Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Trade-Offs Between Precision and Computation Horizon in Real-Time Optimal Control of Switched Systems

Henrik Axelsson, Magnus Egerstedt, and Yorai Wardi

Abstract— **In this paper the problem of real-time optimal control is considered. In particular, we will study dynamical systems that switch between different modes at controlled switching times. Ideally, one would want the switching controller to provide sufficiently good values for the switching times at every instant of time. However, due to the limited computational resources available in many real-time applications, questions concerning trade-offs between the computation horizon and the precision of the solution arise naturally. These trade-offs constitute the main focus of this paper, and a solution will be provided based on the minimization of a conservative bound on the error between the true gradient and the gradient obtained in real-time.**

I. Introduction

As embedded controllers are introduced *en masse* in a number of novel yet resource intense applications, such as active structure control [1], advanced automotive processes [2], and autonomous robotics [3], resource management has become a bottleneck. In this paper, we approach this problem from a computational vantage point and investigate how computationally costly optimal control algorithms can be modified in such a way that they become applicable to real-time scenarios.

By "real-time" we understand hard constraints on the time the control processes have available to them before a result must be delivered. In the context of optimal control, these constraints will be translated into constraints on the accuracy of the numerical algorithms. In particular, we will investigate trade-offs between the horizon over which the solution is obtained and the precision of the numerical algorithm.

The underlying problem that we will address is a so-called switching-time optimization problem in which the controller must switch between a number of predefined control modes. These switches should moreover be such that a given cost-function is minimized. Computationally unconstrained optimal switch-time control is a well-studied topic and previous results include [4], [5], [6], and [7].

What makes the contribution of this paper novel is that exact solutions will not be available to us due to the real-time environment in which the system operates. The outline of this paper is as follows: In Section II, the switch-time optimization problem is presented and some initial results are given. Moreover, the real-time approach to solving this problem is presented. In particular, we will define the performance metric in terms of bounds on the error in the gradient as a function of the solution horizon and the numerical precision. In order to compute these bounds, the behavior of the state and costate equations must be characterized, which is the topic of Section III. This characterization is then put to use in Section IV for optimizing the aforementioned trade-offs, followed by a robot navigation example (Section V) in which an autonomous mobile robot must switch between different modes of operation (or behaviors) in real-time in order to navigate an unknown area.

This work was sponsored by the National Science Foundation through Grants # 0207411 and # 0509064.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 U.S.A., {henrik,magnus,ywardi}@ece.gatech.edu

II. Problem Formulation

Switched mode systems are systems whose dynamic responses change among various modes according to a prescribed supervisory-control law. Let $\{x(t)\}_{t=0}^{t_f}$ denote the state trajectory of the system, and suppose that it evolves according to the equation $\dot{x} = f(x,t)$, where the dynamic-response function $f : \mathbb{R}^n \times [0,t_f] \to \mathbb{R}^n$ comprises a sequential assignment of functions $f_i : \mathbb{R}^n \to \mathbb{R}^n$, $i = 1, 2, \ldots$. Let us fix $t_f > 0$ and suppose that the dynamic response changes N times in the interval $[0, t_f]$. Denoting the switching times by τ_i, $i = 1, \ldots, N$, in increasing order, and defining $\tau_0 := 0$ and $\tau_{N+1} := t_f$, we have that $0 = \tau_0 \leq \tau_1 \leq \ldots, \leq \tau_N \leq \tau_{N+1} = t_f$. Collectively, these switching times constitute a vector in $\mathbb{R}^N$, henceforth denoted by $\bar{\tau} := (\tau_1, \ldots, \tau_N)^T$. Furthermore, suppose that $f(x,t) = f_i(x)$ for every $t \in [\tau_{i-1}, \tau_i)$ and for every $i = 1, \ldots, N+1$. We then have the following differential equation defining the system's dynamics,

$$\dot{x}(t) \;=\; f_i(x(t)), \quad t \in [\tau_{i-1}, \tau_i), \quad i \in \{1, \ldots, N+1\}. \tag{1}$$

We assume throughout that the initial condition $x(0)$ is given and fixed. A more general setting includes an exogenous input $u(t)$ in the right-hand side of (1), but in this paper we concern only the autonomous case where such an input is absent.

The optimal control problem considered is that of optimizing the switching times in order to minimize a cost-function J over a fixed time interval:

$$P \qquad \min_{\bar{\tau}} J(\bar{\tau}) = \int_0^{t_f} L(x(t))dt$$
$$\textit{s.t.} \quad \dot{x}(t) = f_i(x(t)), \quad t \in [\tau_i, \tau_{i+1}),$$

where L is our instantaneous cost. Given some regularity conditions on f and L, an expression for the gradient of J was presented in [8] by the authors, and for the sake of completeness, we recall it below. The derivative of the cost-functional with respect to a switching time τ_i, where $i \in \{1, \ldots, N\}$, is given by

$$\frac{\partial J}{\partial \tau_i}(\bar{\tau}) = p(\tau_i)^T \left(f_{i-1}(x(\tau_i)) - f_i(x(\tau_i)) \right), \tag{2}$$

where $p(t) \in \mathbb{R}^n$ is the costate and satisfies the following differential equation,

$$\dot{p} = -\left(\frac{\partial f_{i+1}}{\partial x}(x(t))\right)^T p - \left(\frac{\partial L}{\partial x}(x(t))\right)^T, t \in (\tau_i, \tau_{i+1}], \quad (3)$$

for $i \in \{1, \ldots, N\}$ with the final condition $p(t_f) = 0$. Once the state trajectory $x(t)$ and the costate trajectory $p(t)$ are calculated, (2) is easily evaluated for all switching times.

The problem considered in this paper is that of solving problem P under the real-time constraint that we only have a finite amount of time available to calculate $x(t)$, $p(t)$ and to evaluate the gradient before we need to update the switches. The calculation of $x(t)$ and $p(t)$ is done through the Euler's-method [11] with step-length δt. Assume that the current time is t_c, where $t_c \in [0, t_f]$, and that we simulate $T \in (0, t_f - t_c]$ seconds into the future using a step-length of δt seconds. The real-time constraint considered in this paper has the following form

$$g(T, \delta t) \leq \Delta, \quad (4)$$

where Δ is the time we are allowed to simulate before we must update our switching time vector, and g is the time it takes to simulate T seconds with a step-length of δt. We will refer to g as the *simulation-time* function. In general $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$ will depend on the processor speed and it will be an increasing function in T, and decreasing in δt. We make the following simplifying assumption regarding the simulation-time function g:

Assumption 1: The simulation-time function is given by

$$g(T, \delta t) = C_{CPU} \frac{T}{\delta t}, \quad (5)$$

where C_{CPU} is a constant relating the CPU speed to the numerical simulation time.

At this point it should be mentioned that we assume that the state of the system is completely observable. This implies that every Δ seconds we can start the numerical integration of $x(t)$ using x at that time as the initial state.

Assuming we are given the dynamics, i.e. the sequence of $f_i's$ and the corresponding switching time vector $\bar{\tau}$, our problem is then to minimize the cost by updating the switches every Δ seconds while choosing T and δt to satisfy the real-time constraint (4). The switches are updated through a gradient descent algorithm with step-size γ, using the calculated gradient denoted by $\nabla \widetilde{J}(\bar{\tau}, T, \delta t)$.

The calculated gradient is given by evaluating (2), with $x(t)$ and $p(t)$ replaced by the state and costate trajectories obtained when solving (1) and (3) numerically. This is done through a numerical integration T seconds in to the future using a step-length of δt seconds, as mentioned earlier. Hence the calculated gradient, $\nabla \widetilde{J}$, will depend on T, δt and $\bar{\tau}$. Furthermore, the calculated gradient will also depend on the current time t_c as the state is updated every Δ seconds, thereby changing the initial condition in the calculation of the state trajectory. However, the dependence on t_c in $\nabla \widetilde{J}$ will be suppressed for notational convenience whenever it is clear from the context.

Having motivated why the calculated gradient depends on T and δt, our approach to optimize the cost in real-time can be presented. As mentioned earlier, we will optimize the cost in real-time through a gradient descent algorithm, with fixed step-size γ, using the calculated gradient. We want the difference in cost between updating the switching times with the true gradient as compared to updating them with the calculated gradient, to be as small as possible. This formulation allows us to minimize the cost with respect to T and δt. Hence, we will choose T and δt to minimize the following term every Δ seconds:

$$J_{diff}(\bar{\tau}, T, \delta t) = |J(\bar{\tau} - \gamma \nabla J(\bar{\tau})) - J(\bar{\tau} - \gamma \nabla \widetilde{J}(\bar{\tau}, T, \delta t))|, \quad (6)$$

subject to (4). Based on the second order Taylor expansion of the two terms in (6), together with the mean-value theorem, we get that

$$\begin{aligned} J(\bar{\tau} - \gamma \nabla J(\bar{\tau})) &= J(\bar{\tau}) - \gamma < \nabla J(\bar{\tau}), \nabla J(\bar{\tau}) > + \\ &\frac{\gamma^2}{2} \nabla J^T(\bar{\tau}) \cdot \frac{\partial^2 J}{\partial \bar{\tau}^2}(d) \cdot \nabla J(\bar{\tau}), \quad (7) \end{aligned}$$

for some vector d on the line segment between τ and $\tau - \gamma \nabla J(\bar{\tau})$. Likewise, for the second term of (6) we get that

$$\begin{aligned} J(\bar{\tau} - \gamma \nabla \widetilde{J}(\bar{\tau}, T, \delta t)) = J(\bar{\tau}) - \gamma < \nabla \widetilde{J}(\bar{\tau}), \nabla J(\tau) > \\ + \frac{\gamma^2}{2} \nabla \widetilde{J}^T(\bar{\tau}) \cdot \frac{\partial^2 J}{\partial \bar{\tau}^2}(d') \cdot \nabla \widetilde{J}(\bar{\tau}), \quad (8) \end{aligned}$$

for some vector d' on the line segment between $\bar{\tau}$ and $\bar{\tau} - \gamma \nabla \widetilde{J}(\bar{\tau}, T, \delta t)$. It was shown in [8] that both $J(\bar{\tau})$ and $\nabla J(\bar{\tau})$ are Lipschitz continuous in $\bar{\tau}$, with Lipschitz constants L_J and $L_{J'}$, and that $||\nabla J(\bar{\tau})||$ is bounded from above by some constant D. It then follows that the second derivative is bounded by some constant K_{J^2}, and the last terms of (7) and (8) are bounded by $\gamma^2 C_1$, where $C_1 = K_{J^2} D^2/2$. Summarizing, we get that

$$\begin{aligned} J_{diff}(\bar{\tau}, T, \delta t) = |\ J(\bar{\tau}) - \gamma \, ||\nabla J(\bar{\tau})||^2 - J(\bar{\tau}) + \\ \gamma < \nabla J(\bar{\tau}), \nabla \widetilde{J}(\bar{\tau}) > | + o(\gamma) C_1, \quad (9) \end{aligned}$$

where $o(\gamma)$ represents little o in the normal way, i.e. $\frac{o(\gamma)}{\gamma} \to 0$ as $\gamma \to 0$. Assuming we pick a small step-size, γ, we can approximate (6) by the following equation,

$$\begin{aligned} J_{diff}(\bar{\tau}, T, \delta t) &\approx \gamma \left| < \nabla J(\bar{\tau}), \nabla J(\bar{\tau}) - \nabla \widetilde{J}(\bar{\tau}) >_Q \right| \\ &\leq \gamma ||\nabla J(\bar{\tau})||_Q ||\nabla \widetilde{J}(\bar{\tau}) - \nabla J(\bar{\tau})||_Q \quad (10) \end{aligned}$$

where the last step follows from Cauchy-Schwartz inequality and the norm $||\cdot||_Q$ is a function of the switching vector and the current time t_c, to be defined in Section IV.

Considering the right hand side of (10), we see that the only term that can be explicitly controlled is $||\nabla \widetilde{J}(\bar{\tau}, T, \delta t) - J(\bar{\tau})||_Q$. This, since our control strategy is to minimize the difference between ∇J and $\nabla \widetilde{J}$ by choosing T and δt, hence we do not explicitly control $\bar{\tau}$. Therefore, we will minimize (6) by minimizing the supremum of the above term over T and δt. We thus have the following problem:

$$\begin{aligned} P_{RT} \qquad & \min_{T, \delta t} \left|\left| \nabla \widetilde{J}(\bar{\tau}, T, \delta t) - \nabla J(\bar{\tau}) \right|\right|_Q, \\ \textit{s.t.} \quad & g(T, \delta t) \leq \Delta, \end{aligned}$$

where the subscript RT denotes "real-time" and g is given in (5).

In order to solve problem P_{RT}, an expression for the upper bound of the norm of the error between the simulated state/costate and the true state/costate will be presented based on T and δt. To this end, the following assumption is needed:

Assumption 2: (i). The functions f_i, $i \in \{1, \ldots, N\}$, and the instantaneous cost L are continuously differentiable on $\mathbb{R}^n$. (ii). There exists a constant $K > 0$ such that, for every $x \in \mathbb{R}^n$, and for every $i \in \{1, \ldots, N\}$,

$$||f_i(x)|| \ \leq \ K(||x|| + 1). \tag{11}$$

Assumption 2 guarantees that both $f(x,t)$ and $\frac{\partial f(x,t)}{\partial x}$ are Lipschitz with Lipschitz constants L_f and $L_{f'}$. The same is true for $L(x(t))$ and $\frac{\partial L}{\partial x}$ with Lipschitz constants L_L and $L_{L'}$. Furthermore, (11) and Bellman-Grönwall's Lemma (see [9]) implies that $||x(t)||,\ t \in (0, t_f)$ is bounded by $(||x_0|| + Kt_f)e^{Kt_f}$. Using Assumption 2 again, we get that $||f|| \leq C_f$, where $C_f := K((||x_0|| + Kt_f)e^{Kt_f} + 1)$. We are now in position to present the upper bound on the error between the state x and the simulated state, denoted by x_s. Before doing that we note that there exists a bounded compact set X such that $x(t) \in X,\ \forall t \in [0, t_f]$ since $x(t)$ is bounded.

III. Bounds on the Error In the State and Costate

The bounds on the error in the state and the costate trajectories presented in this section are well known results in numerical analysis, see for example [10] for results of this nature. Hence, the propositions in this section are given without proofs.

The simulated state, obtained by solving (1) numerically, is denoted by $x_s[t, \delta t]$. Here, δt is the time step between each evaluation of (1) and t is the time variable. We solve for x_s using Euler's method. Starting at time 0, x_s is initialized to be equal to x_0, then, for Δ seconds, the real-time process calculates the state and costate trajectories and then evaluates the calculated gradient in order to update the switching times. After Δ seconds, the process updates x_s according to $x_s[\lfloor\Delta\rfloor, \delta t] = x(\Delta)$ and repeat the above steps for Δ seconds. Here, and in the subsequent part of this paper, $\lfloor\Delta\rfloor$ denotes the closest multiple of δt smaller than or equal to Δ. Defining the set $M(T, \delta t) := \{0, \delta t, 2\delta t, \ldots, \lfloor T\rfloor - \delta t, \lfloor T\rfloor\}$, we can then define the least upper bound of the norm of the error between the real state x and the simulated state x_s, at a time $t \in M(t_f, \delta t)$, to be the function $E[t, \delta t]$. The following proposition gives an upper bound on E that will be used when solving P_{RT}.

Proposition 1: Assume we are given a switching vector $\bar{\tau} \in \mathbb{R}^N$, and the associated dynamic response functions f_i, $i \in \{1, 2 \ldots, N+1\}$, where each f_i satisfies Assumption 2. Let the state trajectory x be given by solving (1) with a fixed initial condition, $x(0) = x_0$. Let $x_s[t, \delta t]$ denote the simulated state at time $t \in M(t_f, \delta t)$ obtained through solving (1) using Euler's method with step-size δt starting at $x_s[0, \delta t] = x_0$. Then the following inequality holds:

$$||x(t) - x_s[t, \delta t]|| \leq E(t, \delta t) \leq \left(e^{L_f t} - 1\right) C_f \delta t. \tag{12}$$

In order to solve P_{RT}, a bound on the norm of the error between the real costate $p(t)$ and the simulated costate, denoted by $p_s[t, T, \delta t]$ and calculated through Euler's method, must be derived. Regarding the error in the simulated costate, there are three sources of errors described below:

1) Firstly, according to (3), $\dot{p}$ depends on $x(t)$, in our case we only have access to $x_s[t, \delta t],\ \forall t \in M(t_f, \delta t)$.
2) Secondly, $p_s[t, T, \delta t]$ is calculated using Euler's method.
3) Finally, if $t_c + T < t_f$ then we do not have access to the state after time $t_c + T$, hence we can not use (3) to solve for p_s between $[t_c + T, t_f]$ as we need x_s in that interval.

We start by assuming that the simulation length T satisfies $t_c + T = t_f$ and derive the error due to the first two cases described above. To this end we note that, by Assumption 2, there exist two constants $C_{L'}$ and $C_{f'}$ such that $\left|\left|\frac{dL}{dx}\right|\right| \leq C_{L'}$, and $\left|\left|\frac{df}{dx}\right|\right| \leq C_{f'}, \forall x \in X$. These two bounds will give us the bounds on $||p||$ and $||\dot{p}||$ derived below, $||p(t)|| = || \int_t^{t_f} \left(\frac{df(x(s),s)}{dx}^T p(s) + \frac{dL(x(s))}{dx}^T\right) ds|| \leq (t_f - t)C_{L'} + \int_t^{t_f} C_{f'} ||p(s)|| ds$. This, together with Bellman-Grönwalls lemma gives the following bound for $||p(t)||$ for all $t \in (0, t_f)$,

$$||p(t)|| \leq (t_f - t)C_{L'}e^{C_{f'}(t_f - t)} \leq t_f C_{L'} e^{C_{f'} t_f}, \tag{13}$$

where we define the right hand side of (13) as C_p and trivially deduce the bound of $||\dot{p}||$ from (3) to be $C_{\dot{p}} := C_{L'} + C_{f'}C_p$. As done above for the error between x and x_s, we define the least upper bound of the norm of the error between the real costate p and the simulated costate p_s, at a time $t \in M(t_f, \delta t)$, to be the function $\tilde{E}[t, T, \delta t]$. The following proposition gives an upper bound on $\tilde{E}$ that will be used when solving P_{RT}.

Proposition 2: Assume we are given a switching vector $\bar{\tau} \in \mathbb{R}^N$, the associated dynamic response functions f_i, $i \in \{1, 2 \ldots, N + 1\}$, and a cost function L, where each f_i and L satisfies Assumption 2. Let the costate trajectory p be given by (3) with a fixed final condition, $p(t_f) = 0$. Let the simulation length T satisfy $T + t_c = t_f$, i.e. we simulate until the final time t_f. Let $p_s[t, T, \delta t]$ denote the simulated costate at time $t \in M(t_f, \delta t)$ obtained through solving (3) backwards using Euler's method, with step-size δt, starting at $p_s[t_f, T, \delta t] = 0$. Then the following inequality holds $\forall t \in M(t_f, \delta t)$,

$$||p(t) - p_s[t, T, \delta t]|| \leq \tilde{E}(t, t_f, \delta t) \leq S_1(\delta t)^2 + S_2 \delta t, \tag{14}$$

where the system constants, S_1 and S_2, are given by

$$S_1 := \quad e^{C_{f'} t_f} \left[L_{L'}([e^{L_f t_f} - 1]C_f + C_f) + 2C_{f'}C_{\dot{p}}\right], \tag{15}$$

$$S_2 := \quad [L_{L'}C_f + C_f C_{\dot{p}} + C_p C_{f'} C_f + (L_{L'} + C_p L_{h'}) \times (e^{L_f t_f} - 1)C_f]\, \frac{e^{C_{f'} t_f} - 1}{C_{f'}}. \tag{16}$$

If we are currently at time $t_c \in (0, t_f)$ when a simulation started, then by replacing t_f by $t_f - t_c$ in the expressions for S_1 and S_2, we get a bound for the error in the costate that reflects the current time.

Regarding the error due to the third source of error described earlier, when we do not simulate all the way until t_f, we set $p_s[t,T,\delta t]=0,\ \forall t\in(t_c+T,t_f)$, and integrate backward in time starting from t_c+T. The error due to this can be found by simply noting that by (13), $||p(t_c+T)||\leq (t_f-(t_c+T))C_{L'}e^{C_{f'}(t_f-(t_c+T))}$ and therefore,

$$||p_s[t,T,\delta t]-p(t)||\leq (t_f-(t_c+T))C_{L'}e^{C_{f'}(t_f-(t_c+T))}, \tag{17}$$

for all $t\in[t_c+T,t_f]$. The total error in $||p-p_s||$ is bounded above by the sum of the two errors presented in (14) and (17). An upper bound on the total error in the costate is given in the following proposition.

Proposition 3: Given the assumptions presented in Proposition 1, an upper bound for $\tilde{E}(t,T,\delta t)$ for a given step-size δt and a given simulation length T and $\forall t\in\{(t_c,t_f)\cap M(t_f,\delta t)\}$ is given by

$$\tilde{E}(t,T,\delta t)\leq S_1(\delta t)^2+S_2\delta t+ (t_f-(t_c+T))C_{L'}e^{C_{f'}(t_f-(t_c+T))}, \tag{18}$$

where S_1 and S_2 are given by (15) and (16) respectively, possibly replacing t_f with t_f-t_c.

Having derived upper bounds for the norm of the errors in the state $E(t,\delta t)$ and costate $\tilde{E}(t,T,\delta t)$, we are now in position to solve problem P_{RT} to the extent of deriving an upper bound of the last term in the right hand side of (10) depending only on T and δt.

IV. Real-Time Optimization

In order to optimize the cost in real-time subject to (4), we showed in Section II that this problem can be approached by minimizing $||\nabla\widetilde{J}(\bar{\tau},T,\delta t)-\nabla J(\bar{\tau})||_Q$, subject to T and δt, every Δ second. We assume that it is more important to optimize switches that are close to the current time t_c, than switches far away in the future and, hence, we let our norm reflect this assumption. Therefore, we will explicitly denote the norm by $||\cdot||_Q$ where

$$||\nabla J(\bar{\tau})-\nabla\widetilde{J}(\bar{\tau},T,\delta t)||_Q := \sum_{\substack{\tau_i\in\bar{\tau}\ s.t.\\ \tau_i>t_c+\Delta}}||\frac{\partial\widetilde{J}}{\partial\tau_i}(\bar{\tau},T,\delta t)-\frac{\partial J}{\partial\tau_i}(\bar{\tau})||\beta^{\tau_i-(t_c+\Delta)}, \tag{19}$$

where $\beta\in(0,1]$ describes how important it is to optimize switches close to t_c relative to switches far away from t_c. If $\beta=1$ then all switches are equally important to optimize. If β is close to zero, we consider it more important to optimize switches close to t_c This is a good choice of norm since the switches close to t_c will be updated fewer times then the switches far away, as a simulation takes Δ seconds.

We define $F(\tau_i,T,\delta t)$ to be an upper bound of $||\frac{\partial\widetilde{J}}{\partial\tau_i}(\bar{\tau},T,\delta t)-\frac{\partial J}{\partial\tau_i}(\bar{\tau})||$ for $\forall\tau_i\in\bar{\tau}\ s.t.\ \tau_i\geq t_c+\Delta$, where the simulated derivative is given by

$$\frac{\partial\widetilde{J}}{\partial\tau_i}(\bar{\tau},T,\delta t)=p_s[\tau_i,T,\delta t]^T(f_{i-1}(x_s[\tau_i,\delta t])-f_i(x_s[\tau_i,\delta t])), \tag{20}$$

and the real derivative is given by (2). Since we are only interested in changing the switches that have not already passed, i.e. we only care abount switches that is after $t_c+\Delta$, $F(\tau_i,T,\delta t)$ is not defined for $\tau_i's$ such that $\tau_i<t_c+\Delta$. Likewise, we note that $\frac{\partial\widetilde{J}}{\partial\tau_i}(\bar{\tau},T,\delta t)=0$ if $\tau_i>t_c+T$ since we set $p_s[t,T,\delta t]=0$ for $t>t_c+T$ in order to minimize the error between p_s and p. Rewriting $||\frac{\partial\widetilde{J}}{\partial\tau_i}(\bar{\tau},T,\delta t)-\frac{\partial J}{\partial\tau_i}(\bar{\tau})||$, we get that

$$\begin{aligned}||\tfrac{\partial\widetilde{J}}{\partial\tau_i}(\bar{\tau},T,\delta t)-\tfrac{\partial J}{\partial\tau_i}(\bar{\tau})||&=\dots\\ =||[p(\tau_i)^T-p_s[\tau_i,T,\delta t]^T][f_{i-1}(x_s[\tau_i,\delta t])&-\\ f_i(x_s[\tau_i,\delta t])]-p(\tau_i)^T[f_{i-1}(x_s[\tau_i,\delta t])&\\ -f_i(x_s[\tau_i,\delta t])-f_{i-1}(x(\tau_i))+f_i(x(\tau_i))]||&,\end{aligned}$$

where an upper bound of the above expression is given by,

$$F(\tau_i,T,\delta t)\leq 2C_f\tilde{E}(\tau_i,T,\delta t)+2C_pL_fE(\tau_i,\delta t), \tag{21}$$

for all $\tau_i<t_c+T$. Likewise,

$$F(\tau_i,T,\delta t)\leq 2C_f(t_f-\tau_i)C_{L'}e^{C_{f'}(t_f-\tau_i)}, \tag{22}$$

if $\tau_i\geq t_c+T$, where the right hand side of (22) is the upper bound of $||\frac{dJ}{d\tau_i}||$ using (13). Using (12) and (18), we can get an upper bound for the right hand side of (21) that do not depend on the switching time τ_i. This upper bound is denoted by $F_1(T,\delta t)$, where

$$\begin{aligned}F_1(T,\delta t):=2C_f\left\{((t_f-(t_c+T))C_{L'}e^{C_{f'}(t_f-(t_c+T))})\right.&\\ \left.+S_1(\delta t)^2+S_2\delta t\right\}+2C_pL_f\left[e^{L_fT}-1\right]C_f\delta t,&\end{aligned} \tag{23}$$

and satisfies $F(\tau_i,T,\delta t)\leq F_1(T,\delta t),\ \forall\tau_i\in\bar{\tau}$ s.t. $\tau_i\geq t_c+\Delta$. We note that since $F_1(T,\delta t)$ do not depend on $\bar{\tau}$, $F_1(T,\delta t)$ can easily be calculated off-line and stored in order for the controller to look it up when the process is running in real-time. This saves computation time, and this is why $F_1(T,\delta t)$ was introduced. We also define the right hand side of (22) by $F_2(\tau_i)$, i.e. $F_2(\tau_i):=2C_f(t_f-\tau_i)C_{L'}e^{C_{f'}(t_f-\tau_i)}$. An upper bound of (19), using $F_1(T,\delta t)$ and $F_2(\tau_i)$, is given by

$$\begin{aligned}||\nabla J(\bar{\tau})-\nabla\widetilde{J}(\bar{\tau})||_Q\leq F_1(T,\delta t)\sum\nolimits_{\substack{\tau_i\in\bar{\tau}\ s.t.\\ t_c+\Delta<\tau_i<t_c+T}}\beta^{\tau_i-(t_c+\Delta)}&\\ +\sum\nolimits_{\substack{\tau_i\in\bar{\tau}\ s.t.\\ \tau_i>t_c+T}}F_2(\tau_i)\beta^{\tau_i-(t_c+\Delta)}.&\end{aligned} \tag{24}$$

Since we assume that we simulate as long as the real-time constraint allow us to, it follows from (5) that $T=\frac{\Delta\cdot\delta t}{C_{CPU}}$ and $F_1(\delta t,T)=F_1(C_{CPU}\frac{T}{\Delta},T)=F_1(T)$ is only a function of the simulation length. Looking at (24), we see that the limits of the summation for the two sums changes each time t_c+T passes by a switching time. The two sums do not change when T is between any two switching points τ_i and τ_{i+1}, therefore, for each interval (τ_i,τ_{i+1}) we only need to consider the minimum value of $F_1(T)$ in that interval. This can be done in real-time since it only requires looking up the T value that minimizes $F_1(T)$ in given intervals (τ_i,τ_{i+1}). There, is no benefit evaluating T more than every δt seconds, i.e. $T\in M(t_f,\delta t)$, since we only have access to x_s and p_s every δt second. From a practical point of view, a significantly coarser grid can be chosen to evaluate T over, i.e. $T\in M(t_f,j\cdot\delta t)$ for some integer $j\gg 1$.

In order to solve problem P_{RT}, the following algorithm can be employed every Δ seconds:

Algorithm 4.1: Real-time switch-time optimization

Given: Δ, C_{CPU}, $\bar{\tau}$, β, a positive integer j, the step-size

γ and the system parameters, i.e. the order of the modal sequence and the cost function to be minimized.
Init: Calculate $F_1(T)\ \forall\ T \in M(t_f, j\cdot\delta t)$, set $t_c = 0$.
Step 1: Evaluate $i^* = \arg\min\{\tau_i \in \bar{\tau}\ |\tau_i > t_c + \Delta\}$. For $\forall i \in \{i^*, \ldots, card(\bar{\tau})\}$, let $T_i^o = \min\{F(T)\ |\ T \in (\tau_i, \tau_{i+1}) \cap M(t_f, j\cdot\delta t)\}$.
Step 2: For all T_i^o calculated in Step 1, evaluate (24), define T^* to be the $T_i^o,\ i \in \{i^*, \ldots, card(\bar{\tau})\}$ that minimizes (24).
Step 3: Using Euler's method, calculate x_s from time t_c to $T^* + t_c$ and p_s from time $T^* + t_c$ to τ_{i^*}. Evaluate $\nabla\widetilde{J}(\bar{\tau}, T, \delta t)$.
Step 4: For all $\tau_i \in \bar{\tau}$ s.t. $t_c + \Delta < \tau_i < t_c + T^*$, update the switch by

$$\tau_i := \max\left\{\tau_i - \gamma\frac{d\widetilde{J}}{d\tau_i},\ t_c + \Delta\right\}. \tag{25}$$

Step 5: If $t_c + \Delta < t_f$, set $t_c := t_c + \Delta$ and go to Step 1, otherwise STOP.

Note that the proposed method is based on the minimization of a bound on the difference between the calculated and the true gradient. A natural question to ask is whether this bound is overly conservative or not. This question depends on the system dynamics, and it is illustrated in the next section.

V. Example

As an application, we consider a robotics example where a mobile robot goes to a goal, denoted by x_g, while avoiding an obstacle located at x_{ob}. This must be achieved by choosing among different behaviors and the times to switch between them. Furthermore, the dynamics of the robot is assumed to be a single integrator, i.e. $\dot{x} = u \in \mathbb{R}^2$ for some control signal or feedback law u. We consider using the following three standard behaviors: *go-to-goal, go-around-obstacle-clockwise*, and *go-around-obstacle-counterclockwise*. The respective behaviors are denoted f_g, $f_{\circlearrowright}$ and $f_{\circlearrowleft}$, and are given by $f_g(x) = v(x_g - x),\ f_{\circlearrowleft}(x) = -f_{\circlearrowright}(x)$, and

$$f_{\circlearrowright}(x) = v\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\frac{x_{ob}-x}{||x_{ob}-x||},$$

where v is a positive scalar equal to 1 and $f_{\circlearrowright}(x)$ and $f_{\circlearrowleft}(x)$ correspond to the robot moving in a circle around the obstacle in the given direction. For example, assume we evolve according to $f_g(x)$ until time τ_1, at which point the robot encounters an obstacle to the left, and therefore switches to $f_{\circlearrowleft}(x)$. It then evolves according to $f_{\circlearrowleft}(x)$ until time τ_2, where it switches back to $f_g(x)$ and evolves according to this behavior until the final time $t_f = 3$ seconds. The trajectory of the robot is then given by evaluating (1) with the following mode-sequence $\{f_g, f_{\circlearrowleft}, f_g\}$. In this case, the state trajectory is completely determined by the switching times, τ_1 and τ_2, and can therefore be calculated analytically. Hence, in this example, there are no errors associated with the calculation of the state trajectory and $E(t, \delta t) = 0$.

In order for the robot not to collide with any obstacles while moving towards the goal, the instantaneous cost $L(\cdot) : \mathbb{R}^2 \to \mathbb{R}_+ \cup \{0\}$ is chosen as

$$L(x(t)) = \rho||x_g - x(t)||^2 + \alpha e^{-\frac{||x_{ob}-x(t)||^2}{\beta}}, \tag{26}$$

where $\rho = 0.01$ is the gain of the goal attraction term, $\alpha = 2$ is the gain of the obstacle avoidance term, and $\beta = 0.1$ is the shaping parameter for the range of the obstacle avoidance term.

As noted earlier, the robot does not know the exact location of the obstacle before it starts running, therefore it can not determine good switching times. However, we do assume that the robot knows that if there is an obstacle, it will be to the left of its trajectory. Therefore, the robot is initialized to evolve according to the following sequence of behaviors $f_g, f_{\circlearrowleft}, f_g$ and the switching vector is initialized to be $\bar{\tau} = (0,\ 0.5,\ 1.5,\ 3)$. The robot starts at $x_0 = (0.05, 0)^T$ and the goal is located at $x_g = (0, 2.5)^T$. We assume that we can only simulate 0.25 seconds into the future before we have to update our switching times, hence $\Delta = 0.25$.

In order to determine how far into the future to simulate, i.e. what T to choose, we will minimize the upper bound of the error between the true gradient and the simulated gradient with respect to how far into the future we simulate according to the right hand side of (24). However, if we apply naive bounds on the constants needed to calculate $F_1(T)$, as described in (23), and $F_2(\tau_i)$, the bound will be very conservative in two ways. Firstly, if we do not have enough computing power, i.e. C_{CPU} is not small enough, the minimum of the right hand side of (24) might be of orders of magnitude bigger than the actual gradient value and minimizing the upper bound would not give us any additional information. Secondly, if we have enough computing power, minimizing the right hand side of (24) will typically tell us to simulate all the way to the end, i.e. T will be such that $t_c + T = t_f$. The problem is that if we do not simulate all the way until t_f the error in p will grow very fast since we have not made any assumption regarding the dynamics of the state. After time τ_3, we have that

$$\dot{p} = -\frac{df_g^T}{dx}(x(t))p(t) - \frac{dL^T}{dx}(x(t)), \tag{27}$$

for $t \in (\tau_2, 3]$, where we note that $\frac{df_g}{dx}$ is the identity matrix and therefore, $-\frac{df_g}{dx}$ is negative definite and the first term in the right hand side of (27) is stable. Regarding the second term in the right hand side of (27), we assume that $||x_{ob} - x(t)|| \geq 0.65$, which is reasonable since we want the robot to avoid the obstacle, and that $||x_g - x(t)|| < 2,\ \forall t \in [\tau_2, 3]$, we then get that $||\frac{dL^T}{dx}(x(t))|| = 2\rho||x_g - x(t)|| + \frac{2\alpha}{\beta}||x_{ob} - x(t)||e^{-\frac{||x_{ob}-x(t)||^2}{\beta}}|| < 4\rho + \frac{2\alpha}{\beta}||x_{ob} - x(t)||e^{-\frac{||x_{ob}-x(t)||^2}{\beta}} \leq 4\rho + \frac{2\alpha}{\beta}\max_z\{z \geq 0.65\ |\ ze^{-\frac{z^2}{\beta}}\} = 0.42,\ \forall t \in [\tau_2, t_f]$. If we define $c_1 = 0.42$ and $c_2 = 1$, and denote the bound on the costate in $(\tau_2, t_f]$ by p_b, we get that $\dot{p}_b = c_1 + p_b c_2$. Solving for p_b backwards it follows that,

$$p_b(t) = \frac{c_1}{c_2}(1 - e^{(t-t_f)c_2}),\quad \forall t \in (\tau_2, t_f].$$

Using this bound for the costate after time τ_2 implies that $||p_s[t, T, \delta t] - p(t)|| \leq \frac{c_1}{c_2}(1 - e^{(t-t_f)c_2})$ for all $t \geq T + t_c$. This bound enable us to get a good result when evaluating (24) using reasonable C_{CPU} values. To summarize, we have the following expression for the bound of the costate,

t_c	$\bar{\tau} = (\tau_1, \tau_2)^T$	$T + t_c$	$x(t_c)$	$J(\bar{\tau})$
0	$(0.5, 1.50)^T$	1.5	$(0.05, 0)^T$	0.0444
0.25	$(0.39, 1.50)^T$	3	$(0.04, 0.55)^T$	0.0397
0.50	$(0.39, 1.50)^T$	3	$(0.11, 0.87)^T$	0.0397
0.75	$(0.39, 1.51)^T$	3	$(0.27, 1.08)^T$	0.0396
1	$(0.39, 1.51)^T$	3	$(0.36, 1.31)^T$	0.0396
1.25	$(0.39, 1.52)^T$	3	$(0.37, 1.57)^T$	0.0395
1.50	$(0.39, 1.52)^T$	3	$(0.32, 1.81)^T$	0.0395

TABLE I

SWITCHING TIMES, ROBOTS POSITION, AND COST AS A FUNCTION OF t_c.

$$C_p(t) = \begin{cases} \frac{c_1}{c_2}(1 - e^{(t-t_f)c_2}), \quad \tau_2 \le t \le t_f, \\ \frac{c_1}{c_2}(1 - e^{(\tau_2-t_f)c_2}) + (\tau_2 - \tau_1)C_{L'}e^{C_{f'}(\tau_2-\tau_1)}, \\ \tau_1 \le t \le \tau_2, \end{cases}$$

and we do not care about p before time τ_1 since τ_1 is the first switching time. Furthermore, we will introduce a time dependence on S_1 and S_2 and evaluate them for different switching times. We then have that

$$\begin{cases} S_{1,\tau_i}(T) = e^{C_{f'}(t_c+T-\tau_i)}\left(2C_{f'}C_{\dot{p}}(t) + L_{L'}C_f\right), \\ S_{2,\tau_i}(T) = \frac{e^{C_{f'}(t_c+T-\tau_i)}-1}{C_{f'}}[L_{L'}C_f + C_{f'}C_{\dot{p}}(t) + \\ C_p(t)L_{h'}C_f], \end{cases}$$

since $E(t, \delta t) = 0$, and $C_{\dot{p}}(t)$ follows from $C_p(t)$ and (3). In our case, we evaluate T every 0.1 seconds. Figure 1 shows a plot of the right hand side of (24) when we are at time $t_c = 0$. At this point, we assume the robot sees the obstacle located at $x_{ob} = (-0.5, 1.5)^T$. The value of the system constants are as follows: $C_f = 1$ since we are only interested in times $t \ge \tau_1$, likewise $C_{f'} = 1$, $C_L = \frac{2^2}{\rho} + 2e^{-\frac{||x_{ob}-x||^2}{\beta}}\Big|_{||x_{ob}-x||=0.65} = 0.07$, $C_{L'} = \frac{2^2}{\rho} + 2^2\frac{0.65}{\beta}e^{-\frac{0.65}{\beta}} = 0.42$, $L_L = C_{L'}$, $L_{L'} = \frac{2}{\rho} + \frac{2^2}{\beta}e^{-\frac{0.65^2}{\beta}} = 0.60$, $L_f = 1$, $L_{h'} = 1$.

It can be seen that the minimum in Figure 1 is obtained at $T = \tau_2$. Therefore the robot calculates p_s backwards from time τ_2, where $p_s[\tau_2, \tau_2, \delta t] = 0$, until time τ_1 and then evaluate the gradient and update the switching vector. Table 1 shows where the minimum T is attained for different $t_c s$, how the switching vector is updated, the robots position at time t_c, and how the cost changes for every iteration of Algorithm 4.1. As can be seen from Table 1, Algorithm 4.1 effectively reduces the cost in real-time. At time $t_c = 0.25$, we see that $t_c + \Delta > \tau_1$, hence the first switching time would have passed before we have a chance to update it. Therefore, in the right hand side of (24) the summation changes and hence, we get a new time T that minimizes the right hand side of (24). Finally, Figure 2 shows the robots final trajectory together with the initial trajectory.

VI. CONCLUSION

In this paper, an approach to real-time optimization of the switching-times in autonomous hybrid systems is proposed. The approach is based on minimizing an upper bound of the norm between the true gradient and the gradient we obtain by simulating the state and the costate trajectories. To this end, upper bounds for the error between the true state/costate and the simulated state/costate trajectories are presented as a function of the computation horizon and the time available to the controller before a result must be delivered. The bound of the norm between the true gradient and the simulated gradient has a simple form and therefore we can minimize this norm in real-time with little computational effort.

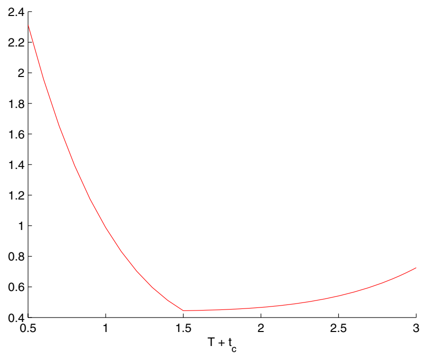

Fig. 1. Right hand side of (24) evaluated at $t_c = 0$.

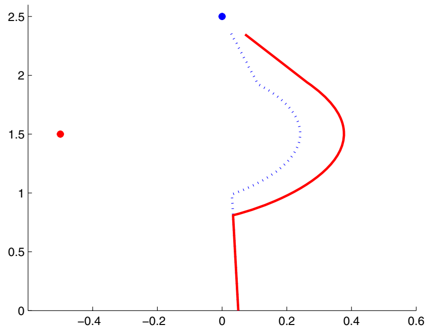

Fig. 2. Robots initial (dotted) and final (solid) trajectory.

REFERENCES

[1] W. K. Gawronski. *Advanced Structural Dynamics and Active Control of Structures*, Springer-Verlag, New-York, 2004.

[2] C. Bohn, H. Karkosch, P.M. Mariefeld, and F. Svaricek. Automotive Applications of Rapid Prototyping for Active Vibration Control. In *IFAC Advances in Automotive Control Engineering Practice*, pp. 1029-1039, 2004.

[3] D. Kortenkamp, R.P Bonasso and R Murphy, *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems*, AAAI Press, Cambridge, Massachusetts, 1998.

[4] A. Bemporad, A. Giua, and C. Seatzu. A Master-Slave Algorithm for the Optimal Control of Continuous-Time Switched Affine Systems. *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 1976-1981, Las Vegas, Nevada, December 2002.

[5] M.S. Branicky, V.S. Borkar, and S.K. Mitter. A Unified Framework for Hybrid Control: Model and Optimal Control Theory. *IEEE Transactions on Automatic Control*, Vol. 43, pp. 31-45, 1998.

[6] A. Rantzer and M. Johansson. Piecewise Linear Quadratic Optimal Control. *IEEE Transactions on Automatic Control*, Vol. 54, pp. 629-637, 2000.

[7] X. Xu and P. Antsaklis. Optimal Control of Switched Autonomous Systems. *IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[8] M. Egerstedt, Y. Wardi, and H. Axelsson, "Transition-Time Optimization for Switched-Mode Dynamical Systems," *IEEE Trans. Automat. Contr.*, vol 51, No. 1, pp. 110-115, Jan. 2006.

[9] E. Polak. *Optimization Algorithms and Consistent Approximations*. Springer-Verlag, New York, 1997.

[10] C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

[11] M.T. Heath. *Scientific Computing, An Introductory Survey*, McGraw-Hill.