitz diagonalizable matrix and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$ where for at least one eigenvalue, say λ_1, it is $Re(\lambda_1) < 0$. Then it is always possible to find a matrix $\boldsymbol{Q}_j \geq \boldsymbol{0}$ such that

$$\int_0^{+\infty} \boldsymbol{x}'\boldsymbol{Q}_j\boldsymbol{x}dt < +\infty.$$

In fact let $\boldsymbol{T} \ : \ \boldsymbol{T}^{-1}\boldsymbol{\Lambda}\boldsymbol{T} = \boldsymbol{A}_j$; obviously it holds (see Appendix B)

$$\int_0^{+\infty} \boldsymbol{x}'\boldsymbol{Q}_j\boldsymbol{x}dt = \boldsymbol{x}_0'\boldsymbol{T}' \int_0^{+\infty} \bar{\boldsymbol{\Lambda}}(t)(\boldsymbol{T}^{-1})'\boldsymbol{Q}_j\boldsymbol{T}^{-1}\bar{\boldsymbol{\Lambda}}(t)dt\,\boldsymbol{T}\boldsymbol{x}_0$$

where $\boldsymbol{x}_0 = \boldsymbol{x}(0)$ and $\boldsymbol{x} = \boldsymbol{x}(t) = \bar{\boldsymbol{A}}(t)\boldsymbol{x}_0$. Now, if we choose the n^2 entries of matrix $\boldsymbol{Q}_j$ such that

$$(\boldsymbol{T}^{-1})'\boldsymbol{Q}_j\boldsymbol{T}^{-1} = \begin{bmatrix} k & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix},$$

then, even if $\boldsymbol{A}_j$ is non Hurwitz

$$\int_0^{+\infty} \boldsymbol{x}'\boldsymbol{Q}_j\boldsymbol{x}dt = -\frac{k}{2Re(\lambda_1)}\boldsymbol{x}_0'\boldsymbol{T}'\boldsymbol{T}\boldsymbol{x}_0 < +\infty. \tag{4.5}$$

Example 4.2 *Let*

$$\boldsymbol{A} = \begin{bmatrix} 3 & 4 \\ 4 & -3 \end{bmatrix}.$$

The matrix $\boldsymbol{A}$ is non Hurwitz because $\boldsymbol{\Lambda} = diag\{-5, 5\}$. The state space transformation $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{x}$ that diagonalizes $\boldsymbol{A}$ is

$$\boldsymbol{T} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix}.$$

Note that $\boldsymbol{T}$ is an orthonormal matrix[2]*. Let us show now that there exists a symmetric semi-definite positive matrix*

$$\boldsymbol{Q} = \begin{bmatrix} q_1 & q \\ q & q_2 \end{bmatrix}$$

such that

$$\int_0^{+\infty} \boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}dt$$

is finite. It is sufficient to find a solution of

$$\boldsymbol{T}\boldsymbol{Q}\boldsymbol{T}' = \begin{bmatrix} k & 0 \\ 0 & 0 \end{bmatrix}.$$

The linear system above becomes

[2]Orthonormal matrix $\boldsymbol{T}$: $\forall\ i, j = 1, \ldots, n < \boldsymbol{t}_i, \boldsymbol{t}_j >= 0$ if $i \neq j$ and $< \boldsymbol{t}_i, \boldsymbol{t}_i >= 1$, where $\boldsymbol{t}_j$ are the columns of $\boldsymbol{T}$. For orthonormal matrices $\boldsymbol{T}^{-1} = \boldsymbol{T}'$.

$$\begin{cases} q_1 \ + 4q + 4q_2 = 5k \\ 2q_1 + 3q - 2q_2 = \ 0 \\ 4q_1 - 4q + \ q_2 \ = \ 0 \end{cases}$$

whose solution gives

$$\boldsymbol{Q} = \frac{k}{5}\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

which is indeed positive semi-definite. Now, according to equation (4.5), being $\boldsymbol{T}' = \boldsymbol{T}^{-1}$, it holds

$$\int_0^{+\infty} \boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}dt = \frac{k}{10}\|\boldsymbol{x}_0\|^2 < +\infty.$$

■

There are even other cases in which Assumption 4.1 may be relaxed preserving a finite value of the cost. The reasons that prompted us to state Assumption 4.1 are the following:

1. Albeit restrictive it is a structural property, thus easily verified;
2. Most of the particular cases such as Case 2 are in general practically irrelevant.

4.3 Switching Table Procedure

In this section we show how to solve the OP (4.1) or equivalently (3.4), for a given switched linear system $H = (\mathcal{L}, act, \mathcal{E}, \mathcal{M})$, as described in Chapter 3, when Assumption 4.1 is satisfied.

In particular we show that the optimal control law for the optimization problem described in the previous section takes the form of a state feedback, i.e., it is only necessary to look at the current system state $\boldsymbol{x}$ in order to determine if a switch from location i_{k-1} to i_k,or equivalently from linear dynamics $\boldsymbol{A}_{i_{k-1}}$ to $\boldsymbol{A}_{i_k}$, should occur.

Remark 4.1 *Before proceeding further we would like to clarify that in the general optimization Problem* (4.1)*, although the number of allowed switches is N, also solutions where only $m < N$ switches effectively occur. More precisely, the number N is an upper bound on the number of available switches. In the next chapters it will be proved that the total cost of the evolution is a decreasing monotone positive function of the number N.* ■

Let us recall the following definition:

Definition 4.1 (Partition of a set) *A* partition of a set *Ω into K subsets Ω_i, $i = 1, \ldots, K$, is such that*

$$\Omega = \bigcup_{i=1}^{K} \Omega_i$$

and

$$\Omega_i \bigcap \Omega_j \equiv \emptyset$$

$\forall\, i \neq j$. ■

The procedure described here considers the model described above (i.e., a switched linear autonomous system) with an annexed optimal control problem with infinite time horizon and finite number of switches. It constructs, for each location i and for the $k-th$ missing switch ($k = 1, \ldots, N$), a table $\mathcal{C}_k^i$ that partitions the state space $\mathbb{R}^n$ into s_i regions $\mathcal{R}_j$'s, where $s_i = |succ(i)| + 1$.

The control law is thus a set of $N \times s$ tables, where $s = |\mathcal{S}|$.

Remark 4.2 *In the following the symbol $\mathcal{C}_k^i$ (switching table) denotes a partition of $\mathbb{R}^n$ viewed from location i, when k switches are still available.* ■

Whenever $i_k = i$ the discrete external controller uses table $\mathcal{C}_k^i$ to determine if a switch should occur: as soon as the continuous state $\boldsymbol{x}$ reaches a point in the region $\mathcal{R}_j$ for a certain $j \in succ(i)$ a switch to mode $i_{k+1} = j$ will occur; on the contrary, no switch will occur while the system's state belongs to $\mathcal{R}_i$.

Example 4.3 *To better illustrate how these tables are used, we propose Figure 4.2. In this figure it is shown the table, obtained for a particular example (Section 4.5), to be used when location i is active and only 1 switch is still available. Whenever the continuous state $\boldsymbol{x}$ is in the orange area, then it is optimal to remain in location i.*

During the evolution with $\boldsymbol{A}_i$ the continuous state may cross the cyan or the yellow region. In this case the last switch should occur towards locations j or h respectively.

One may ask the meaning of the central area in blue (leading to location k). Clearly this area will be never entered from location i, as the state space $\boldsymbol{x}$ is a continuous function. Nevertheless it may be reached directly after the previous switch. Moreover in this particular example a value of $\delta_{\min}(i) \neq 0$ was considered. Thus immediately after the switch the controller is "blind" for the time $\delta_{\min}(i)$. This may cause the continuous state $\boldsymbol{x}$ evolution to silently cross the cyan area and then, when $\delta_{\min}(i)$ is elapsed, $\boldsymbol{x}$ may be in the blue area, thus forcing the controller to switch to k. ■

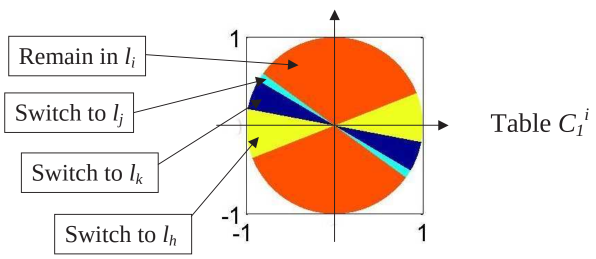


Fig. 4.2. *Partition of the state space $\mathbb{R}^2$ showing how the region $\mathcal{C}_1^i$ (of location i, when 1 switch is available) serves to locate the switching areas. The regions are colored only in the limited plot for obvious reasons, but they cover the entire space maintaining the conic shape.*

This is a fundamental result because it is well known that a state feedback control law has many advantages over an open-loop control law, including that the computation of the control law can be done off line as opposed to being performed on line. On line computations are burdensome, especially if a disturbance acting on the system may cause the system state to deviate from its expected value.

To prove this result, we show constructively how the tables $\mathcal{C}_k^i$ can be computed using a dynamic programming argument.

We first show how the tables $\mathcal{C}_1^i$ ($i \in \mathcal{S}$) for the last switch can be determined. Then, we show by induction how the tables $\mathcal{C}_k^i$ can be computed once the tables $\mathcal{C}_{k-1}^i$ are known.

For simplicity we also assume that dynamics are linear, because affine dynamics can be easily reduced to linear dynamics as shown in (4.2).

In particular we show that for a given mode $i \in \mathcal{S}$ and for a given switch[3] $k = 1, \ldots, N$ it is possible to construct a table $\mathcal{C}_k^i$ that partitions the state space $\mathbb{R}^n$ into $s(i) = |succ(i)| + 1$ regions $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{s(i)}$.

Thus for every mode and for each value of a variable k that we may call "switch counter" the state space is divided into several areas that suggest the optimal switching strategy.

Let us recall here another definition.

Definition 4.2 (Homogeneous function [37]) *A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is* homogeneous *of order m and we say* m-homogeneous *if $\forall\ \boldsymbol{x} \in X \setminus \mathbf{0}$ there exists a $\lambda \in \mathbb{R} \setminus 0$ such that*

$$f(\lambda \boldsymbol{x}) = \lambda^m f(\boldsymbol{x}).$$

■

We will often use the concept of homogeneity of a function. In particular let us observe the following property.

Property 4.1 *(1) Given a LTI autonomous systems of the form*

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$$

the flow function $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ is homogeneous of degree 1.
(2) The quadratic cost functions of the form

$$J(\boldsymbol{x}(0)) = \int_0^\tau \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t) = \boldsymbol{x}'(0)\bar{\boldsymbol{Q}}(\tau)\boldsymbol{x}(0)$$

are homogeneous functions of degree 2.

□

Proof. (1) Trivially follows from the linearity of the LTI systems;
(2) Trivially follows from the quadratic form of the cost functions and the linearity of the system.

■

Finally let us consider the concept of homogeneous region $\mathcal{R} \subseteq \mathbb{R}^n$ of the state space.

Definition 4.3 (Homogeneous region) *A region $\mathcal{R} \subseteq \mathbb{R}^n$ is* homogeneous *if, $\forall\ \boldsymbol{x} \in \mathcal{R}$ and $\forall\ \lambda \in \mathbb{R}$, it holds $\lambda \boldsymbol{x} \in \mathcal{R}$.* ■

We will indifferently use the word *homogeneous* region or *conic* region. A partition of $\mathbb{R}^n$ is entirely homogeneous if all its components $\mathcal{R}_i$ are homogeneous.

Example 4.4 *The partition depicted in Figure 4.2 is homogeneous.* ■

[3]The tables are numbered in anti-chronological order, i.e., the $k - th$ table indicates that k switches are missing, or equivalently, that $N - k$ switches have occurred

4.3.1 Computation of the Switching Tables

Assume generally that:

- the current state of the system is $(\boldsymbol{x}, i)$, where $\boldsymbol{x}$ indicates the continuous state, and i indicates the current location i, i.e., the discrete state;
- the number of missing switches is $k \geq 0$ out of N.

We provide here a method to calculate the switching tables that serve as feedback control law for the class of switched system described in Section 4.2.1, and generally defined in Chapter 3.

To this aim, assume that the switched system evolves according to the following schedule:

$$\begin{array}{l||c|c|c|c} \text{Time intervals} & \varrho_k & \varrho_{k-1} & \cdots & \varrho_0 \\ \hline \text{Indexes} & j_k & j_{k-1} & \cdots & j_0 \end{array} \tag{4.6}$$

with the following constraints:

$$\begin{array}{l} \varrho_0 = +\infty \\ \varrho_h \geq \delta_{\min}(j_h) \\ j_k = i \\ j_h \ \in \ succ(j_{h+1}) \cup \{j_h\} \\ h = 0, \ldots, k-1. \end{array} \tag{4.7}$$

Note that it must be $\boldsymbol{A}_{j_0}$ Hurwitz stable, in agreement with Assumption 4.1.

For sake of clarity we specify that the sequence 4.6 means that the switched system will evolve in location $j_k = i$ for a time ϱ_k, then it will switch to location j_{k-1} where it will remain for a time $\varrho_{k-1} \geq \delta_{\min}(j_{k-1})$ and so on. Finally it will reach the last location j_0 where it will terminate the evolution and it will remain forever, with $\varrho_0 = +\infty$.

Remark 4.3 *We decided to renumber the subscripts of the time intervals and of the location indexes. In particular in this paragraph it appeared more convenient to count them in time backwards. With this idea all definitions, properties and theorems that follow (given by induction) appear more readable.*

Nevertheless, once all things are proved, we will switch back to the previous (more natural) terminology, i.e., the first location visited is indexed by i_0, the last by i_N, and analogously for the switching instants or intervals.

■

The cost associated to any evolution of the system consists of two parts: the cost associated to the *event driven* evolution, i.e., to the number of switches that will occur in the future evolution, and the cost of the *time driven* evolution. We will consider the two parts separately.

Denote the partial sequence of switching time-intervals by

$$\{\varrho_k, \ldots, \varrho_0\}$$

that represents the time driven evolution of the continuous state. More precisely ϱ_h is the time spent in location j_h ($j_k = i$).

Definition 4.4 (Time cost) *The* remaining time cost, *starting from $(\boldsymbol{x}, i = j_k)$ and executing k more switches, is:*

$$\tilde{T}_0(\boldsymbol{x}, j_0, \varrho_0) = \boldsymbol{x}'\bar{\boldsymbol{Q}}_{j_0}(\varrho_0)\boldsymbol{x} = \boldsymbol{x}'\bar{\boldsymbol{Q}}_{j_0}(+\infty)\boldsymbol{x}$$

$$\begin{aligned}
&\tilde{T}_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) \\
&\quad = \boldsymbol{x}'\bar{\boldsymbol{Q}}_{j_k}(\varrho_k)\boldsymbol{x} && (a) \\
&\quad\ +\boldsymbol{x}'\bar{\boldsymbol{A}}'_{j_k}(\varrho_k)\boldsymbol{M}'_{j_k,j}\bar{\boldsymbol{Q}}_j(\delta_{\min}(j))\boldsymbol{M}_{j_k,j}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x} && (b) \\
&\quad\ +\tilde{T}_{k-1}(\boldsymbol{z}, j_{k-1}, \ldots, j_0, \varrho_{k-1}, \ldots, \varrho_0) && (c)
\end{aligned} \tag{4.8}$$

with

$$\boldsymbol{z} = \bar{\boldsymbol{A}}_{j_{k-1}}(\delta_{\min}(j_{k-1}))\boldsymbol{M}_{j_k,j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x}. \tag{4.9}$$

■

Remark 4.4 *Note that the function $\tilde{T}_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0)$ is 2-homogeneous over its variable $\boldsymbol{x}$, in fact it is a quadratic.* ■

Denote the partial sequence of indexes by

$$\{j_k, \ldots, j_0\} \ \in \ 2^{\mathcal{S}}$$

that represents the evolution of the discrete state. In a similar manner as the time cost, we define the cost of the event driven evolution.

Definition 4.5 (Event cost) *The* remaining event cost E_k*, starting from* $(\boldsymbol{x}, i = j_k)$ *and executing k more switches, is (by induction over k):*

$$\begin{aligned}
&E_0(j_0) = 0 \\
&E_k(j_k, \ldots, j_0) = H_{j_k,j_{k-1}} + E_{k-1}(j_{k-1}, \ldots, j_0).\ (d)
\end{aligned} \tag{4.10}$$

■

The total cost of an evolution, that includes both the time-driven and the event-driven cost is thus the function

Definition 4.6 (Total residual cost) *The total residual cost of the evolution scheduled in equation* (4.6) *is defined as*

$$T_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) = \tilde{T}_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) + E_k(j_k, \ldots, j_0).$$

■

The previous definitions require some physical interpretation.

Let us take into account the first item (equation (4.8), $k = 0$) in Definition 4.4. This term is the LQR cost due to the time driven evolution without any switch. It is trivially the area (geometrical interpretation of the integral) below the function $f(t) = \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)$, where $\boldsymbol{x}(t)$ is the solution of the first order LTI differential equation $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$.

By definition this function is nonnegative, because $\boldsymbol{Q} \geq \boldsymbol{0}$. Moreover, since $\boldsymbol{x}(t) \to \boldsymbol{0}$ as $t \to +\infty$ with exponential rate, clearly the value of (4.8), $k = 0$, is finite.

Remark 4.5 *Observe that this may not be the case $\forall\ i \ \in \ \mathcal{S}$, but Assumption 4.1 guarantees that this is true for at least one location in the system. Clearly, if we denote by $\bar{\mathcal{S}} \subseteq \mathcal{S}$ the nonempty subset of $\mathcal{S}$ that verifies the assumption, it holds that the last dynamics j_0 of the switched evolution is such that $j_0 \ \in \ \bar{\mathcal{S}}$.* ■

Let us now comment the second part of equation (4.8), i.e., $k > 0$. It is said that $\tilde{T}(\cdot)$ is a function of:

1. the current discrete state $i = j_k$;
2. the current continuous state $\boldsymbol{x}$;
3. the remaining discrete state evolution $\{j_{k-1}, \dots, j_0\}$;
4. the time $\varrho_k \geq 0$ that will be spent evolving in the current location, starting from point $\boldsymbol{x}$.
5. the remaining time evolution $\{\varrho_{k-1}, \dots, \varrho_0\}$ from the state point $\boldsymbol{z}$ that is reached after ϱ_k and after the minimum permanence time spent in location j_{k-1}.

The subscript k indicates that k switches are still available, or equivalently that $N-k$ switches have already been performed.

Physically this function represents the cost of an evolution that remains for a time ϱ_k in location j_k from point $\boldsymbol{x}$, and switches to location j_{k-1} when the time ϱ_k has elapsed, or equivalently, when the continuous state value has reached the point

$$\boldsymbol{z} = \bar{\boldsymbol{A}}_{j_{k-1}}(\delta_{\min}(j_{k-1}))\boldsymbol{M}_{j_k,j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x}.$$

In particular, the meaning of the terms (a)-(c) can be commented as follows.

(a) This term is the cost of the evolution from point $\boldsymbol{x}$ in location j_k for the finite time ϱ_k. As explained in the Appendix B, it holds

$$\boldsymbol{x}'\bar{\boldsymbol{Q}}_{j_k}(\varrho_k)\boldsymbol{x} = \int_0^{\varrho_k} \boldsymbol{x}'(t)\boldsymbol{Q}_{j_k}\boldsymbol{x}(t)dt,$$

where $\boldsymbol{x}(t) = e^{\boldsymbol{A}_{j_k}t}\boldsymbol{x} = \bar{\boldsymbol{A}}_{j_k}(t)\boldsymbol{x}$.

(b) This term,

$$\boldsymbol{x}'\bar{\boldsymbol{A}}'_{j_k}(\varrho_k)\boldsymbol{M}'_{j_k,j_{k-1}}\bar{\boldsymbol{Q}}_{j_{k-1}}(\delta_{\min}(j_{k-1}))\boldsymbol{M}_{j_k,j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x},$$

is the cost spent in location j_{k-1} during the minimum permanence time $\delta_{\min}(j_{k-1})$. Note that the term (b) is structurally equal to the term (a). In this case however the initial point is

$$\boldsymbol{M}_{j_k,j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x},$$

i.e., the state space reached after a time ϱ_k in location j_k and after the resetting $\boldsymbol{M}_{j_k,j_{k-1}}$ after the switch from j_k to j_{k-1}.

(c) This term,

$$\tilde{T}_{k-1}(\boldsymbol{z}, j_{k-1}, \dots, j_0, \varrho_{k-1}, \dots, \varrho_0),$$

expresses the residual optimal cost when one-switch-less is available, the current discrete state is j_{k-1} and the current continuous state is the point

$$\boldsymbol{z} = \bar{\boldsymbol{A}}_{j_{k-1}}(\delta_{\min}(j_{k-1}))\boldsymbol{M}_{j_k,j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x}$$

reached after the evolution in location j_k and the switch. Note that this term has been already calculated in a recursive mode at the previous step.

(d) This term, in Definition 4.5,

$$H_{j_k,j_{k-1}},$$

is the cost associated to the switch from location j_k to location j_{k-1}.

In the two dimensional case it is possible to provide a graphical representation of terms (a), (b) and (c). This has been done in Figure 4.3.

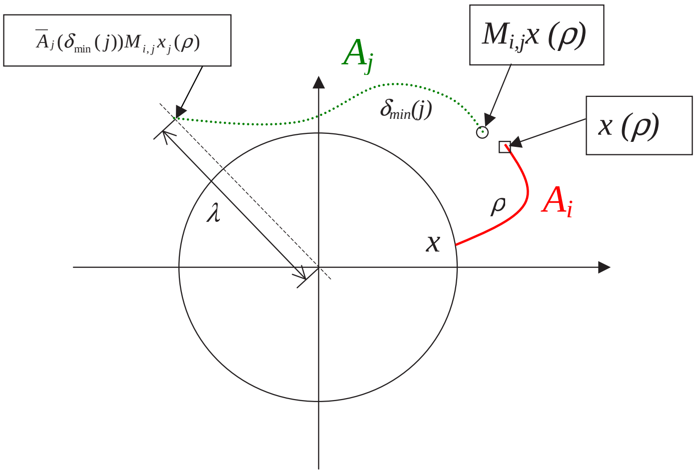


Fig. 4.3. *Sketch of the evolution at the switching instant in* $\mathbb{R}^2$*. The figure illustrates the notation used in equation (4.8). In particular the initial point* $\boldsymbol{x}$ *in the current location* i*, the jump occurred after the switch to mode* j *and the* blind *evolution in location* j *for a time* $\delta_{\min}(j)$ *are represented.*

Minimization of the residual cost

The function

$$T_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) \tag{4.11}$$

depends on $2k$ variables:

$$\{j_{k-1}, \ldots, j_0\} \ \in \ 2^{\mathcal{S}}$$

and

$$\{\varrho_k, \ldots, \varrho_1\}$$

Note that k of them are integer and they take values in a limited set.

We need to compute the optimal residual cost by a suitable choice of these parameters for each couple $(\boldsymbol{x}, i = j_k)$ and for each value of k from 0 to N.

In other words we would like to find the global minimum of equation (4.6) over its $2k$ variables. One possibility is to solve "brute force" the operational research problem

$$T_k^*(\boldsymbol{x}, j_k) = \min_{\substack{\{j_{k-1}, \ldots, j_0\} \\ \{\varrho_k, \ldots, \varrho_1\}}} T_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) \tag{4.12}$$

constrained to

$$\begin{array}{l} \varrho_0 = +\infty \\ \varrho_h \geq \delta_{\min}(j_h) \\ \varrho_k \geq 0 \\ j_k = i \\ j_h \ \in \ succ(j_{h+1}) \cup \{j_h\} \\ h = 0, \ldots, k-1. \end{array} \tag{4.13}$$

There are several reasons to assume that this task is numerically complex. In fact Problem (4.12) is difficult to solve because:

- it is equivalent to a mixed integer quadratic problem. There exist numerical tools (like *CPLEX* or [10]) to solve this type of problems. However when the number of variables, i.e., the number of switches and the number of dynamics, increases these tools exhibit numerical difficulties;
- it is strongly non convex, i.e., local minima are present [79], thus simple descent method, such as *Simplex* or *Newton* or other tools are inefficient.

In these cases the heuristic approaches can be considered, provided that we accept sub-optimal solutions. In fact these approaches (i.e., *Genetic algorithm*, *Simulated annealing*, and so on) are not guaranteed to find or converge to the global minimum.

It is easy to show that, using simple dynamic programming arguments, the optimal cost, converted into a minimum search of the form (4.12), can be computed by solving k times a two-parameter optimizations.

For each value of k and for each couple $(\boldsymbol{x}, i = j_k)$ these two decision variables are:

$$\varrho_k \geq 0, \quad j_{k-1} \ \in \ succ(j_k) \cup \{j_k\},$$

that represent respectively the permanence time in the current location j_k and the index of the next location.

Before proving the main theorem of this section let us observe that function (4.11) can be expressed in terms of the current discrete and continuous state, and of the control variables ϱ_k and j_{k-1}. In fact

$$\begin{gathered} T_k(\boldsymbol{x}, j_k, \ldots, j_0, \varrho_k, \ldots, \varrho_0) = \\ F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), j_{k-1}, \ldots, j_0, \varrho_{k-1}, \ldots, \varrho_0), \end{gathered} \tag{4.14}$$

where

$$\begin{aligned} &F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) = \\ &\quad = \boldsymbol{x}'\bar{\boldsymbol{Q}}_{j_k}(\varrho_k)\boldsymbol{x} \\ &\quad +\boldsymbol{x}'\bar{\boldsymbol{A}}'_{j_k}(\varrho_k)\boldsymbol{M}'_{j_k,j_{k-1}}\bar{\boldsymbol{Q}}_{j_{k-1}}(\delta_{\min}(j_{k-1}))\boldsymbol{M}_{j_k,j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x} \\ &\quad +H_{j_k,j_{k-1}}, \end{aligned} \tag{4.15}$$

i.e., terms (a), (b) of equation (4.8) and term (d) of equation (4.10), and

$$\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k)$$

is the *reached point* as in equation (4.9).

Theorem 4.1 (Optimal remaining cost) *Let us assume that $j_k = i$, i.e., when k switches are missing the current system dynamics is that corresponding to matrix $\boldsymbol{A}_{j_k}$. Let the current state vector be $\boldsymbol{x}$.*

1. *If $k = 0$ then the remaining optimal cost starting from $\boldsymbol{x}$ is:*

$$T_0^*(\boldsymbol{x}, j_0, \varrho_0) = \tilde{T}_0(\boldsymbol{x}, j_0, \varrho_0). \tag{4.16}$$

2. *If $k \in \{1, \ldots, N\}$ then:*
   (i) *the remaining optimal cost starting from $\boldsymbol{x}$ is:*

$$\begin{gathered} T_k^*(\boldsymbol{x}, j_k) = \\ \min_{\substack{j_{k-1} \,\in\, succ(j_k) \,\cup\, \{j_k\} \\ \varrho_k \,\geq\, 0}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}^*(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot); \end{gathered} \tag{4.17}$$

(ii) the next dynamics reached by the optimal evolution is

$$j_{k-1}^{*}(\boldsymbol{x}, j_k) =$$

$$\arg \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\} \\ \varrho_k \geq 0}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}^{*}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot); \tag{4.18}$$

where $j_{k-1}^{*}(\boldsymbol{x}, j_k) = i$ means that no other switch will occur;

(iii) the optimal evolution switches to $\boldsymbol{A}_{j_{k-1}}$ at time $\tau_{N-k+1} = t + \varrho_k^{*}(\boldsymbol{x}, j_k)$, where

$$\varrho_k^{*}(\boldsymbol{x}, j_k) =$$

$$\arg \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\} \\ \varrho_k \geq 0}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}^{*}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot); \tag{4.19}$$

□

Proof. If $k = 0$ the systems is forced to evolve with dynamics $\boldsymbol{A}_{j_k}$ to infinity and the remaining cost (that is also optimal) is the one given in equation (4.16).

If $k > 0$, we have two options. If no future switch occurs then the remaining cost will be $T_k^{*}(\boldsymbol{x}, j_k)$. If at least a future switch will occur, the two decision variables are

- the time before the first switch occurs (parameter $\varrho_k \geq 0$);
- the dynamics reached after the switch (parameter $j_{k-1} \in succ(j_k)$).

In fact, from equations (4.12) and (4.14), it is

$$T_k^{*}(\boldsymbol{x}, j_k) = \min_{\substack{\{j_{k-1}, \ldots, j_0\} \\ \{\varrho_k, \ldots, \varrho_1\}}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot)$$

in force of the principle of optimality [14, 69], we might limit the choice to the current control action (j_{k-1}, ϱ_k) provided that after the switch, it is followed an optimal evolution (T_{k-1}^{*}) from the reached point $\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}^{*}, \varrho_k^{*})$. Hence:

$$T_k^{*}(\boldsymbol{x}, j_k) = \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\} \\ \varrho_k \geq 0}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}^{*}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot),$$

which completes the proof.

■

According to the previous theorem, the optimal remaining cost can be computed recursively, first computing for all vectors $\boldsymbol{x} \in \mathbb{R}^n$ and all dynamics $i \in \mathcal{S}$ the costs $T_0^{*}(\boldsymbol{x}, i)$, then the costs $T_k^{*}(\boldsymbol{x}, i)$, etc.

The procedure may be simplified when all switching costs are zero, as shown in the following proposition.

Proposition 4.1 *Assume that all switching costs are zero, i.e., $H_{i,j} = 0$ for all $e_{i,j} \in \mathcal{E}$. If $\boldsymbol{x}$ is a vector such that $\boldsymbol{x} = \lambda \boldsymbol{y}$, with $\|\boldsymbol{y}\| = 1$ and $\lambda \in \mathbb{R} \backslash \{0\}$, with the notation of Definition 4.2 we have that for all $k \in \{0, \ldots, N\}$ and all $j_k \in \mathcal{S}$*

$$(a) \qquad T_k^*(\boldsymbol{x}, j_k) = \lambda^2 T_k^*(\boldsymbol{y}, j_k), \tag{4.20}$$

$$(b) \qquad \varrho_k^*(\boldsymbol{x}, j_k) = \varrho_k^*(\boldsymbol{y}, j_k), \tag{4.21}$$

$$(c) \qquad j_{k-1}^*(\boldsymbol{x}, j_k) = j_{k-1}^*(\boldsymbol{y}, j_k), \tag{4.22}$$

$$\tag{4.23}$$

□

Proof. (a) To prove this result let us observe that it holds

$$T_k(\boldsymbol{x}, \cdot) = \tilde{T}_k(\boldsymbol{x}, \cdot),$$

in fact $E_k = 0 \; \forall \; k$. Since function $\tilde{T}_k(\cdot)$ is homogeneous of degree 2 (in fact it can be easily shown that it is quadratic) for all values of k, it immediately follows

$$T_k(\boldsymbol{x}, \cdot) = \tilde{T}_k(\boldsymbol{x}, \cdot) = \lambda^2 \tilde{T}_k(\boldsymbol{y}, \cdot).$$

(b)-(c) Under the hypothesis the functions

$$F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + \tilde{T}_{k-1}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot),$$

are homogeneous of degree 2. Thus the minimization problem

$$\tilde{T}_k^*(\boldsymbol{x}, j_k) = \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\} \\ \varrho_k \geq 0}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + \tilde{T}_{k-1}(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot)$$

coincides with

$$\tilde{T}_k^*(\boldsymbol{y}, j_k) = \min_{\substack{j_{k-1} \in succ(j_k) \cup \{j_k\} \\ \varrho_k \geq 0}} F(\boldsymbol{y}, j_k, j_{k-1}, \varrho_k) + \tilde{T}_{k-1}(\boldsymbol{z}(\boldsymbol{y}, j_k, j_{k-1}, \varrho_k), \cdot),$$

by a factor of λ^2.

■

This proposition implies that when all switching costs are zero to determine the optimal costs it is sufficient to evaluate the functions $T_k^*(\boldsymbol{x}, j_k)$ only for vectors $\boldsymbol{x}$ on Σ_n.

Before giving the formal definition of the switching table let us discuss the significance of the optimal argument $\varrho_k^*(\boldsymbol{x}, i) \geq 0$. Its value represents the time that should be spent in the current location i, starting from point $\boldsymbol{x}$, before performing a switch. Therefore if its value is 0, the point $\boldsymbol{x}$ os of immediate switch, else, if it is grater than 0 the point $\boldsymbol{x}$ os not of immediate switch and the evolution continues in the current location i.

Definition 4.7 (Switching table) *The* switching table $\mathcal{C}_k^i$ *is a partition of the state space* $\mathbb{R}^n$ *into* $|succ(i)| + 1$ *regions* $\mathcal{R}_j$ *(for* $j \in succ(i) \cup \{i\}$*) defined as follows*

- *Region*
$$\mathcal{R}_j \equiv \{\boldsymbol{x} \in \mathbb{R}^n \mid \varrho_k^*(\boldsymbol{x}, i) = 0, j_k^*(\boldsymbol{x}, i) = j \neq i\}$$
is the set of points where it is optimal to switch immediately *from location* i *to location* j.
- *The complementary region*
$$\mathcal{R}_i \equiv \mathbb{R}^n \setminus \bigcup_{j \neq i} \mathcal{R}_j,$$
or, equivalently,
$$\mathcal{R}_i \equiv \{\boldsymbol{x} \in \mathbb{R}^n \mid \varrho_k^*(\boldsymbol{x}, i) > 0\},$$
is the region where it is optimal to remain for a time $\varrho_k^*(\boldsymbol{x}, i) > 0$.

■

4.3.2 Lexicographic ordering

In the previous paragraph we highlighted the fact that the switching table procedure requires the solution of a minimization problem.

Let us recall here the problem, taken directly from Theorem 4.1. The system is evolving in the location j_k and the current state space is $\boldsymbol{x}$; furthermore the number of missing switches, out of N, is k.

Theorem 4.1 proves that the optimal choice is to minimize the function

$$T_k^*(\boldsymbol{x}, j_k) = \min_{\substack{j_{k-1} \,\in\, succ(j_k) \cup \{j_k\} \\ \varrho_k \geq 0}} F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}^*(\boldsymbol{z}(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \cdot) \tag{4.24}$$

only in the two variables ϱ_k and j_{k-1}, provided that from the switch on we use an optimal strategy.

The solution of this problem provides the couple

$$\left(j_{k-1}^*, \varrho_k^*\right). \tag{4.25}$$

We omitted here, for sake of clarity, the dependance on $(\boldsymbol{x}, j_k)$ of the arguments j_{k-1}^* and ϱ_k^*. It is important to specify that this couple must be univocally determined. In fact it is possible that there exists several couples of the form (4.25) that minimize the function above to the same value.

Thus we introduced the following lexicographic ordering:

Definition 4.8 (Lexicographic ordering of optimal solution) *Suppose that the problem* (4.24)*, admit α equivalent solutions,*

$$\left(j_{k-1}^*, \varrho_k^*\right)_1, \left(j_{k-1}^*, \varrho_k^*\right)_2, \ldots, \left(j_{k-1}^*, \varrho_k^*\right)_\alpha .$$

These solutions are equivalent in the sense that they all minimize the function (4.24) *to the same value.*

We say

$$\left(j_{k-1}^*, \varrho_k^*\right)_i \prec \left(j_{k-1}^*, \varrho_k^*\right)_h$$

for all $h = 1, \ldots, i-1, i+1, \ldots, \alpha$, iff

$$(j_{k-1}^*)_i < (j_{k-1}^*)_h.$$

Following the definition above the optimal solution has one and only one argument, i.e., the couple $(j_{k-1}^*, \varrho_k^*)_i$ that has smallest index j_{k-1}.

This is particularly important when considering problems with infinite number of switches. In fact in these cases the asymptotical behavior of the cost function over N may generate ambiguity in its minimization search. This criterium ensures that an optimal table is also unique.

4.3.3 Computation of the Table for the initial mode

To decide the optimal initial mode i_0 we may use the knowledge of the cost $T_N^*(\boldsymbol{x}, i)$ (i.e., of the optimal cost to infinity starting from state $\boldsymbol{x}$ with dynamics $j_N = i$) that is evaluated during the construction of the table $\mathcal{C}_N^i$.

Definition 4.1. *Table $\mathcal{C}_{N+1}$ is a partition of the state space $\mathbb{R}^n$ into s regions $\mathcal{R}_i$ ($i \in \mathcal{S}$) where each region is defined as: $\mathcal{R}_i = \{\boldsymbol{x} \mid (\forall j \in \mathcal{S}) T_N^*(\boldsymbol{x}, i) \leq T_N^*(\boldsymbol{x}, j)\}$.*
■

According to this definition, if the initial state belongs to region $\mathcal{R}_i$ we must choose $i_0 = i$ to minimize the total cost.

Note that in some applications the initial discrete state i_0 may not be assigned. However, when this extra degree of freedom on the choice of the initial location is available, this should be done by checking the table $\mathcal{C}_{N+1}$, and choosing i_0 according to the color that corresponds to the initial continuous state $\boldsymbol{x}_0$.

4.3.4 Structure of the Switching Regions

We now discuss the form that the switching regions may take in the case of zero switching costs.

Proposition 4.2 *Consider the case in which $H_{i,j} = 0$ for all $i, j \in \mathcal{S}$. Then any region $\mathcal{R}_j$ of table $\mathcal{C}_k^i$ and of table $\mathcal{C}_{N+1}$ is such that $\boldsymbol{y} \in \mathcal{R}_j \implies (\forall \lambda \in \mathbb{R})\ \lambda \boldsymbol{y} \in \mathcal{R}_j$, i.e., the region $\mathcal{R}_j$ is homogeneous.* □

Proof. When all switching costs are zero, we have shown that equations (4.20) and (4.21) hold. Thus, it follows that in this case $j_k(\boldsymbol{x}, i) = j_k(\boldsymbol{y}, i)$ and $\varrho_k(\boldsymbol{x}, i) = \varrho_k(\boldsymbol{y}, i)$. By Definition 4.7 this implies that all regions of table $\mathcal{C}_k^i$ are homogenous for $k = 1, \ldots, N$.

The table used to select the initial mode has the same property. In fact, assume equation (4.22) holds: taking (as a particular case) $k = 0$ one can see that by Definition 4.1 the regions of table $\mathcal{C}_{N+1}$ are homogenous as well. ■

4.4 Implementation of STP and numerical issues

4.4.1 Algorithm of the STP

We will provide now the algorithm to construct the switching tables. To simplify the notation we decided to show it for the particular case when all switching costs are null. In force of Proposition 4.1 all $\boldsymbol{y}$ can be taken on the Σ_n, because all functions are homogeneous.

We also assume that all jump matrices $\boldsymbol{M}_{i,j}$ are the identity.

Assume that N is the number of available switches and $s = |\mathcal{S}|$.

The algorithm is divided into several steps.

Algorithm 4.1 (Switching table procedure) *The input of this algorithm is the switched system S, its annexed optimal control problem OP and the number of available switches N.*

The output is a set of $N \times s$ tables that the controller can use to provide the feedback control law during the real time evolution.

The list of instructions is depicted in Figure 4.4. ■

Remark 4.6 *The value ∞ in the sixth line of part 2 of the Algorithm represents a sufficiently high value of time. In the practical implementation this is determined by 4 or 5 time constants of the current dynamics.* ■

Note that in the course of the algorithm the function T_k is calculated. Nevertheless this is useful only for the next step. In fact the tables information is contained in the variable $C_k(\boldsymbol{y}, i)$.

This is important in practical applications, where the data should be stored on a PLC whose capacity is usually limited.

1. Initialization: $k = 0$ remaining switches
   For $i = 1 : s$
   Calculate if possible $\boldsymbol{Z}_i \ : \ \boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i = -\boldsymbol{Q}_i$.
   $\forall \ \boldsymbol{y} \ \in \ \Sigma_n$
   Cost assignment
   $T_0(\boldsymbol{y}, i) = \begin{cases} \boldsymbol{y}'\boldsymbol{Z}_i\boldsymbol{y} & \text{if} \quad \exists\, \boldsymbol{Z}_i > \boldsymbol{0} \\ +\infty & \text{else.} \end{cases}$
   Color assignment
   $C_0(\boldsymbol{y}, i) = i$
   end (i)
2. For $k = 1 : N$
   For $i = 1 : s$
   $\forall \ \boldsymbol{y} \ \in \ \Sigma_n$
   Compute the set $succ(i)$;
   Remaining cost:
   For $t = 0 : \infty$
   For $j \in succ(i) \cup \{i\}$
   $\boldsymbol{y}(j, t) = \bar{\boldsymbol{A}}_j(\delta_j)\bar{\boldsymbol{A}}_i'(t)\boldsymbol{y}$
   $\lambda = \|\boldsymbol{y}(j, t)\|$
   $T(\boldsymbol{y}, i, j, t) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_i(t)\boldsymbol{y} + \boldsymbol{y}'\bar{\boldsymbol{A}}_i'(t)\bar{\boldsymbol{Q}}_j(\delta_j)\bar{\boldsymbol{A}}_i'(t)\boldsymbol{y} + \lambda^2 T_{k-1}(\boldsymbol{y}(j, t)/\lambda, j)$
   end (j)
   end (t)
   Cost assignment
   $$T_k(\boldsymbol{y}, i) = \min_{j,t} T(\boldsymbol{y}, i, j, t)$$
   Color assignment
   $$(j^*, t^*) = \arg\min_{j,t} T(\boldsymbol{y}, i, j, t).$$
   $$C_k(\boldsymbol{y}, i) = \begin{cases} j^* & \text{if} \quad t^* = 0 \\ i & \text{if} \quad t^* > 0. \end{cases}$$
   end (i)
   end (k)

Fig. 4.4. *Algorithm for the implementation of the STP*

As an extra advantage we anticipate that it will be proved in the next chapters, that when N grows significantly, the tables converge to the same one (see Chapter 6), thus the data to be passed to a real time controller becomes smaller.

Note that the algorithm is conceptually simple, but calculations become burdensome as the state space dimension increases, since we need to discretize the unitary semisphere. Nevertheless one of the main advantages is that it provide feedback control laws.

Moreover because of the state space discretization, the solution provided by the algorithm is affected by an error. In fact the point $\boldsymbol{y}(j, t)/\lambda$ in function

$$f = \lambda^2 T_{k-1}(\boldsymbol{y}(j, t)/\lambda, j)$$

does not in general belong to the discretization. This forces the algorithm to approximate the value of f with the stored data in the surroundings of $\boldsymbol{y}(j, t)$. Some ideas are suggested in Appendix C.2.

4.4.2 Computational complexity

We discuss here the computational complexity of the STP described above and implemented by the Algorithm 4.1. This results are merely qualitative, and they are described in an intuitive, informal manner.

It has been said that to implement the procedure a state space discretization is required. If the state space is $\mathbb{R}^n$ and we take r samples along each direction, then the discretization set $\mathcal{D}$ has cardinality r^n. In the case when all switching costs are null, and there are no constraints in the state space, then the homogeneity of the functions allow to limite the discretization to Σ_n of $n-1$ dimension, hence the cardinality r^n drops to r^{n-1}.

Proposition 4.3 (Computational complexity of the STP) *The computational complexity of the Algorithm 4.1 is of order $O(Ns(s-1)r^{n-1}N_t)$, where*

- *N is the number of available switches;*
- *s is the cardinality of the set $\mathcal{S}$;*
- *r is the number of samples along each direction of $\mathbb{R}^n$;*
- *N_t is the number of time samples used in the minimization over time.*

□

Proof. Consider the kernel of Algorithm 4.1 reported in Figure 4.5.

0 For $k = 1 : N$
1   For $i = 1 : s$
2     $\forall\, \boldsymbol{y}$
3   For $t = 0 : \infty$
4   For $j \in succ(i) \cup \{i\}$
5         $\boldsymbol{y}(j,t) = \boldsymbol{A}_j(\delta_j)\bar{\boldsymbol{A}}_i'(t)\boldsymbol{y}$
6         $\lambda = \|\boldsymbol{y}(j,t)\|$
7         $T(\boldsymbol{y},i,j,t) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_i(t)\boldsymbol{y} + \boldsymbol{y}'\bar{\boldsymbol{A}}_i'(t)\bar{\boldsymbol{Q}}_j(\delta_j)\bar{\boldsymbol{A}}_i'(t)\boldsymbol{y} + \lambda^2 T_{k-1}(\boldsymbol{y}(j,t)/\lambda, j)$
8   end (j)
9   end (t)
10   Cost assignment

11     $T_k(\boldsymbol{y},i) = \min_{j,t} T(\boldsymbol{y},i,j,t)$

Fig. 4.5. *The kernel of the algorithm for the computation of the switching table.*

By counting the nested *for* cycles we repeat a *minimization search* over time for $N \times s \times r^{n-1}$ times, i.e., for each missing switch k (step (0)), for each location i (step (1)) and for each $\boldsymbol{y}$ on Σ_n (step (2)) we need to:

(a) Take one possible $j \in succ(i)$ (step (4));
(b) Perform a continuous minimization of a regular function [4] (steps (5)-(11));

Let us call μ the complexity of the minimization effort.

Now the complexity of steps (3)-(11) is $O((s-1)\mu)$. In fact the minimization search must be repeated $\forall\; j \in succ(i) \cup \{i\}$, and it should be clear that $|succ(i) \cup \{i\}| \leq s-1$.

[4] The cost functions for this class of systems are linear combinations of exponential functions.

As a minimum search method over time we implemented the exhaustive search[5] over a vector of time steps. Hence if the number of time samples is N_t it holds $\mu \propto N_t$ [71].

Finally the order of magnitude of this algorithm is

$$O(Ns(s-1)r^{n-1}N_t).$$

■

It is important to observe that the complexity is polynomial in N and s. A brute force method that performs a search over all possible switching sequences has complexity of order s^N.

Typical values (example in Section 4.5):

- Number of switches $N = 5$;
- Number of locations $s = 6$;
- State space dimension $n = 2$;
- Discretization samples on Σ_2 $r = 101$;
- Time step exploration $N_t = 300$;

have a computational complexity of the order 10^6.

Note that if we solve by brute force an optimal control problem of the form (4.1) by investigating all admissible switching sequences (they are $(s-1)^N$ in the worst case) the complexity becomes $O(Nr^{n-1}s^N)$ or $O(Nr^n s^N)$ depending on the presence of switching costs.

4.5 Application: a servomechanism with gear-box

As an example we consider the following servomechanism system. It consists of a DC-motor, a gear-box with selectable gear ratios, and a mechanical load. The system setup is depicted in Figure 4.6.

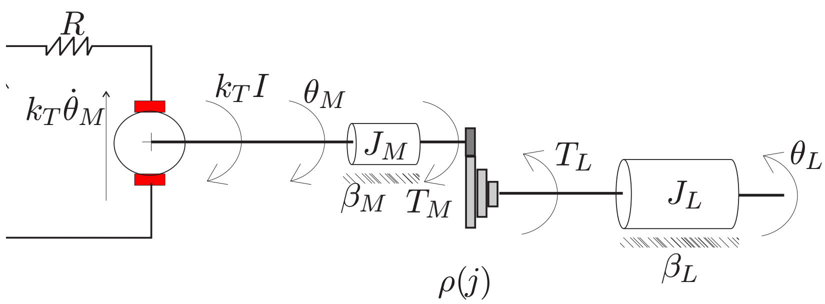


Fig. 4.6. *Servomechanism model with controllable gear ratio.*

The dynamics of the system is described by the relations

$$V = RI + k_T\dot{\theta}_M,$$

$$J_M\ddot{\theta}_M = k_T I - \beta_M\dot{\theta}_M - T_M,$$

$$\dot{\theta}_M = \rho(j)\dot{\theta}_L,$$

[5]It is one variable minimization and the function is only known by points

$$T_L = \rho(j) T_M,$$
$$J_L \ddot{\theta}_L = -\beta_L \dot{\theta}_L + T_L,$$

where

- V is the applied armature voltage,
- I is the armature current,
- R is the armature resistance,
- θ_M, θ_L are the angular position of the motor and load shafts, respectively,
- T_M is the torque developed by the motor,
- k_T is the motor constant,
- J_M and J_L are the equivalent moments of inertia of the motor and load, respectively,
- β_M and β_L are the equivalent viscous frictions coefficients of the motor and load, respectively,
- $\rho(j)$ the gear ratio, $j = 1, 2, 3$.

The above relations can be easily rewritten as the linear differential equation

$$\left[J_L + \rho^2(j) J_M\right] \ddot{\theta}_L + \left[\beta_L + \rho^2(j)\left(\frac{k_T^2}{R} + \beta_M\right)\right] \dot{\theta}_L = \rho(j) \frac{k_T}{R} V.$$

We assume that V can be generated by one of the following PD controllers:

$$V = -k_1(h)\theta_L - k_2(h)\dot{\theta}_L, \ \ h = 1, 2,$$

where $h = 1$ corresponds to a smooth control action, while $h = 2$ corresponds to an aggressive one.

By setting

$$\boldsymbol{x} \triangleq \begin{bmatrix} \theta_L \\ \dot{\theta}_L \end{bmatrix},$$

the overall model can be represented as the autonomous switched linear system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(h, j)\boldsymbol{x},$$

thus

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(h, j)\boldsymbol{x} = \begin{bmatrix} 0 & 1 \\ a_{21}(h, j) & a_{22}(h, j) \end{bmatrix} \boldsymbol{x} \tag{4.26}$$

where

$$a_{21}(h, j) = -\frac{\rho(j)(k_T/R)k_1(h)}{J_L + \rho^2(j) J_M}, \tag{4.27}$$

and

$$a_{22}(h, j) = -\frac{\beta_L + \rho^2(j)\left((k_T^2/R) + \beta_M\right) + \rho(j)(k_T/R)k_2(h)}{J_L + \rho^2(j) J_M}. \tag{4.28}$$

Equivalently, we write

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_i \boldsymbol{x},$$

with

$$i \triangleq 1 + (h - 1) + 2(j - 1),$$
$$\boldsymbol{A}_i \triangleq \boldsymbol{A}(h, j),$$

and $h = 1, 2$, $j = 1, 2, 3$, and consequently $i = 1, \ldots, 6$.

We assume that

(i) the gear shift is sequential, i.e., only transitions $1 \leftrightarrow 3$, $3 \leftrightarrow 5$ are allowed;
(ii) a gear can be shifted only when the smooth control is active, in order to avoid power losses.

The automaton showing all the allowed transitions is depicted in Figure 4.7. The parameters of the system are reported in the table below.

Table 4.1. *Model parameters of the servomechanism system considered in Section 4.5.*

Symbol	Value (IS)	Physical meaning
J_M	1	motor inertia
β_M	0.2	motor friction coefficient
R	50	resistance of armature
k_T	15	motor constant
J_L	50	nominal load inertia
β_L	10	load friction coefficient
ρ	1,2,3	gear ratios
$k_1(1)$	3.2	proportional action (smooth)
$k_1(2)$	31.6	proportional action (aggressive)
$k_2(1)$	3.5	derivative action (smooth)
$k_2(2)$	32.1	derivative action (aggressive)

4.5.1 Numerical simulations

We considered the following numerical values:

- the maximum number of switches is $N = 5$;
- the state $\boldsymbol{x}$ is a continuous function (i.e., $\boldsymbol{M}_{i,j}$ is the identity matrix for any $i, j \in \mathcal{S}$);
- no cost is associated to any switch (i.e. $H_{i,j} = 0$ for any $i, j \in \mathcal{S}$);
- the minimum permanence time in every location is $\delta_{min} = 0.2\ s$;
- the initial state of the system is

$$\boldsymbol{x}_0 = \begin{bmatrix} -1.4 \\ 1.5 \end{bmatrix}$$

- the initial discrete location is 1[6].

Moreover, from equations (4.26), (4.27), (4.28) and Table 4.1, we obtain the following set of dynamics A_i each one associated to location i:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -0.019 & -0.31 \end{bmatrix} A_3 = \begin{bmatrix} 0 & 1 \\ -0.036 & -0.57 \end{bmatrix} A_5 = \begin{bmatrix} 0 & 1 \\ -0.049 & -0.94 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ -0.186 & -0.47 \end{bmatrix} A_4 = \begin{bmatrix} 0 & 1 \\ -0.351 & -0.89 \end{bmatrix} A_6 = \begin{bmatrix} 0 & 1 \\ -0.482 & -1.38 \end{bmatrix} \tag{4.29}$$

It can be easily verified that all dynamics are Hurwitz stable, thus the Assumption 4.1 is verified.

We assumed

[6] Note that if the initial location is not given, we may use the procedure illustrated in Subsection 4.3.3 to evaluate the optimal initial location, given the initial state $\boldsymbol{x}_0$.

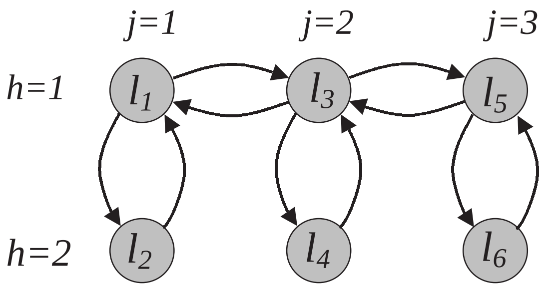


Fig. 4.7. *The hybrid automaton that defines the mode switchings and the set of successors for each location* i*,* $i = 1, \ldots, 6$*.*

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_3 = \boldsymbol{Q}_5 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

and

$$\boldsymbol{Q}_2 = \boldsymbol{Q}_4 = \boldsymbol{Q}_6 = \begin{bmatrix} 3 & 0 \\ 0 & 6 \end{bmatrix}.$$

We evaluate offline the $N \times 6$ switching tables, each of them containing up to $|1 + succ(\cdot)|$ colors.

Provided such tables, the controller is able to estimate the real-time optimal strategy with regard of the described constraints of the system. Knowing the state value $\boldsymbol{x}$, the current location i and the k switches still available, the table $\mathcal{C}_k^i$ will suggest the optimal decision for the system.

From a numerical point of view, the space discretization was of $r = 101$ points along Σ_2.

The time minimization was performed over a time horizon $\tau_{\max}$ equal to three time constant of the slowest mode of matrices A_i, $i = 1, \ldots, 6$, i.e.,

$$\tau_{\max} = 3 \max_{\substack{i\,=\,1,\ldots,6 \\ j\,=\,1,2}} \frac{1}{Re(|\ \lambda_{i,j}\ |)}$$

Table 4.2. *Color mapping of Figure 4.8.*

Location	Color
1	blue
3	red
5	green
6	black

The state trajectory that minimizes the performance index is depicted in Figure 4.8, where the circle indicates the initial state and the squares indicate the values of the state at the switching instants. The color mapping of this trajectory is reported in Table 4.5.1. We found out

$$\mathcal{T}^* = \{0.20, 0.40, 1.47, 4.0, 4.2\},$$

$$\mathcal{I}^* = \{1, 3, 5, 6, 5, 3\},$$

and

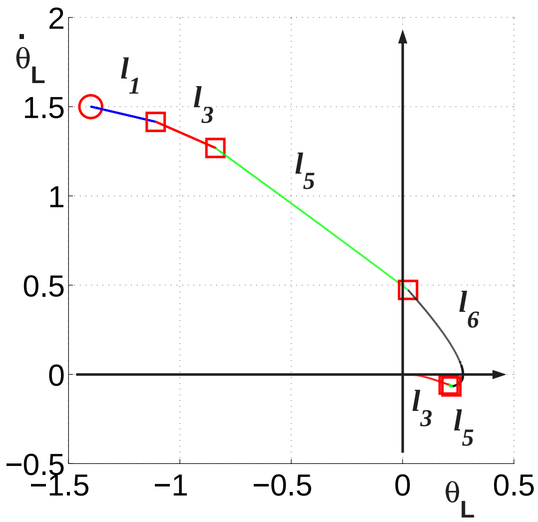


Fig. 4.8. *The system evolution for* $\theta_L(0) = -1.4$, $\dot{\theta}_L(0) = 1.5$, *and initial location* 1.

$$J^* = 4.75.$$

The Figure 4.9 shows, among the 30 tables constructed, only the 5 ones used by the controller during the evolution of the system.

The system initially evolves for the minimum time in location 1. When this time has elapsed, the controller must keep checking the color in table $\mathcal{C}^1_{1,5}$ (see Figure 4.9) corresponding to the current state $\boldsymbol{x}$ (here the state space is the rotational angle of the shaft and its angular velocity). According to this color the controller decides to remain in location 1 or to switch to an adjacent location. In this example an immediate switch to location 3 takes place, since the current state is in the cyan area. Now the controller will wait for the minimum time and then consider table $\mathcal{C}^3_{2,5}$. The same procedure is repeated until all available switches are performed.

It is relevant to notice how the performance of the system is related to the number of available switches. Starting from the same initial conditions (see Figure 4.8), we report the values of the performance index J when $i = 0, \ldots, 6$ switches are available.

Table 4.3. *Values of the performance index upon the number of switches.*

Available switches	Index Value
0	108.62
1	20.78
2	6.69
3	4.84
4	4.84
5	4.75
6	4.69

These results show how the index improves with the number of commutations, but such improvements become negligible after the third switch, when the system has practically reached the origin.

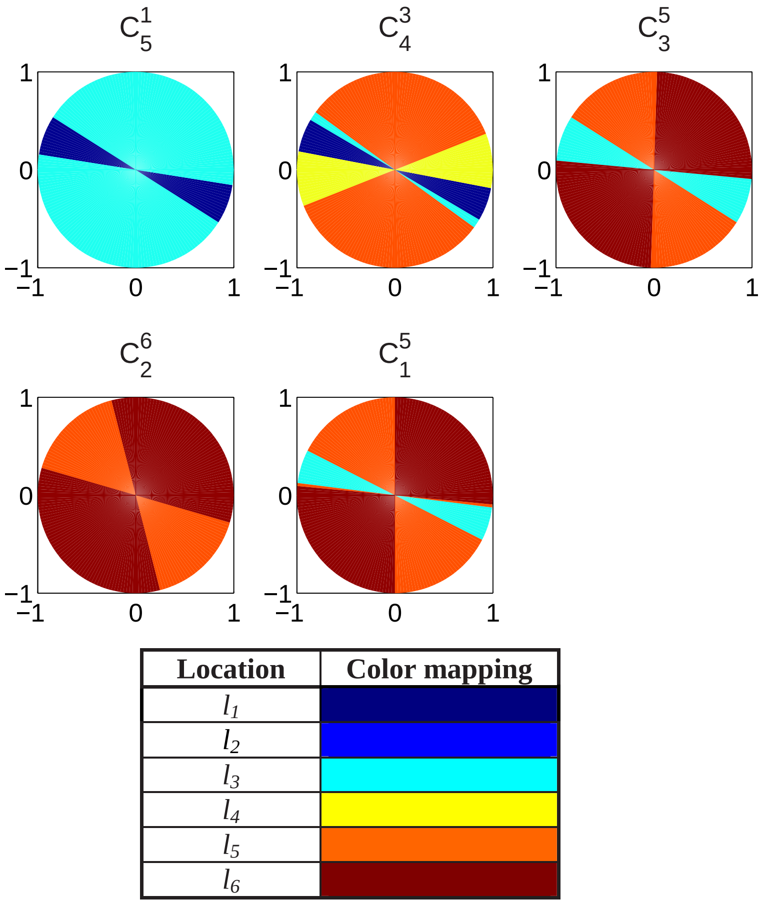


Fig. 4.9. *Tables used by the controller to optimally steer the system to the origin from the initial state* $\boldsymbol{x}_0 = [-1.4, 1.5]'$*, initial location* $i_0 = 1$ *and performing 5 switches in the automaton depicted in Figure 4.7.*

4.6 Particular cases

In this section we will highlight two particular cases of the general optimal control problem applied to switched system described in the previous sections. Let us recall that in this chapter only a finite number of switches N are considered. These two particular cases marked the chronological ordering of the development of the general procedure. We decided however to postpone their description for sake of generality.

The first case appeared initially in [49] and it presents a method for constructing the switching regions for a fixed mode sequence. In this frame there is no degree of freedom in choosing the successor of the current mode, thus the control variables in the optimal control problem are simply the switching instants. The approach is the same (it is based on dynamic programming arguments), but the complexity decreases because at each step of the algorithm only the minimization over the continuous time variable is required.

The second case is described formally in [9]. It presents an extension of the first case, but it allows that at each switching the optimal controller can choose amongst all modes of the system. We might call this method as *arbitrary mode sequence*.

There is a complete degree of freedom in the choice of the successor dynamics. The computational complexity of this approach is certainly higher then the previous one (it is in fact s^2 higher, s is the number of different modes), but definitely lower than the exhaustive search over all possible fixed sequences of length N^7.

We will show in this section, by simple considerations, that both systems can be modelled by an appropriate switched systems and its annexed optimal control problem as described above. Consequently to avoid any redundancy it will not be necessary to repeat the procedure of the table construction for these two particular cases.

In particular the first case represents, in some sense, the simplest way to apply the STP. It is not by chance, indeed, that the majority of properties, propositions and theorems given in the general form, are initially studied on this case and then proved in general.

This is one of the reason why we decided to briefly resume the procedure. Moreover it is helpful for the reader to become more confident with its recursive aspect and its mechanism. For each case some examples and applications are provided. In particular for the first case (fixed mode sequence), an example is provided with non zero switching costs.

4.6.1 Fixed mode sequence

Model and Problem

We consider here the particular switched system SF as in Definition 3.8 whose main characteristic is that the set $succ(i)$ is a singleton or empty for each location i.

As an examples see the oriented graph of one possible SF in Figure 4.10.

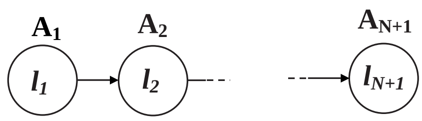


Fig. 4.10. *Oriented graph of a switched system that only admits a fixed sequence of modes. In this case the general optimal control problem is simplified.*

In this case the switching sequence $\mathcal{I} = \{i_1, \ldots, i_{N+1}\}$ is pre-assigned, hence to simplify the notation we denote the state matrices as

$$\boldsymbol{A}_k \triangleq \boldsymbol{A}_{i_k}$$

for $k = 1, \ldots, N + 1$. Moreover the Assumption 4.1 is satisfied.

Remark 4.7 *We may assume that Assumption 4.1 is satisfied, wlg, for dynamics $\boldsymbol{A}_{N+1}$. If this is not the case then there should exist some $m < N + 1$ such that the assumption holds. But if this happens then the problem can be trivially redefined of length m. On the contrary, if there is no such m, then the problem of course is not solvable with finite number of switches.*

[7] If s is the number of different modes and N is the number of switches, an exhaustive search of the optimal sequence of modes by using the fixed mode sequence has exponential growth equal to s^N.

To this system SF associate an optimal control problem $\text{OP}_N(S)$, as defined in Section 3.4.2, reported below.

$$J_N^* \triangleq \min_{\mathcal{T}} \left\{ F(\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k=1}^{N} H_k \right\}$$

$$\begin{array}{ll} \text{s.t.} & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_k\boldsymbol{x}(t) \text{ for } \tau_{k-1} \leq t < \tau_k, \quad k = 1, \ldots, N+1, \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ & 0 = \tau_0 \leq \tau_1 \leq \ldots \leq \tau_{N+1} = +\infty, \\ & \boldsymbol{x}(\tau_k^+) = \boldsymbol{M}_k \ldots \boldsymbol{M}_h \boldsymbol{x}(\tau_h^-) \text{ if } \tau_{h-1} < \tau_h = \ldots = \tau_k < \tau_{k+1}, \end{array} \tag{4.30}$$

We consider the following restrictions:

1. $\forall\, i \in \mathcal{S}\ \delta_{\min}(i) = 0$;
2. $\forall\, e \;\; in\, \mathcal{E}\ \boldsymbol{M}_e = \boldsymbol{I}_n$, i.e., the state space is continuous at the switching instants;
3. $\forall\, e \;\; in\, \mathcal{E}\ H_e = 0$, i.e., all switching costs are null.

Computation of the switching tables

We repeat here the procedure described in general in Section 4.3 for a fixed mode sequence with the additional simplifications given in the end of the last section.

In this paragraph we will present the procedure for this particular case in a simple manner. However the reader can refer to the mentioned Section 4.3 or to Algorithm 4.1, where the pseudo code of the region construction is given for the general case.

Note that the procedure is recursive on k, where k, from now on, represents the number of missing available switches.

The procedure starts in location $N+1$, where 0 switches are available. From every point $\boldsymbol{y}$ on Σ_n, we calculate

$$T_0^*(\boldsymbol{y}) \triangleq \boldsymbol{y}'\boldsymbol{Z}_{N+1}\boldsymbol{y},$$

where $\boldsymbol{Z}_{N+1}$ is the unique solution of the Lyapunov equation $\boldsymbol{A}'_{N+1}\boldsymbol{Z}_{N+1} + \boldsymbol{Z}_{N+1}\boldsymbol{A}_{N+1} = -\boldsymbol{Q}_{N+1}$.

Consider now the location N, where 1 switch is available. We calculate, for each $\boldsymbol{y}$ on Σ_n,

$$T_1(\boldsymbol{y}, \varrho) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_N(\varrho)\boldsymbol{y} + \lambda^2 T_0^* \left(\frac{\bar{\boldsymbol{A}}_N(\varrho)\boldsymbol{y}}{\lambda} \right),$$

where the function T_0^* is calculated in the point reached after a time ϱ, evolving with $\boldsymbol{A}_N$ and starting from point $\boldsymbol{y}$. Note that the factor λ is a scaling factor, due to the fact that the functions T_k's are 2-homogeneous, and therefore calculated only in Σ_n.

Now we minimize. For each point $\boldsymbol{y}$ we look for the value of $\varrho_1^*(\boldsymbol{y}) \geq 0$ such that $T_1(\boldsymbol{y}, \varrho)$ is minimized, i.e.,

$$\varrho_1^*(\boldsymbol{y}) = \arg\min\{T_1(\boldsymbol{y}, \varrho)\}$$

and we call

$$T_1^*(\boldsymbol{y}) \triangleq T_1(\boldsymbol{y}, \varrho_1^*(\boldsymbol{y})).$$

Now we construct the table $\mathcal{C}_1$ by assigning each point $\boldsymbol{y}$ (and all points $\boldsymbol{x} = \lambda\boldsymbol{y}$, $\lambda \in \mathbb{R}$) to

- Region $\mathcal{R}_{st}$ if $\varrho_1^*(\boldsymbol{y}) > 0$;
- Region $\mathcal{R}_{sw}$ if $\varrho_1^*(\boldsymbol{y}) = 0$.

Repeating this assignment for all $\boldsymbol{y}$'s we obtain the switching table $\mathcal{C}_1$.

Suppose now that the same steps are repeated for $k-1$ times and consider now the location $N-k+1$, where k switches are available. We calculate, for each $\boldsymbol{y}$ on Σ_n,

$$T_k(\boldsymbol{y}, \varrho) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_N(\varrho)\boldsymbol{y} + \lambda^2 T_{k-1}^*\left(\frac{\bar{\boldsymbol{A}}_N(\varrho)\boldsymbol{y}}{\lambda}\right),$$

and find, by a single variable time minimization,

$$\varrho_k^*(\boldsymbol{y}) = \arg\min\{T_k(\boldsymbol{y}, \varrho)\}$$

to obtain

$$T_k^*(\boldsymbol{y}) \triangleq T_k(\boldsymbol{y}, \varrho_k^*(\boldsymbol{y})).$$

According to the value of $\varrho_k^*(\boldsymbol{y})$ we assign the point $\boldsymbol{y}$ to $\mathcal{R}_{st}$ or $\mathcal{R}_{sw}$ respectively and build $\mathcal{C}_k$.

Numerical examples

Let us now present the results of some numerical simulations. We consider a second order linear system whose dynamics may only switch between two matrices $\boldsymbol{A}^{(1)}$ and $\boldsymbol{A}^{(2)}$ and the sequence $\mathcal{I}$ is pre-assigned,

$$\mathcal{I} = \{1, 2, 1, 2\}.$$

Thus only three switches are possible ($N = 3$) and the initial system dynamics is $\boldsymbol{A}_1 = \boldsymbol{A}^{(1)}$. Thus, the sequence of switching is

$$\boldsymbol{A}_1 = \boldsymbol{A}^{(1)} \to \boldsymbol{A}_2 = \boldsymbol{A}^{(2)} \to \boldsymbol{A}_3 = \boldsymbol{A}^{(1)} \to \boldsymbol{A}_4 = \boldsymbol{A}^{(2)},$$

where

$$\boldsymbol{A}^{(1)} = \begin{bmatrix} -1 & 1 \\ -18 & -5 \end{bmatrix}, \qquad \boldsymbol{A}^{(2)} = \begin{bmatrix} 1 & -5 \\ 1 & -3 \end{bmatrix}.$$

We also assume that all $\boldsymbol{M}_k$'s are equal to the identity matrix. Finally, we take $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{Q}_3 = \boldsymbol{Q}_4 = \text{diag}\{[1, 2]\}$.

We consider two different cases. We firstly assume that no cost is associated to switches. Secondly, we associate a constant cost to each switch.

First case

The switching regions $\mathcal{C}_k$, $k = 1, 2, 3$, are shown in Figure 4.11 where the following color notation has been used: the lighter (green) region represents the set of states where the system switches to the next dynamics, while the darker (blue) region represents the set of states where the system still evolves with the same dynamics. Note that these regions have only been displayed inside the unit disc because they are homogeneous.

In the bottom right of Figure 4.11 we have shown the system evolution in the case of $\boldsymbol{x}_0 = [0.6, 0.6]'$.

The switching times are

$$\mathcal{T} = \{\tau_1 = 0.01, \tau_2 = 0.35, \tau_3 = 0.40\}$$

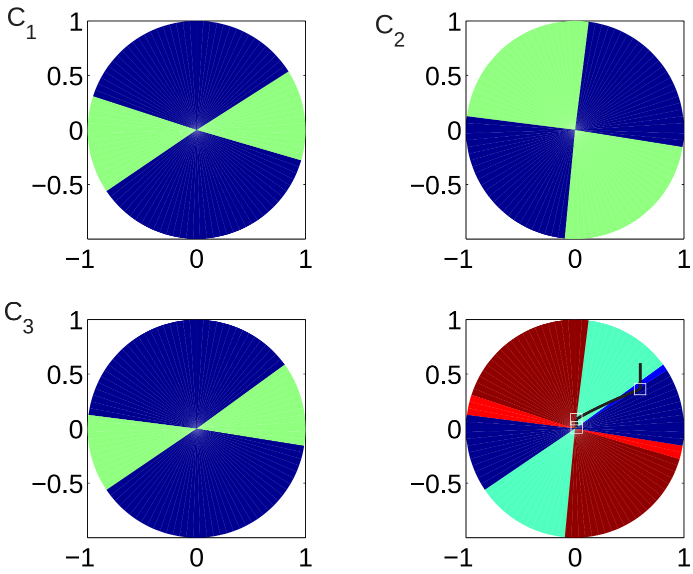


Fig. 4.11. *The switching regions $\mathcal{C}_k$, $k = 1, 2, 3$ in the case of no cost associated to switches, and the system evolution for $\boldsymbol{x}_0 = [0.6, 0.6]'$.*

and the optimal cost is $F(\tau_1, \tau_2, \tau_3) = 0.15$.

Second case

Now, let us assume that non zero costs are associated to switches. In particular, let us assume that $H_1 = H_3 = 0.3$ and $H_2 = 0.1$.

The switching regions $\mathcal{C}_k$, $k = 1, 2, 3$, are shown in Figure 4.12 where we used the same color notation as above, i.e., the lighter (green) region represents the set of states where the system switches to the next dynamics, and the darker (blue) region represents the set of states where the system still evolves with the same dynamics.

In this example $\tilde{\lambda}$ is < 2 and it is sufficient to display the regions within the circle of radius 2.

In the bottom right of Figure 4.12 we have shown the system evolution in the case of $\boldsymbol{x}_0 = [1.3, 1.4]'$. In this case, the switching times are

$$\mathcal{T} = \{\tau_1 = 0.014, \tau_2 = 0.5, \tau_3 = +\infty\}$$

and the optimal cost is $F(\tau_1, \tau_2, \tau_3) = 0.75$.

Modification of the regions

To show how the switching region $\mathcal{C}_k$ may change as H_k varies, we have also computed for this example the region $\mathcal{C}_3$ for different values of $H_3 \in \{0.1, 0.5, 2\}$.

These regions are shown in Figure 4.13, where larger regions correspond to smaller values of H_3.

4.6.2 Arbitrary mode sequence

We consider here the particular switched system SA as in Definition 3.9 whose main characteristic is that the set $succ(i) \equiv \mathcal{S} \setminus \{i\}$ for each location. In this case the automaton has a hyper connected oriented graph, as it can be seen in Figure 4.14.

We annex to the system SA an optimal control problem OP$_N(SA)$, as extensively described in Section 3.4.2, reported below.

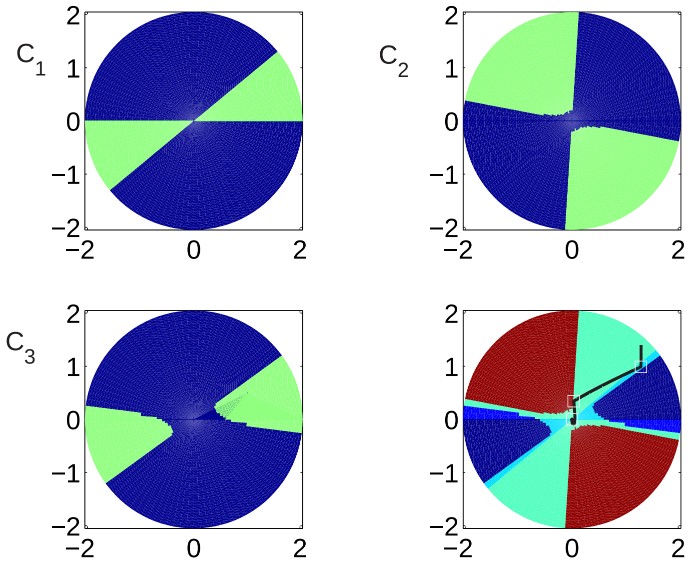


Fig. 4.12. *The switching regions $\mathcal{C}_k$, $k = 1, 2, 3$ in the case of non zero costs associated to switches, and the system evolution for* $\boldsymbol{x}_0 = [1.3, 1.4]'$.

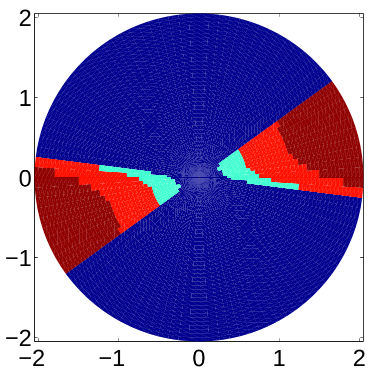


Fig. 4.13. *The switching regions $\mathcal{C}_3$ for different values of the cost* $H_3 \in \{0.1, 0.5, 2\}$.

$$J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt + \sum_{k=1}^{N} H_{i_{k-1},i_k} \right\}$$

$$\begin{array}{lll}
\text{s.t.} & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) & \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 & \\
& i(t) = i_k \quad \text{for } \tau_k \leq t < \tau_{k+1} & k = 0, \ldots, N \\
& i_{k+1} \in \mathcal{S} & k = 0, \ldots, N \\
& \tau_0 = 0, \ \tau_{N+1} = +\infty & \\
& \tau_{k+1} \geq \tau_k + \delta_{\min}(i_k) & k = 0, \ldots, N \\
& \boldsymbol{x}(\tau_k^+) = \boldsymbol{M}_{i_{k-1},i_k}\boldsymbol{x}(\tau_k^-) & k = 1, \ldots, N
\end{array} \qquad (4.31)$$

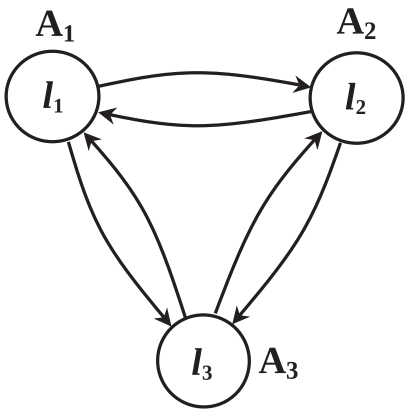


Fig. 4.14. *Oriented graph of a switched system that admits all possible sequence of modes.*

Computation of the switching tables

We will describe here the procedure, in simplified manner, to construct the switching tables. In particular we will highlight the differences with the fixed mode sequence and with the general case. As before, for sake of clarity we will consider the following simplifications:

(i) there is no minimum permanence time in each location;
(ii) all switching costs are null;
(iii) the evolution in the state space is a continuous function of t, i.e., $\forall\ k = 1, \ldots, N+1$ the matrices $M_k = \boldsymbol{I}_n$.

As a first step we calculate and store, for each location i of the automaton and for each point $\boldsymbol{y}$ on Σ_n, the function

$$T_0^*(\boldsymbol{y}, i) \triangleq \begin{cases} \boldsymbol{y}'\boldsymbol{Z}_i\boldsymbol{y}\ if\ \boldsymbol{A}_i\ is\ stable \\ +\infty \qquad\qquad else \end{cases},$$

where $\boldsymbol{Z}_i$ is the unique solution of the Lyapunov equation $\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i = -\boldsymbol{Q}_i$, see also Appendix A.3. By Assumption 4.1, there exists at least one $i \in s$, such that $T_0^*(\boldsymbol{y}, i)$ is finite $\forall\ \boldsymbol{y}$.

Now suppose that 1 switch is available. As before, we evaluate

$$T_1(\boldsymbol{y}, i, j, \varrho) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_i(\varrho)\boldsymbol{y} + \lambda^2 T_0^*\left(\frac{\bar{\boldsymbol{A}}_i(\varrho)\boldsymbol{y}}{\lambda}, j\right),$$

where λ is a normalizing factor of the reached point, that in general will not belong to Σ_n.

Now minimize $T_1(\boldsymbol{y}, i, j, \varrho)$ over the time continuous variable ϱ and the possible successors j of location i, finding

$$T_1^*(\boldsymbol{y}, i) = \min_{\substack{j \in \mathcal{S} \\ \varrho \geq 0}} T_1(\boldsymbol{y}, i, j, \varrho),$$

i.e., whenever the state space is $\boldsymbol{y}$, the current location is i and one switch is still available, then the optimal strategy is to remain in location i for a time $\varrho_1^*(\boldsymbol{y}, i)$ and then switch to location $j_1^*(\boldsymbol{y}, i)$, where

$$[\varrho_1^*(\boldsymbol{y}, i), j_1^*(\boldsymbol{y}, i)] = \arg\min\{T_1(\boldsymbol{y}, i, j, \varrho)\}.$$

The table $\mathcal{C}_1^i$ is constructed as follows:

- if $\varrho_1^*(\boldsymbol{y}, i) = 0$, (i.e., switch immediately) then assign the color of location j_1^*;
- if $\varrho^*(\boldsymbol{y}, i) > 0$, (i.e., stay in location i) then assign the color of location i.

By iteration over the number of switches all the other tables can be constructed.

4.6.3 Numerical Examples

Consider the second order switched linear system with dynamic matrices

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & -10 \\ 100 & 1 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} 1 & -100 \\ 10 & 1 \end{bmatrix}, \quad \boldsymbol{A}_3 = \begin{bmatrix} -0.1 & 0 \\ 0 & -0.1 \end{bmatrix}$$

$(\boldsymbol{f}_1 = \boldsymbol{f}_2 = \boldsymbol{f}_3 = \boldsymbol{0})$ and let $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{Q}_3 = \boldsymbol{I}_2$, $N = 3$, $\boldsymbol{x}_0 = [1, 1]'$. Note that while $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are unstable matrices, $\boldsymbol{A}_3$ is strictly Hurwitz, so that Assumption 4.1 is satisfied.

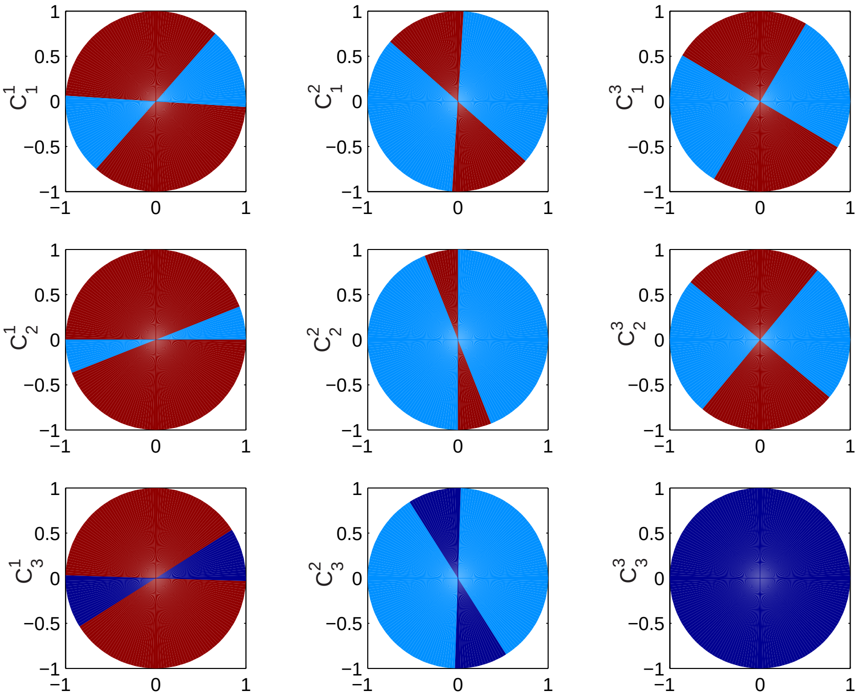


Fig. 4.15. *The set of tables for the numerical example described in Section 4.6.3 where* $N = 3$ *and* $\mathcal{S} = \{1, 2, 3\}$.

We first execute the offline part of the procedure, consisting in the construction of the $N \times s = 9$ tables $\mathcal{C}_k^i$, for $k, i = 1, 2, 3$. Results are reported in Figure 4.15 where the following color notation has been used: Red color (medium gray) is used to denote region $\mathcal{R}_1$, i.e., the set of states where the system either switches to $\boldsymbol{A}_1$ if the current variable of the control variable is $i(t) \neq 1$, or no switch occur if $i(t) = 1$; light blue (light gray) denotes region $\mathcal{R}_2$, and dark blue (dark gray) is used to denote $\mathcal{R}_3$.

As an example, by looking at $\mathcal{C}_1^2$ we know that, if the initial dynamics is $\boldsymbol{A}_2$, then the system may either switch to $\boldsymbol{A}_1$ or still evolve with the same dynamics $\boldsymbol{A}_2$: on the contrary a switch to dynamics $\boldsymbol{A}_3$ may never occur.

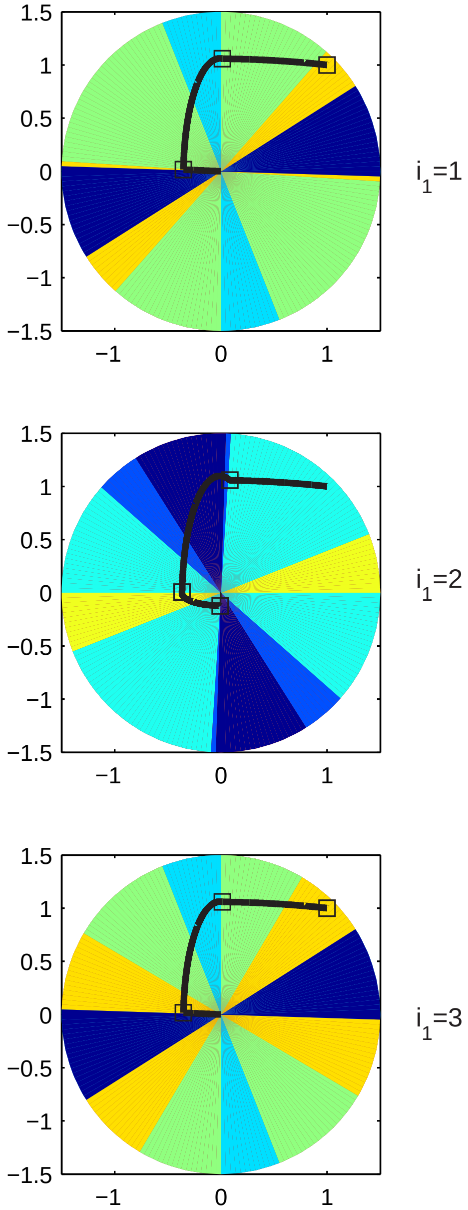


Fig. 4.16. *The system evolution for $\boldsymbol{x}_0 = [1, 1]'$ and i_1 varying in $\mathcal{S}$ for the example described in Section 4.6.3.*

In Figure 4.17 we have reported table $\mathcal{C}_0$ that shows the partition of the state space introduced in Subsection 4.3.3. The same color notation has been used. In particular, this table enables us to conclude that the global optimum may only be reached when the initial system dynamics is either $\boldsymbol{A}_1$ or $\boldsymbol{A}_2$. On the contrary, whenever the initial system dynamics is $\boldsymbol{A}_3$, we may only reach a suboptimal value of the performance index.

Now, let us present the results of some numerical simulation. Let us assume that the initial state is $\boldsymbol{x}_0 = [1,1]'$. We compute the optimal mode sequence for all admissible initial system dynamics, i.e., we assume $i_0 = 1, 2, 3$, respectively.

The results of numerical simulations are reported in Figure 4.16 where switches are highlighted trough a small black square.

Detailed results may be read in Table 4.4 where we have reported the optimal mode sequence, the optimal timing sequence and the corresponding cost value for the different initial dynamics. We may observe that the best solution may only be reached when the initial system dynamic is the second one. In the other cases only a suboptimal value of the cost may be obtained. Note that these results are in accordance with those of Figure 4.17 being $\boldsymbol{x}_0 \in \mathcal{R}_1$.

The correctness of the solution has been validated through an exhaustive inspection of all admissible mode sequences. More precisely, for each admissible mode sequence we have computed the optimizing timing sequence and the corresponding cost value. In such a way we have verified that $J_3^* = 0.126$ is indeed the global optimum.

i_0	i_1	i_2	i_3	τ_1	τ_2	τ_3	V_3
1	2	1	3	0.000	0.009	0.060	0.669
2	1	2	3	0.009	0.062	0.116	0.126
3	2	1	3	0.000	0.009	0.060	0.669

Table 4.4. *Detailed results of the numerical example described in Section 4.6.3 when the initial state is* $\boldsymbol{x}_0 = [1,1]'$.

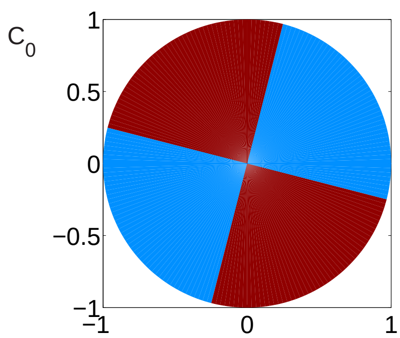


Fig. 4.17. *Table* $\mathcal{C}_4$, *for the computation of the initial mode of the example described in Section 4.6.3.*

4.7 A servomechanism with gear-box

As a final application example, consider the servomechanism system described in Section 4.5. The difference with the model in Section 4.5 are the following:

1. It only has 2 selectable gear ratios;
2. The oriented graph associated to the system is hyper connected

In Figure 4.18 we depicted a sketch of the system.

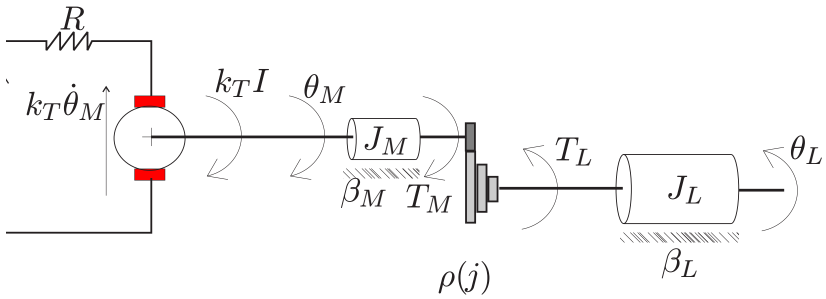


Fig. 4.18. *Servomechanism model with controllable gear ratio.*

For the details and models please refer to Section 4.5 Thus the system may switch between $s = 4$ different LTI modes. In particular

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ -0.019 & -0.309 \end{bmatrix} \quad \boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ -0.186 & -0.477 \end{bmatrix}$$

$$\boldsymbol{A}_3 = \begin{bmatrix} 0 & 1 \\ -0.036 & -0.572 \end{bmatrix} \quad \boldsymbol{A}_4 = \begin{bmatrix} 0 & 1 \\ -0.351 & -0.890 \end{bmatrix},$$

whose eigenvalues are all in the stable half plane. Note that the automaton graph, showing all the allowed transitions is depicted in Fig. 4.19.

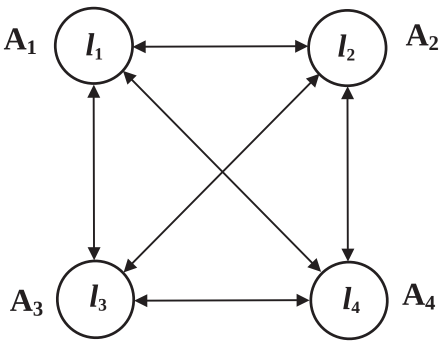


Fig. 4.19. *Graph of the switched system described in Section 4.7.*

4.7.1 Numerical simulations

To complete the simulation setup consider the following numerical values:

- the maximum number of switches is $N = 6$;
- the state $\boldsymbol{x}$ is a continuous function (i.e., $\boldsymbol{M}_{i,j} = \boldsymbol{I}_2 \; \forall \; i, j \in \{1, \ldots, 4\}$);
- no cost is associated to switches (i.e., $H_{i,j} = 0 \; \forall \; i, j \in \{1, \ldots, 4\}$);
- the initial state of the system is $\boldsymbol{x}_0 = [-0.78, 0.63]'$;
- the initial system dynamics is $\boldsymbol{A}_3$.

Finally, we take as weighting matrices

$$\boldsymbol{Q}_1 = \boldsymbol{Q}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \;\; \boldsymbol{Q}_2 = \boldsymbol{Q}_4 = \begin{bmatrix} 3 & 0 \\ 0 & 6 \end{bmatrix},$$

obviously positive definite. We solve this optimization problem using the procedure described above.

4.7.2 Switching table procedure

We evaluate offline the $N \times s = 24$ switching tables, each of them containing up to $s = 4$ colors. A space discretization of $r = 51$ points along Σ_2 and a local minimum search over five time constants were considered sufficiently fine.

The state trajectory that minimizes the performance index is depicted in Figure 4.20, where the circle indicates the initial state and the squares indicate the value of the state at the switching instants. The optimal mode sequence is

$$\mathcal{I}^* = \{3, 4, 1, 3, 4, 1\}$$

and the optimal sequence of switching times is

$$\mathcal{T}^* = \{1.95, 4.75, 39.85, 48.70, 51.45, +\infty\}.$$

The resulting value of the performance index is $J^*(\boldsymbol{x}_0) = 1.263$.

We can observe that the system after three switches has practically reached the origin, thus the complete evolution is no longer visible. The cheapest trajectory is obtained starting with the most aggressive voltage level ($\boldsymbol{A}_3$), then changing gear to $\boldsymbol{A}_4$, and finally going to dynamics $\boldsymbol{A}_1$ in order to drive smoothly the shaft towards the steady state.

One may argue that the number of switches considered for this problem is not appropriately chosen in order to obtain a significant reduction of the performance index. Thus, for the given initial point $\boldsymbol{x}_0$ and initial mode $i = 3$, we reported in table 4.5 the values of $\mathcal{I}^*$, $\mathcal{T}^*$ and J^*, for different growing values of N[8].

Table 4.5 suggests at least three very interesting ideas. First of all it should be remarked that not all the number of available switches is useful to obtain a cost reduction and a different trajectory. This is evident for the two different values of $N = 2$ and $N = 3$, whose corresponding optimal switching sequence and switching times are the same. Secondly it shows that the optimal cost J^*, given the initial conditions, is a non increasing function of N. Finally, there exists a value of N^*, depending on the particular problem[9], that yields to an asymptotic value of the switching cost for the given initial point. These considerations, formally proved in the following chapter, lead to a fundamental theoretical result, that allowed us to deal with $N = \infty$ number of switches and eventually to completely relax Assumption 4.1. For sake of completeness we report in Figure 4.21 a diagram of the first and the last column of Table 4.5.

[8] Number of switches
[9] For this problem $N^* = 3$

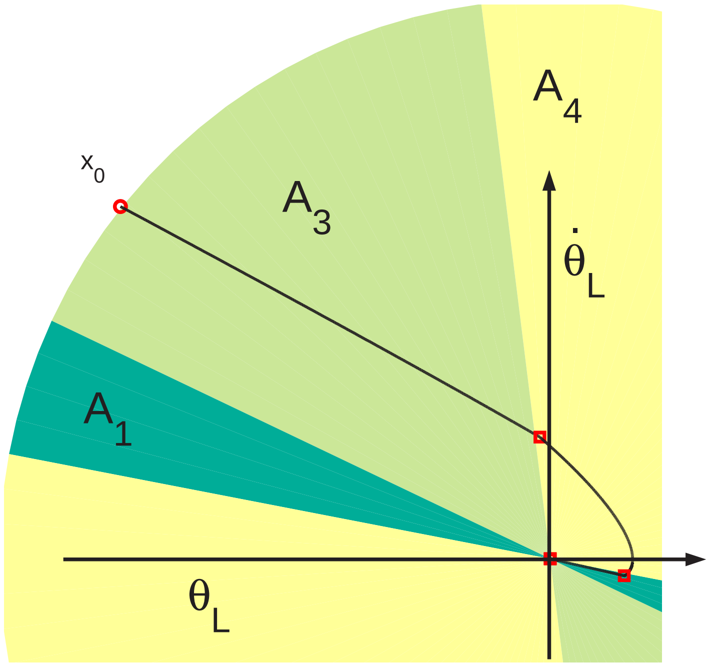


Fig. 4.20. *The system evolution for* $\theta_L(0) = -0.78$, $\dot{\theta}_L(0) = 0.63$, *and initial dynamics* $\boldsymbol{A}_3$.

N	$\mathcal{I}^*$	$\mathcal{T}^*$ (s)	J^*
0	$\{3\}$	$\{+\infty\}$	2.092
1	$\{3,4\}$	$\{1.9,+\infty\}$	1.311
2	$\{3,4,1\}$	$\{1.95,4.75,+\infty\}$	1.263
3	$\{3,4,1\}$	$\{1.95,4.75,+\infty\}$	1.263
4	$\{3,4,1,3,4\}$	$\{1.95,4.75,39.85,48.7,+\infty\}$	1.263
5	$\{3,4,1,3,4,1\}$	$\{1.95,4.75,39.85,48.7,51.45,+\infty\}$	1.263
6	$\{3,4,1,3,4,1\}$	$\{1.95,4.75,39.85,48.7,51.45,+\infty\}$	1.263

Table 4.5. *Optimal solutions of the problem described in Section 4.7 for increasing values of the allowed number of switches* N

4.8 Conclusions

We formally presented in this chapter the kernel of this thesis, i.e., to provide a constructive method for designing a feedback control law for a particular class of switched system that minimizes a given performance index. We have shown that there exist a numerically viable procedure, based on the principle of optimality, that leads to the construction of appropriate switching tables.

The minimization of a given LQR like index, of the model under consideration, takes the form of a state space partition into regions that suggest the optimal switching strategy. In this chapter we restricted the analysis to switched linear affine systems.

It has been shown that the procedure can be applied offline, thus providing the law in feedback form, but as a disadvantage it requires a discretization of the state space, with evident *curse of dimensionality*.

One of the main restrictions on the model and problem considered in this chapter are the finiteness of the of switches and the absence of constraints in the state space.

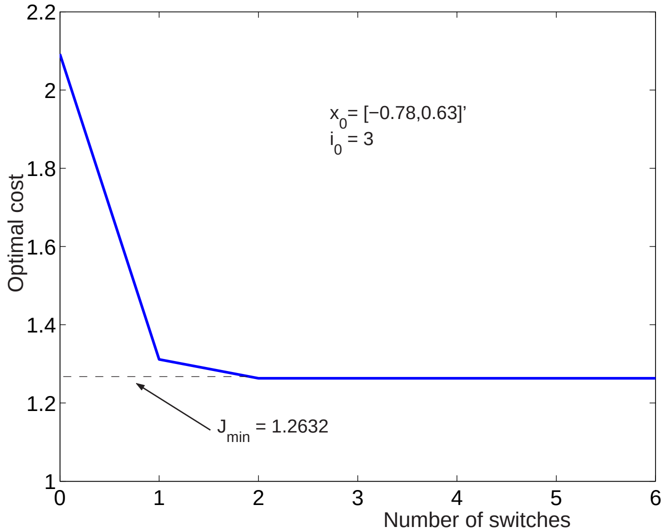


Fig. 4.21. *Convergence of the optimal cost J^* with the number of the N available switches for the given initial conditions.*

The following chapters aim to relax these two restrictions, to provide more general results.

5

Finite number of switches: hybrid automaton

5.1 Introduction

In Chapter 4 we considered switched systems and a particular optimization problem, with an infinite horizon quadratic cost function and a fixed number N of allowed switches.

Here we show how to solve the same optimal control problem for a more general hybrid automata HA. We will show that the STP, described in the previous chapter, can be extended to the problems featured by constraints on the state space.

In a switched system all switches are assumed to be *controllable* (i.e., they can be triggered by the controller). In a hybrid automaton there may also exist *autonomous* switches that are internally forced by the crossing of a given threshold. This type of autonomous switch has also been considered by [122] in a recent work.

This is formalized in Chapter 3 by the introduction in the basic switched system with invariants and guards. We considered, under the described set up of the HA, two different approaches.

In fact, the presence of internal triggers that force the occurrence of switches may be interpreted in two ways:

(a) a subset of edges may fire autonomously, depending upon a set of constraints (*guards*) on the space state $\mathbb{R}^n$, i.e., the discrete controller has no influence on this event;
(b) a switch must occur as a prioritized event, depending upon a set of constraints (*guards*) on the space state $\mathbb{R}^n$, and it is commanded by the discrete controller.

The modelling power of these aspects of an HA can be read in Chapter 3. Here we will limit ourselves to developing the extension of the STP in both cases. We might recall however that, in general, the approach (a) is more suitable for those physical systems with "constructive" constraints. As a trivial example consider a circuit containing a diode where the voltage threshold

$$x_1(t) < 0$$

denotes the condition where the diode behaves as an open circuit.

The approach (b) is more suitable to model cases where the continuous evolution of the system must be restricted to a safe or specification region, i.e., the systems behave under certain safety and liveness constraints. An application to a physical system is discussed in Example 5.5.2.

In particular, based on the notion of l-complete approximations [95, 43], [83] and on the supervisory control theory of Ramadge and Wonham [96] we design a discrete supervisor that guarantees safety and liveness constraints, expressed in terms of an invariant set on the state space, that restricts the switching DOF of the optimal controller. This method was developed by *Gromov et al.* in a joint work [32] and described in Appendix D.

Provided that the behavior of the HA is deterministic, i.e., for each state of the evolution $(\boldsymbol{x}(t), i(t)), \forall\, t \in [0, +\infty)$, it is always possible to model $(\boldsymbol{x}(t+dt), i(t+dt))$ with probability 1, the STP is applicable to both interpretations of the problem.

Note however that the trade-off of this important result is a quite high computational cost (to be performed offline). In fact the whole space discretization is required.

As an advantage, we remark that the investigation of the continuous evolution of the system can be restricted to the invariant set. This leads to some degree of approximation, especially in the case where the invariant set is not limited. In this case some extra information on the physical modelling procedure must be considered.

5.2 The considered model

In this chapter we will deal with the optimal control of the hybrid automaton HA, as in Definition 3.16. Briefly a hybrid automaton the HA considered here is a tuple $HA = (\mathcal{L}, act, inv, \mathcal{E}, \mathcal{M})$, whose entries have the following meaning

- $\mathcal{L}$ is a finite set of locations indexed by $i = 1, \ldots, s$.
- $act : \mathcal{L} \rightarrow$ *Inclusions* is a function that associates to each location i a differential equation of the form $\dot{\boldsymbol{x}} = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{f}_i$.
- $inv : \mathcal{L} \rightarrow$ *Invariants* is a function that associates to each location i an invariant $inv_i \subseteq \mathbb{R}^n$ such that $\boldsymbol{x} \in inv_i$.
- $\mathcal{E} \subset \mathcal{L} \times Guards \times \mathcal{L}$ is the set of edges. The edge $e_{i,j}$ is enabled when the current location is i and the current continuous state is $\boldsymbol{x} \in g_{i,j} \subseteq \mathbb{R}^n$: it may fire reaching the new location j.
- A linear jump relation is $\mathcal{M} \subset \mathbb{R}^n \times \mathbb{R}^n$ associated to an edge $e_{i,j}$. When the edge fires, $\boldsymbol{x}$ is reset to $\tilde{\boldsymbol{x}} = \boldsymbol{M}_{i,j}\boldsymbol{x}$, where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$.

Additionally a *minimum permanence time* $\delta_{\min}(i)$, Definition 3.7, in each location can be considered.

As described in the introduction of this chapter, the presence of *invariant and guards*, Definitions 3.17 and 3.18, associated to edges influences the behavior of the HA, and consequently the problem formulation and its solution should be described consistently. More precisely the presence of these sets have an effect on the switching scheduling, and thus on the designing of the control policy.

It is fundamental for the successful design of the control law by the construction of switching tables, that the system is deterministic, i.e., the *hybrid evolution* $(\boldsymbol{x}(t), i(t))$ is exactly known for any given initial state.

Once this is guaranteed we may analyze two different interpretations of the switching constraints. More precisely we will refer to *autonomous hybrid automaton* AHA, as in Section 3.6.1 (**Case a**) and to *constrained hybrid automaton* CHA, as in Section 3.6.4 (**Case b**).

5.2.1 Case a: autonomous hybrid automaton AHA

This case considers an *autonomous* HA, meaning that this system is subject to sequences of autonomous switches. Detailed description of the AHA and its dynamical behaviors are given in Section 3.6.1.

In this model, not only the time driven evolution $\boldsymbol{x}(t)$ is uncontrolled (we only studied hybrid systems whose continuous control $\boldsymbol{u} = \boldsymbol{0}$), but also the discrete event evolution $i(t)$ is subject to autonomous behaviors according to subsets (named as *guards*) of the state space.

Assume that the current hybrid state is, at a given time t, $(\boldsymbol{x}, i)$. For this state there are two possible conditions:

1. $\boldsymbol{x} \in inv_i$.
2. $\boldsymbol{x} \notin inv_i$.

In case (1) it is possible to define a set of controllable successors, $succ_c(i) \in 2^{\mathcal{S}}$ as in Definition 3.21, each one associated to each *controllable edges* (Definition 3.19) exiting the location i.

While the system is evolving in location i within the corresponding invariant, the DOF of the switcher is defined by $succ_c(i) \cup \{i\}$.

For this particular research we assume that the *guards* associated to the controllable edges coincide with the *invariant* inv_i of location i.

In case (2) the system must leave location i, in agreement with the definition of the invariant. Hence an *autonomous switch* will occur, and the systems falls "spontaneously" into another location, let's say j, which is univocally determined by the guard $g_{i,j}$, according to the Assumption 3.1.

We also recall here the Definition 3.22 of $succ_a(i)$ which denotes the indices associated to the locations reachable from i, by firing an *autonomous edge*.

Here we assume that the number of N available *controllable* switches is finite, but we do not assume the same for the number of *autonomous* switches. Thus, according to the shape of the autonomous guards, the system may

- become unstable with no control;
- exhibit Zenoness.

In Section 3.6.3 we provided sufficient structural conditions on the AHA that avoid these undesirable behaviors.

5.2.2 Case b: constrained hybrid automaton CHA

This case considers a *constrained* HA, CHA, meaning that the switching strategy is influenced by the value of the current continuous state $\boldsymbol{x}$. Detailed description of the CHA and its dynamical behaviors are given in Section 3.6.4.

We suddenly state that in this case there are no autonomous sequences of switches, thus the instability issue and the Zenoness are avoided *a priori*. In fact the number of available switches N (all controllable) is limited.

In this model we consider the guards defined as in Definition 3.28, where the guard $g_{i,j} \equiv inv_j$ is enabled *iff* the state $\boldsymbol{x}$ belongs to the invariant of the location j, destination of a switch.

The invariants are constructed by converting a specification[1], imposed on the quantized output signals of the system, that guarantees safety and liveness of the CHA. The procedure is described in detail in Appendix D.

We recall the Definition 3.29 of the *set of successors*, $succ(\boldsymbol{x}, i) \equiv \{j \in \mathcal{S} : \boldsymbol{x} \in g_{i,j}\}$. Note the dependency of this set from both components of the hybrid state, in opposition to switched systems S in Chapter 4, where the dependency was only on the discrete part i.

[1] Some typical specifications on the dynamical behavior of a HA are for instance the *safety* and the *liveness*.

The dynamical behavior is briefly described as follows. Let us assume that the current hybrid state is, at a given time t, $(\boldsymbol{x}, i)$. For this state there are two possible conditions:

1. $\boldsymbol{x} \in inv_i$.
2. $\boldsymbol{x} \notin inv_i$.

In case (1) the controller can choose to switch to anyone of the locations enabled, through the guards, by the current value $\boldsymbol{x}$, or it can decide to remain in the current location i, since the invariant condition is verified.

In case (2), the system must leave location i, because i is no longer considerable. Hence the DOF of the controller is $succ(\boldsymbol{x}, i)$.

It is meaningful to remark that this model is potentially blocking, as outlined in Remark 3.3. In fact there is the evident possibility that $\boldsymbol{x}$ leaves the invariant and the set becomes $succ(\boldsymbol{x}, i) \equiv \emptyset$.

5.3 Case a: optimal control problem for AHA

The optimal control problem for AHA, OP(AHA) is based on the assumption that the discrete controller has at most N (fixed a priori) controllable switches available. The formal definition of the problem is given in Section 3.6.3, where all the properties, the symbols and the assumption that allow the existence of a solution are extensively described.

Here we shall limit to report the problem formulation, as it appears in Definition 3.26, and we refer the reader to the mentioned Section 3.6.3 for a complete illustration of the formalism.

$$
\begin{aligned}
J_N^* \triangleq\ & \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt \right\} \\
\text{s.t. } & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)} \\
& 0 = \tau_0 \leq \ldots \leq \tau_k \leq \ldots \leq \tau_{N+1} = +\infty \\
& \qquad\qquad \text{(controlled switching times)} \\
& i(0) = i_0 \quad \text{(initial location)} \\
& \boldsymbol{x}(0) = \boldsymbol{x}_0 \quad \text{(initial state)} \\
& i(\tau_k) \in \ succ_c(i(\tau_k^-)) \ \text{(location reached after the } k-th \text{ controlled switch)} \\
& \boldsymbol{x}(\tau_k) = \boldsymbol{M}_{i(\tau_k^-),i(\tau_k)}\boldsymbol{x}(\tau_k^-) \\
& \qquad\qquad \text{(state reached after the } k-th \text{ controlled switch)} \\
& \sigma_k = \sigma(\boldsymbol{x}(\tau_k), i(\tau_k)) \quad \text{(autonomous sequence)} \\
& i(\tau_k+\theta) = \varphi_{\sigma_k}(\theta) \text{ for } \theta \in [0, \tau_{k+1}-\tau_k) \\
& \qquad\qquad \text{(autonomous index trajectory)}
\end{aligned}
\tag{5.1}
$$

Briefly, function $i(t)$ is composed of $N+1$ blocks delimited by the instants τ_k's where the controlled switches occur. Each block is a piecewise constant function: steps internal to the interval $t \in [\tau_k, \tau_{k+1})$ correspond to autonomous switches. More precisely between the occurrence of two controllable switches the location *does not remain constant*, as in switched systems of Chapter 4, but it may be *piecewise* constant, according to the occurrence of autonomous switches.

We named this piecewise constant function of autonomous switches as $\varphi_\sigma(t)$, and an example is depicted in Figure 5.1.

The control variables in this problem are the *sequence of controlled switching times* $\mathcal{T} \triangleq \{\tau_1, \ldots, \tau_N\}$, and the *sequence of location indices* associated with controllable switches $\mathcal{I} \triangleq \{i(\tau_1), \ldots, i(\tau_N)\}$.

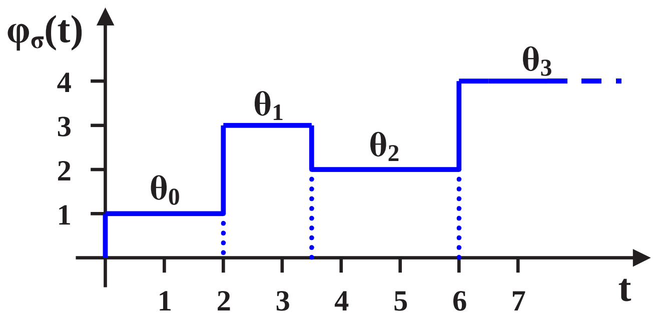


Fig. 5.1. *Function $\varphi_\sigma(t)$ of the autonomous sequence $\sigma(\boldsymbol{x}, i) = \{(1, 2), (3, 1.5), (2, 2.5), (4, +\infty)\}$.*

5.3.1 State feedback control law for AHA

In this section we show that the optimal control law for the optimization problem above takes the form of a *state feedback*, i.e., it is only necessary to look at the current system state $\boldsymbol{x}$ in order to determine if a controllable switch from location i_k to i_{k+1}, or equivalently from linear dynamics $\boldsymbol{A}_{i_k}$ to $\boldsymbol{A}_{i_{k+1}}$, should occur.

In particular, we show that for a given location i and for a given controllable switch $k \in 1, \ldots, N$ it is possible to construct a table $\mathcal{C}_k^i$ that partitions the invariant space inv_i into s_i regions $\mathcal{R}_j$'s, where $s_i = |succ_c(i)| + 1$, i.e., we can write

$$inv_i = \mathcal{R}_i \cup \left(\bigcup_{j \in succ_c(i)} \mathcal{R}_j \right).$$

Whenever $i(\tau_k + \theta) = i$ we use table $\mathcal{C}_k^i$ to determine if a switch should occur: as soon as the state reaches a point in the region $\mathcal{R}_j$ for a certain $j \in succ_c(i)$ a controllable switch will occur and we switch to mode $i(\tau_{k+1}) = j$; finally, no switch will occur while the system's state belongs to $\mathcal{R}_i$.

We simply show how the tables $\mathcal{C}_1^i$ for the last switch can be computed using the cost function associated to an autonomous evolution. The tables for the intermediate switches can also be constructed using the same dynamic programming arguments given in Chapter 4.3.

5.3.2 Computation of the tables for controllable switches

Consider a state $(\boldsymbol{x}, i)$ and let $\sigma(\boldsymbol{x}, i) = \{(i_0, \theta_0), \ldots, (i_h, \theta_h)\}$ (where $i_0 = i$) be the corresponding sequence of autonomous switches. Let us evaluate the following function:

$$\begin{aligned} J_\sigma(\boldsymbol{x}, i, \varrho) &= \textstyle\int_0^\varrho \boldsymbol{x}'(t) \boldsymbol{Q}_{\varphi_\sigma(t)} \boldsymbol{x}(t) dt \\ &= \sum_{k=0}^{\bar{h}-1} \boldsymbol{x}_k' \bar{\boldsymbol{Q}}_{i_k}(\theta_k) \boldsymbol{x}_k + \boldsymbol{x}_{\bar{h}}' \bar{\boldsymbol{Q}}_{i_{\bar{h}}}(\varrho - \sum_{k=0}^{\bar{h}-1} \theta_k) \boldsymbol{x}_{\bar{h}} \end{aligned} \tag{5.2}$$

where $\boldsymbol{x}_0 = \boldsymbol{x}$, $\boldsymbol{x}_{k+1} = \boldsymbol{M}_{i_k, i_{k+1}} \bar{\boldsymbol{A}}_{i_k}(\theta_k) \boldsymbol{x}_k$ and where $0 \leq \bar{h} \leq h$ is an integer value that depends on ϱ through the following inequalities:

$$\sum_{k=0}^{\bar{h}-1} \theta_k \leq \varrho < \sum_{k=0}^{\bar{h}} \theta_k \tag{5.3}$$

The function in (5.2) represents the cost of the evolution of the system, starting from state $(\boldsymbol{x}, i)$ and only subject to autonomous switches, for a time ϱ.

We will first explain how to build the table of the last controlled switch and then proceed recursively for the others. Assume that $i_N = i$, i.e., after $N-1$ controlled switches the current AHA state is $(\boldsymbol{x}, i)$. We show how to compute the table $\mathcal{C}_1^i$. First of all we must create $\sigma(\boldsymbol{x}, i) = \{(i_0, \theta_0), \ldots, (i_h, \theta_h)\}$.

- Consider first the case in which no controlled switch occurs. The remaining cost starting from $\boldsymbol{x}$, due to the time-driven evolution and only subject to autonomous switches is
$$T_i^*(\boldsymbol{x}, i) = J_\sigma(\boldsymbol{x}, i, +\infty). \tag{5.4}$$
- If the system evolves without performing controlled switches for a time ϱ and then a controlled switch to location j occurs, the remaining cost starting from $\boldsymbol{x}$ due to the time-driven evolution is
$$T_i(\boldsymbol{x}, j, \varrho) = J_\sigma(\boldsymbol{x}, i, \varrho) + T_j^*(\bar{\boldsymbol{x}}, j). \tag{5.5}$$
where
  - $j \in succ_c(i_{\bar{h}})$ is a controllable successor of $i_{\bar{h}}$. This set depends on ϱ through $\bar{h}$, as in Equation (5.3)
  - $\bar{\boldsymbol{x}} = \boldsymbol{M}_{i_{\bar{h}},j} \bar{\boldsymbol{A}}_{i_{\bar{h}}} (\varrho - \sum_{k=0}^{\bar{h}-1} \theta_k) \boldsymbol{x}_{\bar{h}}$ is the destination point after $\bar{h}$ autonomous switches.

The minimization of function (5.5) has to be performed over ϱ and over $j \in succ_c(i_{\bar{h}})$ (and note that $\bar{h}$ depends on ϱ). This minimization problem can be written as
$$\min_{0 \le \bar{h} \le h} \ \min_{j \,\in\, succ_c(i_{\bar{h}})} \ \min_{\varrho \in I_{\bar{h}}} \ T_i(\boldsymbol{x}, j, \varrho), \tag{5.6}$$
where $I_{\bar{h}}$ is the time interval defined by the inequalities in (5.3).

Let us denote by $\varrho^*(\boldsymbol{x}, i)$ and $j^*(\boldsymbol{x}, i)$ the values of ϱ and j that minimize (5.6). We may now indicate
$$T_i^*(\boldsymbol{x}, j^*(\boldsymbol{x}, i)) = T_i(\boldsymbol{x}, \varrho^*(\boldsymbol{x}, i), j^*(\boldsymbol{x}, i)) \tag{5.7}$$

We now show how these data are used to construct the tables for the last controllable switch.

In presence of autonomous switching regions the state space available for controllable partitions is only the inv_i. Such subspace will be then partitioned into $\mathcal{R}_j$ regions according to the following criterion:

- $\boldsymbol{x} \in \mathcal{R}_i$ if $\varrho^*(\boldsymbol{x}, i) > 0$; this physically means that the optimal strategy is to remain for a non zero time ϱ in location i;
- $\boldsymbol{x} \in \mathcal{R}_{j^*(\boldsymbol{x},i)}$ if $\varrho^*(\boldsymbol{x}, i) = 0$; this physically means that the optimal strategy is to immediately switch to location j^*.

Once the table for the last switch is constructed, it is simple to build all the others following the principle of dynamic programming and solving problem (5.6) recursively over the total number of allowed controllable switches as in [28].

5.3.3 The homogeneous case

We present now a particular class of AHA where the structure of the guards and invariants is homogeneous. Firstly we recall that a guard g_e is homogeneous if

$$(\forall \boldsymbol{x} \in g_e,\ \forall \lambda \in \mathbb{R})\ \lambda \boldsymbol{x} \in g_e.$$

Such case is meaningful because it allows one to describe guards of the form

$$\boldsymbol{x}'(t)\boldsymbol{Z}\boldsymbol{x}(t) \geq 0,$$

where $\boldsymbol{x}(t)$ is the continuous state of the hybrid system, i.e., guards given by quadratic forms.

A physical example of this is given by an electric system whose threshold

$$x_1(t)x_2(t) > 0$$

(here $x_1(t)$ and $x_2(t)$ are voltage and current, resp.) denotes the condition where the system behaves as a power generator.

Moreover, as we show in the following remark, in such conditions the computational complexity of the offline to compute the switching regions is reduced.

Remark 5.1. For each state $(\boldsymbol{x}, i)$ of an AHA with homogeneous guards, $\sigma(\boldsymbol{x}, i)$ is a homogeneous function with respect to its second variable, i.e., $\forall \lambda \in \mathbb{R} \setminus \{0\}$, $\sigma(\boldsymbol{x}, i) \equiv \sigma(\lambda \boldsymbol{x}, i)$. ■

This obvious fact implies that the residual cost $J_\sigma(\boldsymbol{x}, i, \varrho)$ given in Section 5.3.2 can be calculated only in the points $\boldsymbol{y}$ on Σ_n. In fact, knowing $J_\sigma(\boldsymbol{y}, i, \varrho)$, clearly $J_\sigma(\boldsymbol{x}, i, \varrho) = \lambda^2 J_\sigma(\boldsymbol{y}, i, \varrho)$, $\boldsymbol{x} = \lambda \boldsymbol{y}$.

As a consequence a discretization of the all invariant set inv_i is no longer required, because all the necessary information to construct the optimal switching tables can be calculated along Σ_n. Hence this special case reduces the computational complexity of the construction of table $\mathcal{C}^i_{k,N}$ [28] from $\mathcal{O}((s_i - 1)r^n)$ for the general AHA, to $\mathcal{O}((s_i - 1)r^{n-1})$, where we indicate by s_i the number of controllable edges of location i, r is the discretization sampling along each direction, n is the state space dimension.

5.4 Case b: optimal control problem for CHA

The optimal control problem for a CHA, OP(CHA) is based on the assumption that the discrete controller has at most N (fixed a priori) controllable switches available. The formal definition of the problem is given in Section 3.6.6, where all the properties, the symbols and the assumption that allow the existence of a solution are extensively described.

Here we shall limit to report the problem formulation, as it appears in Definition 3.30, and we refer the reader to the mentioned Section 3.6.6 for a complete illustration of the formalism.

$$J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} F(\mathcal{I},\mathcal{T}) \triangleq \min_{\mathcal{I},\mathcal{T}} \int_0^\infty (\boldsymbol{x}(t) - \boldsymbol{x}_{eq})' \boldsymbol{Q}_{i(t)} (\boldsymbol{x}(t) - \boldsymbol{x}_{eq}) dt$$

$$\begin{aligned} \text{s.t. } & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)} \boldsymbol{x}(t) + \boldsymbol{f}_{i(t)} \\ & i(0) = i_0 \quad \text{(initial location)} \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \quad \text{(initial state)} \\ & i(t^+) \in\ succ(\boldsymbol{x}(t), i(t)) \cup \{i(t)\} \\ & \boldsymbol{x}(t) \in inv_{i(t)} \subset \mathbb{R}^n,\ \ \forall\, t \geq 0 \end{aligned} \tag{5.8}$$

and $\mathcal{T} \triangleq \{\tau_1, \ldots, \tau_N\}$, with $\tau_0 \leq \ldots \leq \tau_k \leq \ldots \leq \tau_{N+1} = +\infty$ are the *controlled switching times* with $\tau_k - \tau_{k-1} \geq \delta_{\min}(i(\tau_{k-1})) \; \forall \; k = 1, \ldots, N+1$, the *minimum permanence time* imposed in each location.

We also have $\boldsymbol{x}(\tau_k) = \boldsymbol{x}_k$, the state reached after the $k-th$ controlled switch[2], and $i(t)$ is a $(N+1)$-piecewise constant function, defining the second set of control variables

$$\mathcal{I} \triangleq \{i(\tau_1), \ldots, i(\tau_N)\}.$$

The performance index described here weights the distance from a *target state* $\boldsymbol{x}_{eq}$. However appealing this is not the formulation of a *hybrid reachability problem*, that requires a completely different framework (see for the case the works of [81, 106]). However some authors proposed a method of solving a reachability problem via a minimization of a HJB equation, thus, to an extent, solving a particular class of optimal control problem [80].

In fact in order to be sure that the cost is finite we are forced to introduce the following assumption:

Assumption 5.1 *There exist a location i in the considered CHA, such that*

$$\boldsymbol{x}_{eq} = -\boldsymbol{A}_i^{-1} \boldsymbol{f}_i$$

with $\boldsymbol{A}_i$ strictly Hurwitz.

Assumption 5.1 is an extension of Assumption 4.1. If verified, the problem mirrors exactly the one described in Chapter 4, with the new set of variables $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}_{eq}$.

Based on the results given in Chapter 4, we show in the sequel how it is possible to construct a partition of the state space in order to determine, in state feedback form, the optimal switching signal $i(t)$, that steers the system to the target state $\boldsymbol{x}_{eq}$ minimizing the performance index of equation (5.8).

5.4.1 Case b: state feedback control law for CHA

The procedure STP that allows to solve problem (5.8) has been extensively described in Section 4.3. Hence we will not repeat here all the derivation, but we shall limit to provide the algorithm.

In fact this case is different in force of the fact that the set of successors of a given location is dependent *also* on the continuous state space $\boldsymbol{x}$, as described in the dynamical behavior of the CHA on Section 3.6.4.

For sake of clarity we report here the following remark.

Remark 5.2. An important caution should be taken when considering the successors of the current location i. In fact, let us consider the edge $e = (i, g_e, j)$. It may be activated when $\boldsymbol{x} \in inv_j$ but two different cases may occur.

1. The continuous state $\boldsymbol{x} \in inv_i \bigcap inv_j$. In this case the discrete controller has the DOF between keeping the evolution in location i or switching to location j.
2. The continuous state $\boldsymbol{x} \notin inv_i$. The evolution cannot continue in location i thus the discrete controller must leave location i.

This implies that the set of "admissible" successors also depends on the current continuous state $\boldsymbol{x}$. ■

[2] Here we assume that the state is continuous thus there are no jumps at the occurrence of a switch.

As in problem (5.8), all switching costs are null, and all jump matrices $\boldsymbol{M}_{i,j}$ are the identity.

In this affine case we can no longer restrict to the unitary semisphere, but we have to discretize the whole state space. Hence we define a rectangular grid $\mathcal{D}$ as it is shown in Figure 5.2.

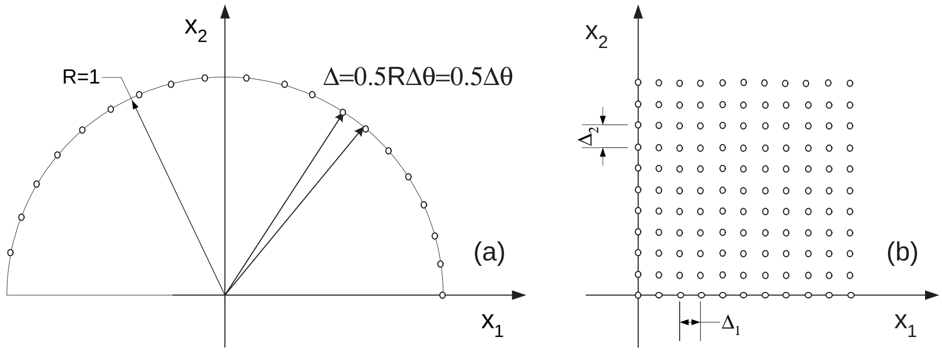


Fig. 5.2. *Different shapes of the discretization pattern in $\mathbb{R}^2$. In particular (a) spherical pattern, used when all affine terms and switching costs are null and (b) grid pattern.*

Assume that N is the number of available switches and $s = |\mathcal{S}|$, and that all dynamics $\boldsymbol{A}_i$ of the automaton are Hurwitz.

The algorithm is divided into several steps.

Algorithm 5.1 (Switching table procedure for CHA) *The input of this algorithm is the constrained hybrid automaton CHA, its target state $\boldsymbol{x}_{eq}$, its annexed optimal control problem OP and the number of available switches N, a tuning parameter $t_{\max}$ that expresses the duration of the future exploration.*

The output is a set of $N \times s$ tables that the controller can use to provide the feedback control law during the real time evolution.

The list of instructions is depicted in Figure 5.3. ■

The main advantage of the proposed procedure may be briefly summarized as follows.

- It is guaranteed to find the optimal solution to problem (5.8).
- It provides the global optimal solution, i.e., the tables may be used to determine the optimal state feedback control law for all initial states.

The optimal control law can be computed as follows. For a given location i and for a given switch $k \in \{1, \ldots, N\}$, it is possible to construct a table $\mathcal{C}_k^i$ that partitions the invariant set inv_i into up to $s_i = |succ(i)| + 1$ regions $\mathcal{R}_j$'s. Whenever $i_{k-1} = i$ we use table $\mathcal{C}_k^i$ to determine if a switch should occur: as soon as the state reaches a point in the region $\mathcal{R}_j$ we will switch to mode $i_k = j$ provided that the minimum permanence time $\delta_{\min}(i)$ has elapsed; on the contrary no switch will occur while the system state belongs to $\mathcal{R}_i$.

5.5 Numerical examples

We provide in this section two numerical examples, one referred to the case (a), i.e., when an autonomous sequence is allowed, the other referred to the case (b), i.e., the set of successors of a given location is state dependent.

1. Initialization: $k = 0$ remaining switches
   Redefine $\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{x}_{eq}$.
   For $i = 1 : s$
   Calculate if possible $\boldsymbol{Z}_i \ : \ \boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i = -\boldsymbol{Q}_i$.
   $\forall\ \boldsymbol{y} \ \in \ \mathcal{D}$
   Cost assignment
   $T_0(\boldsymbol{y}, i) = \begin{cases} \boldsymbol{y}'\boldsymbol{Z}_i\boldsymbol{y} & \text{if} \quad \boldsymbol{f}_i = \boldsymbol{0} \\ +\infty & \text{else.} \end{cases}$
   Color assignment
   $C_0(\boldsymbol{y}, i) = i$
   end (i)
2. For $k = 1 : N$
   For $i = 1 : s$
   $\forall\ \boldsymbol{y} \ \in \ \mathcal{D}$
   Compute the set $succ(\boldsymbol{y}, i)$;
   Remaining cost:
   $t = 0$
   While $t < t_{\max} \wedge \boldsymbol{y} \ \in \ inv_i$
   For $j \in succ(\boldsymbol{y}, i)$
   $\boldsymbol{y} \leftarrow \bar{\boldsymbol{A}}_i'(t)\boldsymbol{y}$
   $\boldsymbol{y}(j, t) = \bar{\boldsymbol{A}}_j(\delta_j)\boldsymbol{y}$
   $T(\boldsymbol{y}, i, j, t) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_i(t)\boldsymbol{y} + \boldsymbol{y}'\bar{\boldsymbol{A}}_i'(t)\bar{\boldsymbol{Q}}_j(\delta_j)\bar{\boldsymbol{A}}_i'(t)\boldsymbol{y} + T_{k-1}(\boldsymbol{y}(j,t), j)$
   end (j)
   $t \leftarrow t + dt$.
   end (t)
   Cost assignment
   $$T_k(\boldsymbol{y}, i) = \min_{j,t} T(\boldsymbol{y}, i, j, t)$$
   Color assignment
   $$(j^*, t^*) = \arg\min_{j,t} T(\boldsymbol{y}, i, j, t).$$
   $$C_k(\boldsymbol{y}, i) = \begin{cases} j^* & \text{if} \quad t^* = 0 \\ i & \text{if} \quad t^* > 0. \end{cases}$$
   end (i)
   end (k)

Fig. 5.3. *Algorithm for the implementation of the STP in presence of state space constraints as in the model CHA.*

5.5.1 Case a: an AHA example in the homogeneous case

Let us consider the AHA whose graph is reported in Figure 5.4 where dashed arrows have been used to denote edges associated to autonomous switches, while continuous arrows have been used to denote edges associated to controllable switches.

In this particular $\mathbb{R}^2$ case, guards and invariants of the automaton are homogeneous. In such a case they may be easily described [90] as quadratic forms of $\boldsymbol{x}$. In particular, we assume that the guards associated to autonomous switches are[3]

$$g_{1,2} = \{\boldsymbol{x} \in \mathbb{R}^2 |\ \boldsymbol{x}'\boldsymbol{G}_{1,2}\boldsymbol{x} \geq 0\},\ \boldsymbol{G}_{1,2} = \begin{bmatrix} -0.2 & 0.6 \\ 0.6 & -1 \end{bmatrix}$$

$$g_{1,3} = \{\boldsymbol{x} \in \mathbb{R}^2 |\ \boldsymbol{x}'\boldsymbol{G}_{1,3}\boldsymbol{x} \geq 0\},\ \boldsymbol{G}_{1,3} = -\begin{bmatrix} 1 & 1.25 \\ 1.25 & 1 \end{bmatrix}$$

[3]To avoid a heavy notation we denote here $g_{i,j}$ the guard associated to edge $e_{i,j}$.

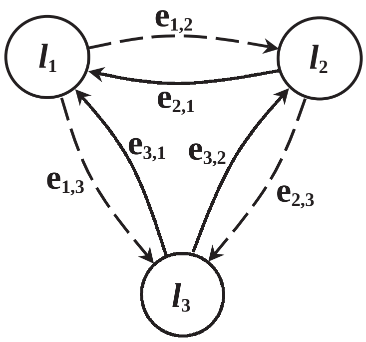


Fig. 5.4. *Oriented graph of the AHA considered in Example 5.5.1. The dashed arcs represent the autonomous edges, while the continuous arcs represent the controllable edges.*

and

$$g_{2,3} = \{\boldsymbol{x} \in \mathbb{R}^2 | \ \boldsymbol{x}'\boldsymbol{G}_{2,3}\boldsymbol{x} \geq 0\}, \ \boldsymbol{G}_{2,3} = \begin{bmatrix} -3 & 0.5 \\ 0.5 & 0 \end{bmatrix}$$

where $g_{1,2} \cap g_{1,3} = \emptyset$.

Consequently, by Assumption 3.1 given in Chapter 3, the invariant sets may be defined as

$$\begin{aligned} &inv_1 = \mathbb{R}^2 \setminus (g_{1,2} \cup g_{1,3}), \\ &inv_2 = \mathbb{R}^2 \setminus g_{2,3}, \qquad\qquad inv_3 = \mathbb{R}^2, \end{aligned}$$

while the guards associated to controllable switches are

$$g_{2,1} = inv_2, \ g_{3,1} = g_{3,2} = inv_3.$$

The above set of guards and invariants are shown in Figure 5.5.

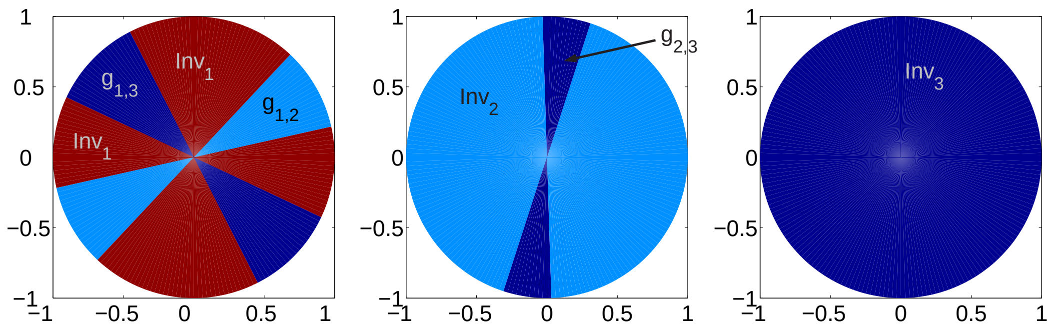


Fig. 5.5. *The guards and invariants of the AHA in Example 5.5.1.*

This automaton is also homogeneous, thus it allows to perform calculations along Σ_2.

Let us assume that the activity functions at the discrete locations are defined by the following matrices:

$$\boldsymbol{A}_1 = \begin{bmatrix} -1.85 & -1 \\ 1 & 0 \end{bmatrix}, \boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ -0.74 & -1.29 \end{bmatrix}, \boldsymbol{A}_3 = \begin{bmatrix} -2.75 & -2.84 \\ 1 & 0 \end{bmatrix}.$$

All jumps are coincident with the identity relation, i.e., $\boldsymbol{M}_{i,j} = \boldsymbol{I}_2$, for all i, j with $i \neq j$, where $\boldsymbol{I}_2$ denotes the second order identity matrix.

Finally we assume that weighting matrices are coincident with the identity matrix, and that $N = 3$ controllable switches are allowed.

To solve the resulting optimal control problem, we first evaluate offline the $N \times s$ controllable switching tables, using the procedure presented in the Subsection 5.3.2.

In this particular case 9 tables have been constructed (3 for every switch).

A space discretization of 101 points along Σ_2 and a local minimum search within five time constants have been considered sufficiently fine.

Provided such tables, the controller/supervisor is ready (and fast) to estimate the optimal strategy in real time mode subject to the constraints of the automaton.

The state trajectory that minimizes the performance index is depicted in Figure 5.6, where the black squares indicate the controllable switches and the red stars indicate the autonomous switches.

Finally, we found out the following values of the switching (both controllable and autonomous) instants $\mathcal{T}$, of the optimal sequence $\mathcal{I}$, and of the optimal cost J:

$$
\begin{aligned}
&\mathcal{T} = \{0.05,\ 0.11,\ 0.11,\ 0.78,\ 0.96,\ 1.505\} \\
&\mathcal{I} = \{3\ \Rightarrow\ 1\ \rightarrow\ 3\ \Rightarrow\ 2\ \rightarrow\ 3\ \Rightarrow\ 2\ \rightarrow\ 3\} \\
&J = 62.15
\end{aligned}
$$

In the subset $\mathcal{I}$ the arrow $\Rightarrow$ indicates a controllable switch, and the arrow $\rightarrow$ indicates an autonomous switch.

The system initially sojourns in location 3 then the supervisor switches to location 1. Tables indicate that it is worth waiting until the autonomous threshold with location 3, in order to go directly to location 2 in zero time. Now it is better to remain in location 2 until the autonomous boundary is reached before using the third controllable switch, which takes place during the evolution in location 3. From now on the system evolves independently towards zero, performing a finite number of autonomous switches.

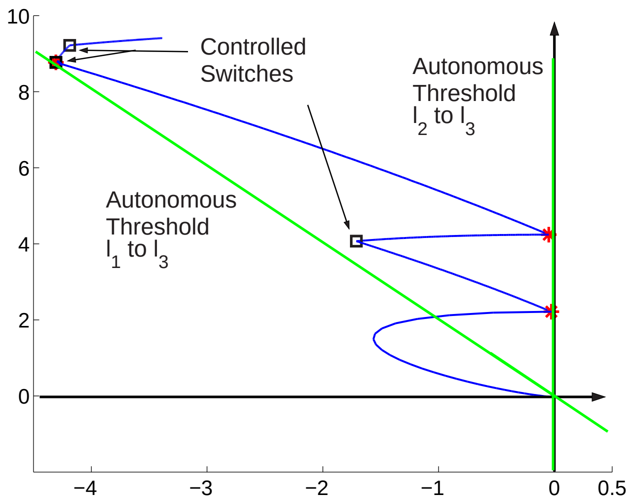


Fig. 5.6. *System evolution for $\boldsymbol{x}(0) = [-3.4, -9.4]'$, and initial location 3. A square denotes a controlled switch. A star denotes an autonomous switch.*

5.5.2 Case b: application and case study

As an example of the described procedure, the following problem is considered. This problem was inspired by [25]. A physical system is composed of two cylindric tanks, equipped with inflow pipe and subject to leakage (see Figure 5.7).

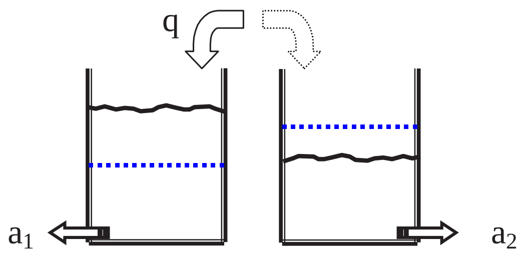


Fig. 5.7. *Schematic view of the physical system considered for the Example 5.5.2.*

The continuous variables of this system are the levels of the fluid in each tank, namely $\boldsymbol{x} = [x_1, x_2]'$.

The physical dimension of the tanks imposes a minimum and a maximum value of the fluid level, i.e.,

$$\boldsymbol{x} \in X = [0, 30] \times [0, 20].$$

The level $\boldsymbol{x}_j$ in each tank is governed by the linearized DE, namely

$$\begin{aligned} \dot{x}_1 &= -a_1 x_1 + f_{i,1}, \\ \dot{x}_2 &= -a_2 x_2 + f_{i,2} \end{aligned}$$

where a_1, a_2 are the flow losses of tank 1 and tank 2, $\boldsymbol{f}_i \triangleq [f_{i,1}, f_{i,2}]'$ is the flow input, described next. We assign in this example $a_1 = 2$ and $a_2 = 3$.

The inflow pipe (Figure 5.7) is capable of a flow rate $q = 60$, but it may only assume *quantized positions* taken from a finite set

$$\mathcal{Q} = \{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_5\}$$

where

$$\boldsymbol{f}_1 = \begin{bmatrix} q \\ q \end{bmatrix}, \boldsymbol{f}_2 = \begin{bmatrix} 0 \\ q \end{bmatrix}, \boldsymbol{f}_3 = \begin{bmatrix} q \\ 0 \end{bmatrix}, \boldsymbol{f}_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{f}_3 = \frac{1}{2} \begin{bmatrix} q \\ q \end{bmatrix}.$$

The global linearized DE of the system is thus

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix} \boldsymbol{x} + \boldsymbol{f}_i,$$

$i = 1 \ldots 5$, where $\boldsymbol{A}$ represents the linearized fluid loss due to static height pressure.

The resulting system can be modelled as a CHA composed of 5 locations, as depicted in Figure 5.8. The structure of the automaton takes into account the order in which the different inflow rates can be changed.

Furthermore we consider an extra specification based on Figure 5.10. In particular the output signal of the system is of three levels:$\{Y_d^+, Y_d^0, Y_d^-\}$, denoting respectively, that the continuous state $\boldsymbol{x}$ is in the *safe* region (i.e., the interior part of X), the *conditionally safe* region (i.e., the sides of X), and the *unsafe* region (i.e., the corners of X).

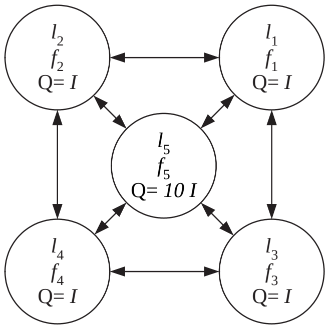


Fig. 5.8. *The HA modelling the considered affine system. The double arrows indicate that both switching directions are allowed.*

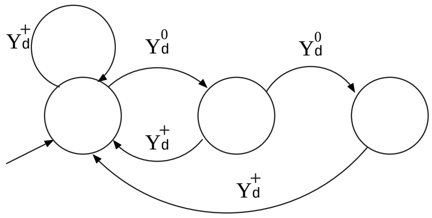


Fig. 5.9. *Specification for the outputs, safe, conditionally safe and unsafe regions.*

We impose a specification on the sequences of outputs represented in Figure 5.9. More precisely this automaton imposes the following requirements:

1. if the state is in the safe region, then the next output symbol can either be Y_d^+ or Y_d^0 (meaning respectively that the state will remain in the safe region or may enter the conditionally safe region;
2. if the state is in the conditionally safe region, then the next output symbol can either be Y_d^+ or Y_d^0 (meaning respectively that the state will go back to the safe region or may remain in the conditionally safe region and potentially the unsafe region);
3. if the state is in the unsafe region, then the next output symbol can only be Y_d^+ (meaning that the state will go back to the safe region in not more than one step).

Thus, the specification requires that the state can belong to the *conditionally* safe region for no longer than two time intervals. After that, the system should stay at least one time interval within the *safe* region producing the corresponding output symbol.

In the low level step, a procedure based on l-complete approximation and supervisory control theory, described in Appendix D the specification of outputs depicted in Figure 5.9 is converted into a set of invariants, i.e., constraints on the state space, which are attached to the swithced system to form the CHA on which we finally apply the Algorithm 5.1.

These invariants restrict the behavior of the overall system to guarantee the safety and liveness conditions. The invariant set of locations $1, \ldots, 5$ are reported in Figure 5.11.

The high-level step requires the solution of an optimal control problem of the form (5.8). The weight matrices $\boldsymbol{Q}_i$ are indicated for each location in Figure 5.8 where $\boldsymbol{I}$ denotes the identity matrix.

The maximum number of switches is $N = 3$. The target state is equal to

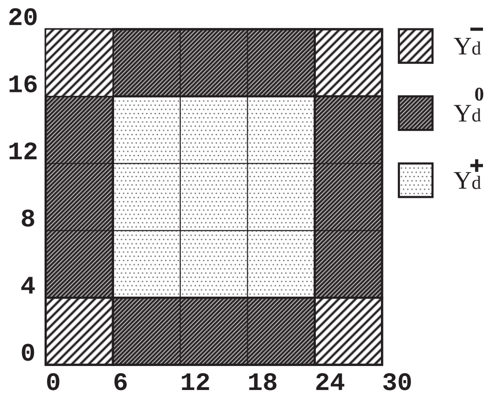


Fig. 5.10. *Partitioning of the state space* X.

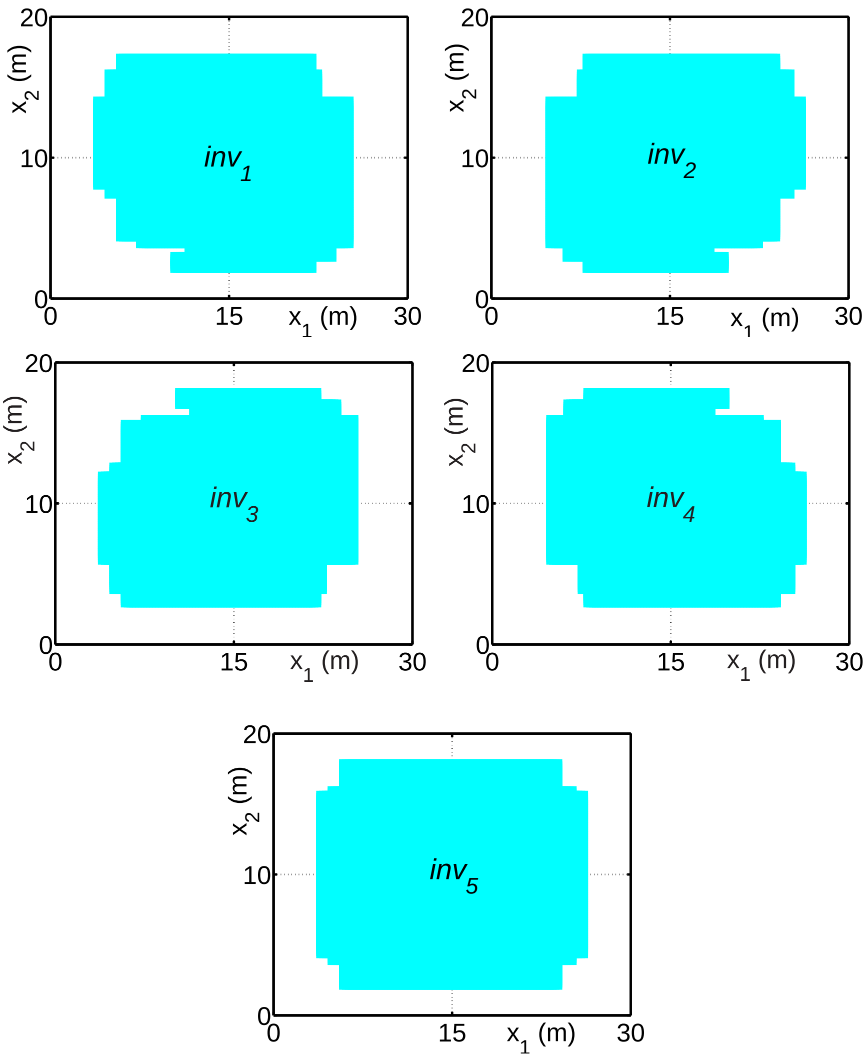


Fig. 5.11. *Invariant regions for locations* $1, \ldots, 5$.

$$\boldsymbol{x}_{eq} = \begin{bmatrix} 15 \\ 10 \end{bmatrix},$$

that satisfies Assumption 5.1 for $i = 5$.

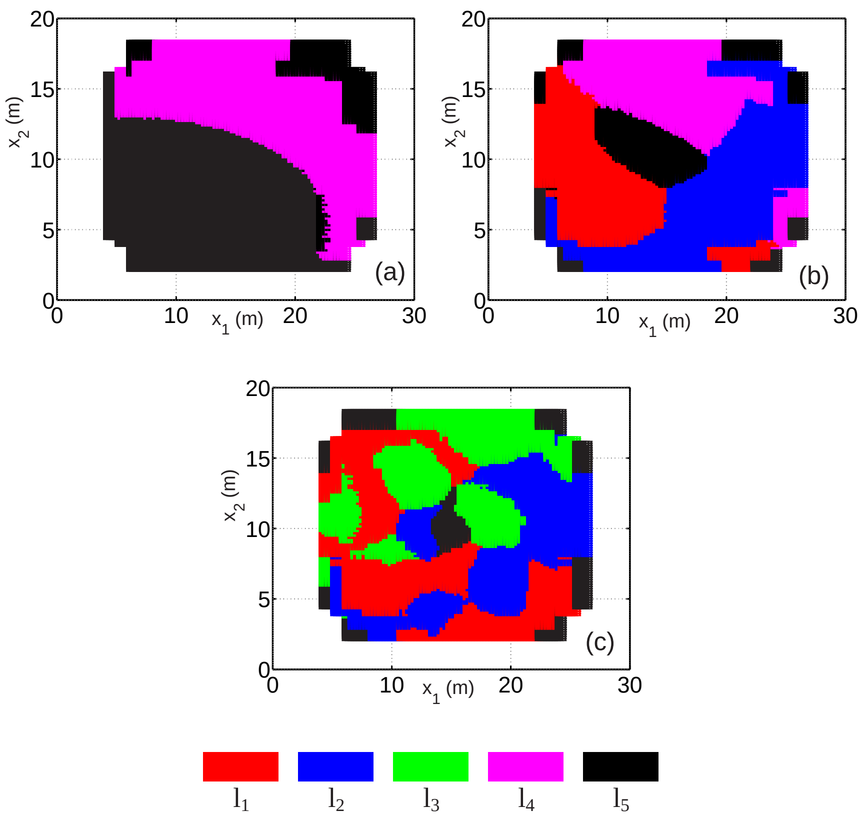


Fig. 5.12. *Switching tables used by the controller during the simulation described in the example. (a) is the table used when 1 switch is available, (b) when 2 switches are available, (c) when 3 switches are available.*

The offline part of this procedure consists in the construction of $5 \times 3 = 15$ tables, one per each location and per number of available switches. A state space discretization is a grid of 125×125 points. The minimum search algorithm works on a time domain of $t = 5s$ with time step $0.1s$. The latter value was chosen to guarantee an appropriate synchronization between the two levels. The offline calculation effort for this step of the problem took approximately 2 hours, on a common commercial laptop with average up to date performances. For sake of brevity we only report some of these tables (depicted in Figure 5.12), i.e., those tables used by the controller during the simulation ran for the initial continuous state

$$\boldsymbol{x} = \begin{bmatrix} 21 \\ 3 \end{bmatrix}$$

and the initial location 1. The trajectory obtained for this particular value of the initial state is plotted in Figure 5.13. The optimal switching sequence and switching times are

$$\mathcal{I} = \{1, 2, 4, 5\}$$

and

$$\mathcal{T} = \{0.121, 0.221, 0.321\},$$

and the optimal cost is

$$J_3^* = 19.47.$$

Note that, due to the minimum permanence time within each location, it may occur that the switching from one discrete location to another, does not necessarily occur as soon as the state trajectory exits the current region. This is the case of the last switching point of the trajectory reported in Figure 5.13.

The controller uses the appropriate switching tables to impose the appropriate switching. The simulation does not require any extra calculations other then observing the state space and compare its value with the switching table (already calculated) corresponding to the current location and to the current number of remaining switches.

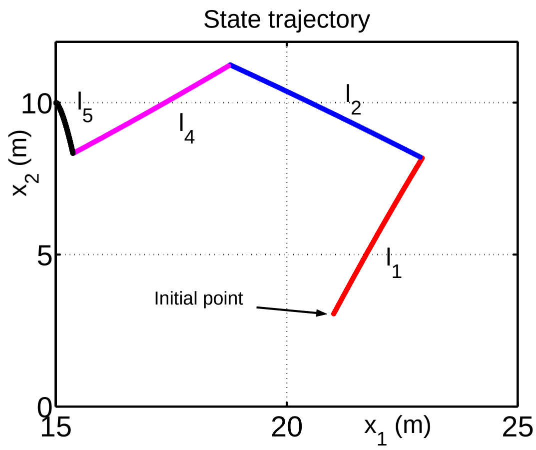


Fig. 5.13. *State space trajectory and discrete location sequence.*

5.6 Conclusions

In this chapter we analyzed the problem of providing a feedback optimal control law for a switched system in presence of state space constraints, that can be seen as a generalization of the class of switched system we have considered in Chapter 4. This led us to the introduction of a the more general model, i.e., the HA, featured by constraints on the state space.

In particular we studied two cases.

5.6.1 Case a

A class of HA that we called Autonomous Hybrid Automata whose main aspect is that not only the continuous time evolution $\boldsymbol{x}(t)$ is autonomous, but also the discrete event evolution $i(t)$ is autonomous and it follows an evolution governed by autonomous, internally forced switches.

This is really a dangerous aspect in the framework of HS, because it is well known that in general an autonomous evolution of discrete events may provoke instability of the system.

We provided sufficient conditions to ensure that this does not occur. Although may not be restrictive, these conditions are structural on the AHA.

In this model there are two types of edges: firstly a controllable edge represents a mode switch that can be triggered by the controller; secondly an autonomous edge represents a mode switch that is triggered by the continuous state of the system as it reaches a given threshold.

We have shown how the special structure of autonomous hybrid automata allows one to solve an infinite horizon quadratic optimization problem with a numerically viable procedure; the optimal control law takes the form of a state-feedback.

The application of the STP is not straightforward for this class. In fact, during the time search subroutine, there exists the possibility of starting an autonomous sequence.

5.6.2 Case b

In this case two approaches based respectively on discrete approximation of continuous systems and optimal control of switched systems, were successfully cast and merged to the framework of a HA.

More precisely a the discrete approximation part, i.e., the low level part, converts some specifications on the output signals of the plant into constraints on the state space. The approach is described in Appendix D.

In this case the autonomous switches are not admissible, but the set of successors, that in Chapter 4 was a function of the current discrete state, is now a function of the hybrid state $(\boldsymbol{x}, i)$.

The oriented graph of the automaton is state dependent, i.e., some arcs may be "forbidden" according to the value of the state space.

The STP of Chapter 4 can be extended provided that now a dynamic value of the set of successors must be taken into account.

Both cases, apart from extremely special shapes of the constraints, require the discretization of the whole state space, or, which is equivalent, the sampling of Σ_{n+1}.

6

Infinite number of switches

6.1 Introduction

In this chapter we focus our attention on the optimal control problem of a switched system when an infinite number of switches is allowed.

In Chapter 4 we assumed that an upper bound on the maximum number N of available switches is imposed. We developed the STP that provides the state feedback optimal control law for this particular case.

Here, under reasonable assumptions, we show how the proposed procedure can be extended to the case of $N = \infty$. In other words we will provide a constructive method to design a switching table that can be used indefinitely until the continuous state $\boldsymbol{x}$ of the switched system has practically reached the origin.

Furthermore, since this switching law is based on the minimization of a piecewise LQ performance index, it is also optimal.

The case $N = \infty$ contains interesting theoretical developments, such as the convergence of the switching tables, and relevant practical applications. In fact the majority of real systems are able to infinitely switch.

As an example the approach has been applied to the servomechanism system studied in Section 4.5. As a real case study we considered the design of a semi active suspension system, to which we dedicated part of this chapter.

6.2 The model and the optimal control problem

In this section we recall the model and the optimal control problem defined in Chapter 3 that we consider in this chapter.

6.2.1 The model: switched system

We consider a switched system $S = (\mathcal{L}, act, \mathcal{E}, \mathcal{M})$, extensively described in Section 3.3, in consistency with Definition 3.2.

We recall that:

— $\mathcal{L}$ is a finite set of locations.

— $act : \mathcal{L} \rightarrow$ *Diff_Eq* is a function that associates to each location i a linear affine DE of the form $\dot{\boldsymbol{x}} = act_i(\boldsymbol{x}) = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{f}_i$.

— $\mathcal{E} \subset \mathcal{L} \times \mathcal{L}$ is the set of edges. An edge $e = (i, j)$ is an arc from location i to j, $i \neq j$.

— $\mathcal{M} : \mathcal{E} \rightarrow \mathbb{R}^{n\times n}$ associates to each edge $e \in E$ a constant matrix in $\mathbb{R}^{n\times n}$, that represents the linear resetting of the state space $\boldsymbol{x}$ at the switching instants.

We recall the definition of the set of successors, as in Definition 3.4, $succ(i)$ which denotes the set of indices associated to the locations reachable from location i.

Once entered in a location i a *minimum permanence time* $\delta_{\min}(i)$ must elapse before the controller may decide the best strategy, whose formal Definition is 3.7.

In this chapter, wlg, we will restrict the analysis to the following class:

1. The *hybrid evolution* $(\boldsymbol{x}(t), i(t))$ is continuous in $\boldsymbol{x}$, i.e., there are no state jumps;
2. The affine terms $\boldsymbol{f}_i$ are all null;
3. The item (i) of Assumption 4.1 is verified, i.e., at least one dynamics of S is strictly Hurwitz.

6.2.2 The optimal control problem: infinite number of switches

For the class of system defined above we consider the optimal control problem OP$_\infty(S)$, as in Definition 3.15, reported below.

$$
\begin{aligned}
&J_\infty^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt \right\} \\
&\text{s.t. } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t) + \boldsymbol{f}_{i(t)}, \ \ \boldsymbol{x}(0) = \boldsymbol{x}_0, \ \ i(0) = i_0 \\
&\qquad i(t) = i_k \ \in \ succ(i_{k-1}) \text{ for } \tau_k \le t < \tau_{k+1}, \\
&\qquad \tau_{k+1} \ge \tau_k + \delta_{\min}(i_k),
\end{aligned}
\tag{6.1}
$$

where all terms, symbols and control variables are described in Section 3.4. Nevertheless we would like to recall that the control variables are

$$\mathcal{T} \triangleq \{\tau_1, \tau_2, \ldots\}$$

and

$$\mathcal{I} \triangleq \{i_1, i_2, \ldots\},$$

where $\mathcal{T}$ is the *sequence of switching times* and $\mathcal{I}$ is the sequence of indices as in Definitions 3.11 and 3.12 respectively. Note that these sets are unlimited.

In fact the subscript ∞ indicates that we relax the restriction considered in the previous chapters allowing that the number of switches may be infinite.

To solve this problem we initially assume that the number of available switches is finite and equals to N, thus we consider a problem OP$_N(S)$, as in Definition 3.14and reported below.

$$
\begin{aligned}
&J_N^* \triangleq \min_{\mathcal{I},\mathcal{T}} \left\{ F(\mathcal{I},\mathcal{T}) \triangleq \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt \right\} \\
&\text{s.t. } \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t), \ \ \boldsymbol{x}(0) = \boldsymbol{x}_0, \ \ i(0) = i_0 \\
&\qquad i(t) = i_k \text{ for } \tau_k \le t < \tau_{k+1}, \quad k = 0, \ldots, N \\
&\qquad \tau_0 = 0, \ \tau_{N+1} = +\infty \\
&\qquad \tau_{k+1} \ge \tau_k + \delta_{\min}(i_k), \qquad\qquad k = 0, \ldots, N \\
&\qquad i_{k+1} \in succ(i_k), \qquad\qquad\qquad k = 0, \ldots, N
\end{aligned}
\tag{6.2}
$$

In the sequel we will refer to the solution of the finite problem as J_N^*, and to the solution of the infinite problem as J_∞^*.

6.3 State feedback control law

We briefly recall the procedure STP that gives the solution of Problem (6.2) in feedback form, described in Section 4.3, derived for a finite number of switches N.

The general idea of the STP is to proceed backward from the last switch $k = 0$, and obtain, for any given hybrid point $(\boldsymbol{x}, j_k)$, when k switches remain, a residual cost of the form

$$\begin{aligned} T_k(\boldsymbol{x}, j_k, \dots, j_0, \varrho_k, \dots, \varrho_0) = F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + \\ + T_{k-1}(\boldsymbol{z}, j_{k-1}, \dots, j_0, \varrho_{k-1}, \dots, \varrho_0) \end{aligned} \tag{6.3}$$

where

$$\boldsymbol{z} = \bar{\boldsymbol{A}}_{j_{k-1}}(\delta_{\min}(j_{k-1}))\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x}$$

expresses the state reached after the evolution for ϱ_k time units in location j_k and a switch to j_{k-1} for $\delta_{\min}(j_{k-1})$ time units[1].

It has been proved that

$$T_k^*(\boldsymbol{x}, j_k) = \min_{j_{k-1}, \varrho_k} T_k(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) \tag{6.4}$$

with $j_{k-1} \in succ(j_k)$ and $\varrho_k \geq 0$, provided that we take

$$T_k(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) = F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) + T_{k-1}^*(\boldsymbol{z}, j_{k-1}, \dots, j_0, \varrho_{k-1}, \dots, \varrho_0),$$

in agreement with the well known principle of optimality [3, 14, 69].

In this case, (all switching costs are null and all jumps are the identity matrix)

$$F(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) = \boldsymbol{x}'\bar{\boldsymbol{Q}}_{j_k}(\varrho_k)\boldsymbol{x} + \boldsymbol{x}'\bar{\boldsymbol{A}}'_{j_k}(\varrho_k)\bar{\boldsymbol{Q}}_{j_{k-1}}\bar{\boldsymbol{A}}_{j_k}(\varrho_k)\boldsymbol{x}, \tag{6.5}$$

as it was defined in Definition 4.4 and explained there on.

The strategy associated to the current hybrid state $(\boldsymbol{x}, j_k)$, when k switches are missing, is thus dependent, as explained in Definition 4.7 on the values

$$\begin{aligned} \varrho_k^*(\boldsymbol{x}, j_k) &= \arg \min_{j_{k-1}, \varrho_k} T_k(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k) \\ j_{k-1}^*(\boldsymbol{x}, j_k) &= \arg \min_{j_{k-1}, \varrho_k} T_k(\boldsymbol{x}, j_k, j_{k-1}, \varrho_k), \end{aligned}$$

wrt the lexicographic ordering of Paragraph 4.3.2.

By induction of k up to N we terminate the procedure of table construction.

Things are numerically simplified when all switching costs and affine terms are null, as it is the case we consider in this chapter. In fact, thanks to the 2-homogeneity (see Definition 4.2) of terms in equation (6.3), the investigation can be limited to Σ_n.

[1]For a detailed definition of all elements of (6.3) see Paragraph 4.3.1.

6.4 Conjecture

The results given in Paragraph 4.3.1, resumed in the previous section, may naturally lead to the following question:

What happens if N keeps increasing?

We may provide the following conjectures:

Conjecture 6.1 (Convergence of the switching tables) *The tables $\mathcal{C}_N^i$, $i \in \mathcal{S}$, constructed with the STP for increasing values of N*, converge *to a final set of tables that we can call $\mathcal{C}_\infty^i$.* ■

This conjecture, formally proved in Section 6.5.2, may be deduced as follows: if the number of available switches is N, where N is a sufficiently large integer, then in a given point $(\boldsymbol{x}, i)$ the optimal strategy, i.e., the color of the table $\mathcal{C}_N^i$ in $\boldsymbol{x}$, should be the same as if we consider the table $\mathcal{C}_{N+1}^i$ in $\boldsymbol{x}$, obtained with the STP applied to a problem with $N+1$ allowed switches.

Conjecture 6.2 (Convergence of the cost) *The optimal cost from point $(\boldsymbol{x} \neq \boldsymbol{0}, i)$, namely $J_N^*(\boldsymbol{x}, i)$, calculated for increasing values of N, is a decreasing function of N and it* converges *to a strictly positive lower bound $J_\infty^*(\boldsymbol{x}, i)$.* ■

The first part of Conjecture 6.2 can be deduced by the following consideration: augmenting the number of available switches is equivalent, to an extent, to relax the number of constraints in a minimization problem. The solution of such a problem (with fewer constraints) can only *improve*, permitting us to foresee that the cost is a decreasing function of N. The second part of the conjecture comes from the fact that any evolution that starts from $\boldsymbol{x} \neq \boldsymbol{0}$, has necessarily a strictly positive cost. This conjecture is formally proved in Section 6.5.1

Assume now that the convergence of the tables is observed when $\bar{N}$ switches are allowed. Conjecture 6.1 allows one to use indefinitely only the tables $\mathcal{C}_\infty^i \equiv \mathcal{C}_{\bar{N}}^i$ during an evolution that admits an infinite number of switches. The cost of this evolution must be J_∞^*.

We may also provide the following conjecture:

Conjecture 6.3 (Cost reduction) *For any point $(\boldsymbol{x} \neq \boldsymbol{0}, i)$ and* for all $N \in \mathbb{N}$, *the costs*

1. *$J_N^*(\boldsymbol{x}, i)$, i.e., the cost obtained performing N switches,*
2. *$J_{N,\infty}^*(\boldsymbol{x}, i)$, i.e., the cost obtained performing ∞ switches, and using only tables $\mathcal{C}_N^i$,*
3. *$J_\infty^*(\boldsymbol{x}, i)$, i.e., the cost obtained performing ∞ switches, and using only tables $\mathcal{C}_\infty^i$,*

are related as

$$J_\infty^*(\boldsymbol{x}, i) \leq J_{N,\infty}^*(\boldsymbol{x}, i) \leq J_N^*(\boldsymbol{x}, i).$$

■

We formally prove the extreme parts of the inequality, i.e., $J_\infty^*(\boldsymbol{x}, i) \leq J_N^*(\boldsymbol{x}, i)$, as stated above. The intermediate property is not proved yet.

This last conjecture states that a reduction of the cost can be obtained albeit the tables haven't converged yet. In fact using indefinitely the last calculated tables, namely $\mathcal{C}_N^i$, it should hold $J_{N,\infty}^*(\boldsymbol{x}, i) \leq J_N^*(\boldsymbol{x}, i)$.

What we find interesting in this conjecture is that it permits to economize in terms of computational effort. In fact if we are not able to compute tables until the convergence is met, we may consider the last calculated ones and assume as optimal the trade off value $J_{N,\infty}^*(\boldsymbol{x}, i)$, which is worst than $J_\infty^*(\boldsymbol{x}, i)$ but better than $J_N^*(\boldsymbol{x}, i)$.

Let us now consider the following example, through which we would like to highlight the results claimed by the three conjectures above.

6.4.1 An example

Let us consider a switched system composed of three locations $i = 1, 2, 3$ and a set of edges $\mathcal{E}$, whose oriented graph is depicted in Figure 6.1(a).

The dynamics associated to each location are:

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 9 \\ -1 & -0.5 \end{bmatrix} \boldsymbol{A}_2 = \begin{bmatrix} 3.09 & 2.78 \\ -7.22 & -3.59 \end{bmatrix} \boldsymbol{A}_3 = \begin{bmatrix} -3.84 & 3.22 \\ -6.78 & -3.34 \end{bmatrix}. \tag{6.6}$$

The single trajectories of each dynamics are juxtaposed in Figure 6.1(b) for three different initial states (a color mapping is given in Figure 6.2). It can be seen that all dynamics are stable. The dynamics $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$ are obtained from $\boldsymbol{A}_1$ by a rotation of $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ respectively.

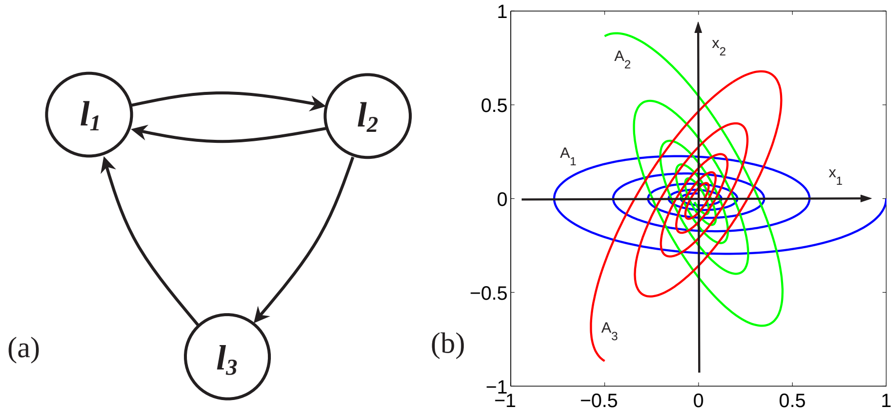


Fig. 6.1. *(a):Oriented graph of the example detailed in Section 6.4.1. (b): Trajectories of the three dynamics considered in Section 6.4.1. The blue evolution is dynamics $\boldsymbol{A}_1$, the green $\boldsymbol{A}_2$, the red $\boldsymbol{A}_3$ (Figure 6.2).*

For sake of completeness:

$$\boldsymbol{A}_2 = \boldsymbol{T}^{-1}\boldsymbol{A}_1\boldsymbol{T}, \;\; \boldsymbol{A}_3 = \boldsymbol{T}^{-1}\boldsymbol{A}_2\boldsymbol{T}$$

where

$$\boldsymbol{T} = \begin{bmatrix} \cos(\frac{2\pi}{3}) & \sin(\frac{2\pi}{3}) \\ -\sin(\frac{2\pi}{3}) & \cos(\frac{2\pi}{3}) \end{bmatrix}.$$

All jumps are the identity matrix, i.e., the time driven evolution $\boldsymbol{x}(t)$ is continuous.

A minimum permanence time is required in each location, thus:

$$\delta_{\min}(1) = 0.1 \quad \delta_{\min}(2) = 0.1 \quad \delta_{\min}(3) = 0.3$$

The optimal control problem is in form (6.2), and the matrices $\boldsymbol{Q}_i$, $i = 1, 2, 3$ are chosen all equal to $\boldsymbol{I}_2$.

Location	Color mapping
l_1	
l_2	
l_3	

Fig. 6.2. *Color mapping of the locations* $1, 2, 3$ *described in Section 6.4.1.*

The general setup is very simple.

Now we fix the parameter $N = 10$ as the maximum number of allowed switches and we start to perform the STP over the variable $k = 0, \ldots, 10$, to construct $\mathcal{C}_k^i$.

It takes approximately 50 minutes with a discretization of 51 points on Σ_2.

The STP produces 30 tables, being $(N = 10) \times (s = 3)$. These tables are used according to the current hybrid state $(\boldsymbol{x}, i)$ and to the number of switches that have been done, as described in Section 4.3.

Note that the tables depicted in Figure 6.3 converge from approximately the value of $N = 7$, thus the Conjecture 6.1 is verified.

Consider now the initial point $\boldsymbol{x}_0 = [0, 1]'$ and initial location $i = 1$. We evaluate the cost from this given point as a function of N. The plot is depicted in Figure 6.4.

It is clear from Figure 6.4 that the cost is a *lower bounded non increasing function* of the number of allowed switches N.

Finally we consider 51 initial points, on Σ_2, parameterized in ϑ, i.e.,

$$\left(\boldsymbol{x}_0(\vartheta_j) = \begin{bmatrix} \cos(\vartheta_j) \\ \sin(\vartheta_j) \end{bmatrix}, i_0 = 1 \right),$$

with

$$\vartheta_j = j \frac{\pi}{50}, \quad j = 0, \ldots, 50.$$

From each one of these point we calculate the cost obtained with a performance of up to N switches, namely $J_N^*(\boldsymbol{x}(\vartheta_j), i = 1)$ and $J_{N,\infty}^*(\boldsymbol{x}(\vartheta_j), i = 1)$ as described in Conjecture 6.3.

The significant result is reported in Figure 6.5 , where we depict the function

$$f_N(\vartheta_j) = \frac{J_N^*(\vartheta_j) - J_{N,\infty}^*(\vartheta_j)}{J_N^*(\vartheta_j)} \%,$$

that represents the *normalized difference*[2] (in percentage) of these two values of the cost.

Note that *for all* values of N and *for all* initial points the function $f_N(\vartheta_j)$ is positive, meaning that $J_N^*(\vartheta_j) \geq J_{N,\infty}^*(\vartheta_j)$ as claimed by Conjecture 6.3.

Furthermore if we consider the last plot in Figure 6.5, i.e., $N = 10$, the highest value reached by this index along Σ_2 is not even $2 \cdot 10^{-4}$, showing that from 10 switches on, we do not obtain significant reductions of the cost value.

When this happens the condition of tables convergence is reached. In other words for this particular problem all tables, from $N = 10$ on, are the same, i.e.,

$$\mathcal{C}_{10}^i \equiv \mathcal{C}_{11}^i \equiv \mathcal{C}_{12}^i \equiv \ldots.$$

This important result is general. In the next sections it will be formally proved and it will permit us to define the table $\mathcal{C}_\infty^i$, i.e., the unique tables that must be used, in each location $i = 1, 2, 3$, when an infinite number of switches are available.

[2] In the next sections we will see that this is a key comparison between the costs.

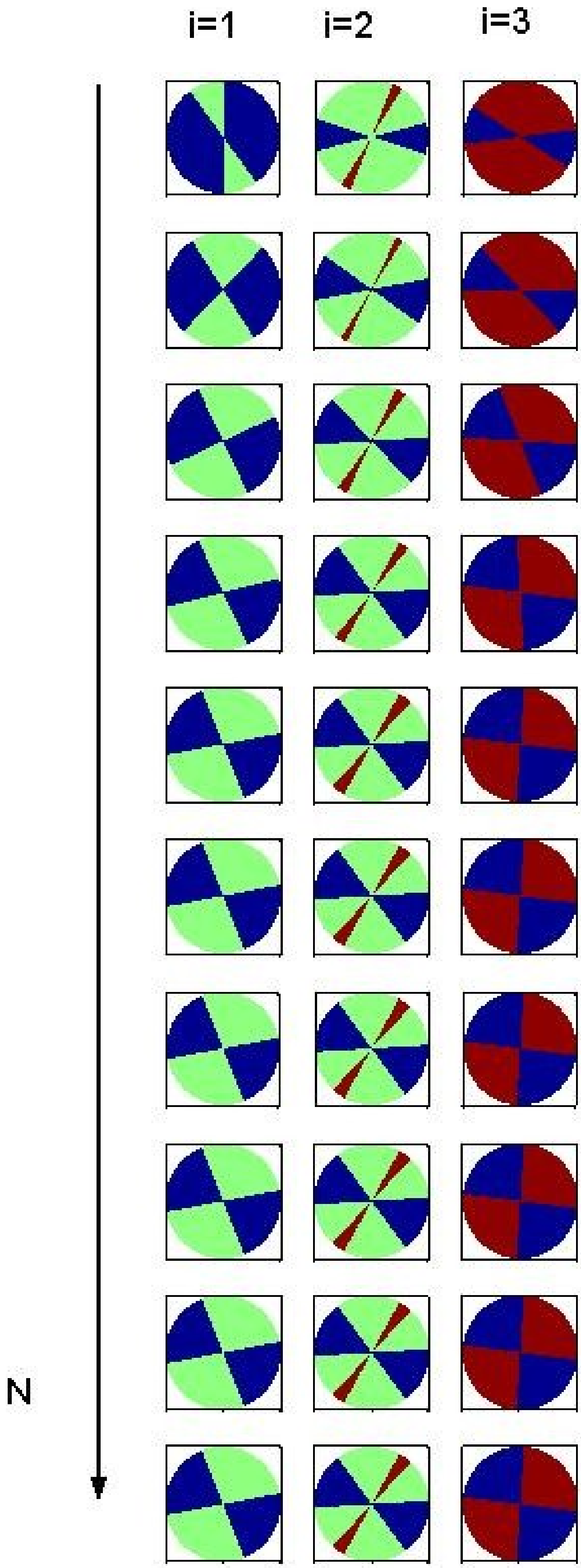


Fig. 6.3. *The 30 tables constructed with the STP for the example described in Section 6.4.1. From left to right location* $1, 2, 3$ *and from top to bottom the tables obtained per increasing* N *until* $N = 10$*. The color mapping (Figure 6.2) is: blue-*1*, green-*2*, red-*3*.*

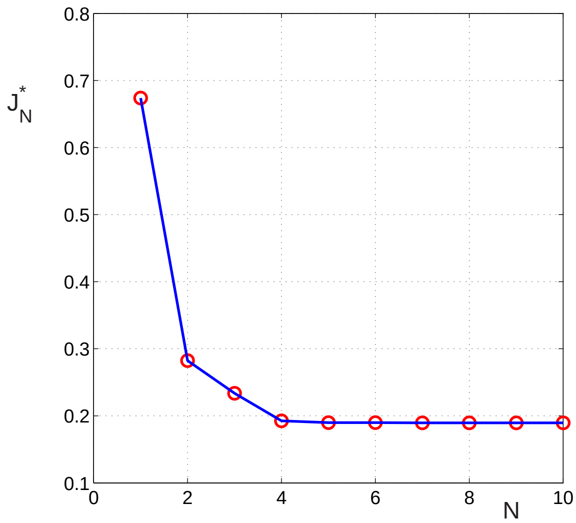


Fig. 6.4. *Asymptotic behavior of the optimal cost as the number of available switches increases for the example described in Section 6.4.1.*

We report in Figure 6.6 the tables $\mathcal{C}_{10}^i \equiv \mathcal{C}_{\infty}^i$, $i = 1, 2, 3$, i.e., the bottom row of Figure 6.3.

For completeness we show in Figure 6.7 the plot of an evolution from the point

$$\left(\boldsymbol{x}_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, i_0 = 2 \right).$$

The optimal variables and cost are:

$$\begin{aligned}
&\mathcal{T}^* = \{0.15, 0.34, 0.26, 0.25, 0.36, 0.26, 0.25, 0.35, 0.28, 0.37, \ldots\} \\
&\mathcal{I}^* = \{2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, \ldots\} \\
&J_{10}^* = 0.1896.
\end{aligned}$$

6.5 An infinite number of switches

In this section we discuss how, under appropriate assumptions, the above conjectures are proved, thus allowing us to efficiently extend the STP to the case of $N = \infty$.

Consider an OP$_\infty(S)$ of the form (6.1) where

(i) there exists $i \in \mathcal{S}$, such that the linear dynamics $\boldsymbol{A}_i$ is stable;
(ii) for all $i \in \mathcal{S}$, $\boldsymbol{Q}_i \geq \boldsymbol{0}$.

In Chapter 7 we will even relax (i) and extend the procedure to the case where all dynamics of the switched systems are unstable.

6.5.1 Convergence of the cost

Let us state initially an obvious monotonicity result.

Property 6.1 (Monotonicity of the cost) *Let $N, N' \in \mathbb{N}$. If $N' > N$ and the switched system evolves along an optimal trajectory, then for some initial hybrid state $(\boldsymbol{x}_0, i_0)$,*

$$J_{N'}^*(\boldsymbol{x}_0, i_0) \leq J_N^*(\boldsymbol{x}_0, i_0) < +\infty.$$

$\square$

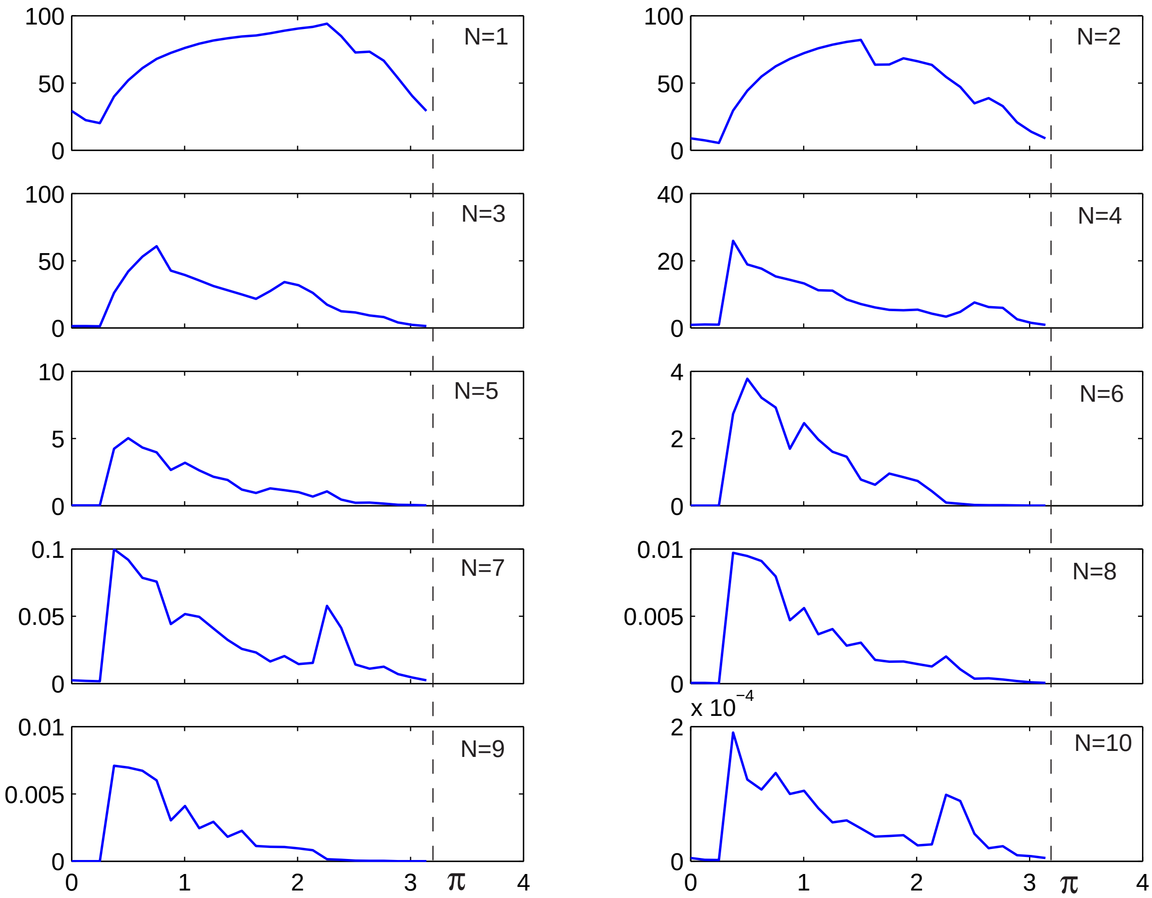


Fig. 6.5. *Example in Section 6.4.1. Percentage relative difference of the total cost of the evolution from 51* initial points *on Σ_2. In particular $J^*_{N,\infty}(\vartheta)$ is the cost of the evolution obtained using indefinitely the same tables $\mathcal{C}^i_N$, $i = 1, 2, 3$, while $J^*_N(\vartheta)$ is the cost of the evolution obtained using all tables $\mathcal{C}^i_j$, $j = N, N-1, \ldots, 1$.*

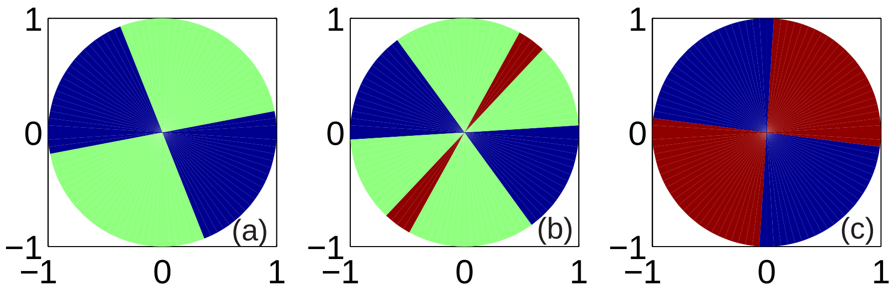


Fig. 6.6. *(a) Table $\mathcal{C}^1_{10}$, (b) Table $\mathcal{C}^2_{10}$, (c) Table $\mathcal{C}^3_{10}$, for the example described in Section 6.4.1. The color mapping (Figure 6.2) is: blue-1, green-2, red-3.*

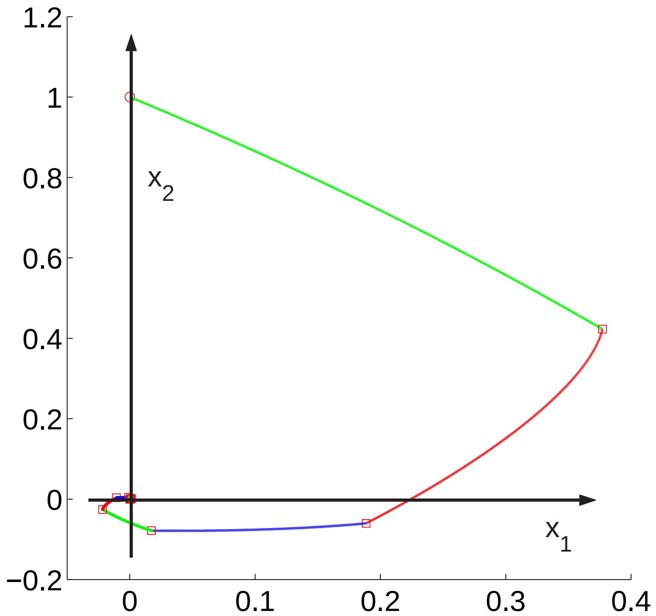


Fig. 6.7. *Plot of the hybrid evolution of the automaton described in Section 6.4.1 from the initial point* $(\boldsymbol{x}_0 = [0, 1]', i_0 = 2)$ *and performing at most 10 switches governed by the tables obtained from the STP of Section 4.3.*

Proof. We first observe that by Assumption (i), there exists a location i_0 such that $J_N^*(\boldsymbol{x}_0, i_0)$ is finite for any $N \geq 1$.

To prove the first inequality we observe that the same evolution that generates $J_N^*(\boldsymbol{x}_0, i_0)$ is also admissible for (6.2) when a larger value N' of switches is allowed. ■

An immediate consequence of Property 6.1 is the following proposition.

Proposition 6.1 (Convergence of the cost) *For all initial state* $(\boldsymbol{x}_0, i_0)$*,* $\boldsymbol{x}_0 \neq \boldsymbol{0}$*, and for all* $\varepsilon' > 0$*,* $\exists\ \bar{N} = \bar{N}(\boldsymbol{x}_0, i_0)$ *such that for all* $N > \bar{N}$*,*

$$J_N^*(\boldsymbol{x}_0, i_0) - J_{\bar{N}}^*(\boldsymbol{x}_0, i_0) < \varepsilon.$$

□

Proof. We first observe that by Assumption (ii) $J_N^*(\boldsymbol{x}_0, i_0)$ is lower bounded by a strictly positive number. Then, the result trivially follows from the monotonicity property above and the fact that J_N^* is lower bounded, hence it is a Cauchy sequence. ■

In other words the proof of the proposition leans on the fact that the cost is decreasing with N (Property 6.1) and on the fact that it is obviously lower bounded by a strictly positive value.

This can only be *iff* the function J_N^* as an asymptotic behavior with N. Let us consider the example described in Section 6.4.1. We depicted in Figure 6.4

$$J_N^*(\boldsymbol{x}_0 = [0, 1]', i_0 = 2)$$

as a function of N, number of available switches.

Its asymptotical behavior for the given initial point requires no further comments.

The reader can also refer to the example described in Section 4.7, and in particular Figure 4.21, where the property was analyzed in the case of the servomechanism model described in Section 4.5.

From the proposition above it is clear that the cost, *given a particular initial point* $(\boldsymbol{x}_0, i_0)$, converges to some finite $\bar{N}(\boldsymbol{x}_0, i_0)$.

One may argue that the dependency of $\bar{N}(\boldsymbol{x}_0, i_0)$ on the initial point might be such that

$$\lim_{\|\boldsymbol{x}_0\| \to +\infty} \bar{N}(\boldsymbol{x}_0, i_0) = +\infty.$$

If this was the case the results above would be useless. In fact it would not be possible to affirm that a unique finite value of $\bar{N}$ can be found. However by the homogeneity property of the cost function, it is easy to show that this is not true, and indeed the Proposition 6.1 can be extended to the normalized values of the costs on Σ_n.

We state formally this important result. We omit here, to avoid a cumbersome notation, the subscript 0 of the initial state, thus $(\boldsymbol{x}_0, i_0) = (\boldsymbol{x}, i)$. We show that, independently from the initial state, a relative tolerance ε on the cost can be found.

Proposition 6.2 (Normalized convergence of the cost) *For any initial state* $(\boldsymbol{x}, i)$, $\boldsymbol{x} \neq \mathbf{0}$*, and for all* $\varepsilon > 0,\ \exists\, \bar{N}$ *such that for all* $N > \bar{N}$*,*

$$\frac{J^*_N(\boldsymbol{x}, i) - J^*_{\bar{N}}(\boldsymbol{x}, i)}{J^*_N(\boldsymbol{x}, i)} < \varepsilon.$$

□

Proof. Since all switching costs are null, the optimal residual costs are 2-homogeneous functions (see Definition 4.2) of $\boldsymbol{x}$.

Thus if $\boldsymbol{x} = \lambda \boldsymbol{y}$, then

$$J^*_N(\boldsymbol{x}, i) = J^*_N(\lambda \boldsymbol{y}, i)$$

and

$$J^*_{\bar{N}}(\lambda \boldsymbol{y}, i) = \lambda^2 J^*_{\bar{N}}(\boldsymbol{y}, i).$$

Moreover, by Proposition 6.1 $\forall\, (\boldsymbol{y}, i)$ and $\forall\, \varepsilon' > 0$, $\exists\, \bar{N}(\boldsymbol{y}, i)$ such that

$$\forall\, N > \bar{N}(\boldsymbol{y}, i),$$
$$J^*_N(\boldsymbol{y}, i) - J^*_{\bar{N}}(\boldsymbol{y}, i) < \varepsilon'.$$

Hence if we define

$$\bar{N} = \max_{\substack{i \in \mathcal{S} \\ \boldsymbol{y} \in \Sigma_n}} \bar{N}(\boldsymbol{y}, i)$$

it holds that

$$\frac{J^*_N(\boldsymbol{x}, i) - J^*_{\bar{N}}(\boldsymbol{x}, i)}{J^*_N(\boldsymbol{x}, i)} = \frac{\lambda^2 [J^*_N(\boldsymbol{y}, i) - J^*_{\bar{N}}(\boldsymbol{y}, i)]}{\lambda^2 J^*_N(\boldsymbol{y}, i)} \leq \frac{\varepsilon'}{\min\limits_{\boldsymbol{y} \in \Sigma_n} J^*_N(\boldsymbol{y}, i)} = \varepsilon.$$

■

According to the above result, one may use a given relative tolerance ε to approximate two cost values, i.e.,

$$\frac{J^*_N(\boldsymbol{x}, i) - J^*_{N'}(\boldsymbol{x}, i)}{J^*_N(\boldsymbol{x}, i)} < \varepsilon \qquad \Longrightarrow \qquad J^*_N(\boldsymbol{x}, i) \cong J^*_{N'}(\boldsymbol{x}, i).$$

6.5.2 Convergence of the switching tables

Finally we can prove the main result of this chapter. All tables, computed with the STP described in Section 4.3, converge to the same one (for each location) for increasing values of N.

We keep omitting the subscript 0 in $(\boldsymbol{x}_0, i_0)$, thus $(\boldsymbol{x}_0, i_0) = (\boldsymbol{x}, i)$.

Theorem 6.1 *Given a fixed relative tolerance ε, if $\bar{N}$ is chosen as in Proposition 6.2 then for all $N > \bar{N} + 1$ it holds that $\mathcal{C}_N^i \equiv \mathcal{C}_{\bar{N}+1}^i$.* □

Proof. By definition $J_k^*(\boldsymbol{x}, i) = T_k^*(\boldsymbol{x}, i)$ for all $k \geq 1$, hence from equations (6.4) and (6.5) it follows that

$$\begin{aligned} &J_N^*(\boldsymbol{x}, i) = \\ &= \min_{\substack{j \in succ(i) \cup \{i\} \\ \varrho \geq 0}} \left\{ \boldsymbol{x}'\bar{\boldsymbol{Q}}_i(\varrho)\boldsymbol{x} + \boldsymbol{x}'\bar{\boldsymbol{A}}_i'(\varrho)\bar{\boldsymbol{Q}}_j(\delta_{min}(j))\bar{\boldsymbol{A}}_i(\varrho)\boldsymbol{x} + J_{N-1}^*(\boldsymbol{z}, j) \right\} \end{aligned}$$

where $\boldsymbol{z} = \bar{\boldsymbol{A}}_j(\delta_{min}(j))\bar{\boldsymbol{A}}_i(\varrho)\boldsymbol{x}$.

Now, being by assumption $N - 1 > \bar{N}$, by virtue of Proposition 6.2 we may approximate

$$J_{N-1}^*(\boldsymbol{z}, j) \cong J_{\bar{N}}^*(\boldsymbol{z}, j)$$

thus

$$\begin{aligned} &J_N^*(\boldsymbol{x}, i) \cong \\ &\cong \min_{\substack{j \in succ(i) \cup \{i\} \\ \varrho \geq 0}} \left\{ \boldsymbol{x}'\bar{\boldsymbol{Q}}_i(\varrho)\boldsymbol{x} + \boldsymbol{x}\bar{\boldsymbol{A}}_i'(\varrho)\bar{\boldsymbol{Q}}_j(\delta_{min}(j))\bar{\boldsymbol{A}}_i(\varrho)\boldsymbol{x} + J_{\bar{N}}^*(\boldsymbol{z}, j) \right\} = \\ &= J_{\bar{N}+1}^*(\boldsymbol{x}, i). \end{aligned}$$

Therefore, the optimal arguments (ϱ^*, j^*) used to compute $\mathcal{C}_N^i$ and $\mathcal{C}_{\bar{N}+1}^i$ are the same. ■

The above result allows one to compute with a finite procedure the optimal tables for a switching law when N goes to infinity.

In such a case, in fact, it holds that

$$\mathcal{C}_\infty^i \equiv \lim_{N \to \infty} \mathcal{C}_N^i \equiv \mathcal{C}_{\bar{N}+1}^i.$$

Hence, we only need to use the tables $\mathcal{C}_\infty^i$, $i \in \mathcal{S}$ for all switches.

We recall that under the assumptions (i) and (ii), the system, optimally controlled with an infinite number of switches, is stable as proved in [50].

6.5.3 A convergence criterion

We have proved in Proposition 6.2 that there exist a finite value of number of switches N such that the tables converge.

It is not clear yet how this value can be found analytically. We know in fact that a value of N exists and it is finite, but we will never be sure, in principle, that the convergence is reached if we do not consider exhaustively all possible values of N.

Since this is impossible in a practical implementation, then our approach consists in constructing tables until a convergence criterion is met.

In the special case where all the matrices of the switched system are stable and $\delta_{\min}(i) \neq 0$ for some i, a criterion may be obtained by simple considerations on the slowest decay time of each dynamics.

In fact it is reasonable to observe that the convergence rate [109] of the switched system (if any) is certainly higher then the slowest mode of the set

$$\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_s\}.$$

We can prove a theorem that establishes an upper bound on the value of $\bar{N}$. Let us first give the following definition:

Definition 6.1 (Slowest decay time) *Consider a switched system S composed of only Hurwitz dynamics. Consider the absolute real part of the slowest of slowest mode of each dynamics,*

$$\nu = \min_{i \in \mathcal{S}} \min_{j=1,\ldots,n} Re(|\lambda_{i,j}|).$$

We define the number $T \triangleq \dfrac{5}{\nu}$ as the slowest decay time *of the given switched system.* ■

Observe that this definition is an extension of the *time constant* for classic linear system. The factor 5 in the definition above is the number of time constants that should be taken in order to obtain a decay of $\boldsymbol{x}$ from any initial state, lower than 1%.

Theorem 6.2 (Upper bound of $\bar{N}$) *Consider a switched system S composed of only Hurwitz dynamics and $\delta_{\min}(i) \neq 0$ for all $i \in \mathcal{S}$. An* upper bound *of $\bar{N}$ is*

$$\bar{N} = \left\lceil \frac{T}{\delta_{\min}} \right\rceil,$$

where $\delta_{\min} = \min_{i \in \mathcal{S}} \delta_{\min}(i)$. □

Proof. If we perform $\bar{N}$ switches it means that the system spends at least the minimum permanence time in the visited dynamics. Hence we are sure that from all initial states we can obtain a decay of the norm of the state space of a factor of *at least* 10^{-2}, in the worst case, evolving only in the slowest dynamics. This shows that with higher values of $N > \bar{N}$ the *normalized cost* of the evolution will improve with the order of 10^{-4}, which can be considered negligible for practical purposes. ■

Note that this criterion is in many cases too restrictive. In the example described in Section 6.4.1 it holds

$$\nu = 4, \ \delta_{\min} = 0.1 \ \Rightarrow \ \bar{N} > 200,$$

which is a very high upper bound compared to $N = 10$, where we start to observe convergence experimentally.

6.6 Computational complexity

The computational complexity of this extension is the same of the STP. In fact, from an implementation point of view, we apply the same method recursively until we meet a convergence criterion.

The interested reader can refer to Section 4.4.2.

6.7 Application: case study

In this section we describe a case study, the design of a semiactive suspension system, that motivated the extensions of the STP described above.

6.7.1 Framework on suspension systems and design

A semiactive suspension [51, 54, 70, 98] consists of a spring and a damper where the value of the *damper coefficient* f[3] can be controlled and updated.

In some types of suspensions, the active ones, it may also be possible to control the elastic constant λ_s of the spring. This case is considered here only as a target of the semiactive one.

A semiactive suspension is a valid trade-off solution because it can be easily realized at a lower cost than that of a fully active one [35, 56].

Note, however, that a semiactive system clearly lacks other important secondary advantages of the fully active one, like the ability to resist downward static forces (due to loads) and to control the altitude of the vehicle.

The optimal control technique known as LQR [87] is probably the simplest way to design an active law for suspension systems and such an idea has been initially proposed by Thompson [111]. In such a case the objective is that of minimizing a given performance index, that consists of a quadratic cost.

The control input is the value $u(t)$ of the force generated by the suspension. The optimal law takes the form of a state feedback law with constant gains, i.e.,

$$u(t) = -\boldsymbol{K}\boldsymbol{x}(t).$$

We can model a semiactive suspension system as a switched system, if we assume that the damping coefficient $f(t)$ may take values within a finite set

$$\mathcal{F} = \{f_1, f_2, \cdots, f_s\}$$

where

$$f_1 < f_2 < \ldots < f_s.$$

In the resulting model a different location corresponds to each value of f. The control input is now the discrete switch: we change the value of f, switching within locations, with the objective of minimizing a given performance index, that consists of a quadratic cost.

The optimal law takes the form of a state feedback law: in fact it has been shown that the optimal switch can be triggered by looking at the current hybrid state $(\boldsymbol{x}, i)$.

As in [48] we assume a time is required to update the damping coefficient. This is modelled by the introduction of a minimum permanence time $\delta_{\min}$.

Furthermore, within this time it is only possible to pass to adjacent values of f, i.e., if $f(t) = f_i$ then

$$f(t + \delta_{\min}) \in \{f_{i-1}, f_i, f_{i+1}\}.$$

The results of some numerical simulations show that the proposed semiactive suspension system always provides a good approximation of a fully active suspension system, while producing significant improvements wrt purely passive suspensions.

[3] Damper coefficient is a technical term. A common term in applied science is *viscous coefficient*, i.e., the proportional factor between Force and Velocity in viscous media.

6.7.2 Dynamical models of the suspension system

We consider a quarter car suspension system and derive two different dynamical models. The first one is a 2-DOF fourth order dynamical model that takes into account the dynamics of the tire. The second one is a 1-DOF second order dynamical model that neglects the effect of the tire.

While the second order model allows one to study the filtering properties of the suspension in terms of passenger comfort, it does not describe the interaction of the tire with the suspended mass and the ground, and thus it cannot be used to evaluate other important features such as road holding.

From an benchmark point of view, however, the reduced order model is extremely useful, because it is possible to give a geometrical representation of the optimal switching regions, thus providing a more intuitive explanation of the proposed approach. This is the main reason that led us to consider both models.

The fourth order dynamical model

Let us now consider the completely active suspension system of a quarter car with two degrees of freedom schematized in Figure 6.8.a.

We used the following notation:

— M_w is the equivalent unsprung mass consisting of the wheel and its moving parts;
— M_s is the sprung mass, i.e., the part of the whole body mass and the load mass pertaining to only one wheel;
— λ_t is the elastic constant of the tire, whose damping characteristics have been neglected. Note that this is in line with almost all researchers who have investigated synthesis of active suspensions for motor vehicles as the tire damping is minimal;
— λ_s is the elastic constant of the spring;
— $x_1(t)$ is the deformation of the suspension wrt the static equilibrium configuration, taken as positive when elongating;
— $x_2(t)$ is the vertical absolute velocity of the sprung mass M_s;
— $x_3(t)$ is the deformation of the tire wrt the static equilibrium configuration, taken as positive when elongating;
— $x_4(t)$ is the vertical absolute velocity of the unsprung mass M_w;
— $u(t)$ is the control force produced by the actuator.

It is readily shown that the state variable mathematical model of the system under study is given by [35]

$$\dot{\boldsymbol{x}}(t) = \tilde{\boldsymbol{A}}\boldsymbol{x}(t) + \tilde{\boldsymbol{B}}u(t) \tag{6.7}$$

where

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \end{bmatrix}$$

is the state, and the constant matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ have the following structure:

$$\tilde{\boldsymbol{A}} = -\begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & \frac{\lambda_t}{M_w} & 0 \end{bmatrix}, \qquad \tilde{\boldsymbol{B}} = \begin{bmatrix} 0 \\ \frac{1}{M_s} \\ 0 \\ -\frac{1}{M_w} \end{bmatrix}.$$

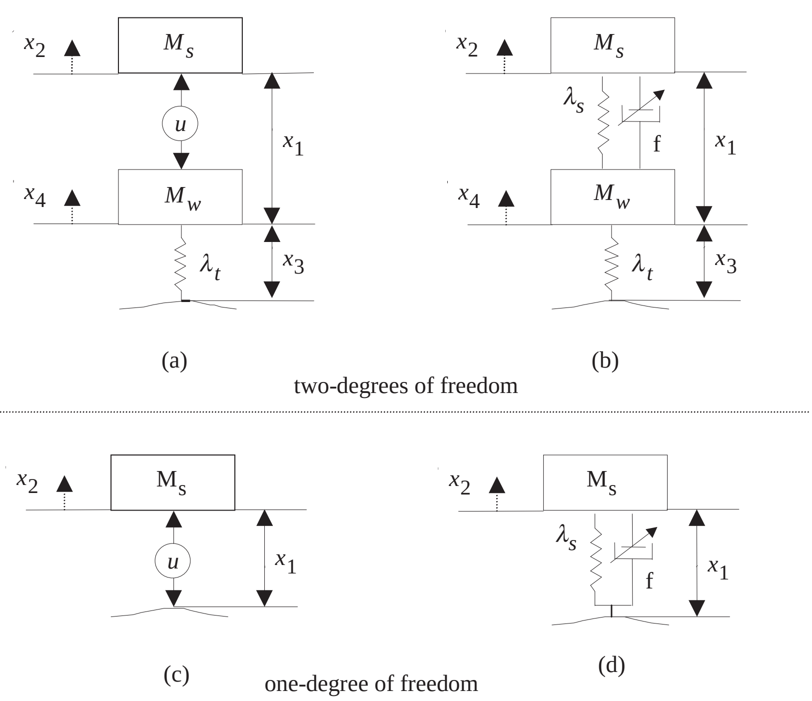


Fig. 6.8. *Scheme of the 2-DOF suspension: (a) active suspension; (b) semiactive suspension. Scheme of the 1-DOF suspension: (c) active suspension; (d) semiactive suspension.*

Now, let us consider Figure 6.8.b that represents a conventional semiactive suspension composed of a spring and a damper with adaptive characteristic coefficient $f = f(t)$.

The effect of this suspension is equivalent to that of a control force

$$u_s(t) = -\left[\lambda_s \ f(t) \ 0 \ -f(t)\right] \boldsymbol{x}(t). \tag{6.8}$$

Note that, as f may vary, $u_s(t)$ is both a function of $f(t)$ and of $\boldsymbol{x}(t)$. It is immediate to verify that the state variable mathematical model of the semiactive suspension is still given by equation (6.7) where $u(t)$ is replaced by $u_s(t)$.

Therefore, in such a case the system dynamics is regulated by the following state equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -\frac{\lambda_s}{M_s} & -\frac{f(t)}{M_s} & 0 & \frac{f(t)}{M_s} \\ 0 & 0 & 0 & 1 \\ \frac{\lambda_s}{M_w} & \frac{f(t)}{M_w} & -\frac{\lambda_t}{M_w} & -\frac{f(t)}{M_w} \end{bmatrix} \boldsymbol{x}(t). \tag{6.9}$$

The second order dynamical model

If the dynamics of the tire is completely neglected, the suspension system of a quarter car can be schematized as shown in Figures 6.8.c and d. More precisely, Figure c provides the scheme of a completely active suspension system, while Figure d provides the scheme of a semiactive suspension system, where the physical meaning of all variables is the same as in the 2-DOF case.

The state variable mathematical model of the active system is still given by a linear DE of the form (6.7), where the state is

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix},$$

and the constant matrices $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ have the following structure:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad \tilde{\boldsymbol{B}} = \begin{bmatrix} 0 \\ \frac{1}{M_s} \end{bmatrix}.$$

The effect of the semiactive suspension is equivalent to that of a control force

$$u_s(t) = -\begin{bmatrix} \lambda_s & f(t) \end{bmatrix} \boldsymbol{x}(t). \tag{6.10}$$

Thus, the system dynamics of a semiactive suspension is regulated by the following state equation:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{\lambda_s}{M_s} & -\frac{f(t)}{M_s} \end{bmatrix} \boldsymbol{x}(t). \tag{6.11}$$

6.7.3 Semiactive suspension design

Now, let us discuss in detail how the proposed methodology can be successfully used to design a semiactive suspension system.

As already said in Section 6.7.1, we assume that the value of the damping coefficient f may take values within a finite set

$$\mathcal{F} = \{f_1, f_2, \ldots, f_s\}$$

where

$$f_1 < f_2 < \ldots, f_s.$$

We select the value of f in $\mathcal{F}$ so as to minimize a given performance index, consisting of a quadratic cost depending on the time evolution.

Moreover, we assume that:

(A1) the state is measurable;

(A2) whenever f is updated, its value remains the same within a given time interval $\delta_{\min}$, that does not depend on the current value of f;

(A3) if at time t the damping coefficient is updated to

$$f(t) = f_i \in \mathcal{F},$$

then at time $t + \delta_{\min}$ the value of f may either remain the same or it may switch to an "adjacent" value, namely,

$$f(t + \delta_{\min}) \in \begin{cases} \{f_i, f_{i+1}\} & i = 1 \\ \{f_{i-1}, f_i, f_{i+1}\} & i = 2, \cdots, s-1 \\ \{f_{i-1}, f_i\} & i = s \end{cases} \tag{6.12}$$

Note that assumption (A2) enables us to take into account the fact that the damping coefficient f cannot be updated at an arbitrarily high frequency. Clearly, the amplitude of the time interval $\delta_{\min}$ depends on the particular physical damper.

As an example, in the case of a solenoid valve damper [48, 99], under the above assumption (A2) an admissible value is $\delta_{\min} = 0.007$ [48].

If the assumption (A3) is removed, and we assume that the value of f may arbitrarily change from any value to any other one, a larger $\delta_{\min}$ should be considered, e.g., $\delta_{\min} = 0.03$ [48].

Under the assumptions (A1) to (A3), the considered optimal control problem can be written as in (6.2).

The matrices $\boldsymbol{A}_{i(t)}$ are uniquely defined given the value of f according to equations (6.9) or (6.11), depending on the considered dynamical model.

More precisely, to each value of f in $\mathcal{F}$ it corresponds a matrix $\boldsymbol{A}(f(t))$ that specifies the discrete state (location) of the switched system.

Note that we consider a particular case where the minimum permanence time in the discrete locations is the same for all locations.

Moreover, from the assumption (A3), the oriented graph of the switches system that shows all the arcs, has the structure of a *birth-death process* [47] and is shown in Figure 6.9.

Fig. 6.9. *The oriented graph of the switched system that models the semiactive suspension described in Section 6.7.3.*

In the following we present the results of some numerical simulations carried out on both the second order and the fourth order dynamical system.

In particular, we first assume that a finite number N of switches is available, then we allow the system to perform an infinite number of switches.

6.7.4 Application example

The proposed procedure has been applied to the quarter car suspension shown in Figure 6.8, with values of the parameters taken from [51], and reported in Table 6.1.

Table 6.1. *Model parameters of the suspension system considered in Section 6.7.3.*

Symbol	Value (IS)	Physical meaning
M_s	288.90	mass of the quarter car
M_w	28.58	mass of the wheel
λ_s	14345	elastic coefficient of the spring
λ_t	155900	elastic coefficient of the tire

The damping coefficient f[4] may take values within the finite set

$$\mathcal{F} = \{800, 1500, 2300, 3000\}$$

while the minimum permanence time is taken $\delta_{\min} = 0.007$. The oriented graph of the switched system is depicted in Figure 6.10.

[4]In the IS the damper coefficient is measured in Ns/m.

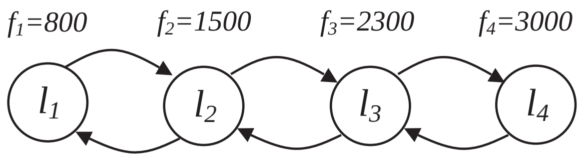


Fig. 6.10. *The oriented graph of the switched system that models the semiactive suspension described in Section 6.7.3 with the corresponding numerical values.*

6.7.5 Simulations on the second order model

We first present the results of some numerical simulations carried out on the second order dynamical model of the suspension system.

A different weighting matrix is associated to each discrete location, or equivalently to each value of f. In particular, we assume that

$$\boldsymbol{Q}_{i(t)} = \boldsymbol{Q}(f(t)) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + 0.8 \cdot 10^{-9} \cdot \begin{bmatrix} \lambda_s \\ f(t) \end{bmatrix} [\lambda_s, f(t)].$$

In such a way, by virtue of equation (6.10), we can perform a significant comparison, in terms of performance index, among the proposed semiactive suspension and an active suspension system, considered as a target.

The purely active suspension can be obtained by solving an LQR problem where

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \ \boldsymbol{R} = 0.8 \cdot 10^{-9}.$$

Note that the numerical values of the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are the same as in [51].

Simulation 1: $\boldsymbol{N = 6}$

We first assume that a finite number $N = 6$ of switches is available. We evaluate offline the $N \times s$ switching tables. A state space discretization of $r = 100$ points along Σ_2 and a minimum local search over three time constants were considered sufficiently fine.

We assume that the initial state is

$$\boldsymbol{x}_0 = \begin{bmatrix} 0.1 \\ 0 \end{bmatrix}, \ i_0 = 1.$$

The state trajectory that minimizes the performance index is depicted in Figure 6.11, where the circle indicates the initial state and the squares indicate the values of the state at the switching times. We found out

$$\begin{aligned} &\mathcal{T}^* = \{0.096, 0.1370, 0.222, 0.473, 0.482, 0.646\} \\ &\mathcal{I}^* = \{1, 2, 3, 4, 3, 2, 3\} \\ &J_6^* = 1.419 \cdot 10^{-3}. \end{aligned}$$

Figure 6.12 shows, among the 24 tables constructed, only the 6 ones used by the controller during the evolution of the system.

The system initially evolves in location 1. When the minimum permanence time $\delta_{\min}$ has elapsed, the controller must keep checking the color in table $\mathcal{C}_6^1$ (see Figure 6.12) corresponding to the current state $(\boldsymbol{x}, 1)$.

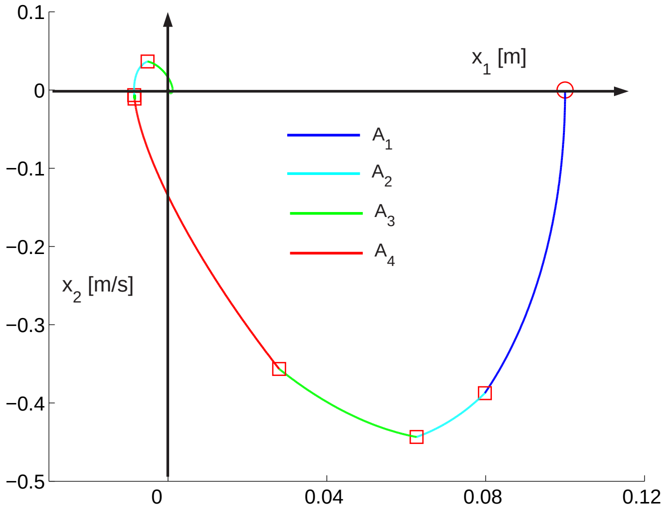


Fig. 6.11. *The results of Simulation 1: the state trajectory.*

According to this color the controller decides whether to remain in location 1 or to switch to the adjacent location 2. In this case, no switch occurs until a time $\tau_1 = 0.096$ has elapsed, when the continuous state reaches the cyan area relative to location 2. Now the controller will wait for the minimum permanence time and then consider table $\mathcal{C}_5^2$. The same procedure is repeated until all the available switches are performed.

Note that, given the structure of the automaton, while the switching tables associated to discrete locations 2 and 3 may have up to 3 colors, the tables associated to locations 1 and 4 may have at most two different colors.

To better appreciate the performance of the proposed semiactive suspension it is necessary to look at the time evolution of the sprung mass displacement. This curve is reported in Figure 6.13.a where we can also visualize the evolution of the fully active suspension considered as a target, and that of a completely passive suspension obtained using a value of $f = 1918$ Ns/m [34].

In Figure 6.13.b we have reported the different values of the damping coefficient f during the simulation.

In Table 6.7.5 we compare the values of the quadratic performance index obtained using the active suspension (considered as a target), the semiactive suspension in the case of $N = 6$ ($i_0 = 1$ in all cases), and the passive suspension system obtained using $f = 1918$, chosen as in [34].

The results of Table 6.7.5 enable us to conclude that the proposed semiactive suspension exhibits an intermediate behavior between the passive suspension and the considered active one, even if a small number of switches is allowed.

Simulation 2: $N = \infty$

As already discussed in Section 6.5, for a sufficiently large value of N, the tables relative to the first switches always converge to the same one, only depending on the discrete location $l \in \mathcal{L}$.

As an example, assume $N = 8$ and consider the discrete location 3. The tables relative to the first 6 switches, namely $\mathcal{C}_k^3$, $k = 8, 7, \ldots, 3$, are reported in Figure 6.14.

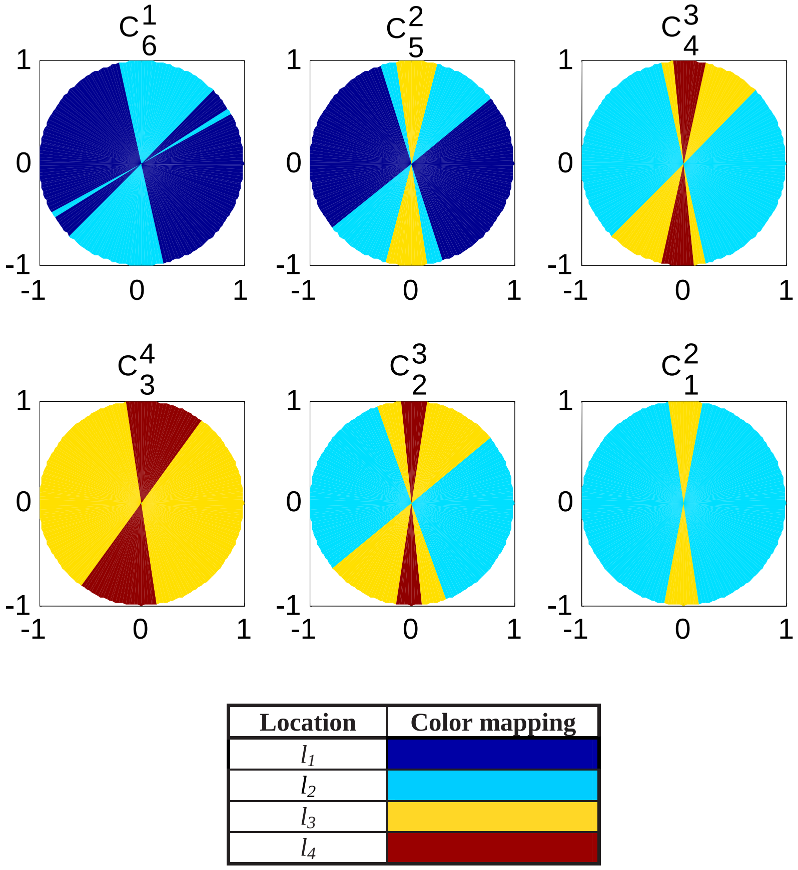


Fig. 6.12. *Tables used by the controller to compute the state evolution in Figure 6.11.*

Table 6.2. *Different values of the performance index in the case of some numerical simulations carried out on the second order model.*

x_0	semiactive ($N=6$)	semiactive ($N=\infty$)	active	passive
$[0.100\ \ 0.000]'$	$1.419 \cdot 10^{-3}$	$1.419 \cdot 10^{-3}$	$1.278 \cdot 10^{-3}$	$1.546 \cdot 10^{-3}$
$[0.045\ \ 0.090]'$	$3.960 \cdot 10^{-4}$	$3.959 \cdot 10^{-4}$	$3.294 \cdot 10^{-4}$	$4.189 \cdot 10^{-4}$
$[-0.015\ \ 0.100]'$	$1.493 \cdot 10^{-5}$	$1.492 \cdot 10^{-5}$	$1.437 \cdot 10^{-5}$	$1.905 \cdot 10^{-5}$
$[-0.057\ \ 0.080]'$	$3.719 \cdot 10^{-4}$	$3.717 \cdot 10^{-4}$	$3.506 \cdot 10^{-4}$	$4.114 \cdot 10^{-4}$

We may observe that, as the number of available switches increases, i.e., k goes from 3 to 8, the tables converge. In particular, in this case the tables relative to the first two switches, namely $\mathcal{C}_8^3$ and $\mathcal{C}_7^3$, are the same.

Now, if we consider a larger value of N, i.e., $N = 9$ (10), and look at the tables relative to location 3, we may observe that $\mathcal{C}_9^3$ ($\mathcal{C}_{10}^3$) tables coincide with $\mathcal{C}_8^3$ and $\mathcal{C}_7^3$. Using the notation introduced in Section 6.5, we denote these tables as $\mathcal{C}_\infty^3$.

Analogous considerations may be repeated for all the other discrete locations.

Now, let us consider the $\mathrm{OP}_\infty(S)$ (6.1) with no bound on the maximum number of available switches.

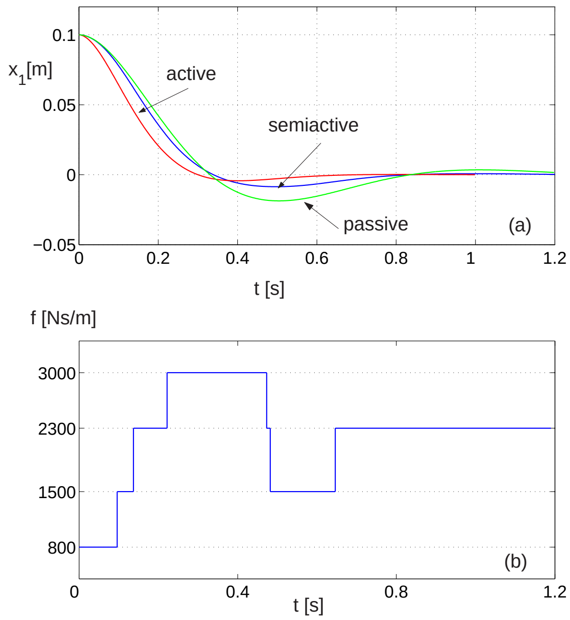


Fig. 6.13. *The results of Simulation 1: (a) the time evolution of the sprung mass displacement; (b) the different values of f used by the semiactive suspension.*

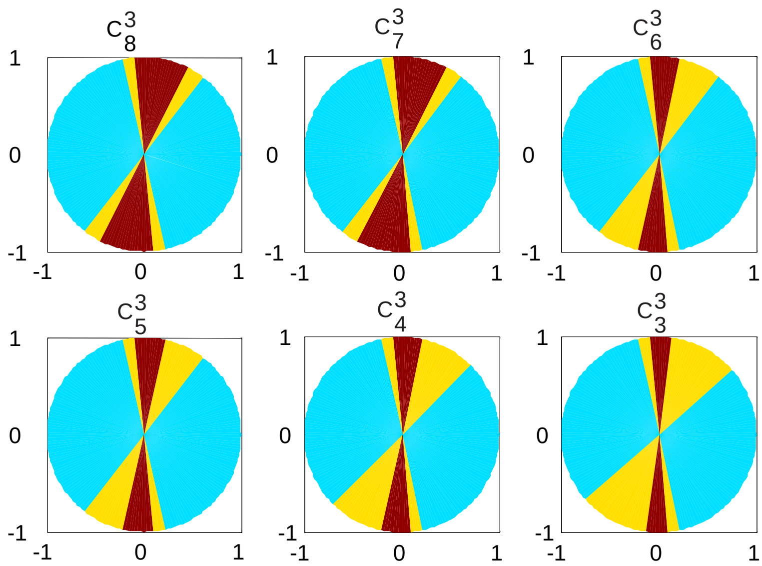


Fig. 6.14. *The first 6 switching tables for location 3 and $N = 8$. Color mapping is in Figure 6.15.*

Location	Color mapping
l_1	
l_2	
l_3	
l_4	

Fig. 6.15. *Color mapping of the locations* $1, 2, 3, 4$ *of the semiactive suspension system design described in Section 6.7.3.*

By virtue of the above convergence properties, this problem can be solved by using only the tables $\mathcal{C}_\infty^i$, for $i \in \mathcal{S}$, as described in Section 6.5.

We report these tables in Figure 6.16.

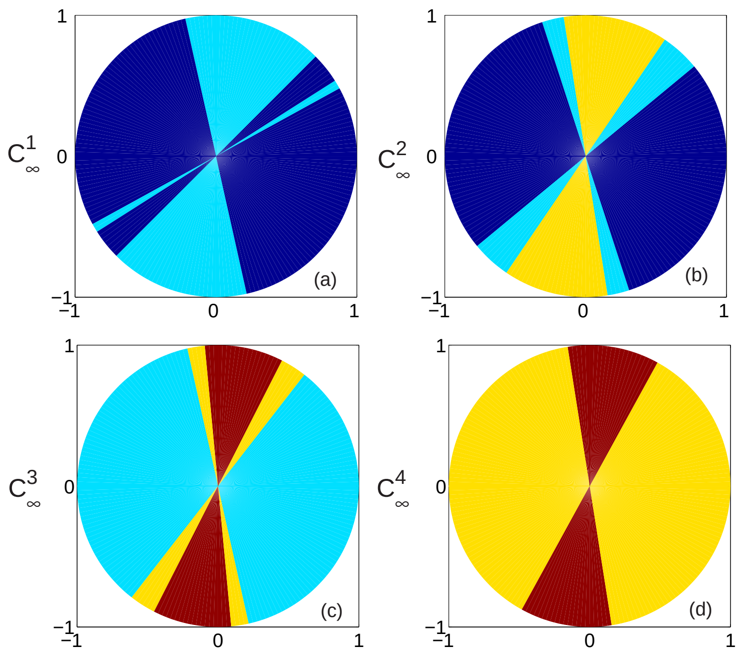


Fig. 6.16. *Convergence tables* $\mathcal{C}_\infty^1, \ldots, \mathcal{C}_\infty^4$ *(a), (b), (c), (d), respectively, for the semiactive suspension design with an infinite number of switches. Color mapping is in Figure 6.15.*

Assume that the initial state is still equal to $\boldsymbol{x}_0 = [0.1\ 0]'$ and $i_0 = 1$.

The state trajectory that minimizes the performance index is reported in Figure 6.17 where the circle indicates the initial state and the squares indicate the values of the state at the switching times.

It can be easily observed that this trajectory is practically coincident with that in Figure 6.11.

This clearly occurs because after the first 6 switches, the system has practically reached the origin. As a consequence, the optimal value of the performance index J^* is practically the same, as it can be read in Table 6.7.5.

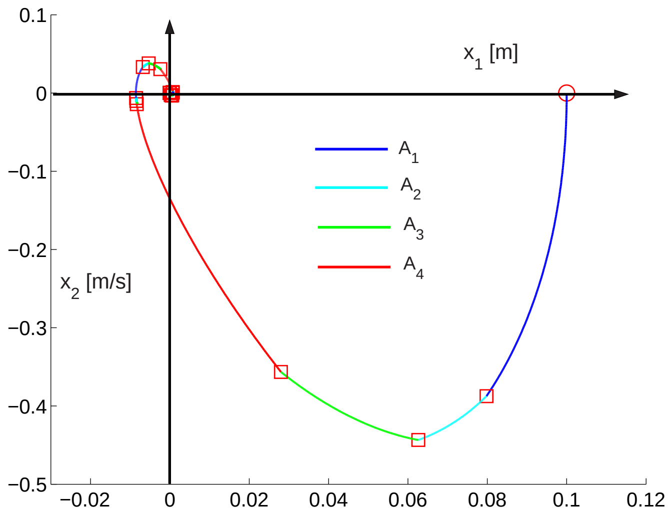


Fig. 6.17. *The results of Simulation 2: the state trajectory.*

In Figure 6.18.a we have reported the sprung mass displacement of the semiactive suspension together with that of the fully active suspension considered as a target, and that of a completely passive suspension [34].

In Figure 6.18.b we can see the different values of the damping coefficient f during the numerical simulation.

Note that the periodicity of the switching sequence is a consequence of the particular example (second order system, rotating dynamics), but it is not a general result.

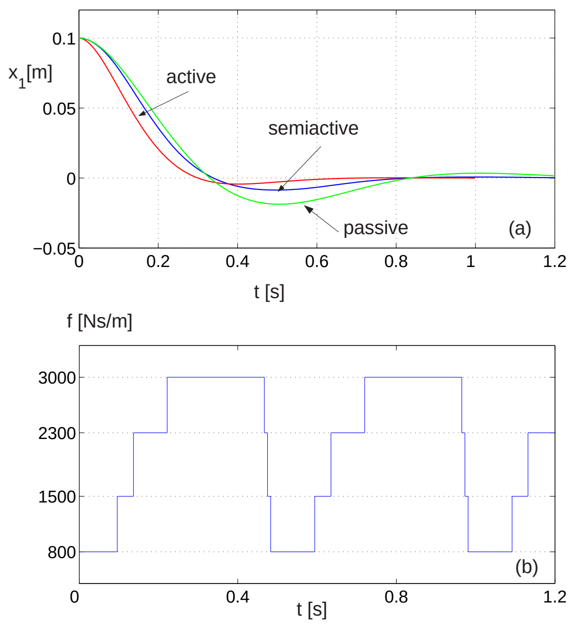


Fig. 6.18. *The results of Simulation 2: (a) the time evolution of the sprung mass displacement; (b) the different values of f used by the semiactive suspension.*

6.7.6 Simulations on the fourth order model

Now, let us present the results of some numerical simulations carried out on the fourth order dynamical model of the suspension system.

As in the previous case, a different weighting matrix is associated to each discrete location, or equivalently to each value of f. In particular, we assume that

$$\boldsymbol{Q}_{i(t)} = \boldsymbol{Q}(f(t)) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + 0.8 \cdot 10^{-9} \cdot \begin{bmatrix} \lambda_s \\ f(t) \\ 0 \\ -f(t) \end{bmatrix} \cdot [\lambda_s, f(t), 0, -f(t)].$$

In such a way, by virtue of equation (6.8), we can perform a significant comparison, in terms of performance index, among the proposed semiactive suspension and an active suspension system, considered as a target, and obtained by solving an LQR problem where $\boldsymbol{Q} = \text{diag}\{1,\ 0,\ 10,\ 0\}$ and $\boldsymbol{R} = 0.8 \cdot 10^{-9}$ [51].

We consider straightforward the most realistic case of $N = \infty$.

As already explained above, we first compute the $N \times s$ switching tables for a "sufficiently" large value of N until we observe that there exists a $k < N$ such that for all $i \in \mathcal{S}$,

$$\mathcal{C}_k^i = \mathcal{C}_{k+1}^i = \cdots = \mathcal{C}_N^i.$$

In this case we took $N = 6$ and we observed that the convergence occurs for $k = 5$. Thus, we can reasonably assume

$$\mathcal{C}_\infty^i = \mathcal{C}_6^i,\ i = 1, \ldots, 4.$$

These switching tables are not reported here because a significant graphical representation is not possible.

The STP applied for this $4 - th$ dimensional case was an interesting challenging problem from an implementation point of view. See the Appendices C.1 and C.2 for a brief description of the algorithm that allowed the numerical construction of the tables in $\mathbb{R}^4$.

The three angles ξ, φ, ϑ that describe Σ_4 in spherical coordinates (Appendix C.1) are appropriately sampled.

A trade-off value was found in $N_\xi = 15$, and it produces, with the criteria described in Appendix C.1, 8581 points.

Note that with a constant discretization, $N_\xi = 15$ would have produced 27000 points, without providing a denser information.

Moreover the criteria in Appendix C.2 was important in order to allow us to take such a small value of N_ξ.

Running in MATLAB environment on a pentium III 450 MHz the computational time per switch is about 60 hours. Note however that these burdensome calculations are performed offline.

Assume that the initial state is

$$\boldsymbol{x}_0 = \begin{bmatrix} 0.1 \\ 0 \\ 0.01 \\ 0 \end{bmatrix},\ i_0 = 1.$$

In Figures 6.19.a and b we have reported the sprung mass and the unsprung mass displacement of the semiactive suspension together with that of the fully active suspension considered as a target, and that of a completely passive suspension [34].

In particular, by looking at plot (a) that shows the most significant variable, we can conclude that the semiactive system guarantees better performance than the passive one.

In fact, in such a case, the behavior of the semiactive suspension system in terms of the sprung mass displacement, is quite similar to that obtained using the purely active system. Finally, in Figure 6.19.c we can see the different values of the damping coefficient f during the numerical simulation.

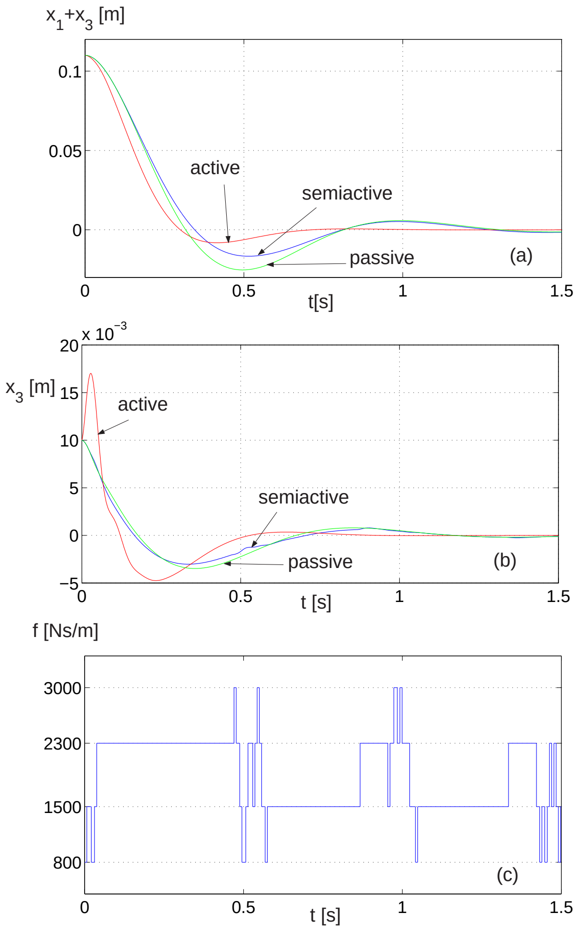


Fig. 6.19. *The results of the simulation carried out on the fourth order model: (a) the time evolution of the sprung mass displacement ($x_1 + x_3$); (b) the unsprung mass displacement x_3; (c) the different values of f used by the semiactive suspension.*

A comparison among the semiactive, the active and the passive suspension in terms of performance index is given in Table 6.3, for a small group of significant initial points.

We may conclude, as in the 1-DOF case, that the proposed semiactive suspension provides an intermediate performance between that of the passive suspension and that of the purely active one.

Table 6.3. *The results of the numerical simulations carried out on the fourth order model.*

$\boldsymbol{x}_0$	semiactive	active	passive
$\boldsymbol{x}_0 = [0.100\ \ 0\ \ 0.010\ \ 0]'$	$1.775 \cdot 10^{-3}$	$1.591 \cdot 10^{-3}$	$1.829 \cdot 10^{-3}$
$\boldsymbol{x}_0 = [-0.050\ \ 0.300\ \ -0.005\ \ 0.010]'$	$2.423 \cdot 10^{-4}$	$2.374 \cdot 10^{-4}$	$2.976 \cdot 10^{-4}$
$\boldsymbol{x}_0 = [0.050\ \ 0.300\ \ 0.005\ \ 0.010]'$	$1.011 \cdot 10^{-3}$	$8.200 \cdot 10^{-4}$	$1.052 \cdot 10^{-3}$
$\boldsymbol{x}_0 = [0.010\ \ -0.300\ \ 0.010\ \ 0.100]'$	$1.678 \cdot 10^{-4}$	$1.164 \cdot 10^{-4}$	$2.175 \cdot 10^{-4}$
$\boldsymbol{x}_0 = [0\ \ 0.400\ \ 0.010\ \ 0.300]'$	$3.513 \cdot 10^{-4}$	$3.109 \cdot 10^{-4}$	$4.312 \cdot 10^{-4}$
$\boldsymbol{x}_0 = [-0.080\ \ -0.100\ \ 0.012\ \ 0.400]'$	$1.144 \cdot 10^{-3}$	$8.903 \cdot 10^{-4}$	$1.151 \cdot 10^{-3}$

6.8 Conclusions

In this chapter the problem of infinite number of switches has been examined. We proved a convergence behavior of the switching tables under particular assumptions.

In particular we formally shown that the cost function, is a decreasing function of the number of switches, and that there exists a sufficiently great number $\bar{N}$, independent from the initial point, such that if the systems performs more than $\bar{N}$ switches the relative improvements on the cost are irrelevant.

Such result permitted us to demonstrate that the tables must converge, and moreover we provide a constructive way to design them.

Once these tables are constructed the controller is allowed to *indefinitely* use the last calculated table $\mathcal{C}^i_{\bar{N}}$, for an infinite number of switches.

This result, in junction with the STP was applied to a literature and industrial case study, i.e., the design of a particular semiactive suspension.

The possibility of performing an infinite number of switches and to design the optimal control law with a finite procedure, allowed the authors to explore possible relations with other important issues concerning the switched systems, in particular the stabilizability issue. This will be done in Chapter 7.

7

Infinite number of switches: optimal control and stability

7.1 Introduction

We have dealt in the previous chapters with the problem of designing a feedback control law for a class of switched system and a class of hybrid automaton. To this aim we have developed a recursive procedure, called STP, that under particular assumptions, provides a partition of the state space into time invariant switching regions.

The procedure was initially developed under the constraint that the number of allowed switches N is finite. Then we observed a convergence behavior of the switching tables with an increasing number of allowed switches, leading us to deal also with an infinite number of switches.

In the last chapter we proved this important aspect of the STP formally. This allows the construction of the feedback control law for hybrid automata that optimally drives the system to the origin performing an infinite number of switches.

In both cases (N finite and N infinite) we introduced the fundamental Assumption 4.1 that *basically* guarantees the existence of a switching sequence, finite or infinite, whose corresponding cost is finite.

In this chapter we use the STP to obtain a stabilizing switching sequence, that is also optimal.

Note that some switched systems composed of only unstable modes can be stabilized by appropriate switching surfaces (as a reference see for instance [17, 90] among many others) of conic shape. It is reasonable to assume that the quadratic LQR cost of these *asymptotically stable* solutions is finite.

These simple considerations suggest, for $N = \infty$, to relax the Assumption 4.1 and see if we can find a finite cost solution for a switched system composed of only unstable modes.

In [50] it is proved that if a switched system can be optimally controlled with a finite cost, then the closed loop system is *asymptotically stable*.

In such a framework the STP becomes a numerically viable approach to designing a stabilizing control law, which is indeed a significant issue in the context of the autonomous switched systems.

Moreover we prove that if the switched system is *exponentially stabilizable*, then the STP can always find an optimal control law with a finite cost that makes the closed loop system at least asymptotically stable.

The method of using the STP to provide a stabilizing switching sequence is based on the consideration outlined below.

Once all tables $\mathcal{C}_\infty^i$ (defined in Section 6.5.2) are constructed, it may happen that the region $\mathcal{R}_j$ associated to a given dynamics j never appears.

In this case, the optimal evolution for the given switched system is equivalent to the same switched system where the location j (that never appears) is removed.

This in particular, may allow us to compute an optimal control law for an unstable system introducing a dummy stable dynamics $\tilde{\boldsymbol{A}}$, provided that the corresponding region does not appear in the tables $\mathcal{C}_\infty^i$.

The result is significant. In fact, although there is a rich literature on stability *analysis* of hybrid systems, there are very few results on the *design* of stabilizing laws and they usually apply to restricted classes of systems or give only sufficient conditions.

7.2 The considered model

In this chapter we will derive the sufficient conditions for the existence of a stabilizing switching law for the class of switched system SA described in Section 3.3.1, Definition 3.9.

Briefly this is a particular case of S where all locations are featured by an autonomous linear dynamics, the set of edges is *complete*. This signifies that the oriented graph of the system is completely connected and that the set $succ(i) \equiv \mathcal{S} \setminus \{i\}$

Moreover we set $\forall\ i \neq j\ \in\ \mathcal{S},\ \boldsymbol{M}_{i,j} = \boldsymbol{I}_n$ (the state is continuous at the switch), and $\forall\ i\ \in\ \mathcal{S},\ \delta_{\min}(i) = 0$ (no minimum permanence time required in each location). We recall that the evolution is given by

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}, t) \triangleq \boldsymbol{A}_{i(t)}\boldsymbol{x}(t), \quad i \in \mathcal{S} \equiv \{1, \ldots, s\}, \tag{7.1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$, $i(t) \in \mathcal{S}$ is the current mode and represents a control variable, $\mathcal{S}$ is a finite set of integers, each one associated with a matrix $\boldsymbol{A}_i \in \mathbb{R}^{n\times n}$.

Moreover, $\forall\ i \in \mathcal{S}$, the dynamics $\boldsymbol{A}_i$ are non Hurwitz. We show how it is possible to design stabilizing laws for these SA, by extending the optimal control technique developed for stable switched systems.

For brevity of notation we refer to this particular class of switched systems as $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$.

7.3 Problem formulation

The general problem of this chapter is to design a stabilizing law for a switched system of the form (7.1). Before proceeding further it appears useful to recall some basic definitions that will occur in the following. For more details we address to [68].

7.3.1 The notion of stability

Consider the *non autonomous system*

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}, t) \tag{7.2}$$

where $f : D \times [0, \infty) \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in $\boldsymbol{x}$ on $D \times [0, \infty)$, and $D \subset \mathbb{R}^n$ is a domain that contains the origin $\boldsymbol{x} = \boldsymbol{0}$.

Definition 7.1 (Equilibrium point) *The* origin *is an* equilibrium point *for (7.2) if*

$$f(\boldsymbol{0}, t) = 0, \quad \forall t \geq 0. \tag{7.3}$$

■

Definition 7.2 (Stability of the equilibrium point) *The equilibrium point* $\boldsymbol{x} = \boldsymbol{0}$ *of (7.2) is*

- stable *if, for all* $\varepsilon > 0$*, there exists* $\delta = \delta(\varepsilon, t_0) > 0$ *such that*

$$\|\boldsymbol{x}(t_0)\| < \delta \quad \Rightarrow \quad \|\boldsymbol{x}(t)\| < \varepsilon, \;\; \forall t \geq t_0 \geq 0; \tag{7.4}$$

- unstable *if it is not stable;*
- asymptotically stable *(AS) if it is stable and there exists a positive constant* $\delta = \delta(t_0)$ *such that* $\boldsymbol{x}(t) \to \boldsymbol{0}$ *as* $t \to \infty$*, for all* $\|\boldsymbol{x}(t_0)\| < \delta$*;*
- exponentially stable *(ES) if there exist positive constants* δ*,* K*, and* λ *such that*

$$\|\boldsymbol{x}(t)\| \leq K \|\boldsymbol{x}(t_0)\| \, e^{-\lambda(t-t_0)}, \quad \forall \, \|\boldsymbol{x}(t_0)\| < \delta. \tag{7.5}$$

If asymptotic (or exponential) stability holds for any initial state, the equilibrium point is said to be globally *asymptotically (or exponentially) stable.* ■

Note that exponential stability implies asymptotic stability, which in turn implies stability.

Definition 7.3 (Global stabilizability) *The switched system* $\{\boldsymbol{A}_i\}_{i \in \mathcal{S}}$ *is said to be* globally stabilizable *if there exists a switching control law* $i(\boldsymbol{x}, t)$ *such that the controlled system is globally stable. Analogous definitions hold for global asymptotic (or exponential) stabilizability.* ■

Note that if at least one dynamics $\boldsymbol{A}_i$ is Hurwitz, then the system $\{\boldsymbol{A}_i\}_{i \in \mathcal{S}}$ is obviously globally exponentially stabilizable.

We show ho to compute a conic switching law $i(\boldsymbol{x}, t)$, when it does exist, such that the controlled system $\{\boldsymbol{A}_i\}_{i \in \mathcal{S}}$ is globally asymptotically stable. In particular, we provide a procedure that guarantees to determine a globally asymptotically stable switching law whenever the system is globally exponentially stabilizable.

7.3.2 The optimal control problem

The proposed stabilizing procedure is based on the solution of an optimal control problem of the form, in consistency with Definition 3.15 and the restriction listed in Section 7.2.

$$\begin{aligned}
& J_\infty^* \triangleq \min_{\mathcal{I}, \mathcal{T}} F(\mathcal{I}, \mathcal{T}) \triangleq \int_0^\infty \boldsymbol{x}'(t) \boldsymbol{Q}_{i(t)} \boldsymbol{x}(t) dt \\
& \text{s.t. } \; \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)} \boldsymbol{x}(t), \;\; \boldsymbol{x}(0) = \boldsymbol{x}_0, \;\; i(0) = i_0 \\
& \qquad i(t) = i_k \; \in \; succ(i_{k-1}) \text{ for } \tau_k \leq t < \tau_{k+1}, \\
& \qquad \tau_{k+1} \geq \tau_k,
\end{aligned} \tag{7.6}$$

$k \;\in \mathbb{N}$. We will denote this optimal control problem annexed to the switched system SA with the simplified notation OP($\mathcal{S}$), omitting the subscript ∞ and the class restriction SA, that will be assumed valid in the rest of the chapter.

For further explanation on the model and on the problem refer to Sections 3.3 and 3.4.

As in Chapter 6 we will build the result on infinite number of switches by considering the extension of a finite number of switches in the form:

$$
\begin{aligned}
J_N^*(\boldsymbol{x}_0, i_0) &\triangleq \min_{\mathcal{I},\mathcal{T}} F(\mathcal{I},\mathcal{T}) \triangleq \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt \\
\text{s.t. } & \dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{i(t)}\boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad i(0) = i_0 \\
& i(t) = i_k \in succ(i_{k-1}) \text{ for } \tau_k \leq t < \tau_{k+1}, k = 0, \ldots, N \\
& \tau_0 = 0, \ \tau_{N+1} = +\infty
\end{aligned}
\tag{7.7}
$$

We denote by $i^*(t)$, $t \in [0, +\infty)$, $i^*(t) = i_k^*$ for $\tau_k^* \leq t < \tau_{k+1}^*$ the switching trajectory solving (7.7), and $\mathcal{I}^*$, $\mathcal{T}^*$ the corresponding optimal sequences.

7.3.3 State feedback control law

We recall the STP presented in Section 4.3 and analyzed in the particular case of Section 4.6.2.

The optimal control law for the optimization problem (7.7) takes the form of a state feedback, i.e., it is only necessary to look at the current system state $\boldsymbol{x}$ in order to determine if a switch from linear dynamics $\boldsymbol{A}_{i_{k-1}}$ to $\boldsymbol{A}_{i_k}$, should occur.

More precisely, for a given mode $i \in \mathcal{S}$ when k switches are still available, it is possible to construct a table $\mathcal{C}_k^i$ that partitions the state space $\mathbb{R}^n$ into s regions $\mathcal{R}_j$'s, $j = 1, \ldots, s = |\mathcal{S}|$.

Whenever $i(t) = i$ and k switches are remaining, we use table $\mathcal{C}_k^i$ to determine if a switch should occur: as soon as the state reaches a point in the region $\mathcal{R}_j$ for a certain $j \in \mathcal{S} \setminus \{i\}$ we will switch to mode j; on the contrary, no switch will occur while the system's state belongs to $\mathcal{R}_i$.

The procedure that shows how to construct the table $\mathcal{C}_k^i$, for all $i \in \mathcal{S}$ and all $k = 1, \ldots, N$ for the switched system $\{\boldsymbol{A}_i\}_{i \in \mathcal{S}}$ is described in detail in Section 4.6.2, or in [29].

The procedure is based on the principle of optimality and it construct recursively a partition of the hybrid space $(\boldsymbol{x}, i)$, $\boldsymbol{x} \in \mathbb{R}^n$ (in this case $\boldsymbol{x} \in \Sigma_n$ is sufficient) and $i \in \mathcal{S}$, for each value of k remaining switches.

This partition is based on the information already known when $k-1$ switches are missing. Proceeding backwards in a recursive procedure all partitions $\mathcal{C}_k^i$ can be constructed.

The key function of the procedure is the *residual cost*, as in Definition 4.4, that we report here. Note that in absence of switching costs it holds $T_k(\cdot) = \tilde{T}_k(\cdot)$, from Definition 4.6.

Assume that k switches are missing and the current hybrid state is $(\boldsymbol{y}, i)$, where $\boldsymbol{y} \in \Sigma_n$.

The residual cost is:

$$
T_k(\boldsymbol{y}, i, j, \varrho) = \boldsymbol{y}'\bar{\boldsymbol{Q}}_i(\varrho)\boldsymbol{y} + T_{k-1}^*(\boldsymbol{z}, j) \tag{7.8}
$$

where $\varrho \geq 0$ and $j \in \mathcal{S}$ are the current control variables, $\boldsymbol{z} \in \mathbb{R}^n$ is $\boldsymbol{z} = \bar{\boldsymbol{A}}_i(\varrho)\boldsymbol{y}$, i.e., the point reached after a permanence in mode i for a time ϱ.

The two members of the sum that defines $T_k(\boldsymbol{y}, i, j, \varrho)$ have the following physical meaning: the first one is the cost of the evolution with the current dynamics $\boldsymbol{A}_i$ for a time ϱ, the second one is the optimal residual cost from point $\boldsymbol{z}$ to infinity and its value has been determined at the previous step of the algorithm, when $k-1$ switches remain.

Its meaning can be easily understood once the function

$$
T_k(\boldsymbol{y}, i)^* = \min_{\substack{j \in \mathcal{S} \\ \varrho \geq 0}} T_k(\boldsymbol{y}, i, j, \varrho), \tag{7.9}
$$

optimal residual cost, is defined[1].

To complete the procedure the function $T_0^*(\boldsymbol{y}, i)$ is defined as follows:

$$T_0^*(\boldsymbol{y}, i) \triangleq \begin{cases} \boldsymbol{y}'\boldsymbol{Z}_i\boldsymbol{y} \; if \; \boldsymbol{A}_i \; is \; stable \\ +\infty \qquad\qquad else \end{cases}, \tag{7.10}$$

Hence, defined $\varrho^*(\boldsymbol{x}, i)$ and $j^*(\boldsymbol{x}, i)$ as in equations (4.19) and (4.18) respectively, we obtain, $\forall\, \boldsymbol{y} \in \Sigma_n$, the table $\mathcal{C}_k^i$, in agreement with the Definition 4.7.

When all dynamics of $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ are unstable the equation (7.10) is badly posed. In fact in such a case if we apply brute force the STP the last residual cost will be equal to infinite and consequently all the other function $T_k^*(\cdot) = +\infty$.

In the sequel we will explain and formally prove that this inconsistency can be avoided by the introduction in $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ of a *dummy* dynamics $\tilde{\boldsymbol{A}}_{s+1}$ (Hurwitz) that serves *only* to give a finite value to the function $T_0^*(\boldsymbol{y}, i)$.

In other words we consider an *augmented* system, defined in the sequel, $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$, that obeys to Assumption 4.1, but such that the table $\tilde{\mathcal{C}}_\infty$[2], $i = 1, \ldots, s$ *does not include* the region $\mathcal{R}_{s+1}$.

Informally the new dynamics can be seen as a *launch pad* for the STP. Once the tables have converged, it can be removed, because the system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ has reached its *orbital equilibrium*.

7.3.4 Lexicographic ordering and uniqueness

In general the couple $(j^*(\boldsymbol{y}, i), \varrho^*(\boldsymbol{y}, i))$, arguments that minimizes (7.9), may be not unique.

Hence a state $\boldsymbol{y}$ may be assigned to different regions $\mathcal{R}_j$, for $j \in \mathcal{S}'$.

To remove this source of nondeterminism we will refer to the lexicographic ordering of the couples $(j^*(\boldsymbol{y}, i), \varrho^*(\boldsymbol{y}, i))$ as in Definition 4.8. This ensures that an optimal table is also unique.

There is, however, another issue related to this problem that must be addressed

Consider the case in which the optimal arguments of (7.9) from point $\boldsymbol{y}$ in location i are $\varrho^*(\boldsymbol{y}, i) = 0$ and $j^*(\boldsymbol{y}, i) = j$.

This signifies that an immediate switch towards j^* is required.

It may be the case that the system, once entered in location j^* requires an immediate switch to another location, say p, causing the presence of 2 switches in zero time.

This behavior is undesirable, because it leads to a potential risk of a Zenoness when the number of available switches goes to infinite.

To avoid this it is sufficient to reset

$$j^* = \arg\min \; T_{k-1}(\boldsymbol{x}, j^*, j, \varrho).$$

This choice signifies that the next location of i must coincide with the optimal switching strategy obtained from j^* at the previous iteration problem.

In fact when $T_k(\boldsymbol{x}, i, j, \varrho)$ is minimized with $\varrho = 0$ it clearly holds

---

[1]In general $T_{k-1}^*(\boldsymbol{z}, j)$ is calculated only on Σ_n, and $\boldsymbol{z} \notin \Sigma_n$. However, without switching costs, in force of Property 4.1 it trivially holds

$$T_{k-1}^*(\boldsymbol{z}, j) = \lambda^2 T_{k-1}^*\left(\frac{\boldsymbol{z}}{\lambda}, j\right),$$

where $\lambda = \|\boldsymbol{z}\|$.

[2]Later on it will be proved that for this class of systems it even holds $\mathcal{C}_\infty^i \equiv \mathcal{C}_\infty^j$ for all $i, j \in \mathcal{S}$, thus we will refer only to table $\mathcal{C}_\infty$.

$$T_k^*(\boldsymbol{y}, i) = T_{k-1}^*(\boldsymbol{y}, j).$$

When this extra precaution is taken, we can ensure that a spacing condition

$$\tau_{k+1} - \tau_k > 0$$

is always verified during an optimal evolution.

We observe that this *simple idea* is significant only if the function $\delta_{\min}(i) = 0$, $\forall\, i \in \mathcal{S}$. Moreover it is applicable only to arbitrary mode switched systems SA. In the general case, i.e., when

$$j^*(\boldsymbol{y}, i) \ \in \ succ(i),$$

it is not possible to reset $j^* = h \triangleq \arg\min \ T_{k-1}(\boldsymbol{x}, j^*, j, \varrho)$. unless also $h \in succ(i)$.

For this reason the *simple idea* was not introduced in Section 6.3.

7.4 The optimal control problem with an infinite number of switches

We will recall some results obtained in Chapter 6 that hold for the STP when the Assumption 4.1 is satisfied and when the number of switches is allowed to grow to infinity.

Here we shall only give the statements; proofs and explanations are in the mentioned Chapter 6.

Moreover in this particular case we will use the notation $J_k^*(\boldsymbol{x}, i)$ to indicate the residual cost $T_k^*(\boldsymbol{x}, i)$. Initially, when the number of switches was fixed a priori the notation $J_N^*(\boldsymbol{x}, i)$ indicated the total cost and $T_k^*(\boldsymbol{x}, i)$ the intermediate residual cost. Now, where the number of switches is a varying parameter, this distinction becomes senseless.

Property 7.1 (Monotonicity of the cost) *Let $N, N' \in \mathbb{N}$. If $N' > N$ and the switched system evolves along an optimal trajectory, then for some initial hybrid state $(\boldsymbol{x}_0, i_0)$,*

$$J_{N'}^*(\boldsymbol{x}_0, i_0) \leq J_N^*(\boldsymbol{x}_0, i_0) < +\infty.$$

■

Proposition 7.1 (Convergence of the cost) *For some initial state $(\boldsymbol{x}_0, i_0)$, $\boldsymbol{x}_0 \neq \mathbf{0}$, and*

$$\forall\, \varepsilon' > 0,\ \exists\, \bar{N} = \bar{N}(\boldsymbol{x}_0, i_0)$$

such that $\forall\, N > \bar{N}$,

$$J_N^*(\boldsymbol{x}_0, i_0) - J_{\bar{N}}^*(\boldsymbol{x}_0, i_0) < \varepsilon.$$

■

Proposition 7.2 (Normalized convergence of the cost) *For any continuous initial state $\boldsymbol{x}_0$, $\boldsymbol{x}_0 \neq 0$, and $\forall\, \varepsilon > 0$, $\exists\, \bar{N}$ such that for all $N > \bar{N}$,*

$$\frac{J_N^*(\boldsymbol{x}_0, i) - J_{\bar{N}}{}^*(\boldsymbol{x}_0, j)}{J_N^*(\boldsymbol{x}_0, i)} < \varepsilon,$$

for all $i, j \in \mathcal{S}$. ■

Theorem 7.1 *Given a fixed relative tolerance ε, if $\bar{N}$ is chosen as in Proposition 7.2 then for all $N > \bar{N} + 1$ it holds that*

$$\mathcal{C}^i_N \equiv \mathcal{C}^i_{\bar{N}+1}.$$

■

The above result allows one to compute with a finite procedure the optimal tables for a switching law when N goes to infinity. In such a case, in fact, it holds that for all $i \in \mathcal{S}$,

$$\mathcal{C}^i_\infty \equiv \lim_{N\to\infty} \mathcal{C}^i_N \equiv \mathcal{C}^i_{\bar{N}+1}.$$

We are now ready to formally prove a useful result for the switched system considered in this chapter.

Theorem 7.2 *Given a fixed relative tolerance ε, if $\bar{N}$ is chosen as in Proposition 7.2 then for all $i, j \in \mathcal{S}$ it holds that*

$$\mathcal{C}^i_{\bar{N}+1} \equiv \mathcal{C}^j_{\bar{N}+1}.$$

□

Proof. It follows from the fact that, by Proposition 7.2,

$$J^*_{\bar{N}+1}(\boldsymbol{x}_0, i) = J^*_{\bar{N}+1}(\boldsymbol{x}_0, j)$$

for all $i, j \in \mathcal{S}$, and from the uniqueness of the optimal tables as discussed in Section 7.3.4. ■

This result also allows one to conclude that for all $i \in \mathcal{S}$

$$\mathcal{C}_\infty \equiv \lim_{N\to\infty} \mathcal{C}^i_N,$$

i.e., all tables converge to the same one.

Remark 7.1 *Note that this result does not hold for the general switched system, but merely for the class considered here, i.e., completely connected and all* minimum permanence times *set to 0. If one of this condition are violated then the tables will converge differently from one location to another.* ■

This result is in fact very intuitive. In fact in the completely connected automaton every location has the same point of view of the rest of the systems.

Thus if from the hybrid state $(\boldsymbol{x}, i)$ it is better to switch to location j, where there will occur a non trivial evolution, this must be true for all locations $h = 1, \ldots, s$, $h \neq i, j$ and in location j the optimal strategy is to remain in j.

This is in force of the uniqueness of the optimal solution defined in Section 7.3.4.

Hence the optimal strategy in a point $(\boldsymbol{x}, i)$ is independent from the current location i.

To construct the table $\mathcal{C}_\infty$ the value of $\bar{N}$ is needed. We do not provide so far any analytical way to compute $\bar{N}$, therefore our approach consists in constructing tables until a convergence criterion is met.

Table $\mathcal{C}_\infty$ can be used to compute the optimal feedback control law that solves an optimal control problem of the form (7.7) with $N = \infty$.

More precisely, when an infinite number of switches is available, we only need to keep track of the table $\mathcal{C}_\infty$.

If the current continuous state is $\boldsymbol{x}$ and the current location is i, on the basis of the knowledge of the color of $\mathcal{C}_\infty$ in $\boldsymbol{x}$, we decide if it is better to still evolve with the current dynamics $\boldsymbol{A}_i$ or switch to a different dynamics, that is univocally determined by the color of the table in $\boldsymbol{x}$.

Remark 7.1. Note that the table $\mathcal{C}_\infty$ is *Zeno-free*, i.e., it guarantees that no Zeno instability may occur when it is used to compute the optimal feedback control law. This property is guaranteed by the procedure used for their construction as discussed in Section 7.3.4. ■

7.5 Stabilizability of unstable switched systems

In this section we deal with the problem of stabilizing a switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ whose linear dynamics $\boldsymbol{A}_i$ are not stable.

In particular, we show that a solution to this problem — when it does exist — can be obtained by solving an optimal control problem of the form (7.7) with $N = \infty$.

More precisely, we show how this problem can be solved by applying the switching table procedure to a *dummy* problem that satisfies the assumption that at least one dynamics $\boldsymbol{A}_i$ is Hurwitz.

When the original switched system is stabilizable, we select among all stabilizing laws a switching law that minimizes a given quadratic performance index.

7.5.1 Intuitive notions

It should be clear that we propose a method that is able to find a stabilizing switching law from a given initial state of conic shape *only if* the switched systems admits at least a stabilizing solution from a given initial state.

Definition 7.4 (Order of convergence) *Consider a function $f(t)$ such that*

$$\lim_{t\to\infty} f(t) = 0,$$

we define the order of convergence k the value such that

$$\lim_{t\to\infty} t^k f(t) = l < \infty.$$

■

Proposition 7.3 *The STP gives a stabilizing solution* only if *the system admits an AS solution such that*

$$\|\boldsymbol{x}(t)\|^2 \to 0$$

with order $k > 1$. □

Proof. In fact the STP is based on the finiteness of an infinite time horizon integral that weights the square norm of vector $\boldsymbol{x}$ weighted by matrices $\boldsymbol{Q}_i$. If the square norm converge to zero with order $k \leq 1$ the integral is no longer finite, hence the STP is not applicable. ■

To better illustrate this statement we provide some intuitive examples.

Example 7.1 *The function*

$$f(t) = \|\boldsymbol{x}(t)\| = \frac{\|\boldsymbol{x}_0\|}{\sqrt{t+1}}$$

goes to 0 as $t \to +\infty$ *with order* $k = \frac{1}{2}$.

If the norm of the vector field $\boldsymbol{x}(t)$ *of a switched system has this convergence property, it is asymptotically stable but our procedure will fail, i.e., it will not find a conic switching law that stabilizes the system.*

In fact its fundamental mechanism is the minimization of the integral of a norm[3], *whose sum is expected to be finite (in the Lebesgue sense), but*

$$\min \int_0^\infty \boldsymbol{x}'(t)\boldsymbol{x}(t)dt = \min \int_0^\infty \|\boldsymbol{x}(t)\|^2 dt = \|\boldsymbol{x}_0\|^2 \min \int_0^\infty \frac{1}{t+1} dt = +\infty.$$

■

Example 7.2 *The function*

$$f(t) = \|\boldsymbol{x}(t)\| = \frac{\|\boldsymbol{x}_0\|}{t+1}$$

goes to 0 as $t \to +\infty$ *with order* $k = 1$. *In this case the STP will work because*

$$\int_0^\infty \|\boldsymbol{x}(t)\|^2 dt < +\infty.$$

■

Example 7.3 *The function*

$$f(t) = \|\boldsymbol{x}(t)\| = \|\boldsymbol{x}_0\| e^{-\gamma^2 t}$$

goes to 0 as $t \to +\infty$ *with order* $k = +\infty$. Fortiori *in this case the STP will work because*

$$\int_0^\infty \|\boldsymbol{x}(t)\|^2 dt < +\infty.$$

■

On the contrary, if our procedure works, it does not guarantee straightforward the order k. Thus we can only ensure an asymptotic stability.

In [109] it is proved that for some classes of switched system the order of convergence of the vector field is exponential.

7.5.2 Theoretical results

We present the following preliminary result that is essential for the rest of the derivation.

Still we deal with a switched system of the form $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$.

For simplicity of notation let us indicate with OP($\mathcal{S}$) the optimal control problem of the form (7.7), associated to $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$.

Consider also a set $\tilde{\mathcal{S}} \subset \mathcal{S}$ and the corresponding switched system $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$. This system is trivially obtained from $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ by *refining* the oriented graph.

Figure 7.1 shows graphically this operation.

The corresponding OP($\tilde{\mathcal{S}}$) is the same optimal control problem that associates to each location h, $h \in \tilde{\mathcal{S}}$, the same matrix $\boldsymbol{Q}_h$ as in OP($\mathcal{S}$).

[3] Here wlg we considered $\boldsymbol{Q}_{i(t)} = \boldsymbol{I}_n$.

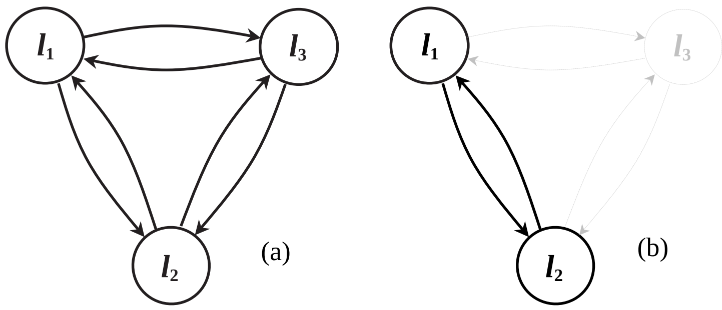


Fig. 7.1. *Refining operation of a switched system. (a) Switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$, $\mathcal{S} \equiv \{1,2,3\}$.(b) Switched system $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$, $\tilde{\mathcal{S}} \equiv \{1,2\} \subset \mathcal{S}$, refined from (a).*

Proposition 7.4 *Let us consider an OP($\mathcal{S}$) of the form (7.7) with $N = \infty$. If the table $\mathcal{C}_\infty$, solution of OP($\mathcal{S}$), is a partition of $\mathcal{R}_j$, $j \in \tilde{\mathcal{S}} \subset \mathcal{S}$, then $\tilde{\mathcal{C}}_\infty$, solution of OP($\tilde{\mathcal{S}}$) coincides with $\mathcal{C}_\infty$, i.e.,*

$$\mathcal{C}_\infty \equiv \tilde{\mathcal{C}}_\infty.$$

□

Proof. The validity of the statement follows from the definition of the table $\mathcal{C}_\infty$ and the possibility of using it to derive an optimal feedback control law for OP($\mathcal{S}$).

If the region $\mathcal{R}_h$ corresponding to a certain mode $\boldsymbol{A}_h$ does not appear in $\mathcal{C}_\infty$, it means that it is never convenient to switch to mode $\boldsymbol{A}_h$, or to evolve with $\boldsymbol{A}_h$ if it is the initial mode.

Hence, for any initial state, an optimal solution of OP($\mathcal{S}$) is also an optimal solution for OP($\tilde{\mathcal{S}}$), and in force of the uniqueness of the switching tables it can only be

$$\mathcal{C}_\infty \equiv \tilde{\mathcal{C}}_\infty.$$

regardless of the current continuous state and the current dynamics. ■

The above result enables us to use the switching table procedure to compute a stabilizing switching law, if it does exist, for switched systems whose dynamics are unstable.

In particular, the proposed approach is based on the construction of an *augmented system* and an *augmented OP* that are defined as follows.

Definition 7.5 (Augmented system) *Consider a switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$. We define the* augmented system *a switched system of the same class $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$, such that*

- $|\tilde{\mathcal{S}}| = |\mathcal{S}| + 1$*;*
- $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$ *is composed of the same dynamics as* $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$*;*
- $\boldsymbol{A}_{s+1} \in \{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$ *is Hurwitz.*

■

The augmented system $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$ coincides with the system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$, but it contains an extra dynamics, $\boldsymbol{A}_{s+1}$ that is Hurwitz.

As an example consider the switched system whose oriented graph is depicted in Figure 7.1. If the dynamics associated to location 3 is Hurwitz, then the system in Figure 7.1.(a) is an augmented system of Figure 7.1.(b).

Definition 7.6 (Augmented OP) *Let us consider an OP($\mathcal{S}$) with $N = \infty$. Assume that all possible modes $\boldsymbol{A}_i$, $i \in \mathcal{S}$, are not stable and the corresponding weighting matrices $\boldsymbol{Q}_i$, $i \in \mathcal{S}$, are strictly positive definite.*

Let $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$ be an augmented system of $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ as in Definition 7.5.

We define augmented OP($\tilde{\mathcal{S}}$) *of OP($\mathcal{S}$)an optimal control problem of the form (7.7) with $N = \infty$, and $\boldsymbol{Q}_{s+1} = q\tilde{\boldsymbol{Q}}$, $q \in \mathbb{R}^+$ and $\tilde{\boldsymbol{Q}} > \boldsymbol{0}$.* ■

In other terms, for all $i = 1, \ldots, s$ all $\boldsymbol{Q}_i$ coincide for both systems. The new $\boldsymbol{Q}_{s+1}$ is a strictly positive matrix multiplied by a factor q, whose role will be clear in the sequel.

Now, let us prove the following proposition.

Proposition 7.5 *Consider an OP($\mathcal{S}$) with $N = \infty$. Let $J^*_\infty(\boldsymbol{x}_0, i_0)$ be the optimal cost value solution of OP($\mathcal{S}$) when the initial state is $(\boldsymbol{x}_0, i_0)$.*

Now let $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ be the optimal cost value of the augmented OP($\tilde{\mathcal{S}}$), as in Definition 7.6, for the same initial state.

The optimal cost $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ is a strictly increasing function of q for all values of q such that the Hurwitz dynamics $\boldsymbol{A}_{s+1}$ appears in the optimal evolution of OP($\tilde{\mathcal{S}}$). □

Proof. We prove this by contradiction. Consider two different augmented problems, OP$^{(1)}(\tilde{\mathcal{S}})$ and OP$^{(2)}(\tilde{\mathcal{S}})$ that differ for their value of q.

In particular, let $q^{(1)}$ and $q^{(2)}$ be the values of the coefficient q associated to OP$^{(1)}(\tilde{\mathcal{S}})$ and OP$^{(2)}(\tilde{\mathcal{S}})$ respectively, and let $q^{(1)} > q^{(2)}$.

Assume that

$$\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q^{(1)}) = \tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q^{(2)}),$$

respectively the costs of the evolutions $(\boldsymbol{x}(t), i(t))^{(1)}$ and $(\boldsymbol{x}(t), i(t))^{(2)}$ are the same.

We will show how this assumption brings to a contradiction. In fact, if we use the solution of OP$^{(1)}(\tilde{\mathcal{S}})$ and compute the cost of the evolution $(\boldsymbol{x}(t), i(t))^{(1)}$ for a generic initial state using the weights of OP$^{(2)}(\tilde{\mathcal{S}})$ we obtain a value that is smaller than $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q^{(1)})$.

For the absurd assumption this value is also smaller than $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q^{(2)})$, and this is a contradiction, because by definition $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q^{(2)})$, obtained solving OP$^{(2)}(\tilde{\mathcal{S}})$, is the minimum value. ■

If a switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ composed of exclusively unstable dynamics is stabilizable, then a stabilizing switching law can always be computed using the STP.

We do this by annexing to the switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ an OP($\mathcal{S}$) and by considering the solution of an augmented OP($\tilde{\mathcal{S}}$).

The main feature of the computed switching law is that it stabilizes the system and at the same time it minimizes the annexed quadratic performance index.

This is very appealing, because this method provides a criterium to design a stabilizing switching law (which is itself a major goal in system theory) and furthermore the feedback stabilizing law minimizes an index.

7.5.3 Theorem on the stabilizability

Let us now state and prove the main theorem of this chapter.

Theorem 7.3 (Stabilizability of unstable switched system) *Given a switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$, let us consider an optimal control problem of the form (7.7) with $N = \infty$ and weighting matrices $\boldsymbol{Q}_i > 0$, $i \in \mathcal{S}$. Then, let us define an augmented $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$ and an augmented OP($\tilde{\mathcal{S}}$), as in Definitions 7.5 and 7.6.*

(i) The switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ *is globally exponentially stabilizable* $\Longrightarrow \exists\ \tilde{q} \in \mathbb{R}^+$ *such that the table* $\mathcal{C}_\infty$*, computed by solving OP(*$\tilde{\mathcal{S}}$*), does not contain the color associated to* $\tilde{\boldsymbol{A}}$*.*

(ii) The switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ *is asymptotically stabilizable* $\Longleftarrow \exists\ \tilde{q} \in \mathbb{R}^+$ *such that the table* $\mathcal{C}_\infty$*, computed by solving OP(*$\tilde{\mathcal{S}}$*), does not contain the color associated to* $\tilde{\boldsymbol{A}}$*.*

□

Proof. We denote $J^*_\infty(\boldsymbol{x}_0, i_0)$ the optimal cost of the optimal control problem for the system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ when the initial state is $(\boldsymbol{x}_0, i_0)$, and $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ the corresponding optimal cost of the augmented system $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$.

The cost $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ is obviously finite for all finite values of q because $\tilde{\boldsymbol{A}}$ is stable.

Moreover, it is upper limited by the value of $J^*_\infty(\boldsymbol{x}_0, i_0)$, i.e., $\forall q \in \mathbb{R}^+$,

$$\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q) \le J^*_\infty(\boldsymbol{x}_0, i_0).$$

Finally, $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ is a quadratic function of $\boldsymbol{x}_0$, i.e., if $\boldsymbol{x}_0 = \lambda \boldsymbol{y}_0$ then

$$\tilde{J}^*_\infty(\lambda\boldsymbol{y}_0, i_0, q) = \lambda^2 \tilde{J}^*_\infty(\boldsymbol{y}_0, i_0, q).$$

(i) Assume that the switched system $\{\boldsymbol{A}_i\}_{\in\mathcal{S}}$, is globally exponentially stabilizable.

This implies that $J^*_\infty(\boldsymbol{x}_0, i_0) < +\infty$, for all $\boldsymbol{x}_0 \in \mathbb{R}^n$ and for all $i_0 \in \mathcal{S}$.

In fact, any control law that is exponentially stable implies that along any trajectory it holds

$$\int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt = \int_0^\infty \boldsymbol{y}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{y}(t)\ \|\boldsymbol{x}(t)\|^2 dt$$

$$\le K\int_0^\infty \|\boldsymbol{x}(t)\|^2 dt \le Kc^2\|\boldsymbol{x}_0\|^2 \int_0^\infty e^{-2\lambda t}dt \le +\infty,$$

where we have written $\boldsymbol{x}(t) = \boldsymbol{y}(t)\ \|\boldsymbol{x}(t)\|$ with $\|\boldsymbol{y}(t)\| = 1$,

$$K = \min_{\substack{i\,\in\,\mathcal{S}\\ \boldsymbol{y}\,\in\,\Sigma_n}} \boldsymbol{y}'\boldsymbol{Q}_i\boldsymbol{y},$$

and $c, \lambda \in \mathbb{R}^+$.

By Proposition 7.5 we know that $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ is an increasing function of q for all values of q such that $\boldsymbol{A}_{s+1}$ appears in the optimal evolution.

Therefore, we may conclude that if $\{\boldsymbol{A}_i\}_{\in\mathcal{S}}$ is globally exponentially stabilizable then $\exists\ q^{(1)}(\boldsymbol{x}_0, i_0) \in \mathbb{R}^+$ such that

$$\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q^{(1)}(\boldsymbol{x}_0, i_0)) = J^*_\infty(\boldsymbol{x}_0, i_0).$$

Moreover, if the equality holds for a certain value of $q = q^{(1)}(\boldsymbol{x}_0, i_0)$, then it also holds for all $q > q^{(1)}(\boldsymbol{x}_0, i_0)$.

In fact, the above equality implies that the optimal control law of the augmented OP($\tilde{\mathcal{S}}$) requires no evolution with the stable mode $\boldsymbol{A}_{s+1}$.

If this is the case when its weighting matrix is $\boldsymbol{Q}_{s+1} = q^{(1)}\tilde{\boldsymbol{Q}}$, then *fortiori* when its weighting matrix is $\boldsymbol{Q}_{s+1} = q\tilde{\boldsymbol{Q}}$ with $q > q^{(1)}(\boldsymbol{x}_0, i_0)$.

Now, the result holds if we let

$$\tilde{q} = \max_{\substack{i_0 \in \mathcal{S} \\ x_0 \in \mathbb{R}^n}} q^{(1)}(\boldsymbol{x}_0, i_0) = \max_{\substack{i_0 \in \mathcal{S} \\ y_0 \in \Sigma_n}} q^{(1)}(y_0, i_0),$$

where the second equality follows from the fact that $\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, q)$ is a quadratic function of $\boldsymbol{x}_0$.

If we define the augmented OP($\tilde{\mathcal{S}}$) with $\boldsymbol{Q}_{s+1} = \tilde{q}\tilde{\boldsymbol{Q}}$, then for all values of $\boldsymbol{x}_0 \in \mathbb{R}^n$ and all $i_0 \in \mathcal{S}$, it holds that

$$\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, \tilde{q}) = J^*_\infty(\boldsymbol{x}_0, i_0),$$

i.e., the controlled system never switches to dynamics $\boldsymbol{A}_{s+1}$, neither evolves with $\boldsymbol{A}_{s+1}$ if it is the initial mode.

This obviously implies that the table $\mathcal{C}_\infty$, computed applying the switching table procedure to the augmented OP($\tilde{\mathcal{S}}$) with $\boldsymbol{Q}_{s+1} = \tilde{q}\tilde{\boldsymbol{Q}}$, does not contain the color associated to the stable mode $\boldsymbol{A}_{s+1} = \tilde{\boldsymbol{A}}$.

(ii) Assume that $\exists\ \tilde{q}$ such that the switching table $\mathcal{C}_\infty$, computed applying the switching table procedure to the OP($\tilde{\mathcal{S}}$), does not contain the color associated to the stable mode $\boldsymbol{A}_{s+1} = \tilde{\boldsymbol{A}}$.

By Proposition 7.4 this implies that the control law that results using table $\tilde{\mathcal{C}}_\infty$ is also optimal for the OP($\mathcal{S}$).

Therefore, being

$$\tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, \tilde{q}) < +\infty,$$

and $J^*_\infty(\boldsymbol{x}_0, i_0) = \tilde{J}^*_\infty(\boldsymbol{x}_0, i_0, \tilde{q})$ for all $\boldsymbol{x}_0 \in \mathbb{R}^n$ and all $i_0 \in \mathcal{S}$, it follows that $J^*_\infty(\boldsymbol{x}_0, i_0) < +\infty$.

It is not difficult to show, with the same argument we used in [50], that the finite value of the optimal cost for all initial states and dynamics implies that the switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ is globally asymptotically stabilizable. ■

The above theorem provides an efficient way to deal with the problem of determining an asymptotic stabilizing switching law for a switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ with linear unstable modes, that can be summarized in the following steps.

1. associate to the switched system an OP($\mathcal{S}$) with $N = \infty$;
2. define an augmented system $\{\boldsymbol{A}_i\}_{i\in\tilde{\mathcal{S}}}$ and OP($\tilde{\mathcal{S}}$) choosing q very large positive real number;
3. construct the table $\tilde{\mathcal{C}}_\infty$ solving OP($\tilde{\mathcal{S}}$);
4. If this table does not contain the color associated to the stable mode $\boldsymbol{A}_{s+1}$, by Theorem 7.3(ii), we conclude that the original switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ is globally asymptotically stabilizable. In such a case, we compute the stabilizing feedback control law that minimizes the chosen quadratic performance index using table $\mathcal{C}_\infty$.

Note, finally, that the procedure may also find control laws that locally stabilizes a system, as shown in the examples described in Sections 7.6.1 and 7.6.2.

We do not provide an a priori rule to establish if the switched system is stabilizable and in such a case, an analytical way to compute an appropriate value of q. In this case the solution of the problem of *knowing* if the system is stabilizable remains open.

Nevertheless in all numerical examples taken from the literature, we found out that if the system is stabilizable it was sufficient to use a large value of q ($10^{10} \div 10^{20}$) to compute stabilizing laws.

7.6 Numerical examples

In this section we will provide some numerical specific examples and in particular we will use the theoretical results to provide stabilizing switching laws of switched systems that minimize a performance index.

Three examples are taken from the literature, and known to be stabilizable. In particular one is inspired by the famous example of *Branicky* [17], another one is inspired by the example described by *Pettersson et al.* in [40] and stabilized with LMI approaches. The third one is taken from a work of *Colaneri et al.* who studied stabilizability via optimal issues for a switched system of the same class considered here. All examples are composed of strictly unstable dynamics.

We then show what happens to the STP when the system is evidently non *globally stabilizable*. The region associated to the stable dynamics, albeit an extremely high associated weights in the OP($\tilde{\mathcal{S}}$), it does not disappear.

Finally we provide an example that can be even solved *analytically*, providing the possibility of comparing the results obtained with the STP.

7.6.1 Examples from literature

Example 1: from *Branicky*

As a first example of the described approach we choose a variant of a very well-known switched system [17] $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$, with $s = 3$ and

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & -10 \\ 100 & 1 \end{bmatrix},\ \boldsymbol{A}_2 = \begin{bmatrix} 39.97 & -77.50 \\ 32.50 & 37.97 \end{bmatrix},\ \boldsymbol{A}_3 = \begin{bmatrix} -37.97 & -77.50 \\ 32.50 & 39.97 \end{bmatrix}.$$

Note that dynamics $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$ are obtained from dynamics $\boldsymbol{A}_1$ by an axis rotation of $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ degrees respectively.

All dynamics $\boldsymbol{A}_i$'s are unstable.

To determine a stabilizing switching law we first associate to the switched system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$ an optimal control problem of the form (7.7) with $N = \infty$.

In particular, we take $\boldsymbol{Q}_i = \boldsymbol{I}_2$, $i = 1, 2, 3$.

We define an augmented OP with the stable dynamics

$$\boldsymbol{A}_4 = -\boldsymbol{A}_1$$

and weighting matrix

$$\boldsymbol{Q}_4 = 10^5\tilde{\boldsymbol{Q}},$$

where $\tilde{\boldsymbol{Q}} = \boldsymbol{I}_2$.

We construct the table $\mathcal{C}_\infty$. More precisely, we apply the procedure to construct the tables $\mathcal{C}_N^i$ for finite values of N and we find out that, for a sufficiently large value of N, namely $N = 15$, the tables converge to the same one. The table $\mathcal{C}_\infty$ is reported in Figure 7.2.

We can immediately observe that the color associated to the stable dynamics $\boldsymbol{A}_4$ never appears. This means that, regardless of the initial state, the optimal trajectory of the augmented OP is obtained by infinitely switching among unstable dynamics $\boldsymbol{A}_i$, $i = 1, 2, 3$.

This allows one to conclude that the switched system $\{\boldsymbol{A}_i\}_{i\in\{1,2,3\}}$ is globally asymptotically stabilizable. Moreover, the table $\mathcal{C}_\infty$ can be used to compute the stabilizing feedback control law that minimizes the chosen quadratic performance index.

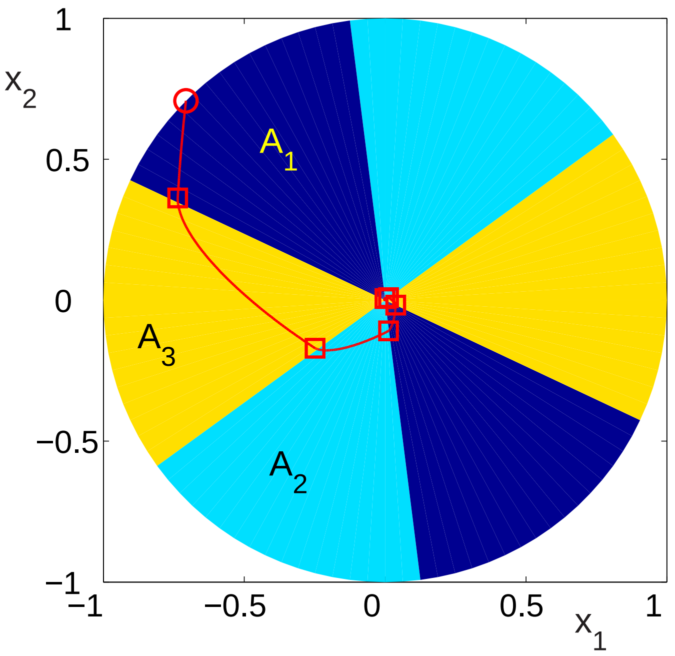


Fig. 7.2. *The optimal trajectory of the switched system* $\{\boldsymbol{A}_i\}_{i\in\{1,2,3\}}$ *of Example 7.6.1.*

An example of an optimal trajectory is reported in Figure 7.2 when the initial state is

$$\boldsymbol{x}_0 = \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}, \; i_0 = 1.$$

The stabilizing and optimal sequences are

$$\begin{array}{lcll} \mathcal{I}^* & = & & \{\;\; 1 \quad\; 3 \quad\; 2 \quad\; 1 \quad\; 3 \quad\; 2 \quad\; 1 \;\; \ldots\} \\ \mathcal{T}^* & = & 10^{-2} & \{\, 0.48\; 3.84\; 3.78\; 3.42\; 3.72\; 3.48\; 3.18 \ldots\} \\ J^*_\infty & = 0.0208. & & \end{array}$$

Note that the system, because of the homogeneous regions, presents a periodic behavior.

Example 2: from *Pettersson et al.*

This example is taken from [40], where it was stabilized via a LMI approach. The same examples was also analyzed in [64], where the switching rule is obtained via a probabilistic gradient based algorithm.

The system dynamics are:

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 10 \\ 0 & 0 \end{bmatrix}, \; \boldsymbol{A}_2 = \begin{bmatrix} 1.5 & 2 \\ -2 & -0.5 \end{bmatrix}.$$

The annexed optimal control problem associates to each dynamics the weight identity matrices, and allows $N = \infty$ switches. Hence $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{I}_2$.

As in the other examples we consider an augmented OP, given by the couple

$$\boldsymbol{A}_3 = -\boldsymbol{A}_2, \; \boldsymbol{Q}_3 = 10^5 \boldsymbol{Q}_1. \tag{7.11}$$

The STP converges, in this case, after $N = 16$ switches, and the table $\tilde{\mathcal{C}}_\infty$ does not contain the color of the dummy dynamics, hence it is the stabilizing one of the original system, only composed of unstable dynamics.

The table, juxtaposed with an exemplificative trajectory, is depicted in Figure 7.3.

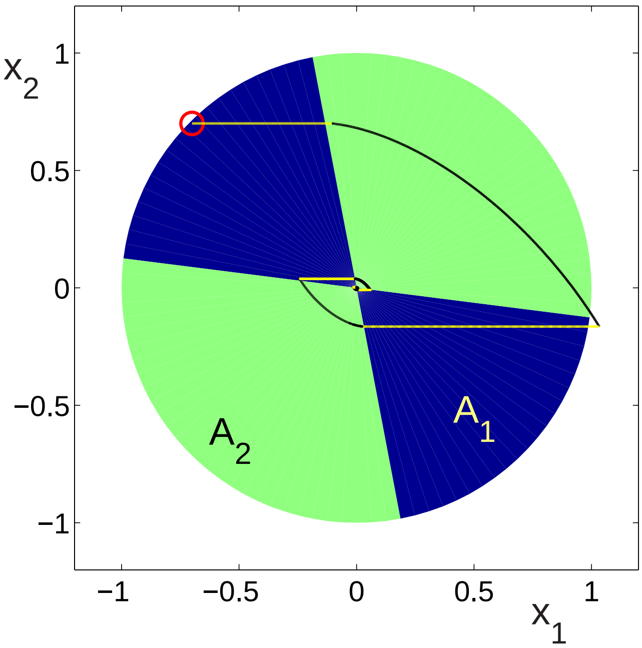


Fig. 7.3. *Table $\mathcal{C}_\infty$ for the switched system taken from [40] and described in Example 7.6.1. The obtained table stabilizes the system and it minimizes an LQR like performance index.*

An example of an optimal trajectory is reported in Figure 7.3 when the initial state is

$$\boldsymbol{x}_0 = \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix},\ i_0 = 1.$$

The stabilizing and optimal sequences, according to the given index, are

$$\begin{array}{lcl} \mathcal{I}^* &=& \{ \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \ \ldots\} \\ \mathcal{T}^* &=& \{ \ 0.09\ 0.77\ 0.61\ 0.77\ 0.61\ 0.77\ 0.61 \ldots\} \\ J^*_\infty = 0.76. && \end{array}$$

We take advantage of this example to highlight the fact that there might exists other stabilizing laws, but their performance is lower in terms of the considered index.

For sake of completeness we provided another switching law, reported in Figure 7.4. We may easily observe that this law, however stabilizing, is not optimal in the sense of minimizing the given performance index.

The trajectory starting from the same initial point is completely different and it is described by the following schedule:

$$\begin{array}{lcl} \mathcal{I} &=& \{ \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \quad 2 \quad 1 \ \ldots\} \\ \mathcal{T} &=& \{ \ 0.08\ 0.86\ 0.27\ 0.86\ 0.27\ 0.86\ 0.27 \ldots\} \\ J_\infty = 1.26. && \end{array}$$

and obviously its cost $J_\infty > J^*_\infty$.

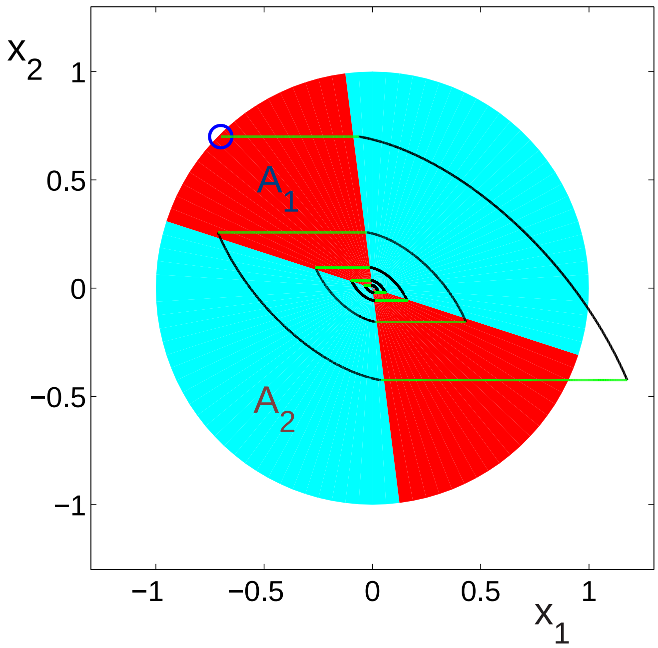


Fig. 7.4. *Stabilizing, however non optimal, switching table for the switched system taken from [40] and described in Example 7.6.1.*

Example 3: from *Colaneri et al.*

This example is taken from a recent work of *Colaneri and Geromel* [26], to appear at the triennial IFAC (International Federation of Automatic Control) conference 2005. In this paper the authors consider the issue of providing a stabilizing switching sequence. Their approach is based on Lyapunov like methods.

We recall now the result provided by Colaneri and Geromel that interestingly suits our research. The reader is referred to [26] for proofs and details.

Given a switched system of the form $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$, i.e., a switched system composed of autonomous dynamics and whose oriented graph is completely connected, the goal is to search for a set of matrices $\boldsymbol{Z}_i$ that satisfies the Lyapunov-Metzler equation, namely

$$\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i + \sum_{j=1}^{s}\pi_{j,i}\boldsymbol{Z}_j < \boldsymbol{0}, \tag{7.12}$$

for $i = 1,\dots,s$, where $s = |\mathcal{S}|$ and π_{ji} are the entries of a matrix of Metzler class[4].

The solution of equation (7.12) provides the switching signal $i(\boldsymbol{x}(t))$ in feedback form as

$$i(\boldsymbol{x}(t)) = \arg\min_{i=1,\dots,s} \boldsymbol{x}'(t)\boldsymbol{Z}_i\boldsymbol{x}(t). \tag{7.13}$$

[4]The class of Metzler matrices [26] is constituted by all matrices $\Pi \in \mathbb{R}^{s\times s}$, with elements π_{ij} such that $\pi_{i,j} \geq 0,\ i \neq j$ and $\sum_{i=1}^{s}\pi_{i,j} = 0,\ \forall\, j$.

Furthermore if the equation (7.12) becomes

$$\boldsymbol{A}_i'\boldsymbol{Z}_i + \boldsymbol{Z}_i\boldsymbol{A}_i + \sum_{j=1}^{s}\pi_{j,i}\boldsymbol{Z}_j + \boldsymbol{Q} < \boldsymbol{0}, \tag{7.14}$$

with $\boldsymbol{Q} \geq \boldsymbol{0}$, then

$$\int_0^\infty \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt \leq \min_{i=1,\ldots,s} \boldsymbol{x}_0'\boldsymbol{Z}_i\boldsymbol{x}_0, \tag{7.15}$$

We expect that the switching signal $i(\boldsymbol{x}(t)) = \arg\min_{i=1,\ldots,s} \boldsymbol{x}'(t)\boldsymbol{Z}_i\boldsymbol{x}(t)$ where $\boldsymbol{Z}_i$'s solve (7.14), is the optimal one, i.e., it minimizes the integral in equation (7.15).

Thus we applied both procedure to the example presented in the paper.

A switched system is composed of 2 dynamics, namely

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ 2 & -9 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} 0 & 1 \\ -2 & 2 \end{bmatrix},$$

both *unstable* and an annexed OP with weight matrices $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{Q} = \boldsymbol{I}_2$.

Colaneri and Geromel solve 7.14 and obtain the following matrices:

$$\boldsymbol{Z}_1 = \begin{bmatrix} 6.7196 & 1.6293 \\ 1.6293 & 1.0222 \end{bmatrix}, \quad \boldsymbol{Z}_2 = \begin{bmatrix} 6.0825 & 2.1293 \\ 2.1293 & 2.2206 \end{bmatrix},$$

and using equation (7.13) we can obtain the switching region depicted in Figure 7.5(a).

Now we apply to the given setup the STP, described in this chapter. To this purpose we define the augmented problem

$$\boldsymbol{A}_3 = -\boldsymbol{A}_2, \ \boldsymbol{Q}_3 = 10^5\boldsymbol{Q},$$

and we keep constructing switching tables until the convergence is met. For this example we established a convergence after $N = 15$ switches. The last table, $\mathcal{C}_{15} \equiv \mathcal{C}_\infty$, is depicted in Figure 7.5(b).

Observe that the tables obtained with the two different methods are the same. We choose to give different colors to remark that they are obtained with different approaches.

To conclude we simulated these switching tables starting from two different initial points.

The first one is $\boldsymbol{x}_0 = [-0.7, 0]'$ and initial dynamics $i_0 = 2$. We obtained the following values:

- Optimal switching sequence: $\mathcal{I} = \{2, 1, 2, 1, \ldots\}$;
- Optimal switching times: $\mathcal{T} = \{0.37, 0.01, 0.02, 0.01, \ldots\}$;
- Optimal cost $J^*(\boldsymbol{x}_0, i_0) = 0.561$.

The trajectory is depicted in *red* in Figure 7.5(b), or in Figure 7.6, where it has been zoomed.

The second one is $\boldsymbol{x}_0 = [0, 0.7]'$ and initial dynamics $i_0 = 1$. We obtained the following values:

- Optimal switching sequence: $\mathcal{I} = \{1, 2, 1, 2, 1, \ldots\}$;
- Optimal switching times: $\mathcal{T} = \{0.42, 0.62, 0.01, 0.02, 0.01, \ldots\}$;

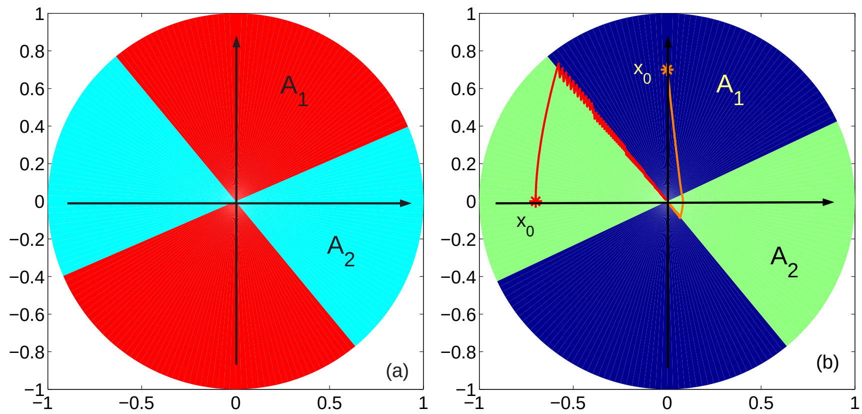


Fig. 7.5. *Example studied in [26] reported in Section 7.6.1: (a) Switching table obtained with the approach described by* Colaneri and Geromel, *(b) Switching table obtained with the STP described in this chapter and optimal trajectories. Note that these switching manifolds admit sliding motions.*

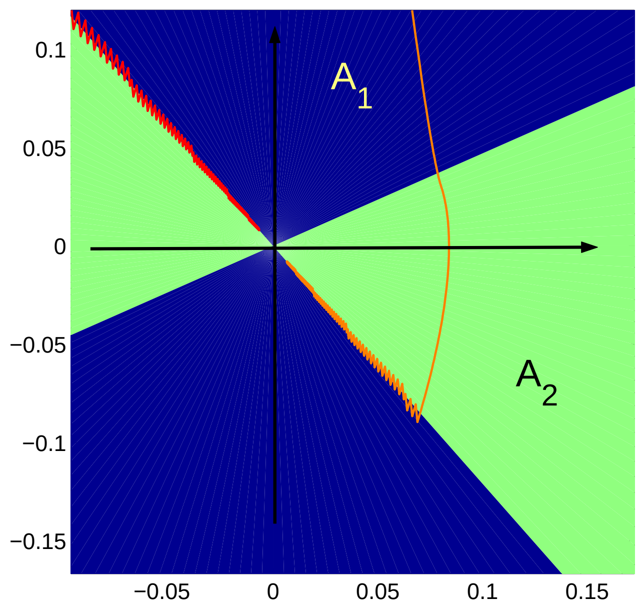


Fig. 7.6. *Zoomed picture of Figure 7.5(b) representing two optimal trajectories of the system described in Section 7.6.1. Observe the sliding motion around a switching surface, a frequent behavior of switched systems.*

- Optimal cost $J^*(\boldsymbol{x}_0, i_0) = 0.0394$.

The trajectory is depicted in *orange* in Figure 7.5(b), or in Figure 7.6, where it has been zoomed.

7.6.2 Non stabilizable case

We shall present another example to illustrate the behavior of our procedure when the switched system is not globally stabilizable.

This example was inspired by [64].

We consider a system $\{\boldsymbol{A}_i\}_{i\in\mathcal{S}}$, with $s = 2$ and

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Clearly this system is not stabilizable in the I and III quadrant, where both vector fields diverge in independent directions ($\boldsymbol{A}_1$'s flow is parallel to the $\boldsymbol{x}_1$ direction and $\boldsymbol{A}_2$'s to the $\boldsymbol{x}_2$ one).

A more convincing argument can be found by looking at the picture in Figure 7.7 of these two vector fields.

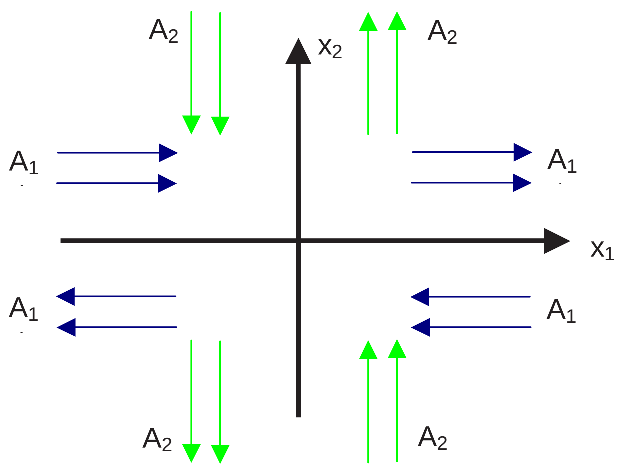


Fig. 7.7. *Vector fields of the dynamics considered in the example in Section 7.6.2. Note that in the first and third quadrant there is no possibility to obtain a switching strategy that stabilizes the switched system.*

In the II and IV quadrant, where the direction of the flow is opposite, we expect the presence of an optimal switching sequence, that stabilizes the system.

To maintain the high symmetry of the system we consider

$$\boldsymbol{Q}_i = \boldsymbol{I}_2,$$

$i = 1, 2$, and we define an augmented optimal control problem with the stable dynamics

$$\boldsymbol{A}_3 = -\boldsymbol{I}_2, \ \boldsymbol{Q}_3 = 10^{13}\boldsymbol{Q}_1.$$

The converging switching table of the augmented problem $\tilde{\mathcal{C}}_\infty$ is achieved after $N = 10$ switches.

Its construction required a finer discretization of Σ_2 (150 points, versus 50 of the examples in Section 7.6.1) and it is reported in Figure 7.8.

Note that when $x_1 x_2 \geq 0$ the minimum cost is obtained by performing the evolution exclusively with the stable dynamics $\boldsymbol{A}_3$, despite the extremely high value of the weighting matrix $\boldsymbol{Q}_3$.

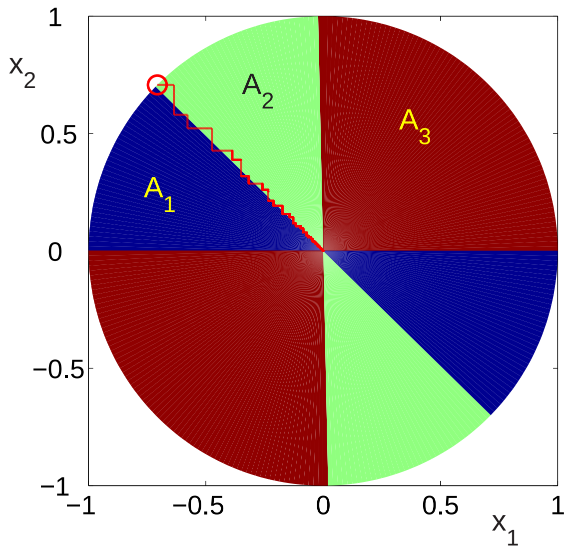


Fig. 7.8. *Non globally stabilizable example. Note that, however extremely expensive, the controller cannot find a stabilizing sequence of only dynamics $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ in the odd quadrants, thus the color of the augmented dynamics $\boldsymbol{A}_3$ cannot disappear from $\tilde{\mathcal{C}}_\infty$. Note also the stabilizing sliding motion along the switching attractive manifold $x_2 = -x_1$.*

On the contrary, when $x_1 x_2 < 0$, there exists a stabilizing switching law that collapses into a sliding mode along the surface $x_2 = -x_1$.

This intuitive result can also be obtained analytically by providing the expression of the cost as a function of the angle variable on Σ_2.

Moreover it is possible to identify analytically the equivalent dynamics $\boldsymbol{A}_e = -0.5\boldsymbol{I}_2$ along the sliding surface.

Let us consider now the initial point

$$\boldsymbol{x}_0 = \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}, \; i_0 = 1.$$

The trajectory is plotted in Figure 7.8, and it has the typical chattering shape of the sliding mode.

The corresponding cost (sampling step $dt = 0.1$) is

$$J^*_\infty(\boldsymbol{x}_0, i_0) = 1.0025.$$

Obviously, when the sampling step of the simulation program goes to zero, the evolution follows the equivalent dynamics $\boldsymbol{A}_e$ with equivalent weighting matrix $\boldsymbol{Q}_e = \boldsymbol{I}_2$ whose quadratic cost is trivially equal to 1.

7.6.3 Analytical example

In this section we describe a particular switched system whose stabilizing control law that minimizes the corresponding OP can be computed analytically. Furthermore we will apply the procedure described in this chapter to show the equivalency of the approaches.

Consider the switched system featured by the unstable dynamics

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; \boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}.$$

It is trivially the same of Section 7.6.2, except for the fact that $\boldsymbol{A}_2$ has opposite sign.

However this is sufficient to guarantee the existence of a stabilizing control law.

This can be seen from the vector fields of these dynamics that are parallel to the coordinate axis as depicted in Figure 7.11a.

Our purpose, with this very simple example, is to obtain the switching law analytically, and then show that it coincides with the one obtained with the STP.

Analytical construction of the optimal switching region

To develop the conic switching surfaces analytically it is necessary to choose a general initial state and a general switching surface.

We can choose

$$\boldsymbol{x}_0 = \begin{bmatrix} -1 \\ \alpha \end{bmatrix}, \; i_0 = 2.$$

Note that this point is *general* because one DOF can be omitted in force of the 2-homogeneity of the cost.

We parameterize the switching surfaces by their slopes $m_1, m_2 \; \in (-\infty, +\infty)$, i.e.,

$$\begin{cases} (1) \; x_2 = m_1 x_1 \\ (2) \; x_2 = m_2 x_1, \end{cases}$$

and such that we use $\boldsymbol{A}_1$ ($\boldsymbol{A}_2$) if $\boldsymbol{x}'\boldsymbol{G}\boldsymbol{x} > 0$ (< 0), where

$$\boldsymbol{G} = \begin{bmatrix} m_1 m_2 & -\frac{m_1+m_2}{2} \\ -\frac{m_1+m_2}{2} & 1 \end{bmatrix}$$

defines the conic regions depicted in Figure 7.9. Note that according to this set up it is $m_1 > 0$ and $m_2 < 0$.

We first calculate the sequence of switching states, namely $\boldsymbol{x}_k$, $k = 0, 1, 2, 3, \ldots$. To avoid confusion we recall that the vectors are in *bold*, while the components of the vectors aren't. Let us initially give $\boldsymbol{x}(t)$ as a function of the state transition matrices and a generic initial state $\boldsymbol{x}_k$:

$$\begin{aligned} \boldsymbol{x}(t) = \bar{\boldsymbol{A}}_1(t)\boldsymbol{x}_k = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_k, \;\; (a) \\ \boldsymbol{x}(t) = \bar{\boldsymbol{A}}_2(t)\boldsymbol{x}_k = \begin{bmatrix} 1 & 0 \\ -t & 1 \end{bmatrix} \boldsymbol{x}_k. \; (b) \end{aligned} \tag{7.16}$$

We start from the point

$$\boldsymbol{x}_0 = \begin{bmatrix} -1 \\ \alpha \end{bmatrix}, \; i_0 = 2,$$

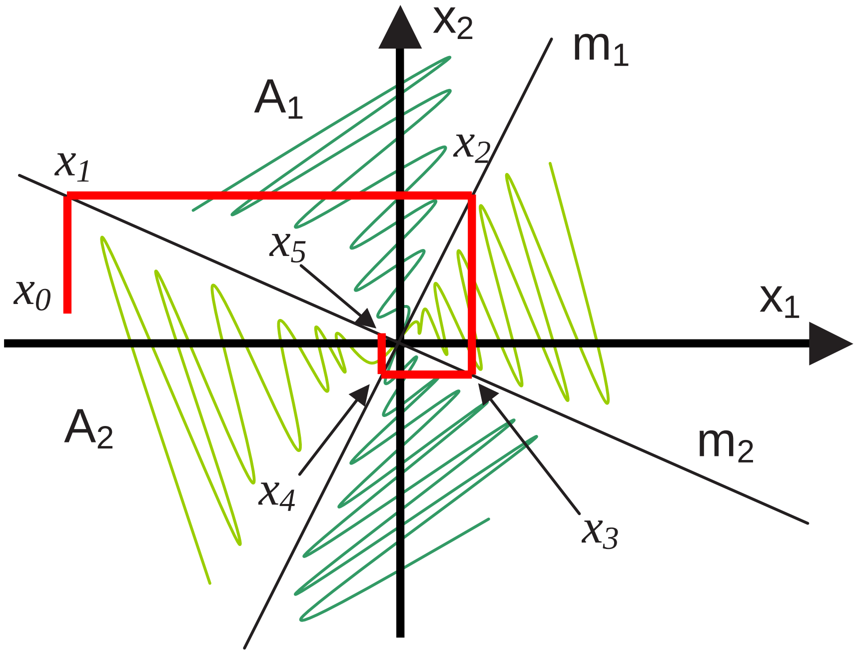


Fig. 7.9. *Parameterization of the switching surfaces for the example of Section 7.6.3.*

with $\alpha < -m_2$. Evolving with dynamics $\boldsymbol{A}_2$ we will hit the first surface (of slope m_2) after a time $\delta_0 = -m_2 - \alpha$. This can be obtained using equation (7.16.b) as follows:

$$\boldsymbol{x}(t) = \begin{bmatrix} 1 & 0 \\ -t & 1 \end{bmatrix} \begin{bmatrix} -1 \\ \alpha \end{bmatrix},$$

and imposing that $\boldsymbol{x}(t)$ belong to the switching surface of slope m_2, hence $\boldsymbol{x}(t) = [-1, -m_2]'$.

We obtain the first switching point

$$\boldsymbol{x}_1 = \begin{bmatrix} -1 \\ -m_2 \end{bmatrix}.$$

From this point we now switch into dynamics $\boldsymbol{A}_1$. With analogous calculations, i.e., using equation (7.16.a) with initial point $\boldsymbol{x}_1$, the next switching point is

$$\boldsymbol{x}_2 = \frac{m_2}{m_1} \begin{bmatrix} -1 \\ -m_1 \end{bmatrix}.$$

The time spent with dynamics $\boldsymbol{A}_1$ is the time necessary to cross the *cone* between the two manifolds parameterized by m_2 and m_1. For a well known property of the linear systems in $\mathbb{R}^2$, this time is *independent* from the particular initial state. Its value is $\delta_H = \dfrac{m_2 - m_1}{m_2 m_1}$.

Now from point $\boldsymbol{x}_2$ the system switches again into $\boldsymbol{A}_2$ and in analogy with the previous calculations we obtain the point

$$\boldsymbol{x}_3 = \frac{m_2}{m_1}\begin{bmatrix} -1 \\ -m_2 \end{bmatrix} = \frac{m_2}{m_1}\boldsymbol{x}_1$$

after a time $\delta_V = m_1 - m_2$.

Now from point $\boldsymbol{x}_3$ the system switches again into $\boldsymbol{A}_1$ and in analogy with the previous calculations we obtain the point

$$\boldsymbol{x}_4 = \left(\frac{m_2}{m_1}\right)^2 \begin{bmatrix} -1 \\ -m_1 \end{bmatrix} = \frac{m_2}{m_1}\boldsymbol{x}_2,$$

after a time δ_H.

It goes now straightforward that the sequences of switching points are governed geometrically. In particular:

$$\begin{aligned}&\boldsymbol{x}_0 = \begin{bmatrix} -1 \\ \alpha \end{bmatrix},\ \boldsymbol{x}_1 = \begin{bmatrix} -1 \\ -m_2 \end{bmatrix},\ \boldsymbol{x}_2 = \tfrac{m_2}{m_1}\begin{bmatrix} -1 \\ -m_1 \end{bmatrix},\ \boldsymbol{x}_3 = \tfrac{m_2}{m_1}\boldsymbol{x}_1,\ \boldsymbol{x}_4 = \tfrac{m_2}{m_1}\boldsymbol{x}_2,\\ &\ldots,\ \boldsymbol{x}_{2k+1} = \left(\tfrac{m_2}{m_1}\right)^k \boldsymbol{x}_1,\ \boldsymbol{x}_{2k+2} = \left(\tfrac{m_2}{m_1}\right)^k \boldsymbol{x}_2,\ \ldots,\end{aligned} \tag{7.17}$$

with $k = 1, \ldots, \infty$.

Simple considerations on the switching sequences and on the geometrical behavior of the given dynamics lead to assert the following remark.

Remark 7.2 *The conditions of stability for this particular system are the following:*

- *if*
  1. $m_2 m_1 \geq 0$*;*
  2. $\dfrac{m_2}{m_1} < -1$*;*

  the system is unstable. In fact in case (1), i.e., the switching surfaces are both in I and III quadrants or II and IV quadrants, the system has no possibility to perform any stabilizing rotation. In case (2) the switching sequence in (7.17) *diverges;*
- *if* $\dfrac{m_2}{m_1} = -1$ *the system is at its limit cycle [46, 53]. In fact the switching sequence in* (7.17) *becomes stationary;*
- *if* $-1 < \dfrac{m_2}{m_1} < 0$ *the system is ES;*

■

The conic switching surfaces, whose corresponding stabilizing switching sequence minimizes the performance index

$$J = \int_0^{+\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt,$$

can be calculated analytically for the simple case $\boldsymbol{Q}_1 = \boldsymbol{Q}_2 = \boldsymbol{I}_2 = \boldsymbol{Q}$.

To this aim we need to express the cost J as a function of the initial point and the parameters of the switching surfaces m_1 and m_2. This is appealing, because, the trajectory is composed of 2 kinds of homothetic branches, the vertical and the horizontal ones.

Since the initial points of the vertical (horizontal) branches have the form $\boldsymbol{x}_{2k+1} = \left(\frac{m_2}{m_1}\right)^k \boldsymbol{x}_1$ ($\boldsymbol{x}_{2k+2} = \left(\frac{m_2}{m_1}\right)^k \boldsymbol{x}_2$) we expect to formulate J as a combination of geometrical series indexed by k.

Moreover, as stated above, the time spent in the vertical (horizontal) branch is $\delta_V = m_1 - m_2$ ($\delta_H = \dfrac{m_2 - m_1}{m_2 m_1}$), *regardless* of the particular value of k.

Hence the cost expression is:

$$J(\alpha, 2, m_1, m_2) = \boldsymbol{x}_0' \bar{\boldsymbol{Q}}_2(\delta_0)\boldsymbol{x}_0 + \sum_{k=1}^{+\infty} \boldsymbol{x}_{2k-1}' \bar{\boldsymbol{Q}}_1(\delta_H)\boldsymbol{x}_{2k-1} + \sum_{k=1}^{+\infty} \boldsymbol{x}_{2k}' \bar{\boldsymbol{Q}}_2(\delta_V)\boldsymbol{x}_{2k}, \tag{7.18}$$

where 2 is the index of the initial dynamics.

We may calculate analytically

$$\begin{aligned} \bar{\boldsymbol{Q}}_2(\delta_0) &= \int_0^{\delta} \bar{\boldsymbol{A}}_1'(t)\bar{\boldsymbol{A}}_1(t)dt, \\ \bar{\boldsymbol{Q}}_1(\delta_H) &= \int_0^{\delta_H} \bar{\boldsymbol{A}}_2'(t)\bar{\boldsymbol{A}}_2(t)dt, \\ \bar{\boldsymbol{Q}}_2(\delta_V) &= \int_0^{\delta_V} \bar{\boldsymbol{A}}_1'(t)\bar{\boldsymbol{A}}_1(t)dt. \end{aligned} \tag{7.19}$$

and substitute the values $\boldsymbol{x}_k$ given in equation (7.17). Now all terms in (7.18) can be expressed as a function of (α, m_1, m_2). The global expression becomes long and complicated, but conceptually simple. In fact, once we have the $J(\alpha, 2, m_1, m_2)$ all is left to do is to find its global minimum, in the two variables m_1 and m_2, in the stability range provided in Remark 7.2.

The minimization task, via partial derivative, is not analytically feasible, because in the cost expression there appears polynomial terms of order 6, but it can be done numerically.

A *smart* shortcut can be achieved considering that it must hold $m_1 m_2 = -1$, for sake of *symmetry*.

In fact there is no significant reason to believe that the system prefer to sojourn longer in one specific dynamics, because they are both weighted with the same matrix $\boldsymbol{Q}$.

Thus it can be $m_1 = m > 1$ and $m_2 = -\frac{1}{m}$ and this is a significant simplification, because we pass from a two variables problem into a single variable problem.

Now we would like to recalculate the function

$$J = J(\boldsymbol{x}_0, i_0, m) = J(\alpha, 2, m) = \int_0^{+\infty} \boldsymbol{x}'(t)\boldsymbol{Q}_{i(t)}\boldsymbol{x}(t)dt. \tag{7.20}$$

Using the sum expression of the cost we obtain that J can be written as

$$J(\alpha, 2, m) = \boldsymbol{x}_0' \bar{\boldsymbol{Q}}_{i_0}(\delta_0)\boldsymbol{x}_0 + \sum_{k=1}^{+\infty} \boldsymbol{x}_k' \bar{\boldsymbol{Q}}_{i_k}(\delta)\boldsymbol{x}_k, \tag{7.21}$$

where

- $\boldsymbol{x}_k$ are the switching states of equation (7.17) that can be seen also in Figure 7.9;
- $\delta_0 = \frac{1}{m} - \alpha > 0$;
- $\delta = \delta_H = \delta_V = m + \frac{1}{m} > 0$ is the time spent dynamics $\boldsymbol{A}_1$ or $\boldsymbol{A}_2$, from $\boldsymbol{x}_1$ on.

We can also recalculate the switching states $\boldsymbol{x}_k$:

$$\boldsymbol{x}_0 = \begin{bmatrix} -1 \\ \alpha \end{bmatrix},\ \boldsymbol{x}_1 = \begin{bmatrix} -1 \\ \frac{1}{m} \end{bmatrix},\ \boldsymbol{x}_2 = -\tfrac{1}{m^2}\begin{bmatrix} -1 \\ -m \end{bmatrix},\ \boldsymbol{x}_3 = -\tfrac{1}{m^2}\boldsymbol{x}_1,\ \boldsymbol{x}_4 = -\tfrac{1}{m^2}\boldsymbol{x}_2,$$
$$\ldots,\ \boldsymbol{x}_{2k+1} = \left(-\tfrac{1}{m^2}\right)^k \boldsymbol{x}_1,\ \boldsymbol{x}_{2k+2} = \left(-\tfrac{1}{m^2}\right)^k \boldsymbol{x}_2,\ \ldots, \tag{7.22}$$

with $k = 1, \ldots, \infty$.

Hence equation (7.21) can be rewritten as

$$J(\alpha, 2, m) = \boldsymbol{x}_0' \bar{\boldsymbol{Q}}_{i_0}(\delta_0)\boldsymbol{x}_0 + (\boldsymbol{x}_1' \bar{\boldsymbol{Q}}_{i_1}(\delta)\boldsymbol{x}_1 + \boldsymbol{x}_2' \bar{\boldsymbol{Q}}_{i_2}(\delta)\boldsymbol{x}_2)\sum_{k=0}^{+\infty} m^{-4k}. \tag{7.23}$$

Now, being

$$\bar{\boldsymbol{Q}}_{i_0}(\delta_0) = \begin{bmatrix} \delta_0 + \frac{\delta_0^3}{3} & -\frac{\delta_0^2}{2} \\ -\frac{\delta_0^2}{2} & \delta_0 \end{bmatrix},\ \bar{\boldsymbol{Q}}_{i_1}(\delta) = \begin{bmatrix} \delta & \frac{\delta^2}{2} \\ \frac{\delta^2}{2} & \delta + \frac{\delta^3}{3} \end{bmatrix},\ \bar{\boldsymbol{Q}}_{i_2}(\delta) = \begin{bmatrix} \delta + \frac{\delta^3}{3} & -\frac{\delta^2}{2} \\ -\frac{\delta^2}{2} & \delta \end{bmatrix}.$$

after some quite long calculations we obtain

$$J(\alpha, 2, m) = -\frac{\alpha}{3}\left(3 + \alpha^2\right) + \left(\frac{1}{m} + \frac{1}{3m^3} + \frac{(m^2+1)^3}{3m^3(m^2-1)}\right). \tag{7.24}$$

Finally, solving

$$\frac{\partial J}{\partial m} = 0,$$

and considering *acceptable* only the stabilizing solutions $m > 1$, in force of Remark 7.2, we find a unique solution

$$m^* \cong 3.146, \quad \frac{1}{m^*}$$

that represent the slope of the switching surface in the I and III quadrants, and in the II and IV quadrants respectively. Note that, as expected, the value of m^* is independent from α, i.e., from the initial point.

We depict in Figure 7.10 the equation (7.24) plotted in function of the design parameter m in the surrounding of its minimum value. The function cost reaches the minimum when $m = m^*$.

Numerical construction of the optimal switching region via STP

The same result can be obtained by applying the STP. To do this we consider the augmented problem, provided that its weight in the performance index expression is high. In particular

$$\boldsymbol{A}_3 = \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix}, \quad \boldsymbol{Q}_3 = 10^4 \boldsymbol{I}_2$$

Convergence is met in $N = 12$ switches. In Figure 7.11(b) the switching region is depicted, with two trajectories for different initial points. It goes straightforward that the optimal sequences, are, for the initial point

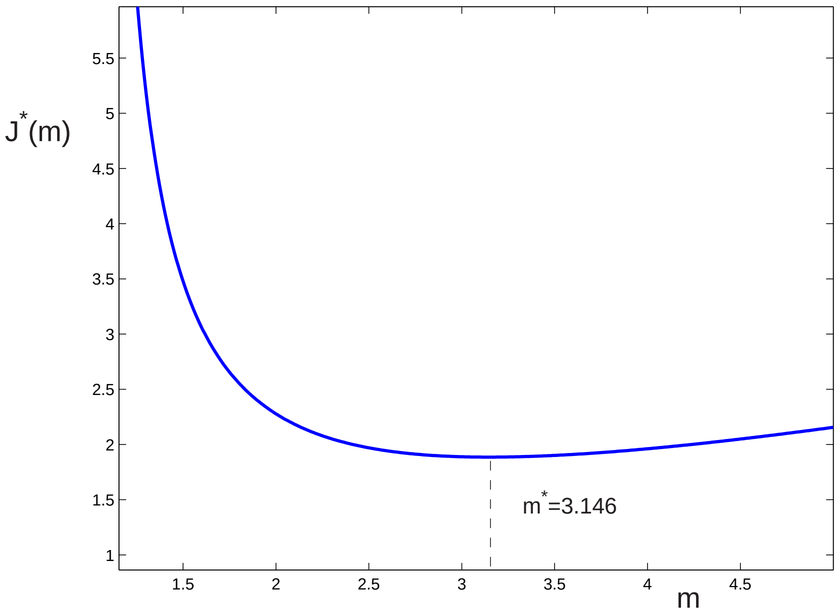


Fig. 7.10. *Plot of function* (7.24) *in the surrounding of its global minimum.*

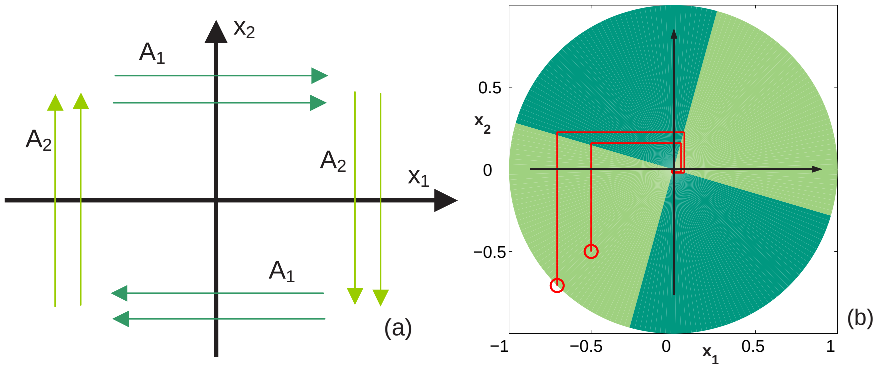


Fig. 7.11. *Example studied in Section 7.6.3: (a) Vector field of dynamics $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, (b) Optimal switching surfaces and two trajectories.*

$$\boldsymbol{x}_0 = \begin{bmatrix} -0.707 \\ -0.707 \end{bmatrix}, \; i_0 = 2,$$

$$\begin{aligned} \mathcal{I}^* &= \{\ 2\ \ 1\ 2\ 1\ 2\ 1\ 2\ \ldots\} \\ \mathcal{T}^* &= \{\ \delta_0\ \delta\ \delta\ \delta\ \delta\ \delta\ \delta\ \ldots\}. \end{aligned}$$

The cost of the trajectory obtained with the STP is $J_\infty^{*(STP)} = 1.6099$, while the exact value, from equation (7.24), with $m = m^* = 3.146$, $\alpha = -1$, normalized on Σ_2 is $J_\infty^* = 1.6095$

7.7 Conclusions

Based on the results of the optimal control of switched systems with a infinite number of admissible switches and at least one Hurwitz dynamics, we showed that this approach can also be efficiently applied when all LTI dynamics are not stable. This is done by solving an appropriate optimal control law, called the augmented OP, that contains a Hurwitz dynamics. In particular, we show that if the switched system with unstable modes is globally exponentially stabilizable, then an optimal feedback control law can be computed, that guarantees the closed-loop system to be globally asymptotically stable.

8

Concluding notes, open issues and future research interest

In this chapter we will summarize the contributions of this thesis, illustrate the open issues and propose some directions for future research.

8.1 Summary of contributions

The research contributions of this thesis are set out in Chapters 3, 4, 5, 6 and 7. We will subsequently summarize their contents.

In Chapter 3 the considered model is defined. We dealt with a subclass of the recent dynamic models known as *hybrid systems*, a mathematical formalization that integrates *event driven* dynamics with *time driven* dynamics. The subclass of interest is the *hybrid automata*, a model composed of a set of locations (nodes), each associated with a mode governed by a linear affine differential equation, and a set of edges (arcs), whose firing is the occurrence of a discrete event that provokes the mode switching. In this chapter we formally defined the hybrid automata, its *state*, composed of the continuous state $\boldsymbol{x} \in \mathbb{R}^n$ and the discrete state i the current active mode, its *properties* and its *dynamical behavior*. In parallel we defined an annexed optimal control problem, that is formulated as the minimization of a performance index based on the quadratic cost of the continuous state $\boldsymbol{x}$ and the sum of a cost associated to the event driven dynamics. The control variable is the piecewise function $i(t)$, namely the sequence of locations and switching instants. In a hybrid automaton the degree of freedom of the function $i(t)$ may be limited by the continuous state $\boldsymbol{x}(t)$. Besides we described in detail a subclass of the hybrid automata, commonly denoted as *switched system*, characterized by a function $i(t)$ independent from the continuous state $\boldsymbol{x}$.

In Chapter 4 we developed a procedure, the *switching table procedure* STP, that solves the optimal control problem defined in Chapter 3, for a switched system and for a finite number of switches. The procedure, based on dynamic programming arguments, consists in the construction of a set of tables that partition the state space into several regions which suggest the optimal switching strategy in feedback form. We proved that the thus obtained tables guarantee to find the global optimum of the given performance index. Moreover the dynamic programming principle bounds the computational complexity of the table construction, that is linear in the number of switches and quadratic in the number of modes of the switched system. The main drawback of the STP resides in the necessity of discretizing the *whole* state space, so practically limiting the application to low dimensional examples. Despite this, we showed that under particular conditions, specifically when all switching costs and affine terms are null, the regions are homogeneous, thus permitting to discretize only along the unitary semisphere.

In Chapter 5 we applied the STP to hybrid automata and to the case of finite number of switches. Namely we demonstrated that the procedure is still applicable even in the case when the design function $i(t)$ is allowed to take values from a set constrained by the current continuous value $\boldsymbol{x}$ of the hybrid evolution. In particular we analyzed in detail two special cases, defined in Chapter 3. The former is the *autonomous hybrid automata* AHA, where there may occur *internally forced* [122] spontaneous switchings, according to the current value $\boldsymbol{x}$. For this model, characterized by uncontrollable edges, we prove the conditions under which the STP may be applicable and the optimal solution is finite. The latter is the *constrained hybrid automata* CHA, where the choice of the controller may be restricted in order to respond to *safety* specifications on the output signal. This problem is solved *hierarchically*, namely a *low level* procedure deals with the specifications, and a *high level* procedure, STP, with the optimal control within the remaining degree of freedom left by the low level. In both cases a whole state space discretization is required, unless for the peculiar case where guards and invariants of the hybrid automaton, defined in Chapter 3, are homogeneous.

In Chapter 6 we considered the same model and problem studied in Chapter 4, i.e., switched systems, but we relaxed the condition that the number of switches is upper bounded by a finite value N. In other words we applied the STP for increasing values of N, pointing to *infinite*. Then we observed and formally proved that there *always* exists a *sufficiently large* value $\bar{N}$, independent from the particular initial state and initial mode, that marks the convergence of the switching tables. So the tables obtained for greater values than $\bar{N}$ are the same. This is a significant result because it allows one to use *only* the tables $\mathcal{C}^i_{\bar{N}} = \mathcal{C}^i_\infty$ indefinitely for $t \in [0, \infty)$. Furthermore the method has been successfully applied to the design of a semiactive suspension system of a quarter car model, and it appeared that its performances are consistently intermediate between a purely active and a purely passive one.

Lastly, in Chapter 7 we investigated the possibility of using the STP, for a switched system, restricted to the case of *completely connected* automaton, with *infinite* number of switches, as a design tool of a stabilizing switching signal. We demonstrated that this is possible provided that the switched system is *globally exponentially* stabilizable. In fact this is a sufficient condition to guarantee that the performance index, an integral of the quadratic norm of the state $\boldsymbol{x}(t)$, is finite. From this viewpoint the STP appears an alternative synthesis method of a feedback stabilizing control law for a switched system. Another significant *added value* proposed in Chapter 7 is the extension of the STP to the cases where all modes of the switched system are non Hurwitz. This is obtained by the addition of a slow and expensive Hurwitz dynamics in the original switched system, that serves as a *launching pad* for the STP, whose presence may disappear from table $\mathcal{C}_\infty$[1]. If this happens then $\mathcal{C}_\infty$ is the stabilizing switching table for the unstable given system. Specific examples from literature have been considered for comparison.

The software that implements the STP is described in Appendix E and it may be downloaded from the web site

http://www.diee.unica.it/∼dcorona/thesis.html.

8.2 Open issues

In this paragraph we will briefly describe some issues that still remained open in the development of the STP.

[1] If the switched system is completely connected we proved that $\mathcal{C}^i_\infty \equiv \mathcal{C}^j_\infty$, for all modes i, j of the system. Hence simply $\mathcal{C}^i_\infty \equiv \mathcal{C}_\infty$.

8.2.1 Estimation of the value of N in the table convergence issue

In Chapter 6 and 7 we showed how the STP can be extended to the case where the value of N, the number of available switches, becomes *infinite*. We proved formally that there exists a threshold value of $\bar{N}$ such that all tables constructed for greater values of $\bar{N}$ are the same. This is a relevant theoretical result, but the practical implementation may require an estimation or an upper bound of $\bar{N}$. In Chapter 6, Theorem 6.2, we provided a method, yet not sufficiently general. We argue that this value of $\bar{N}$ may be obtained by knowing the convergence rate of the state $\boldsymbol{x}$, but so far, to our knowledge, there are no general results for switched systems[2].

8.2.2 Estimation of the value of q in the table convergence issue

A similar problem arises in Chapter 7. The stabilizing switching signal for a switched system composed of only unstable dynamics is subject to the existence of a *sufficiently* large value of a parameter q that weights the stable augmented dynamics, see Theorem 7.3. This problem is similar to the one described in the previous Section 8.2.1. In fact if we know that the system is stabilizable then we *might* find an appropriate q such that the color of the stable dynamics disappears from $\mathcal{C}_\infty$. Conversely, if the color does not disappear it may be due to an erroneous choice of q, i.e., too small, or to the effective non stabilizability of the system. We might conclude that the existence of q is a criterion that guarantees stabilizability *a priori*. Hence a method that gives an upper bound on q would be a relevant result. In fact, to our knowledge, apart from special cases, i.e., quadratically stable systems [42], the available methods to assert stabilizability, based on the existence of multiple Lyapunov functions, are often burdensome.

8.2.3 Analytical estimation of the switching tables

Another open issue is the analytical calculation of the optimal switching tables. This is possible only in the extremely special case of $N = 1$ switch, because it descends straightforward from the solution of the Lyapunov equation, defined in A.3. We are quite sure that there must be some methods, based on LMI applied to multiple Lyapunov functions, that should solve this problem. A hint can be found in [26], however the setup is not exactly coincident with ours. Furthermore we are interested in calculating all tables, while a Lyapunov based method at most gives the table of convergence. Succeeding in this task is very significant, since it means that the procedure may be easily applied to higher dimensional problems.

8.2.4 Analytical calculation of the residual cost

From Chapter 4 on, while applying the STP, a recursive procedure, at step k we always encountered the problem of estimating the optimal residual cost, obtained numerically at step $k-1$, from a point $\boldsymbol{x}$ and given mode i. It would be extremely useful to have an *analytical* expression of the residual cost whenever more than one switch is still available. This would not only reduce the computational time of the procedure, but it may also gain in precision, because of a reduction of the error propagation. Note that this step has been partially done by considering an approximation, via linear spline interpolation, of the value of the cost in a point with the surrounding values in the discretization points. This is described in Appendix C.2. Finally we

[2] For example, Sun, in [109], calculates the stabilizing convergence rate for the class of *switched triangular systems*.

have the feeling that this issue and the one deployed in the previous Subsection 8.2.3 must be somehow related.

8.3 Future research interest

Here we will briefly describe how the STP can be further extended, and in addition some of my personal research interest are listed.

8.3.1 Stabilizability and optimal control of hybrid automaton

A natural extension fills a *gap* contained in this thesis. In fact we considered optimal control with *finite* number of switches for switched systems in Chapter 4 and for hybrid automata in Chapter 5. Then we considered *infinite* number of switches for switched systems in Chapter 6. Consistently, the extension to the infinite number of switches for hybrid automata is missing. This extension is still in progress and some preliminary results have been submitted at the IEEE International Symposium on Intelligent Control 2005 [33]. The possibility of performing an infinite number of switches in presence of constraints on the state space may cause the activation of undesirable behaviors such as *blocking* or *Zenoness*. Hence, before extending the STP to this case we considered important to set ahead a procedure that guarantees the *liveness* of the hybrid automaton. Briefly it consists in the design of a stabilizing switching table, for an unstable hybrid automaton, that *avoids* obstacles, i.e., forbidden regions of the state space. Here, rather than performing the calculations in continuous time, we passed to discrete time systems, which has been a necessary condition for the synchronization of the two approaches. However original and interesting the results are not, in our opinion, complete. In fact a structural criterion that permits us to conclude the existence of a stabilizing switching table is still missing.

8.3.2 Optimal quantized control

Another interesting extension is offered by the fact that we always considered autonomous dynamics. This may be considered too restrictive in many applications, hence we would be interested in building the feedback optimal control law for a non autonomous switched system, namely $\dot{\boldsymbol{x}} = \boldsymbol{A}_{i(t)}\boldsymbol{x} + \boldsymbol{B}_{i(t)}\boldsymbol{u}$, where $\boldsymbol{u} \in \mathbb{R}^m$ is a continuous control input. In these cases two controllers may be active: the switching signal $i(t)$ and the continuous action $\boldsymbol{u}(t)$. As it is, the problem appears complex, hence we recently started to consider restrictive cases. In particular, as a first step, we abandoned the *switched system* and considered a *quantized*, see, among many others, [73, 92], discrete time single input optimal control problem of the form $\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}u_j(k)$, where $u_j \in \{u_1, u_2, \ldots, u_s\}$ is a finite set of quantized inputs. The optimal control problem is a classical discrete LQR, with the additional constraint that u is quantized and bounded. Note that this problem is, to an extent, a particular affine system, hence the STP should be applicable straightforwardly. The numerical results, although very close to those obtained by Borrelli *et al.* [55] via model predictive control methods, present strong numerical disturbances.

8.3.3 Extensions of the result to classes of non linear vector fields

The STP was explicitly designed for linear switched systems. However there are several interesting non linear problems that may be considered. In this case the procedure would not work properly, but there might be some classes of non linear switched systems in which the procedure may still be applicable. Another theoretical development may be to approximate a non linear vector fields with a linear switched systems and then exploit the STP to design a suboptimal control law for the original non linear problem.

8.3.4 Extensions of the result to classes of uncertain switched systems

Currently we are also interested in developing algorithms to synthesize control laws for switched or more generally hybrid systems in presence of uncertainties. This might appear both in the model parameters, often affected by measurement disturbances or time variational as well as in the non deterministic occurrence of a discrete event. We believe that results in this field would be relevant to physical applications.

A

The linear quadratic optimal control

We will recall here the fundamental theoretical results of the *linear quadratic optimal* control. These results are taken from [44].

A.1 LQR feedback control law: the Riccati equation

Given an LTI system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

and a quadratic performance index

$$J(t) = \int_t^T (\boldsymbol{x}'(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau) + \boldsymbol{u}'(\tau)\boldsymbol{R}\boldsymbol{u}(\tau))\, d\tau, \tag{A.1}$$

$\boldsymbol{Q}, \boldsymbol{R}$ symmetric and semi definite positive, we seek for a suitable *gain matrix* $\boldsymbol{K}$, such that index $J(t)$ is minimized whenever

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t).$$

By defining

$$\boldsymbol{A}_c \triangleq \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K},$$

(closed loop system) we have

$$\boldsymbol{x}(\tau) = \bar{\boldsymbol{A}}_c(\tau - t)\boldsymbol{x}(t),$$

and substituting in (A.1) we obtain

$$J(t) = \boldsymbol{x}'(t)\int_t^T \left(\bar{\boldsymbol{A}}_c'(\tau - t)(\boldsymbol{Q} + \boldsymbol{K}'\boldsymbol{R}\boldsymbol{K})\bar{\boldsymbol{A}}_c(\tau - t)\right) d\tau \boldsymbol{x}(t) = \boldsymbol{x}'(t)\boldsymbol{Z}(t,T)\boldsymbol{x}(t), \tag{A.2}$$

$\boldsymbol{Z}(t,T)$ symmetric.

By definition of integral

$$\dot{J} = -\boldsymbol{x}'(\tau)\boldsymbol{L}\boldsymbol{x}(\tau)|_{\tau=t} = -\boldsymbol{x}'(t)\boldsymbol{L}\boldsymbol{x}(t)$$

with $\boldsymbol{L} = \boldsymbol{Q} + \boldsymbol{K}'\boldsymbol{R}\boldsymbol{K}$, and from direct derivation of (A.2) we have

$$\dot{J} = \dot{\boldsymbol{x}}'(t)\boldsymbol{Z}(t,T)\boldsymbol{x}(t) + \boldsymbol{x}'(t)\boldsymbol{Z}(t,T)\dot{\boldsymbol{x}}(t) + \boldsymbol{x}'(t)\dot{\boldsymbol{Z}}(t,T)\boldsymbol{x}(t)$$

and being $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_c \boldsymbol{x}(t)$ we obtain

$$\dot{J} = \boldsymbol{x}'(t) \left(\boldsymbol{A}_c' \boldsymbol{Z}(t,T) + \boldsymbol{Z}(t,T)\boldsymbol{A}_c + \dot{\boldsymbol{Z}}(t,T) \right) \boldsymbol{x}(t).$$

Hence

$$\boldsymbol{A}_c' \boldsymbol{Z}(t,T) + \boldsymbol{Z}(t,T)\boldsymbol{A}_c + \dot{\boldsymbol{Z}}(t,T) = -\boldsymbol{L}. \tag{A.3}$$

It can be proved that $J(t)$ is minimized provided that

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{Z}(t,T).$$

This particular gain, substituted in (A.3) yields (after trivial passages) to

$$\boldsymbol{A}\boldsymbol{Z}(t,T) + \boldsymbol{Z}(t,T)\boldsymbol{A} + \dot{\boldsymbol{Z}}(t,T) - \boldsymbol{Z}(t,T)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{Z}(t,T) = -\boldsymbol{Q} \tag{A.4}$$

known as the *Riccati* equation.

The Riccati equation, integrated with terminal condition $\boldsymbol{Z}(T,T) = \boldsymbol{0}$, gives the state feedback optimal control law $\boldsymbol{u}(t) = -\boldsymbol{K}(t)\boldsymbol{x}(t)$ for $t < T$.

A.2 LQR feedback control law: steady state solution

Consider the infinite time horizon, i.e.

$$J(t) = \lim_{T\to\infty} \int_t^T \left(\boldsymbol{x}'(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau) + \boldsymbol{u}'(\tau)\boldsymbol{R}\boldsymbol{u}(\tau) \right) d\tau = \lim_{T\to\infty} \boldsymbol{x}'(t)\boldsymbol{Z}(t,T)\boldsymbol{x}(t). \tag{A.5}$$

If the value of $J(t)$ is limited then $\boldsymbol{Z}(t,T)$ will converge to a constant matrix $\boldsymbol{Z}$, and eventually $\dot{\boldsymbol{Z}}(t,T) \to \boldsymbol{0}$.

In this case the equation (A.3) becomes

$$\boldsymbol{A}_c'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A}_c = -\boldsymbol{L}, \tag{A.6}$$

that yields to the *algebraic Riccati equation*, ARE,

$$\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{Z} = -\boldsymbol{Q} \tag{A.7}$$

whenever $\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{Z}$.

If the control action is chosen $\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t)$ then the function (A.5) is minimized.

Let us report here for completeness two sufficient conditions for the uniqueness of the solution, relevant in most of the applications.

Theorem A.1 *If the system is asymptotically stable (AS), then the ARE* (A.7) *has a unique positive definite solution that minimizes* (A.5). ■

Theorem A.2 *If the system* $(\boldsymbol{A}, \boldsymbol{B})$ *is controllable, and the couple* $(\boldsymbol{A}, \boldsymbol{C})$*, with* $\boldsymbol{C}$ *any orthogonal decomposition of* $\boldsymbol{Q}$*, i.e.,* $\boldsymbol{C}'\boldsymbol{C} = \boldsymbol{Q}$*, then the ARE* (A.7) *has a unique positive definite solution that minimizes* (A.5). ■

A.3 Evaluation of the LQ cost in the autonomous case

Let us now show a crucial result in the theoretical and implementative aspects of the STP.

Theorem A.3 (Value of the finite time horizon cost for autonomous system) *Given a LTI system of the form* $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$ *and a matrix* $\boldsymbol{Q} \geq \boldsymbol{0}$ *then the cost of a trajectory*

$$J(t) = \int_t^T \boldsymbol{x}'(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau)\; d\tau \tag{A.8}$$

is equal to

$$J(t) = \boldsymbol{x}'(t)\left[\boldsymbol{Z} - \bar{\boldsymbol{A}}'(T-t)\boldsymbol{Z}\bar{\boldsymbol{A}}(T-t)\right]\boldsymbol{x}(t),$$

whenever the Lyapunov equation $\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A} = -\boldsymbol{Q}$ *admits a unique solution.*

□

Proof. By the Lyapunov equation $\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A} = -\boldsymbol{Q}$,

$$J(t) = -\int_t^T \boldsymbol{x}'(\tau)(\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A})\boldsymbol{x}(\tau)\; d\tau$$

and being $\dot{\boldsymbol{x}}(\tau) = \boldsymbol{A}\boldsymbol{x}(\tau)$,

$$J(t) = -\int_t^T \dot{\boldsymbol{x}}'(\tau)\boldsymbol{Z}\boldsymbol{x}(\tau) + \boldsymbol{x}'(\tau)\boldsymbol{Z}\dot{\boldsymbol{x}}(\tau)\; d\tau.$$

Evidently the last equation can be expressed as

$$J(t) = -\int_t^T \frac{d\boldsymbol{x}'(\tau)\boldsymbol{Z}\boldsymbol{x}(\tau)}{d\tau}\; d\tau = -\boldsymbol{x}'(\tau)\boldsymbol{Z}\boldsymbol{x}(\tau)\Big|_t^T = \boldsymbol{x}'(t)\boldsymbol{Z}\boldsymbol{x}(t) - \boldsymbol{x}'(T)\boldsymbol{Z}\boldsymbol{x}(T), \tag{A.9}$$

hence

$$J(t) = \boldsymbol{x}'(t)\left[\boldsymbol{Z} - \bar{\boldsymbol{A}}'(T-t)\boldsymbol{Z}\bar{\boldsymbol{A}}(T-t)\right]\boldsymbol{x}(t),$$

because $\boldsymbol{x}(T) = \bar{\boldsymbol{A}}(T-t)\boldsymbol{x}(t)$. ■

Note that the cost is a *quadratic* function of the initial state and of the measure of the time interval.

Moreover we can state the following corollary.

Corollary A.1 (Value of the infinite time horizon cost for autonomous system). *Given a LTI system of the form* $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}$*, with* $\boldsymbol{A}$ *strictly Hurwitz, and a matrix* $\boldsymbol{Q} \geq \boldsymbol{0}$ *then the* infinite time horizon cost *of a trajectory*

$$J(t) = \lim_{T\to\infty}\int_t^T \boldsymbol{x}'(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau)\; d\tau \tag{A.10}$$

is equal to

$$J(t) = \boldsymbol{x}'(t)\boldsymbol{Z}\boldsymbol{x}(t),$$

where $\boldsymbol{Z} > \boldsymbol{0}$ *is the unique solution of the Lyapunov equation* $\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A} = -\boldsymbol{Q}$*.*

□

Proof. From Theorem A.1 the Lyapunov equation, the special case of equation (A.6) when $\boldsymbol{R} = \boldsymbol{B} = \boldsymbol{0}$, admits a unique positive definite solution. Furthermore from Theorem A.3 we have

$$J(t) = \lim_{T\to\infty}\int_t^T \boldsymbol{x}'(\tau)\boldsymbol{Q}\boldsymbol{x}(\tau)\,d\tau = \lim_{T\to\infty} \boldsymbol{x}'(t)\boldsymbol{Z}\boldsymbol{x}(t) - \boldsymbol{x}'(T)\boldsymbol{Z}\boldsymbol{x}(T) = \boldsymbol{x}'(t)\boldsymbol{Z}\boldsymbol{x}(t).$$

In fact, being $\boldsymbol{A}$ strictly Hurwitz,

$$\lim_{T\to\infty} \boldsymbol{x}(T) = \boldsymbol{0}.$$

■

Remarks:

- the Lyapunov equation (A.9) is the special case of equation (A.6) where $\boldsymbol{B} = \boldsymbol{R} = \boldsymbol{0}$;
- if $\boldsymbol{A}$ is Hurwitz the infinite time horizon cost is a quadratic function of the initial state and of the unique solution $\boldsymbol{Z} > \boldsymbol{0}$ of the Lyapunov equation;
- if $\boldsymbol{A}$ is non Hurwitz, then the *infinite time horizon* cost is infinite.
- if $\boldsymbol{A}$ is non Hurwitz, then the *finite time horizon* cost is finite, and it can be calculated from Theorem A.3, provided that the Lyapunov equation has a unique solution (but not necessarily positive definite)[1];
- in all cases the cost is a quadratic function of the initial state and, in the finite time horizon cases it is a function of the measure of the time interval.

[1]If all the real parts of the eigenvalues of $\boldsymbol{A}$ have the same sign then the Lyapunov equation admits a unique solution.

B

Computation of the performance index

The piecewise LQR problems presented in this thesis required to calculate the following integral:

$$\int_0^{\delta} \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt$$

subject to $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{f}$. It can be shown that for any initial state $\boldsymbol{x}_0 = \boldsymbol{x}(0)$, it holds

$$\int_0^{\delta} \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt = \boldsymbol{x}_0'\bar{\boldsymbol{Q}}(\delta)\boldsymbol{x}_0 + \bar{\boldsymbol{c}}(\delta)\boldsymbol{x}_0 + \bar{\alpha}(\delta)$$

where

$$\begin{aligned}
\bar{\boldsymbol{Q}}(\delta) &= \int_0^{\delta} \bar{\boldsymbol{A}}'(t)\ \boldsymbol{Q}\ \bar{\boldsymbol{A}}(t)dt, \\
\bar{\boldsymbol{c}}(\delta) &= 2\boldsymbol{f}'\ \int_0^{\delta}\left(\int_0^{t} \bar{\boldsymbol{A}}'(\tau)d\tau\right)\boldsymbol{Q}\ \bar{\boldsymbol{A}}(t)dt, \\
\bar{\alpha}(\delta) &= \boldsymbol{f}'\ \left[\int_0^{\delta}\left(\int_0^{t} \bar{\boldsymbol{A}}'(\tau)d\tau\right)\boldsymbol{Q}\left(\int_0^{t} \bar{\boldsymbol{A}}(t)d\tau\right)dt\right]\ \boldsymbol{f}
\end{aligned}$$

In general cases it is not easy to provide analytical expressions for $\bar{\boldsymbol{Q}}(\delta)$, $\bar{\boldsymbol{c}}(\delta)$, and $\bar{\alpha}(\delta)$, thus numerical integration is needed. On the contrary, under appropriate assumptions on $\boldsymbol{A}$ and $\boldsymbol{f}$, these analytical expressions can be easily determined. As an example, let us consider the following two cases.

- Assume $\boldsymbol{A}$ is strictly Hurwitz and $\boldsymbol{f} = \boldsymbol{0}$. In such a case

$$\begin{aligned}
&\bar{\boldsymbol{Q}}(\delta) = \boldsymbol{Z} - \bar{\boldsymbol{A}}'(\delta)\boldsymbol{Z}\bar{\boldsymbol{A}}(\delta), \\
&\bar{\boldsymbol{c}}(\delta) = \boldsymbol{0}, \\
&\bar{\alpha}(\delta) = 0,
\end{aligned}$$

  where $\boldsymbol{Z}$ is the unique solution of the Lyapunov equation $\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A} = -\boldsymbol{Q}$. The same computation is valid when the eigenvalues of $\boldsymbol{A}$ are all unstable.
- Assume that $\boldsymbol{A}$ is diagonalizable. In such a case, $\boldsymbol{A} = \boldsymbol{T}^{-1}\boldsymbol{\Lambda}\boldsymbol{T}$, where $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$ and λ_j, $j = 1, \ldots, n$ are the eigenvalues of $\boldsymbol{A}$. We obtain:

$$\begin{aligned}
\bar{\boldsymbol{Q}}(\delta) &= \boldsymbol{T}'\left(\int_0^{\delta} \bar{\boldsymbol{\Lambda}}(t)(\boldsymbol{T}^{-1})'\ \boldsymbol{Q}\ \boldsymbol{T}^{-1}\bar{\boldsymbol{\Lambda}}(t)dt\right)\boldsymbol{T}, \\
\bar{\boldsymbol{c}}(\delta) &= 2\boldsymbol{f}'\ \boldsymbol{T}'\left[\int_0^{\delta}\left(\int_0^{t} \bar{\boldsymbol{\Lambda}}(\tau)d\tau\right)(\boldsymbol{T}^{-1})'\ Q\ \boldsymbol{T}^{-1}\bar{\boldsymbol{\Lambda}}(t)dt\right]\boldsymbol{T}, \\
\bar{\alpha}(\delta) &= \boldsymbol{f}'\ \boldsymbol{T}'\left[\int_0^{\delta}\left(\int_0^{t} \bar{\boldsymbol{\Lambda}}(\tau)d\tau\right)(\boldsymbol{T}^{-1})'\ \boldsymbol{Q}\ \boldsymbol{T}^{-1}\left(\int_0^{t} \bar{\boldsymbol{\Lambda}}(\tau)d\tau\right)dt\right]\boldsymbol{T}\ \boldsymbol{f},
\end{aligned}$$

and it is straightforward to symbolically compute the integrals exploiting the simple form of the exponential diagonal matrix.

C

Issues on state space discretization

C.1 Discretization of n dimensional unitary semisphere Σ_n

The main computational effort in the construction of the switching tables is the discretization of the state space. There are several ways to discretize the state space and it is important to identify the best one case by case.

An unappropriate discretization of the state space can provoke explosions in the computational time and in terms of needed resources. Note that the most intuitive one (i.e., the cartesian grid) is not always the better idea. In some cases a polar discretization (evidently when the homogeneous property holds) is more suitable.

We provide in the following the first step to construct the relation between polar and cartesian system in $\mathbb{R}^n$. The n polar coordinates are composed of 1 radius ρ_n and $n-1$ angles $\theta_2, \ldots, \theta_n$. Given a point $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]'$, such relation is

$$\begin{cases} x_n = \rho_n \sin(\theta_n) \\ x_{n-1} = \rho_{n-1} \sin(\theta_{n-1}) \\ \vdots \\ x_3 = \rho_3 \sin(\theta_3) \\ x_2 = \rho_2 \sin(\theta_2) \\ x_1 = \rho_2 \cos(\theta_2) \end{cases}$$

where $\rho_n = \|\boldsymbol{x}\|$, $\rho_i = \rho_{i+1} \cos(\theta_i)$ for $i = n-1, \ldots, 2$. To describe $\mathbb{R}^n$, variables must range in: $\rho_n \in [0, +\infty)$, $\theta_2 \in [0, 2\pi)$, and $\theta_3, \ldots, \theta_n \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. To describe Σ_n we choose $\rho_n = 1$, $\theta_2 \in [0, 2\pi)$ $\theta_3, \ldots, \theta_{n-1} \in [-\frac{\pi}{2}, \frac{\pi}{2})$, and $\theta_n \in [0, \frac{\pi}{2}]$.

Note that a uniform discretization for each angle brings to areas with high density of points (think of the grid on the earth surface at the poles), as it can be seen in Figure C.1. This aspect is useless: there is no need at all to increase the density around a point. In $\mathbb{R}^3$ one may suggest to take the vertexes of a regular polyhedron, such as hexahedron, octahedron, dodecahedron, icosahedron, but these ones at most contain 20 points. Moreover there is no further method for higher dimensions. Thus we provide an approximation, named as *constant arc length*.

An approximately equal spaced grid can be obtained with a reduced number of points using the following criterion, that provides constant arc length.

As an example, assume $n = 4$. Let us call $\theta_4 = \xi$, $\theta_3 = \varphi$ and $\theta_2 = \vartheta$.

1. Define nominal values of discretization N_ϑ, N_φ, N_ξ; since $\vartheta \in [0, 2\pi)$, $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2})$ and $\xi \in [0, \frac{\pi}{2}]$ we choose $N_\vartheta = 2N_\varphi = 4N_\xi$ proportional to the respective range of each variable;

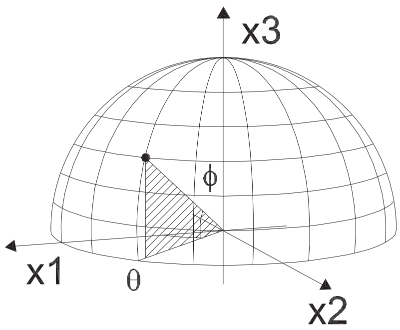


Fig. C.1. *Uniform discretization of Σ_3. Observe that the density of point is not uniform when approaching the North pole.*

2. discretize ξ uniformly, i.e., $\xi_i = i\frac{\pi}{2N_\xi}$, $i = 0, \cdots, N_\xi$;
3. denoted by round$(\cdot)$ a function that approximates to the closest integer, for every ξ_i define $\bar{N}_\varphi = \text{round}(N_\varphi \cos(\xi_i))$ and discretize φ uniformly, i.e., $\varphi_j = -\frac{\pi}{2} + j\frac{\pi}{\bar{N}_\varphi}$, $j = 0, \cdots, \bar{N}_\varphi - 1$;
4. for every ξ_i and φ_j define $\bar{N}_\vartheta = \text{round}(N_\vartheta \cos(\xi_i)\cos(\varphi_j))$ and discretize ϑ uniformly, i.e., $\vartheta_k = k\frac{2\pi}{\bar{N}_\vartheta}$, $k = 0, \cdots, \bar{N}_\vartheta - 1$;

With such criteria we obtain a grid of $N \cong \frac{N_\xi N_\varphi N_\vartheta}{3}$.

A geometrical representation can be given in $\mathbb{R}^3$. In this case the *zenith* angle φ is divided into N_φ uniform samples. For each value of φ we obtain a *disc* (parallel to the equator), whose radius is evidently $R = \sin(\varphi)$. We divide the equator disc ($R = 1$) into N_ϑ sectors and then all the others are divided in such a way that the arc length is constant and equal to the biggest. The number of points on each disc will then decrease as the latitude increases.

In Figure C.2 we reported, as an example, the aerial view of the discretization along each parallel disc in a case where the zenith interval is divided in 3 sectors (0, $\frac{\pi}{6}$, $\frac{\pi}{3}$, respectively).

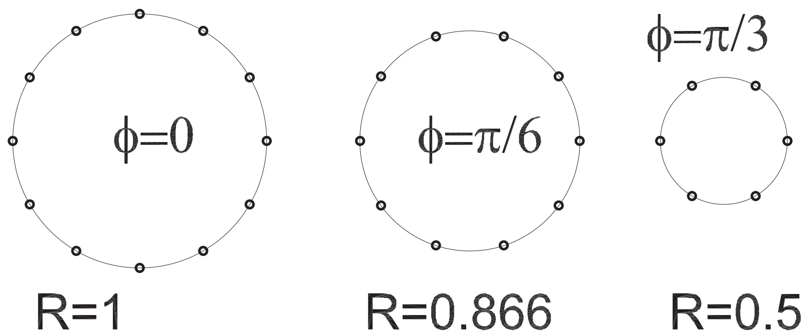


Fig. C.2. *Discretization of the parallel discs of Σ_3 according to the criteria of constant arc length.*

C.2 Interpolation of the value of the cost

The algorithm of the tables construction is based on the calculation of a function T_k, k is the number of available switches, in each point of a space discretization grid $\mathcal{D}$. This value is obtained by the sum of the cost with the current dynamics and the residual optimal cost T^*_{k-1} in a point $\boldsymbol{x}$ that in general does not belong to the discretization set. Unfortunately only $T^*_{k-1}(\boldsymbol{x}_i)$ is available, thus we approximate $T^*_{k-1}(\boldsymbol{x})$ with the values of the points of $\mathcal{D}$ around $\boldsymbol{x}$.

The simplest approach is the nearest neighbor policy, i.e., $T^*_{k-1}(\boldsymbol{x}) \simeq T^*_{k-1}(\boldsymbol{x}_i)$, where $\boldsymbol{x}_i$ is the nearest (in the Euclidean sense) point to $\boldsymbol{x}$. We describe here another approach, that was successfully used in the STP of the fourth dimensional case. It should be remarked that both approaches are valid only if the functions $T^*_{k-1}(\boldsymbol{x})$ are continuous.

To contain the level of discretization and to guarantee an acceptable accuracy on $T^*_{k-1}(\boldsymbol{x})$, an interpolation criteria is required. When a point $\boldsymbol{x}$ isn't in the grid, we consider the residual optimal cost values T^*_{k-1} in H points of the grid around $\boldsymbol{x}$, namely $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_H$. For example, we report a picture of the discretization $\mathcal{D}$ on Σ_2 (Figure C.3) and Σ_3 (Figure C.4).

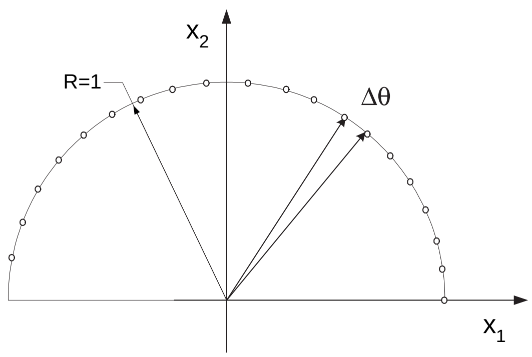


Fig. C.3. *Spherical discretization pattern in* $\mathbb{R}^2$.

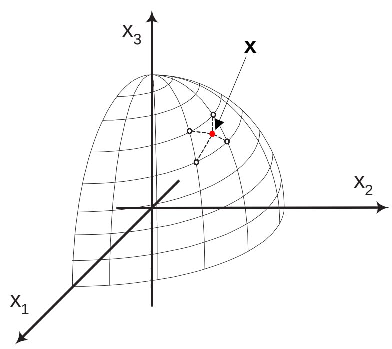


Fig. C.4. *Neighborhood of the point* $\boldsymbol{x}$ *on* Σ_3. *Note that the number of points of the closest neighborhood is at most equal to 4.*

Let us observe that in general, in Σ_n, the number of points H around $\boldsymbol{x}$ is equal to 2^{n-1}. If a cartesian grid is used, this number is equal to 2^n.

In this framework let us define $d_i = \|\boldsymbol{x} - \boldsymbol{x}_i\|^{-1},\ i = 1 \dots H$. An estimation of the value of $T^*_{k-1}(\boldsymbol{x})$ can be obtained by the average of the values of $T^*_{k-1}(\boldsymbol{x}_i)$, $i = 1, \dots, H$, weighted with the reciprocal distance from the given point $\boldsymbol{x}$.

$$T^*_{k-1}(\boldsymbol{x}) = \frac{\displaystyle\sum_{i=1}^{H} d_i T^*_{k-1}(\boldsymbol{x}_i)}{\displaystyle\sum_{i=1}^{H} d_i}.$$

From a geometrical point of view, this is equivalent to substitute the function with the hyperplane (named as *spline*) that passes in all the points $(\boldsymbol{x}_i, T_{k-1}(\boldsymbol{x}_i))$, $i = 1, \dots, H$. This approximation was considered better than the nearest neighbor policy. In fact it can be proved [27] that the error on the estimation of the function T, in a one dimensional discretization, is proportional to Δ^2, where Δ is the parameter of the grid. As a disadvantage, this interpolation introduces more calculations. Note that in the one dimensional manifold (like on Σ_2) this is a "linear interpolation", as it can be seen in Figure C.5.

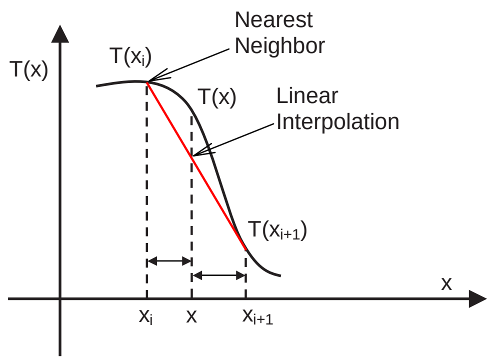


Fig. C.5. *Linear interpolation and nearest neighbor method for the estimation of the value of the residual cost T in a point that does not belong to the discretization. This one dimensional case also holds in $\mathbb{R}^2$ if the sampling along Σ_2 is sufficient.*

D

The l-complete approximation

We provide here the approach, developed by *Raisch et al.* that converts the specifications[1] on the dynamical behavior of the outputs signals of a hybrid automaton HA into constraints on the state space.

These constraints can be considered invariants, i.e., $inv_i \subseteq \mathbb{R}^n$ where the state space $\boldsymbol{x}$ must remain when evolving in location i.

Some constraints on the admissible state trajectory can be expressed via a discrete automaton that is based on a partition of the state space $X \subset \mathbb{R}^n$. To each element of this partition we associate an output signal Y_d. A discrete automaton SP_Y is used to restrict the set of the admissible sequences of output signals.

In the following we call *safety constrains* the constraints that originate from the structure of the HA, and the constraints on the output sequences given by the discrete automaton SP_Y. An example of this specification will be given in Section 5.5.2.

The low-level step consists in the definition of the invariant sets that guarantee that the discrete output sequences obey an imposed specification modelled by a discrete event automaton SP_Y. The resulting hybrid automaton does not posses blocking states, i.e. it guarantees the liveness of the overall system. The resulting system is considered as an input for the top-level optimization procedure.

The hybrid plant model is converted to a purely discrete one via the l-complete approximation approach [95, 83]. Subsequently, Ramadge and Wonham's supervisory control theory [96] is implemented to synthesize a least restrictive supervisor. If the hybrid plant is interpreted as a hybrid automaton, attaching the supervisor is equivalent to adding invariants to this automaton.

D.1 Ordered set of discrete abstractions

Let us now restrict our attention to the class of switched affine systems that evolve in discrete time and generate discrete-valued outputs. The sampling interval is denoted by Δt. Furthermore we assume that all processes in our overall system are synchronized (i.e. they operate on a common time scale). The model of the plant is then described by the set of time-invariant difference equations

$$\begin{aligned} \boldsymbol{x}(t_{k+1}) &= f_{\psi(t_k)}(\boldsymbol{x}(t_k)), \\ y_d(t_{k+1}) &= q_y(\boldsymbol{x}(t_{k+1})) \end{aligned} \tag{D.1}$$

[1] Some typical specifications on the dynamical behavior of a HA are for instance the *safety* and the *liveness*.

where $k \in \mathbb{N}_0$ is the time index, $t_k \in \mathbb{T} = \{k\Delta t\}_{k\in\mathbb{N}_{\not{0}}}$; $\boldsymbol{x} \in X \subset \mathbb{R}^n$, $\{f_{\psi^{(i)}}(\boldsymbol{x}) = \boldsymbol{A}_i\boldsymbol{x} + \boldsymbol{b}_i \,:\, \psi^{(i)} \in \Psi\}$ is a family of affine state transition maps from X into $\mathbb{R}^n$ that is parameterized by some finite index set $\Psi = \{\psi^{(1)}, \ldots, \psi^{(\alpha)}\}$; $\psi : \mathbb{T} \to \Psi$ is a switching signal which can be interpreted as a discrete control input.

$y_d \in Y_d^{\mathbb{T}}$ is a discrete-valued measurement signal. The set of output symbols, Y_d, is assumed to be finite: $Y_d = \{y_d^{(1)}, \ldots, y_d^{(\beta)}\}$, and $q_y : X \to Y_d$ is the output map. Without loss of generality, the latter is supposed to be surjective (*onto*). The output map partitions the state space into a set of disjoint subsets $Y^{(i)} \subset X,\ i = 1, \ldots, \beta$, i.e.

$$\bigcup_{i=1}^{\beta} Y^{(i)} = X,$$

$$Y^{(i)} \cap Y^{(j)} = \emptyset \quad \forall i \neq j\,.$$

To implement supervisory control theory, the hybrid plant model is approximated by a purely discrete one. This is done using the method of l-complete approximation [95, 83], which is described in the following paragraphs.

Denote the behavior of the hybrid plant model by $\mathcal{B}_{plant}$, i.e. $\mathcal{B}_{plant} \subseteq (\Psi \times Y_d)^{\mathbb{T}}$ is the set of all pairs of (discrete valued) input/output signals $w = (\psi, y_d)$ that (D.1) admits. In general, a time-invariant system with behavior $\mathcal{B}$ is called l-complete if $w \in \mathcal{B} \Leftrightarrow \sigma^t w|_{[t_0,t_l]} \in \mathcal{B}|_{[t_0,t_l]}\ \forall t \in \mathbb{T}$, where σ is the backward shift operator and $w|_{[t_0,t_l]}$ denotes the restriction of the signal w to the domain $[t_0, t_l]$ [116]. Hence, for l-complete systems we can decide whether a signal belongs to the system behavior by looking at intervals of length l. Clearly, an l-complete system can be represented by a difference equation in its external variables with lag l. The hybrid plant model (D.1) is, except for trivial cases, not l-complete. For such systems, the notion of *strongest l-complete approximation* has been introduced in [83]: a time-invariant dynamical system with behavior $\mathcal{B}_l$ is called strongest l-complete approximation for $\mathcal{B}_{plant}$ if

(i) $\mathcal{B}_l \supseteq \mathcal{B}_{plant}$,
(ii) $\mathcal{B}_l$ is l-complete,
(iii) $\mathcal{B}_l \subseteq \tilde{\mathcal{B}}_l$ for any other l-complete $\tilde{\mathcal{B}}_l \supseteq \mathcal{B}_{plant}$,

i.e. if it is the "smallest" l-complete behavior containing $\mathcal{B}_{plant}$. Obviously, $\mathcal{B}_l \supseteq \mathcal{B}_{l+1}\ \forall l \in \mathbb{N}$, hence the proposed approximation procedure may generate an ordered set of abstractions. Clearly, $w \in \mathcal{B}_l \Leftrightarrow w|_{[t_0,t_l]} \in \mathcal{B}_{plant}|_{[t_0,t_l]}$. For $w|_{[t_0,t_l]} = (\psi^{i_0}, \ldots, \psi^{i_l}, y_d^{i_0}, \ldots, y_d^{i_l})$ this is equivalent to

$$\begin{aligned} &f_{\psi^{(i_{l-1})}}\left(\ldots f_{\psi^{(i_1)}}\left(f_{\psi^{(i_0)}}\left(q_y^{-1}(y_d^{(i_0)})\right) \cap \left(q_y^{-1}(y_d^{(i_1)})\right)\right)\right. \\ &\left.\ldots (q_y^{-1}(y_d^{i_{(l-1)}})\right) \cap q_y^{-1}(y_d^{(i_l)}) := X(w|_{[t_0,t_l]}) \neq \emptyset. \end{aligned} \tag{D.2}$$

Note that for a given string $w|_{[t_0,t_l]}$, $X(w|_{[t_0,t_l]})$ represents the set of possible values for the continuous state variable $\boldsymbol{x}(t_l)$ and that (D.2) does not depend on $\psi^{(i_l)}$. For switched affine systems evolving on discrete time $\mathbb{T}$, (D.2) can be checked *exactly*. For switched affine system evolving on continuous time and special classes of nonlinear systems, $X(w|_{[t_0,t_l]})$ can be safely over approximated, hence (D.2) can be checked "conservatively" (e.g. [43, 84]). This will still lead to an l-complete approximation but, in general, not a strongest l-complete approximation.

As both input and output signal evolve on finite sets, Ψ and Y_d, $\mathcal{B}_l$ can be realized by a (nondeterministic) finite automaton. In [95, 83], a particularly intuitive realization is suggested, where the approximation state variable stores information on past values of ψ and y_d. More precisely, the automaton state set can be defined as

$$X_d := \bigcup_{j=0}^{l-1} X_{d_j}$$

where

$$X_{d_0} = Y_d$$

and

$$X_{dj} = \{(\psi^{(i_0)}, \ldots, \psi^{(i_{j-1})}, y_d^{(i_0)}, \ldots, y_d^{(i_j)})\}$$

such that

$$\exists \psi^{(i_j)} \in \Psi : (\psi^{(i_0)}, \ldots, \psi^{(i_j)}, y_d^{(i_0)}, \ldots, y_d^{(i_j)}) \in \mathcal{B}_l|_{[t_0, t_j]}.$$

As the states $\boldsymbol{x}_d^{(j)}$ of the approximation realization are strings of input and output symbols, we can associate $\boldsymbol{x}_d^{(j)}$ with a set of continuous states, $X(\boldsymbol{x}_d^{(j)})$, in completely the same way as in (D.2).

Note that the transition function $\delta : X_d \times \Psi \to 2^{X_d}$ follows immediately from $\mathcal{B}_l$ and that we can associate $y_d^{(i_j)}$ as the unique output for each discrete state $\boldsymbol{x}_d^{(j)} \in X_d$. The resulting (non deterministic) Moore-automaton $M_l = (X_d, \Psi, Y_d, \delta, \mu, X_{d_0})$ with state set X_d, input set Ψ, output set Y_d, transition function δ, output function μ, and initial state set X_{d_0} is then a realization of $\mathcal{B}_l$. Note that the state of M_l is instantly deducible from observed variables.

To recover the framework of supervisory control theory [96] as closely as possible, we finally convert M_l into an equivalent automaton without outputs, $G_l = (\tilde{X}_d, \Psi \times Y_d, \tilde{\delta}, \tilde{X}_{d_0})$, where Ψ represents the set of controllable events and Y_d the set of uncontrollable events.

D.2 Specification and supervisor design

Safety requirements can often be formalized as a set of acceptable pairs of input/output signals. In many applications we have independent specification behaviors for both inputs and outputs, $\mathcal{B}_\Psi \subseteq \Psi^{\mathbb{T}}$, $\mathcal{B}_{Y_d} \subseteq Y_d^{\mathbb{T}}$, which are assumed to be m_Ψ and m_Y-complete. They can hence be realized by finite automata $SP_\Psi = (S_\Psi, \Psi, \delta_\Psi, S_{\Psi 0})$ and $SP_Y = (S_Y, Y_d, \delta_Y, S_{Y0})$.

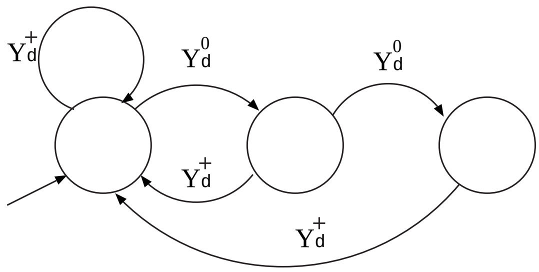


Fig. D.1. *Specification for the outputs*

The overall specification is then easily obtained by forming the shuffle product of SP_Ψ and SP_Y (e.g. [22]),

$$SP = (S, \Psi \cup Y_d, \delta_{SP}, S_0)$$

where $S = S_\Psi \times S_Y$, $S_0 = S_{\Psi 0} \times S_{Y0}$. SP realizes the concurrent behavior of SP_Ψ and SP_Y.

Given an approximating automaton G_l and a specification automaton SP, supervisory control theory checks, whether there exists a nonblocking supervisor and, if the answer is affirmative, provides a least restrictive supervisor SUP via "trimming" of the product of G_l and SP. Hence the state set of the supervisor, X_{SUP}, is a subset of $\tilde{X} \times S$.

The functioning of the resulting supervisor is very simple. At time t_k it "receives" a measurement symbol which triggers a state transition. In its new state $\boldsymbol{x}_{sup}^{(j)}$, it enables a subset $\Gamma(\boldsymbol{x}_{sup}^{(j)}) \subseteq \Psi$ and waits for the next feedback from the plant. As shown in [83], the supervisor will enforce the specifications not only for the approximation, but also for the underlying hybrid plant model (D.1).

In the following, we will be interested in the special case of *quasi-static* specifications. To explain this notion, let $p_{app} : X_{SUP} \rightarrow \tilde{X}$ denote the projection of $X_{SUP} \subseteq \tilde{X} \times S$ onto its first component. If p_{app} is injective, the specification automaton is called quasi-static with respect to the approximation automaton G_l.

Proposition D.1. *S is quasi-static with respect to G_l if*

$$l \geq max(m_\Psi, m_Y). \tag{D.3}$$

D.3 Closed loop model

We now interprete the hybrid plant model (D.1) as a hybrid automaton with locations $\psi^{(1)}, \ldots, \psi^{(\alpha)}$ and attach the supervisor SUP. For the case of quasi-static specifications, each supervisor state $p_{app}(\boldsymbol{x}_{sup}^{(i)})$ corresponds exactly to a state $\tilde{\boldsymbol{x}}_d^{(i)} = p_{app}(\boldsymbol{x}_{sup}^{(i)})$ of the approximating automaton, which, in turn, can be associated with a set $X(\tilde{\boldsymbol{x}}_d^{(i)}) = X(p_{app}(\boldsymbol{x}_{sup}^{(i)}))$.

Attaching the supervisor to the hybrid plant automaton therefore boils down to adding invariants to each location

$$inv(\psi^{(j)}) = \bigcup_{\substack{i,\, p_{app}(\boldsymbol{x}_{sup}^{(i)}) \in \tilde{X}_{d_{l-1}} \\ \psi^{(j)} \in \Gamma(\boldsymbol{x}_s^{(i)})}} X(p_{app}(\boldsymbol{x}_{sup}^{(i)})),$$

where $\tilde{X}_{d_{l-1}} = X_{d_{l-1}}$. Union of all invariants forms the refined, safe state space that contains only safe points, i.e. points for which exists at least one sequence of control symbols such that the resulted behavior satisfies the specification.

The resulting hybrid automaton is guaranteed to obey the specification but retains degrees of freedom, which can be used in a separate optimal control layer.

E

Software user-guide

This brief appendix aims to present some of the software packages, implemented during this doctoral period at the University of Cagliari. This software was used to obtain part of the results described in this thesis.

The software is implemented in MATLAB, version 6.0, Release 12. It can be downloaded from

http://www.diee.unica.it/∼dcorona/thesis.html.

With the package *STP2_corona.zip* you can:

1. Construct the *switching tables* for a $\mathbb{R}^2$ hybrid automata, modelled in this thesis, that provide the feedback switching law of the considered optimal control and stability problem (Chapters 3,4,7) – Function ***regions.m***.
2. Visualize graphically these tables – Function ***plot_tables.m***.
3. Obtain the number of switches $\bar{N}$ (Chapters 6,7) where the convergence is achieved. The convergence criterion has not been automatized, hence this decision is up to the user, by direct visualization of the tables.
4. Simulate the use of the switching tables, in both cases of N *finite* and *infinite* – Function ***simulation.m***.
5. In addition an efficient general function that calculates the LQR cost from a given initial point and for a given time interval is proposed – Function ***index.m***.

Additionally with the package *STP4_corona.zip* you can implement the STP in $\mathbb{R}^4$. We decided to provide separate files because of the more complex data structure in $\mathbb{R}^4$ compared to $\mathbb{R}^2$. The STP4 software guide is in Section E.5.

E.1 Function *regions.m*

This software is for two dimensional use, i.e., the state space $\boldsymbol{x} \in \mathbb{R}^2$. This implies that matrices $\boldsymbol{A}$, $\boldsymbol{Q}$ and $Jump$ are matrices of class $\mathbb{R}^2$.

The function receives in input the following data:

1. The *hybrid automata* (dynamics, jumps, edges and minimum permanence time);
2. The *optimal control problem* (weight matrices and number of available switches);
3. The discretization data (Time and space discretization);
4. Eventually, to be used when you need to keep increasing N to meet convergence, the previously calculated table.

Before proceeding further with the help of the software we provide the notion of MATLAB *matrix array*.

A matrix array is an array whose elements are matrices. A set of matrices $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3$ can be collected in a unique data structure

$>>$ $\boldsymbol{A}$= $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3\}$

Each element is recalled with the typing of the required index within brackets $\{\}$. For example, the command

$>>$ $\boldsymbol{A}\{1\}$

shows matrix elements of $\boldsymbol{A}_1$. In general also matrices of matrices can be represented. For example

$>>$ $\boldsymbol{M} = \{\boldsymbol{M}_{1,1}, \boldsymbol{M}_{1,2}; \boldsymbol{M}_{2,1}, \boldsymbol{M}_{2,2}\}$

where $\boldsymbol{M}_{i,j}$, $i, j = 1, 2$ are matrices. The command

$>>$ $\boldsymbol{M}\{2, 2\}$

shows matrix $\boldsymbol{M}_{2,2}$.

E.1.1 Initial use: no tables are calculated

At the MATLAB prompt type:

$>>$ $Table = regions(A, Q, G, Jump, d\_min, N_\vartheta, \tau_M, N_t, N)$

where

INPUT

1. $\boldsymbol{A}$ is a *matrix array* $1 \times s$, s is the number of locations, that contains all dynamics of the automaton. Note that the software, in this preliminary version, *does not have internal checks* on the stability of each element, hence make sure that at least one matrix of the array $\boldsymbol{A}$ is stable.
2. $\boldsymbol{Q}$ is a *matrix array* $1 \times s$, that contains all weights in the LQR cost. Note that *all* Q must be *non negative* definite.
3. G is a matrix $s \times s$ of edges. If there exists an arc from location i to location j then set the element $G(i, j) = j$, else set $G(i, j) = 0$. Note that in this context there are no self-loops, hence set $G(i, i) = 0$.
4. $Jump$ is a ***matrix array*** $s \times s$ of ***switching jump matrices*** $Jump\{i, j\}$. In case of continuous state set $Jump\{i, j\} = eye(2)$.
5. d_min is a row vector that contains the minimum permanence times in each location. Note that each element of this vector must be positive or null.
6. N_ϑ is the number of *equally* spaced points on the unitary semisphere. Typical values are 51, 71, 101, 151. You might prefer odd values in order to represent the point on the x_2 axis. Point 1 corresponds to $[1, 0]'$, point $\frac{N_\vartheta+1}{2}$ corresponds to $[0, 1]'$, point N_ϑ to $[-1, 0]'$.
7. τ_M maximum time exploration, in theory infinite. Note that an appropriate value should be 4 or 5 times the slowest time constants of the stable matrices $\boldsymbol{A}_i$

8. N_t number of points in the time exploration. Note that this number should be coordinated with τ_M in order to obtain a fine enough time step $dt = \frac{\tau_M}{N_t}$. Unless $\boldsymbol{A}$'s have high frequencies modes, values between $dt = 10^{-2}, 10^{-3}$ are acceptable.
9. N number of allowed switches.

Remark E.1 (Stabilization usage) *If you are aiming to use the software to design a table that* stabilizes *a switched system where all dynamics are unstable, make sure the following:*

1. *Insert in the matrix array $\boldsymbol{A}$ a stable dynamics $\boldsymbol{A}\{s+1\}$, typically, a good choice is to take one of the unstable dynamics (with all positive real parts eigenvalues) with opposite sign. Rotating stable dynamics are observed to behave better.*
2. *Insert in the matrix array $\boldsymbol{Q}$ an extremely expensive positive definite matrix. Good examples*[1] *are*

   $$>>\ \boldsymbol{Q}\{s+1\} = 1e10 * eye(2).$$

3. *The matrix G must be complete, i.e., all its terms out of diagonal are non null. If a switched system (augmented) of three locations is considered then*

   $$G = \begin{bmatrix} 0 & 2 & 3 \\ 1 & 0 & 3 \\ 1 & 2 & 0 \end{bmatrix}.$$

4. *All Jump matrices are the identity.*

■

OUTPUT

The calculations may be long, hence a sequence of dots are visualized to confirm that the software is effectively running. Each *carriage return* in the line of dots indicates that a new location i is being processed, hence the table $\mathcal{C}_k^i$ is in progress of construction.

In the end the lines of dots will be $(N+1)\times s$, where s is the number of locations.

The data, residual cost and color from each point of the unitary semisphere and for each dynamics, is collected in a matrix array $Table$. The data structure requires further explanations.

- $Table\{k\}$, $k = 1, \ldots, N+1$, contains the information relative to $k-1$ remaining switches. For example $Table\{3\}$ contains the residual cost and the color from each point of the semisphere, when 2 switches are left. Hence $Table\{N+1\}$ is the last calculated one, for N available switches. The command

  $$>>\ Table$$

  lists this matrix array.
- Each element of $Table$, $Table\{k\}$, is a matrix of s rows and $2N_\vartheta$ columns. In other words each row i, i.e., the current dynamics, is a vector of $2N_\vartheta$ elements. This should be seen as a list of N_ϑ *couples*, (a couple per point of the discretization) where the *odd* element is the residual cost, and the even *element* is an integer $j = 1, \ldots, s$ that indicates the switching strategy.

[1] MATLAB exponential number notation.

Example E.1 *Assume that* $N_\vartheta = 101$, $N = 10$ *and* $s = 4$. *Then the element*

$$>> \ Table\{k\}(i, 2h-1)$$

contains the residual cost when $k-1$ *switches are left, from location* i *and from point indexed* $h = 1, \ldots, N_\vartheta$ *on the semisphere, and*

$$>> \ Table\{k\}(i, 2h)$$

contains the location index where it is optimal to switch. ■

Remark E.2 *It is a good habit, when the function* **region.m** *has terminated, to save the data by running the MATLAB command*

$$>> \ save \ < file\_name > \ Table, A, Q, G, Jump, d\_min, N_\vartheta, \tau_M, N_t, N$$

that stores the input and the calculated data in a file called $< file\_name >$. *To load this file type the MATLAB command*

$$>> \ load \ < file\_name >$$

from the belonging directory.

■

E.1.2 Iterative use: a set of tables are available

Use this modality when you want to keep calculating tables for increasing values of N. More specifically: the program has constructed already N tables, Tab. Probably you want to calculate $M = N + \tilde{N}$, without losing the previous effort.

Hence, at the MATLAB prompt type:

$$>> \ Table = regions(A, Q, G, Jump, d\_min, N_\vartheta, \tau_M, N_t, M, Tab)$$

where:

INPUT

All elements *must* be the same as in the previous section, except the last two.

1. M: new number of allowed switches, greater then the previous one.
2. Tab is the set of tables previously calculated.

OUTPUT

See previous section.

E.2 Function *plot_tables.m*

This function prepares the data to plot the switching tables in $\mathbb{R}^2$.

We assume that the program region has been executed and a set of tables has been calculated. At the MATLAB prompt type

$$>> \ [X, Y, T] = plot\_table(Table);$$

The data stored in X, Y, T contains the information for the input of the MATLAB built-in function ***pcolor***.

If you want to visualize the table $\mathcal{C}^i_{k-1}$ ($k-1$ switches are left from location i) then type

$$>> \ pcolor([X\{k\}; 0\ 0; 0\ 0], [Y\{k\}; 0\ 0; 0\ 0], [T\{k,i\}; 2\ 2; 1\ 1])$$

if the automaton has 2 locations,

$$>> \ pcolor([X\{k\}; 0\ 0; 0\ 0; 0\ 0], [Y\{k\}; 0\ 0; 0\ 0; 0\ 0], [T\{k,i\}; 3\ 3; 2\ 2; 1\ 1])$$

if the automaton has 3 locations,

$$>> \ pcolor([X\{k\}; 0\ 0; \ldots; 0\ 0], [Y\{k\}; 0\ 0; \ldots; 0\ 0], [T\{k,i\}; s\ s; \ldots; 1\ 1])$$

if the automaton has s locations.

Remark E.3 (Color mapping) *The color associated to location i is the color of table $Table\{1\}(i,:)$ and you may visualize it with command*

$$>> \ pcolor([X\{1\}; 0\ 0; \ldots; 0\ 0], [Y\{1\}; 0\ 0; \ldots; 0\ 0], [T\{1,i\}; s\ s; \ldots; 1\ 1]).$$

This command produces a disk with the color associated to the $i-th$ location. ■

E.3 Function *simulation.m*

From a given initial hybrid state $(\boldsymbol{x}, i)$ it is possible to use the tables, calculated by function ***regions.m***, to calculate the optimal switching intervals, the optimal switching sequence, the optimal cost and the optimal trajectory.

At the MATLAB prompt type

$$>> \ [T, I, J, X] = simulation(A, Q, Jump, d\_min, \tau_M, N_t, \boldsymbol{x}, i, Table, op, th);$$

where all input variables have been defined in Section E.1.1, except for

1. $Table$ is the output of program ***regions.m***;
2. op, a parameter so defined:
   - $op = 0$, *finite* number of switches, uses all tables;
   - $op = 1$, *infinite* number of switches, uses only the last calculated tables.
3. th is a terminating criterion on the norm of the continuous state $\boldsymbol{x}$, usually values $th = 10^{-3}, 10^{-4}$ are acceptable.

OUTPUT

The subroutine ***simulation.m*** outputs the following data:

1. T is an array of the permanence time in each location;

2. I is an array of the visited locations during the evolution; $I(k)$ is the index of the location when k switches are available, while $T(k)$ is the time interval spent in location $I(k)$.
3. J is the total cost of the evolution;
4. X is a sequence of points $\boldsymbol{x}$ that describes the evolution.
   To sketch the plot of the evolution type the command

   $$>> \ plot(X(1,:), X(2,:))$$

Example E.2 *In Section 4.5.1 we implemented the switched system model of a servomechanism with gear box. We firstly run the function* **regions.m** *with the given numerical values, hence we simulated an evolution from the therein given initial point. The function* **simulation.m** *exited the following numerical values:*

1. $T = [0.20, 0.20, 1.07, 2.53, 0.20]$,
2. $I = [1, 3, 5, 6, 5, 3]$,
3. $J = 4.75$.

Note that the array T, the switching intervals, has been converted into $\mathcal{T}^*$ that expresses the switching instants in an absolute time scale.

In addition vector X has been used to sketch the evolution depicted in Figure 4.8.

■

E.4 Function *index.m*

This function serves to calculate the integral

$$J = \int_0^{\varrho} \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt, \tag{E.1}$$

subject to $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)$ and $\boldsymbol{x}(0) = \boldsymbol{x}_0$.

Albeit not directly involved in the STP its usage is crucial for the described functions. Moreover it is quite general, hence we decided to describe it better.

In this paragraph $\boldsymbol{A}$, $\boldsymbol{Q}$ are general[2] square matrices.

To calculate the value of the integral (E.1), we preliminary need to solve the Lyapunov matrix equation. To this aim, at the MATLAB prompt, type

$$>> \ [\boldsymbol{Z}, flag] = lyap\_mod(\boldsymbol{A}', \boldsymbol{Q})$$

that solves the Lyapunov matrix equation $\boldsymbol{A}'\boldsymbol{Z} + \boldsymbol{Z}\boldsymbol{A} = -\boldsymbol{Q}$ and returns a $flag$ whose value is

- -1 if $\boldsymbol{Z}$ does not exists or it is not unique;
- 0 if $\boldsymbol{Z}$ exists and the matrix $\boldsymbol{A}$ is Hurwitz;
- 1 if $\boldsymbol{Z}$ exists and the matrix $\boldsymbol{A}$ is non Hurwitz.

The $flag$ variable is needed because when it assumes the values $0, 1$ it is possible to solve the integral *analytically*, so gaining in precision and computational time. In fact, as described in Appendix B, it holds

[2]Except for $\boldsymbol{Q} \geq \boldsymbol{0}$.

$$J = \int_0^{\varrho} \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt = \boldsymbol{x}_0'(\boldsymbol{Z} - \bar{\boldsymbol{A}}'(\varrho)\boldsymbol{Z}\bar{\boldsymbol{A}}(\varrho))\boldsymbol{x}_0.$$

In case $flag = -1$ then the Lyapunov equation has non unique or non existing solution, hence the cost must be calculated *numerically*. To this purpose we found satisfactory the implementation of a *constant step trapezoidal method* [27].

Now type

$>> \ J = index(\boldsymbol{A}, \boldsymbol{Q}, \boldsymbol{x}_0, \boldsymbol{Z}, flag, \varrho, dt)$

where dt is an appropriately chosen time interval, useful if the integral is calculated numerically.

Note that the call to function ***lyap_mod.m*** may be done inside the function ***index.m***, but in this case the computational time of ***index.m*** would increase. This is undesirable, because this function is called by ***regions.m*** at least $Ns^2 N_\vartheta N_t$ times.

Similarly the execution of

$>> \ J = index(\boldsymbol{A}, \boldsymbol{Q}, \boldsymbol{x}_0, \boldsymbol{Z}, flag)$

calculates

$$J = \int_0^{\infty} \boldsymbol{x}'(t)\boldsymbol{Q}\boldsymbol{x}(t)dt, \tag{E.2}$$

subject to $\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)$ and $\boldsymbol{x}(0) = \boldsymbol{x}_0$. In this case the computation is simplified. In fact it holds:

1. if $flag = 0$ then $J = \boldsymbol{x}_0'\boldsymbol{Z}\boldsymbol{x}_0$;
2. if $flag = -1, 1$ then, immediately, $J = +Inf$.

E.5 Function *regions4.m*

Download the file *STP4_corona.zip* from

http://www.diee.unica.it/∼dcorona/thesis.html.

The function *main* is called ***regions4.m***. It implements the STP in the fourth dimensional case. Hence all dynamics and weight matrices are of class $\mathbb{R}^{4\times 4}$. For this case we neglected the state jumps.

At the MATLAB prompt type

$>> \ [Table, X] = regions4(A, Q, G, d\_min, N_\xi, N_\varphi, N_\vartheta, N)$

where the input data A, Q, G, d_min, N have already been described in Section E.1.1.

The input variables $N_\xi, N_\varphi, N_\vartheta$ represent the discretization of the unitary semisphere in $\mathbb{R}^4$. To have a better interpretation of these values see also C.1.

An appropriate choice of these values should be $N_\vartheta = 2N_\varphi = 4N_\xi$, as it has been motivated in Appendix C.1. In the example implemented in Section 6.7.6 we chose $N_\xi = 15$. This choice leads to a discretization of 8581 points, sparse in Σ_4, that was considered acceptable.

OUTPUT

The calculations are long, hence a sequence of dots are visualized to confirm that the software is effectively running.

The data, residual cost and switching strategy, from each point of the unitary semisphere and for each dynamics, is collected in a matrix array $Table$, whose structure requires further explanations.

It is a matrix array $Table\{k\}$, $k = 1, ..., N+1$.

- $Table\{k\}$, $k = 1, \ldots, N+1$, contains the information relative to $k-1$ remaining switches. For example $Table\{3\}$ contains the residual cost and the color from each point of the semisphere, when 2 switches are left. Hence $Table\{N+1\}$ is the last calculated one, for N available switches. The command

  $$>> \ Table$$

  lists this matrix array.
- The element $Table\{k+1\}(i, h, 1)$ is the residual cost from point indexed by h (semisphere, see X) and from location i.
- The element $Table\{k+1\}(i, h, 2)$ is the color mapping of point indexed by h (semisphere, see X) and from location i.
- The data structure X is a matrix whose rows represent the polar angles of the unitary semisphere in $\mathbb{R}^4$. Hence $X(h, :)$ is a point in $\mathbb{R}^4$ in polar coordinates and $\varrho = 1$.

Remark E.4 *It is a good habit, when the function* **region4.m** *has terminated, to save the data by running the MATLAB command*

$$>> \ save \ < file\_name > \ Table, X, A, Q, G, d\_min, N_\xi, N$$

that stores the input and the calculated data in a file called $< file\_name >$*. To load this file type the MATLAB command*

$$>> \ load \ < file\_name >$$

from the belonging directory. ∎

E.6 Function *simulation4.m*

From a given initial hybrid state $(\boldsymbol{x}, i)$ it is possible to use the tables, calculated by function ***regions4.m***, to calculate the optimal switching intervals, the optimal switching sequence, the optimal cost and the optimal trajectory.

At the MATLAB prompt type

$$>> \ [T, I, J, X1] = simulation4(\boldsymbol{x}, i, d\_min, A, Q, Table, X, Ncsi, dt, op, th)$$

where all input variables have been defined in Section E.1.1 and E.5, except for

1. $Table, X$ are the output of program ***regions4.m***;
2. dt represents the time step of the simulation, usually $dt = 10^{-3}, 10^{-4}$ are efficiently fine;
3. op, a parameter so defined:

- $op = 0$, *finite* number of switches, uses all tables;
- $op = 1$, *infinite* number of switches, uses only the last calculated tables.

4. th is a terminating criterion on the norm of the continuous state $\boldsymbol{x}$, usually values $th = 10^{-3}, 10^{-4}$ are acceptable.

OUTPUT

The subroutine ***simulation4.m*** outputs the following data:

1. T is an array of the permanence time in each location;
2. I is an array of the visited locations during the evolution; $I(k)$ is the index of the location when k switches are available, while $T(k)$ is the time interval spent in location $I(k)$.
3. J is the total cost of the evolution;
4. $X1$ is a sequence of points $\boldsymbol{x}$ that describes the evolution.

In this case the trajectory of the evolution has no geometrical interpretation. However it is possible to sketch each row of matrix $X1$, i.e., $X1(i,:)$, $i = 1,2,3,4$ that represents the time evolution $x_i(t)$, with time step dt.

F

Notation, Symbols and Acronyms

Unless differently specified, notation, symbols an acronyms used in this thesis have the meaning detailed in the following tables.

F.1 Acronyms

Acronym	Significance
AHA	Autonomous Hybrid Automaton
ARE	Algebraic Riccati Equation
AS	Asymptotically Stable
CHA	Constrained Hybrid Automaton
DE	Differential Equation
DOF	Degree Of Freedom
e.g.	Latin *exempli gratia*
ES	Exponentially Stable
GHA	General Hybrid Automaton
HA	Hybrid Automaton
HJB	Hamilton Jacobi Bellman
HS	Hybrid System(s)
i.e.	Latin *id est*
IS	International System of measurements
LMI	Linear Matrix Inequality
LQ	Linear Quadratic
LQR	Linear Quadratic Regulator
LTI	Linear Time Invariant
OC	Optimal Cost
OP	Optimal Control Problem
OS	Operating System
PLC	Programmable Logical Controller
RF	Radio Frequency
S	Switched System
SA	Switched System of Arbitrary mode sequence
SF	Switched System of Fixed mode sequence
STP	Switching Table Procedure
wlg	without loss of generality
wrt	with regard to

F.2 Units

All measurements, when omitted, are intended to be in the International System. Angles are in radiants, angular velocity in radiants per second.

F.3 Notation

Symbol	Significance
$\|\cdot\|$	Norm$_2$
$\vert \cdot \vert$	Absolute value if $\cdot$ is a scalar, Norm$_\infty$ if $\cdot$ is a vector
$\vert \cdot \vert$	Cardinality of a set
$\lceil \cdot \rceil$	Approximation to higher integer
$Re(a)$	Real part of $a \in \mathbb{C}$
$Im(a)$	Imaginary part of $a \in \mathbb{C}$
$\cdot'$	Matrix transposition
$\cdot^*$	Optimal result or argument
$\boldsymbol{x}$	Vectors are **bold** in small letter
x_i, $x(i)$	i-th element of vector $\boldsymbol{x}$
$\boldsymbol{x}_i$	Particular vector $\boldsymbol{x}$
$\boldsymbol{A}$	Matrices are **bold** in capital letter
a_{ij}, $a(i, j)$	i-th row, j-th column of matrix $\boldsymbol{A}$
diag$\{\boldsymbol{a}\}$	Diagonal matrix whose main diagonal is the ordered vector $\boldsymbol{a}$
N, λ	Scalars
$\mathcal{S}$	Sets are in mathematical calligraphic
	(with the exception of numerical sets such as $\mathbb{N}$ or $\mathbb{R}^n$)
$\{\ldots\}$	Environment where sets are defined or listed (if countable)
$[x_1, x_2, \ldots, x_n]$	Row vector
$\triangleq$	Definition via equation
$\equiv$	Equivalence for sets or elements of sets
$\cong$	Approximated equality
$\propto$	Proportional
$\geq$	Greater or equal, for matrices semi-definite positive
$\prec$	Smaller, for elements of a set, in a lexicographic ordering sense
$\vee$	Logical OR
$\wedge$	Logical AND
$!$	Singleton, uniqueness
iff	If and only if $\Leftrightarrow$
$:, \vert$	Such that
$< \boldsymbol{v}_1, \boldsymbol{v}_2 >$	Scalar, dot, internal, inner product between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ of appropriate dimension
$\blacksquare$	Closes environments (theorems, definitions, algorithms and so on)
$\square$	Separates statements from proofs

F.4 Symbols

Symbol	Significance
t	Continuous time
t_0	Initial time
τ	Time instant
ϱ	Time, as a variable
δ	Time interval
n	State space dimension
$\boldsymbol{x}(t)$	State space evolution in $\mathbb{R}^n$
$\dot{\boldsymbol{x}}(t)$, $\dot{\boldsymbol{x}}$	Time derivative in $\mathbb{R}^n$ of vector $\boldsymbol{x}$
$\boldsymbol{x}$	Values of the state space in $\mathbb{R}^n$
$\boldsymbol{x}_k$, $\boldsymbol{x}(\tau_k)$	Values of the state space in $\mathbb{R}^n$ at time τ_k
$\boldsymbol{x}(\tau^+)$	$\lim_{t\to\tau^+} \boldsymbol{x}(t)$
$\boldsymbol{x}(\tau^-)$	$\lim_{t\to\tau^-} \boldsymbol{x}(t)$
$\boldsymbol{y}$, $\boldsymbol{y}_0$	State space such that $\|y\| = 1$
$(\boldsymbol{x}, i)$	Hybrid state, featured by a continuous term and a discrete term (indicating a location)
$(\boldsymbol{x}(t), i(t))$	Hybrid evolution, featured by a continuous evolution and a discrete evolution (indicating a sequence of location)
$\boldsymbol{u}(t)$	Continuous control input
$i(t)$	Discrete control input
$\boldsymbol{f}$	Affine term of $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{f}$
$\boldsymbol{A}$	Linear dynamics
$\boldsymbol{B}$	Control matrix in state space representation of linear systems $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$
$\boldsymbol{Q}$	Weight matrix for $\boldsymbol{x}$, usually $\boldsymbol{Q} \geq 0$
$\boldsymbol{R}$	Weight matrix for $\boldsymbol{u}$, usually $\boldsymbol{R} \geq 0$
$\boldsymbol{K}$	Proportional term for feedback optimal control of a LQR problem
J	Performance index
H	Switching cost
$\boldsymbol{M}$	State reset matrix
$\bar{\boldsymbol{A}}(t)$	Exponential matrix $\bar{\boldsymbol{A}}(t) = e^{\boldsymbol{A}t}$
$\bar{\boldsymbol{Q}}(t)$	Value of the integral $\int_0^t \boldsymbol{x}'(\varrho)Qx(\varrho)d\varrho$
$\bar{\boldsymbol{f}}(t)$	Value of the integral $\int_0^t \bar{\boldsymbol{A}}(\varrho)\boldsymbol{f}d\varrho$
$\boldsymbol{I}_n$	Identity matrix in $\mathbb{R}^n$
$\mathcal{C}$	Switching table: union of partitions of the state space
$\mathcal{C}_k^i$	Switching table of location i and k switches *to be performed*
$\mathcal{C}_\infty^i$	Switching table of location i and missing ∞ switches
$\mathcal{C}_\infty$	Common switching table for ∞ switches
$\mathcal{R}_i$	Partition of the state space with the color of the dynamics $\boldsymbol{A}_i$
$\mathcal{D}$	Set of discretization points of the state space
$\mathcal{L}$	Set of locations
$\mathcal{S}$	Set of location indexes
$\mathcal{E}$	Set of edges
$\mathcal{T}$	Set of ordered switching instants
$\mathcal{I}$	Set of switching indexes
O	Order of magnitude (especially used in computational complexity)
Σ_n	Unitary semisphere in $\mathbb{R}^n$, $\sum_1^n x_i^2 = 1$ and $x_n \geq 0$

References

1. P.J. Antsaklis (Ed.). Special issue on Hybrid Systems. *Proceedings of the IEEE 88(7)*. 2000.
2. E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pneuli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88(7):1011–1025, 2000.
3. R.E. Bellman. *Dynamic programming*. Princeton University Press, 1957.
4. A. Bemporad. Effective conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Transaction Automatic Control*, 49(5):832–838, 2004.
5. A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. In *Proceedings IEEE American Control Conference*, pages 1190–1194, Chicago, Illinois USA, June 2000.
6. A. Bemporad, F. Borrelli, and M. Morari. On the optimal control law for linear discrete time hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 2289, pages 105–119. Springer Verlag, 2002.
7. A. Bemporad, L. Giovanardi, and F.D. Torrisi. Performance driven reachability analysis for optimal scheduling and control of hybrid systems. Technical Report AUT00-15, Automatic Control Laboratory, ETH Zurich, Switzerland, September 2000.
8. A. Bemporad, A. Giua, and C. Seatzu. A master–slave algorithm for the optimal control of continuous–time switched affine systems. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 1976–1981, Las Vegas, Nevada USA, December 2002.
9. A. Bemporad, A. Giua, and C. Seatzu. Synthesis of state-feedback optimal controllers for continuous time switched linear systems. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 3182–3187, Las Vegas, USA, December 2002.
10. A. Bemporad and D. Mignone. *MIQP.M: A Matlab function for solving mixed integer quadratic programs*. ETH Zurich, 2000. code available at `http://control.ethz.ch/~hybrid/miqp`.
11. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.
12. A. Bemporad, B. De Schutter, and W.P.M.H. Heemels. On the equivalence classes of hybrid dynamical models. In *Proceedings 40th IEEE Conference on Decision and Control*, pages 364–369, Orlando, Florida USA, December 2001.
13. S.C. Bengea and R.A. DeCarlo. Conditions for the existence of a solution to a two-switched/hybrid optimal control problem. In *Proceedings IFAC Conference on Analysis and Design of Hybrid Systems*, pages 419–424, St. Malò, France, 2003.
14. D.P. Bertsekas. *Dynamic programming and optimal control, Vol. 1*. Athena Scientific, second edition, 2000.
15. F. Borrelli, M. Baotic, A. Bemporad, and M. Morari. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *Proceedings IEEE American Control Conference*, pages 4717–4722, Denver, Colorado USA, June 2003.
16. U. Boscain and G. Charlot. Stability of planar nonlinear systems. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 4283–4284, Maui, Hawaii USA, December 2003.

17. M.S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transaction Automatic Control*, 43(4):475–482, 1998.
18. M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Transaction Automatic Control*, 43(1):31–45, 1998.
19. M.S. Branicky and S.K. Mitter. Algorithms for optimal hybrid control. In *Proceedings 34th IEEE Conference on Decision and Control*, pages 2661–2666, New Orleans, Lousiana USA, December 1995.
20. P.E. Caines and Y.J. Wei. Hierarchical hybrid control systems: a lattice theoretical formulation. *IEEE Transaction Automatic Control*, 43(4):501–508, 1998.
21. C. G. Cassandras, Q. Liu, and K. Gokbayrak. Optimal control of a two-stage hybrid manufacturing system model. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 450–455, Phoenix, Arizona USA, December 1999.
22. C.C. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Kluwer Academic Publishers, Boston, 1999.
23. C.G. Cassandras, D.L. Pepyne, and Y. Wardi. Optimal control of a class of hybrid systems. *IEEE Transaction Automatic Control*, 46(3):398–415, 2001.
24. M.K. Çamlibel, W.P.M.H. Heemels, A.J. van der Schaft, and J.M. Schumacher. Solution concepts for hybrid dynamical systems. In *Proceedings IFAC World Congress*, Barcelona, Spain, 2002.
25. C. Chase, J.Serrano, and P.J.Ramadge. Periodicity and chaos from switched flow systems: contrasting examples of discretely controlled continuous systems. *IEEE Transaction Automatic Control*, 38:70–83, 1993.
26. P. Colaneri and J. Geromel. Stabilization of continuous time switched systems. In *Proceedings IFAC World Congress*, Prague, The Czech Republic, 2005. To appear.
27. V. Comincioli. *Analisi numerica*. McGraw-Hill, Milano, 1990. in italian.
28. D. Corona, A. Bemporad, A. Giua, and C. Seatzu. Optimal state-feedback quadratic regulation of linear hybrid automata. In *Proceedings IFAC Conference on Analysis and Design of Hybrid Systems*, pages 407–412, St. Malò, France, 2003.
29. D. Corona, A. Bemporad, A. Giua, and C. Seatzu. Optimal control of continuous-time switched affine systems. *IEEE Transaction Automatic Control*, 2004. Conditionally accepted as a regular paper.
30. D. Corona, A. Giua, and C. Seatzu. Optimal feedback switching laws for autonomous hybrid automata. In *Proceedings IEEE International Symposium on Intelligent Control*, pages 31–36, Taipei, Taiwan, September 2004.
31. D. Corona, A. Giua, and C. Seatzu. Stabilization of switched systems via optimal control. In *Proceedings IFAC World Congress*, Prague, The Czech Republic, 2005. To appear.
32. D. Corona, A. Giua, C. Seatzu, D. Gromov, E. Mayer, and J. Raisch. Optimal hybrid control for switched linear systems under safety and liveness constraints. In *Proceedings IEEE Symposium on Computer Aided Control System Design*, pages 35–40, Taipei, Taiwan, September 2004.
33. D. Corona, A. Giua, C. Seatzu, D. Gromov, E. Mayer, and J. Raisch. Optimal control of discrete time hybrid automata under safety and liveness constraints. In *Proceedings IEEE International Symposium on Intelligent Control*, Limassol, Cyprus, June 2005. Submitted.
34. G. Corriga, A. Giua, and G. Usai. An H_2 formulation for the design of a passive vibration-isolation system for cars. *Vehicle Systems Dynamics*, 26:381–393, 1996.
35. G. Corriga, S. Sanna, and G. Usai. An optimal tandem active-passive suspension for road vehicles with minimum power consumption. *IEEE Transaction Industrial Electronics*, 38(3):210–216, 1991.
36. J. Daafouz and G. Millerioux. Parameter dependent Lyapunov function for global chaos synchronization of discrete time hybrid systems. In *Proceedings IEEE Conference on Control of Oscillation and Chaos*, pages 339–342, St. Petersburg, Russia, July 2000.
37. P. De Leenheer, H. Nijmeijer, and B. De Jager. Stabilizing hybrid control law. In *Proceedings IFAC World Congress*, Barcelona, Spain, 2002.
38. B. De Schutter. Optimal control of a class of linear hybrid systems with saturation. 39(3):835–851, 2000.

39. B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, 2001.
40. R. DeCarlo, M.S. Branicky, S. Pettersson, and B. Lennartson. Perspectives and results on the stability and stabilizability of hybrid systems. *Proceedings of the IEEE*, 88(7):1069–1082, 2000.
41. M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal control of switching times in switched dynamical systems. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 2138–2143, Maui, Hawaii USA, December 2003.
42. E. Feron. Quadratic stabilizability of switched systems via state and output feedback. Technical Report CICS P-468, Center for Intelligent Control Systems, Massachusetts Institute of Technology, February 1996.
43. D. Franke, T. Moor, and J. Raisch. Supervisory control of switched linear systems. *at–Automatisierungstechnik.* Special Issue on Hybrid Systems I: Analysis and Control, 48(9):460–469, 2000.
44. B. Friedland. *Control system design, an introduction to state-space methods*. McGraw-Hill, Singapore, 1987.
45. M. Garavello and B. Piccoli. Hybrid necessary principle. In *Proceedings IFAC Conference on Analysis and Design of Hybrid Systems*, pages 289–294, St. Malò, France, June 2003.
46. A. Girard. Computation and stability analysis of limit cycles in piecewise linear hybrid systems. In *Proceedings IFAC Conference on Analysis and Design of Hybrid Systems*, pages 217–222, St. Malò, France, 2003.
47. A. Giua and A. Di Febbraro. *Sistemi ad eventi discreti*. McGraw-Hill, Milano, 2002. in italian.
48. A. Giua, M. Melas, C. Seatzu, and G. Usai. Design of a predictive semiactive suspension system. *Vehicle Systems Dynamics*, 41:277–300, 2003.
49. A. Giua, C. Seatzu, and C. Van Der Mee. Optimal control of autonomous linear systems switched with a pre–assigned finite sequence. In *Proceedings IEEE International Symposium on Intelligent Control*, pages 144–149, Mexico City, Mexico, September 2001.
50. A. Giua, C. Seatzu, and C. Van Der Mee. Optimal control of switched autonomous linear systems. In *Proceedings 40th IEEE Conference on Decision and Control*, pages 2472–2477, Orlando, Florida USA, December 2001.
51. A. Giua, C. Seatzu, and G. Usai. Semiactive suspension design with an optimal gain switching target. *Vehicle Systems Dynamics*, 31:213–232, 1999.
52. K. Gokbayrak and C.G. Cassandras. A hierarchical decomposition method for optimal control of hybrid systems. In *Proceedings 39th IEEE Conference on Decision and Control*, pages 1816–1821, Sydney, Australia, December 2000.
53. J.M. Gonçalves. Regions of stability for limit cycles of piecewise linear systems. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 651–656, Maui, Hawaii USA, December 2003.
54. E. Göring, E.C. von Glasner, and P. Schützner R. Povel. Intelligent suspension systems for commercial vehicles. In *Proceedings Congress MV2, Active Control in Mechanical Engineering*, pages 1–12, Lyon, France, 1993.
55. P. Grieder, F. Borrelli, F.D. Torrisi, and M. Morari. Computation of the constrained infinite time linear quadratic regulator. In *Proceedings IEEE American Control Conference*, pages 4711–4716, Denver, Colorado USA, June 2003.
56. A. Hac. Suspension optimization of a 2-dof vehicle model using stochastic optimal control technique. *Journal of Sound and Vibration*, 100(3):343–357, 1985.
57. S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 3972–3976, Phoenix, Arizona USA, December 1999.
58. S. Hedlund and A. Rantzer. Convex dynamic programming for hybrid systems. *IEEE Transaction Automatic Control*, 47(9):1536–1540, 2002.
59. W.P.M.H. Heemels, J.M. Schumacher, and S. Weiland. Well-posedness of linear complementarity systems. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 3037–3042, Phoenix, Arizona USA, December 1999.

60. J.P. Hespanha and A.S. Morse. Stability of switched system based on average dwell time. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 2655–2670, Phoenix, Arizona USA, December 1999.
61. M. Heymann, F. Lin, G. Meyer, and S. Resmerita. Analysis of zeno behaviours in hybrid systems. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 2379–2383, Las Vegas, Nevada USA, December 2002.
62. B. Hu, X. Xu, P.J. Antsaklis, and A.N. Michel. Robust stabilizing control law for a class of second-order switched systems. *Systems & Control Letters*, 38:197–207, 1999.
63. B. Hu, G. Zhai, and A.N. Michel. Common quadratic Lyapunov function with associated switching regions for two unstable second-order LTI systems. In *Proceedings 39th IEEE Conference on Decision and Control*, pages 1391–1396, Sydney, Australia, December 2000.
64. H. Ishii, T. Başar, and R. Tempo. Synthesis of switching rules for switched linear systems through randomized algorithms. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 4788–4793, Maui, Hawaii USA, December 2003.
65. K.H. Johansson, J. Lygeros, S. Sastry, and M. Egerstedt. On the existence and uniqueness of executions of hybrid automata. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 2249–2254, Phoenix, Arizona USA, December 1999.
66. K.H. Johansson, J. Lygeros, S. Sastry, and M. Egerstedt. Simulation of zeno hybrid automata. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 3538–3543, Phoenix, Arizona USA, December 1999.
67. M. Johansson and A. Rantzer. Computation of piecewise linear quadratic Lyapunov functions for hybrid systems. *IEEE Transaction Automatic Control*, 48(1):2–17, 2004.
68. H.K. Khalil. *Nonlinear systems*. Prentice-Hall, Upper Saddle River, New Jersey, third edition, 2002.
69. D.E. Kirk. *Optimal control theory: an introduction*. Prentice-Hall, Englewood Cliffs, N.J., 1970.
70. K.J. Kitching, D.J. Cole, and D. Cebon. Performance of semi-active damper for heavy vehicles. *ASME Journal of Dynamic Systems Measurement and Control*, 122:498–506, 2000.
71. D.E. Knuth. *The art of computer programming, Vol. 1*. Addison-Wesley, second edition, Reading, MA, 1973.
72. T. Kousaka, S. Tahara, T. Ueta, M. Abe, and H. Kawakami. Chaos in simple hybrid system and its control. *Electronic Letters*, 37(1):1–2, 2001.
73. D. Liberzon. Hybrid feedback stabilization of systems with quantized signals. *Automatica*, 39(7):1543–1554, 2003.
74. D. Liberzon and A.S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, 19(5):59–70, 1999.
75. D. Liberzon and R. Tempo. Gradient algorithms for finding common Lyapunov functions. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 4782–4787, Maui, Hawaii USA, December 2003.
76. B. Lincoln and B. Bernhardsson. LQR optimization of linear systems switching. *IEEE Transaction Automatic Control*, 47(10):1071–1074, 2002.
77. B. Lincoln and A. Rantzer. Optimizing linear system switching. In *Proceedings 40th IEEE Conference on Decision and Control*, pages 2063–2068, Orlando, Florida USA, December 2001.
78. B. Lincoln and A. Rantzer. Suboptimal dynamic programming with error bounds. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 2354–2359, Las Vegas, Nevada USA, December 2002.
79. B. Lincoln and A. Rantzer. Relaxed optimal control of piecewise linear systems. In *Proceedings IFAC Conference on Analysis and Design of Hybrid Systems*, pages 425–430, St. Malò, France, 2003.
80. J. Lygeros. On the relation of reachability to minimum cost optimal control. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 1910–1915, Las Vegas, Nevada USA, December 2002.
81. J. Lygeros, K.H. Johansson, S.N. Simič, J. Zhang, and S.S. Sastry. Dynamical properties of hybrid automata. *IEEE Transaction Automatic Control*, 48(1):2–17, 2004.

82. A.N. Michel and B. Hu. Toward a stability theory of general hybrid dynamical systems. *Automatica*, 35:371–384, 1999.
83. T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioural framework. *Systems & Control Letters*, 38:157–166, 1999. Special issue on *Hybrid Control Systems*.
84. T. Moor and J. Raisch. Abstraction based supervisory controller synthesis for high order monotone continuous systems. In S. Engell, G. Frehse, and E. Schnieder, editors, *Modelling, Analysis and Design of Hybrid Systems*, volume 279 of *Lecture notes in computer science*, pages 247–265. Springer-Verlag, 2002.
85. T. Moor, J. Raisch, and S. O'Young. Discrete supervisory control of hybrid systems by l-complete approximations. *Journal of Discrete Event Dynamic Systems*, 12(1):83–107, 2002.
86. X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approach to the description and analysis of hybrid systems. In *Hybrid Systems*, LNCS 736, pages 149–178. Springer Verlag, 1993.
87. K. Ogata. *Modern control engineering*. Prentice Hall International Editions, second edition, Englewood Cliffs, N.J., 1990.
88. C.C. Pantelides, M.P. Avraam, and N. Shah. Optimization of hybrid dynamic processes. In *Proceedings IEEE American Control Conference*, Chicago, Illinois USA, June 2000.
89. G. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. In *Proceedings 37th IEEE Conference on Decision and Control*, pages 4336–4341, Tampa, Florida USA, December 1998.
90. S. Pettersson. *Analysis and Design of Hybrid Systems*. PhD thesis, Chalmers University of Technology, Department of signals and systems, Control Engineering Laboratory, Göteborg, Sweden, 1999.
91. S. Pettersson. Synthesis of switched linear systems. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 5283–5288, Maui, Hawaii USA, December 2003.
92. B. Picasso and A. Bicchi. Stabilization of LTI systems with quantized state - quantized input static feedback. In O. Maler and A. Pneuli, editors, *Hybrid Systems: Computation and Control*, number 2623 in LNCS, pages 405–416, Heidelberg, 2003. Springer Verlag.
93. B. Piccoli. Necessary conditions for hybrid optimization. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 410–415, Phoenix, Arizona USA, December 1999.
94. J. Raisch. A hierarchy of discrete abstraction for a hybrid plant. *Journal Européen des Systèmes Automatisés*, 32:1073–1095, 1998.
95. J. Raisch and S. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, 43(4):569–573, April 1998.
96. P.J. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings IEEE, Special Issue on Discrete Events Dynamic Systems*, 77(1):81–98, 1989.
97. P. Riedinger, F. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 3059–3064, Phoenix, Arizona USA, December 1999.
98. V. Roberti, B. Ouyahia, and A. Devallet. Oleopneumatic suspension with preview semi-active control law. In *Proceedings Congress MV2, Active Control in Mechanical Engineering*, Lyon, France, 1993.
99. D. Sammier, O. Sename, and L. Dugard. Commande par placement de poles de sospensions automobiles. In *Conference Internationale Francophone d'Automatique*, Nantes, France, July 2002. (in french).
100. E. De Santis, M.D. Di Benedetto, and L.Berardi. Computation of maximal safe sets for switching systems. *IEEE Transaction Automatic Control*, 49(2):184–195, 2004.
101. M. Shaikh. *Optimal control of hybrid systems: theory and algorithms*. PhD thesis, Department of electrical and computer engineering, Univerity of Mc Gill, Montreal, Canada, 2004.
102. M.S. Shaikh and P.E. Caines. On trajectory optimization for hybrid systems: theory and algorithms for fixed schedules. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 1997–1998, Las Vegas, Nevada USA, December 2002.

103. M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: analysis and zonal algorithm for trajectory and schedule optimization. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 2144–2149, Maui, Hawaii USA, December 2003.
104. M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: optimization of switching times and combinatoric schedules. In *Proceedings IEEE American Control Conference*, pages 2773–2778, Denver, Colorado USA, June 2003.
105. M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: optimization of trajectories, switching times and location schedules. In O. Maler and A. Pneuli, editors, *Hybrid Systems: Computation and Control*, number 2623 in LNCS, pages 466–481, Berlin, 2003. Springer Verlag.
106. M. Spathopoulos and A. Trontis. Target control for hybrid systems with linear continuous dynamics. In *Proceedings 40th IEEE Conference on Decision and Control*, pages 1229–1234, Orlando, Florida USA, December 2001.
107. Z. Sun. Stability and insensitivity of switched linear systems. *IEEE Transaction Automatic Control*, 49(7):1133–1137, 2004.
108. Z. Sun. Stabilization and optimal switching of switched linear systems. *Automatica*, 2004. Submitted.
109. Z. Sun and R. Shorten. On convergence rates of switched linear systems. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 4800–4805, Maui, Hawaii USA, December 2003.
110. H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Proceedings 38th IEEE Conference on Decision and Control*, pages 425–430, Phoenix, Arizona USA, December 1999.
111. A.G. Thompson. An active suspension with optimal linear state feedback. *Journal of Sound and Vibration*, 5:187–203, 1976.
112. H.R. Visser and P.P.J. van den Bosch. Modelling of periodically switching networks. In *Proceedings IEEE Conference Power electronics specialists*, pages 67–73, June 1991.
113. Y. Wardi, M. Egerstedt, M. Boccadoro, and F. Delmotte. Optimal control of switching surfaces. In *Proceedings 43rd IEEE Conference on Decision and Control*, pages 1854–1859, Atlantis, Bahamas, December 2004.
114. Y. Wardi, D.L. Pepyne, and C.G. Cassandras. A backward algorithm for computing optimal control for single-stage hybrid manufacturing systems. *International Journal Prod. Res.*, 32(2):369–394, 2001.
115. M. Wicks, P. Peleties, and R. DeCarlo. Switched controller synthesis for the quadratic stabilization of a pair of unstable linear systems. *European Journal of Control*, 4:140–147, 1998.
116. J.C. Willems. Models for dynamics. *Dynamics reported*, 2:172–269, 1989.
117. G. Xie and L. Wang. Controllability and stabilizability of switched linear systems. *Systems & Control Letters*, 48(2):135–155, 2003.
118. X. Xu. *Analysis and Design of Switched Systems*. PhD thesis, Univerity of Notre Dame, Notre Dame, Indiana USA, 2001.
119. X. Xu and P.J. Antsaklis. Design of stabilizing control laws for second-order switched systems. In *Proceedings IFAC World Congress*, pages 181–186, Beijing, China, 1999.
120. X. Xu and P.J. Antsaklis. A dynamic programming approach for optimal control of switched systems. In *Proceedings 39th IEEE Conference on Decision and Control*, pages 1822–1827, Sydney, Australia, December 2000.
121. X. Xu and P.J. Antsaklis. An approach for solving general switched linear quadratic optimal control problems. In *Proceedings 40th IEEE Conference on Decision and Control*, pages 2478–2483, Orlando, Florida USA, December 2001.
122. X. Xu and P.J. Antsaklis. An approach to optimal control of switched systems with internally forced switchings. In *Proceedings IEEE American Control Conference*, Anchorage, Alaska USA, May 2002.
123. X. Xu and P.J. Antsaklis. An approach to switched systems optimal control based on parameterization of the switching instants. In *Proceedings IFAC World Congress*, pages 2280–2285, Barcelona, Spain, 2002.

124. X. Xu and P.J. Antsaklis. Optimal control of switched autonomous systems. In *Proceedings 41st IEEE Conference on Decision and Control*, pages 4401–4406, Las Vegas, Nevada USA, December 2002.
125. X. Xu and P.J. Antsaklis. Optimal control of switched systems via nonlinear optimization based on direct differentiations of value functions. *International Journal of Control*, 75(16/17):1406–1426, 2002.
126. X. Xu and P.J. Antsaklis. A linear programming approach to time optimal control of integrator switched systems with state constraints. In *Proceedings 42nd IEEE Conference on Decision and Control*, pages 2168–2173, Maui, Hawaii USA, December 2003.
127. X. Xu and P.J. Antsaklis. Quadratic optimal control problems for hybrid linear autonomous systems with state jumps. In *Proceedings IEEE American Control Conference*, pages 3393–3398, Denver, Colorado USA, June 2003.
128. X. Xu and P.J. Antsaklis. Results and perspectives on computational methods for optimal control of switched systems. In *6th Int. Workshop on Hybrid Systems: Computation and Control*, Prague, The Czech Republic, 2003.
129. X. Xu and P.J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Transaction Automatic Control*, 49(1):2–16, 2004.
130. H. Ye, A.N. Michel, and L. Hou. Stability theory for hybrid dynamical systems. *IEEE Transaction Automatic Control*, 43(4):461–474, 1998.
131. G. Zhai, H. Lin, and P.J. Antsaklis. Quadratic stabilizability of switched linear systems with polytopic uncertainties. *International Journal of Control*, 76(7):747–753, 2003.