Linear Quadratic Regulation of Switched Systems Using Informed Policies

Duarte Antunes, *Member, IEEE*, and W. P. Maurice Heemels, *Fellow, IEEE*

***Abstract*—The problem of designing a switching and control policy for regulating the state of a switched linear system to zero while minimizing a quadratic cost appears in numerous applications. However, obtaining the optimal policy is in general computationally intractable. Here, we propose a class of suboptimal policies that exploit information, in terms of upper or lower bounds, on the optimal cost. We analyze the performance of these novel policies, obtaining new bounds on the optimal cost which are tighter than the initial ones. The usefulness of these policies and performance bounds is illustrated in the context of resource-aware control.**

***Index Terms*—Optimal control, dynamic programming, approximation algorithms, switched systems, linear quadratic regulator.**

I. Introduction

IN THIS paper we consider the following switched linear system (SLS)

$$x_{k+1} = A_{\sigma_k} x_k + B_{\sigma_k} u_k, \quad k \in \mathbb{N}_0, \qquad (1)$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^{n_u}$ is the control input, and $\sigma_k \in \{1, 2, \dots, m\}$ is the switching input at discrete time $k \in \mathbb{N}_0 := \{0, 1, 2, \dots\}$. Our goal is to design a policy for the control and the switching input that regulates the state to zero while minimizing a quadratic cost.

The control of SLSs arises in various applications such as mixing of fluids [2], DC-DC power conversion [3], event-triggered control [4], supervisory control [5], task scheduling for real-time control [6], viral mutation in HIV treatment [7], mobile sensor networks [8], damping of vibrating structures [9], multihop networked control [10], protocol design for networked control [11], and several others. Model (1) captures many of these applications and can be used as an approximation to study several others by linearization and discretization of related nonlinear and continuous-time models. The problem of minimizing a quadratic cost parallels the well-known LQR problem [12], which is one the tools per excellence for regulating/stabilizing (1) in the special case $m = 1$, where only the control input is to be designed.

However, computing the optimal policy for the control and the switching inputs when $m > 1$ is in general computationally intractable. In fact, a direct application of the well-known iterative dynamic programming algorithm leads to value functions of exponentially growing complexity [13]. Thus, one must settle for suboptimal policies.

The works [13]–[15] propose suboptimal policies under very mild assumptions. The policy in [14] results from an iterative relaxed dynamic programming algorithm; a relaxation parameter, specifying an acceptable loss of performance (measured by the quadratic cost) with respect to the optimal strategy, allows to trade performance and computation complexity. The work [13] proposes a similar relaxed value function iterative method, formally establishing that a stabilizing stationary policy can be found after enough iterations if the relaxation parameter is sufficiently small. In addition, [13] provides a bound on the cost of such a policy with respect to the optimal policy. Relaxed dynamic programming is also used in [15] to propose an explicit receding horizon policy to the problem. However, the computation complexity of these methods can still be large.

Other works in the literature propose policies with reduced complexity under additional assumptions. In [16], [17], assuming the existence of a stabilizing base policy corresponding to a periodic switching input, a low complexity rollout policy [18, Ch. 6] is proposed whose performance is upper bounded by that of the base policy. For the special case where there is no control input (the input matrices B_j in (1) are identically zero), and assuming the existence of a solution to a set of Lyapunov-Metzler inequalities, [19] provides a simple stabilizing policy that also achieves an upper bound on an infinite horizon quadratic cost. Similar results have been recently reported in [20] for the general case where the B_j are not zero, often referred to as the codesign problem, imposing a linear state-feedback policy for the control input. Under the more restrictive condition that at least one of the subsystems is stable (when $\sigma_k = i$, for a given i and for all k, and gains B_j are all identically zero), [21], [22] propose a related consistent policy, i.e., a policy that achieves a smaller (or equal) quadratic cost than the one achieved by each of the (stable) subsystems.

Manuscript received November 6, 2015; revised July 24, 2016 and July 26, 2016; accepted September 26, 2016. Date of publication October 11, 2016; date of current version May 25, 2017. A subset of the results was presented in [1]. The present work provides new results and extensions in several directions. This work is supported by the Dutch Science Foundation (STW) and the Dutch Organization for Scientific Research (NWO) under the VICI grant "Wireless controls systems: A new frontier in Automation" (No. 11382). Recommended by Associate Editor R. M. Jungers.

The Authors are with the Control Systems Technology Group, Department of Mechanical Engineering,Eindhoven University of Technology, The Netherlands (e-mail: D.Antunes@tue.nl; M.Heemels@tue.nl).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2016.2616388

There are several other related works that consider more general models and different problems from the ones considered here and in [13]–[17], [19]–[22]. For example, model predictive control, considering state and input constraints, has been proposed to tackle switching and control problems for the large class of mixed logical systems in [23] and in the context of hierarchical control systems [24]; several results are given in [25], [26] for dual switching systems for which the matrices A_i and B_i in (1) also depend on a stochastic parameter modeled by a Markov chain; [27], [28] study a similar model to (1) but with additive white disturbances. The results in these papers are mostly of a different nature to the ones provided here and in [13]–[17], [19]–[22], since the latter pertain to a stronger characterization of performance for (1) considering deterministic, rather than stochastic, notions of performance.

In this paper we propose and analyze a class of linear quadratic regulators parameterized by a set of positive semidefinite matrices, which defines a piecewise quadratic function $\hat{J}$ approximating the optimal cost. Our aim is to provide a framework for deriving policies with lower complexity and tighter performance guarantees than in previous works, by allowing some information on the optimal cost, typically available in applications, to be incorporated in the approximation $\hat{J}$. To this effect, our novel results provide upper and lower bounds on the cost difference between $\hat{J}$ and the cost of the proposed policy. Choosing $\hat{J}$ as the cost of a given base policy we recover the rollout policies in [4], [16], [17]. We can then use our novel results to assert the gain obtained by a rollout policy over a base policy, tightening the bound provided by $\hat{J}$ on the optimal cost. On the other hand, choosing $\hat{J}$ as a lower bound on the optimal cost, we can bound the distance of the resulting policy from the optimal. We term the latter policies by optimistic policies and the general class by informed policies. This terminology is inspired by a connection with informed search methods [29, Ch. 3], which is discussed below (see Section IV).

We illustrate the usefulness of the novel policies and results in two applications of resource-aware control, namely in an event-triggered control setting proposed in [4] and in the context of simultaneously controlling several processes assuming that only one can be controlled at a given time [6], [14]. The results in this paper lead to policies with reduced complexity and better performance than in previous works.

The paper is organized as follows. Section II presents the proposed class of linear quadratic regulators considering an infinite horizon cost and Section III provides several results for analyzing the performance of these regulators. Section IV briefly discusses the case of a finite horizon cost. Section V presents numerical examples and Section VI concludes the paper. The proofs of the theorems are given in the Appendix.

II. A Class of Linear Quadratic Regulators

Consider the quadratic cost

$$\sum_{k=0}^{\infty} g(x_k, u_k, \sigma_k), \tag{2}$$

for (1), where

$$g(x,u,i) := \begin{bmatrix} x^\intercal & u^\intercal \end{bmatrix} \begin{bmatrix} Q_i & S_i \\ S_i^\intercal & R_i \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{3}$$

is assumed to be positive semidefinite for every $i \in \mathcal{M} := \{1, 2, \dots, m\}$. The optimal cost for an initial condition x_0, obtained by taking the infimum of (2) with respect to $\{u_k, \sigma_k | k \in \mathbb{N}_0\}$, is denoted by $J^*(x_0)$. The objective is to find a *feedback* policy μ for (1), consisting of a function from the state to the control and switching inputs, such that (2) is minimized when

$$(u_k, \sigma_k) = \mu(x_k), \quad k \in \mathbb{N}_0. \tag{4}$$

Note that we consider μ to be independent of time. In Section IV we also consider a finite horizon version of this problem where we allow μ to be time-dependent.

As explained in [13], if (1) is exponentially stabilizable (see [30]) it is possible to find a policy for which the cost (2) is arbitrarily close to the optimal cost for every initial condition x_0 when the control and switching inputs are given by (4). Such a policy can be obtained by running the iterative dynamic programming algorithm

$$V_{k+1}(x) = \min_{u \in \mathbb{R}^{n_u}, i \in \mathcal{M}} g(x,u,i) + V_k(A_i x + B_i u),$$

with $V_0(x) = 0$, for a typically large number of iterations $k \in \{0, 1 \dots, N\}$, and is specified by

$$\mu^*(x) \in \operatorname*{argmin}_{u \in \mathbb{R}^{n_u}, i \in \mathcal{M}} g(x,u,i) + V_{N+1}(A_i x + B_i u), \tag{5}$$

where we use $\in$ to indicate that the minimum might not be unique. However, one can show that $V_k(x) = \min_{X \in \Pi_k} x^\intercal X x$ where Π_k are sets of positive semidefinite matrices whose cardinality grows exponentially with k (cf. [13]). This indicates that finding an optimal policy is computationally intractable.

Motivated by the difficulties in obtaining the optimal policy, in this paper, we propose and analyze a class of suboptimal stationary policies (4), parameterized by a set of n_T positive semidefinite matrices T_ℓ, $\ell \in \mathcal{T} := \{1, 2, \dots, n_T\}$. Given this set we define

$$\hat{J}(x) = \min_{\ell \in \mathcal{T}} x^\intercal T_\ell x, \tag{6}$$

and

$$\bar{J}_\mu(x) := \min_{u \in \mathbb{R}^{n_u}, i \in \mathcal{M}} g(x,u,i) + \hat{J}(A_i x + B_i u). \tag{7}$$

The function $\hat{J}$ can be interpreted as an *approximation* of the optimal cost and the proposed policy μ selects one of the (possibly many) control and switching inputs that achieve the minimum in (7), for each state x. Note that the choice $\hat{J} = V_{N+1}$ would lead to the (high complexity) policy μ^*, described above. However, we can obtain several low complexity policies by allowing $\hat{J}$ to incorporate information on the optimal cost, and in particular by choosing $\hat{J}$ as a lower or upper bound on J^*. Several examples will be discussed shortly.

The proposed policy μ can be characterized more explicitly as follows. For a square matrix $T \in \mathbb{R}^{n \times n}$, and $i \in \mathcal{M}$, let

$$G_i(T) := -(B_i^\intercal T B_i + R_i)^\dagger (B_i^\intercal T A_i + S_i^\intercal),$$

where the symbol † denotes the pseudo-inverse, and

$$F_i(T) := A_i^\intercal T A_i + Q_i - G_i(T)^\intercal (B_i^\intercal T B_i + R_i) G_i(T). \tag{8}$$

Moreover, for $n_P := m n_T$ and $\mathcal{P} := \{1, 2, \ldots, n_P\}$, let

$$\{P_j | j \in \mathcal{P}\}, \quad \{K_j | j \in \mathcal{P}\}, \tag{9}$$

be indexations of the sets

$$\{F_i(T_\ell) | \ell \in \mathcal{T}, i \in \mathcal{M}\}, \quad \{G_i(T_\ell) | \ell \in \mathcal{T}, i \in \mathcal{M}\},$$

respectively. Each value $j \in \mathcal{P}$ corresponds to a unique T_ℓ, $\ell \in \mathcal{T}$, and to a unique $i = \pi(j) \in \mathcal{M}$, where $\pi : \mathcal{P} \to \mathcal{M}$ is a map characterizing the latter correspondence. Then the minimum in (7) is achieved by $(u, i) = \mu(x)$, for

$$\mu(x) = (K_{\iota(x)} x, \pi(\iota(x))), \tag{10}$$

where

$$\iota(x) = \min \operatorname*{argmin}_{j \in \mathcal{P}} \; x^\intercal P_j x. \tag{11}$$

Then (7) can be written as $\bar{J}_\mu(x) = x^\intercal P_{\iota(x)} x$. Note that in (11) we have arbitrated that the smallest index is selected if for a given x the minimum of $x^\intercal P_j x$ is achieved by more than one index $j \in \mathcal{P}$, to make the policy well-defined. However, any other choice would be possible without influencing the results of the present paper. Note also that (10), (11) is in general a choice among the control inputs that achieve the minimum in (7). One such case of interest is when $B_i = 0$ for every $i \in \mathcal{M}$, i.e., only a policy for the switching input is to be designed.

The complexity of the proposed policy both in terms of memory required and computational time is dictated by n_P. In fact, to evaluate (10), (11), one needs to compute *a priori* the two sets of n_P matrices (9) and evaluate n_P quadratic functions (11) at each time step k, requiring $n(n+1)/2$ multiplications to compute the products $x_i x_j$ and $n_P n(n+1)/2$ multiplications to evaluate (11).

Next we show how, by picking $\hat{J}$, we can define informed policies, such as rollout and optimistic policies, which will be the focus of the investigation in the paper. The terminology for these policies is further justified in Section IV.

A. Rollout Policies

Suppose that we know a *stabilizing* policy characterized by a fixed sequence of schedules $b = (b_0, b_1, \ldots)$, i.e., $\sigma_k = b_k$, $k \in \mathbb{N}_0$, *independent* of the state, and a linear feedback control input policy $u_k = K_k x_k$, $k \in \mathbb{N}_0$. For concreteness, we consider that b is periodic and the optimal control input policy is picked, although we could also select other sequences and linear feedback policies. We assume (1) is stabilizable (defined for periodic systems in [31, Def. 4.5]) when $\sigma_k = b_k$, for $b_k = b_{k+T}$, $k \in \mathbb{N}_0$, and a period T. In this case, the optimal control input stabilizes (1) and $K_k = G_{b_{k \bmod T}}(X_{b_{k+1 \bmod T}})$ where k mod T denotes the remainder after division of k by T, and X_i are the unique positive definite solutions to

$$X_i = F_{b_i}(X_{i+1}), \quad i \in \{0, 1, \ldots, T-2\}, \tag{12}$$

and $X_{T-1} = F_{b_{T-1}}(X_0)$ (cf. [31, Prop. 13.9]). We refer to this policy as the base policy and note that in many problems of interest for SLSs there exist natural choices of base policies. For instance, in event-triggered control a natural base policy is optimal periodic control [4]; in mixing of fluids, periodic mixing [2]; in protocol design for networked control, round-robin protocols [11], [17].

The cost (2) when this *base policy* is taken for (1) is $x_0^\intercal M_b x_0$, where M_b equals the first matrix X_0 of the solution to (12); we use the notation M_b to indicate the dependence on b. We then pick (6) as

$$\hat{J}(x_0) = \min_{b \in \mathcal{B}} x_0^\intercal M_b x_0, \tag{13}$$

where $\mathcal{B}$ denotes a set of sequences each characterizing a stabilizing base policy. We can see (13) as a special base policy consisting of the minimum of base policies and the associated policy (10) can then be seen as a rollout policy using a one-step lookahead horizon [18, Ch. 6]. Rollout policies share common features with model predictive control (MPC) policies [32] and in particular optimize control (and in this case also switching) inputs over a lookahead time horizon, operating in a receding horizon fashion [18]. However, for rollout policies after the lookahead horizon, a base policy is assumed to be used. Note that the one-step lookahead optimization in (10) takes an explicit dependence on the state as in explicit model predictive control [33]; h- step lookahead policies for $h \in \mathbb{N}_{\geq 1}$ will be discussed below.

For one-step lookahead policies we can conclude that

$$\bar{J}_\mu(x_0) = \min_{c \in \mathcal{C}} x_0^\intercal M_c x_0,$$

where, as before, $x_0^\intercal M_c x_0$ is the cost of a base policy associated with c and

$$\mathcal{C} := \{(i, b) | i \in \mathcal{M}, b \in \mathcal{B}\} \tag{14}$$

is determined by the choice of $\mathcal{B}$. We will often assume that the set $\mathcal{B}$ is such that

$$\mathcal{B} \subseteq \mathcal{C}. \tag{15}$$

For example, if $m = 2$, $\mathcal{B} = \{(1, 2, 1, 2, 1, \ldots)\}$ does not satisfy (15), since in such a case $\mathcal{C} = \{(1, 1, 2, 1, \ldots), (2, 1, 2, 1, \ldots)\}$, but $\mathcal{B} = \{(2, 1, 2, 1, 2, \ldots), (1, 2, 1, 2, 1, \ldots)\}$ does, since in such a case $\mathcal{C}$ includes $\mathcal{B}$. Note that this assumption implies that $\bar{J}_\mu(x) \leq \hat{J}(x)$, for every $x \in \mathbb{R}^n$. This property will turn out to be crucial to analyze the stability and performance of rollout policies.

We discuss next two variants of rollout policies also captured by (10).

1) h-Step Lookahead Horizon Policies: For rollout policies with h-step lookahead horizon, at each time k, the switching and control inputs are optimized over a horizon $k, k + 1 \ldots, k + h - 1$ assuming that after time $k + h - 1$ the base policy is used. The first control and switching inputs are then applied and the process is repeated at time $k + 1$ in a receding horizon fashion. Formally, we can write this policy in terms of the first switching and control inputs,

$$\mu(x) = (u_{k|k}(x), \sigma_{k|k}(x)),$$

resulting from the following optimization problem:

$$\min_{u_{k|k},\sigma_{k|k},\dots,u_{k+h-1|k},\sigma_{k+h-1|k}} \sum_{\ell=k}^{k+h-1} g(x_{\ell|k},u_{\ell|k},\sigma_{\ell|k}) + \min_{b\in\mathcal{B}} x_{k+h|k}^{\intercal} M_b x_{k+h|k}$$

with initial condition $x_{k|k}=x$, where the state $x_{\ell|k}$, control inputs $u_{\ell|k}$, and switching inputs $\sigma_{\ell|k}$ satisfy (1) for $\ell \in \{k, k+1, \dots, k+h-1\}$. One can conclude that $u_{k|k}, \sigma_{k|k}$ can equivalently be obtained by solving

$$\min_{u,i} g(x,u,i) + \min_{d\in\mathcal{D}} (A_i x + B_i u)^{\intercal} M_d (A_i x + B_i u),$$

where $\mathcal{D} := \{(i_1,\dots,i_{h-1},b) | i_\ell \in \mathcal{M}, b \in \mathcal{B}\}$, which takes the form (10) with

$$\hat{J}(x) = \min_{d\in\mathcal{D}} x^{\intercal} M_d x_d. \tag{16}$$

We then conclude that a general h-lookahead horizon policy can be interpreted as a 1-step lookahead horizon policy with a different base policy associated with the switching sequences in $\mathcal{D}$. Note also that

$$\{M_d | d\in\mathcal{D}\} = \{F_{i_1}(F_{i_2}(\dots F_{i_h-1}(M_b))) | i_\ell \in \mathcal{M}, b\in\mathcal{B}\}, \tag{17}$$

where the map F was defined in (8).

2) h-Lifted Policies: In h-lifted rollout policies the optimization to choose control and switching inputs at given times $k \in \{0, h, 2h, \dots\}$ is similar to the one in h-step lookahead policies. However, these inputs are actually applied at times $k+1, k+2, \dots, k+h-1$ (in addition to time k), instead of being recomputed. We can formalize this policy using the lifted system of (1) over h time steps. For $h \in \mathbb{N}_{\geq 2}$ we write (1) as

$$\underline{x}_{\ell+1} = \underline{A}_{\underline{\sigma}_\ell} \underline{x}_\ell + \underline{B}_{\underline{\sigma}_\ell} \underline{u}_\ell, \quad \ell \in \mathbb{N}_0, \tag{18}$$

where $\underline{x}_\ell = x_{\ell h}$, $\underline{u}_\ell = [u_{\ell h}^{\intercal} \dots u_{(\ell+1)h-1}^{\intercal}]^{\intercal}$, and $\underline{\sigma}_\ell \in \mathcal{M}_h := \{1, 2, \dots, m^h\}$, is such that each $\underline{\sigma}_\ell = \underline{i} \in \mathcal{M}_h$ corresponds to a unique $(\sigma_0, \dots, \sigma_{h-1}) \in \mathcal{M}^h := \mathcal{M} \times \cdots \times \mathcal{M}$. Moreover, (2) can be written as

$$\sum_{\ell=0}^{\infty} \underline{g}(\underline{x}_\ell, \underline{u}_\ell, \underline{\sigma}_\ell) \tag{19}$$

for a positive semidefinite quadratic function $\underline{g}$. The expressions for $\underline{A}_{\underline{i}}$, $\underline{B}_{\underline{i}}$, $\underline{i} \in \mathcal{M}_h$, and $\underline{g}$ are omitted for brevity. We define a policy for this lifted system as in (10)–(11), i.e.,

$$\underline{\mu}(\underline{x}) = (\underline{u}(\underline{x}), \underline{i}(\underline{x})) \tag{20}$$

where for $\underline{x} \in \mathbb{R}^n$, $\underline{u}(\underline{x}) \in \mathbb{R}^{n_u h}$, $\underline{i}(\underline{x}) \in \mathcal{M}_h$ are control and switching input values attaining the minimum of $\underline{g}(\underline{x}, \underline{u}, \underline{i}) + \hat{J}(\underline{A}_{\underline{i}} \underline{x} + \underline{B}_{\underline{i}} \underline{u})$ and $\hat{J}$ is given by the general form (6), which for rollout policies takes the form (13). We then define the *lifted policy* for the *original* system (1) by applying the control and schedules

$$(u_k, \sigma_k) = (\underline{v}_{k-\ell h}, \underline{s}_{k-\ell h}),$$

for $k \in \{\ell h, \dots, (\ell+1)h - 1\}$, where

$$(\underline{v}, \underline{i}_s) = \underline{\mu}(x_{\ell h}), \tag{21}$$

and $\underline{s}$ is the element in $\mathcal{M}^h$ corresponding to $\underline{i}_s \in \mathcal{M}_h$. Note that (21) is only computed at times $k = \ell h$, $\ell \in \mathbb{N}_0$.

B. Optimistic Policies

We denote by optimistic policies those for which $\hat{J}(x)$ is a lower bound on the optimal cost (2), i.e.,

$$\hat{J}(x) \leq J^*(x) \tag{22}$$

for every $x \in \mathbb{R}^n$. Such lower bounds are known in many applications of SLSs. Prime examples are applications in which (1), (2) result from the discretization of a continuous-time optimal control problem

$$\min_{u_C(t)} \int_0^\infty x_C(t)^{\intercal} Q_C x_C(t) + u_C(t)^{\intercal} R_C u_C(t) dt \tag{23}$$

for positive-definite Q_C and R_C and a controllable system

$$\dot{x}_C(t) = A_C x_C(t) + B_C u_C(t), \tag{24}$$

for an initial condition $x_C(0) = x_0$. The optimal input for this problem is given by $u_C(t) = K_C x_C(t)$, where $K_C = -R_C^{-1} B_C^{\intercal} P_C x_C(t)$ and P_C is the solution to a Riccati equation and results in a cost $x_0^{\intercal} P_C x_0$ (cf. [18, Ch. 3]). However, in several applications this optimal control input cannot be applied due to constraints in the problem. This is the case, e.g., in networked control where the controller communicates with the process via a resource constrained network [1], [11], [34], or in the context of the control of several linear processes assuming that only one process can be controlled at a given time [6], [14]. These constraints can be modeled by a switching input σ_k determining, for example, if a packet transmission occurs (and/or which communication node is allowed to transmit), or which process is being controlled at discrete-time steps $k\tau$, $k \in \mathbb{N}_0$, where τ is a sampling period. This typically requires also a parameterization of the feasible control inputs between discrete time steps such as a standard hold device $u(t) = u_k$, $t \in [k\tau, (k+1)\tau)$. The control input $u_C(t)$ is then constrained to a set $\mathcal{C}(u, \sigma)$, where $u = (u_0, u_1, \dots)$ and $\sigma = (\sigma_0, \sigma_1, \dots)$ and the continuous-time model and objective function can be discretized to take the form (1) and (2). Then we can use

$$\hat{J}(x) = x^{\intercal} P_C x \tag{25}$$

as the approximation of the cost-to-go, which satisfies (22), and derive a policy taking the form (10).

Inspired by the h-step lookahead horizon rollout policies we can define other optimistic policies resulting from the following cost approximation: $\hat{J}(x) = \min_{Y\in\mathcal{Y}} x^{\intercal} Y x$, where $\mathcal{Y} = \{F_{i_1}(F_{i_2}(\dots F_{i_N-1}(P_C))) | i_\ell \in \mathcal{M}\}$ is defined in a similar fashion as in (17). Moreover, we can define lifted policies with the cost approximation (25). One can check that both of these variants are also optimistic policies in the sense that they also satisfy (22).

III. Performance Analysis

The evolution of the state of the switched linear system (1) with policy μ taking the form (4), (10) is described by

$$x_{k+1} = \Phi_{\iota(x_k)} x_k, \tag{26}$$

where

$$\Phi_j := A_{\pi(j)} + B_{\pi(j)}K_j, \quad j \in \mathcal{P}.$$

We say that (26) is exponentially stable if there exist constants $c > 0$ and $0 \le \alpha < 1$ such that $\|x_k\| \le c\alpha^k \|x_0\|$ for every $k \in \mathbb{N}_0$. In such a case we say that the policy (4), (10) (for a given choice of $\hat{J}$) is exponentially stabilizing, or simply stabilizing for brevity. Conditions to test exponential stability will be discussed shortly.

Let $J_\mu(x_0)$ denote the cost (2) for policy μ when (4), (10) is applied to (1) initialized at x_0. Our first result relates this cost to $\hat{J}(x_0)$ through the function $\bar{J}_\mu(x)$.

Theorem 1: Let $\{x_k\}_{k\in\mathbb{N}_0}$ denote the solution to (1) for an initial state x_0 and for a stabilizing μ taking the form (4), (10). Then

$$J_\mu(x_0) = \hat{J}(x_0) + \sum_{k=0}^{\infty}(\bar{J}_\mu(x_k) - \hat{J}(x_k)). \tag{27}$$

■

We can interpret each term in the summation (27) as a gain if

$$\hat{J}(x_k) - \bar{J}_\mu(x_k) \ge 0, \tag{28}$$

or as a loss if

$$\bar{J}_\mu(x_k) - \hat{J}(x_k) \ge 0. \tag{29}$$

By convention the terms gains and losses will always refer to positive quantities. Computing the gains (28) and losses (29) along the trajectories of the SLS controlled by μ provides the cost difference between the estimate $\hat{J}$ and the true cost J_μ of policy μ. For rollout policies for which (15) holds, we have that (28) holds for every x_k, and this will be a crucial fact to analyze the stability and performance of these policies. For the optimistic policies discussed before, resulting from the discretization of a continuous-time model, one can show that (29) holds for every x_k. However, this condition is not crucial to analyze optimistic policies.

The next result provides upper and lower bounds on the cost difference between $J_\mu(x_0)$ and $\hat{J}(x_0)$.

Theorem 2: Suppose that μ is stabilizing. If there exist a function $J_U : \mathbb{R}^n \to \mathbb{R}$, and a nonnegative scalar $\lambda < 1$ such that, for every $x \in \mathbb{R}^n$,

$$\bar{J}_\mu(x) \le \hat{J}(x) + J_U(x), \tag{30}$$

and, for every $x \in \mathbb{R}^n$,

$$J_U(\Phi_{\iota(x)}x) \le \lambda J_U(x), \tag{31}$$

then, for every $x_0 \in \mathbb{R}^n$,

$$J_\mu(x_0) \le \hat{J}(x_0) + \frac{1}{1-\lambda}J_U(x_0). \tag{32}$$

Moreover, if there exist a function $J_W : \mathbb{R}^n \to \mathbb{R}$ and a nonnegative scalar $\gamma < 1$ such that, for every $x \in \mathbb{R}^n$,

$$\bar{J}_\mu(x) \ge \hat{J}(x) + J_W(x), \tag{33}$$

and, for every $x \in \mathbb{R}^n$,

$$J_W(\Phi_{\iota(x)}x) \ge \gamma J_W(x), \tag{34}$$

then, for every $x_0 \in \mathbb{R}^n$,

$$J_\mu(x_0) \ge \hat{J}(x_0) + \frac{1}{1-\gamma}J_W(x_0). \tag{35}$$

■

We can use this result in three directions to analyze the performance of rollout and optimistic policies.

First, we can use the first part of Theorem 2 to lower-bound the performance gain of a rollout policy over the corresponding base policy. This is clearly interesting in the applications mentioned before (event-triggered control, mixing of fluids, etc) as it gives a guaranteed cost gain over the base policies. Moreover, since the cost of the base policy is an upper bound on the optimal cost, computing such gains provides a stricter bound on the optimal cost. A preliminary analysis can be obtained with the parameterization $J_U(x) = -\alpha\hat{J}(x)$. Picking the largest nonnegative α such that (30) holds and the largest λ smaller than one such that (31) holds, we conclude that

$$J^*(x) \le J_\mu(x) \le \left(1 - \frac{\alpha}{1-\lambda}\right)\hat{J}(x). \tag{36}$$

In this paper, we will mostly focus on the parameterization $J_U(x) = -x^\intercal Ux$. For this parameterization, the conditions (30) and (31) boil down to

$$\min_{i\in\mathcal{P}} x^\intercal P_i x \le \min_{\ell\in\mathcal{T}} x^\intercal T_\ell x - x^\intercal Ux, \tag{37}$$

and

$$x^\intercal \Phi_{\iota(x)}^\intercal U \Phi_{\iota(x)} x \ge \lambda x^\intercal Ux, \tag{38}$$

respectively, for every $x \in \mathbb{R}^n$, where

$$\{P_i | i \in \mathcal{P}\} = \{M_c | c \in \mathcal{C}\}, \quad \{T_\ell | \ell \in \mathcal{T}\} = \{M_b | b \in \mathcal{B}\}, \tag{39}$$

and the cost *gain* of a rollout policy (10) over the cost (2) of the base policy (13) in (32) is then (at least) given by

$$\frac{1}{1-\lambda}x_0^\intercal Ux_0. \tag{40}$$

In Section III-A we shall discuss the use of these condition for the different variants of rollout policies, and we shall present shortly numerical methods to test (37), (38) in terms of linear matrix inequalities (LMIs). Other parameterizations for J_U (and J_W considered in the sequel) can be selected. For example if $J_U(x) = -\min_{r\in\mathcal{U}} x^\intercal U_r x$ for some set of positive semidefinite matrices $\mathcal{U}$, we can derive conditions to test (30) and (31) in terms of bilinear matrix inequalities. However, for the sake of brevity we do not pursue this here.

Second, we can use the second part of Theorem 2 to assert how far is the performance of the base policy from that of a corresponding rollout policy. For example, in the context of ETC [4] this is useful to guarantee that a periodic control strategy (typically easier to implement) performs already well enough with respect to a rollout (even-triggered) policy, which in that case might be dispensable. Considering again the simple parameterization $J_W(x) = -\alpha\hat{J}(x)$ we conclude that

$$\hat{J}(x) \le J_\mu(x)/(1 - \frac{\alpha}{1-\gamma}),$$

where α is the smallest nonnegative constant which satisfies (33) and γ is the smallest nonnegative value which satisfies (34). Picking $J_W(x) = -x^\intercal W x$ for a positive semidefinite matrix W, (33) and (34) boil down to asserting if for every $x \in \mathbb{R}^n$,

$$\min_{i\in\mathcal{P}} x^\intercal P_i x \geq \min_{\ell\in\mathcal{T}} x^\intercal T_\ell x - x^\intercal W x, \tag{41}$$

and

$$x^\intercal \Phi_{\iota(x)}^\intercal W \Phi_{\iota(x)} x \leq \gamma x^\intercal W x. \tag{42}$$

We are now interested in *minimizing* the upper bound on the cost difference $\hat{J}(x_0) - J_\mu(x_0)$ between the rollout and base policies, which taking into account (35) is given by

$$\frac{1}{1-\gamma} x_0^\intercal W x_0. \tag{43}$$

We shall further discuss this case in Section III-B.

Third, for optimistic policies we can use the first part of Theorem 2 to upper bound the distance $J_\mu(x) - \hat{J}(x)$, which is also a bound on the distance of the cost of μ from the optimal since $J_\mu(x) - J^*(x) \leq J_\mu(x) - \hat{J}(x)$ for every x. To this effect, if we pick the simple parameterization $J_U(x) = \alpha\hat{J}(x)$ we obtain that

$$J_\mu(x) \leq (1 + \frac{\alpha}{1-\lambda})\hat{J}(x) \overset{(22)}{\leq} (1 + \frac{\alpha}{1-\lambda})J^*(x),$$

where α is the smallest nonnegative constant that satisfies (30), and λ is the smallest nonnegative constant that satisfies (31). Considering the parameterization $J_U(x) = x^\intercal U x$ for a positive definite U, we can write the conditions (30) and (31) as

$$\min_{i\in\mathcal{P}} x^\intercal P_i x \leq \min_{\ell\in\mathcal{T}} x^\intercal T_\ell x + x^\intercal U x, \tag{44}$$

and

$$x^\intercal \Phi_{\iota(x)}^\intercal U \Phi_{\iota(x)} x \leq \lambda x^\intercal U x, \tag{45}$$

respectively, for every $x \in \mathbb{R}^n$, and the cost *loss* of an optimistic policy (10) over the optimal cost (2) is bounded by

$$\frac{1}{1-\lambda} x_0^\intercal U x_0.$$

This case will be further discussed in Section III-C. We can also use the second part of Theorem 2 to lower bound the difference between $\hat{J}(x)$ and $J_\mu(x)$. However, this does not correspond to a lower bound on the difference between J^* and J_μ, and therefore we do not elaborate on this.

The premise for these analyses is that μ is stabilizing. As we show next, for rollout policies, stability can be guaranteed under mild assumptions. Stability of optimistic policies will be discussed in Section III-D.

Theorem 3: Suppose that there exists a positive constant d such that $g(x, u, i) \geq d\|x\|^2$ for every $x \in \mathbb{R}^n$, $u \in \mathbb{R}^{n_u}$, $i \in \mathcal{M}$. Then, a rollout policy (4), (10) resulting from a stabilizing base policy characterized by $\hat{J}$, described by (13) and such that (15) holds, is stabilizing.

The following theorem allows to test the conditions of Theorem 1 for quadratic parameterizations of J_U and J_W in terms of LMIs.

Theorem 4: The following statements hold.

1) For a given $\lambda > 0$, if there exist nonnegative $\alpha_{\ell i}$, $i \in \mathcal{P}$, with $\sum_{i\in\mathcal{P}} \alpha_{\ell i} = 1$ for each $\ell \in \mathcal{T}$, nonnegative β_{ji}, $j, i \in \mathcal{P}$, and a positive semidefinite U such that

$$\sum_{i\in\mathcal{P}} \alpha_{\ell i} P_i \leq T_\ell - U, \quad \text{for everys } \ell \in \mathcal{T}, \tag{46}$$

and

$$\Phi_i^\intercal U \Phi_i + \sum_{j\in\mathcal{P}} \beta_{ji}(P_i - P_j) \geq \lambda U, \quad \text{for every } i \in \mathcal{P}, \tag{47}$$

then (37) and (38) hold.

2) For a given $\gamma > 0$, if there exist nonnegative $\alpha_{\ell i}$, with $\sum_{\ell\in\mathcal{T}} \alpha_{\ell i} = 1$, for each $i \in \mathcal{P}$, nonnegative β_{ji}, $j, i \in \mathcal{P}$, and a positive semidefinite W such that

$$P_i \geq \sum_{\ell=1}^{n_T} \alpha_{\ell i} T_\ell - W, \quad \text{for every } i \in \mathcal{P},$$

and

$$\Phi_i^\intercal W \Phi_i + \sum_{j\in\mathcal{P}} \beta_{ji}(P_j - P_i) \leq \gamma W, \quad \text{for every } i \in \mathcal{P},$$

then (41) and (42) hold.

3) For a given $\lambda > 0$, if there exist nonnegative scalars $\alpha_{\ell i}$, $i \in \mathcal{P}$, with $\sum_{i\in\mathcal{P}} \alpha_{\ell i} = 1$, for each $\ell \in \mathcal{T}$, nonnegative scalars β_{ji}, $j, i \in \mathcal{P}$, and a positive semidefinite U such that

$$\sum_{i\in\mathcal{P}} \alpha_{\ell i} P_i \leq T_\ell + U, \quad \text{for every } \ell \in \mathcal{T}, \tag{48}$$

and

$$\Phi_i^\intercal U \Phi_i + \sum_{j\in\mathcal{P}} \beta_{ji}(P_j - P_i) \leq \lambda U, \quad \text{for every } i \in \mathcal{P}, \tag{49}$$

then (44) and (45) hold.

■

A. Lower-Bounding the Gain of Rollout Policies

We start by considering one-step lookahead policies. Note that, for such a case, (38) is always met for $\lambda = 0$ as $U \geq 0$, in which case (32) is a known bound to compare base and rollout policies, see [18, p. 338]. Condition (38) imposes that the gain (28) at time $k+1$ must be always at least λ times the gain at time k for every time k and state x_k. In case $\lambda > 0$, this is only possible if the gain $y^\intercal U y$, $y = \Phi_{\iota(x)} x$, is never zero for any nonzero state x, which typically amounts to requiring $U > 0$ in (37). In turn, if (37) is achieved with $U > 0$ then there exists a strictly positive gain for every nonzero state x, which is in general too much to expect from the one-step lookahead rollout policy. In fact, this would imply that for every nonzero x the choice in (11) for (39), would correspond to a sequence in $\mathcal{C}$ not contained in $\mathcal{B}$.

On the other hand, if we consider lifted policies the set $\mathcal{C}$, described by (14) with $\mathcal{M}$ replaced by the much larger set $\mathcal{M}^h$ will generally become sufficiently rich as h increases so that this latter condition can be met. For example, if

$m = 2$ and $\mathcal{B} = \{b\}$ then $\mathcal{C} = \{(1,b),(2,b)\}$ for a one-step improvement policy, but for a lifted policy with $h = 2$ we have $\mathcal{C} = \{(1,1,b),(2,1,b),(1,2,b),(2,2,b)\}$. By considering lifted policies we mean that we analyze conditions (37), (38) for the lifted switched system (18)–(20), i.e., the matrices P_i, Φ_i should be replaced by $\underline{P}_i \in \underline{\mathcal{P}}$ and

$$\underline{\Phi}_j := \underline{A}_{\underline{\pi}(j)} + \underline{B}_{\underline{\pi}(j)}\underline{K}_j, \quad j \in \underline{\mathcal{P}},$$

where $\underline{\mathcal{P}}$, $\underline{K}_j$, and $\underline{\pi}$ are defined as for (1), (2) but now for (18), (19). We will use this reference to lifted policies several times below with this meaning, but without explicitly stating it.

Note that there is a trade-off in maximizing the gain (40) by augmenting h for lifted policies. In fact, increasing h leads to larger U, in the sense that $U > \gamma I$ for larger γ, but reduces the largest λ that satisfies (38) since $\underline{x}_\ell = x_{\ell h}$ will converge to zero faster for larger h. Moreover, since condition (38) is not easy to test, we need to test it for every i (see (47)) creating many more constraints (exponentially in h) for establishing (38). To mitigate the latter issue, we propose two methods.

The *first* method is to prune the set $\mathcal{P}$, since there may exist redundant matrices in this set that never correspond to the minimum in (11), but still impose a constraint while testing (38) using or (47) below. A matrix P_j, $j \in \mathcal{P}$ can be pruned if the ellipsoid $\mathcal{E}_j := \{x | x^\mathsf{T} P_j x \leq 1\}$ is covered by the ellipsoids corresponding to the other elements in $\mathcal{P}$, i.e., $\mathcal{E}_j \subseteq \cup_{i \in \mathcal{P}\setminus\{j\}} \mathcal{E}_i$. A sufficient condition to prune P_j (cf. [13], [14]) is the existence of nonnegative scalars α_ℓ, $\ell \in \mathcal{P}$, adding up to one, such that

$$\sum_{\ell \in \mathcal{P}\setminus\{j\}} \alpha_\ell P_\ell \leq P_j. \tag{50}$$

Inspired by [13], one can add ϵI, for a small $\epsilon > 0$, to the right-hand side of (50), leading to more pruned matrices in $\mathcal{P}$, at the expense of less gain guarantees.

The *second* method, which we shall follow in the numerical example, is to compute $\ell^{(\kappa)} = \text{argmin}_{j \in \mathcal{P}} x^{(\kappa)\mathsf{T}} P_j x^{(\kappa)}$ for a representative set of $x^{(\kappa)}$, $\kappa \in 1, \ldots, K$, and consider only the switching sequence associated with $P_{\ell^{(\kappa)}}$. Such a representative set can be chosen randomly or by simulating the trajectory of (1) driven by the rollout policy for given initial conditions. Note that to guarantee stability of the rollout policy using the arguments mentioned before one should make sure that (15) still holds after pruning by either method.

Considering now rollout policies with h-step lookahead horizon, for $h > 1$, Theorem 1 allows to estimate the difference between the true cost of such a policy and a different $\hat{J}$, given by (16), which is not the cost of the base policy, denote here by J_{base}. To compare the true cost with the cost of the base policy we can write $J_\mu - J_{\text{base}} = (J_\mu - \hat{J}) + (\hat{J} - J_{\text{base}})$, where the first term can be estimated by Theorem 1. Typically, as h increases $J_{\text{base}} - \hat{J}$ increases since $\hat{J}$ becomes closer to the optimal cost and $J_\mu - \hat{J}$ becomes smaller. In particular, it becomes harder for condition (38) to be satisfied with λ different from zero.

1) Numerical Method to Maximize the Gain: To maximize (40) for a particular x_0 we can pick a dense grid of values for λ in the interval $[0,1)$ and for each λ solve the LMI problem:

Problem 1 max $x_0^\mathsf{T} U x_0$

$$\text{s.t.} \quad U > 0, \ (46), \ (47).$$

On the other hand, if we are interested in maximizing a lower bound on the gains obtained for every initial condition we can consider:

Problem 2 max ξ

$$\text{s.t.} \quad U > \xi I, \xi > 0, \ (46), \ (47).$$

The maximum lower bound on the gain is then obtained by plotting (40) as a function of λ, where U results from the solutions to these LMI problems. If these LMI problems are unfeasible even for $\lambda = 0$ then condition (46) is not satisfied, meaning that U cannot be picked (strictly) positive definite.

B. Upper-Bounding the Gain of Rollout Policies

Here (42) imposes the most stringent condition, as there may not exist a constant $\gamma < 1$ satisfying (42) whereas (41) is always met for sufficiently large W. Note, however, that for lifted policies we can pick h large enough to reduce γ satisfying (42) and achieve $\gamma < 1$, although this in general leads to a larger W that satisfies the equivalent version of (41) for the lifted problem. Hence, a trade-off is also present here.

To compare the cost of rollout policies with h-step lookahead horizon, for $h > 1$, with that of the base policy, denoted now by J_{base} we can again write $J_\mu - J_{\text{base}} = (J_\mu - \hat{J}) + (\hat{J} - J_{\text{base}})$. Theorem 1 allows to estimate the difference between $(\hat{J} - J_\mu)$, which typically increases as h increases. Numerical methods to minimize the loss function (43) can be derived using the same ideas as in Section III-A, involving a line search over the parameter γ.

C. Upper-Bounding the Distance of Optimistic Policies to the Optimal

Again we consider only a quadratic parameterization for J_U, for the sake of brevity. We now assume that $J_U = x^\mathsf{T} U x$ for some positive semidefinite U in (30)–(32). To minimize the cost loss $\frac{1}{1-\lambda} x_0^\mathsf{T} U x_0$ of policy μ with respect to $x_0^\mathsf{T} T x_0$, for a particular initial condition x_0, we can solve:

Problem 3 min $x_0^\mathsf{T} U x_0$

$$\text{s.t.} \quad U > 0, \ (48), \ (49),$$

for each fixed $\lambda \in [0,1)$. This is also an upper bound on the loss of μ with respect to the optimal policy. The tighter bound is obtained by a line search over λ. If we are interested in bounding this loss for every initial condition, we can consider

Problem 4 min ξ

$$\text{s.t.} \quad 0 < U < \xi I, \xi > 0, (48), \ (49).$$

These problems can be unfeasible since (49) might not be met. However, one can show that by considering lifted policies feasibility will always be achieved for sufficient large h, and one can also see that as h increases one can pick smaller λ to satisfy (49). Yet, as h increases, U satisfying (49) becomes larger, and thus there is (again) a trade-off in minimizing the loss.

If we consider an optimistic policy with h- step lookahead horizon, we have that $\hat{J}$ typically becomes closer to the optimal cost as h increases and therefore the difference between $\bar{J}_\mu$ and $\hat{J}$ becomes smaller. This means that typically when h increases, the cost of the (increasingly more computationally complex) optimistic policy approaches the cost of the optimal policy.

D. Stability of Optimistic Policies

From standard (converse) Lyapunov arguments, (26) is globally exponentially stable if and only if there exists a positive definite Lyapunov function V such that, for every $x \in \mathbb{R}^n$,

$$V(\Phi_{\iota(x)} x) - V(x) \le -d\|x\|^2 \qquad (51)$$

and $c_1\|x\|^2 \le V(x) \le c_2\|x\|^2$ for some positive constants d, c_1, c_2 (cf. [30]). A natural candidate for a Lyapunov function to test if an optimistic policy μ is stabilizing is $\hat{J}(x)$, especially in the mentioned cases where (1), (2) result from the discretization of a continuous-time model. That is, if $\hat{J}(x)$ is positive definite and if

$$\hat{J}(\Phi_{\iota(x)} x) - \hat{J}(x) \le -a\|x\|^2, \qquad (52)$$

for every $x \in \mathbb{R}^n$ and for some $a > 0$, then μ is stabilizing. One can conclude that a sufficient condition to test this when $n_T = 1$ and $T_1 = T$ is the existence of nonnegative β_{ij}, $i, j \in \mathcal{P}$, such that

$$\Phi_i^\mathsf{T} T \Phi_i + \sum_{j \in \mathcal{P}} \beta_{ij}(P_j - P_i) - T < 0, \quad \text{for every } i \in \mathcal{P}.$$

This follows from similar arguments from the ones used in the proof of Theorem 4. Similar conditions can be derived for the case where $n_T > 1$. Considering lifted policies using $\hat{J}(x) = x^\mathsf{T} T x$, one can show that (52) holds for large enough h, provided that the SLS is exponentially stabilizable, i.e., for every x_0 there exist a control and a switching input sequence that drives the state exponentially to zero (see [30]).

An alternative way to test stability still using (52) is provided in the next result.

Theorem 5: Suppose that there exists some positive constant η such that $g(x, u, i) \ge \eta\|x\|^2$ for every $x \in \mathbb{R}^n$, $u \in \mathbb{R}^{n_u}$ and $i \in \mathcal{M}$. If there exists $0 \le \epsilon < 1$ such that

$$\bar{J}_\mu(x) \le \hat{J}(x) + \epsilon g(x, K_{\iota(x)} x, \pi(\iota(x))) \qquad (53)$$

then (52) holds. ■

For example, in the special case where there is no control input (B_i, S_i, R_i are identically zero), we can test this condition by asserting the existence of nonnegative $\alpha_{\ell j}$ adding up to one for each $\ell \in \mathcal{T}$ and a nonnegative ϵ smaller than one such that for every $i \in \mathcal{M}$ and $\ell \in \mathcal{T}$, we have $\sum_{j \in \mathcal{P}} \alpha_{\ell j} P_j \le T_\ell + \epsilon Q_i$.

IV. Finite-Horizon Cost

In this section we discuss how to extend the proposed policies and results to the case where a finite-horizon cost

$$\sum_{k=0}^{L-1} g(x_k, u_k, \sigma_k) + g_L(x_L), \qquad (54)$$

for $g_L(x_L) := x_L^\mathsf{T} P_L x_L$ and for g as in (3), is to be minimized for (1) instead of the infinite horizon cost considered so far. The optimal policy for this case is a time-varying policy $(u_k, \sigma_k) = \mu_k(x_k)$ described by

$$\mu_k(x) \in \operatorname*{argmin}_{u \in \mathbb{R}^{n_u}, i \in \mathcal{M}} g(x, u, i) + V_{k+1}(A_i x + B_i u), \qquad (55)$$

where $V_k(x) = \min_{X \in \mathcal{X}_k} x^\mathsf{T} X x$,

$$\mathcal{X}_k = \{F_{i_k}(F_{i_{k+1}}(\ldots F_{i_{L-1}}(P_L)))| i_\ell \in \mathcal{M}\}, \qquad (56)$$

for $k \in \{0, 1, \ldots, L-1\}$ (cf. [13]). Each matrix X of the set $\mathcal{X}_k$ corresponds to a finite sequence of switching inputs $(i_k, \ldots, i_{L-1})$ and $x_0^\mathsf{T} X x_0$ is the cost (54) of applying such a sequence and associated optimal control inputs when the initial condition is x_0. The switching inputs at time k specified by the optimal policy result then from an *exhaustive search* for the sequence that leads to the minimum cost from time k to time L.

Similarly to the case of an infinite horizon cost, we can consider an approximation of the *cost-to-go* $\sum_{\kappa=k}^{L-1} g(x_\kappa, u_\kappa, \sigma_\kappa) + g_L(x_L)$ incurred after a given time step k. This approximation is denoted by $\hat{J}^k(x)$, which now depends on the time step k. We can then define suboptimal policies obtained by replacing V_{k+1} by $\hat{J}^{k+1}(x)$ in (55). For rollout policies, this approximation corresponds to the cost associated with one or several finite sequences and corresponding optimal control inputs, whereas for optimistic policies this approximation corresponds to a lower bound on the *optimal* cost-to-go $V_k(x)$. Both optimize over a lookahead horizon taking into account this approximation for the cost-to-go after the horizon. We can interpret this as a heuristic search method which replaces the exhaustive (and often computational intractable) optimal search described above. Moreover, the fact that after the horizon some information is used on the optimal cost, in terms of lower or upper bounds, to guide the search, parallels similar ideas of informed search methods, such as the A* search [29].

The cost approximation for rollout policies can be either computed *a priori*, leading to an explicit policy as we have considered so far, or computed online, involving an optimization for the switching inputs which leads to a good cost approximation. In the former case, the expression is given by $\hat{J}^k(x) := \min_{b^k \in \mathcal{B}^k} x^\mathsf{T} M_{b^k} x$, where $\mathcal{B}^k$ is a set of sequences taking values in $\{k, k+1, \ldots, L-1\}$ and $x^\mathsf{T} M_{b^k} x$ is the cost-to-go when the sequence b^k is used along with optimal control inputs. These sequences might, for example, result from an initial sequence $(b_0, b_1, \ldots, b_{L-1})$, which is truncated either by removing the first schedules leading to $(b_k, \ldots, b_{T-1})$ or the last schedules giving $(b_0, \ldots, b_{L-1-k})$. In the case of online optimization, the cost-to-go are evaluated by running (1) for the sets of switching input sequence corresponding to the base policy (and corresponding optimal control inputs when $B_i \neq 0$). This might be needed in applications, where it is not convenient or even impossible to store the matrices $\{M_{b^k} | b^k \in \mathcal{B}^k\}$. This is, for instance, the case in the context of mixing of fluids where these matrices can have over $50\,000 \times 50\,000$ entries, while running (1) is possible due to the sparsity of the matrices A_i (cf. [2]).

Similarly to Theorem 1 one can conclude that

$$J_\mu(x_0) = \hat{J}^0(x_0) + \sum_{k=0}^{L-1} (\bar{J}_\mu^k(x_k) - \hat{J}^k(x_k)), \qquad (57)$$

where $J_\mu(x_0)$ denotes the cost (54) of the policy $(\mu_0, \mu_1, \dots, \mu_{L-1})$ for an initial condition x_0 and $\bar{J}^k_\mu(x_0)$ is defined as in (7) by replacing $\hat{J}$ by $\hat{J}^{k+1}$ if $k \in \{0, 1, \dots, L-2\}$ and by replacing $\hat{J}$ by g_L if $k = L-1$. Again the difference between the cost of the proposed policy $J_\mu(x_0)$ and that of the initial approximation $\hat{J}^0(x_0)$ is the sum between gains or losses along the trajectory of the SLS. However, extending Theorem 2 would entail testing similar conditions to the ones presented there for each $k \in \{0, 1, \dots, L-1\}$, which might be impractical in general.

Instead, we can use time-invariant policies for the finite-horizon problem resulting from a time-invariant cost approximation $\hat{J}(x)$ and use our previous results to characterize the performance of such a policy. In fact, if we let $J^L_\mu(x_0)$ and $J^\infty_\mu(x_0)$ denote the finite-horizon cost (54) and infinite-horizon cost (2) obtained when a *time-invariant policy* μ, described by (4), (10), is applied to (1) initialized at x_0, we can conclude that

$$J^\infty_\mu(x_0) = J^L_\mu(x_0) - g_L(x_L) + J^\infty_\mu(x_L).$$

If the horizon is large, $g_L(x_L)$ and $J_\mu(x_L)$ are close to zero. Thus, a characterization the performance of $J^\infty_\mu(x_0)$, made possible by Theorems 1 and 2 is also an approximate characterization of the performance of $J^L_\mu(x_0)$. The use of time-invariant policies for the finite-horizon problem can also be justified by the fact that by construction the proposed policy for the finite-horizon problem μ_k, for a fixed k, converges to the proposed policy for the infinite horizon problem when $\hat{J}^{k+1}(x)$ converges to $\hat{J}(x)$. This is the case when the horizon is sufficiently large and the same periodic switching input sequence is used to compute $\hat{J}^k(x)$ and $\hat{J}(x)$.

V. Numerical Examples

In Section V-A we consider the event-triggered control of a simple two dimensional system, allowing us to illustrate the main concepts and results of the paper graphically. In Section V-B, we consider the control of two linear processes assuming that only one process can be controlled at a given time. This constraint may arise from limited resources in terms of communication (e.g., wireless settings) [17] or computation (e.g., shared processor) [6] resources. For this example, our main goal is to compare the complexity and the performance of policies proposed in this paper with previous approaches [13], [14].

A. Event-Triggered Control

Consider the following linearization model of an inverted pendulum

$$\frac{d}{dt}\begin{bmatrix}\theta(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}\dot{\theta}(t)\\ \gamma_m\theta + T_m(t)\end{bmatrix}, \tag{58}$$

where $\theta(t)$ is the displacement angle, $\dot{\theta}(t)$ is the angular velocity, and $T_m(t)$ is the torque input at time $t \in \mathbb{R}_{\geq 0}$, and γ_m is a positive constant. The actuators are connected to a controller collocated with the sensors via a communication network. The controller can sample the state sensors, providing full state measurements periodically at times $t_k := k\tau$, $k \in \mathbb{N}_0$, for some sampling period $\tau > 0$. However, to save communication resources, the controller sends control values to the actuators only at a subset of times t_k, $k \in \mathbb{N}_0$, determined by the switching input $\{\sigma_k | k \in \mathbb{N}_0\}$, with $\sigma_k = 1$ when a transmission occurs and $\sigma_k = 2$ otherwise. The control objective is to minimize the quadratic cost

$$\int_0^\infty (\theta(t)^2 + \dot{\theta}(t)^2 + r_c T_m(t)^2)dt. \tag{59}$$

Note that in the absence of a network the optimal control input would be the solution to the well-known LQR problem, given by $T_m(t) = K_C x_C(t)$, where $x_C(t) = [\theta(t)\,\dot{\theta}(t)]^\intercal$ leading to a cost $x_C(0)^\intercal P_C x_C(0)$. For the numerical values $\gamma_m = 3$ and $r_c = 0.1$ this yields $K_C = [-7.3589 \quad -4.9717]$ and

$$P_C = \begin{bmatrix}2.1671 & 0.7359\\ 0.7359 & 0.4972\end{bmatrix}.$$

At the actuators side, the control is set to zero if there is no transmission at time t_k, and is given by a linear state feedback law with gain K_C otherwise, i.e.,

$$T_m(t) = \begin{cases} 0, \text{ if } \sigma_k = 2, \\ K_C x_k, \text{ if } \sigma_k = 1, \end{cases} \quad t \in [t_k, t_{t_k+1}), \tag{60}$$

where $x_k := x_C(t_k)$, $k \in \mathbb{N}_0$. Writing the equations for x_k we obtain a SLSs (1) with two modes $m = 2$ (transmit or not transmit) and $B_i = 0$, $i \in \{1, 2\}$. After exactly discretizing (58), (59) for $\tau = 0.1$ and taking into account (60) we obtain

$$A_1 = \begin{bmatrix}0.9782 & 0.0756\\ -0.4381 & 0.5154\end{bmatrix}, \quad A_2 = \begin{bmatrix}1.0150 & 0.1005\\ 0.3015 & 1.0150\end{bmatrix},$$

and

$$Q_1 = \begin{bmatrix}0.6465 & 0.3553\\ 0.3553 & 0.3065\end{bmatrix}, \quad Q_2 = \begin{bmatrix}0.1040 & 0.0202\\ 0.0202 & 0.1013\end{bmatrix}.$$

We are interested in computing a policy for $\{\sigma_k | k \in \mathbb{N}_0\}$ that minimizes the quadratic cost subject to a transmission rate constraint to save communication resources. Here we consider that the controller can only transmit on average at half of the rate it can measure the state, i.e., $\frac{1}{2\tau}$. To achieve this we use a similar scheme to [16], [4] considering a lifted policy (in the sense of Section II-A2) that incorporates this constraint. We pick $h = 6$ and note that there are 20 scheduling options in $\{1, 2\}^6$ that satisfy the transmission constraint, e.g., (1, 2, 1, 2, 1, 2) and (2, 1, 1, 2, 2, 1). We consider the following policies: i) a policy obtained by considering the lower bound $x_0^\intercal P_C x_0$ as a cost estimate in (6), (10); ii) a rollout policy with base policy (1, 2, 1, 2, 1, 2) which we shall refer to as periodic switching; we consider also a rollout policy resulting from the base policy (2, 1, 1, 1, 2, 2).

1) Optimistic Policies: The costs obtained by simulation and by the bounds obtained by solving Problem 3 are summarized in the next table for three initial conditions

$$x_0^{(1)} = \begin{bmatrix}1\\ 0\end{bmatrix}, \quad x_0^{(2)} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}, \quad x_0^{(3)} = \frac{2}{\sqrt{13}}\begin{bmatrix}1\\ -\frac{3}{2}\end{bmatrix}.$$

We also show the values obtained by an optimal continuous-time policy and a periodic switching policy obtained by repeating (1, 2, 1, 2, 1, 2) for comparison.

Ini. Cond.	J_μ	Problem 3	$x_0^\intercal P_C x_0$	Periodic
$x_0^{(1)}$	3.1836	3.2015	2.1671	3.6032
$x_0^{(2)}$	3.0697	3.1822	2.0680	3.6618
$x_0^{(3)}$	0.3625	0.3925	0.3317	0.3720

To obtain these values we restricted the options in $\{1, 2\}^6$ choosing a representative set of states and checking which scheduling options correspond to the choice in (20). With this method we restricted the scheduling options to (1, 2, 1, 2, 1, 2), (2, 1, 1, 1, 2, 2), (1, 1, 1, 2, 2, 2), (2, 1, 1, 2, 1, 2), (1, 1, 2, 1, 2, 2). Note that the guaranteed bounds are reasonably tight to the values J_μ obtained by simulation.

Considering Problem 4, we obtain $\lambda = 0.53$ and

$$U = \begin{bmatrix} 0.5151 & 0.1617 \\ 0.1617 & 0.2552 \end{bmatrix}$$

which allows us to conclude that

$$J_\mu \le x_0^\intercal P_C x_0 + \frac{1}{1-\lambda} x_0^\intercal U x_0 = x_0^\intercal \begin{bmatrix} 3.2630 & 1.0799 \\ 1.0799 & 1.0401 \end{bmatrix} x_0.$$

*2) **Rollout:*** The costs obtained by a rollout policy considering the base policy obtained by periodically repeating (1, 2, 1, 2, 1, 2) for the three initial states are summarized in the next table, where we repeat the values obtained with the periodic policy for comparison.

Ini. Cond.	J_μ(rollout)	Periodic(base)
$x_0^{(1)}$	3.1986	3.6032
$x_0^{(2)}$	3.0734	3.6618
$x_0^{(3)}$	0.3519	0.3720

Note that the values obtained with the rollout policy are considerably better than with the base policy. Yet, Problems 1 and 2 are unfeasible, preventing us from providing a guaranteed bound on such gain other than the trivial bound, as previously discussed. This illustrates that the bounds can be sometimes conservative, especially if the base policy is already well-performing as in this case.

Motivated by this fact we consider a different base policy associated with the schedules (2, 1, 1, 1, 2, 2), which do not perform as well as the previous base policy. The new values for the cost are summarized next.

Ini. Cond.	J_μ(rollout)	Problem 1	Periodic(base)
$x_0^{(1)}$	3.2160	3.9896	4.1450
$x_0^{(2)}$	3.0901	4.1427	4.3627
$x_0^{(3)}$	0.3626	0.3742	0.3789

We considered the following schedules to obtain the bound with Problems 1 and 2: (2, 1, 1, 2, 1, 2), (1, 2, 1, 2, 1, 2), (2, 1, 1, 1, 2, 2), (1, 1, 2, 1, 2, 2), (1, 1, 1, 2, 2, 2). These schedules were again obtained by choosing a representative set of states and checking which scheduling options correspond to the choice in (20). The values obtained by solving Problem 1 are reasonably tight to the values obtained via simulation. Solving Problem 4, we obtain $\xi = 0.045$ and

$$W = \begin{bmatrix} 0.0105 & 0.0041 \\ 0.0041 & 0.0057 \end{bmatrix}.$$

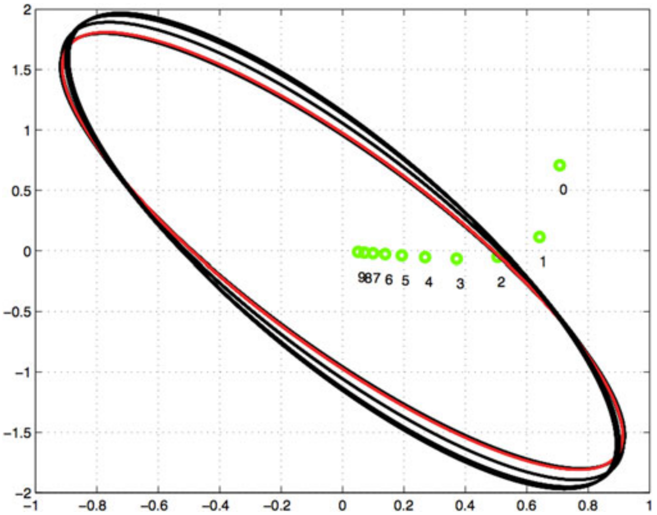


Fig. 1. Illustration of the set $\mathcal{P}$ and of the trajectory associated with $x_0^{(2)}$. Time $k = 0, 1, \ldots, 9$ is indicated next to each point of the trajectory.

In Fig. 1 we plot the ellipsoids $\{x | x^\intercal P_i x = 1\}$ where $P_i \in \mathcal{P}$ are 5 matrices associated with each of the set of schedules mentioned above. In grey (or red in the colored version of the present paper) we show the ellipsoid associated with the base policy. We can see that it is covered by the remaining ellipsoids, meaning that in fact it is never picked in (20) (and thus we can guarantee a strict gain). We also show a trajectory of the rollout policy in Fig. 1. Note that by Theorem 1 we know that the overall gain is the sum of each of the gains (28) along the trajectory. We can obtain these gains with the help of the plotted ellipsoids yielding for the first seven time steps the following:

k	0	1	2	3	4	5	6
Gain (28)	0.803	0.262	0.106	0.050	0.025	0.013	0.007

Adding up these gains we obtain 1.266, which is a value already close to the overall gain associated with $x_0^{(2)}$, which is $4.3627 - 3.0901 = 1.2726$.

B. Control of Several Processes With Constrained Actuation

In this subsection we consider the simultaneous control of m processes assuming that only one process can be controlled at a given time. We assume that each of the processes can be described by a linear model taking the form (24) and for each of the processes we wish to minimize an independent cost function taking the form (23). The overall objective is to minimize the sum of these costs. The actuation value for each process $j \in \{1, \ldots, m\}$ is hold constant between times $t_k = k\tau$, for a period τ, and is only updated if the actual values for the other processes are not. Formally, we have that in the interval $t \in [t_k, t_{k+1})$,

$$u_C^j(t) = \begin{cases} u_k^j \text{ if } \sigma_k = j \\ u_C^j(t_k^-) \text{ otherwise,} \end{cases}$$

where $\sigma_k \in \{1, 2, \ldots, m\}$ is the switching input, which equals j if the actuation value of process j is updated, $u_C^j(t_k^-)$ denotes the actuation value immediately before time t_k, and u_k^j denotes the possible actuation update value of the process j at time t_k, taking any other value otherwise. If we define the state of process j at time t_k as $x_k^j := [x_C(t_k)^\intercal \;\; u_k^{j\intercal}]^\intercal$, we can conclude that it evolves according to (1), where

$$A_j^j = \begin{bmatrix} A_d^j & 0 \\ 0 & 0 \end{bmatrix}, A_i^j = \begin{bmatrix} A_d^j & B_d^j \\ 0 & I \end{bmatrix},$$

$$B_j^j = \begin{bmatrix} B_d^j \\ I \end{bmatrix}, B_i^j = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

for $i \neq j$ and $A_d^j = e^{A_C^j \tau}$, $B_d^j = \int_0^\tau e^{A_C^j s} B_C^j ds$. Moreover, the cost for each of the processes takes the form (2) where

$$Q_j^j := \begin{bmatrix} \bar{Q} & 0 \\ 0 & 0 \end{bmatrix}, Q_i^j := \begin{bmatrix} \bar{Q} & \bar{S} \\ \bar{S}^\intercal & \bar{R} \end{bmatrix}, \; S_j^j := \begin{bmatrix} \bar{S} \\ 0 \end{bmatrix}, \; S_i^j := \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$R_j^j := \bar{R}$, $R_i^j := 0$, for $i \neq j$, and

$$\begin{bmatrix} \bar{Q} & \bar{S} \\ \bar{S}^\intercal & \bar{R} \end{bmatrix} := \int_0^\tau e^{\begin{bmatrix} A_C & B_C \\ 0 & 0 \end{bmatrix}^\intercal s} \begin{bmatrix} Q_C & 0 \\ 0 & R_C \end{bmatrix} e^{\begin{bmatrix} A_C & B_C \\ 0 & 0 \end{bmatrix} s} ds. \tag{61}$$

We can then combine these models and obtain an augmented model taking the form (1) where $x_k = [x_k^{1\intercal} \ldots x_k^{m\intercal}]^\intercal$ with $A_j = \text{blkdiag}(A_j^1, \ldots, A_j^m)$, $B_j = \text{blkdiag}(B_j^1, B_j^2, \ldots, B_j^m)$, where $\text{blkdiag}(M_1, \ldots, M_m)$ denotes a block diagonal matrix with blocks M_i, and an augmented cost taking the form (2) where the cost matrices can also be easily obtained.

We consider that the models and costs associated with each process are identical and given by (58) and (59), respectively, for $\gamma_m = 3$ and $r_c = 0.1$. The sampling period is set to $\tau = 0.1$.

We compare four policies: i) h-step lookahead rollout policies with a base policy obtained by taking the minimum of m periodic base policies, each corresponding to actuating the processes in a round-robin fashion starting at actuator $i \in \{1, 2, \ldots, m\}$ and associated with the periodic sequence $b_k = 1 + ((i + k - 1) \bmod m)$; ii) h-step lookahead optimistic policies with $\hat{J}(x_0) = \underline{J}(x_0)$ for

$$\underline{J}(x) = x^\intercal(\text{blkdiag}(P_C, P_C, \ldots, P_C))x$$

where P_C is such that $x_0^{j\intercal} P_C x_0^j$ is the optimal cost for each process j which would be obtained by using continuous-time control (in the absence of actuator restrictions), i.e., all the actuators can be used/updated simultaneously; iii) policy (5) for a given N resulting from the dynamic programming algorithm described in Section II; iv) a relaxed method combining ideas of [13] and [14]. The starting point of this relaxed method is the dynamic programming algorithm described in Section II. However, at each step of the algorithm the complexity of the value functions is reduced according to a relaxation method. This method boils down to considering $V_k^\epsilon(x) := \min_{H \in \mathcal{H}_k} x^\intercal H x$ for

$$\mathcal{H}_{k+1} = \mathcal{C}_\epsilon(\mathcal{F}(\mathcal{H}_k)), \;\; \mathcal{H}_0 = \{0\}, \;\; k \in \{0, 1, \ldots, N\},$$

and taking the policy (5) with V_{N+1} replaced by V_{N+1}^ϵ, where $\mathcal{F}(\mathcal{H}) := \{F_i(H) | H \in \mathcal{H}, i \in \mathcal{M}\}$ and $\mathcal{C}_\epsilon$ is a pruning operator that selects matrices from its input set based on the following rule: matrix $P_j \in \mathcal{H}$ is pruned if there exist nonnegative scalars α_ℓ, adding up to one, such that

$$\sum_{P_\ell \in \mathcal{H} \setminus \{P_j\}} \alpha_\ell P_\ell \leq P_j + \epsilon I. \tag{62}$$

This pruning method coincides with the one in [13], whereas in [14] the relaxation term ϵI is instead proportional to the stage cost. In [13] a stopping criterion for the algorithm (choice of N) is proposed, which assures that the resulting policy (5) with $V_N(x)$ replaced by $V_N^\epsilon(x)$ is stabilizing for sufficiently small ϵ. However, motivated by the difficulties in stopping the algorithm with this stopping criterion, we run the iterations (62) for a given relaxation parameter ϵ for a large N, in a similar fashion to [14]. Besides, as proposed in [14] at each step of the algorithm the matrices $H \in \mathcal{H}_k$ are ordered in increasing order of trace(H) to speed up the algorithm.

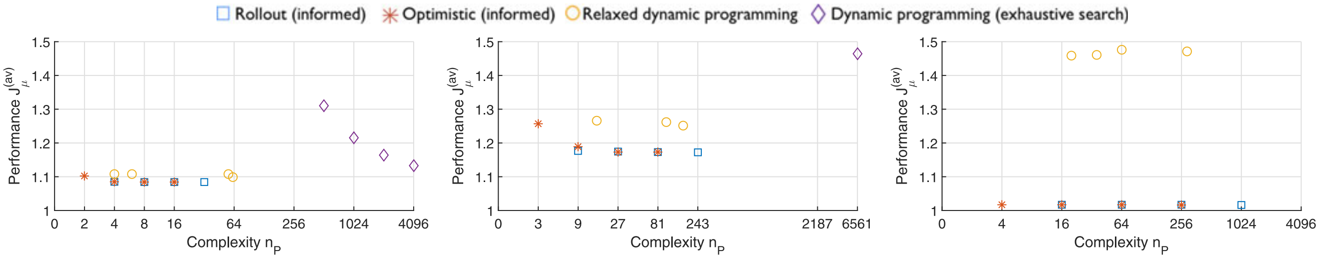
We compare the performance versus complexity of these policies by plotting an average performance over several initial conditions

$$\text{Average performance: } J_\mu^{\text{av}} = \sum_l \frac{J_\mu(x_0^{(l)})}{\underline{J}(x_0^{(l)})}$$

normalized with respect to the optimal performance in the absence of actuator constraints $\underline{J}$. The set of initial conditions results from all the 6^m combinations obtained by taking the initial condition of each of the m processes from the set $(\theta(0), \dot{\theta}(0)) \in \{\frac{\pi}{24}, \frac{\pi}{3}\} \times \{-\frac{1}{2}, 0, \frac{1}{2}\}$. The complexity of the policy is measured by the number of matrices in the set $\mathcal{P}$

$$\text{Complexity: } n_P.$$

The results are plotted in Fig. 2 for $m \in \{2, 3, 4\}$, where the parameters corresponding to each policy are also indicated. Both rollout and optimistic policies achieve already a very good performance for small values of h corresponding to very low complexity policies. Finding a stabilizing policy using the dynamic programming algorithm is still possible for $m \in \{2, 3\}$, but, e.g., for $m = 2$ even with $n_P = 4096$, corresponding to $N = 11$ iterations of the dynamic programming algorithm, the performance is far from the other policies. The relaxed dynamic programming when $m = 2$ can still find a low complexity stabilizing policy with a performance close to that of the informed policies if we set the relaxation parameter large ($\epsilon > 0.1$). However, if we try to approach the optimal performance by setting the relaxation parameter to smaller levels, the complexity of V^ϵ becomes larger and it becomes harder (and at some point even impractical) to test (62). This is what we also see for $m \in \{3, 4\}$: it is possible to find a stabilizing policy with a large relaxation parameter and with a performance far from optimal, but as we



(a) (b) (c)

Fig. 2. Normalized performances averaged over several initial conditions. For rollout policies and optimistic policies we used $h \in \{1, 2, 3, 4\}$ for every m; each value of h corresponds to stars and arrows in the plot, respectively, from left to right. The number iterations used for dynamic programming were $N \in \{7, 8, 9, 10\}$ for $m = 2$ and $N \in \{6\}$ for $m = 2$ corresponding to the depicted diamonds from left to right. The parameters used for relaxed dynamic programming corresponding to the circles from left to right were: for $m = 2$, $(\epsilon, N) \in \{(0.05, 30), (0.025, 30), (0.015, 30), (0.01, 30)\}$; for $m = 3$, $(\epsilon, N) \in \{(0.1, 30), (0.05, 20), (0.04, 14)\}$; for $m = 4$, $(\epsilon, N) \in \{(0.5, 15), (0.25, 15), (0.1, 10), (0.1, 15)\}$. (a) $m = 2$, (b) $m = 3$, (c) $m = 4$.

try to reduce this relaxation parameter, it is computationally impractical to run relaxed dynamic programming. As a consequence, it becomes harder to approximate the cost of informed policies. This is also the trend for larger values of m. It is clear that informed policies allow considerably better performance with less complexity than the dynamic programming and the relaxed dynamic programming methods.

VI. Conclusions and Discussion

In this paper we have proposed and analyzed a class of suboptimal policies for the linear quadratic regulation of switched systems. These policies are especially intended for problems where some information on the optimal cost is available, such as a lower bound or the knowledge of a stabilizing base policy. We have provided examples of applications (resource-aware control, mixing of fluids, etc.) where such information is available and we have concluded via numerical examples that for such applications informed policies lead in general to policies with a lower complexity or/and better performance than the ones proposed in previous papers.

For a general application of SLS (e.g. [2]–[10]), finding such information or an approximation of the optimal cost is the crucial step to use the policies and results of the present paper. Stabilizing rollout policies are typically easier to find, since they only need the existence of a stabilizing base policy associated with a fixed sequence of schedules. For each initial condition such sequence must naturally exist if the system is (exponentially) stabilizable. The crucial point is to assert the existence of a single sequence for *every* initial condition. Easy to test conditions to assert the existence of a periodic stabilizing policy can be found in [35] for the case where $B_i = 0$. However, such a periodic base policy might not exist, even if the system is stabilizable, as shown in [35] by an example. This can be viewed as a limitation of rollout policies, although in most practical applications a periodic base policy is available. In turn, optimistic policies rely on a lower bound on the optimal cost, which always exists since the optimal cost is nonnegative. The crucial point for these policies is to find a lower bound such that the resulting policy is stabilizing. As discussed above, this can always be guaranteed for h-lifted policies, but might require a large h leading to computationally complex policies.

Appendix
Proof of Theorem 1

We note that for $\{x_k\}_{k\in\mathbb{N}_0}$ it holds that

$$\bar{J}_\mu(x_k) = g(x_k, \bar{u}(x_k), \bar{\sigma}(x_k)) + \hat{J}(x_{k+1}).$$

for every $k \in \mathbb{N}_0$, where $(\bar{u}(x), \bar{\sigma}(x)) := (K_{\iota(x)}x, \pi(\iota(x)))$. By adding up from $k = 0$ to $k = N$ we obtain

$$\sum_{k=0}^{N} g(x_k, \bar{u}(x_k), \bar{\sigma}(x_k))$$
$$= \hat{J}(x_0) + \sum_{k=0}^{N} (\bar{J}_\mu(x_k) - \hat{J}(x_k)) - \hat{J}(x_{N+1}).$$

Letting $N \to \infty$ and noting that, under the assumption that μ is stabilizing, the state exponentially converges to zero (and so does $\hat{J}(x_{N+1})$), we obtain (27).

Proof of Theorem 2

To prove (32) we note that

$$J_\mu(x_0) - \hat{J}(x_0) = \sum_{k=0}^{\infty} (\bar{J}_\mu(x_k) - \hat{J}(x_k))$$
$$\leq \sum_{k=0}^{\infty} J_U(x_k) \leq \sum_{k=0}^{\infty} \lambda^k J_U(x_0) = \frac{1}{1-\lambda} J_U(x_0),$$

where in the first equality we used (27), in the first inequality we used (30), and in the second inequality we used (31). The proof of (35) is similar and is therefore omitted.

Proof of Theorem 3

From standard (converse) Lyapunov arguments, (26) is exponentially stable if and only if there exists a positive definite Lyapunov function V such that

$$V(\Phi_{\iota(x)}x) - V(x) \leq -a\|x\|^2 \tag{63}$$

and $c_1\|x\|^2 \leq V(x) \leq c_2\|x\|^2$ for some positive constants a, c_1, c_2 (cf. [30]). If (15) holds then (28) holds for every x_k. From this we conclude that for every $x \in \mathbb{R}^n$,

$$g(x, K_{\iota(x)}x, \pi(\iota(x))) + \hat{J}(\Phi_{\iota(x)}x) \leq \hat{J}(x)$$

Then $\hat{J}$ is a Lyapunov function which is bounded and positive definite (since g is positive definite) and satisfies (63) for some $a = d$, where d is as in the hypothesis of the theorem. Therefore, μ is stabilizing.

Proof of Theorem 4

We prove only (i) since the proofs of (ii) and (iii) are similar. It is clear that (37) holds if

$$\min_{i\in\mathcal{P}} x^\intercal P_i x \leq x^\intercal (T_\ell - U)x, \quad \text{for every } \ell \in \mathcal{T}.$$

and since $\min_{i\in\mathcal{P}} x^\intercal P_i x \leq \sum_{i\in\mathcal{P}} \alpha_{\ell i} x^\intercal P_i x$, where the $\alpha_{\ell i}$ are as described in the theorem statement, we conclude that (37) holds if (46) holds. It is also clear that (38) holds if for every $i \in \mathcal{P}$,

$$x^\intercal \Phi_i^\intercal U \Phi_i x \geq \lambda x^\intercal U x$$

holds when x is such that

$$x^\intercal (P_i - P_j)x \leq 0, \text{ for every } j \in \mathcal{P} \setminus \{i\}$$

Then (47) follows from a straightforward application of the S-procedure [36].

Proof of Theorem 5

Using (7) in (53) we obtain

$$\begin{aligned}\hat{J}(\Phi_{\iota(x)}x) &\leq \hat{J}(x) + (\epsilon - 1)g(x, K_{\iota(x)}x, \pi(\iota(x))) \\ &\leq \hat{J}(x) - \eta(1 - \epsilon)\|x\|^2.\end{aligned}$$

Then (52) holds with $a = \eta(1 - \epsilon)$.

References

[1] D. Antunes and W. Heemels, "Performance analysis of a class of linear quadratic regulators for switched linear systems," in *Proc. 2014 IEEE 53rd Annu. Conf. Decision and Control (CDC)*, Dec. 2014, pp. 5475–5480.

[2] V. Dolk, "A switching feedback approach to optimize mixing of fluids," Master's thesis, Eindhoven Univ. Technol. (TU/e), 2013.

[3] S. Mariethoz, S. Almer, M. Baja, A. Beccuti, D. Patino, A. Wernrud, J. Buisson, H. Cormerais, T. Geyer, H. Fujioka, U. Jonsson, C.-Y. Kao, M. Morari, G. Papafotiou, A. Rantzer, and P. Riedinger, "Comparison of hybrid control techniques for buck and boost DC-DC converters," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 5, pp. 1126–1145, Sep. 2010.

[4] D. Antunes and W. Heemels, "Rollout event-triggered control: Beyond periodic control performance," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3296–3311, Dec. 2014.

[5] W. Zhang, J. Hu, and J. Lian, "Quadratic optimal control of switched linear stochastic systems," *Syst. Control Lett.*, vol. 59, no. 11, pp. 736–744, 2010.

[6] A. Cervin and P. Alriksson, "Optimal on-line scheduling of multiple control tasks: A case study," in *Proc. 18th Euromicro Conf. Real-Time Systems, 2006* 2006, pp. 10–150.

[7] E. Hernandez-Vargas, P. Colaneri, R. Middleton, and F. Blanchini, "Discrete-time control for switched positive systems with application to mitigating viral escape," *Int. J. Robust Nonlinear Control*, vol. 21, no. 10, pp. 1093–1111, 2011.

[8] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[9] F. Blanchini, D. Casagrande, P. Gardonio, and S. Miani, "Constant and switching gains in semi-active damping of vibrating structures," *Int. J. Control*, vol. 85, no. 12, pp. 1886–1897, 2012.

[10] R. Alur, A. D'Innocenzo, K. Johansson, G. Pappas, and G. Weiss, "Compositional modeling and analysis of multi-hop control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 10, pp. 2345–2357, Oct. 2011.

[11] D. Dacic and D. Nesic, "Observer design for wired linear networked control systems using matrix inequalities," *Automatica*, vol. 44, no. 11, pp. 2840–2848, 2008.

[12] F. L. Lewis, *Applied Optimal Control and Estimation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[13] W. Zhang, J. Hu, and A. Abate, "Infinite-horizon switched LQR problems in discrete time: A suboptimal algorithm with performance analysis," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1815–1821, Jul. 2012.

[14] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1249–1260, Aug. 2006.

[15] D. Gorges, M. Izak, and S. Liu, "Optimal control and scheduling of switched systems," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 135–140, Jan. 2011.

[16] D. Antunes, W. P. M. H. Heemels, and P. Tabuada, "Dynamic programming formulation of periodic event-triggered control: Performance guarantees and co-design," in *Proc. IEEE 51st Annu. Conf. Decision and Control (CDC)*, Dec. 2012, pp. 7212–7217.

[17] D. Antunes, W. P. M. H. Heemels, J. Hespanha, and C. Silvestre, "Scheduling measurements and controls over networks—Part I: Rollout strategies for protocol design," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 2036–2041.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 2005.

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

[20] G. S. Deaecto, M. Souza, and J. C. Geromel, "Discrete-time switched linear systems state feedback design with application to networked control," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 877–881, Mar. 2015.

[21] J. Geromel, G. Deaecto, and J. Daafouz, "Suboptimal switching control consistency analysis for switched linear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 7, pp. 1857–1861, Jul. 2013.

[22] G. S. Deaecto, A. R. Fioravanti, and J. C. Geromel, "Suboptimal switching control consistency analysis for discrete-time switched linear systems," *Eur. J. Control*, vol. 19, no. 3, pp. 214–219, 2013.

[23] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[24] B. Picasso, D. D. Vito, R. Scattolini, and P. Colaneri, "An MPC approach to the design of two-layer hierarchical control systems," *Automatica*, vol. 46, no. 5, pp. 823–831, 2010.

[25] P. Bolzern, P. Colaneri, and G. D. Nicolao, "Design of stabilizing strategies for discrete-time dual switching linear systems," *Automatica*, vol. 69, pp. 93–100, 2016.

[26] P. Bolzern, P. Colaneri, and G. D. Nicolao, "H ∞; co-design for discrete-time dual switching linear systems," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 1742–1747.

[27] J. W. Lee, "Infinite-horizon joint lqg synthesis of switching and feedback in discrete time," *IEEE Trans. Autom. Control*, vol. 54, no. 8, pp. 1945–1951, Aug. 2009.

[28] B. Lincoln and B. Bernhardsson, "Lqr optimization of linear system switching," *IEEE Trans. Autom. Control*, vol. 47, no. 10, pp. 1701–1705, Oct. 2002.

[29] S. J. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach* (3rd ed.). Upper Saddle River, NJ, USA: Prentice-Hall, 2009.

[30] W. Zhang, A. Abate, J. Hu, and M. P. Vitus, "Exponential stabilization of discrete-time switched linear systems," *Automatica*, vol. 45, no. 11, pp. 2526–2536, 2009.

[31] S. Bittanti and P. Colaneri, *Periodic Systems: Filtering and Control*. New York, NY, USA: Springer, 2009.

[32] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[33] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear Model Predictive Control*, ser. Lecture Notes in Control and Information Sciences, L. Magni, D. Raimondo, and F. Allgöwer, Eds. Berlin, Germany: Springer, 2009, vol. 384, pp. 345–369.

[34] M. Donkers, W. Heemels, N. van de Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Trans. Autom. Control*, vol. 56, no. 9, pp. 2101–2115, Sep. 2011.

[35] M. Fiacchini, A. Girard, and M. Jungers, "On the stabilizability of discrete-time switched linear systems: Novel conditions and comparisons," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1181–1193, May 2016.

[36] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Studies in Applied Mathematics, vol. 15). Philadelphia, PA, USA: SIAM, 1994.

Duarte Antunes (M'13) was born in Viseu, Portugal, in 1982. He received the Licenciatura degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal, in 2005, and the Ph.D. degree in automatic control from IST in 2011.

From 2011 to 2013 he held a postdoctoral position at the Eindhoven University of Technology (TU/e), The Netherlands. He is currently an Assistant Professor at the Department of Mechanical Engineering of TU/e. His research interests include networked control systems, stochastic control, dynamic programming, and systems biology.

W. P. Maurice Heemels (F'16) received the M.Sc. degree in mathematics and the Ph.D. degree in control theory (both *summa cum laude*) from the Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, in 1995 and 1999, respectively.

After being an Assistant Professor at the electrical engineering department at TU/e and a Research Fellow at the Embedded Systems Institute (ESI), he is currently a Full Professor in the Control Systems Technology Group in the mechanical engineering department at TU/e. He held visiting research positions at the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001), at Océ, Venlo, The Netherlands (2004), and at the University of California at Santa Barbara, CA, USA (2008). His current research interests include general system and control theory, hybrid and cyber-physical systems, networked and event-triggered control, and constrained systems including model predictive control.

Dr. Heemels is an Associate Editor for the journals *Nonlinear Analysis: Hybrid Systems* and *Automatica*. He was the recipient of a prestigious VICI grant of the Dutch Technology Foundation (STW) and the Netherlands Organisation for Scientific Research (NWO). In addition, he served as the general chair of the 4th IFAC Conference on Analysis and Design of Hybrid Systems (ADHS) 2012 in Eindhoven, The Netherlands, and was the IPC chair for the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS) 2013 in Koblenz, Germany.