On-line Optimal Timing Control of Switched Systems

Y. Wardi, P. Twu, and M. Egerstedt

***Abstract*— This paper considers a real-time algorithm for performance optimization of switched-mode hybrid dynamical systems. The controlled parameter consists of the switching times between the modes, and the cost criterion has the form of the integral of a performance function defined on the system's state trajectory. The dynamic response functions (state equations) associated with the modes are not known in advance; rather, at each time t, they are estimated for all future times $s \geq t$. A first-order algorithm is proposed and its behavior is analyzed in terms of its convergence rate. Finally, an example of a mobile robot tracking a moving target while avoiding obstacles is presented.**

I. Introduction

Consider the switched-mode hybrid dynamical system characterized by the following state equation,

$$\dot{x} = f(x, v, t), \tag{1}$$

where $t \in [0, T]$ for a given $T > 0$, $x \in \mathbb{R}^n$ is the state variable, and the input $v : [0, T] \to \Gamma$ is a discrete control having values in a finite set Γ. In this equation T is assumed to be fixed, the initial state is $x(0) = x_0$ for a given $x_0 \in \mathbb{R}^n$, and $f : \mathbb{R}^n \times \Gamma \times [0, T] \to \mathbb{R}^n$ is a suitable function guaranteeing the existence of a unique solution of the state equation (1) for a class of admissible controls $v : \Gamma \to [0, T]$. The problem of optimally controlling such systems arises in various applications (see, e.g., [1] for a survey), and often it amounts to minimizing a cost-functional of the form

$$J = \int_0^T L(x, v, t)dt \tag{2}$$

for a given performance function $L : \mathbb{R}^n \times \Gamma \times [0, T] :\to \mathbb{R}$. The objective of this paper is the development and investigation of an algorithm for this problem in a real-time setting.

This general nonlinear optimal control problem has been formulated in [2] in the setting of hybrid systems. Several variants of the maximum principle were derived in [3], [4], [5], and subsequently various algorithms were developed in [6], [1], [7], [5], [8], [9]. Most of these results concern the case where the controlled variable consists of the switching times of the discrete control v among a fixed sequence of values in Γ, but the case of variable sequence-values was considered as well [9], [8]. More recently the question of real-time optimization has begun to be addressed, and initial results were obtained in [10], [11], [12].

The need for real-time algorithms typically arises when complete information about the system is not available a priori but the algorithm can acquire partial information about it in real time. In these situations the objective is not to optimize the cost functional defined by Equation (2), but rather to reduce the cost to go at certain times.

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. Email: ywardi@ece.gatech.edu, ptwu@gatech.edu, magnus@ece.gatech.edu

In previously-published papers on real-time algorithms [10], [11], [12] the uncertainty in the system's parameters is due to state estimation via observers or the lack of complete knowledge of the cost functions, and for these situations, second-order algorithms are suitable. In contrast, in this paper the uncertainty is due to the lack of knowledge of the system's dynamic response functions and the need to estimate the functional form of the state equations, and this imposes far-greater efforts to compute the Hessians. In fact, the Hessians require numerical solutions of the state transition matrices and their derivatives at each time, and these may not be computable in real time. Therefore the algorithm considered in this paper is a first-order technique. Furthermore, its use may have other advantages over second-order algorithms, including global convergence and the absence of the need to compute inverses of the Hessian matrices or approximations thereto.

Similarly to earlier works on real-time algorithms [10], [11], [12], this paper assumes that the mode-sequence is given and the variable parameter consists of the switching times between the modes. The more difficult case where the mode-sequence is a part of the optimization variable can be handled by ad-hoc techniques as in [11], and its systematic analysis will be done elsewhere.

The rest of the paper is organized as follows. Section 2 formulates the problem and recounts some relevant results, Section 3 presents the algorithm, and Section 4 contains a numerical example. Due to space limitations, the proof of some auxiliary results (lemmas) are omitted and can be found in [13].

II. Problem formulation

Consider a fixed sequence of values of the control $v \in \Gamma$, $\{v_i\}$, $i = 1, \ldots, N+1$, for a given $N \geq 0$. For every $i = 1, \ldots, N+1$, we will use the notation $f_i(x, t) := f(x, v_i, t)$. Assuming a given final time T, let us denote by $\tau_1, \ldots, \tau_N$ the times at which the values of v are changed: $v = v_1$ for $t \in [0, \tau_1)$, $v = v_2$ for $t \in [\tau_1, \tau_2)$, etc. Denote these switching times collectively by the vector $\bar{\tau} := (\tau_1, \ldots, \tau_N)^T \in \mathbb{R}^N$, and for the sake of notation, define $\tau_0 := 0$ and $\tau_{N+1} := T$. We will consider the state equation (1) as depending on the switching-times vector $\bar{\tau}$. To formalize this dependence, define the function $F(x, t, \bar{\tau})$ by $F(x, t, \bar{\tau}) = f_i(x, t)\ \forall\ t \in$

$[\tau_{i-1}, \tau_i),\ \forall\ i = 1, \ldots, N+1$. Then, Equation (1) becomes

$$\dot{x} = F(x, t, \bar{\tau}) := f_i(x, t),\ \forall\, t \in [\tau_{i-1}, \tau_i),\ i = 1, \ldots, N+1; \tag{3}$$

and we assume a given initial condition $x(0) = x_0$. Given a function $L : \mathbb{R}^n \times [0, T] \to \mathbb{R}$, we define the cost functional J, as a function of the switching times, by

$$J(\bar{\tau}) \ := \ \int_0^T L(x, t) dt. \tag{4}$$

The functions $f_i,\ i = 1, \ldots, N+1$, correspond to the system's modes and hence are called the modal functions; we will use the simpler term "modes" when referring to them. We make the following assumption.

Assumption 1: The functions f_i and L are C^∞, namely they have continuous derivatives of all orders.

Since the mode-sequence is fixed and $\tau_{i+1} \geq \tau_i$, we define the feasible set for $\bar{\tau}$ by

$$\Lambda = \{\bar{\tau} \in \mathbb{R}^N : 0 := \tau_0 \leq \tau_1 \leq \ldots, \leq \tau_N \leq \tau_{N+1} := T\}. \tag{5}$$

Observe that we allow for the case $\tau_{i-1} = \tau_i$, which corresponds to the presence of mode i for a time interval of duration 0. This is done in order to ensure that the feasible set is closed. Define the optimal control (or, optimization) problem Π by

$$\Pi : \quad \min\{J(\bar{\tau}) \ : \ \bar{\tau} \in \Lambda\}. \tag{6}$$

Suppose now that the modal functions are not known in the future but have to be estimated. More precisely, for a given time $t \in [0, T)$, the functions f_i are not known for future times $s > t$, but are estimated by functions $\tilde{f}_i$. These are modal functions of a state variable $\tilde{x}$, time t, and the future time $s \geq t$, and will be denoted by $\tilde{f}_i(\tilde{x}, s, t)$. To be precise, this notation has the following meaning: $\tilde{f}_i$ approximates f_i in a way that depends on some data available at time t, and it is a modal function of the future state variable $\tilde{x}$ and future times $s \geq t$. Now fix $t \in [0, T)$, a state $x \in \mathbb{R}^n$, and a switching-times vector $\bar{\tau} \in \mathbb{R}^N$. Consider the evolution of the future state variable $\tilde{x}$ in future times $s \geq t$, according to the modal functions $\tilde{f}_i$ and the switching schedule $\bar{\tau}$, starting from the initial condition x at time t. This future state variable depends on t, $s \geq t$, x, and $\bar{\tau}$, and hence is denoted by $\tilde{x}(s, t, x, \bar{\tau})$. $\tilde{x}(s, t, x, \bar{\tau})$ is defined by the following equation, which is similar to (3) except that the initial time is t:

$$\frac{\partial \tilde{x}}{\partial s}(s, t, x, \bar{\tau}) \ = \ \tilde{f}_i\big(\tilde{x}(s, t, x, \bar{\tau}), s, t\big),$$
$$\forall\ s \in [t, T] \cap [\tau_{i-1}, \tau_i),\ i = 1, \ldots, N+1, \tag{7}$$

with the boundary condition $\tilde{x}(t, t, x, \bar{\tau}) = x$. We make the following observations about this equation. First, the vector $\bar{\tau}$ need not have a constant value but may be a function of time, and hence denoted by $\bar{\tau}(t) = (\tau_1(t), \ldots, \tau_N(t))^T \in \mathbb{R}^N$. Second, past modes that are no longer active at time t are irrelevant to the evolution of the future state. Define $q(t) := \max\{i = 1, \ldots, N \ : \ \tau_{i-1}(t) < t\}$, then Eq. (7) has the following form,

$$\frac{\partial \tilde{x}}{\partial s}(s, t, x, \bar{\tau}(t)) \ = \ F(\tilde{x}, s, t, \bar{\tau}(t)) \ :=$$
$$\tilde{f}_i\big(\tilde{x}(s, t, x, \bar{\tau}(t)), s, t\big),$$
$$\forall\ s \in [t, T],\ \forall\ i = q(t), \ldots, N+1, \tag{8}$$

with the boundary condition $\tilde{x}(s, s, x, \bar{\tau}(t)) = x$. We will be particularly interested in the case where $x = x(t)$, the state of the system defined by Eq. (3) with the slight modification that $\bar{\tau}(t)$ replaces $\bar{\tau}$. Finally, we mention that the time t and state $x(t)$ are called the *real* time and *real* state, whereas the times $s > t$ and future states $\tilde{x}(s, t, x(t), \bar{\tau}(t))$ are called the *simulated* time and state.

Suppose next that the performance function $L(x, t)$ is not known at future times $s \geq t$ but rather approximated (estimated) by a function $\tilde{L}(x, s, t)$. This notation indicates that the approximated function at the simulated time $s \geq t$ depends on the time the approximation is performed, which is t. Similarly to Eq. (4), given a switching-time vector $\bar{\tau}$, we define the cost-to-go performance function by

$$\tilde{J}(t, x(t), \bar{\tau}) \ = \ \int_t^T \tilde{L}\big(\tilde{x}(s, t, x(t), \bar{\tau}), s, t\big) ds. \tag{9}$$

Given $t \in [0, T]$ and the system's state $x(t)$, consider $\tilde{J}(t, x(t), \cdot)$ as a function of $\bar{\tau}$. Past switching times, namely those switching times that occurred before time t, cannot be modified, and therefore $\tilde{J}(t, x(t), \bar{\tau})$ can be only a function of $\tau_i,\ i = q(t), \ldots, N$, but not of $\tau_j,\ j = 1, \ldots, q(t) - 1$. Denoting by Λ_t the constraint-set

$$\Lambda_t \ := \ \{\bar{\tau} \ : \ t \leq \tau_{q(t)} \leq \tau_{q(t)+1} \ldots \leq \tau_{N+1} = T\}, \tag{10}$$

we further denote by Π_t the problem of minimizing $\tilde{J}(t, x(t), \bar{\tau})$ with respect to $\bar{\tau}$, subject to the constraint $\bar{\tau} \in \Lambda_t$, where by a slight abuse of notation we regard $\bar{\tau}$ as the vector $\bar{\tau} = (\tau_{q(t)}, \ldots, \tau_N)^T \in \mathbb{R}^{N-q(t)+1}$ whose dimension is time-varying. This is the problem that will concern us in the sequel, and we point out that, as is common in nonlinear programming, we seek as solution points vectors $\bar{\tau}$ that satisfy necessary local-optimality conditions, like stationary or Kuhn-Tucker, and not global solutions.

Now we will consider the case where the switching-time vector, $\bar{\tau}(t)$, is a function of t. Ideally we would like to choose $\bar{\tau}(t)$ to be a solution point for Π_t for every $t \in [0, T]$, but this of course is infeasible since it would require infinite computing speed and precision. Instead, we define and compute $\bar{\tau}(t)$ by performing a single step of a gradient-descent algorithm that aims at solving Π_t. The required computation takes a positive amount of time, and hence $\bar{\tau}(t)$ will be computed only for a finite number of time-points $t \in [0, T]$. This will be made clear in the discussion on the algorithm that will be carried out in the next section.

III. Real-Time Algorithm

The algorithm considered in this section is the gradient-descent technique with Armijo step size, adapted to the real-time setting and particular constraints considered in this paper. Gradient-descent algorithms with Armijo step size have been extensively tested on optimal control problems, and they have the properties of global convergence and linear convergence rate [14]. We recall from [14] the basic algorithm in the abstract setting of minimizing a continuously-differentiable (C^1) function $g(z) : \mathbb{R}^n \rightarrow \mathbb{R}$:

Algorithm 1: Fix constants $\alpha \in (0, 1)$ and $\beta \in (0, 1)$.
Step 0: Fix $z_0 \in \mathbb{R}^n$, and set $j = 0$.
Step 1: Compute $h_j := -\nabla g(z_j)$.
Step 2: Compute k_j, defined by

$$k_j \ := \ \min\left\{k \geq 0 \ : \ g(z_j+\beta^k h_j)-g(z_j) \leq -\alpha\beta^k ||h_j||^2\right\}. \tag{11}$$

Set $\lambda_j := \beta^{k_j}$.
Step 3: Set $z_{j+1} = z_j + \lambda_j h_j$, set $j = j + 1$, and go to Step 1.

Reference [14] presents convergence analysis and practical implementation details. The main convergence results are that a sequence of iteration points computed by the algorithm, $\{z_j\}_{j=1}^{\infty}$, converges to stationary points at a linear rate. Specifically, if $lim_{j\rightarrow\infty} z_j = \hat{z}$, then (i) $\nabla g(\hat{z}) = 0$; (ii) there exist constants $A > 0$ and $c \in (0, 1)$ such that, for all $j = 1, 2, \ldots$,

$$||z_j - \hat{z}|| \ \leq \ Ac^j; \tag{12}$$

and (iii) there exists a constant $\gamma \in (0, 1)$ such that, for all j large enough,

$$g(z_{j+1}) - g(\hat{z}) \ \leq \ \gamma\big(g(z_j) - g(\hat{z})\big). \tag{13}$$

Various extensions of the algorithm and its analysis to constrained problems can be found in [14].

Consider now the problem Π_t for a given $t \in [0, T)$. Given a switching-time vector $\bar{\tau}(t)$, we next define the feasible descent direction, $h(t)$, and a step size, $\lambda(t)$, that will be used in the algorithm presented later. $h(t)$ is defined as the projection of the vector $-\frac{\partial\tilde{J}}{\partial\bar{\tau}}(t, x(t), \bar{\tau}(t))$ onto the set $\Lambda_t - \{\bar{\tau}(t)\}$; an explicit formula for it is contained in Reference [15]. Note that if $\bar{\tau}$ is contained in the interior of Λ_t then $h(t) = -\frac{\partial\tilde{J}}{\partial\bar{\tau}}(t, x(t), \bar{\tau}(t))$. Regarding the step size $\lambda(t)$, we modify Equation (11) to ensure feasibility in the following way. Given constants $\alpha \in (0, 1)$ and $\beta \in (0, 1)$. Define $\mu(t) := \max\{\lambda \geq 0 : \bar{\tau}(t) + \lambda h(t) \in \Lambda_t\}$, and then define $\lambda_{max}(t) := \min\{\mu(t), 1\}$. This term, $\lambda_{max}(t)$, is the largest-possible step size to be considered in the step-size computation, and it ensures feasibility of the computed switching-time vectors. Define $k(t)$ by

$$\begin{aligned} k(t) \ : \ \min\Big\{k \geq 0 \ : \\ \tilde{J}\big(t, x(t), \bar{\tau}(t) + \lambda_{max}(t)\beta^k h(t)\big) - \tilde{J}\big(t, x(t), \bar{\tau}(t)\big) \\ \leq \ \alpha\lambda_{max}(t)\beta^k\Big\langle h(t), \frac{\partial\tilde{J}}{\partial\bar{\tau}}\big(t, x(t), \bar{\tau}(t)\big)\Big\rangle\Big\}, \end{aligned} \tag{14}$$

where $\langle\cdot, \cdot\rangle$ denoted inner product in $\mathbb{R}^{N-q(t)+1}$. Then define $\lambda(t) := \lambda_{max}(t)\beta^{k(t)}$.

These computations take a positive amount of time, and this must be reflected in the on-line algorithm presented below. A typical iteration at time t starts at $\bar{\tau}(t)$ and computes the next switching-time vector by following a gradient descent with Armijo step size. Assuming that the computation takes Δt seconds, the iteration is based on the formula $\bar{\tau}(t + \Delta t) = \bar{\tau}(t) + \lambda(t)h(t)$. Thus, the algorithm can compute only a finite number of iterations in the time-interval $[0, T]$.

Algorithm 2: Fix constants $\alpha \in (0, 1)$ and $\beta \in (0, 1)$.
Step 0: Define $t_0 := 0$, fix $\bar{\tau}(t_0) = 0$, and set $j = 0$.
Step 1: Compute $\frac{\partial\tilde{J}}{\partial\bar{\tau}}(t_j, x(t_j), \bar{\tau}(t_j))$.
Step 2: Compute $h(t_j)$, defined as the projection of -$\frac{\partial\tilde{J}}{\partial\bar{\tau}}(t_j, x(t_j), \bar{\tau}(t_j))$ onto the set $\Lambda_{t_j} - \{\bar{\tau}(t_j)\}$.
Step 3: Compute

$$\mu(t_j) \ := \ \max\{\lambda \geq 0 \ : \ \bar{\tau}(t_j) + \lambda h(t_j) \in \Lambda_{t_j}\}, \tag{15}$$

and compute $\lambda_{max}(t_j) := \min\{\mu(t_j), 1\}$.
Step 4: Compute $k(t_j)$, defined by

$$\begin{aligned} k(t_j) \ : \ \min\Big\{k \geq 0 \ : \\ \tilde{J}\big(t_j, x(t_j), \bar{\tau}(t_j) + \lambda_{max}(t_j)\beta^k h(t_j)\big) - \tilde{J}\big(t_j, x(t_j), \bar{\tau}(t_j)\big) \\ \leq \ \alpha\lambda_{max}(t_j)\beta^k\Big\langle h(t_j), \frac{\partial\tilde{J}}{\partial\bar{\tau}}\big(t_j, x(t_j), \bar{\tau}(t_j)\big)\Big\rangle\Big\}. \end{aligned} \tag{16}$$

Step 5: Set $\lambda(t_j) := \lambda_{max}(t_j)\beta^{k(t_j)}$ and set $\bar{\tau}(t_{j+1}) = \bar{\tau}(t_j) + \lambda(t_j)h(t_j)$. Set $t_{j+1} := t_j + \Delta t$. If $t_{j+1} \geq T$ then stop; otherwise set $j = j + 1$ and go to Step 1.

The algorithm's behavior cannot be described by asymptotic concepts since it computes but a finite number of points, and therefore we analyze it in terms of *convergence rate* by modifying equations (12) and (13).

To start with, let us define $J(t, x(t), \bar{\tau})$ to be the "true" cost to go, namely

$$J(t, x(t), \bar{\tau}) \ := \ \int_t^T L(x(s), s)ds, \tag{17}$$

where, for a given switching vector $\bar{\tau}$, $x(s)$ is the state trajectory of the system as defined in (3). The following result relates certain minimum points for $J(0, x_0, \cdot)$ to those for $J(t, x(t), \cdot)$.

Lemma 1: Let $\hat{\tau} := (\hat{\tau}_1, \ldots, \hat{\tau}_N)^T$ be a minimum point for $J(0, x_0, \cdot)$, and assume that $\hat{\tau}_N < T$. Let $\{x(t)\}$ be the state trajectory associated with $\hat{\tau}$. Suppose that the Hessian matrix $\frac{\partial^2 J}{\partial\bar{\tau}^2}(0, x_0, \hat{\tau})$ is positive definite. Then, for every $t \in [0, \bar{\tau}_N]$, (i) $\bar{\tau}$ is a minimum point for $J(t, x(t), \cdot)$, and (ii) there exist $m > 0$ and $M \geq m$ such that for every $t \in [0, \hat{\tau}_N]$, all the eigenvalues of $\frac{\partial^2 J}{\partial\bar{\tau}^2}(t, x(t), \hat{\tau})$ lie between m and M.

Proof: Please see [13]. ■

The source of errors between the real cost J and the simulated cost $\tilde{J}$ are the discrepancies between the dynamic response functions F and $\tilde{F}$, and between the performance functions L and $\tilde{L}$. These discrepancies can be quantified in terms of upper bounds on the L^1 norms of the error terms $\tilde{F} - F$ and $\tilde{L} - L$, and their derivatives, along the system's state trajectories, which translate into upper bounds on the L^∞ norms of the error function $\tilde{J} - J$ and its derivatives. The following analysis is carried out in terms of the latter bounds. Thus, for every $t \in [0,T)$, define $\varepsilon_0(t) := ||\tilde{J}(t,x(t),\cdot) - J(t,x(t),\cdot)||_{L^\infty}$ and $\varepsilon_1(t) := ||\frac{\partial \tilde{J}}{\partial \bar{\tau}}(t,x(t),\cdot) - \frac{\partial J}{\partial \bar{\tau}}(t,x(t),\cdot)||_{L^\infty}$ (where the L^∞ norm is with respect to the variable $\bar{\tau} \in \Lambda_t$), and define, for $\ell = 0,1$,

$$E_\ell(t) \ := \ \max\{\varepsilon_\ell(s) : s \in [t,T]\}. \qquad (18)$$

Our main results concerning the convergence rate of Algorithm 2 is along the lines of Equations (12) and (13). Let $\hat{\tau}$ be a solution point for Π, and recall that $\bar{\tau}(t_j)$ is the jth iteration point computed by the algorithm. Now $\bar{\tau}(t_j)$ and $\bar{\tau}(t_{j+1})$ may have different dimensions since the controlled variable consists only of future and present switching times, and to make a comparison meaningful, we will assume that they have the same dimension, i.e., $q(t_j) = q(t_{j+1})$. Thus, we will implicitly assume throughout the following discussion that for all time t and $\bar{\tau} := (\tau_1,\ldots,\tau_N)^T$ under consideration, $t < \tau_1$. Furthermore, the argument is greatly complicated when points $\bar{\tau}(t_j)$ lying on the boundary of the feasible set Λ are explicitly considered; therefore, and due to space limitations, we will consider only the case where $\bar{\tau}(t_j)$ is contained in the interior of Λ while deferring the general case to a later publication that permits more space.

The following assumption will be made.

Assumption 2: (i). For every $t \in [0,T)$ and for every state $x(t)$, the functions $J(t,x(t),\cdot)$ and $\tilde{J}(t,x(t),\cdot)$ are three-times continuously differentiable. (ii). There exists $K_0 > 0$ such that, for every $t \in [0,T)$, state $x(t)$, and $\bar{\tau} \in \Lambda$, $||\frac{\partial^2 J}{\partial \bar{\tau}^2}(t,x(t),\bar{\tau})|| \leq K_0$ and $||\frac{\partial^2 \tilde{J}}{\partial \bar{\tau}^2}(t,x(t),\bar{\tau})|| \leq K_0$.

Lemma 2: Let $\hat{z} \in \mathbb{R}^n$ be a local minimum for $g(\cdot)$, and suppose that $H(\hat{z})$ is positive definite. There exist $m > 0$, $M \geq m$, and $\epsilon > 0$ such that, if $z \in B(\hat{z},\epsilon)$, then the following two equations are in force,

$$\frac{1}{2}m||z-\hat{z}||^2 \ \leq \ g(z) - g(\hat{z}) \ \leq \ \frac{1}{2}M||z-\hat{z}||^2, \qquad (19)$$

and

$$0 \ \leq \ g(z) - g(\hat{z}) \ \leq \ \frac{1}{2m}||\nabla g(z)||^2. \qquad (20)$$

Proof: Please see Lemma 1.3.6 in [14] ■

The next result implies that the Armijo step size is bounded from below by a positive quantity that depends only on an upper bound on the second derivative of $g(\cdot)$, but not on any other particular feature of $g(\cdot)$.

Lemma 3: Let $\Omega \subset \mathbb{R}^n$ be a convex, open set containing $\hat{z}$, and suppose that there exists $K_0 > 0$ such that, for every $z \in \Omega$, $||H(z)|| \leq K_0$. If $z_j \in \Omega$ and $z_j - \nabla g(z_j) \in \Omega$ for some $j = 1,2,\ldots$, then

$$\lambda_j \ \geq \ \frac{2}{K_0}\beta(1-\alpha), \qquad (21)$$

where λ_j is the step size computed in Step 2 of Algorithm 1, and β and α are the constants set for that algorithm.

Proof: Please see [13] ■

The following is the main result of the paper.

Proposition 1: Suppose that Assumption 2 is in force. Let $\hat{\tau}$ be a solution point for Π lying in the interior of Λ, and suppose that the matrix $\frac{\partial^2 J}{\partial \bar{\tau}^2}(0,x_0,\hat{\tau})$ is positive definite. Then there exist $\epsilon > 0$, $\gamma \in (0,1)$, $c \in (0,1)$, and constants $K_1 > 0$, $K_2 > 0$, $K_3 > 0$, and $K_4 > 0$, such that, if $\bar{\tau}(t_j) \in B(\hat{\tau},\epsilon)$, and $\bar{\tau}(t_{j+1}) \in B(\hat{\tau},\epsilon)$; $\lambda_{max}(t_j) = 1$ (see Step 3); and $t_{j+1} < \hat{\tau}_1 - \epsilon$, then,

$$\begin{aligned} J\big(t_{j+1},x(t_{j+1}),\bar{\tau}(t_{j+1})\big) - J\big(t_{j+1},x(t_{j+1}),\hat{\tau}\big) \ \leq \\ \gamma\Big(J\big(t_j,x(t_j),\bar{\tau}(t_j)\big) - J\big(t_j,x(t_j),\hat{\tau})\big)\Big) + \\ 2E_0(t_j) + K_1E_1(t_j). \end{aligned} \qquad (22)$$

Furthermore, if $\bar{\tau}(t_j) \in B(\hat{\tau},\epsilon)$ for all $j = j_0, j_0+1,\ldots$, for some $j_0 \geq 0$, then

$$||\bar{\tau}(t_j) - \hat{\tau}|| \ \leq \ K_2c^{j-j_0} + K_3\big(E_0(t_j)\big)^{1/2} + K_4\big(E_1(t_j)\big)^{1/2}. \qquad (23)$$

Proof: For every $\epsilon > 0$, if $t_{j+1} < \hat{\tau}_1 - \epsilon$, $||\bar{\tau}(t_{j+1}) - \hat{\tau}|| < \epsilon$, and $||\bar{\tau}(t_j) - \hat{\tau}|| < \epsilon$, then simple algebra yields that $\tau_1(t_{j+1}) > \hat{\tau}_1 - \epsilon$ and $t_{j+1} < \tau_1(t_{j+1})$, and likewise, $\tau_1(t_j) > \hat{\tau}_1 - \epsilon$ and $t_j < \tau_1(t_j)$. This implies that, when the real system evolves with either one of the switching times $\bar{\tau}(t_j)$, $\bar{\tau}(t_{j+1})$, or $\hat{\tau}$, at all time $t \leq t_{j+1}$, all of the switching times are in the future. Consequently (see (3)), the state variables of these respective systems are identical for all $t \leq t_{j+1}$, and hence. and by (4),

$$\begin{aligned} J\big(t_{j+1},x(t_{j+1}),\bar{\tau}(t_{j+1})\big) - J\big(t_{j+1},x(t_{j+1}),\hat{\tau}\big) \ = \\ J\big(t_j,x(t_j),\bar{\tau}(t_{j+1})\big) - J\big(t_j,x(t_j),\hat{\tau}\big). \end{aligned} \qquad (24)$$

Consequently, and by (17),

$$\begin{aligned} J\big(t_{j+1},x(t_{j+1}),\bar{\tau}(t_{j+1})\big) - J\big(t_{j+1},x(t_{j+1}),\hat{\tau}\big) \ \leq \\ \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_{j+1})\big) - J\big(t_j,x(t_j),\hat{\tau}\big) + E_0(t_j). \end{aligned} \qquad (25)$$

Furthermore, by subtracting and adding $\tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_j)\big) + J\big(t_j,x(t_j),\bar{\tau}(t_j)\big)$ to the difference term in the RHS of (25),

$$\begin{aligned} \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_{j+1})\big) - J\big(t_j,x(t_j),\hat{\tau}\big) \ \leq \\ \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_{j+1})\big) - \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_j)\big) + \\ J\big(t_j,x(t_j),\bar{\tau}(t_j)\big) - J\big(t_j,x(t_j),\hat{\tau})\big) + E_0(t_j). \end{aligned} \qquad (26)$$

By Lemma 3 and Assumption 2, there exists $\bar{\lambda} > 0$ such that (see Steps 4-5),

$$\begin{aligned} \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_{j+1}\big) - \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_j)\big) \ < \\ -\alpha\bar{\lambda}||\frac{\partial \tilde{J}}{\partial \bar{\tau}}\big(t_j,x(t_j),\bar{\tau}(t_j)\big)||^2. \end{aligned} \qquad (27)$$

By Lemma 2 and Assumption 2 there exist $\epsilon > 0$, $B_1 > 0$, and $B_2 > 0$ such that, if $||\bar{\tau}(t_j) - \hat{\tau}|| < \epsilon$ then

$$\begin{aligned} ||\frac{\partial J}{\partial \bar{\tau}}\big(t_j,x(t_j),\bar{\tau}(t_j)\big)||^2 \ \geq \\ B_1\Big(J\big(t_j,x(t_j),\bar{\tau}(t_j)\big) - J\big(t_j,x(t_j),\hat{\tau}\big)\Big), \end{aligned} \qquad (28)$$

and

$$\begin{aligned} \Big| \ ||\frac{\partial \tilde{J}}{\partial \bar{\tau}}\big(t_j,x(t_j),\bar{\tau}(t_j)\big)||^2 - ||\frac{\partial J}{\partial \bar{\tau}}\big(t_j,x(t_j),\bar{\tau}(t_j)\big)||^2 \ \Big| \\ \leq \ B_2E_1(t_j). \end{aligned} \qquad (29)$$

Finally, by (17),

$$\begin{aligned} \tilde{J}\big(t_j,x(t_j),\bar{\tau}(t_j)\big) - \tilde{J}\big(t_j,x(t_j),\hat{\tau}\big) \ \leq \\ J\big(t_j,x(t_j),\bar{\tau}(t_j)\big) - J\big(t_j,x(t_j),\hat{\tau}\big) + 2E_0(t_j). \end{aligned} \qquad (30)$$

Now putting together Equations (25) - (30) we obtain,

$$J\big(t_{j+1}, x(t_{j+1}), \bar{\tau}(t_{j+1})\big) - J(t_{j+1}, x(t_{j+1}), \hat{\tau}\big) \leq (1-\alpha\bar{\lambda}B_1)\Big(J\big(t_j, x(t_j), \bar{\tau}(t_j)\big) - J(t_j, x(t_j), \hat{\tau}\big)\Big) + +2E_0(t_j) + \alpha\bar{\lambda}B_2E_1(t_j). \quad (31)$$

Define $K_1 := \alpha\bar{\lambda}B_2$. Define $\gamma := 1-\alpha\bar{\lambda}B_1$ if $1-\alpha\bar{\lambda}B_1 > 0$, and otherwise choose any $\gamma \in (0,1)$. Since $J\big(t_{j+1}, x(t_{j+1}), \bar{\tau}(t_{j+1})\big) - J(t_{j+1}, x(t_{j+1}), \hat{\tau}\big) \geq 0$ and $J\big(t_j, x(t_j), \bar{\tau}(t_j)\big) - J(t_j, x(t_j), \hat{\tau}\big)) \geq 0$, it follows that (22) is satisfied with these values of K_1 and γ.

Next, suppose that $||\bar{\tau}(t_j) - \hat{\tau}|| < \epsilon$ for all $j \geq j_0$ for some $j_0 \geq 0$. Define $A := J\big(t_{j_0}, x(t_{j_0}), \bar{\tau}(t_{j_0})\big) - J\big(t_{j_0}, x(t_{j_0}), \hat{\tau}\big)$. By (24), for all $j \geq j_0$,

$$J\big(t_j, x(t_j), \bar{\tau}(t_j)\big) - J\big(t_j, x(t_j), \hat{\tau}\big) \leq A\gamma^{j-j_0} + \frac{1}{1-\gamma}\big(2E_0(t_j) + K_1E_1(t_j)\big). \quad (32)$$

By Lemma 2 (Equation (19)) we can assume (by reducing ϵ if necessary) that there exists $B > 0$ such that, for all $j \geq j_0$,

$$||\bar{\tau}(t_j) - \hat{\tau}||^2 \leq B\Big(J\big(t_j, x(t_j), \bar{\tau}(t_j)\big) - J\big(t_j, x(t_j), \hat{\tau}\big)\Big). \quad (33)$$

Define $c := \sqrt{\gamma}$, $K_2 := (AB)^{1/2}$, $K_3 := \left(\frac{2B}{1-\gamma}\right)^{1/2}$, and $K_4 := \left(\frac{K_1B}{1-\gamma}\right)^{1/2}$. By (33) and (32), with a bit of algebra, Equation (23) follows. ■

Observe that each one of the bounds in Equations () and () contains two terms: The first term characterizes linear convergence rate when the functions and their gradients could be computed exactly, and the last respective terms are due to the estimation errors.

IV. Simulation Example

This section considers the scenario of a mobile robot tracking a moving target while avoiding two obstacles along the way. The robot does not know the future planned trajectory of the target. However, it can sense the target's position and velocity at each time t, and it uses this information to estimate the future trajectory of the target via linear interpolation.

Let $x_R(t) \in \mathbb{R}^2$ and be the position (location) of the mobile robot and $x_G(t) \in \mathbb{R}^2$ be the location of the target at each time $t \in [0,T]$. Let $x_\Phi, x_\Psi \in \mathbb{R}^2$ be the fixed positions of Obstacle 1 and Obstacle 2, respectively. Furthermore, at each time $t \in [0,T]$, the estimated position of the target at future times $s \geq t$ is defined via the following linear interpolation,

$$\tilde{x}_G(s,t,x_G(t)) := x_G(t) + \dot{x}_G(t)(s-t). \quad (34)$$

The state of the robot is given by $x(t) = (x_R(t)^T \;\; x_3(t))^T$, where $x_3(t) \in [0, 2\pi)$ denotes its heading angle. We assume that it moves at a constant speed $\bar{v}$ unless $||x_R - x_G|| < r$ for a given (small) $r > 0$, in which case its speed is $\frac{\bar{v}}{r}||x_R - x_G||$. Thus, denoting the robot's actual speed by v, its dynamic equation of motion is

$$\dot{x}(t) = \begin{bmatrix} v\,cos\,(x_3(t)) \\ v\,sin\,(x_3(t)) \\ u - x_3(t) \end{bmatrix}. \quad (35)$$

Note that the third coordinate of this equation steers the heading angle $x_3(t)$ towards a desired heading u. The control u depends on the robot's mode of operation, and there are three modes: Go to Goal, Avoid Obstacle 1, and Avoid Obstacle 2, henceforth denoted by *G2G*, *Avoid1*, and *Avoid2*, respectively. In the G2G mode the angle u is defined by

$$u = tan^{-1}\left(\frac{x_{G,2}(t) - x_{R,2}(t)}{x_{G,1}(t) - x_{R,1}(t)}\right), \quad (36)$$

where the subscript indices 1 and 2 indicate the first and second elements of the vector, respectively. It can be seen by Eq. (35) that the robot's heading angle approaches the direction of the goal in this mode. For *Avoid1*, define the control as

$$u = \begin{cases} \phi_\Phi - \frac{\pi}{2}, & if\ \phi_\Phi - x_3(t) \geq 0 \\ \phi_\Phi + \frac{\pi}{2}, & if\ \phi_\Phi - x_3(t) < 0 \end{cases}, \quad (37)$$

where $\phi_\Phi = tan^{-1}\left(\frac{x_{\Phi,2} - x_{R,2}(t)}{x_{\Phi,1} - x_{R,1}(t)}\right)$. In this mode, the robot circumvents the obstacle counterclockwise or clockwise according to the sign of the angle $\phi_\Phi - x_3(t)$. A similar equation holds for the *Avoid2* mode except that the subscript Φ is replaced by Ψ.

The problem Π that we consider is to compute the switching times between modes in a given sequence that minimize the integral of a performance function $L(x(t), t)$ which penalizes proximity to the obstacles and distance from the goal. We chose the performance function to be

$$\begin{aligned} L(x(t), t) &= \rho\,||x_R(t) - x_G(t)||^2 \\ &\quad + \alpha_1\, exp\left(-\frac{||x_R(t) - x_\Phi||^2}{\beta_1}\right) \\ &\quad + \alpha_2\, exp\left(-\frac{||x_R(t) - x_\Psi||^2}{\beta_2}\right), \end{aligned} \quad (38)$$

for some given $\rho, \alpha_1, \alpha_2, \beta_1, \beta_2 > 0$. However, in the real-time scenario of this paper the robot does not know the planned trajectory of the goal, and in the *G2G* mode it attempts to approach $\tilde{x}_G(s,t,x_G(t))$ as defined by (34). Let us denote the future simulated state of the robot from time t onward by $\tilde{x}(s,t,x(t),\bar{\tau}(t))$, which is defined by the following equation,

$$\frac{\partial}{\partial s}\tilde{x}(s,t,x(t),\bar{\tau}(t)) = \begin{bmatrix} v\,cos\,(\tilde{x}_3(s,t,x(t),\bar{\tau}(t))) \\ v\,sin\,(\tilde{x}_3(s,t,x(t),\bar{\tau}(t))) \\ u - \tilde{x}_3(s,t,x(t),\bar{\tau}(t)) \end{bmatrix}, \quad (39)$$

where the speed v is the constant $\bar{v}$ as long as $||\tilde{x}(s,t,x(t),\bar{\tau}(t)) - \tilde{x}_G(s,t,x_G(t))|| \geq r$, and $\frac{\bar{v}}{r}||\tilde{x}(s,t,x(t),\bar{\tau}(t)) - \tilde{x}_G(s,t,x_G(t))||$ if $||\tilde{x}(s,t,x(t),\bar{\tau}(t)) - \tilde{x}_G(s,t,x_G(t))|| < r$. The control u is defined via Equations (36)-(37) except that $x_R(t)$ and $x_G(t)$ are replaced by $\tilde{x}_R(s,t,x(t),\bar{\tau}(t))$ and $\tilde{x}_G(s,t,x_G(g))$,

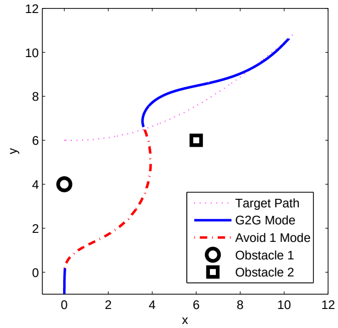

Fig. 1. Robot Trajectory

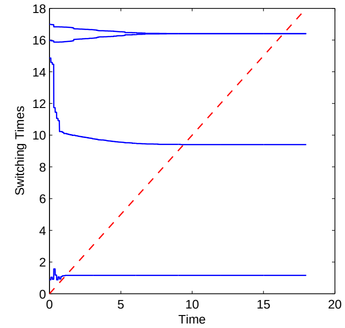

Fig. 2. Evolution of Switching Times

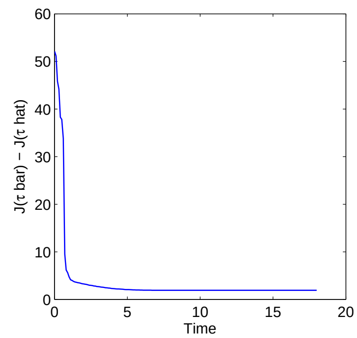

Fig. 3. Cost Functional

respectively, and similar replacements to (38) define the simulated performance function $\tilde{L}\left(\tilde{x}\left(s,t,x\left(t\right),\bar{\tau}\left(t\right)\right),s,t\right)$.

Consider the scenario shown in Figure 1, where the positions of the obstacles 1 and 2 are indicated by the circle and square, respectively, and the trajectory of the target is shown by the dotted curve. Initially $x(0)=(0,-1,\frac{\pi}{2})^T$, and we made the (reasonable) choice of the mode-sequence to be $\{G2G, Avoid1, G2G, Avoid2, G2G\}$. We ran Algorithm 2 with the initial switching-time vector $\bar{\tau}(0)=(1\ 15\ 16\ 17)^T$. The final time is $T=18$ and the iteration-computation time is $\Delta t=0.1$, so the algorithm ran for 180 iterations. We chose the values $\alpha=\beta=0.5$ for the Armijo step size, $\bar{v}=1$, $r=0.5$, and for Equation (38), $\rho=0.1$, $\alpha_1=\alpha_2=500$, and $\beta_1=\beta_2=0.8$.

Figure 2 shows the graphs of the switching times $\tau_k(t)$, $k=1,2,3,4$, for a run of the algorithm. The diagonal line in the figure represents the real time, and once a switching time crosses it it "freezes", namely remains a constant thereafter. We can see that τ_1 freezes early, at about $t=1.0$, while the remaining switching times have enough time to stabilize before freezing. We note that τ_3 and τ_4 coalesce at about $t=7$; at this point the descent direction in Algorithm 2 becomes the projected gradient as defined in Step 2. This coalescence eliminates the $4th$ mode, namely *Avoid2*, from the system's state, and the mode-sequence becomes $\{G2G, Avoid1, G2G\}$.

We also computed ("off-line") a local-minimum point for J, $\hat{\tau}=(1.5233, 9.0680, 16.4365, 16.8769)^T$, where $J(\hat{\tau})=22.8588$. Figure 3 shows the graph of $J(\bar{\tau}(t))-J(\hat{\tau}(t))$ which, not surprisingly, is monotone decreasing. The fact that it does not go to 0 is due to the early freezing of τ_1 which results in a positive estimation error at all future times.

REFERENCES

[1] X. Xu and P. Antsaklis, "Optimal Control of Switched Autonomous Systems", in *Proc. IEEE Conference on Decision and Control*, Las Vegas, Nevada, December 10-13, 2002.

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

[3] B. Piccoli, "Hybrid Systems and Optimal Control", *Proc. IEEE Conference on Decision and Control*, Tampa, Florida, December 16-18, pp. 13-18, 1998.

[4] H.J. Sussmann, "Set-Valued Differentials and the Hybrid Maximum Principle", in *Proc. IEEE Conf. on Decision and Control*, Sydney, Australia, December 12-15, pp. 558–563, 2000.

[5] M.S. Shaikh and P. Caines, "On Trajectory Optimization for Hybrid Systems: Theory and Algorithms for Fixed Schedules", in *Proc. IEEE Conf. on Decision and Control*, Las Vegas, Nevada, December 10-13, pp. 1997–1998, 2002.

[6] S. Hedlund and A. Rantzer, "Optimal Control of Hybrid Systems", in *Proc. IEEE Conf. on Decision and Control*, Phoenix, Arizona, December 7-10, pp. 3972-3977, 1999.

[7] X. Xu and P.J. Antsaklis, "Optimal Control of Switched Systems via Nonlinear Optimization Based on Direct Differentiations of Value Functions", *Int. J. of Control*, Vol. 75, pp. 1406-1426, 2002.

[8] P. Caines and M.S. Shaikh, "Optimality Zone Algorithms for Hybrid Systems Computation and Control: Exponential to Linear Complexity", in *Proc. 13th Mediterranean Conference on Control and Automation*, Limassol, Cyprus, pp. 1292-1297, June 27-29, 2005.

[9] S.A. Attia, M. Alamir, and C. Canudas de Wit, "Sub Optimal Control of Switched Nonlinear Systems Under Location and Switching Constraints", in *Proc. 16th IFAC World Congress*, Prague, the Czech Republic, July 3-8, 2005.

[10] S. Azuma, M. Egerstedt, and Y. Wardi, "Output-Based Optimal Timing Control of Switched Systems", in *Proc. Hybrid Systems: Computation and Control*, Springer-Verlag, pp. 64-78, Santa Barbara, CA, March 2006.

[11] X.C. Ding, Y. Wardi, and M. Egerstedt, "On-Line Optimization of Switched-Mode Dynamical Systems", *IEEE Transactions on Automatic Control*, Vol. 54, No. 9, pp. 2266-2271, 2009.

[12] X.C. Ding, Y. Wardi, and M. Egerstedt, "On-line Adaptive Optimal Timing Control of Switched Systems", in *Proc. IEEE CDC*, Shanghai, China, Dec. 16-18, 2009.

[13] Y. Wardi, P. Twu, and M. Egerstedt, "On-line Optimal Timing Control of Switched Systems", Technical Memorandum, *http://www.ece.gatech.edu/∼magnus/OnlineOptimalTiming.pdf*.

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

[15] H. Axelsson, M. Egerstedt, G. Vachtsevanos, and Y. Wardi, "Algorithms for Switching-Time Optimization in Hybrid Systems", in *Proc. 13th Mediterranean Conference on Control and Automation*, Limassol, Cyprus, June 27-29, 2005.