An Iterative Algorithm for the Optimal Control of Continuous–Time Switched Linear Systems

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

Alessandro Giua, Carla Seatzu
Dip. di Ingegneria Elettrica ed Elettronica, Università di Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{giua,seatzu}@diee.unica.it

Abstract

For continuous-time switched linear systems, this paper proposes an approach for solving infinite-horizon optimal control problems where the decision variables are the switching instants and the sequence of operating modes. 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 effectiveness of the approach is shown through simple simulation examples.

1 Introduction

Switched systems are a particular class of hybrid systems that switch between many operating modes, where each mode is governed by its own characteristic dynamical law [1]. Typically, mode transitions are triggered by variables crossing specific thresholds (state events), by the elapse of certain time periods (time events), or by external inputs (input events). The problem of determining optimal control laws for this class of hybrid systems has been widely investigated in the last years and many results can be found in the control and computer science literature. For continuous-time hybrid systems, most of the literature is focused on the study of necessary conditions for a trajectory to be optimal [16], and on the computation of optimal/suboptimal solutions by means of dynamic programming or the maximum principle [5,6,10,13,14,18]. 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. In [9] the authors use a hierarchical decomposition approach to break down the overall problem into smaller ones. In so doing, discretization is not involved and the main computational complexity arises from a higher-level nonlinear programming problem.

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, in [4] we introduced a rather general hybrid modeling framework and showed how mixed-integer quadratic programming (MIQP) can be efficiently used to determine optimal control sequences. We also showed that when optimal control is implemented in a receding horizon fashion by repeatedly solving MIQPs on-line, this leads to an asymptotically stabilizing control law. For those cases where on-line optimization is not viable, in [2] we proposed multiparametric programming as an effective means for synthesizing piecewise affine optimal controllers, that solve in state-feedback form the finite-time hybrid optimal control problem with performance criteria based on linear (1 or infinity) norms, or, as shown more recently, on quadratic norms. Such a control design flow for hybrid systems was applied to several industrial case studies, in particular to automotive problems where the simplicity of the control law is essential for its applicability.

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, in [3] we proposed a technique that rules out switching sequences that are either not optimal or simply not compatible with the evolution of the dynamical system. An algorithm to optimize switching sequences that has an arbitrary degree of suboptimality was presented in [11].

In the continuous-time case, and in particular for switched linear systems composed by stable autonomous dynamics, by assuming that the switching sequence is preassigned (and, therefore, that the only decision variables to be optimized are the switching instants), in [7,8] we proved an important result: the control law is a state-feedback and there exists a numerically viable procedure to compute the switching regions $\mathcal{C}_{i,N}$, i.e., the points of the state space where the i-th switch of a sequence of length N should occur.

In this paper we solve the optimal control problem for continuous-time switched linear systems in which both the switching instants and the switching sequence are decision variables. 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. A few simple heuristics can be added to the algorithm to improve its performance. Although the final optimal switching policy is computed for a given initial state, as a by-product of the algorithm it has a state-feedback nature, which is only valid however for "small" perturbations of the initial state such that the optimal switching se-

Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02)
0-7695-1683-1/02 $17.00 © 2002 **IEEE**

quence does not change. 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 [12]. A two-stage procedure was also proposed in [18].

Although we formally prove that the algorithm always converges to a local minimum, the global minimum is not always reached. We run the algorithm on about 100 random tests and observed that it converges extremely quickly to the global optimum on about 95% of the problems, and in the remaining 5% the difference between the solution and the global solution (computed via enumeration) is below 1%.

2 Problem Formulation

In this paper we consider the following class of hybrid systems

$$\dot{x}(t) = A_{i(t)}x(t),\ i(t) \in \mathcal{S} \tag{1}$$

that we denote as *switched linear systems*, where $x(t) \in \mathbb{R}^n$, $i(t) \in \mathcal{S}$ is the control variable, and $\mathcal{S} \triangleq \{1, 2, \dots, s\}$ is a finite set of integers, each one associated with a linear dynamics. We assume that all matrices A_i ($i \in \mathcal{S}$) are strictly Hurwitz (i.e., all dynamics are asymptotically stable) and that a positive semi-definite weight matrix Q_i is associated to each dynamics. For such a class of hybrid systems we want to solve the following optimal control problem

$$\begin{array}{ll} V_N^* \triangleq \min\limits_{I,\mathcal{T}} & \left\{ F(I,\mathcal{T}) \triangleq \displaystyle\int_0^\infty x'(t)Q_{i(t)}x(t)dt \right\} \\ \text{s.t.} & \dot{x}(t) = A_{i(t)}x(t) \\ & x(0) = x_0 \\ & i(t) = i_k \text{ for } \tau_{k-1} \le t < \tau_k \\ & i_k \in \mathcal{S},\ k = 1, \dots, N+1 \\ & \tau_0 = 0,\ \tau_{N+1} = +\infty \\ & \tau_k \in \mathbb{R}_{\ge 0}\ \forall k = 1, \dots, N \end{array} \tag{2}$$

where N is the maximum allowed number of switches (fixed a priori), $\mathcal{T} \triangleq \{\tau_1, \dots, \tau_N\}$ is a finite sequence of switching times, $I \triangleq \{i_1, \dots, i_{N+1}\}$ is a finite sequence of switching indices, and x_0 is the initial state of the system. We denote by $i^*(t)$, $t \in [0, +\infty)$, $i^*(t) = i_k^*$ for $\tau_{k-1}^* \le t < \tau_k^*$ the switching trajectory solving (2), and I^*, $\mathcal{T}^*$ the corresponding optimal sequences.

This framework may be easily generalized. One may assume that whenever at time τ_k a switch from i_k to i_{k+1} occurs, the state should jump from $x(\tau_k^-)$ to $x(\tau_k^+) = M_k x(\tau_k^-)$ as in [8]. One may also assume that a cost is associated to each switch as in [7]. However, to avoid heavy notation in this paper we only restrict to the basic framework.

The optimal control problem (2) may also be rewritten as:

$$\begin{array}{ll} \min\limits_{I,\mathcal{T}} & \displaystyle\sum_{k=1}^{N+1} x'_{k-1} \left[Z_{i_k} - \bar{A}'_{i_k}(k) Z_{i_k} \bar{A}_{i_k}(k) \right] x_{k-1} \\ \text{s.t.} & x_k = \bar{A}_{i_k}(k) x_{k-1},\ k = 1, \dots, N+1 \\ & x_0 = x(0) \end{array} \tag{3}$$

where

$$\bar{A}_i(k) \triangleq e^{(\bar{\tau}_k - \bar{\tau}_{k-1})A_i}, \tag{4}$$

and Z_i is the unique[1] solution of the Lyapunov equation

$$A_i' Z_i + Z_i A_i = -Q_i. \tag{5}$$

Consider a decomposition of (3) into the following "master" and "slave" subproblems:

Problem 1 (Master) *For a fixed sequence of switched times $\bar{\tau}_1, \dots, \bar{\tau}_N$, solve the optimal control problem (3) with respect to $i_1, \dots, i_{N+1}$. Denote by*

$$\{i_1, \dots, i_{N+1}\} = f_M(\bar{\tau}_1, \dots, \bar{\tau}_N) \tag{6}$$

and $V_M(\bar{\tau}_1, \dots, \bar{\tau}_N)$ the optimizing index sequence and optimal value, respectively.

Problem 2 (Slave) *For a fixed sequence of switching indices $\bar{\imath}_1, \dots, \bar{\imath}_{N+1}$, solve the optimal control problem (3) with respect to $\tau_1, \dots, \tau_N$. Denote by*

$$\{\tau_1, \dots, \tau_N\} = f_S(\bar{\imath}_1, \dots, \bar{\imath}_{N+1}) \tag{7}$$

and $V_S(\bar{\imath}_1, \dots, \bar{\imath}_{N+1})$ the optimizing timing sequence and optimal value, respectively.

3 Master Algorithm

For a fixed sequence of switched times $\bar{\tau}_1, \dots, \bar{\tau}_N$, the master algorithm solves the optimal control problem (3) with respect to $i_1, \dots, i_{N+1}$. It is a purely combinatorial problem that can be rephrased as:

$$\begin{array}{ll} \min\limits_{i_k \in \mathcal{S}} & \displaystyle\sum_{k=1}^{N+1} x'_{k-1} \bar{Q}_{i_k}(k) x_{k-1} \\ \text{s.t.} & x_k = \bar{A}_{i_k}(k) x_{k-1},\ k = 1, \dots, N+1 \\ & x_0 = x(0), \end{array} \tag{8}$$

where

$$\bar{Q}_{i_k}(k) \triangleq Z_{i_k} - \bar{A}'_{i_k}(k) Z_{i_k} \bar{A}_{i_k}(k). \tag{9}$$

Problem (8) can be efficiently solved via Mixed-Integer Quadratic Programming (MIQP) (see e.g. [15] or the free Matlab solver available at `http://control.ethz.ch/~hybrid/miqp`).

[1] Because each A_i is asymptotically stable.

Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02)
0-7695-1683-1/02 $17.00 © 2002 IEEE

To this end, we need to introduce binary variables $\gamma_i^k \in \{0,1\}$ and continuous variables $z_i^k \in \mathbb{R}^n$, $i \in \mathcal{S}$, $k = 1, \ldots, N+1$, where

$$
\begin{aligned}
[\gamma_i^k = 1] \quad &\leftrightarrow \quad [i(k) = i], \quad \begin{matrix}\forall k = 1, \ldots, N+1, \\ \forall i \in \mathcal{S}\end{matrix} && \text{(10a)} \\
z_i^{k+1} \quad &= \quad \bar{A}_i(k) x_{k-1} \gamma_i^k, \quad \begin{matrix}\forall k = 1, \ldots, N, \\ \forall i \in \mathcal{S}\end{matrix} && \text{(10b)} \\
z_i^1 \quad &= \quad x_0 \gamma_i^1, \ \forall i \in \mathcal{S} && \text{(10c)} \\
x_k \quad &= \quad \sum_{i=1}^{s} z_i^{k+1}, \ \ \forall k = 0, \ldots, N && \text{(10d)} \\
\bigoplus_{i=1}^{s} \gamma_i^k \quad &= \quad 1, \ \forall k = 1, \ldots, N+1 && \text{(10e)}
\end{aligned}
$$

where the last exclusive-or constraint follows by the fact that only one dynamics can be active in each interval k. Constraints (10b) can be transformed into the following set of mixed-integer linear inequalities by using the so-called "big-M" technique (see e.g. [4, 17] for details):

$$
\begin{aligned}
z_i^k &\le M \gamma_i^k, \ \ \forall k = 1, \ldots, N+1 && \text{(11a)} \\
-z_i^k &\le M \gamma_i^k, \ \ \forall k = 1, \ldots, N+1 && \text{(11b)} \\
z_i^{k+1} &\le \bar{A}_i(k) x_{k-1} + M(1 - \gamma_i^k), \quad \forall k = 1, \ldots, N && \text{(11c)} \\
-z_i^{k+1} &\le -\bar{A}_i(k) x_{k-1} + M(1 - \gamma_i^k), \quad \forall k = 1, \ldots, N && \text{(11d)} \\
z_i^1 &\le x_0 + M(1 - \gamma_i^1) && \text{(11e)} \\
-z_i^1 &\le -x_0 + M(1 - \gamma_i^1) && \text{(11f)}
\end{aligned}
$$

for all $i \in \mathcal{S}$, 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 $A_i(k)x_{k-1} = x_k$, for all $k = 2, \ldots, N+1, i \in \mathcal{S}$. Usually M can be estimated on the basis of physical considerations on the hybrid system. Eq. (10e) can be instead expressed as

$$
\sum_{i=1}^{s} \gamma_i^k = 1, \ \ \forall k = 1, \ldots, N+1. \tag{12}
$$

Summing up, the master problem (8) is equivalent to the MIQP

$$
\min_{\substack{x_k, \gamma_i^k, z_i^k \\ k = 1, \ldots, N+1 \\ i = 1, \ldots, s}} \quad \sum_{k=1}^{N+1} \sum_{i=1}^{s} (z_i^k)' \bar{Q}_i(k) z_i^k \tag{13}
$$

$$
\text{s.t.} \quad \text{(10d), (11), (12).}
$$

4 Slave Algorithm

For a fixed sequence of switching indices $\bar{\imath}_1, \ldots, \bar{\imath}_{N+1}$, the slave algorithm solves the optimal control problem (3) with respect to $\tau_1, \ldots, \tau_N$.

A solution to this problem where the switching sequence is pre-assigned was already presented in [8] where it was shown that the optimal control law turns out to be a "homogeneous feedback", in the sense that for all $k \le N$: (a) it is possible to identify a region $\mathcal{C}_{k,N}$ of the state space such that the k–th switch should occur if and only if we are within this region; (b) this region is homogeneous, i.e., if $x \in \mathcal{C}_{k,N}$, then $\lambda x \in \mathcal{C}_{k,N}$, for all real numbers λ.

We have also provided an algorithmic way to construct the regions. In fact, we observed that it is sufficient to determine which points on the unitary semi–sphere belong to a region to completely determine the region itself (because it is a homogeneous space). In [8] we have also shown that these *switching regions* have to be computed starting from the last one. More precisely, let us first define the residual cost from the k-th to the N-th switch, given a state x, as:

$$
F_k(x, \delta_k, \delta_{k+1}, \ldots, \delta_N) = \textstyle\sum_{j=k}^{N+1} x'_{j-1} \bar{Q}_{i_j}(j) x_{j-1} \tag{14}
$$

where $\delta_j = \tau_j - \tau_{j-1}$ is the j–th switching interval and $x_{k-1} = x$. We also define the corresponding k-th optimal switching interval as:

$$
\delta_k^*(x) = \arg \min_{\delta_k \in \mathbb{R}_0^+} F_k(x, \delta_k, \delta_{k+1}^*(x_k), \ldots, \delta_N^*(x_{N-1}))
$$

where $x_j = e^{A_j \delta_j^*(x_{j-1})} x_{j-1}$. Finally we can write that

$$
\mathcal{C}_{k,N} = \{x \mid \delta_k^*(x) = 0\} \quad k = 1, \ldots, N. \tag{15}
$$

Thus, we choose a suitable discretization step and for each point x on the unitary semi–sphere, determine if it belongs to $\mathcal{C}_{N,N}$, $\mathcal{C}_{N-1,N}$, etc., also computing step by step the corresponding values of the remaining cost.

The output of this procedure is the set of switching regions. To determine the optimal switching instants, the evolution of the system is simulated starting from the initial state x_0, by switching as soon as the next switching region is reached.

5 Master-Slave Algorithm

The proposed master-slave algorithm is structured as follows:

Algorithm 1

1. Initialize $\mathcal{T}(0) \leftarrow \{\tau_1, \ldots, \tau_N\}$ (e.g., τ_k are randomly or uniformly distributed), $k = 1$, $I(0) = \{-1, \ldots, -1\}$; Let $\epsilon > 0$ a given tolerance;
2. Solve the master problem $I(k) \leftarrow f_M(\mathcal{T}(k-1))$;
3. If $|F(\mathcal{T}(k-1), I(k)) - F(\mathcal{T}(k-1), I(k-1))| \le \epsilon$ 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 $\{\tau_1, \ldots, \tau_N\} \leftarrow \mathcal{T}(k)$, $\{i_1, \ldots, i_{N+1}\} \leftarrow I(k)$;
8. End

Proposition 1 *Algorithm 1 stops after a finite number of steps* $N_{\rm stop}$.

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

$$V(k-1) = F(\mathcal{T}(k-1), I(k-1)) \geq F(\mathcal{T}(k-1), I(k)) \geq F(\mathcal{T}(k), I(k)) = V(k).$$

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 3. is satisfied after a finite number of iteration k_ϵ for any given positive tolerance ϵ. □

Assumption 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* $F(I_1, \mathcal{T}) = F(I_2, \mathcal{T})$, time-degenerate *if there exist a sequence* I *and* $\mathcal{T}_1 \neq \mathcal{T}_2$ *such that* $F(I, \mathcal{T}_1) = F(I, \mathcal{T}_2)$.

Proposition 2 *Let* $\epsilon = 0$ *and assume problem (2) is not switch-degenerate. Let step 3. be modified as follows*

3'. If $I(k) = I(k-1)$ go to 7.;

Then Algorithm 1 stops after a finite number of steps $N_{\rm stop}$.

Proof: By contradiction, assume the stopping criterion 3'. is never met. Since the number of possible sequences I is finite, there must exist indices j, k such that $I(j) = I(k)$. Since step 3'. never succeeds, necessarily $|j-k| \geq 2$, and moreover $I(j+1) \neq I(j)$. As (2) is not switch-degenerate,

$$\begin{array}{rcl} V(j) &=& F(\mathcal{T}(j), I(j)) > F(\mathcal{T}(j), I(j+1)) \\ &\geq& V(j+1) \geq V(k) = F(\mathcal{T}(k), I(k)) \\ &=& F(\mathcal{T}(k), I(j)), \end{array}$$

which contradicts the optimality of $\mathcal{T}(j)$ for the slave problem at step j. □

Note that Proposition 2 proves that Algorithm 1 cannot cycle over the same switching sequences, and $I(k) \neq I(j)$ for all $j \neq k$, $j, k \in \{1, \ldots, N_{\rm stop} - 1\}$.

We remark that although Algorithm 1 converges to a solution $I, \mathcal{T}$ after a finite number $N_{\rm stop}$ 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. As we will exemplify later, our computational experience with about 100 random tests shows convergence to the global optimum on about 95% of the problems, and in the remaining 5% the difference between the solution and the global solution (computed via enumeration) is below 1%.

Algorithm 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 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.

5.1 Degeneracies

We 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 the switching index i_k is undetermined (multiple solutions for I)

6 Numerical Examples

Example 1 Let us consider a second order linear system whose dynamics may be chosen within a finite set $\{A_1, A_2, A_3\}$. In particular, we assume:

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

$$A_3 = \begin{bmatrix} -2.414 & -1.414 \\ 1 & 0 \end{bmatrix}$$

Each dynamics has an associated weighting matrix: $Q_1 = \mathrm{diag}\{1,\ 1\}$, $Q_2 = \mathrm{diag}\{8,\ 2\}$, $Q_3 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$.

We also assume that only three switchings are possible, thus $N = 3$ and the control variable $i(t)$ may only take values from the finite set of integers $\mathcal{S} = \{1, 2, 3\}$. The initial state vector has been taken equal to $x_0 = [1\ 1]^T$.

We apply the master–slave algorithm to determine the optimal index sequence. The initial timing sequence has been randomly generated and taken equal to $\mathcal{T}_0 = \{0.290, 0.498, 0.672\}$. The master–slave algorithm finds out that the optimizing index 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 we may also observe that the procedure converges after only 5 steps. This also implies that the most burdensome part of the algorithm, i.e., the slave problem, has only been solved twice.

The correctness of the solution has been validated through an exhaustive inspection of all admissible index sequences. More precisely, for each admissible index sequence we have computed the optimizing timing sequence and the corresponding cost value using the slave algorithm. In such a way we have verified $V_3^* = 1.44026$ is indeed the global optimum. Obviously, being only two

Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02)
0-7695-1683-1/02 $17.00 © 2002 **IEEE**

step		τ_1	τ_2	τ_3	i_1	i_2	i_3	i_4	$F(I,T)$
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 1: Detailed results of the numerical example 1.

the switches required to optimize the cost value, the minimum cost may also be obtained by using other index sequences. As an example, if we consider $I = \{3, 1, 2, 3\}$ and $\mathcal{T} = \{0, 0.180, 0.240\}$, this solution is optimal as well.

In Figure 1 we have reported the switching regions $\mathcal{C}_{j,3}$, $j = 1, 2, 3$, when the index sequence is the optimal one. The darker region represents the set of states where the system still evolves with the same dynamics, while the lighter region represents the set of states where the system switches to the next dynamics. Clearly, in $\mathcal{C}_{3,3}$ we have no light portion because it corresponds to a non–effective switch, being $i(3) = i(4) = 3$.

Finally, in the bottom right of Figure 1 we have shown the system evolution for the chosen initial state $x_0 = [1\ 1]^T$.

On the basis of several random tests we performed, we observed that the convergence of the algorithm to a global minimum is heavily influenced by two factors. Firstly, 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 minor number of switches — is usually computed. Secondly, 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 In this second example we present an heuristics that in some cases improves the performance of the algorithm.

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

$$A_1 = \begin{bmatrix} -1.851 & -1.002 \\ 1 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -0.7445 & -1.289 \\ 1 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} -2.754 & -2.839 \\ 1 & 0 \end{bmatrix}.$$

To each dynamic we associate a weighting matrix: $Q_1 = \text{diag}\{1,\ 1\}$, $Q_2 = \text{diag}\{0.466,\ 0.4186\}$, $Q_3 = \text{diag}\{2,\ 8\}$.

As in the previous example we assume that $N = 3$, i.e., at most three switchings are possible, and $x_0 = [1\ 1]^T$.

We take as initial timing sequence $\mathcal{T}_0 = \{0.011, 0.145, 0.234\}$ and apply the master–slave algorithm to determine the optimal index sequence. The provided solution is $I = \{1, 1, 1, 2\}$ and the corresponding

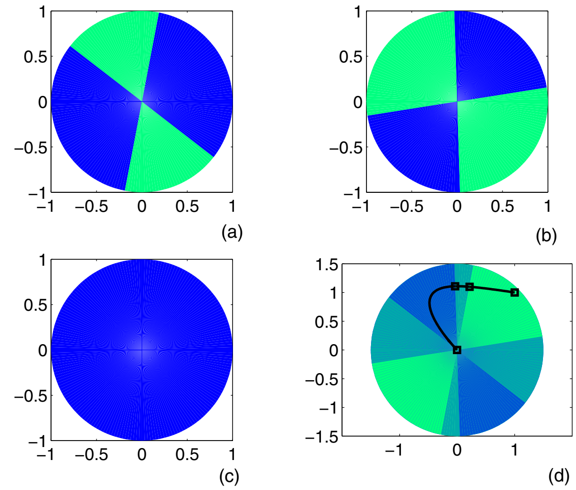

Figure 1: The switching regions for the system in example 1 when the initial state is $x_0 = [1\ 1]^T$ and the index sequence is the optimal one: (a) $\mathcal{C}_{1,3}$, (b) $\mathcal{C}_{2,3}$, (c) $\mathcal{C}_{3,3}$. (c) The system evolution for $x_0 = [1\ 1]^T$.

step		τ_1	τ_2	τ_3	i_1	i_2	i_3	i_4	$F(I,T)$
1	M	0.011	0.145	0.234	2	2	2	2	1.66123
1	S	0.000	0.000	0.000	2	2	2	2	1.66123
2	M	0.000	0.000	0.000	1	1	1	2	1.66123
2	S	0.000	0.000	0.000	1	1	1	2	1.66123
3	M	0.000	0.000	0.000	1	1	1	2	1.66123

Table 2: Detailed results of the numerical example 2 when the master–slave algorithm is applied in its original form.

performance index is $V_3 = 1.66123$. Detailed results are reported in Table 2. Nevertheless, this solution is not optimal and this may be easily verified through an exhaustive inspection of all admissible index sequences.

A careful examination of the solution, suggests us that such a case may be considered as degenerate, being $\{1, 1, 1, 2\}$ an index sequence that corresponds to only one switch. Thus, when it is used by the slave algorithm at the fourth step, 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 numerical examples we have examined — may be summarized as follows. The index sequence computed via the master algorithm that corresponds to a number of switches that is less that N, is modified before being used by the slave algorithm. In particular, we suggest to arbitrarily change the index sequence so that the original sequence is still contained in the new one but two consecutive indices should never be the same.

In the numerical case at hand the results of the master–slave algorithm, when the above caution is adopted, are those reported in Table 3. In particular, we may observe that at the second step of the whole procedure, the slave algorithm does not examine the index sequence firstly computed by the master algorithm, but computes the optimal

Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02)
0-7695-1683-1/02 $17.00 © 2002 **IEEE**

step		τ_1	τ_2	τ_3	i_1	i_2	i_3	i_4	$F(I,T)$
1	M	0.011	0.145	0.234	2	2	2	2	1.66123
1	S	0.000	0.000	0.000	1	3	1	2	1.66123
2	M	0.000	0.000	0.000	1	1	1	2	1.66123
2	S	1.780	3.130	4.860	1	3	1	2	1.51340
3	M	1.780	3.130	4.860	1	3	1	2	1.51340

Table 3: Detailed results of the numerical example 2 when the master–slave algorithm is applied with the proposed heuristic.

timing sequence corresponding to a new index sequence $I = \{1, 3, 1, 2\}$, that has been randomly generated by arbitrarily modifying all indices — apart from the last one — so as to avoid switch degeneracy. At this step, the value of the performance index does not decrease, thus the artifice is useless. The same reasoning is repeated at the fourth step and in this case we find out a better value of the cost and at the following fifth step the procedure stops. Moreover, the results of the exhaustive search show that the computed solution is optimal thus revealing the effectiveness of the modified procedure.

Although this heuristic is not always effective, it often improves the performance of the algorithm while it may never make it worse. Its only drawback is that, to avoid cycling, it is necessary to add a stopping condition that detects loops.

7 Conclusions

In this paper we have proposed an approach for solving infinite-horizon optimal control problems for continuous-time switched linear systems, where both the switching instants and the sequence of operating modes must be determined.1

There are several ways in which the proposed approach can be extended. In many practical applications a direct switch from dynamics i to dynamics j may not be admissible. Generalizing, it may be useful to restrict the set of switching sequences based on reachability sets of finite automata. This additional constraint may be easily taken into account in the master algorithm. We believe that it should be possible to relax the assumptions (required by the slave algorithm) that all dynamics are stable and linear. In particular we plan to explore the case in which all subsystems are affine systems.

References

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

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

[3] A. Bemporad, L. Giovanardi, and F. Torrisi. Performance driven reachability analysis for optimal scheduling and control of hybrid systems. Technical Report AUT00-15, Automatic Control Laboratory, ETH Zurich, Switzerland, 2000.

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

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

[6] M. Branicky and S. Mitter. Algorithms for optimal hybrid control. In *Proc. 34th IEEE Conf. on Decision and Control*, pages 2661–2666, New Orleans, LA, USA, 1995.

[7] A. Giua, C. Seatzu, and C. V. D. 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.

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

[9] K. Gokbayrak and C. 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, USA, December 1999.

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

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

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

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

[14] 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, 1999.

[15] N. V. Sahinidis. BARON — Branch And Reduce Optimization Navigator. Technical report, University of Illinois at Urbana-Champaign, Dept. of Chemical Engineering, Urbana, IL, USA, 2000.

[16] H. Sussmann. A maximum principle for hybrid optimal control problems. In *Proc. 38th IEEE Conf. on Decision and Control*, pages 425–430, Phoenix, AZ, USA, 1999.

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

[18] X. Xu and P. Antsaklis. An approach for solving general switched linear quadratic optimal control problems. In *Proc. 40th IEEE Conf. on Decision and Control*, pages 2478–2483, Orlando, FL, USA, 2001.