Planning methods for the optimal control and performance certification of general nonlinear switched systems

Lucian Buşoniu, Marcos Cesar Bragagnolo, Jamal Daafouz, and Irinel-Constantin Morărescu

***Abstract*— We consider two problems for discrete-time switched systems with autonomous, general nonlinear modes. The first is optimal control of the switches so as to minimize the discounted infinite-horizon sum of the costs. The second problem occurs when switches are a disturbance, and the worst-case cost under any sequence of switches is sought. We use an optimistic planning (OP) algorithm that can solve general optimal control with discrete inputs such as switches. We extend the analysis of OP to provide sequences of switches with certification (upper and lower) bounds on the optimal and worst-case costs, and to characterize the convergence rate of the gap between these bounds. Since a minimum dwell time between switches must often be ensured, we introduce a new optimistic planning variant that can handle this case, and analyze its convergence rate. Simulations for linear and nonlinear modes illustrate that the approach works in practice.**

I. Introduction

Switched systems consist of a set of linear or nonlinear dynamics called modes, together with a law for switching between these modes [13]. They are employed to model real-world systems that are subject to known or unknown abrupt parameter changes, e.g. embedded systems in automotive industry, aerospace, and energy management. This important class of hybrid systems is therefore heavily studied, with a main focus on stability and stabilization, see surveys [14], [19]. Performance optimization for switched systems has also been investigated, see e.g. the survey [22]. Hybrid versions of the Pontryagin Maximum Principle or dynamic programming have been proposed [17], [18], with the drawback of lacking efficient numerical algorithms. Suboptimal solutions with guaranteed performance include [21], [9]. The former efficiently represents the approximate value function using relaxations. The latter proves that the so-called min-switching strategies are consistent, i.e. that they improve performance with respect to non-switching strategies. Certification bounds [6] (lower and upper bounds on performance) are provided for linear switched systems with a dwell time assumption in [10]. In [7], the problem is treated by introducing modal occupation measures, which allow relaxation to a primal linear programming (LP) formulation. Overall, however, optimal control remains unsolved for general switched systems.

L. Buşoniu is with the Automation Department, Technical University of Cluj-Napoca, Romania (lucian@busoniu.net). M. Bragagnolo, J. Daafouz, and I.C. Morărescu are with Université de Lorraine, CRAN, UMR 7039 and CNRS, CRAN, UMR 7039, 2 Avenue de la Forêt de Haye, Vandœuvre-lès-Nancy, France. This work was supported by a Programme Hubert Curien-Brancusi cooperation grant, CNCS-UEFISCDI contract no. 781/2014 and Campus France grant no. 32610SE; by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PNII-RU-TE-2012-3-0040; and by the ANR project "Computation Aware Control Systems", ANR-13-BS03-004-02. We are grateful to Koppány Máthé for providing ideas on analyzing the algorithm with dwell-time constraints.

In this paper, we propose an approach inspired from the field of planning in artificial intelligence, to search either for the best performance when the switches are controllable, or for the worst-case performance when switching acts as a disturbance. We consider a set of autonomous, general *nonlinear* modes, and a performance index consisting of the discounted infinite-horizon sum of general, nonquadratic stage costs. Optimistic planning [8], [15] is used to search the space of possible sequences of switches. Since a minimum dwell time between switches must often be guaranteed, we introduce a new optimistic planner that handles this constraint, and analyze its convergence rate. In both the worst-case or optimal-control settings, our approach designs a sequence that guarantees certification on the performance, while optionally enforcing a minimum dwell-time.

Compared to the optimal control methods reviewed above, the advantages of our approach include: a procedure to design a worst-case sequence, a characterization of the certification bounds, and a method for the case of constrained switching. While a high computational complexity is unavoidable due to the generality of the switched system considered, our analysis is focused precisely on characterizing the relation between computation and quality of the bounds. Note also that we focus on optimality, without analyzing stability – which is a separate, difficult problem for discounted cost [12], [16]. In certain cases stability conditions exist, e.g. for linear modes and a minimum dwell time [5], and our approach can handle dwell-time constraints. Moreover, our method and guarantees may be useful even if stability cannot be formally decided.

Next, Section II formalizes the problem and Section III gives the necessary background. Section IV describes the proposed approach, and Section V evaluates it in two linear examples and a nonlinear one. Section VI concludes.

II. Problem statement

Consider a discrete-time nonlinear switched system with states $x \in X$ that can be at each step k in one of M modes $u \in U = \{u^1, \ldots, u^M\}$, where each mode is autonomous:

$$x_{k+1} = f_{u_k}(x_k) \tag{1}$$

The dwell time is defined as the number of steps during which the mode remains unchanged after a switch. A function $g(x_k, u_k)$ assigns a numerical stage cost to each state-mode pair. Under a fixed initial state x_0, define an infinitely-long sequence of modes $\boldsymbol{u}_\infty = (u_0, u_1, \ldots)$ and the infinite-horizon discounted cost of this sequence:

$$J(\boldsymbol{u}_\infty) = \sum_{k=0}^{\infty} \gamma^k g(x_k, u_k) \tag{2}$$

where $\gamma \in (0,1)$ is the discount factor and $x_{k+1} = f_{u_k}(x_k)$. Discounting is necessary for our planning algorithms, and many other works use it, e.g. [1], [4], [11]. No specific form is required for dynamics f, and a closed form need not even exist (e.g. f may be represented as a computer program).

In this context, we define two different problems:

PO. **Optimal control:** Find the optimal value $\underline{J} = \inf_{\boldsymbol{u}_\infty} J(\boldsymbol{u}_\infty)$ and a corresponding sequence of switches that achieves it.

PW. **Worst-case switches**: Find the largest possible cost: $\overline{J} = \sup_{\boldsymbol{u}_\infty} J(\boldsymbol{u}_\infty)$, and a corresponding sequence of switches that achieves it.

PO is useful when the switches can be controlled, while PW is interesting when they are a disturbance and we are interested in the performance under the worst-case disturbance. We rely on a central assumption of cost boundedness.

Assumption 1: The stage costs are bounded in $[0, G]$.

Due to the discounting, this guarantees that the infinite-horizon cost J in (2) is bounded to $[0, \frac{G}{1-\gamma}]$.

Example 1: A classical switched system is obtained when the modes are linear and the cost is quadratic. A typical way of ensuring Assumption 1 is to saturate the cost to G:

$$f_{u_k} = A_{u_k} x_k$$
$$g(x_k, u_k) = \min\{x_k^\top Q x_k, G\}$$

This changes the optimal solution, but is often sufficient in practice. For this system, Theorem 1 in [5] provides a minimum dwell time which, if obeyed, guarantees stability for any switching sequence. We provide an algorithm that enforces a minimum dwell-time in Section IV-B. □

Of course, PO usually only makes sense if the system is stabilizable, and PW additionally requires that it is stable under any switching sequence. We focus here on optimality, so we do not analyze stability. While stability conditions sometimes exist, as in Example 1, stability under discounted cost is generally a difficult problem [12], [16]. Nevertheless, even when conditions to guarantee stability analytically cannot be found, PO and PW may still make sense, and our algorithms can be applied. If stability is empirically verified by the trajectory generated, then our near-optimality bounds remain meaningful.

III. Background: Optimistic planning for deterministic systems

This section introduces optimistic planning for deterministic systems (OP) [8], [15], which forms the basis of our approach: it supplies independence of the mode dynamics, as well as a way to design sequences with known lower and upper bounds on the performance. Both PO and PW will be encompassed as variants of an optimal control problem that involves maximizing a *reward* function $\rho : X \times U \to [0,1]$, where U is the discrete set of M actions. Given an initial state x_0, the return of a sequence is:

$$v(\boldsymbol{u}_\infty) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k) \tag{3}$$

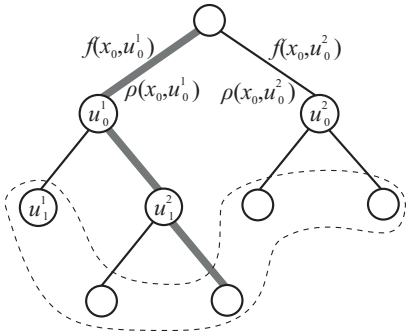

Fig. 1. Illustration of an OP tree $\mathcal{T}$. Nodes are labeled by actions, arcs represent transitions and are labeled by the resulting states and rewards. Subscripts are depths, superscripts index the M possible actions/transitions from a node (here, $M = 2$). The leaves are enclosed in a dashed line, while the gray path denotes a sequence.

Algorithm 1 Optimistic planning.

1: initialize tree $\mathcal{T} \leftarrow \{\boldsymbol{u}_0\}$
2: **for** $t = 1, \ldots, n$ **do**
3: find optimistic leaf: $\boldsymbol{u}^\dagger \leftarrow \arg\max_{\boldsymbol{u} \in \mathcal{L}(\mathcal{T})} b(\boldsymbol{u})$
4: add to $\mathcal{T}$ the children of $\boldsymbol{u}^\dagger$, labeled by $u^1, \ldots, u^M$
5: **end for**
6: **return** sequence $\boldsymbol{u}_d^* = \arg\max_{\boldsymbol{u} \in \mathcal{L}(\mathcal{T})} l(\boldsymbol{u})$, lower bound $l^* = l(\boldsymbol{u}_d^*)$, upper bound $b^* = \max_{\boldsymbol{u} \in \mathcal{L}(\mathcal{T})} b(\boldsymbol{u})$

and the optimal return is $v^* = \sup_{\boldsymbol{u}_\infty} v(\boldsymbol{u}_\infty)$. Under mild technical conditions, this optimum exists, together with a sequence that achieves it [2]. Define a finite-length sequence of d actions as $\boldsymbol{u}_d = (u_0, \ldots, u_{d-1})$.

OP explores a tree representation of the possible action sequences from the current system state, as illustrated in Figure 1. Each node at some depth d is reached via a unique path through the tree, corresponding to a unique action sequence $\boldsymbol{u}_d$ of length d. We denote the nodes by their corresponding action sequences, the current tree by $\mathcal{T}$, and its leaves by $\mathcal{L}(\mathcal{T})$. For any node/sequence $\boldsymbol{u}_d$, because all the rewards at depths larger than d are in $[0,1]$, the following upper bound holds for the returns of all infinite action sequences that share the initial subsequence up to $\boldsymbol{u}_d$:

$$b(\boldsymbol{u}_d) = \sum_{i=0}^{d-1} \gamma^i \rho(x_i, u_i) + \frac{\gamma^d}{1-\gamma} =: l(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}$$

where $l(\boldsymbol{u}_d)$ is a lower bound. Here, $x_k, k \geq 1$ is the state sequence obtained by applying $\boldsymbol{u}_d$.

OP starts with a root node representing the empty sequence, and iteratively expands n nodes, where expanding a node adds M new children nodes, one for each possible discrete action. This corresponds to appending each action to the sequence of the expanded node. The expansion rule is optimistic since it expands at each iteration the most promising sequence: the one with the largest upper bound. After n node expansions, a "safe" sequence that maximizes l among the leaves is returned, together with bounds on the performance, see Algorithm 1. While OP is a type of nonlinear model-predictive control, its AI origins lead to some atypical near-optimality guarantees, described next.

To analyze the complexity of finding the optimal sequence from x_0, define the near-optimal subtree:

$$\mathcal{T}^* = \{\boldsymbol{u}_d \,|\, d \geq 0, v^* - v(\boldsymbol{u}_d) \leq \frac{\gamma^d}{1-\gamma}\} \tag{4}$$

where the value of a finitely long sequence is $v(\boldsymbol{u}_d) := \sup_{\boldsymbol{u}_\infty} v((\boldsymbol{u}_d, \boldsymbol{u}_\infty))$, with $(\cdot,\cdot)$ denoting sequence concatenation. A core property of OP is that it only expands nodes in $\mathcal{T}^*$. This subtree can be significantly smaller than the complete tree containing all sequences, and to measure its size let $\mathcal{T}_d^*$ be the set of nodes at depth d on $\mathcal{T}^*$ and $|\cdot|$ denote set cardinality. Then, define the asymptotic branching factor as $\kappa = \limsup_{d\to\infty} |\mathcal{T}_d^*|^{1/d}$. This is a complexity measure for the problem, see below for more intuition.

The upcoming theorem is a consequence of the analysis in [8], [15]. Parts (i), (ii) show that OP returns a long, near-optimal sequence with known performance bounds, and part (ii) quantifies the length and bounds via branching factor κ.

Theorem 2: When OP is called with budget n:

(i) The optimal value v^*, as well as the value $v(\boldsymbol{u}_d^*)$ of the sequence returned, are in the interval $[l^*, b^*]$. Further, the gap $\varepsilon := b^* - l^*$ satisfies $\varepsilon \leq \frac{\gamma^{d^*}}{1-\gamma}$ where d^* is the largest depth of any node expanded.

(ii) The length d of sequence $\boldsymbol{u}_d^*$ is at least d^*.

(iii) If $\kappa > 1$, OP will reach a depth of $d^* = \Omega(\frac{\log n}{\log \kappa})$, and $\varepsilon = O(n^{-\frac{\log 1/\gamma}{\log \kappa}})$. If $\kappa = 1$, $d^* = \Omega(n)$ and $\varepsilon = O(\gamma^{cn})$, where c is a problem-dependent constant.[1]

Proof: Part (ii) follows from the proof of Theorem 2 in [8], and (iii) from the proofs of Theorems 2 and 3 in [8]. Part (i) is stated here in a new form that brings out the lower and upper bounds, so we will prove it.

Clearly, $l^* \leq v(\boldsymbol{u}_d^*) \leq v^*$ by definition. Consider now a leaf sequence $\boldsymbol{u}'$ on the final tree, that is an initial subsequence of an optimal sequence. Since b^* is the largest upper bound, $b^* \geq b(\boldsymbol{u}') \geq v^*$, so combined with the first inequality we get $v^*, v(\boldsymbol{u}_d^*) \in [l^*, b^*]$. Further, by expanding nodes the largest b-value on the tree can only decrease. Hence, for any node $\boldsymbol{u}^\dagger$ previously expanded, found at some depth d, we have $b^* \leq b(\boldsymbol{u}^\dagger)$ and also $\ell^* \geq \ell(\boldsymbol{u}^\dagger)$, so $\varepsilon = b^* - l^* \leq b(\boldsymbol{u}^\dagger) - l(\boldsymbol{u}^\dagger) = \frac{\gamma^d}{1-\gamma}$. One such node is at d^*, so $\varepsilon \leq \frac{\gamma^{d^*}}{1-\gamma}$. ■

Note that d^* is the depth of the developed tree minus 1. The smaller κ, the better OP does. The best case is $\kappa = 1$, obtained e.g. when a single sequence always obtains rewards of 1, and all the other rewards on the tree are 0. In this case the algorithm must only develop this sequence, and the gap decreases exponentially. In the worst case, $\kappa = M$, obtained e.g. when all the sequences have the same value, and the algorithm must explore the complete tree in a uniform fashion, expanding nodes in order of their depth.

[1]For $g, h : [0,\infty) \to [0,\infty)$, $g(t) = O(h(t))$ (or $g(t) = \Omega(h(t))$) means $\exists t_0, c > 0$ so that $g(t) \leq ch(t)$ (or $g(t) \geq ch(t)$) $\forall t \geq t_0$.

IV. Approach and analysis

A. Applying OP to switched systems

OP can be applied to the system in Section II by interpreting the mode switches as discrete actions. To solve the optimal control problem PO and the worst-case problem PW, the reward function is taken, respectively, as:

$$\underline{\rho}(x,u) := 1 - \frac{g(x,u)}{G}, \quad \overline{\rho}(x,u) = \frac{g(x,u)}{G} \tag{5}$$

so that maximizing $\underline{\rho}$ is equivalent to minimizing costs g, and maximizing $\overline{\rho}$ to maximizing costs g. We use underline to denote quantities under PO and overline for PW, e.g. $\underline{\kappa}$ and $\overline{\kappa}$ are the complexity measures (branching factors) in the two problems. Then OP is simply applied with either of these two reward functions, and it will produce certification bounds and design a sequence that achieves them, as described next.

Corrolary 3: (i) When applied to PO, OP returns bounds $\underline{l}, \underline{b}$ so that the optimal value $\underline{J}$ is in the interval $[G(\frac{1}{1-\gamma} - \underline{b}), G(\frac{1}{1-\gamma} - \underline{l})]$, as well as a sequence $\underline{\boldsymbol{u}}$ that achieves these bounds. The gap (interval size) is $G\underline{\varepsilon} = O(n^{-\frac{\log 1/\gamma}{\log \underline{\kappa}}})$ when $\underline{\kappa} > 1$, or $O(\gamma^{\underline{c}n})$ when $\underline{\kappa} = 1$.

(ii) When applied to PW, OP returns bounds $\overline{l}, \overline{b}$ so that the worst-case value $\overline{J}$ is in the interval $[G\overline{l}, G\overline{b}]$, as well as a sequence $\overline{\boldsymbol{u}}$ that achieves these bounds. The gap is $G\overline{\varepsilon} = O(n^{-\frac{\log 1/\gamma}{\log \overline{\kappa}}})$ when $\overline{\kappa} > 1$, or $O(\gamma^{\overline{c}n})$ when $\overline{\kappa} = 1$.

Proof: For any infinitely long sequence $\boldsymbol{u}_\infty$, it is easily seen that the value under $\underline{\rho}$ is $\underline{v}(\boldsymbol{u}_\infty) = \frac{1}{1-\gamma} - \frac{1}{G}J(\boldsymbol{u}_\infty)$, and so $\underline{v}^* = \frac{1}{1-\gamma} - \frac{1}{G}\underline{J}$. Using this fact and Theorem 2, Part (i) is derived immediately. We similarly observe $\overline{v}(\boldsymbol{u}_\infty) = \frac{1}{G}J(\boldsymbol{u}_\infty)$ and easily derive Part (ii). ■

The results above are for a single sequence starting at x_0. In practice (and in our examples below), the algorithm is used in receding horizon, by only applying the first action u_0 of the sequence, then recomputing a new sequence from x_1 and applying its first action u_1, etc.

B. Enforcing a dwell-time constraint

It is often important to ensure that after switching, the system remains in the same mode for a certain number of steps – the dwell time. This is because for some systems fundamental properties (stability, performance, etc.) can be guaranteed only under dwell time constraints, see e.g. Example 1 and [5]. Another reason is that in practice, it may be unsuitable or impossible to switch arbitrarily fast, so the designer must guarantee by construction a minimum dwell time.

Therefore, we introduce and analyze an algorithm that enforces a dwell time of at least δ along any sequence. We start from OP and introduce a function $\Delta(\boldsymbol{u})$, which takes as input any finite-length sequence $\boldsymbol{u}$ and provides the last dwell time at the end of the sequence. Then, the dwell-time condition is checked for every node to be expanded. If the dwell time is at least δ, a switch can occur, and so children are created for all the actions (modes). Otherwise, a switch is not allowed, so only the child that keeps the mode constant is created. Figure 2 illustrates the complete tree down to depth

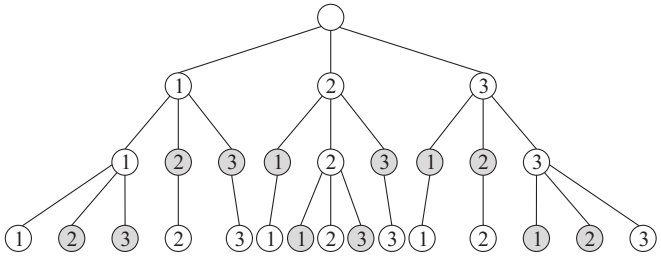

Fig. 2. Illustration of a constrained, OPδ tree for $\delta = 2$. Nodes are labeled by integer actions. The gray nodes have dwell time 1, so they are allowed only one child, the one keeping the action unchanged. The children of the gray nodes have dwell time 2, so they are eligible for full expansion. If δ were 3 instead, then these children would not satisfy the constraint either.

Algorithm 2 OP with a dwell-time constraint.

initialize tree $\mathcal{T} \leftarrow \{\boldsymbol{u}_0\}$
for $t = 1, \dots, n$ **do**
  find optimistic leaf: $\boldsymbol{u}^\dagger \leftarrow \arg\max_{\boldsymbol{u} \in \mathcal{L}(\mathcal{T})} b(\boldsymbol{u})$
  if $\Delta(\boldsymbol{u}^\dagger) \geq \delta$ **then**
    create all children of $\boldsymbol{u}^\dagger$, labeled by $u^1, \dots, u^M$
  else
    create one child, labeled by last action u on $\boldsymbol{u}^\dagger$
  end if
end for
return $\boldsymbol{u}_d^* = \arg\max_{\boldsymbol{u} \in \mathcal{L}(\mathcal{T})} l(\boldsymbol{u}), l^* = l(\boldsymbol{u}_d^*), b^* = \max_{\boldsymbol{u} \in \mathcal{L}(\mathcal{T})} b(\boldsymbol{u})$

3 (the algorithm may only create some of these nodes). By convention, it is assumed that the dwell time condition is satisfied at $d = 1$, see also the discussion on closed-loop application at the end of the section. The resulting procedure is called OP with a dwell-time constraint (OPδ) and shown in Algorithm 2.

Denote now by $\boldsymbol{U}_\delta$ the set of sequences satisfying the constraint, and the constrained optimal values:

$$v_\delta^* = \sup_{\boldsymbol{u}_\infty \in \boldsymbol{U}_\delta} v(\boldsymbol{u}_\infty)$$
$$v_\delta(\boldsymbol{u}_d) = \sup_{\boldsymbol{u}_\infty \text{ s.t. } (\boldsymbol{u}_d, \boldsymbol{u}_\infty) \in \boldsymbol{U}_\delta} v((\boldsymbol{u}_d, \boldsymbol{u}_\infty))$$

Of course, the constrained optimum is generally worse than the unconstrained one, $v_\delta^* \leq v^*$, so enforcing the constraint comes at a price. We analyze in the sequel the bounds and gap provided by OPδ, in the general case of a reward function ρ. Then, by choosing the rewards as in (5), we can solve either PO or PW under the dwell-time constraint. Note that whenever the distinction between unconstrained and constrained values is not clear, we explicitly add the subscript δ to the quantity in the constrained problem.

As for OP, define the near-optimal constrained subtree:

$$\mathcal{T}_\delta^* = \{\boldsymbol{u}_d \,|\, d \geq 0, \boldsymbol{u}_d \in \boldsymbol{U}_\delta, v_\delta^* - v_\delta(\boldsymbol{u}_d) \leq \frac{\gamma^d}{1-\gamma}\} \quad (6)$$

where $\boldsymbol{u}_d \in \boldsymbol{U}_\delta$ means that there exist some infinite constrained sequence starting with $\boldsymbol{u}_d$. Then, the following properties similar to OP also hold in the constrained case.

Lemma 4: OPδ only expands nodes in $\mathcal{T}_\delta^*$, and the optimal constrained value v_δ^*, as well as the value $v_\delta(\boldsymbol{u}_d^*)$ of the sequence returned, are in the interval $[l^*, b^*]$. Further, the gap $[l^*, b^*]$ satisfies $\varepsilon \leq \frac{\gamma^{d^*}}{1-\gamma}$ where d^* is the largest depth of any node expanded by OPδ.

Proof: By definition of the algorithm all sequences expanded satisfy the first condition $\boldsymbol{u}_d \in \boldsymbol{U}_\delta$. Further, for any finite tree there exists some leaf sequence $\boldsymbol{u}'$ so that $b(\boldsymbol{u}') \geq v_\delta^*$, and since $\boldsymbol{u}^\dagger$ maximizes the b-value, $b(\boldsymbol{u}^\dagger) \geq v_\delta^*$, or equivalently $l(\boldsymbol{u}^\dagger) + \frac{\gamma^d}{1-\gamma} \geq v_\delta^*$. This implies $v_\delta(\boldsymbol{u}^\dagger) + \frac{\gamma^d}{1-\gamma} \geq v_\delta^*$, the same as the second condition in (6). So finally $\boldsymbol{u}^\dagger \in \mathcal{T}_\delta^*$. The remainder of the lemma is proven like Theorem 2(i), by replacing all the unconstrained values v by constrained ones. ■

So far the analysis simply established that OPδ preserves some interesting properties of OP. The main novelty in our analysis follows: studying the new gap ε obtained by the constrained algorithm. To this end, the cardinality of the near-optimal tree must be characterized using a new complexity measure, which is defined as follows.

Definition 5: The complexity measure is the smallest value of K for which there exists a constant $c > 0$ so that $\left|\mathcal{T}_{d,\delta}^*\right| \leq c \cdot K^{d/\delta},\ \forall d \geq 0$.

Here $\mathcal{T}_{d,\delta}^*$ denotes the nodes of $\mathcal{T}_\delta^*$ at depth d. Note that due to the special cases below, a K always exists and belongs to the interval $[1, M\delta]$ (it may be non-integer). Constant K plays a similar role to the branching factor κ in the unconstrained problem, and in some cases a relationship between the two quantities can be found, as we show later. Our results hold for any pair c, K, but we take the smallest K. Using K, the gap ε is characterized as follows.

Theorem 6: Given a computational budget n, OPδ algorithm produces a gap $\varepsilon = O(n^{-\delta \frac{\log 1/\gamma}{\log K}})$ if $K > 1$, and $\varepsilon = O(\gamma^{\frac{n}{c}})$ when $K = 1$, where c is the constant from the definition of K.

Proof: Define d_n to be the smallest depth so that $n \leq \sum_{i=0}^{d_n} \left|\mathcal{T}_{d,\delta}^*\right|$; this means the algorithm has expanded nodes at d_n (perhaps not yet at $d_n + 1$), so $d^* \geq d_n$ and $\varepsilon \leq \frac{\gamma^{d_n}}{1-\gamma}$.

If $K > 1$, then[2] $n \leq \sum_{i=0}^{d_n} cK^{d/\delta} = c\frac{(K^{1/\delta})^{d+1}-1}{K^{1/\delta}-1} \leq c_1 K^{d/\delta}$, from which $d_n \geq \delta \frac{(\log n - \log c_1)}{\log K} \geq \delta \log n / \log K - c_2$. Thus, after some manipulations $\varepsilon \leq c_3 n^{-\delta \frac{\log 1/\gamma}{\log K}}$.

If $K = 1$, then $n \leq \sum_{i=0}^{d_n} c \leq c(d_n + 1)$, and $d_n \geq \frac{n-1}{c}$ leading to $\varepsilon \leq \gamma^{\frac{n-1}{c}}$. The theorem is proven. ■

While we measure complexity by the number n of nodes expanded, the number of children of a node may be either 1 or M, so the computational cost of expansion varies. Nevertheless, this only amounts to a constant factor in the relationships, and so it does not affect the asymptotic analysis. Next, we find the complexity measure K and illustrate its relation to κ in two interesting cases.

Case 1: All sequences optimal: Consider a problem where all the rewards are identical, say equal to 1 or to 0. While any sequence is optimal in this problem, it is nevertheless an interesting worst case, which highlights the (correct) behavior of the algorithm in general. In this case

[2]We denote by c_i positive constants whose value is irrelevant to the asymptotic analysis.

the algorithm must explore the entire tree uniformly, in the order of depth. Counting the size of this tree is tedious but straightforward, and we skip the proof. The result is $\left|\mathcal{T}^*_{d,\delta}\right| \leq M^2\delta(\delta M)^{\frac{d}{\delta}}$, so $K = M\delta$. Since $\mathcal{T}^*_{d,\delta}$ has the largest possible size, $M\delta$ is also the upper limit of the possible values of K.

Comparing to OP, K equals δ times the branching factor M in the OP tree. The two algorithms explore trees that grow exponentially with the depth, but have different size. Consider the resulting rates: $\varepsilon = O(n^{-\frac{\log 1/\gamma}{\log M}})$ for OP, and $\varepsilon = O(n^{-\delta\frac{\log 1/\gamma}{\log M\delta}})$ for OPδ. Since $\delta\frac{\log 1/\gamma}{\log M\delta} \geq \frac{\log 1/\gamma}{\log M}$ for all $M, \delta \geq 2$, OPδ converges faster in this worst-case sense. Of course, this does not mean that OPδ is faster for any given particular problem, and in fact the relationship varies. □

Case 2: One optimal sequence: In this case, a single sequence has maximal rewards (equal to 1), and all other transitions have a reward of 0. Here, two situations are possible. If the optimal sequence is within the constrained set, OPδ always expands this sequence further, we have $\left|\mathcal{T}^*_{d,\delta}\right| = 1$ and $K = 1$, the easiest type of problem. This also leads to the lower limit of 1 for K. In this situation, the original OP explores the same path so $\kappa = 1$. Thus the best-case convergence rate of the two algorithms is the same – exponential.

Otherwise, the optimal sequence leaves the constrained set at a node $\boldsymbol{u}_0$ at some finite depth, and then the algorithm must explore uniformly the subtree having $\boldsymbol{u}_0$ at the root (perhaps in addition to some other nodes). Then, since the analysis is asymptotic, for large depths, K has the maximal value of $M\delta$ again. Since OP is still allowed to refine the optimal sequence, $\kappa = 1$ and here introducing the constraint has made the problem significantly *more* complex. □

Having completed the analysis of generic OPδ, its properties in the context of PO and PW for switched systems are summarized in the following direct adaptation of Corrolary 3. The differences are that the values become constrained, and the convergence rates change to those of OPδ.

Corrolary 7: (i) When applied to PO, OPδ returns bounds $\underline{l}, \underline{b}$ so that the optimal value $\underline{J}_\delta := \inf_{\boldsymbol{u}_\infty \in \boldsymbol{U}_\delta} J(\boldsymbol{u}_\infty)$ is in the interval $[G(\frac{1}{1-\gamma} - \underline{b}), G(\frac{1}{1-\gamma} - \underline{l})]$, as well as a sequence $\underline{\boldsymbol{u}}$ that achieves these bounds. The gap is $G\underline{\varepsilon} = O(n^{-\delta\frac{\log 1/\gamma}{\log \underline{K}}})$ when $\underline{K} > 1$, or $O(\gamma^{n/c})$ when $\underline{K} = 1$.

(ii) In PW, OPδ returns bounds $\overline{l}, \overline{b}$ so that the worst-case value $\overline{J}_\delta := \sup_{\boldsymbol{u}_\infty \in \boldsymbol{U}_\delta} J(\boldsymbol{u}_\infty)$ is in the interval $[G\overline{l}, G\overline{b}]$, as well as a sequence $\overline{\boldsymbol{u}}$ that achieves these bounds. The gap is $G\overline{\varepsilon} = O(n^{-\delta\frac{\log 1/\gamma}{\log \overline{K}}})$ when $\overline{K} > 1$, or $O(\gamma^{n/c})$ when $\overline{K} = 1$.

The following remark is important for receding-horizon applications. If a switch occurs at step k, then to guarantee the dwell-time constraint the mode must be kept constant (keeping the loop open) until $k + \delta - 1$, so OPδ only needs to be called again at step $k + \delta$. Note that this means some of the nodes at $d = 1$ have dwell time 1 so they become constrained, unlike our simplifying assumption stated before Algorithm 2, which holds at $k = 0$. However this is easy to take into account in the implementation.

V. Simulation Results

We evaluate our approach for an optimal control problem PO in a nonlinear switched system, and for worst-case disturbance PW in a linear switched system.

A. Optimal control for nonlinear modes

Consider PO for the double-tank system with nonlinear modes from [20]. The two states of the system correspond to the fluid levels in an upper and a lower tank. The output of the upper tank flows into the lower tank, the output of the lower tank exits the system, and the flow into the upper tank is restricted to be either 1 or 2. The two modes have continuous-time dynamics:

$$\dot{x}(t) = \begin{bmatrix} 1 - \sqrt{x_1(t)} \\ \sqrt{x_1(t)} - \sqrt{x_2(t)} \end{bmatrix}, \dot{x}(t) = \begin{bmatrix} 2 - \sqrt{x_1(t)} \\ \sqrt{x_1(t)} - \sqrt{x_2(t)} \end{bmatrix}$$

The cost is defined as $(x - x^*)^\top Q(x - x^*)$ with $x^* = [0, 3]^\top$, and $Q = \mathrm{diag}(0, 2)$, so the first state is not optimized and the second must reach value 3. Different from [20], we numerically integrate the dynamics over sampling intervals of $T_\mathrm{s} = 0.1$ to obtain the discrete-time modes f_1 and f_2, see again (1).

We examine the effect of the computational budget n on performance, which is measured by the undiscounted cost along the trajectory in order to be consistent with usual formulations of optimal control of switched systems. OP is run for a range of budgets from 10 to 200 in increments of 2, using the initial state $x_0 = [2, 2]^\top$, $\gamma = 0.98$ and a trajectory length of 20 s. Figure 3, top reports the results, showing that the cost decreases with larger budgets as expected, although the differences are small, showing that the problem is simple enough to be solved well with small budgets. Note that the cost no longer decreases for significantly larger budgets, which indicates the solution is likely already optimal (for discounted costs). Figure 3, bottom shows the trajectory for $n = 200$, which stabilizes the level to 3 in around 6 s, like the approach in [20].

B. Worst-case disturbance with a dwell-time constraint

Next, we illustrate a problem of type PW: worst-case disturbance. We borrow the example of [5], having two linear modes $A_1 = e^{B_1 T_\mathrm{s}}$ and $A_2 = e^{B_2 T_\mathrm{s}}$ with:

$$B_1 = \begin{bmatrix} 0 & 1 \\ -10 & -1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 1 \\ -0.1 & -0.5 \end{bmatrix}$$

The sampling time is $T_\mathrm{s} = 0.5$, the cost is quadratic with $Q = I$ and the initial state is $x_0 = [1, 1]^\top$. In [5] stability is guaranteed under a minimum dwell-time of $\delta = 6$, and we take advantage of this guarantee by applying the constrained algorithm OPδ with $\delta = 6$, and keeping the very first mode constant for 6 steps. We select $\gamma = 0.98$, a budget $n = 2000$, an experiment length of 300 s, and derive G by taking bounds of 30 for both state variables; these bounds are never reached in the experiments. The undiscounted cost obtained by running the algorithm in receding horizon is 142.10, close to the upper bound 152.17 obtained by [5]. Note that we reached our value by *designing* a switching control, whereas

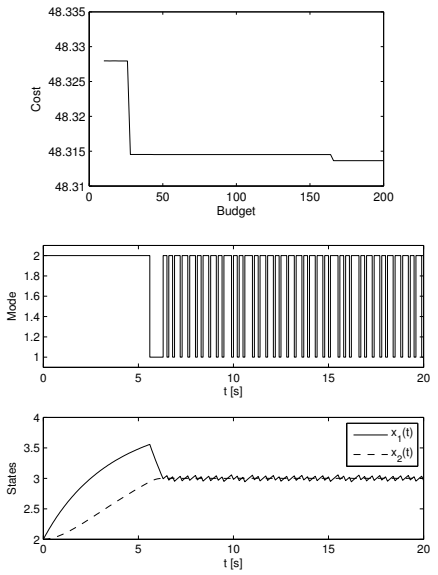

Fig. 3. Top: Influence of computational budget for the nonlinear tanks. Bottom: Control and state trajectories in the same problem.

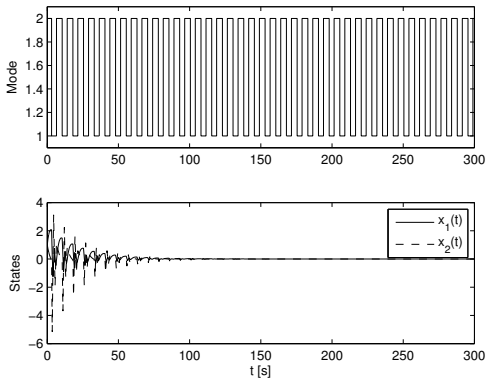

Fig. 4. Controlled trajectory for worst-case disturbance.

[5] do not (but have the advantage of providing a stability analysis). Figure 4 illustrates the results.

VI. Conclusions and future work

We provide an approach to optimize discounted costs in discrete-time switched systems with possibly nonlinear modes, which is able to optionally include a minimum dwell-time constraint. This approach provides upper and lower bounds on either the optimal cost when the switches are controlled, or on the worst-case cost when the switches are a disturbance. The convergence rate of the gap between bounds as a function of computation is characterized.

An important future direction is an explicit treatment of stability guarantees, either by connecting with existing conditions in the switched systems literature, or with our approach for systems without switches in [16]. Approaches can be developed for systems where switches occur stochastically, or where some of the switches are controlled and some are a disturbance. Suitable planning methods exist [3], [15]. OPδ can also be modified to handle a maximum dwell-time, which requires novel complexity analysis as in Section IV-B.

References

[1] D. Antunes, W. Heemels, and P. Tabuada, "Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design," in *IEEE Conference on Decision and Control, Hawai: U.S.A.*, 2012, pp. 7212–7217.

[2] D. P. Bertsekas and S. E. Shreve, *Stochastic Optimal Control: The Discrete Time Case*. Academic Press, 1978.

[3] L. Buşoniu, E. Páll, and R. Munos, "An analysis of optimistic, best-first search for minimax sequential decision making," in *2014 IEEE International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL-14)*, Orlando, 10–12 December 2014.

[4] J. Filar, V. Gaitsgory, and A. Haurie, "Control of singularly perturbed hybrid stochastic systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 2, pp. 179–190, 2001.

[5] J. C. Geromel and P. Colaneri, "Stability and stabilization of discrete-time switched systems." *International Journal of Control*, vol. 79, no. 7, pp. 719–728, 2006.

[6] J. C. Geromel and R. H. Korogui, "H2 robust filter design with performance certificate via convex programming," *Automatica*, vol. 44, pp. 937–948, 2008.

[7] D. Henrion, J. Daafouz, and M. Claeys, "Optimal switching control design for polynomial systems: an lmi approach," in *Proceedings of the IEEE Conference on Decision and Control (CDC-13)*, 2013.

[8] J.-F. Hren and R. Munos, "Optimistic planning of deterministic systems," in *Proceedings 8th European Workshop on Reinforcement Learning (EWRL-08)*, Villeneuve d'Ascq, France, 30 June – 3 July 2008, pp. 151–164.

[9] G. D. J. C. Geromel and J. Daafouz, "Suboptimal switching control consistency analysis for switched linear systems," *IEEE Transactions on Automatic Control*, vol. 58, pp. 1857–1861, 2013.

[10] M. Jungers and J. Daafouz, "Guaranteed cost certification for discrete-time linear switched systems with a dwell time," *IEEE Transactions on Automatic Control*, vol. 58, no. 3, pp. 768–772, 2013.

[11] K. Katsikopoulos and S. Engelbrecht, "Markov decision processes with delays and asynchronous cost collection," *IEEE Transactions on Automatic Control*, vol. 48, no. 4, pp. 568–574, 2003.

[12] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M.-B. Naghibi-Sistani, "Reinforcement q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, 2014, appeared online.

[13] D. Liberzon, *Switching in Systems and Control.*, ser. Systems and Control: Foundations and Applications. Birkhauser, 2003.

[14] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 308–322, 2009.

[15] R. Munos, "The optimistic principle applied to games, optimization and planning: Towards foundations of Monte-Carlo tree search," *Foundations and Trends in Machine Learning*, vol. 7, no. 1, pp. 1–130, 2014.

[16] R. Postoyan, L. Buşoniu, D. Nešić, and J. Daafouz, "Stability of infinite-horizon optimal control with discounted cost," in *Proceedings 53nd Conference on Decision and Control (CDC-14)*, Los Angeles, USA, 15–17 December 2014.

[17] P. Riedinger, C. Iung, and F. Kratz., "An optimal control approach for hybrid systems," *European Journal of Control*, vol. 9, pp. 449–458, 2003.

[18] M. S. Shaikh and P. Caines, "On the hybrid optimal control problem: theory and algorithms," *IEEE Transactions on Automatic Control*, vol. 52, pp. 1587–1603, 2007.

[19] R. Shorten, F. Wirth, O. Mason, K. Wulff, and C. King, "Stability criteria for switched and hybrid systems," *Automatica*, vol. 49, no. 7, pp. 545–592, 2007.

[20] R. Vasudevan, H. Gonzalez, R. Bajcsy, and S. S. Sastry, "Consistent approximations for the optimal control of constrained switched systems," *SIAM Journal on Control and Optimization*, 2012.

[21] J. H. W. Zhang and A. Abate, "Infinite-horizon switched lqr problems in discrete time: A suboptimal algorithm with performance analysis," *IEEE Transactions on Automatic Control*, vol. 57, pp. 1815–1821, 2012.

[22] F. Zhu and P. J. Antsaklis, "Optimal control of switched hybrid systems: a brief survey," *Discrete Event Dynamic Systems*, vol. 25, no. 3, pp. 345–364, 2015.