Optimal Control of Continuous-Time Switched Affine Systems

Carla Seatzu, Daniele Corona, Alessandro Giua
Dip. Ingegneria Elettrica ed Elettronica, Università di Cagliari,
Piazza D'Armi, 09123 Cagliari, Italy
e-mail: {seatzu,daniele.corona,giua}@diee.unica.it

Alberto Bemporad
Dip. Ingegneria dell'Informazione, Università di Siena
Via Roma 56, 53100 Siena, Italy
e-mail: bemporad@dii.unisi.it

***Abstract*— The paper deals with optimal control of switched piecewise affine autonomous systems, where the objective is to minimize a performance index over an infinite time horizon. We assume that the switching sequence has a finite length, and that the decision variables are the switching instants and the sequence of operating modes. We present two different approaches for solving such an optimal control problem. The first approach iterates between a procedure that finds an optimal switching sequence of modes, and a procedure that finds the optimal switching instants. The second approach is inspired by dynamic programming and identifies the regions of the state space where an optimal mode switch should occur, therefore providing a state feedback control law.**

I. Introduction

Recent technological innovations have caused an ever increasing interest in the study of *hybrid systems* and many significant results have appeared in the literature [9], [11], [26], [40].

A hybrid system has many operating modes, each one governed by its own characteristic dynamical law [15]. Typically, mode transitions are classified as *autonomous* when they are triggered by variables crossing specific thresholds (state events) or by the elapse of certain time periods (time events), or *controlled* when they are triggered by external input events.

A. Optimal control of Hybrid Systems: state of the art

The problem of determining optimal control laws for hybrid systems and in particular for switched systems, has been extensively investigated in the last years and many results can be found in the control and computer science literature. Many authors considered very general problem statements so that it is only possible to derive necessary conditions for optimality, other authors provided methods for computing open-loop optimal trajectories. However very few practical algorithms were given for computing a state-feedback optimal control law. Here we review the most relevant literature.

For continuous-time hybrid systems, Branicky and Mitter [10] compare several algorithms for optimal control, while Branicky *et al.* [9] discuss general conditions for the existence of optimal control laws for hybrid systems.

Necessary optimality conditions for a trajectory of a switched system are derived using the maximum principle by Sussmann [33] and Piccoli [26], who consider a fixed sequence of finite length. A similar approach is used by Riedinger *et al.* [28], who restrict the attention to linear quadratic cost functionals but considering both autonomous and controlled switches.

Hedlund and Rantzer [20], [21] use convex dynamic programming to approximate hybrid optimal control laws and to compute lower and upper bounds of the optimal cost, while the case of piecewise-affine systems is discussed by Rantzer and Johansson [27]. For determining the optimal feedback control law these techniques require the discretization of the state space in order to solve the corresponding Hamilton-Jacobi-Bellman equations.

Gokbayrak and Cassandras [19] use a hierarchical decomposition approach to break down the overall optimal control problem into smaller ones. In so doing, discretization is not involved and the main computational complexity arises from a higher-level nonlinear programming problem. In [11] Cassandras *et al.* consider a particular hybrid optimization problem of relevant importance in the manufacturing domain and develop efficient solution algorithms for classes of problems.

Xu and Antsaklis have surveyed several interesting optimal control problems and methods for switched systems in [38]. Among them, we mention an approach based on the parametrization of the switching instants [39] and one based on the differentiation of the cost function [36]. Using similar approaches, a problem of optimal control of switched autonomous systems is studied in [35]. However the method encounters major computational difficulties when the number of available switches grows. In [37] the authors consider a switched autonomous linear system with linear discontinuities (jumps) on the state space and a finite time performance index penalizing switching and a final penalty, and study the problem only for fixed mode sequences.

Bengea and De Carlo [7] apply the maximum principle to an embedded system governed by a logic variable and a continuous control. The provided control law is open loop, however

To appear on the *IEEE Trans. on Automatic Control*, 2006.

some necessary and sufficient conditions are introduced for optimality.

Shaikh and Caines [32] consider a finite-time hybrid optimal control problem and give necessary optimality conditions for a fixed sequence of modes using the maximum principle. In [31] these results are extended to non-fixed sequences by using a suboptimal result based on the Hamming distance permutations of an initial given sequence. Finally, in [30], the authors derive a feedback law (similar to that one considered in this paper) but for a finite time LQR problem whose solutions are strongly dependent of the initial conditions, thus providing open-loop solutions.

Egerstedt, Wardi and Delmotte in [16] considered an optimal control problem for switched dynamical systems, where the objective is to minimize a cost functional defined on the state and where the control variable consists of the switching times. A gradient-descent algorithm is proposed based on an especially simple form of the gradient of the cost functional.

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 [6] introduce a hybrid modeling framework that, in particular, handles both internal 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), and showed how mixed-integer quadratic programming (MIQP) 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 MIQPs on-line, an asymptotically stabilizing control law is obtained. For those cases where on-line optimization is not viable, Bemporad *et al.* [1], [2] and Borrelli *et al.* [8] propose multiparametric 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.* [3] propose a technique that rules out switching sequences that are either not optimal or simply not compatible with the evolution of the dynamical system.

For a special class of discrete-event systems, De Schutter and van den Boom [13], [29] proposed an optimal receding-horizon strategy that can be implemented via linear programming.

An algorithm to optimize switching sequences that has an arbitrary degree of suboptimality was presented by Lincoln and Rantzer in [23], and in [24] the same authors consider a quadratic optimization problem for systems where all switches are autonomous.

B. The proposed approach

In this paper we focus our attention on *switched systems*, that are a particular class of hybrid systems in which all switches are controlled by external inputs. Giua *et al.* [17], [18] considered the optimal control of continuous-time switched systems composed by linear and stable autonomous dynamics with a piecewise-quadratic cost function. The assumption is that the switching sequence has finite length N and that the mode sequence is fixed, so that only the switching instants must be optimized. They provided a numerically viable way of computing a state-feedback control law in the form of N *switching regions*, where the k-th switching region, $k = 1, \ldots, N$ is the set of states where the k-th switch must occur.

In this paper, we solve an optimal control problem for continuous-time switched affine systems with a piece-wise quadratic cost function. We first present the procedure to construct the optimal switching regions for a finite-length fixed sequence. Secondly, we propose two different approaches to solve a similar optimal control problem when in the finite-length switching sequence both the *switching instants* and the *mode sequence* are decision variables.

The first approach is called *master-slave procedure* (MSP) [4] and exploits a synergy of discrete-time and continuous-time techniques alternating between two different procedures. 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 is based on the construction of the switching regions and finds the optimal switching instants, assuming the mode sequence is known. Although we formally prove that the algorithm always converges, the global minimum may not always be reached. A few simple heuristics can be added to the algorithm to improve its performance. A related approach that optimizes hybrid processes by combining mixed-integer linear programming (MILP) to obtain a candidate switching sequence and dynamic simulation was proposed in [25]. A two-stage procedure which exploits the derivatives of the optimal cost with respect to the switching instants was proposed in [39].

The second approach, called *switching table procedure* (STP) [5], is based on dynamic programming ideas and allows one to avoid the explosion of the computational burden with the number of possible switching sequences. It relies on the construction of switching tables and can be seen as a generalization of the slave procedure. In fact, the switching tables not not only specify the regions of the state space where an optimal switch should occur, but also what the optimal next dynamical mode must be. The solution is always globally optimal and in state-feedback form. A similar approach based on the construction of "optimal switching zones" was also used in [30].

To summarize: STP is guaranteed to find the optimal solution and provides a "global" closed-loop solution, i.e., the tables may be used to determine the optimal state feedback law for all initial states. On the other hand, MSP is not guaranteed to converge to a global optimum: it only provides an open-loop solution for a given initial state. Note however that, as a by-product, it also provides a "local" state-feedback solution, as the slave procedure consists of tables that may be used to determine a state feedback policy (this, however, is only optimal for small perturbations around the given initial state). Furthermore, MSP handles more general cost functions than STP, such as penalties associated with mode switching, and requires a lighter computational effort. Henceforth, both procedures have pros and cons, and preferring one to another will depend on the application at hand.

Finally, in the last section of the paper we discuss the computational complexity of the two approaches.

The problem considered in this paper where the mode sequence is a decision variable is significant in all those applications where the controller has the ability of choosing among several dynamics. There are many cases in which it is necessary to consider finite length sequences as we do in the paper. As an example, when there is a cost associated to each switch an infinite switching sequence would lead to an infinite cost. Furthermore, we also feel that solving the optimization problem in the case of a finite length sequence is a necessary first step to derive an optimal control law for an infinite time horizon and an infinite number of available switches: in the literature this setting is typically considered when addressing stabilizability issues of switched linear autonomous systems (see for instance [14]).

The paper is structured as follows. In Section 2 we provide the exact problem formulation. In Section 3 we restrict our attention to the case of a fixed sequence of modes. In Section 4 we illustrate the master-slave procedure. The switching table procedure is then illustrated in Section 5. A detailed comparison among the two proposed approaches is performed in Section 6.

II. Problem Formulation

We consider the following class of continuous-time hybrid systems, commonly denoted as *switched affine systems*,

$$\begin{aligned} \dot{x}(t) &= A_{i(t)}x(t) + f_{i(t)},\ i(t) \in \mathcal{S}, && \text{(1a)}\\ x(t^+) &= J_{i,j}x(t^-) \quad \text{if} \quad i(t^-) = i,\ i(t^+) = j, && \text{(1b)}\end{aligned}$$

where $x(t) \in \mathbb{R}^n$, $i(t) \in \mathcal{S}$ is the current mode and represents a control variable, $\mathcal{S} \triangleq \{1, 2, \ldots, s\}$ is a finite set of integers, each one associated with a matrix $A_i \in \mathbb{R}^{n\times n}$ and a vector $f_i \in \mathbb{R}^n$, $i \in \mathcal{S}$. Equation (1b) models a reset condition, by allowing that whenever at time t a switch from mode $i(t^-) = i$ to mode $i(t^+) = j$ occurs, the state jumps from $x(t^-)$ to $x(t^+) = J_{i,j}x(t^-)$, where $J_{i,j} \in \mathbb{R}^{n\times n}$ are constant matrices, $i, j \in \mathcal{S}$, $J_{i,i} = I$, and the continuity of the state trajectory at the switching instant from mode i to mode j is preserved if $J_{i,j} = I$.

In order to make (1) stabilizable on the origin, we assume the following:

Assumption 2.1: There exists at least one mode $i \in \mathcal{S}$ such that A_i is strictly Hurwitz and $f_i = 0$. ■

A. Optimal control problem

The main objective of this paper is to solve the optimal control problem

$$\begin{aligned} V_N^* &\triangleq \min_{I,\mathcal{T}} \Big\{ F(I,\mathcal{T}) \\ &\triangleq \int_0^\infty x'(t)Q_{i(t)}x(t)dt + \sum_{k=1}^{N} H_{i_{k-1},i_k} \Big\} \\ \text{s.t.}\quad & \dot{x}(t) = A_{i(t)}x(t) + f_{i(t)} \\ & x(0) = x_0 \\ & i(t) = i_k \in \mathcal{S} \text{ for } \tau_k \le t < \tau_{k+1}, \quad k = 0, \ldots, N, \\ & 0 = \tau_0 \le \tau_1 \le \ldots \le \tau_N < \tau_{N+1} = +\infty, \\ & x(\tau_k^+) = J_{i_{k-1},i_k} \ldots J_{i_{h-1},i_h} x(\tau_h^-) \\ & \qquad \text{if } \tau_{h-1} < \tau_h = \ldots = \tau_k < \tau_{k+1}, \end{aligned} \tag{2}$$

where Q_i are positive semi-definite matrices, and x_0 is the initial state of the system.

In this optimization problem there two types of decision variables:

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

Accordingly, the cost 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 $H_{i,i} = 0$, $\forall i \in \mathcal{S}$.

Note that in the last equation of (2), we slightly generalized model (1b) by allowing that more than one consecutive switch may occur at the same time instant $\tau_h = \tau_{h+1} = \cdots = \tau_k$.

Denote by $i^*(t)$, $t \in [0, +\infty)$, $i^*(t) = i_k^*$ for $\tau_{k-1}^* \le t < \tau_k^*$, $k = 0, \ldots, N$, the switching trajectory solving (2), and by I^*, $\mathcal{T}^*$ the corresponding optimal sequences. By letting $\delta_k \triangleq \tau_k - \tau_{k-1} \ge 0$ be the time interval elapsed between two consecutive switches, $k = 1, \ldots, N$, the optimal control problem (2) can be rewritten as

$$\begin{aligned} V_N^* \triangleq\ & \min_{I,\mathcal{T}} \Big\{ \sum_{k=0}^{N} \big[x_k' \bar{Q}_{i_k}(\delta_{k+1}) x_k + \bar{c}_{i_k}(\delta_{k+1}) x_k \\ & \qquad + \bar{\alpha}_{i_k}(\delta_{k+1}) \big] + \textstyle\sum_{k=1}^{N} H_{i_{k-1},i_k} \Big\} \\ \text{s.t.}\quad & x_{k+1} = J_{i_k,i_{k+1}} \bar{A}_{i_k}(\delta_{k+1}) x_k + \bar{f}_{i_k}(\delta_{k+1}), \\ & \qquad k = 0, \ldots, N-1 \\ & x_0 = x(0) \end{aligned} \tag{3}$$

where

$$\begin{aligned} \bar{A}_i(\delta) &\triangleq e^{A_i\delta}, \\ \bar{f}_i(\delta) &\triangleq \int_0^\delta e^{A_i t} f_i dt, \end{aligned} \tag{4}$$

$\bar{Q}_i(\delta)$, $\bar{c}_i(\delta)$, $\bar{\alpha}_i(\delta)$ can be obtained by simple integration and linear algebra, as reported in Appendix A, or even resorting to numerical integration.

B. Affine versus linear models

Before proceeding further, we observe that the original affine dynamics (1a) can be rewritten as a linear dynamics by simply augmenting the state space from $\mathbb{R}^n$ to $\mathbb{R}^{n+1}$:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} = \begin{bmatrix} A_{i(t)} & f_{i(t)} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix} \tag{5}$$

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 matrices of the form

$$\begin{bmatrix} Q_i & 0 \\ 0 & 0 \end{bmatrix}$$

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

Henceforth, without loss of generality, in the rest of the paper when necessary we will deal with the optimal control of switched piecewise *linear* systems whose performance index is the same as in the previous formulation (2). This will be done in Section 3 and Section 5, where the procedures based on the construction of the switching tables are presented.

On the contrary, the master-phase of the procedure discussed in Section 5 may *explicitly* handle affine models and we prefer to deal with the general form (1a) when presenting this approach.

We also observe that Assumption 2.1 is sufficient to ensure that the system is stabilizable on the origin (and hence that the optimal control problem we consider is solvable with a finite cost) but it is not strictly necessary. As an example, assume that all dynamics A_i have a displacement term $f_i \neq 0$ but that at least one dynamics, say A_j, is Hurwitz. One can make a linear state-coordinate transformation $x \to z + A_j^{-1} f_j$ and penalize — whenever mode i is active — the deviation from the target state through the quadratic term $(x - A_j^{-1} f_j)' Q_i (x - A_j^{-1} f_j) = z' Q_i z$.

III. Fixed Mode Sequence

For ease of exposition, in this section the attention is focused onto the case in which the sequence of operating modes is fixed and only the switching instants are decision variables, which is a common ingredient both for the slave-phase of the master-slave algorithm and for the switching table procedure presented in the next sections.

We consider linear dynamics as explained in Subsection II-B. Given that the switching sequence $I = \{i_0, \ldots, i_N\}$ is pre-assigned, to simplify the notation we denote the state matrices as $A_k \triangleq A_{i_k}$ for $k = 0, \cdots, N$. We also denote the jump (resp., switching cost) matrices as $J_k \triangleq J_{i_{k-1}, i_k}$ (resp., $H_k \triangleq H_{i_{k-1}, i_k}$) for $k = 1, \cdots, N$.

Let us also observe that in the general optimization problem (2) although the number of allowed switches is N, it may possible to consider also solutions where only $m < N$ switches effectively occur. This can be done choosing $i_m = i_{m+1} = \cdots = i_N$: in this case the switching cost only depends on the first m switches, because by definition $H_{i_m, i_{m+1}} = \ldots = H_{i_{N-1}, i_N} = 0$ (recall that $H_{i,i} = 0$). We also want to keep this additional feature when restricting our attention to the fixed mode sequence problem formulation. Thus we explicitly introduce a new variable $m \leq N$ that is equal to the number of switches effectively occurring in an optimal solution. Accordingly, we assume that $\tau_{m+1} = \cdots = \tau_N = \tau_{N+1} = +\infty$ so that once the system switches to mode A_m at time τ_m, it will always remain in that mode.

Summarizing, the optimization problem we consider in this section is

$$\begin{aligned} V_N^* \ & \triangleq \min_{\mathcal{T}} \left\{ F(\mathcal{T}) \triangleq \int_0^\infty x'(t) Q_{i(t)} x(t) dt + \sum_{k=1}^m H_k \right\} \\ \text{s.t.} \ & \dot{x}(t) = A_k x(t) \text{ for } \tau_k \leq t < \tau_{k+1}, \quad k = 0, \ldots, m, \\ & x(0) = x_0 \\ & 0 = \tau_0 \leq \tau_1 \leq \ldots \leq \tau_m < \tau_{m+1} = +\infty, \\ & x(\tau_k^+) = J_k \ldots J_h x(\tau_h^-) \\ & \qquad \text{if } \ \tau_{h-1} < \tau_h = \ldots = \tau_k < \tau_{k+1}, \\ & m \leq N \end{aligned} \tag{6}$$

or equivalently

$$\begin{aligned} & \min_{\mathcal{T}} \ \left\{ \sum_{k=0}^m x_k' \bar{Q}_k(\delta_{k+1}) x_k + \sum_{k=1}^m H_k \right\} \\ & \text{s.t.} \quad x_{k+1} = J_{k+1} \bar{A}_k(\delta_{k+1}) x_k, \ \ k = 0, \ldots, m-1 \\ & \qquad \ x_0 = x(0) \\ & \qquad \ m \leq N \end{aligned} \tag{7}$$

where for ease of notation we have set

$$\begin{aligned} \bar{A}_k(\delta_{k+1}) &\triangleq e^{A_k(\tau_{k+1} - \tau_k)}, \ \ k = 0, \ldots, m-1, \\ \bar{Q}_k(\delta_{k+1}) &\triangleq \bar{Q}_{i_k}(\tau_{k+1} - \tau_k), \ \ k = 0, \ldots, m. \end{aligned} \tag{8}$$

Now, we show that the optimal control law for the above optimization problem takes the form of a state–feedback, i.e., to determine if a switch from A_{k-1} to A_k should occur it is only necessary to look at the current system state x. More precisely, the optimization problem can be solved computing a set of tables $\mathcal{C}_k$ ($k = 1, \ldots, N$). Each table represents a portion of the state space $\mathbb{R}^n$ in two regions: $\mathcal{R}$ and $\mathcal{R}'$. If the current system dynamics is A_{k-1} we will switch to A_k as soon as the state reaches a point in the region $\mathcal{R}$ of the table $\mathcal{C}_k$, for $k = 1, \ldots, N$.

To prove this result, we also show constructively how the tables $\mathcal{C}_k$ can be computed.

A. Computation of the Switching Tables

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 occur, and the cost of the time–driven evolution. We will consider the two parts separately.

Definition 3.1 (Event cost): Let us assume that after k switches the current system dynamics is that corresponding to matrix A_k: in the future evolution up to $N - k$ switches may occur. We define the *remaining event cost* starting from dynamics A_k and executing $j = 0, 1, \ldots, N-k$ more switches recursively as follows:

$$E_{k,0} = 0, \qquad \text{and} \qquad E_{k,j} = E_{k,j-1} + H_{k+j} \text{ for } j \geq 1.$$

■

In a similar manner we would also like to compute the cost of the time–driven evolution.

Definition 3.2 (Time cost): Let us assume that after k switches the current system dynamics is that corresponding to matrix A_k and the current state vector is x. The *remaining time cost* starting from the initial state x with dynamics A_k and executing $j = 0, 1, \ldots, N-k$ additional switches is

$$\begin{aligned}&\tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_j) = x'\bar{Q}_k(\varrho_1)x \\ &\quad + x' e^{A_k' \varrho_1} J_{k+1}' \bar{Q}_{k+1}(\varrho_2) J_{k+1} e^{A_k \varrho_1} x \\ &\quad + \ldots \\ &\quad + x' e^{A_k' \varrho_1} J_{k+1}' \cdots e^{A_{k+j-1}' \varrho_j} J_{k+j}' \bar{Q}_{k+j}(+\infty) \cdot \\ &\qquad \cdot J_{k+j} e^{A_{k+j-1}\varrho_j} \cdots J_{k+1} e^{A_k \varrho_1} x,\end{aligned}$$

where $\varrho_i = \tau_{k+i} - \tau_{k+i-1} \geq 0$ represents the time spent within dynamics A_{k+i-1}. ■

Function $\tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_j)$ depends on j different parameters ϱ_i ($i = 1, \ldots, j$). We need to compute the optimal time cost by a suitable choice of these parameters. It is easy to prove using simple dynamic programming arguments that this optimal cost can be computed by solving j one-parameter optimizations.

Procedure 1: The optimal value

$$\tilde{T}^*_{k,j}(x) = \min_{\varrho_1, \ldots, \varrho_j \geq 0} \tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_j)$$

can be recursively computed as follows.

- Assume that $j = 0$, i.e., no future switch occurs. Then there is only one possible future evolution (henceforth it is optimal): the system evolves for ever with dynamics A_k. The *optimal remaining cost* is
$$\tilde{T}^*_{k,0}(x) = x'\bar{Q}_k(+\infty)x. \tag{9}$$
We also define, with a notation that will be clear in the sequel, $\varrho_{k,0}(x) = +\infty$.
- Assume that j additional switches occur, with $j = 1, \ldots, N-k$. We first consider a remaining evolution such that: (a) the system evolves with dynamics A_k for a time ϱ; (b) a switch to A_{k+1} occurs after ϱ; (c) the future evolution is the optimal evolution that allows only $j-1$ additional switches. The remaining cost starting from x due to this time–driven evolution is
$$T_{k,j}(x, \varrho) = x'\bar{Q}_k(\varrho)x + \tilde{T}^*_{k+1,j-1}(J_{k+1} e^{A_k \varrho} x). \tag{10}$$
We define the value of ϱ that minimizes (10) for all values of $j = 1, \ldots, N-k$ [1]:
$$\varrho_{k,j}(x) = \arg\min_{\varrho \geq 0} T_{k,j}(x, \varrho), \tag{11}$$
and denote the optimal value as
$$T^*_{k,j}(x) = T_{k,j}(x, \varrho_{k,j}(x)). \tag{12}$$

It is obvious that

$$\begin{aligned}T^*_{k,j}(x) &\triangleq \min\nolimits_{\varrho \geq 0} T_{k,j}(x, \varrho) \\ &= \min\nolimits_{\varrho_1, \ldots, \varrho_j \geq 0} \tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_j) \triangleq \tilde{T}^*_{k,j}(x).\end{aligned}$$

■

[1] We can have that $\varrho_{k,j}(x)$ is a set (not a single value). In this case we assume a single value (i.e., the minimum) is taken.

Let us now state an elementary result.

Proposition 3.1: If x is a vector such that $x = \lambda y$, with $\|y\|_2 = 1$ and $\lambda \in \mathbb{R} \setminus \{0\}$, the cost function of the time-driven evolution is such that

$$(a) \qquad T^*_{k,j}(x) = \lambda^2 T^*_{k,j}(y); \tag{13}$$

$$(b) \qquad \varrho_{k,j}(x) = \varrho_{k,j}(y). \tag{14}$$

Proof: This can be proved inductively.

Clearly the result holds for $j = 0$ (base step) because by definition

$$T^*_{k,0}(x) = x'\bar{Q}_k(+\infty)x = \lambda^2 y'\bar{Q}_k(+\infty)y = \lambda^2 T^*_{k,0}(y),$$

and

$$\varrho_{k,0}(x) = \varrho_{k,0}(y) = \infty.$$

Assume the result holds for $T^*_{k,j-1}(x)$; we show that it also holds for $T^*_{k,j}(x)$ (inductive step). In fact:

$$\begin{aligned}T_{k,j}(x, \varrho) &= x'\bar{Q}_k(\varrho)x + T^*_{k+1,j-1}(J_{k+1}e^{A_k \varrho}x) \\ &= \lambda^2 y'\bar{Q}_k(\varrho)y + \lambda^2 T^*_{k+1,j-1}(J_{k+1}e^{A_k \varrho}y) \\ &= \lambda^2 T_{k,j}(y, \varrho),\end{aligned}$$

and from this it immediately follows that:

$$T^*_{k,j}(x) = \lambda^2 T^*_{k,j}(y),$$

and

$$\begin{aligned}\varrho_{k,j}(x) &= \arg\min_{\varrho \geq 0} T_{k,j}(x, \varrho) = \arg\min_{\varrho \geq 0} \lambda^2 T_{k,j}(y, \varrho) \\ &= \arg\min_{\varrho \geq 0} T_{k,j}(y, \varrho) = \varrho_{k,j}(y).\end{aligned}$$

■

Having introduced the necessary notation, we can finally compute the optimal evolution that starts from a vector x, in terms of the optimal evolution that starts from a vector y on the unitary semi-sphere and parallel to x.

Theorem 3.1: Let A_k be the current dynamics and let the current state vector at time t be $x = \lambda y$ with $\|y\|_2 = 1$ and $\lambda \in \mathbb{R} \setminus \{0\}$.

i) The optimal remaining cost starting from x is
$$F_k(x) = \min_{j=0,\ldots,N-k} \{\lambda^2 T^*_{k,j}(y) + E_{k,j}\}, \tag{15}$$

ii) The optimal remaining number of switches starting from x is
$$j_k(x) = \arg\min_{j=0,\ldots,N-k} \{\lambda^2 T^*_{k,j}(y) + E_{k,j}\} \tag{16}$$

iii) The optimal evolution switches to A_{k+1} at time $\tau_{k+1} = t + \delta_{k+1}(x)$, where
$$\delta_{k+1}(x) = \varrho_{k,j_k(x)}(x) = \varrho_{k,j_k(x)}(y). \tag{17}$$

Proof: The optimal remaining cost starting from x depends on a discrete decision variable, i.e., the number of switches j that in this case belongs to the set $\{0, \cdots, N-k\}$ and on the continuous decision variables ϱ_i's. Thus:

$$\begin{aligned} F_k(x) &= \min_{\substack{j=0,\cdots,N-k \\ \varrho_1,\ldots,\varrho_j \geq 0}} \{\tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_j) + E_{k,j}\} \\ &= \min_{j=0,\cdots,N-k} \{ \min_{\varrho_1,\ldots,\varrho_j \geq 0} \tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_j) + E_{k,j}\} \\ &= \min_{j=0,\cdots,N-k} \{\min_{\varrho \geq 0} T_{k,j}(x, \varrho) + E_{k,j}\} \\ &= \min_{j=0,\cdots,N-k} \{T^*_{k,j}(x) + E_{k,j}\} \\ &= \min_{j=0,\cdots,N-k} \{\lambda^2 T^*_{k,j}(y) + E_{k,j}\} \end{aligned}$$

and the other two statements trivially follow from this. □

According to the previous proposition, assume that k switches have occurred and A_k is the current dynamics with current state vector x. Then, the remaining cost is minimized by an evolution that executes immediately the $k+1$-th switch to A_{k+1} if and only if $\delta_{k+1}(x) = 0$. This leads to the definition of a switching table.

Definition 3.3: The table $\mathcal{C}_k$ is a partition of the state space in two regions:

- the optimal region $\mathcal{R}$ for the $k-$th switch is defined as follows: $\forall\ x \neq 0$, $x \in \mathcal{R}$ if $\delta_k(x) = 0$. Moreover, since we want $\mathcal{R}$ to be a closed region, we assume that $0 \in \mathcal{R}$ if $0 \in \partial\mathcal{R}$ where $\partial\mathcal{R}$ is the boundary of $\mathcal{R}$.
- The complementary region: $\mathcal{R}' = \mathbb{R}^n \setminus \mathcal{R}$.

■

The following result, that immediately follows from Theorem 3.1 and Definition 3.3, characterizes the switching regions.

Corollary 3.1: A state vector $x = \lambda y$, with $\|y\|_2 = 1$ and $\lambda \in \mathbb{R} \setminus \{0\}$, belongs to region $\mathcal{R}$ of $\mathcal{C}_{k+1}$ if and only if there exists an index $j^* \in \{1, \cdots, N-k\}$ such that the following two conditions are both verified:

$$\text{(a)}\quad \lambda^2 T^*_{k,j^*}(y) + E_{k,j^*} = \min_{j=0,\cdots,N-k} \left\{\lambda^2 T^*_{k,j}(y) + E_{k,j}\right\} \tag{18}$$

$$\text{(b)}\quad \varrho_{k,j^*}(y) = 0. \tag{19}$$

■

A final observation. To compute the switching table $\mathcal{C}_{k+1}$ and to determine the optimal remaining cost $F_k(x)$, we only need to compute the value $\varrho_{k,j}(y)$ with a one-parameter optimization (see equations (11) and (14)) for all y on the unitary semi-sphere: this can be easily done with a numerical procedure.

We discretize the unitary semisphere with the procedure described in Appendix B. In particular, in all the examples examined the number of samples has been heuristically chosen to ensure a reasonable tradeoff between accuracy and computational cost.

The corresponding values of $T^*_{k,j}(y)$ can be obtained applying equation (12), while to determine if a vector $x = \lambda y$ belongs to $\mathcal{R}$ and to compute the corresponding optimal remaining cost we only need to apply equations (18) and (19).

B. Structure of the Switching Regions

We now discuss the form that the switching regions may take.

Let us first consider the case of zero switching costs.

Proposition 3.1: Consider the case in which $H_1 = \cdots = H_N = 0$. Then for all $k = 0, \ldots, N-1$ the switching region $\mathcal{R}$ of table $\mathcal{C}_{k+1}$ is such that:

$$y \in \mathcal{R} \Longrightarrow (\forall \lambda \in \mathbb{R})\ \lambda y \in \mathcal{R}$$

i.e., the region $\mathcal{R}$ is homogeneous.

Proof: Thanks to equation (16), it is immediate to see that if all costs are zero

$$\begin{aligned} j_k(x) &= \arg\min_{j=0,\ldots,N-k} \lambda^2 T^*_{k,j}(y) \\ &= \arg\min_{j=0,\ldots,N-k} T^*_{k,j}(y) = j_k(y) \end{aligned}$$

and by equation (17) $\delta_{k+1}(x) = \delta_k(y)$. Thus $\delta_{k+1}(y) = 0 \Longrightarrow \delta_{k+1}(x) = 0$. □

We now consider the case of non-zero switching costs. Let us first state a trivial fact.

Fact 3.1: For all $x \in \mathbb{R}^n$, $k = 1, \ldots, N$ and $j = 1, \ldots, N-k$ it holds: $T^*_{k,j}(x) \leq T^*_{k,j-1}(x)$.

Proof: This can be easily shown using the definition of time cost. In fact

$$\begin{aligned} T^*_{k,j}(x) &= \tilde{T}^*_{k,j}(x) = \min_{\varrho_1,\ldots,\varrho_{j-1},\varrho_j \geq 0} \tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_{j-1}, \varrho_j) \\ &\leq \min_{\varrho_1,\ldots,\varrho_{j-1} \geq 0} \tilde{T}_{k,j}(x, \varrho_1, \ldots, \varrho_{j-1}, +\infty) \\ &= \min_{\varrho_1,\ldots,\varrho_{j-1} \geq 0} \tilde{T}_{k,j-1}(x, \varrho_1, \ldots, \varrho_{j-1}) \\ &= \tilde{T}^*_{k,j-1}(x) = T^*_{k,j-1}(x) \end{aligned}$$

□

We can finally state the following result.

Proposition 3.2: For all $k = 0, \ldots, N-1$, and for all $y \in \mathbb{R}^n$ (with $\|y\|_2 = 1$) there exists a finite $\tilde{\lambda}(k, y) > 0$, that depends on k and y, such that the switching region $\mathcal{R}$ of table $\mathcal{C}_{k+1}$ has the following property:

$$\tilde{\lambda}(k, y) y \in \mathcal{R} \Longrightarrow (\forall \lambda :\ |\lambda| \geq \tilde{\lambda}(k, y))\ \lambda y \in \mathcal{R}.$$

Proof: For all k and for all y on the unitary semi-sphere we can define

$$\tilde{\jmath} = \min\{j \mid T^*_{k,j}(y) = T^*_{k,N-k}(y)\}.$$

This allows us to write, also according to Fact 3.1, that the optimal remaining time costs can be ordered as follows

$$T^*_{k,0}(y) \geq T^*_{k,1}(y) \geq \ldots > T^*_{k,\tilde{\jmath}}(y) = \cdots = T^*_{k,N-k}(y),$$

while by definition the remaining event costs can be ordered as follows

$$E_{k,0} \leq E_{k,1} \leq \ldots \leq E_{k,\tilde{\jmath}_k(y)} \leq \cdots \leq E_{k,N-k+1}.$$

We now define

$$\tilde{\lambda}(k, y) = \begin{cases} 0 & \text{if } \tilde{\jmath} = 0 \\ \max\limits_{j=0,\ldots,\tilde{\jmath}-1} \sqrt{\dfrac{E_{k,\tilde{\jmath}} - E_{k,j}}{T^*_{k,j}(y) - T^*_{k,\tilde{\jmath}}(y)}} & \text{otherwise} \end{cases}$$

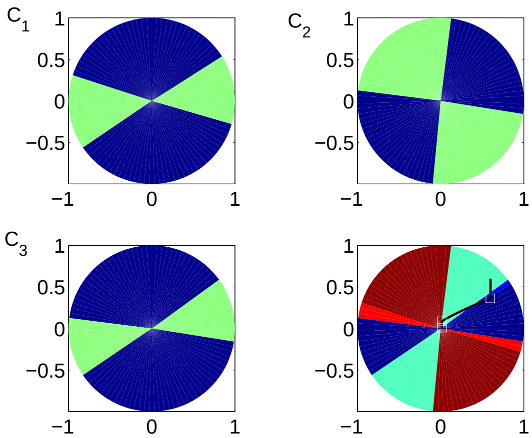

Fig. 1. The switching regions $\mathcal{C}_k$, $k = 1, 2, 3$ in the case of no cost associated to switches, and the system evolution for $x_0 = [0.6\ 0.6]'$.

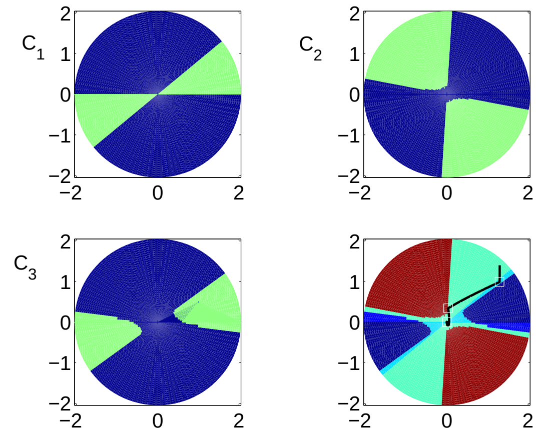

Fig. 2. The switching regions $\mathcal{C}_k$, $k = 1, 2, 3$ in the case of non-zero costs associated to switches, and the system evolution for $x_0 = [1.3\ 1.4]'$.

so that we can write that for all $\lambda \in \mathbb{R}$ with $|\lambda| \geq \tilde{\lambda}(k, y)$ and for all $j = 0, \ldots, N - k + 1$, it holds

$$\lambda^2 T^*_{k,\tilde{j}}(y) + E_{k,\tilde{j}} \leq \lambda^2 T^*_{k,j}(y) + E_{k,j}.$$

Thus the optimal remaining evolution starting from $x = \lambda y$ with $|\lambda| \geq \tilde{\lambda}(k, y)$ will contain $\tilde{j}$ more switches and x belongs to $\mathcal{R}$ if and only if $\varrho_k(y) = 0$. □

We finally denote

$$\tilde{\lambda} = \max_{k=1,\ldots,N} \max_{|y|=1} \tilde{\lambda}(k, y). \tag{20}$$

C. Numerical Examples

Let us now present the results of some numerical simulations. In particular, we consider a second order linear system whose dynamics may only switch between two matrices $A^{(1)}$ and $A^{(2)}$. We also assume that only three switches are possible ($N = 3$) and the initial system dynamics is $A_0 = A^{(1)}$. Thus, the sequence of switching is $A_0 = A^{(1)} \to A_1 = A^{(2)} \to A_2 = A^{(1)} \to A_3 = A^{(2)}$, where

$$A^{(1)} = \begin{bmatrix} -1 & 1 \\ -18 & -5 \end{bmatrix}, \qquad A^{(2)} = \begin{bmatrix} 1 & -5 \\ 1 & -3 \end{bmatrix}.$$

We also assume that all J_k's are equal to the identity matrix. Finally, we take $Q_0 = Q_1 = Q_2 = Q_3 = \mathrm{diag}\{1\ \ 2\}$.

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 1 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 1 we have shown the system evolution in the case of $x_0 = [0.6\ 0.6]'$.

The switching times are $\tau_1 = 0.01$, $\tau_2 = 0.35$ and $\tau_3 = 0.40$, and the optimal cost is $F(\tau_1, \tau_2, \tau_3) = 0.15$.

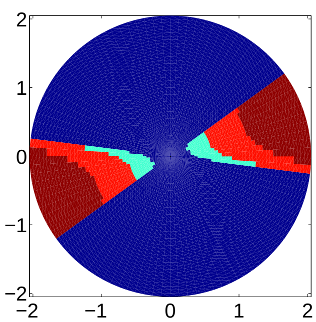

Fig. 3. The switching regions $\mathcal{C}_3$ for different values of the cost $H_3 \in \{0.1, 0.5, 2\}$.

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 2 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 2 we have shown the system evolution in the case of $x_0 = [1.3\ \ 1.4]'$. In this case, the switching times are $\tau_1 = 0.014$, $\tau_2 = 0.5$ and $\tau_3 = +\infty$, 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 regions $\mathcal{C}_3$ for different values of $H_3 \in \{0.1, 0.5, 2\}$.

These regions are shown in Figure 3, where larger regions correspond to smaller values of H_3.

IV. Master-Slave Procedure

In this section we propose a solution to the optimal control problem (2) in which both the switching instants and the switching sequence are decision variables, and the initial state $x(0)$ is given. The procedure exploits a synergy of discrete-time and continuous-time techniques, by alternating between a "master" procedure that finds an optimal switching sequence and a "slave" procedure that finds the optimal switching instants.

Problem 1 (Master): For a fixed sequence of switching times $\bar{\tau}_1, \ldots, \bar{\tau}_N$, solve the optimal control problem (3) with respect to $i_0, \ldots, i_N$. Denote by

$$\{i_0, \ldots, i_N\} = f_M(\bar{\tau}_1, \ldots, \bar{\tau}_N) \tag{21}$$

and $V_M(\bar{\tau}_1, \ldots, \bar{\tau}_N)$ the optimizing mode sequence and the corresponding optimal value, respectively. ■

Problem 2 (Slave): For a fixed sequence of switching indices $\bar{\imath}_0, \ldots, \bar{\imath}_N$, solve the optimal control problem (3) with respect to $\tau_1, \ldots, \tau_N$. Denote by

$$\{\tau_1, \ldots, \tau_N\} = f_S(\bar{\imath}_0, \ldots, \bar{\imath}_N) \tag{22}$$

and $V_S(\bar{\imath}_0, \ldots, \bar{\imath}_N)$ the optimizing timing sequence and the corresponding optimal value, respectively. ■

An approach for solving the slave phase has been extensively discussed in the previous section. Here below we describe a way to solving the master phase.

A. Master Algorithm

For a fixed sequence of switching times $\bar{\tau}_1, \ldots, \bar{\tau}_N$, the master algorithm solves the optimal control problem (3) with respect to $i_0, \ldots, i_N$. We assume here that $\bar{\tau}_N < +\infty$; if such is not the case the problem can be simplified discarding the infinity switching times $\bar{\tau}_{k+1} = \cdots = \bar{\tau}_N = +\infty$ and working on the shorter prefix $\bar{\tau}_1 \leq \cdots \leq \bar{\tau}_k < +\infty$.

It is a purely combinatorial problem that can be rephrased as:

$$\begin{aligned} \min_I \quad & \Big\{ F(I) \triangleq \sum_{k=0}^{N} \left[x_k' \bar{Q}_{i_k}(k) x_k + \bar{c}_{i_k}(k) x_k + \bar{\alpha}_{i_k}(k) \right] \\ & \qquad + \textstyle\sum_{k=1}^{N} H_{i_{k-1}, i_k} \Big\} \\ \text{s.t.} \quad & x_{k+1} = J_{i_k, i_{k+1}} \bar{A}_{i_k}(k) x_k + \bar{f}_{i_k}(k), \\ & \qquad k = 0, \ldots, N-1 \\ & x_0 = x(0), \end{aligned} \tag{23}$$

where

$$\begin{aligned} \bar{A}_{i_k}(k) &\triangleq e^{A_{i_k}(\bar{\tau}_k - \bar{\tau}_{k-1})}, \\ \bar{f}_{i_k}(k) &\triangleq \int_{\bar{\tau}_{k-1}}^{\bar{\tau}_k} e^{A_{i_k}(\bar{\tau}_k - t)} f_{i_k} dt, \\ \bar{Q}_{i_k}(k) &\triangleq \bar{Q}_{i_k}(\bar{\tau}_k - \bar{\tau}_{k-1}) \\ \bar{c}_{i_k}(k) &\triangleq \bar{c}_{i_k}(\bar{\tau}_k - \bar{\tau}_{k-1}) \\ \bar{\alpha}_{i_k}(k) &\triangleq \bar{\alpha}_{i_k}(\bar{\tau}_k - \bar{\tau}_{k-1}) \end{aligned} \tag{24}$$

and where $\bar{Q}_{i_k}(\delta_k)$, $\bar{c}_{i_k}(\delta_k)$ and $\bar{\alpha}_{i_k}(\delta_k)$ can be computed as outlined in Appendix A or resorting to numerical integration.

Problem (23) can be efficiently solved via Mixed-Integer Quadratic Programming (MIQP) [22], [12]. To this end, we need to introduce the binary variables $\gamma_i^k \in \{0, 1\}$, where $\gamma_i^k = 1$ if and only if the system is in the i-th mode between time τ_k and τ_{k+1}:

$$[\gamma_i^k = 1] \quad \leftrightarrow \quad [i_k = i], \; \forall k = 0, \ldots, N, \; \forall i \in \mathcal{S}, \tag{25a}$$

$$\bigoplus_{i=1}^{s} \gamma_i^k \quad = \quad 1, \quad \forall k = 0, \ldots, N, \tag{25b}$$

$$\bigoplus_{i \in \mathcal{S}_{as}} \gamma_i^N \quad = \quad 1, \tag{25c}$$

where the exclusive-or constraint (25b) follows by the fact that only one dynamics can be active in each interval $[\tau_k, \tau_{k+1})$, and in (25c) $\mathcal{S}_{as}$ is the set of indices $i \in \mathcal{S}$ such that A_i is strictly Hurwitz and $f_i = 0$, so that the last dynamics be asymptotically stable and linear ($\mathcal{S}_{as}$ is nonempty by Assumption 2.1).

We also need to introduce the continuous variables $z_i^0 \in \mathbb{R}^n$, where $z_i^0 = x(0)$ if $i_0 = i$ or zero otherwise,

$$z_i^0 = x_0 \gamma_i^0, \; \forall i \in \mathcal{S}, \tag{26}$$

and the continuous variables $z_{i,j}^k \in \mathbb{R}^n$, $j \in \mathcal{S}$, $k = 1, \ldots, N$, where $z_{i,j}^k = x(\tau_k^+)$ if $i_{k-1} = i$, $i_k = j$, or zero otherwise:

$$\begin{aligned} z_{i,j}^k \quad &= \quad J_{i,j}(\bar{A}_i(k) x_k + \bar{f}_i(k)) \gamma_i^{k-1} \gamma_j^k, \\ & \qquad \forall k = 1, \ldots, N, \; \forall i, j \in \mathcal{S}, \end{aligned} \tag{27a}$$

$$x_k \quad = \quad \textstyle\sum_{i,j=1}^{s} z_{i,j}^k, \quad \forall k = 1, \ldots, N. \tag{27b}$$

Constraint (26) can be transformed into the following set of mixed-integer linear inequalities by using the so-called "big-M" technique [34], [6]:

$$\begin{aligned} z_i^0 &\leq x_0 + M(1 - \gamma_i^0), \; i \in \mathcal{S}, \\ -z_i^0 &\leq -x_0 + M(1 - \gamma_i^0), \; i \in \mathcal{S}, \\ z_i^0 &\geq -M \gamma_i^0, \\ z_i^0 &\leq M \gamma_i^0 \end{aligned} \tag{28}$$

where $M \in \mathbb{R}^n$ is an upper bound on the state vector x (more precisely, the j-th component M^j of M is an upper bound on $|x^j|$, where x^j is the j-th component of the state vector), and therefore an upper bound on x_0 and on $x_{k+1}^{i,j} = J_{i,j}(A_i(k) x_k + \bar{f}_i(k))$, for all $k = 0, \ldots, N-1$, $i, j \in \mathcal{S}$, where $x_{k+1}^{i,j}$ would be the state at time τ_{k+1} if at time τ_k the system switches from the ith to the jth mode. Usually a suitable M can be chosen on the basis of physical considerations on the switched system.

Constraint (27a) can be also transformed into a set of mixed-integer linear inequalities by generalizing the above transformation to the case of the product of an affine function of continuous variable and two integer variables as in (27a), namely by transforming the conditions

$$[\gamma_i^{k-1} = 0] \vee [\gamma_j^k = 0] \;\; \rightarrow [z_{i,j}^k = 0], \tag{29a}$$

$$[\gamma_i^{k-1} = 1] \wedge [\gamma_j^k = 1] \;\; \rightarrow \;\; \begin{array}{r} [z_{i,j}^k = J_{i,j}(\bar{A}_i(k) x_k \\ + \bar{f}_i(k))], \end{array} \tag{29b}$$

where $\vee$, $\wedge$ denote the logic "or" and "and", respectively, into the linear mixed-integer inequalities

$$z_{i,j}^k \le M\gamma_i^{k-1}, \quad (30a)$$
$$z_{i,j}^k \le M\gamma_j^k, \quad (30b)$$
$$-z_{i,j}^k \le M\gamma_i^{k-1}, \quad (30c)$$
$$-z_{i,j}^k \le M\gamma_j^k, \quad (30d)$$
$$z_{i,j}^k \le J_{i,j}(\bar{A}_i(k)x_k + \bar{f}_i(k)) + M(2-\gamma_i^{k-1}-\gamma_j^k), \quad (30e)$$
$$-z_{i,j}^k \le -J_{i,j}(\bar{A}_i(k)x_k - \bar{f}_i(k)) + M(2-\gamma_i^{k-1}-\gamma_j^k), \quad (30f)$$
$$\forall k = 1,\dots,N,\ \forall i,j \in \mathcal{S}.$$

Note that in the absence of resets ($J = I$), it is not necessary to introduce Ns^2 real vectors $z_{i,j}^k$, as by proceeding as in (26),(28) Ns vectors z_i^k would be sufficient.

Finally, constraints (25b)–(25c) can be expressed as

$$\begin{aligned}&\sum_{i=1}^s \gamma_i^k = 1, \quad \forall k = 0,\dots,N,\\ &\gamma_i^N = 0, \quad \forall i \notin \mathcal{S}_{as}.\end{aligned} \quad (31)$$

The terms $H_{i_k,i_{k+1}}$ in (23) can be instead expressed as a quadratic function of the γ variables,

$$H_{i_k,i_{k+1}} = \begin{bmatrix}\gamma_k^1\\ \cdots \\ \gamma_k^s\end{bmatrix}' \begin{bmatrix} 0 & H_{12} & \dots & H_{1s}\\ H_{21} & 0 & \dots & H_{2s}\\ \vdots & \vdots & \ddots & \vdots\\ H_{s1} & H_{s2} & \dots & 0\end{bmatrix} \begin{bmatrix}\gamma_{k+1}^1\\ \cdots \\ \gamma_{k+1}^s\end{bmatrix}. \quad (32)$$

Note that (32) only contains terms $\gamma_k^i, \gamma_{k+1}^j$, and therefore the cost term $\sum_{k=0}^N H_{i_k,i_{k+1}}$ leads to a nonconvex quadratic function of the vector of integer variables. In order to overcome this issue[2], we add a constant term K in the cost function in (23), $K \in \mathbb{R}$. This clearly does not change the optimal switching sequence. Then, by virtue of (31), we substitute

$$\begin{aligned} K &= \sum_{\substack{k=0\dots N\\ i=1\dots s}} \frac{K}{(N+1)}\gamma_k^i && (33a)\\ &= \sum_{\substack{k=0\dots N\\ i=1\dots s}} \frac{K}{(N+1)}(\gamma_k^i)^2 && (33b)\end{aligned}$$

and choose K large enough so that the overall cost function is a convex function.

Summing up, the master problem (23) is equivalent to the MIQP

$$\begin{aligned}\min_{\substack{x_k,\gamma_i^k,z_i^k\\ k=1,\dots,N+1\\ i=1,\dots,s}} \ & \sum_{i=1}^s \left[(z_i^0)'\bar{Q}_i(0)z_i^0 + \bar{c}_i(0)z_i^0 + \bar{\alpha}_i(0)\gamma_i^0\right]\\ & + \sum_{k=1}^N \sum_{i=1}^s \bar{\alpha}_i(k)\gamma_i^k + \sum_{j=1}^s \left[(z_{i,j}^k)'\bar{Q}_i(k)z_{i,j}^k \right.\\ & \left. + \bar{c}_i(k)z_{i,j}^k\right] + (32) + (33b)\\ \text{s.t.} \quad & (27b),\ (28),\ (30),\ (31).\end{aligned} \quad (34)$$

[2]Mixed-integer quadratic programming solvers determine the solution by solving a sequence of standard quadratic programs in which the integer variables are either fixed or relaxed between the whole interval $[0,1]$. Such QP problems must be convex problems in order to be solved efficiently.

Note that additional constrains on the possible mode switches can be easily embedded in (34) as linear integer constraints. For example, $[\gamma_i^{k-1} = 1] \rightarrow [\gamma_j^k = 0]$ is equivalent to $\gamma_i^{k-1} - (1-\gamma_j^k) \le 0$.

Another interesting case is that in which the initial mode $i_0 = \hat{\imath}$ is assigned. In this case we add the constraint $\gamma_{\hat{\imath}}^0 = 1$.

B. Slave Algorithm

The slave algorithm has been extensively discussed in the previous Section 3. Note however that in the problem formulation (6) an integer variable $m \le N$ has been introduced to keep into account the number of switches effectively occurring in an optimal solution. Now, a sequence of τ's, namely, $\{\tau_1,\dots,\tau_N\}$, of fixed length N is needed to iterate between the two procedures. This has two important consequences.

- Firstly, in the case of $m < N$, we complete the time sequence by simply taking $\tau_{m+1} = \ldots = \tau_N = +\infty$.
- Secondly, all switching costs H_k, $k = 1,\dots,N$ contribute to the performance index $F(\mathcal{T})$. As a consequence, the term $\sum_{k=1}^m H_k$ in (6) should be replaced by the *constant* term $\sum_{k=1}^N H_k$. Thus, being this term constant, it can be obviously neglected when solving the slave phase.

C. Master-Slave Algorithm

The proposed master-slave algorithm is structured as follows:

Algorithm 4.1:

1. Initialize $\mathcal{T}(0) \leftarrow \{\tau_1,\dots,\tau_N\}$, $k = 1$, $I(0) =$, $\{-1,\dots,-1\}$;
(e.g., τ_k are randomly or uniformly distributed); Let $\epsilon > 0$ a given tolerance;
2. Solve the master problem $I(k) \leftarrow f_M(\mathcal{T}(k-1))$;
3. If
$$|F(\mathcal{T}(k-1), I(k)) - F(\mathcal{T}(k-1), I(k-1))| \le \epsilon \quad (35)$$
set $\mathcal{T}(k) \leftarrow \mathcal{T}(k-1)$ and go to 7.
4. Solve the slave problem $\mathcal{T}(k) \leftarrow f_S(I(k))$;
5. $k \leftarrow k+1$;
6. Go to 2.;
7. Set $\{\tau_1,\dots,\tau_N\} \leftarrow \mathcal{T}(k)$, $\{i_0,\dots,i_N\} \leftarrow I(k)$;
8. End

Proposition 4.1: Algorithm 4.1 stops after a finite number of steps N_{stop}.

Proof: Let $V(k) \triangleq F(\mathcal{T}(k), I(k))$. Clearly,

$$\begin{aligned}V(k-1) = F(\mathcal{T}(k-1), I(k-1)) &\ge F(\mathcal{T}(k-1), I(k))\\ &\ge F(\mathcal{T}(k), I(k)) = V(k).\end{aligned}$$

Since $\{V(k)\}$ is a monotonically nonincreasing sequence bounded between $V(0)$ and 0, it admits a limit as $k \to \infty$. Therefore, $V(k) - V(k-1) \to 0$ as $k \to \infty$, and hence (35) is satisfied after a finite number k_ϵ of iterations for any given positive tolerance ϵ. ■

Definition 4.1: The optimal control problem (2) is said *switch-degenerate* if there exist a sequence $\mathcal{T}$ and $I_1 \neq I_2$ such that $F(I_1, \mathcal{T}) = F(I_2, \mathcal{T})$. ■

Definition 4.2: The optimal control problem (2) is said *time-degenerate* if there exist a sequence I and $\mathcal{T}_1 \neq \mathcal{T}_2$ such that $F(I, \mathcal{T}_1) = F(I, \mathcal{T}_2)$. ■

Note that the result of Proposition 4.1 also holds in the case of degeneracy.

We remark that although Algorithm 4.1 converges to a solution $(I, \mathcal{T})$ after a finite number N_{stop} of steps, such a solution may not be the optimal one, as it may be a local minimum where both the master and the slave problems do not give any further improvement. Note that the global solution can be computed by enumeration by solving a slave problem for all possible s^N switching sequences I.

Algorithm 4.1 computes the optimal switching policy for a given initial state. On the other hand, for small enough perturbations of the initial state such that the optimal switching sequence does not change, the optimal time-switching policy is immediately available as a by-product of the slave algorithm, because of its state-feedback nature.

We finally remark that Algorithm 4.1 may be formulated by optimizing with respect to $\mathcal{T}$ first, for a given initialization of the switching sequence I. The advantage of switching between the master and slave procedures depends on the information available a priori about the optimal solution. For instance when the algorithm is solved repeatedly for subsequent values of the state vector (such as in a receding horizon scheme), it may be useful to use the previous switching sequence as a warm start and optimize with respect to $\mathcal{T}$ first.

We finally remark the following about degeneracies:

1) Time-degeneracy: $i_k = i_{k-1}$ implies that the switching instant τ_k is undetermined (multiple solutions for $\mathcal{T}$)
2) Switch-degeneracy: $\tau_k = \tau_{k-1}$ implies that mode i_k is undetermined (multiple solutions for I)

Ways to handle such degenerate cases will be highlighted in the next section.

D. Numerical Examples

Example 1

Consider a second order linear system whose dynamics may be chosen within a finite set $\{A_1, A_2, A_3\}$, where

$$A_1 = \begin{bmatrix} -5.179 & -1.414 \\ 1 & 0 \end{bmatrix},\ A_2 = \begin{bmatrix} -10.115 & -3.082 \\ 2 & 0 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} -2.414 & -1.414 \\ 1 & 0 \end{bmatrix}.$$

We associate to each dynamics a weight matrix:

$$Q_1 = \text{diag}\{1,\ 1\}, \quad Q_2 = \text{diag}\{8,\ 2\}, \quad Q_3 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}.$$

As only three modes are possible, $s = 3$, the control variable $i(t)$ only takes values from the finite set of integers $\mathcal{S} = \{1, 2, 3\}$. Let the initial state vector be $x_0 = [1\ \ 1]'$ and $N = 3$ be the number of allowed switches.

step		τ_1	τ_2	τ_3	i_0	i_1	i_2	i_3	$F(I, \mathcal{T})$
1	M	0.290	0.498	0.672	1	3	3	3	1.44619
1	S	0.280	0.290	0.300	1	3	3	3	1.44615
2	M	0.280	0.290	0.300	1	2	3	3	1.44459
2	S	0.180	0.240	0.240	1	2	3	3	1.44026
3	M	0.180	0.240	0.240	1	2	3	3	1.44026

TABLE I
RESULTS OF EXAMPLE 1.

Algorithm 4.1 is applied to determine the optimal mode and timing sequence with the initial timing sequence $\mathcal{T}_0 = \{0.290, 0.498, 0.672\}$ (randomly generated). The resulting optimized mode sequence is $I^* = \{1, 2, 3, 3\}$ and the optimal cost value is $V_3^* = 1.44026$. Note that in this case only two switches are required to get the optimal cost value.

Detailed intermediate results are reported in Table 1, where one may also observe that the procedure converges after only 3 steps. This also implies that the most burdensome part of the algorithm, i.e., the slave problem, is only solved twice.

The correctness of the solution has been validated through an exhaustive inspection of all admissible mode sequences. More precisely, for each admissible mode sequence the corresponding optimal timing sequence and cost value were computed by the slave algorithm. As a result, it turns out that $V_3^* = 1.44026$ is indeed a global minimum. Obviously, being only two the switches required to optimize the cost value, the minimum cost may also be obtained by using other mode sequences. As an example, $I = \{3, 1, 2, 3\}$ and $\mathcal{T} = \{0, 0.180, 0.240\}$ is an optimal solution as well.

Several random tests highlighted that the convergence of the algorithm to a global minimum is heavily influenced by two factors. First, the initial switching times sequence should be such that $\tau_k > \tau_{k-1}$. In fact, if $\tau_k = \tau_{k+1}$ for some k, only a suboptimal solution — that corresponds to a smaller number of switches — is usually computed. Second, the first switching time should not be greater than two or three times the maximum time constant associated to each dynamics: if this is not the case, only degenerate solutions with no switch are usually found.

Example 2

We present here an heuristics that in many cases improves the performance of the Algorithm 4.1. Consider the second order switched linear system with dynamic matrices

$$A_1 = \begin{bmatrix} 1 & -10 \\ 100 & 1 \end{bmatrix},\ A_2 = \begin{bmatrix} 1 & -100 \\ 10 & 1 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} -0.1 & 0 \\ 0 & -0.1 \end{bmatrix}$$

($f_1 = f_2 = f_3 = 0$) and let $Q_1 = Q_2 = Q_3 = I$, $N = 3$, $x_0 = [1\ \ 1]'$. Note that while A_1 and A_2 are unstable matrices, A_3 is strictly Hurwitz, so that Assumption 2.1 is satisfied.

We take as initial timing sequence $\mathcal{T}_0 = \{0.001, 0.005, 0.010\}$ and apply the master-slave algorithm to determine the optimal mode sequence. The provided solution is $I = \{1, 1, 2, 3\}$ and the corresponding performance index is $V_3 = 1.42998$. Detailed results are reported in Table II.

step		τ_1	τ_2	τ_3	i_0	i_1	i_2	i_3	F(I,T)
1	M	0.001	0.005	0.010	2	2	2	3	5.63017
1	S	0.000	0.000	0.009	2	2	2	3	5.63017
2	M	0.000	0.000	0.009	1	1	2	3	5.63017
2	S	0.000	0.089	0.143	1	1	2	3	1.42998
3	M	0.000	0.089	0.143	1	1	2	3	1.42998

TABLE II
RESULTS OF EXAMPLE 2 WHEN THE MASTER-SLAVE ALGORITHM IS APPLIED IN ITS ORIGINAL FORM.

step		τ_1	τ_2	τ_3	i_0	i_1	i_2	i_3	F(I,T)
1	M	0.001	0.005	0.010	2	2	2	3	5.63017
1	S	0.000	0.089	0.143	3	1	2	3	1.42998
2	M	0.000	0.089	0.143	1	1	2	3	1.42998
2	S	0.009	0.062	0.116	2	1	2	3	0.12569
3	M	0.009	0.062	0.116	2	1	2	3	0.12569

TABLE III
RESULTS OF EXAMPLE 2 WHEN THE MASTER-SLAVE ALGORITHM IS APPLIED IN ITS MODIFIED FORM.

Nevertheless, this solution is not optimal and this may be easily verified through an exhaustive inspection of all admissible switching sequences.

A careful examination of the solution suggests the presence of time-degeneracy, being $I = \{1, 1, 2, 3\}$ a switching sequence that corresponds to only two switches. Thus, when it is used by the slave algorithm, it may only compute a suboptimal solution.

A simple heuristic solution to this problem — that is effective in this case, as well as in many other instances that were examined — consists of modifying the switching sequence computed via the master algorithm that corresponds to a number of switches that is less that N before running the slave algorithm. In particular, we suggest to arbitrarily change the mode sequence so that the original sequence is still contained in the new one but two consecutive indices are never the same.

Using such an heuristics, the results reported in Table III were obtained. In particular, observe that at the first step of the whole procedure the slave algorithm does not examine the switching sequence firstly computed by the master algorithm, but computes the optimal timing sequence corresponding to a new sequence $I = \{3, 1, 2, 3\}$, that has been randomly generated by arbitrarily modifying the first mode so as to avoid time-degeneracy. At this step, the value of the performance index decreases but the optimum is not computed yet. The same reasoning is repeated at the third step and in this case the optimal value of the cost is found and the procedure stops. The results of an exhaustive search show that the computed solution is optimal thus revealing the effectiveness of the modified procedure.

Although the above heuristics is effective in most instances, a condition is required that stops the algorithm whenever a loop is detected, so that cycling is avoided.

V. SWITCHING TABLE PROCEDURE

Let us consider again the optimal control problem (2) or equivalently (3) and assume that Assumption 1 is satisfied. In this section we show how to solve this problem generalizing the procedure derived in Section 3 to compute the optimal control law when the sequence is fixed. Again for simplicity we assume that dynamics are linear, because affine dynamics can be easily reduced to linear dynamics as shown in (5).

In particular we show that for a given mode $i \in \mathcal{S}$ and for a given switch $k \in 1, \ldots, N$ it is possible to construct a table $\mathcal{C}_k^i$ that partitions the state space $\mathbb{R}^n$ into s regions $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_s$. 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 $\mathcal{R}_j$ (with $j \neq i$) we will switch to mode $i_k = j$, while no switch will occur while the system's state belongs to $\mathcal{R}_i$.

A. Computation of the Switching Tables

In this section we denote $F_{k,i}(x)$ the optimal cost to infinity of an evolution that starts with dynamics $i_k = i$ from the state x. We will show later how this value can be computed.

We can also define the following cost functions.

Definition 5.1: Let us assume that $i_k = i$, i.e., after k switches the current system dynamics is that corresponding to matrix A_i, and let the current state vector be x.

- For $k \in \{0, \ldots, N\}$ and $i \in \mathcal{S}$ we define:
$$T_{k,i}^*(x, i) = x'\bar{Q}_i(+\infty)x \tag{36}$$
the remaining cost of an evolution that starts with dynamics A_i from x and never switches and we denote
$$\varrho_{k,i}(x, i) = +\infty. \tag{37}$$
- For $k \in \{0, \ldots, N-1\}$ and $i, j \in \mathcal{S}$ with $i \neq j$, we define
$$T_{k,i}(x, j, \varrho) = x'\bar{Q}_i(\varrho)x + F_{k+1,j}(J_{i,j}e^{A_i\varrho}x) + H_{i,j} \tag{38}$$
the remaining cost of an evolution that starts with dynamics A_i from x, switches after a time ϱ to A_j and from then on follows an optimal evolution.
We also denote
$$\varrho_{k,i}(x, j) = \arg\min_{\varrho \geq 0} T_{k,i}(x, j, \varrho), \tag{39}$$
the value of ϱ that minimizes (38) while the corresponding minimum is
$$T_{k,i}^*(x, j) = T_{k,i}(x, j, \varrho_{k,i}(x, j)). \tag{40}$$

■

Proposition 5.1 (Optimal remaining cost): Let us assume that $i_k = i$, i.e., after k switches the current system dynamics is that corresponding to matrix A_i, and let the current state vector be $x = \lambda y$ with $\|y\| = 1$ and $\lambda \in \mathbb{R} \setminus \{0\}$.

1) If $k = N$ then the remaining optimal cost starting from x is:
$$F_{N,i}(x) = T_{N,i}^*(x, i). \tag{41}$$
2) If $k \in \{0, \ldots, N-1\}$ then:

i) the remaining optimal cost starting from x is:

$$F_{k,i}(x) = \min_{j\in\mathcal{S}} T^*_{k,i}(x,j); \quad (42)$$

ii) the next dynamics reached by the optimal evolution is

$$j_{k+1,i}(x) = \arg\min_{j\in\mathcal{S}} T^*_{k,i}(x,j) \quad (43)$$

where $j_{k+1,i}(x) = i$ means that no other switch will occur;

iii) the optimal evolution switches to $A_{j_{k+1,i}}(x)$ at time $\tau_{k+1} = t + \delta_{k+1,i}(x)$, where

$$\delta_{k+1,i}(x) = \varrho_{k,i}(x, j_{k+1,i}(x)). \quad (44)$$

Proof:

If $k = N$ the systems is forced to evolve with dynamics A_i to infinity and the remaining cost (that is also optimal) is the one given in equation (41).

If $k<N$, we have two options. If no future switch occurs then the remaining cost will be $T^*_{k,i}(x,i)$. If at least a future switch will occur, the two decision variables are the time before the first switch occurs (parameter $\varrho \geq 0$) and the dynamics reached after the switch (parameter $j \in \mathcal{S}\setminus\{i\}$), while after the switch it is necessary to follow an optimal evolution such that the remaining cost is minimized. Hence:

$$\begin{aligned} F_{k,i}(x) &= \min\{T^*_{k,i}(x,i), \min_{\substack{j\in\mathcal{S}\setminus\{i\}\\ \varrho\geq 0}} \{T_{k,i}(x,j,\varrho)\}\} \\ &= \min\left\{T^*_{k,i}(x,i), \min_{j\in\mathcal{S}\setminus\{i\}} \{T^*_{k,i}(x,j)\}\right\} \\ &= \min\nolimits_{j\in\mathcal{S}} T^*_{k,i}(x,j). \end{aligned}$$

□

According to the previous proposition, the optimal remaining cost can be computed recursively, first computing for all vectors $x \in \mathbb{R}^n$ and all dynamics $i \in \mathcal{S}$ the costs $F_{N,i}(x)$, then the costs $F_{N-1,i}(x)$, etc.

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

Proposition 5.2: Assume that all switching costs are zero, i.e., $H_{i,j} = 0$ for all $i,j \in \mathcal{S}$. If x is a vector such that $x = \lambda y$, with $\|y\|_2 = 1$ and $\lambda \in \mathbb{R}\setminus\{0\}$, with the notation of Definition 5.1 we have that for all $k \in \{0,\dots,N\}$ and all $i,j \in \mathcal{S}$

$$(a) \qquad T^*_{k,i}(x,j) = \lambda^2 T^*_{k,i}(y,j), \quad (45)$$

$$(b) \qquad \varrho_{k,i}(x,j) = \varrho_{k,i}(y,j), \quad (46)$$

$$(c) \qquad F_{k,i}(x) = \lambda^2 F_{k,i}(y), \quad (47)$$

Proof: Clearly, according to equations (36) and (37), the results (a) and (b) hold for $k = N$ and $i = j$. This also implies, by equation (41), that the result (c) holds for $k = N$.

We now recursively show that results (a)-(c) hold for all values of k. Assume in fact that $F_{k,i}(x) = \lambda^2 F_{k,i}(y)$. By equations (38) and (40), it also holds that $T^*_{k-1,i}(x,j) = \lambda^2 T^*_{k-1,i}(y,j)$, hence by equation (39), $\varrho_{k-1,i}(x,j) = \varrho_{k-1,i}(y,j)$ and finally by equation (42) $F_{k-1,i}(x) = \lambda^2 F_{k-1,i}(y)$. □

This proposition implies that when all switching costs are zero to determine the optimal costs it is sufficient to evaluate the functions $F_{k,i}(x)$ only for vectors x on the unitary semi-sphere.

We can finally extend the definition of table given in Section 2.

Definition 5.2: The switching table $\mathcal{C}^i_k$ is a partition of the state space in s regions $\mathcal{R}_j$ (for $j \in \mathcal{S}$) defined as follows

- The region $\mathcal{R}_j = \{x \in \mathbb{R}^n \mid \delta_{k,i}(x) = 0, j_{k,i}(x) = j \neq i\}$ is the set of points where it is optimal to switch from $i_{k-1} = i$ to $i_k = j \neq i$.
- The complementary region is $\mathcal{R}_i = \mathbb{R}^n \setminus \bigcup_{j\neq i} \mathcal{R}_j$. ■

B. Computation of the Table for the Initial Mode

To decide the optimal initial mode i_0 we may use the knowledge of the cost $F_{0,i}(x)$ (i.e., of the optimal cost to infinity starting from state x with dynamics $i_0 = i$) that is evaluated during the construction of the table $\mathcal{C}^i_1$.

Definition 5.3: Table $\mathcal{C}_0$ 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: $\mathcal{R}_i = \{x \mid (\forall j \in \mathcal{S}) F_{0,i}(x) \leq F^*_{0,j}(x)\}$. ■

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.

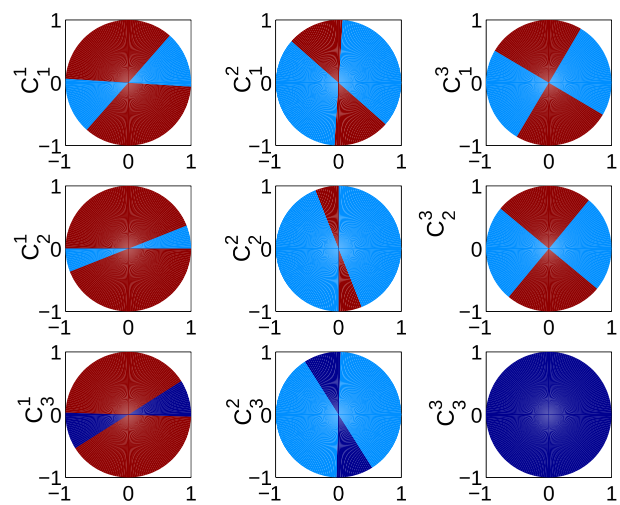

Fig. 4. The set of tables for the numerical example where $N = 3$ and $\mathcal{S} = \{1,2,3\}$.

C. Structure of the Switching Regions

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

Proposition 5.1: Consider the case in which $H_{i,j} = 0$ for all $i,j \in \mathcal{S}$. Then any region $\mathcal{R}_j$ of table $\mathcal{C}^i_k$ and of table $\mathcal{C}_0$ is such that $y \in \mathcal{R}_j \implies (\forall\lambda \in \mathbb{R})\ \lambda y \in \mathcal{R}_j$, i.e., the region $\mathcal{R}_j$ is homogeneous.

Proof: When all switching costs are zero, we have shown that equations (45) and (46) hold. Thus, is immediately follows that in this case $j_{k,i}(x) = j_{k,i}(y)$ and $\delta_{k,i}(x) = \delta_{k,i}(y)$. By Definition 5.2 this implies that all regions of table $\mathcal{C}^i_k$ are homogenous for $k = 1,\dots,N$.

The table used to select the initial mode has the same property. In fact, assume equation (47) holds: taking (as a particular case) $k = 0$ one can see that by Definition 5.3 the regions of table $\mathcal{C}_0$ are homogenous as well. □

D. Numerical Examples

Let us consider again the second order linear system considered in Subsection 4.3, Example 2.

We first execute the off-line 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 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 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 A_2, then the system may either switch to A_1 or still evolve with the same dynamics A_2: on the contrary a switch to dynamics A_3 may never occur.

In Figure 5 we have reported table $\mathcal{C}_0$ that shows the partition of the state space introduced in Subsection 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 A_1 or A_2. On the contrary, whenever the initial system dynamics is 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 $x_0 = [1 \;\; 1]'$. We compute the optimal mode sequence for all admissible initial system dynamics, i.e., we assume $i_0 = 1, 2, 3$, respectively. Detailed results may be read in Table IV 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 5 being $x_0 \in \mathcal{R}_1$.

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 $V_3^* = 0.126$ 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 IV
RESULTS OF THE NUMERICAL EXAMPLE WHEN THE INITIAL STATE IS $x_0 = [1 \;\; 1]'$.

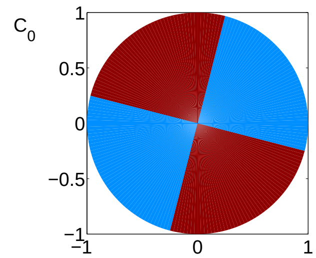

Fig. 5. Table $\mathcal{C}_0$.

VI. Discussion on computational complexity

A. Theoretical analysis

Let us first discuss the computational complexity involved in the construction of the tables for a fixed mode sequence in terms of operations required. In this case, a single table $\mathcal{C}_k$ is associated to the k-th switch and it contains only two regions: $\mathcal{R}_{i_{k-1}}$, i.e., the set of state vectors where we continue using mode i_{k-1}, and $\mathcal{R}_{i_k}$, i.e., the set of state vectors where we switch to mode i_k.

We have to distinguish two cases.

- Assuming that not all switching costs are zero, it is necessary to discretize the state space within a region centered around the origin and large enough to contains all vectors of length smaller than or equal to $\tilde{\lambda}$ as defined in eq. (20). The two regions of the table can be determined by solving a one-parameter optimization problem for each discretization point. If the state space is $\mathbb{R}^n$ and we assume a uniform sampling with r samples along each dimension the complexity for constructing each table is $r^n c$, being c the number of operations required to find the value of ϱ that minimizes (10).
- On the contrary, if all switching costs are zero it is sufficient to grid the unitary semisphere. In this case we can assume that the complexity for constructing each table is $r^{n-1}c$.

Thus, the complexity of solving the optimal control problem for a fixed sequence of length $N+1$ is $f_1(N) = N r^{n-1} c$ or $f_1'(N) = N r^n c$, in the case of zero and non-zero switching costs, respectively, because for each switch a new table must be determined.

It is important to observe that the choice of the number of samples r is a trade-off. The proposed procedure can always be applied even when the number of samples is small: in this case, however, the switching sequence that leads to the final stable mode will most likely only be sub-optimal. By increasing the number of samples one can get as close as desired to the optimal solution.

When the switching sequence is not fixed, using the switching table procedure STP given in Section 5, for each switch it is necessary to compute s tables, one for each possible mode. Now, for sake of simplicity assume that all switching costs are zero. The complexity of computing the generic table $\mathcal{C}_k^i$ is $(s-1)r^{n-1}c$. In fact each table contains s regions that can be determined solving $s-1$ one-parameter optimization problems for each vector y on the unitary semi-sphere. Thus the complexity of solving the optimal control problem (2) for a sequence of length $N+1$ is $f_2(N,s) = Ns(s-1)r^{n-1}c$. On the contrary, if not all switching costs are zero, following the same argument one can immediately show that the complexity is $f_2'(N,s) = Ns(s-1)r^n c$. In any case, the complexity is a quadratic function of the number of possible dynamics.

Note that the amount of data required to construct the switching tables is equal to $g(s) = 2sr^{n-1}$ or $g'(s) = 2sr^n$, depending on the switching costs. In fact, when computing the tables relative to the k-th switch we only look at the s tables relative to the $k+1$-th switch, where for each grid point we keep track of two values: the region it belongs to and the optimal remaining costs when $N-k+1$ switches are available. Note however that once the switching tables have been constructed (the off-line part of the procedure is finished) we do not need to keep memory of the optimal remaining costs.

Let us finally discuss the computational complexity of the master-slave procedure MSP. The complexity of solving MSP is $f_3(N) = \alpha f_1(N) + d = \alpha N r^{n-1} c + d$, where α is the number of times the slave procedure is executed and d is the cost for solving the master MIQP problems. Note in fact, that the slave algorithm is always invoked with zero switching costs. In theory, $d = d(N,s)$ and in the worst case it grows exponentially with N and s. Parameter α depends in general on all the data of the problem. However, for problems with relatively small s and N, as those shown in the paper, it turns out that $d \ll \alpha N r^{n-1} c$ and $\alpha \leq 5$.

From all these considerations, one may conclude that from a computational point of view MSP offers the best performance. For all other aspects, STP is better. In fact STP always finds an optimal solution while MSP may converge to a local minimum. Furthermore, the solutions provided by MSP are local, i.e., the optimal sequence and the corresponding tables are valid for a *given* initial state $x(0)$ and for bounded disturbances. On the contrary, the tables constructed by STP provide the optimal state feedback law for all initial states.

B. Numerical simulations

We conclude this section with a brief presentation of some numerical examples randomly generated, in order to provide a better idea of the time required to perform the off-line phase of the STP. Calculations have been done using the software Matlab 7, on an Intel Pentium 4 with 2 GHz and 256 Mb RAM.

We consider 3 different switched linear systems composed of 4 stable dynamics. The state space of the three systems has dimension $n = 2, 3, 4$, respectively. No cost is associated to the switches, thus we used a discretization of the unitary semisphere as explained in Appendix B. We compare the time requested to compute a single table assuming that the sequence is not fixed. We need not specify the maximum number of allowed switches N because the time spent for the computation of each table does not depend on N or on the remaining number of switches.

To compare the three cases, the optimal value of ϱ that minimizes the remaining cost (see eq. (39)) has been determined by looking at a finite time horizon: $0 \leq \varrho \leq \varrho_{\max} = 3$ that is the same for all systems. A finite uniformly distributed number of ϱ are considered: $\varrho = k \cdot \Delta\varrho$, with $\Delta\varrho = 0.01$ and $k = 0, 1, \ldots, 300$.

dimension	time [s]	parameters of the discretization	samplings
$n=2$	32	$N_\theta = 101$	101
$n=3$	3593	$N_\theta = 120$, $N_\varphi = 30$	2293
$n=4$	25978	$N_\theta = 60$, $N_\varphi = 30$, $N_\xi = 15$	8581

TABLE V

RESULTS OF THE NUMERICAL EXAMPLES IN SUBSECTION VI-B.

The time (in seconds) required to compute a switching table for all the 3 cases examined are given in Table V. In this table we have also reported, using the notation of Appendix B, the parameters relative to the considered discretization of the unitary semi-sphere.

VII. CONCLUSIONS

We have considered a simple class of switched piecewise affine autonomous linear systems with the objective of minimizing a quadratic performance index over an infinite time horizon. We have assumed that the switching sequence has a finite length, and that the decision variables are the switching instants and the sequence of operating modes.

We have presented two different approaches for solving such an optimal control problem. The first approach iterates between a "master" procedure that finds an optimal switching sequence of modes, and a "slave" procedure that finds the optimal switching instants. The second approach is inspired by dynamic programming and allows one to compute a state feedback control law, i.e., it is possible to identify the regions of the state space where an optimal switch should occur whenever the state trajectory enters them.

There are several ways in which this research may be extended.

Firstly, we will consider the case in which an *infinite* number of switches may occur.

Secondly, we will consider the optimal control of switched systems whose switching sequence is determined by a controlled automaton whose discrete dynamics restrict the possible switches from a given location to an adjacent location.

Finally, we may assume that in the automaton there are two types of edges. A controllable edge represents a mode switch that can be triggered by the controller; an autonomous edge represents a mode switch that is triggered by the continuous state of the system as it reaches a given threshold.

REFERENCES

[1] A. Bemporad, F. Borrelli, and M. Morari. Piecewise linear optimal controllers for hybrid systems. In *Proc. American Contr. Conf.*, pages 1190–1194, Chicago, IL, June 2000.

[2] A. Bemporad, F. Borrelli, and M. Morari. On the optimal control law for linear discrete time hybrid systems. In M. Greenstreet and C. Tomlin, editors, *Hybrid Systems: Computation and Control*, number 2289 in Lecture Notes in Computer Science, pages 105–119. Springer-Verlag, 2002.

[3] 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.

[4] A. Bemporad, A. Giua, and C. Seatzu. A master–slave algorithm for the optimal control of continuous–time switched affine systems. In *Proc. 41th IEEE Conf. on Decision and Control*, pages 1976–1981, Las Vegas, Nevada, December 2002.

[5] A. Bemporad, A. Giua, and C. Seatzu. Synthesis of state–feedback optimal controllers for switched linear system. In *Proc. 41th IEEE Conf. on Decision and Control*, pages 3182–3187, Las Vegas, Nevada, December 2002.

[6] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.

[7] S.C. Bengea and R.A. DeCarlo. Conditions for the existence of a solution to a two-switched/hybrid optimal control problem. In *IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 419–424, Saint Malo, France, June 2003.

[8] 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 *Proc. American Contr. Conf.*, Denver, Colorado, 2003.

[9] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: model and optimal control theory. *IEEE Trans. Automatic Control*, 43(1):31–45, 1998.

[10] M.S. Branicky and S.K. Mitter. Algorithms for optimal hybrid control. In *Proc. 34th IEEE Conf. on Decision and Control*, New Orleans, USA, December 1995.

[11] C.G. Cassandras, D.L. Pepyne, and Y. Wardi. Optimal control of a class of hybrid systems. *IEEE Trans. Automatic Control*, 46(3):398–415, 2001.

[12] Dash Associates. *XPRESS-MP User Guide*, 2003. `http://www.dashopt.com`.

[13] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear systems. In *Proc. American Contr. Conf.*, pages 4046–4050, 2000.

[14] 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.

[15] P.J. Antsaklis (ed.). Special issue on hybrid systems: Theory and applications. *Proceedings of the IEEE*, 88(7), July 2000.

[16] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal control of switching times in swithced dynamical systems. In *Proc. 42th IEEE Conf. on Decision and Control*, pages 2138–2143, Maui, Hawaii, USA, December 2003.

[17] A. Giua, C. Seatzu, and C. Van Der Mee. Optimal control of autonomous linear systems switched with a pre–assigned finite sequence. In *Proc. 2001 IEEE Int. Symp. on Intelligent Control*, pages 144–149, Mexico City, Mexico, 2001.

[18] A. Giua, C. Seatzu, and C. Van Der Mee. Optimal control of switched autonomous linear systems. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 2472–2477, Orlando, Florida USA, 2001.

[19] K. Gokbayrak and C.G. Cassandras. A hierarchical decomposition method for optimal control of hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 1816–1821, Phoenix, AZ, December 1999.

[20] S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 3972–3976, Phoenix, AZ, December 1999.

[21] S. Hedlund and A. Rantzer. Convex dynamic programming for hybrid systems. *IEEE Trans. Automatic Control*, 47(9):1536–1540, September 2002.

[22] ILOG, Inc. *CPLEX 8.1 User Manual*. Gentilly Cedex, France, 2003.

[23] B. Lincoln and A. Rantzer. Optimizing linear system switching. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 2063–2068, 2001.

[24] B. Lincoln and A. Rantzer. Relaxed optimal control of piecewise linear systems. In *IFAC Conf. on Analysis and Design of Hybrid Systems*, Saint Malo, France, June 2003.

[25] C.C. Pantelides, M.P. Avraam, and N. Shah. Optimization of hybrid dynamic processes. In *Proc. American Contr. Conf.*, 2000. Available upon request from the authors.

[26] B. Piccoli. Necessary conditions for hybrid optimization. In *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, Arizona USA, December 1999.

[27] A. Rantzer and M. Johansson. Piecewise linear quadratic optimal control. *IEEE Trans. Automatic Control*, 45(4):629–637, April 2000.

[28] P. Riedinger, F. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 2093–2098, Phoenix, AZ, USA, December 1999.

[29] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[30] M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: analysis and zonal algorithm for trajectory and schedule optimization. In *Proc. 42th IEEE Conf. on Decision and Control*, pages 2144–2149, Maui, USA, December 2003.

[31] M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: optimization of switching times and combinatoric schedules. In *Proc. American Contr. Conf.*, Denver, USA, 2003.

[32] M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: optimization of trajectories, switching times and location schedules. In *6th Int. Workshop on Hybrid Systems: Computation and Control*, Prague, The Czech Republic, 2003.

[33] H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Proc. 38th IEEE Conf. on Decision and Control*, Phoenix, Arizona USA, December 1999.

[34] H.P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, Third Edition, 1993.

[35] X. Xu and P.J. Antsaklis. Optimal control of switched autonomous systems. In *Proc. 41th IEEE Conf. on Decision and Control*, Las Vegas, USA, 2002.

[36] X. Xu and P.J. Antsaklis. Optimal control of switched systems via non-linear optimization based on direct differentiations of value functions. *Int. J. Control*, 75(16/17):1406–1426, 2002.

[37] X. Xu and P.J. Antsaklis. Quadratic optimal control problems for hybrid linear autonomous systems with state jumps. In *Proc. American Contr. Conf.*, Denver, USA, 2003.

[38] 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.

[39] X. Xu and P.J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. Automatic Control*, 49(1):2–16, 2004.

[40] P. Zhang and C.G. Cassandras. An improved forward algorithm for optimal control of a class of hybrid systems. *IEEE Trans. Automatic Control*, 47(10):1735–1739, 2002.

Appendix A: Computation of the Cost Function Matrices

In the problem formulation (3) we assumed

$$\int_0^\delta x'(t)Q_i x(t)dt = x_0'\bar{Q}_i(\delta)x_0 + \bar{c}_i(\delta)x_0 + \bar{\alpha}_i(\delta)$$

for any initial state $x_0 = x(0)$, where

$$\begin{aligned}
\bar{Q}_i(\delta) &= \int_0^\delta e^{A_i' t}\ Q_i\ e^{A_i t}dt, \\
\bar{c}_i(\delta) &= 2f_i'\ \int_0^\delta \left(\int_0^t e^{A_i'\tau}d\tau\right) Q_i\ e^{A_i t}dt, \\
\bar{\alpha}_i(\delta) &= f_i'\ \left[\int_0^\delta \left(\int_0^t e^{A_i'\tau}d\tau\right) Q_i \left(\int_0^t e^{A_i\tau}d\tau\right) dt\right]\ f_i
\end{aligned}$$

In general cases it is not easy to provide analytical expressions for $\bar{Q}_i(\delta)$, $\bar{c}_i(\delta)$, and $\bar{\alpha}_i(\delta)$, thus numerical integration is needed. On the contrary, under appropriate assumptions on A_i and f_i, these analytical expressions can be easily determined. As an example, let us consider the following two cases.

- Assume A_i is strictly Hurwitz and $f_i = 0$. In such a case
$$\begin{aligned}
&\bar{Q}_i(\delta) = Z_i - e^{A_i'\delta} Z_i e^{A_i\delta}, \\
&\bar{c}_i(\delta) = 0, \\
&\bar{\alpha}_i(\delta) = 0,
\end{aligned}$$
where Z_i is the unique solution of the Lyapunov equation $A_i'Z_i + Z_iA_i = -Q_i$. The same computation is valid when the eigenvalues of A_i are all unstable.

- Assume that A_i is diagonalizable. In such a case, $A_i = T_i^{-1}\Lambda_i T_i$, where $\Lambda_i = \text{Diag}\{\lambda_i^1,\dots,\lambda_i^n\}$ and λ_i^j, $j = 1,\dots,n$ are the eigenvalues of A_i. We obtain:

$$\bar{Q}_i(\delta) = T_i' \left(\int_0^\delta e^{\Lambda_i t} (T_i^{-1})' \; Q_i \; T_i^{-1} e^{\Lambda_i t} dt \right) T_i,$$

$$\bar{c}_i(\delta) = 2 f_i' \; T_i' \left[\int_0^\delta \left(\int_0^t e^{\Lambda_i \tau} d\tau \right) (T_i^{-1})' \; Q_i \; \cdot \right.$$

$$\left. \cdot \; T_i^{-1} e^{\Lambda_i t} dt \right] T_i,$$

$$\bar{\alpha}_i(\delta) = f_i' \; T_i' \left[\int_0^\delta \left(\int_0^t e^{\Lambda_i \tau} d\tau \right) (T_i^{-1})' \; Q_i \; \cdot \right.$$

$$\left. \cdot \; T_i^{-1} \left(\int_0^t e^{\Lambda_i \tau} d\tau \right) dt \right] T_i \; f_i,$$

and it is straightforward to symbolically compute the integrals given the simple form the exponential of a diagonal matrix takes.

Appendix B: Sampling the "hyper" semi-sphere

The main computational effort in the construction of the switching tables is the discretization of the state space. The first step is 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 $\theta_2, \dots, \theta_n$. Given a point $x = [x_1 \; x_2 \; \dots \; x_n]'$, 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 $\rho_n = \|x\|$, $\rho_i = \rho_{i+1} \; \cos(\theta_i)$ for $i = n-1,\dots,2$. To describe $\mathbb{R}^n$, variables must range in: $\rho_n \in [0,+\infty)$, $\theta_2 \in [0, 2\pi)$, and $\theta_3, \dots, \theta_n \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. To describe the unitary "hyper" semi-sphere we choose $\rho_n = 1$, $\theta_2 \in [0, 2\pi)$ $\theta_3, \dots, \theta_{n-1} \in [-\frac{\pi}{2}, \frac{\pi}{2})$, 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). An equally 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;
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 $\bar{N}_\varphi =$ 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 $\bar{N}_\vartheta =$ round$(N_\vartheta \cos(\xi_i)\cos(\varphi_j))$ and discretize ϑ uniformly, i.e., $\vartheta_k = k\frac{2\pi}{\bar{N}_\vartheta}$, $k = 0, \cdots, \bar{N}_\vartheta - 1$.