ON-LINE OPTIMIZATION OF SWITCHED-MODE DYNAMICAL SYSTEMS[1]

X.C. Ding, Y. Wardi[2], and M. Egerstedt

Abstract

This paper considers an optimization problem in the setting of switched-mode hybrid dynamical systems, where the control variable (independent variable) consists of the mode-switching times, and the performance criterion is a cost functional defined on the system's state trajectory. The system is deterministic, nonlinear, and autonomous, and its state variable cannot be measured and hence it has to be estimated. We propose an on-line, Newton-like optimization algorithm that recomputes the control variable by attempting to optimize the cost-to-go at equally-spaced epochs. The main result concerns the algorithm's convergence rate, which can vary from sublinear to quadratic depending on its computing rate and the state estimation error.

I. Introduction

Consider a switched-mode dynamical system having the form

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

where $x \in R^n$ is the state variable; τ_i, $i = 0, \ldots, N+1$ is a monotone-nondecreasing set of time-points; and $f_i : R^n \rightarrow R^n$ are given functions. We further assume that $\tau_0 = 0$, $\tau_{N+1} = T$ for a given final time $T > 0$, the functions f_i, $i = 1, \ldots, N+1$ are C^∞ (namely, having continuous derivatives of all orders), the initial state $x_0 := x(0)$ is given, and the state trajectory $\{x(t)\}$, $t \in [0, T]$, is continuous, so that $x(\tau_i^+) = x(\tau_i^-)$ for all $i = 1, \ldots, N$. The functions f_i correspond to various modes of the system and hence are called the *modal functions*. Let $L : R^n \rightarrow R$ be a C^∞ function, and define the cost functional J by

$$J \;:=\; \int_0^T L(x)dt. \tag{2}$$

The optimization problem of minimizing J as a function of the mode-switching times τ_i, $i = 1, \ldots, N$, has been investigated extensively. Reference [6] derived a variant of the maximum

[1] Research supported in part by the National Science Foundation under Grant #0509064.

[2] Corresponding author.

principle in a general setting of optimal control. Later, [3] defined a general framework for optimal control in switched-mode hybrid systems, where the modal functions f_i have a control input u, and the cost functional J is a function of the switching times as well as the control input. Subsequently, [8], [12], [10], [11] derived variants of the maximum principle, and [14], [15], [10], [4], [1], [5], [2] investigated various optimization algorithms. References [14], [15], [5] considered the simpler optimization problem defined by (1) and (2), where the absence of the input control u simplifies the analysis while capturing the salient features of switched-mode systems. Labeled the *timing optimization problem*, it arises in a variety of application domains; see [5] for a survey. A more difficult problem, called the *scheduling optimization problem*, has its variable comprised of the schedule of modes, namely the functions f_i and the switching times among them; several algorithmic approaches for this problem have been proposed and analyzed in [4], [1], [2].

This paper focuses on an on-line variant of the timing optimization problem. It considers an abstract setting where the state variable $x(t)$ cannot be measured or computed at time t, and it has to be estimated by a suitable state observer.[3] Correspondingly, our focus is not on minimizing the cost functional J defined by (2), but rather on the problem of minimizing the cost-to-go at each time $t \in [0, T]$, given the state estimator at that time. This problem requires a real-time algorithm, and to this end we propose a variant of the Newton-Raphson method. The main contribution of the paper is in its convergence analysis under real-time conditions. Specifically, assuming that computing times cannot be neglected, the algorithm cannot compute an infinite iteration-sequence during a given finite-length interval, and therefore the standard asymptotic notions of convergence cannot be applied. Instead, we propose an alternative concept for characterizing the algorithm's approach towards a minimum, which is based on convergence rates. The main result is that, if the state estimation error is small, then, the rate at which the algorithm approaches a minimum tends to be quadratic if its computation is fast, and linear if its computation is slow.

The rest of the paper is organized as follows. Section II defines the real-time, timing optimization problem, and Section III describes our algorithm and its approach towards (local)

[3] The paper focuses on optimization algorithms under conditions of state-estimation errors. It considers state observers only as a cause for such errors, but the discussion and analysis pertain to other state estimation techniques. For this reason it does not specify the state-estimation algorithm, nor does it address the question of stable hybrid observers.

solutions. Section IV presents numerical results and suggests possible extensions of the algorithm to the scheduling optimization problem, and Section V concludes the paper. We mention that a preliminary version of the paper has been published in the Proceedings of the 46th CDC [13].

II. Problem Formulation

Consider the timing optimization problem as defined by Equations (1) and (2), and let us denote the switching times by the vector notation $\bar{\tau} := (\tau_1, \ldots, \tau_N)^T \in R^N$. Suppose that the initial state x_0 is fixed, so that the state trajectory $\{x(t)\}_{t \in [0,T]}$ is determined by $\bar{\tau}$. Moreover, the cost functional J is a function of $\bar{\tau}$ as well, and hence will be denoted by $J(\bar{\tau})$. Thus, the timing optimization problem is essentially the nonlinear programming problem of minimizing $J(\bar{\tau})$, subject to the constraint $\bar{\tau} \in \Phi$, where the feasible set Φ is defined as

$$\Phi \;=\; \{\bar{\tau} \in R^N \;:\; 0 \leq \tau_1 \leq \ldots \leq \tau_N \leq T\}. \tag{3}$$

We point out that formulas for the gradient $\nabla J(\bar{\tau})$ have been derived based on variational principles, and used in nonlinear programming algorithms for computing Kuhn-Tucker points for the optimization problem [6], [14], [10], [5].

Suppose now that x_0 is not known, and the state trajectory $\{x(t)\}$ has to estimated by a suitable state observer, like the Luenberger observer in the case of linear systems, or the Moraal-Grizzle observer [7] in the nonlinear case. Denote the resulting state estimator by $\hat{x}(t)$, and let $\Pi_{t,\hat{x}(t)}$ denote the problem of optimizing the cost-to-go at time t with the initial state $\hat{x}(t)$. Formally, $\Pi_{t,\hat{x}(t)}$ is defined as follows. Given a switching-times vector $\bar{\tau} := (\tau_1, \ldots, \tau_N)^T \in \Phi$ such that $\tau_1 \geq t$, consider the equation

$$\dot{\tilde{x}}(\xi) \;=\; f_i(\tilde{x}(\xi)), \qquad \xi \in [\tau_{i-1}, \tau_i), \qquad i = 1, \ldots, N+1, \tag{4}$$

on the time-interval $\xi \in [t, T]$, with the initial condition $\tilde{x}(t) = \hat{x}(t)$. Define the cost-to-go performance functional, denoted by $J(t, \hat{x}(t), \bar{\tau})$, by

$$J(t, \hat{x}(t), \bar{\tau}) \;:=\; \int_t^T L(\tilde{x}(\xi)) d\xi. \tag{5}$$

The problem $\Pi_{t,\hat{x}(t)}$ is to minimize $J(t, \hat{x}(t), \bar{\tau})$ over $\bar{\tau} \in \Phi(t)$, where the constraint set $\Phi(t)$ is defined by

$$\Phi(t) \;=\; \{\bar{\tau} \in R^N \;:\; t \leq \tau_1 \leq \ldots \leq \tau_N \leq T\}. \tag{6}$$

Observe the difference of notation between the variables x, $\hat{x}$, and $\tilde{x}$. $x(t)$ is the state of the system as defined in (1) with the (unknown) initial condition (state) $x_0 = x(0)$. $\hat{x}(t)$ is the state estimator, defined via the dynamic equation (1) with the initial condition $\hat{x}(0)$. Finally, the variable $\tilde{x}(\xi)$ is the trajectory defined by Equation (4) for $\xi \geq t$, with the initial condition $\tilde{x}(t) = \hat{x}(t)$. Note that, if $\hat{x}(0) = x_0$ then $\hat{x}(t) = x(t)$ for all $t \geq 0$, and hence, for every $t \in [0, T]$ and $\xi \geq t$, $\tilde{x}(\xi) = x(\xi)$.

The algorithm that we will later describe is iterative, and it computes a sequence of switching-time vectors $\bar{\tau}$. Since we are concerned with its real-time aspects, we denote by $\bar{\tau}(t) := (\tau_1(t), \ldots, \tau_N(t))^T$ the variable $\bar{\tau}$ it has computed at time t. At this time it performs a computation of the next iteration point by using available information, including the state estimator $\hat{x}(t)$. Let us consider, for a moment, the conceptual case where that computation is instantaneous, since it serves to clarify some of the salient points of the analysis; later we will consider the realistic, implementable case requiring positive computing times. In the conceptual setting, the algorithm is underscored by the following three processes: the time process $\{t\}_{t=0}^{T}$, the state-estimator process $\{\hat{x}(t)\}_{t=0}^{T}$, and the optimization process $\{\bar{\tau}(t)\}_{t=0}^{T}$. The first process, $\{t\}$, acts as a temporal platform for computation, that is, at any time t a certain computation may be performed. The state-estimator process consists of the output of the observer subsystem, and the optimization process acts to minimize the cost-to-go at each time t. Formally, suppose that at time t the algorithm performs a computation of the form

$$\bar{\tau}(t + dt) \;=\; \bar{\tau}(t) + h(t), \tag{7}$$

for a suitable vector $h(t) \in R^N$, and this computation is completed by the time $t + dt$. We point out that whereas the processes $\{t\}$ and $\{\hat{x}(t)\}$ are continuous, the process $\{\bar{\tau}(t)\}$ is not meant to be continuous at any time t.

We observe that due to its real-time nature, the algorithm cannot modify past switching times. This means that, if $\tau_j(t) < t$ for some $j = 1, 2, \ldots$, then the jth switching has already occurred in the past, and hence it cannot be modified by the algorithm at time t. Starting at time $t = 0$, the variable of the algorithm is $\bar{\tau}(t) = (\tau_1(t), \ldots, \tau_N(t))^T$. However, if at some time $t_1 > 0$ it happens that $\tau_1(t_1) \leq t_1$, then from this time onward $\tau_1(t)$ will remain fixed, and the dimension of the variable parameter will be reduced. Furthermore, if this happens early enough then the fixed value of $\tau_1(t)$ can be far off from the optimal value. Later (in Section IV.B) we describe an ad-hoc way to circumvent this problem by inserting new modes from time to time, but now we continue the discussion and carry out the analysis under the condition that $\tau_1(t) > t$.

III. Newton-Raphson Method for Optimizing the Cost-to-Go

In order to guarantee the smoothness properties required by the Newton-Raphson method we make the following assumption.

Assumption 3.1: The functions f_i, $i = 1, \ldots, N+1$, and L, are C^∞.

The following result is an immediate corollary which can be proved by standard arguments from the theory of differential equations (see, e.g., the appendix of [9] for such arguments), and hence its proof is omitted.

Corollary 3.1: The function $J(t, \hat{x}, \bar{\tau})$ is C^∞.

Throughout the discussion in this section we assume that $\bar{\tau}(t) \in \Phi(t)$, and that $\tau_1(t) > t$ for all $t \in [0, T]$; this guarantees that no switching times become a part of the past, and hence they all can be modified by the optimization process. The algorithm that we consider is based on (7), where $h(t)$ is defined in the following way. Define the $N \times N$ matrix $H(t)$ by $H(t) = \frac{\partial^2 J}{\partial \tau^2}\big(t, \hat{x}(t), \bar{\tau}(t)\big)$, and suppose that $H(t)$ is positive definite (otherwise it makes no sense to consider Newton-like algorithms). According to the Newton-Raphsom method the direction vector from $\bar{\tau}(t)$ would be $\tilde{g}(t) := -H(t)^{-1}\frac{\partial J}{\partial \tau}(t+dt, \hat{x}(t+dt), \bar{\tau}(t))$, but since our problem has constraints, we define the direction vector, denoted by $\tilde{h}(t)$, to be the projection of $\tilde{g}(t)$ onto the set $\Phi(t) - \{\bar{\tau}(t)\}$. Note that $\tilde{h}(t)$ is a feasible direction from $\bar{\tau}(t)$, and if $\bar{\tau}(t)$ lies in the interior of $\Phi(t)$ then $\tilde{h}(t) = \tilde{g}(t)$. Now setting $h(t) = \tilde{h}(t)$ in (7) may overshoot the feasible set, and hence we define the step size $\gamma(t)$ by $\gamma(t) := \max\{\gamma \leq 1 : \bar{\tau}(t) + \gamma\tilde{h}(t) \in \Phi(t)\}$, and we set $h(t)$ to be $h(t) = \gamma(t)\tilde{h}(t)$. We remark that the computations of the projection of $\tilde{g}(t)$ onto the set $\Phi(t) - \{\bar{\tau}(t)\}$, and of $\gamma(t)$, are quite simple due to the specific structure of the constraint set $\Phi(t)$ (6).

In the theory of nonlinear programming, convergence of algorithms usually is proved for local minima, or for stationary or Kuhn-Tucker points. In our case, stationarity of a point $\bar{\tau}$ for the problem defined by Equations (1) and (2) means that $\frac{\partial J}{\partial \tau}(0, x_0, \bar{\tau}) = 0$, where $x(t)$ is the "true" state of the system and $x_0 = x(0)$. More generally, stationarity of a point $\bar{\tau} \in R^N$ for the problem $\Pi_{t,x(t)}$ or $\Pi_{t,\hat{x}(t)}$ means that $\frac{\partial J}{\partial \tau}(t, x(t), \bar{\tau}) = 0$ or $\frac{\partial J}{\partial \tau}(t, \hat{x}(t), \bar{\tau}) = 0$, respectively. Note that the optimization process (7) with $h(t)$ as defined above, computes one iteration of a variant of the Newton-Raphson method as applied to the problem $\Pi_{t,\hat{x}(t)}$, and this approach is justified by the following result.

Proposition 3.1: Suppose that Assumption 3.1 is in force; for every $t \in [0,T)$, $H(t)$ is positive definite; $-H(t)^{-1}\frac{\partial J}{\partial \tau}(t+dt, \hat{x}(t+dt), \bar{\tau}(t))$ lies in the interior of the set $\Phi(t+dt) - \{\bar{\tau}(t)\}$; $\gamma(t) = 1$ and hence

$$\bar{\tau}(t+dt) \; = \; \bar{\tau}(t) - H(t)^{-1}\frac{\partial J}{\partial \tau}(t+dt, \hat{x}(t+dt), \bar{\tau}(t)); \tag{8}$$

and $\bar{\tau}(t+dt)$ lies in the interior of the set $\Phi(t+dt)$. Then, if $\bar{\tau}(0)$ is a stationary point for $\Pi_{0,\hat{x}(0)}$, then for every $t \in [0,T]$, $\bar{\tau}(t)$ is a stationary point for $\Pi_{t,\hat{x}(t)}$.

Proof: Immediate by Bellman's optimality principle. ■

We mention that an extension of this result to the case involving Kuhn-Tucker points is straightforward. However, in order to keep the discussion simple, and without loss of generality, we will focus it on stationary points and the corresponding case where $\bar{\tau}(t)$ lies in the interior of the feasible set.

In order to turn the optimization process defined by (7) into an algorithm, we must consider the fact that it takes a positive amount of time to compute $h(t)$, and hence replace dt by a finite quantity $\Delta t > 0$. This results in the following procedure for computing $\bar{\tau}(t+\Delta t)$ from $\bar{\tau}(t)$.

Algorithm 3.1: Step 1. Compute $\tilde{h}(t)$, defined as the projection of $-H(t)^{-1}\frac{\partial J}{\partial \tau}(t+\Delta t, \hat{x}(t+\Delta t), \bar{\tau}(t))$ onto the set $\Phi(t+\Delta t) - \{\bar{\tau}(t)\}$.

Step 2. Compute $\gamma(t) := \max\{\gamma \leq 1 \; : \; \bar{\tau}(t) + \gamma\tilde{h}(t) \in \Phi(t+\Delta t)\}$.

Step 3. Define $h(t) := \gamma(t)\tilde{h}(t)$, and set

$$\bar{\tau}(t+\Delta t) \; := \; \bar{\tau}(t) + h(t). \tag{9}$$

The choice of Δt comprises a balance: on one hand, a single iteration of the algorithm should be computable within Δt seconds, and on the other hand, smaller values of Δt would result in faster convergence. This point will be made clear in the forthcoming analysis, which concerns the algorithm's convergence rate in terms of Δt and the estimation error $e(t)$.

Asymptotic convergence is a minimal requirement of an algorithm, and typically it means that every accumulation point of an iteration-sequence computed by the algorithm satisfies a suitable optimality condition, like being stationary or a Kuhn-Tucker point. However, Algorithm 3.1 does not compute an infinite sequence of iteration points $\bar{\tau}$ because $\Delta t > 0$ and $T < \infty$. For this reason we characterize convergence of the algorithm not in an asymptotic sense, but rather in terms of convergence rate, as will be explained in the next paragraph.

Consider the Newton-Raphson method for minimizing a twice-continuously differentiable function $f : R^n \rightarrow R$:

$$x_{k+1} \;=\; x_k - H(x_k)^{-1}\nabla f(x_k), \qquad k = 1, 2, \ldots, \tag{10}$$

where $H(x) := \frac{d^2 f}{dx^2}(x)$ is the Hessian. Then, if $\lim_{k\rightarrow\infty} x_k = x^*$ for some $x^* \in R^n$, and if $H(x^*)$ is positive definite, then x^* is a local minimum for f; moreover, there exist $\delta > 0$ and $K > 0$ such that, if $||x_k - x^*|| < \delta$, then $||x_{k+1} - x^*|| \leq K||x_k - x^*||^2$. This means that the algorithm has a quadratic convergence rate (see [9]).

The convergence rate that we establish is not quadratic due to the terms $e(t) := x(t) - \hat{x}(t)$ and Δt. In fact, the error term $e(t)$ implies a nonzero offset, and even if $e(t) = 0$, the convergence rate is first-order due to the fact that $\Delta t > 0$. Only if $e(t) = 0$ and $\Delta t \rightarrow 0$ (as is the case for the optimization process (7)), will the resulting convergence rate be quadratic. To make these observations precise, fix a time-point $t \in [0, T)$ and consider a vector $\bar{\tau}$ lying in the interior of the feasible set $\Phi(t)$, namely $t < \tau_1 < \ldots < \tau_N < T$. Suppose that $\bar{\tau}$ is a stationary point for the problem $\Pi_{0,x(0)}$, and hence (by Proposition 3.1) for $\Pi_{t,x(t)}$ as well; consequently $\frac{\partial J}{\partial \tau}(t, x(t), \bar{\tau}) = 0$. Our main convergence result is in terms of the proximity of the computed points $\bar{\tau}(t)$ to such a stationary point $\bar{\tau}$.

Denote the Hessian $\frac{\partial^2 J}{\partial \tau^2}(t, x(t), \bar{\tau})$ by H.

Remark. The independence of H on t requires a clarification. We assume that all the switching times are in the future with respect to t, namely, $\tau_1(t) > t$. If two points, t_1 and $t_2 > t_1$ satisfy this assumption, then $J(t_1, x(t_1), \bar{\tau}) - J(t_2, x(t_2), \bar{\tau}) = \int_{t_1}^{t_2} L(x(\tau))d\tau$. This difference term does not involve the future switching-times vector $\bar{\tau}$, and hence all derivatives of $J(t, x(t), \bar{\tau})$ with respect to $\bar{\tau}$ are the same at $t = t_1$ and $t = t_2$. Note that this argument hinges on the fact that the initial condition for the cost-to-go at time t is the "true" state variable $x(t)$.

Proposition 3.2: Suppose that $\bar{\tau}$ is a stationary point for $\Pi_{t,x(t)}$ lying in the interior of the feasible set $\Phi(t)$, and that H is positive definite. There exist constants $\delta > 0$ and $K > 0$ such that, if $||\bar{\tau}(t) - \bar{\tau}|| < \delta$; $\Delta t < \delta$; $||e(t)|| < \delta$ and $||e(t + \Delta t)|| < \delta$; $\gamma(t) = 1$ in Step 3; and

$\tau_1(t) > t$ and $\tau_1(t+\Delta t) > t+\Delta t$, then

$$\begin{aligned} & ||\bar{\tau}(t+\Delta t) - \bar{\tau}|| \\ \leq\ & K\Big(||\bar{\tau}(t) - \bar{\tau}||^2 + \Delta t ||\bar{\tau}(t) - \bar{\tau}|| + ||e(t+\Delta t)|| \Big). \end{aligned} \tag{11}$$

Proof: Since $\bar{\tau}$ lies in the interior of the feasible set $\Phi(t)$, there exists $\delta_1 > 0$ such that if $||\bar{\tau}(t) - \bar{\tau}|| < \delta_1$, $\Delta t < \delta_1$, $||e(t+\Delta t)|| < \delta_1$, and $\gamma(t) = 1$, then

$$\bar{\tau}(t+\Delta t) \ = \ \bar{\tau}(t) - H(t)^{-1}\frac{\partial J}{\partial \tau}(t+\Delta t, \hat{x}(t+\Delta t), \bar{\tau}(t)); \tag{12}$$

this is evident by following Steps 1 – 3 of the algorithm. Subtracting $\bar{\tau}$ from both sides of this equation yields,

$$\bar{\tau}(t+\Delta t) - \bar{\tau} = \ \bar{\tau}(t) - \bar{\tau} - H(t)^{-1}\frac{\partial J}{\partial \tau}\big(t+\Delta t, \hat{x}(t+\Delta t), \bar{\tau}(t)\big). \tag{13}$$

By assumption $\bar{\tau}$ is a local minimum for $J(t, x(t), \cdot)$. Therefore, and by Bellman's optimality principle, $\bar{\tau}$ is a local minimum for $J(t+\Delta t, x(t+\Delta t), \cdot)$ as well, and hence

$$\frac{\partial J}{\partial \tau}\big(t+\Delta t, x(t+\Delta t), \bar{\tau}\big) \ = \ 0. \tag{14}$$

Consequently, and by (13),

$$\bar{\tau}(t+\Delta t) - \bar{\tau} = \ \bar{\tau}(t) - \bar{\tau} - H(t)^{-1}\Big(\frac{\partial J}{\partial \tau}\big(t+\Delta t, \hat{x}(t+\Delta t), \bar{\tau}(t)\big) - \frac{\partial J}{\partial \tau}\big(t+\Delta t, x(t+\Delta t), \bar{\tau}\big)\Big). \tag{15}$$

By Corollary 3.1, the last difference term in the RHS of (15) can be written as

$$\begin{aligned} & \frac{\partial J}{\partial \tau}\big(t+\Delta t, \hat{x}(t+\Delta t), \bar{\tau}(t)\big) - \frac{\partial J}{\partial \tau}\big(t+\Delta t, x(t+\Delta t), \bar{\tau}\big) \\ = \ & \frac{\partial^2 J}{\partial \tau \partial x}\big(t+\Delta t, x(t+\Delta t), \bar{\tau}\big) e(t+\Delta t) + O\big(||e(t+\Delta t)||^2\big) \\ + \ & \frac{\partial^2 J}{\partial \tau^2}\big(t+\Delta t, x(t+\Delta t), \bar{\tau}\big)\big(\bar{\tau}(t) - \bar{\tau}\big) + O\big(||\bar{\tau}(t) - \bar{\tau}||^2\big), \end{aligned} \tag{16}$$

where $\limsup_{\eta \to 0} \frac{O(\eta^2)}{\eta^2} < \infty$.

Next, recall that $H(t) := \frac{\partial^2 J}{\partial \tau^2}(t, \hat{x}(t), \bar{\tau}(t))$, and that $H := \frac{\partial^2 J}{\partial \tau^2}(t, x(t), \bar{\tau})$ is positive definite by assumption. Therefore, and by Corollary 3.1 there exists $\delta_2 \in (0, \delta_1)$ and $K_1 > 0$ such that, if $\Delta t < \delta_2$, $||e(t+\Delta t)|| < \delta_2$, and $||\bar{\tau}(t) - \bar{\tau}|| < \delta_2$, then

$$||H(t)^{-1} - \Big(\frac{\partial^2 J}{\partial \tau^2}\big(t+\Delta t, x(t+\Delta t), \bar{\tau}\big)\Big)^{-1}|| \leq \ K_1\Big(\Delta t + ||e(t+\Delta t)|| + ||\bar{\tau}(t) - \bar{\tau}||\Big). \tag{17}$$

Now plug the RHS of (16) in (15) to obtain,

$$\begin{aligned}
&\bar{\tau}(t+\Delta t)-\bar{\tau} \\
=\ &\bar{\tau}(t)-\bar{\tau}-H(t)^{-1}\Big(\frac{\partial^2 J}{\partial\tau\partial x}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)e(t+\Delta t)+O\big(||e(t+\Delta t)||^2\big) \\
&+\frac{\partial^2 J}{\partial\tau^2}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)\big(\bar{\tau}(t)-\bar{\tau}\big)+O\big(||\bar{\tau}(t)-\bar{\tau}||^2\big)\Big). \qquad (18)
\end{aligned}$$

Subtracting and adding the term $\Big(\frac{\partial^2 J}{\partial\tau^2}(t+\Delta t,x(t+\Delta t),\bar{\tau})\Big)^{-1}$ from $H(t)^{-1}$ in (18) we obtain,

$$\begin{aligned}
\bar{\tau}(t+\Delta t)-\bar{\tau}=\ &\bar{\tau}(t)-\bar{\tau}-\Big(H(t)^{-1}-\big(\frac{\partial^2 J}{\partial\tau^2}(t+\Delta t,x(t+\Delta t),\bar{\tau})\big)^{-1}\Big) \\
&\times\Big(\frac{\partial^2 J}{\partial\tau\partial x}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)e(t+\Delta t)+O\big(||e(t+\Delta t)||^2\big) \\
&+\frac{\partial^2 J}{\partial\tau^2}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)\big(\bar{\tau}(t)-\bar{\tau}\big)+O\big(||\bar{\tau}(t)-\bar{\tau}||^2\big)\Big) \\
&-\big(\frac{\partial^2 J}{\partial\tau^2}(t+\Delta t,x(t+\Delta t),\bar{\tau})\big)^{-1} \\
&\times\Big(\frac{\partial^2 J}{\partial\tau\partial x}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)e(t+\Delta t)+O\big(||e(t+\Delta t)||^2\big) \\
&+\frac{\partial^2 J}{\partial\tau^2}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)\big(\bar{\tau}(t)-\bar{\tau}\big)+O\big(||\bar{\tau}(t)-\bar{\tau}||^2\big)\Big), \qquad (19)
\end{aligned}$$

and after a bit of algebra, the RHS of (19) can be seen to have the form

$$\begin{aligned}
\bar{\tau}(t+\Delta t)-\bar{\tau}=\ &-\Big(H(t)^{-1}-\big(\frac{\partial^2 J}{\partial\tau^2}(t+\Delta t,x(t+\Delta t),\bar{\tau})\big)^{-1}\Big) \\
&\times\Big(\frac{\partial^2 J}{\partial\tau\partial x}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)e(t+\Delta t)+O\big(||e(t+\Delta t)||^2\big) \\
&+\frac{\partial^2 J}{\partial\tau^2}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)\big(\bar{\tau}(t)-\bar{\tau}\big)+O\big(||\bar{\tau}(t)-\bar{\tau}||^2\big)\Big) \\
&-\big(\frac{\partial^2 J}{\partial\tau^2}(t+\Delta t,x(t+\Delta t),\bar{\tau})\big)^{-1} \\
&\times\Big(\frac{\partial^2 J}{\partial\tau\partial x}\big(t+\Delta t,x(t+\Delta t),\bar{\tau}\big)e(t+\Delta t)+O\big(||e(t+\Delta t)||^2\big)+O\big(||\bar{\tau}(t)-\bar{\tau}||^2\big)\Big). \qquad (20)
\end{aligned}$$

Consequently, and by (17) and Corollary 3.1, there exist $K_2>0$ and $\delta_3\in(0,\delta_2)$ such that, if $\Delta t<\delta_3$, $||e(t)||<\delta_3$, $||e(t+\Delta t)||<\delta_3$, and $||\bar{\tau}(t)-\bar{\tau}||<\delta_3$, then

$$\begin{aligned}
||\bar{\tau}(t+\Delta t)-\bar{\tau}||<\ &K_2\Big(\Delta t+||e(t+\Delta t)||+||\bar{\tau}(t)-\bar{\tau}||\Big)\times\Big(||e(t+\Delta t)|| \\
&+||\bar{\tau}(t)-\bar{\tau}||\Big)+\ K_2\Big(||e(t+\Delta t)||+||\bar{\tau}(t)-\bar{\tau}||^2\Big). \qquad (21)
\end{aligned}$$

Finally, by reducing K_2 and δ_3 if necessary, there exist $K>0$ and $\delta>0$ such that, if the assumptions of the proposition are satisfied, then Equation (11) is in force. ■

The significance of this result can be seen in the relative size of the three terms in the RHS of Equation (11). The first term is inherent in the Newton-Raphson method, and if it dominates the two other terms, then practically the algorithm would have a quadratic convergence rate. However, if Δt is large enough so that the second term dominates the RHS of (11), then the algorithm's convergence rate would be linear. If the third term dominates the other two terms then the convergence rate of the algorithm is bounded by the convergence rate of the estimation error to 0, and this rate may be made faster by the suitable choice of the state observer. Some evidence of all of this will be presented in the next section. We point out that the choice of Δt generally is independent of the state observer, and it can be bounded from below by the available computing rate and the computational requirements of a single iteration of the algorithm, especially in Step 1. If the estimation error is known to converge to zero fast, then the choice of a small Δt can speed up the algorithm's convergence rate albeit at large computing costs. However, if the estimation error is known to be large enough so that the third term in the RHS of (11) dominates the other two terms, then allowing for larger values of Δt could save computational resources while not necessarily slowing down the algorithm's convergence.

IV. Numerical Experiments

This section presents some numerical results of Algorithm 3.1. First we test the algorithm on a simple problem and highlight the significance of Equation (11), then we extend the setting of the algorithm to permit the addition of new modes, where we test it on a tracking problem.

A. Example: Convergence Rate

Consider the two-dimensional, linear system defined by $\dot{x} = A_i x$, $y = c_i x$, $i = 1, 2$, where

$$A_1 = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}, \qquad \text{and } c_1 = c_2 = \begin{pmatrix} 1 & 1 \end{pmatrix},$$

and let the cost functional be $J = \int_0^T ||x(t)||^2 dt$. We use the Luenberger observer so that the equation for the state estimator is $\dot{\hat{x}} = A_i\hat{x} - \ell_i c_i(x - \hat{x})$; the observer matrices ℓ_i will be specified later. The initial conditions are $x_0 = (0.55, 0.55)^T$ and $\hat{x}_0 = (0.3, 0.7)^T$, and we chose $T = 5.0$s. We start at mode 1, namely $\dot{x} = A_1 x$, and consider a single switching to mode 2 at a time $\tau \in [0, T]$, where τ serves as the variable for the optimization problem. The initial value for

τ, where the algorithm starts, is at $\tau = 2.5$, and the optimal point, denoted by $\tau^\star$, computed analytically, is $\tau^\star = 1.71$.

We made four runs of Algorithm 3.1, with two values of Δt and two sets of observer matrices ℓ_i, and the results are shown in Figure 1. In part A of the figure, $\Delta t = 0.05$, and the observer matrices are $\ell_1 = (-1\ 5)^T$ and $\ell_2 = (3.333\ 0.667)^T$ so that the eigenvalues of the observer subsystems are at $-2 \pm j$ for both $i = 1$ and $i = 2$. The graph in the figure is of the variable τ computed by the algorithm, as a function of time t, while the horizontal line indicates the optimal value $\tau = \tau^\star = 1.71$. The dashed diagonal line is the graph of t, and the vertical line indicates the time t where $\tau = t$. Prior to that time $\tau > t$, namely the switching time is in the future (with respect to t) and hence τ can be modified by the algorithm. After that time $\tau < t$, the switching has already occurred and hence τ cannot be changed by the algorithm. This is apparent from the figure, where τ approaches the value of (about) 1.6, and it remains a constant once the switching has occurred and become part of the past.

The algorithm's convergence can be sped up by reducing Δt or choosing a more aggressive observer. First, we kept Δt at 0.05 (as in part A) and changed the observer matrices to $\ell_1 = (-37\ 53)^T$ and $\ell_2 = (35.333\ \ -19.333)^T$, so that the eigenvalues of the matrices $A_i - \ell_i c_i$ are at $-8 \pm 5j$ for both $i = 1$ and $i = 2$. The results are shown in part B of the figure, and they indicate better convergence than in part A. Next, we reduced Δt from 0.05 to 0.005 and used the same observer as in part A, and the results, shown in part C of the figure, also indicate an improvement over those in part A. For best results, shown in part D, we used the observer of part B and Δt of part C.

It is evident from the figure that the use of a more aggressive observer yields faster convergence of the algorithm. Less pronounced but also evident is the fact that smaller Δt yields faster convergence. To further test this point, we set the estimation error to 0 (by setting $\hat{x}(0) = x(0)$), and ran the algorithm with three values of Δt: 0.1, 0.01, and 0.001. Starting at $\tau(0) = 2.5$, Table I shows the quantities $\ln(|\tau_k - \tau^\star|)$ for the first 7 iterations of the algorithm and for all three values of Δt, where τ_k is the switching time at the start of the kth iteration. Recall that Δt is not the integration interval (in all three runs the integration interval was 0.001), but the time between two consecutive iterations of the algorithm. As mentioned earlier, if the estimation error can be neglected, then with a small Δt the RHS of (11) would be dominated by the term $||\bar{\tau}(t) - \bar{\tau}||^2$ and hence indicate a quadratic convergence rate, while if Δt is large then the RHS

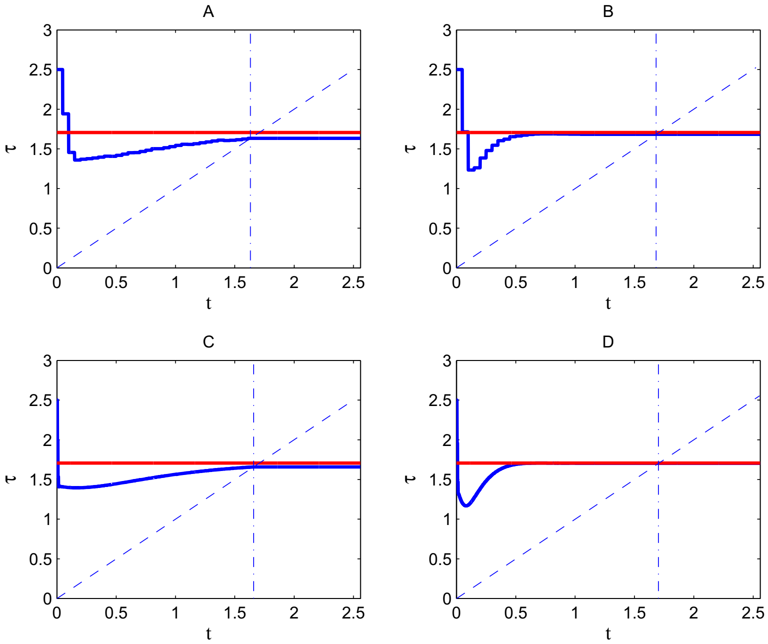

Fig. 1. Results of Algorithm 3.1 with One Switching Variable

of (11) would be dominated by $\Delta t||\bar{\tau}(t) - \bar{\tau}||$, indicating a linear convergence rate. It is hard to measure convergence rate by experimental means, but the results of Table I clearly show faster convergence for smaller Δt.

k	0	1	2	3	4	5	6
$\Delta t = 0.1$	-0.2307	-1.4002	-2.4229	-2.8109	-2.9854	-3.0477	-3.0679
$\Delta t = 0.01$	-0.2307	-1.2050	-3.9477	-5.4536	-5.1579	-5.0772	-5.7839
$\Delta t = 0.001$	-0.2307	-1.1899	-4.3522	-7.6708	-8.6631	-9.4915	-10.0498

TABLE I

$\ln(|\tau_k - \tau^\star|)$ FOR $k = 0, \ldots, 6$

B. Example: A Tracking Problem

Consider the two-dimensional system having the modes $\dot{x} = Ax$ or $\dot{x} = Ax + b$ (denoted by Mode 1 and Mode 2, respectively), where

$$A \; = \; \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix}$$

and $b = (1\ 1)^T$. The problem under study is to track the linear curve $\{r(t)\}_{t \in [0,10]}$, defined by $r(t) = (0.1t, 0.025t)^T$, by the state trajectory $\{x(t)\}$. Thus, we attempt to minimize the cost functional $J := \frac{1}{2}\int_0^T ||x(t) - r(t)||^2 dt$, with $T = 10$, as a function of the switching times. The number of the switchings is not fixed a priori, and it is modified in the manner described below. The system starts at Mode 1 and then alternates between the two modes. The output equation of the system is $y = cx$ with $c = (1\ 1)$, and the (Luenberger) observer matrix is $\ell = (0.5\ \ -0.5)^T$, so that the eigenvalues of the observer's system matrix $A - \ell c$ are both at -1. The initial state is $x(0) = (0.55, 0.55)^T$, and the initial state estimator is $\hat{x}(0) = (0, 2)^T$.

It is not unlikely that the optimal tracking would involve a sliding-mode switching control policy requiring infinite switchings between Mode 1 and Mode 2. However, as is the case in many applications, the number of switchings has to be limited for practical reasons, and therefore we restrict it, somewhat arbitrarily, to 5. Thus, starting at a pre-specified initial point $\bar{\tau}(0)$, the algorithm computes iteration-points $\bar{\tau}(t) = (\tau_1(t), \ldots, \tau_5(t))^T$ at times $k\Delta t$, $k = 1, 2, \ldots$. However, if at some point $t_1 < T$ it happens that $\tau_1(t_1) \leq t_1$, then ahead of this point the first switching time is a part of the past and cannot be modified; we say that the switching time τ_1 freezes and it is no longer a part of the variable parameter whose dimension is reduced to 4. If τ_2 later freezes at a future time $t_2 \in (t_1, T]$, then the variable parameter becomes 3-dimensional. Continuing in this way, all of the switching times may freeze, and this can happen prematurely before the algorithm computes an adequate approximation to the optimal schedule.

To get around this problem we insert two switching times when the dimension of the variable becomes 3. It is done in the following way. Given a time $\tilde{\tau} \in (0, T)$ which is not a switching time, let $\lambda \geq 0$ be small enough so that no switching time is contained in the interval $[\tilde{\tau} - \lambda, \tilde{\tau} + \lambda]$, and consider inserting, in that interval, the complementary mode to the mode that is active at time $\tilde{\tau}$. This insertion results in adding to the schedule two switching times at the two end-points of the interval. Denoting the cost functional J as a function of λ by $\tilde{J}(\lambda)$, let us define

$D(\tilde{\tau}) := \frac{d\tilde{J}}{d\lambda^+}(0)$. Thus, if $D(\tilde{\tau}) < 0$ then such an insertion at a small-enough interval would reduce the value of the cost functional J. We chose the point $\tilde{\tau}$ to be the point τ where $D(\tau)$ obtained its minimum value over a suitable grid of the interval $[0, T]$, as long as this minimum was negative (which it was, in our case). We then inserted the complementary mode at time $\tilde{\tau}$ for an interval of duration $\lambda = 0$, namely both new switching times are at $\tilde{\tau}$, and the fact that $D(\tilde{\tau}) < 0$ ensures that the algorithm will act to separate them.

The results of the algorithm run are shown in Figure 2. The upper graph shows the evolution of the vector $\bar{\tau}(t)$ as a function of time t. Starting at $\bar{\tau}(0) = (1.7, 3.3, 5.0, 6.7, 8.3)^T$, the switching times are drawn by the thick curves. The piecewise-horizontal lines show the optimal switching times for a given schedule, computed off line as points of reference. τ_1 freezes at time $t_1 = 2.93$; at this time the problem is changed, its dimension reduced to 4, and its solution point and the horizontal lines that mark it are changed as well. τ_2 later freezes at time $t_2 = 3.59$, and at this point we perform a mode-insertion at the time $\tilde{\tau} = 5.75$. Now the number of variables is restored to 5, and the algorithm continues its course until the switching times freeze one-at-a-time, and this happens at the time-points 4.55, 5.24, 5.68, 6.49, and 6.86. The graph suggests that these points are close to the optimal points as indicated by the horizontal lines, and therefore we do not insert any more new modes. We mention that computing the optimal solution (marked by the horizontal lines) is done only to gauge the algorithm's performance and may be infeasible in applications. The lower graph of the figure shows the evolution of the two components of the state trajectory (thick curves), the two components of the state estimator (dashed curves), and the two co-ordinates of the line that has to be tracked (straight lines). We clearly discern a trend toward improving the tracking objective.

We close this section by commenting on the mode-insertion technique defined above. The example that we discussed involved only two modes, and hence there is no ambiguity about which mode to insert at a given time. However, if the system has more than two modes, then the algorithm has to choose not only the insertion time but also the mode that will be inserted at that time. A reasonable way to choose the mode and insertion time is to consider terms like $D(\tau)$ for every modal function, and to choose the mode and time τ that minimize (from a given grid) that term, as long as the minimum is negative. Reference [2] analyzed such an algorithm, and proved convergence for suboptimal switching schedules for problems with variable numbers of switching times.

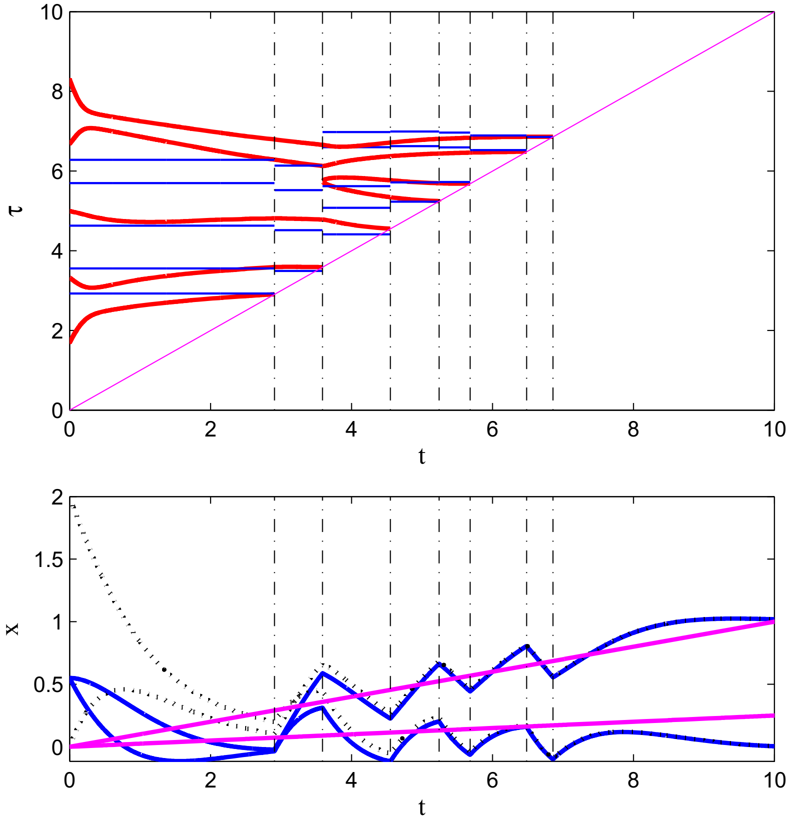

Fig. 2. Results of Algorithm 3.1 for a Tracking Problem

V. Conclusions

This paper proposes a real-time, Newton-like algorithm for timing optimization problems defined on hybrid dynamical systems, where the variable parameter consists of the switching times among the modes. The state variable is estimated by a state observer, and the algorithm attempts to minimize the cost-to-go at each time t. The main result is a characterization of the algorithm's convergence rate, which may vary from sublinear to quadratic, depending on its computing rate and the state estimation error. Numerical examples support the theoretical developments.

References

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

[2] H. Axelsson, Y. Wardi, M. Egerstedt, and E.I. Verriest, “Gradient Descent Approach to Optimize Mode Sequencing in Hybrid Dynamical Systems”, *Journal of Optimization Theory and Applications*, Vol. 136, No. 2, pp. 167-186, 2008.

[3] 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.

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

[5] M. Egerstedt, Y. Wardi, and H. Axelsson, “Transition-Time Optimization for Switched Systems”, *IEEE Transactions on Automatic Control*, Vol. AC-51, pp. 110-115, 2006.

[6] H. Halkin, “Necessary Conditions for Optimal Control Problems with Differenetiable or Nondifferentiable Data”, in *Proc. Conference on Mathematical Control Theory*, Canberra, Australia, August 23 – September 2, 1977; Lecture Notes in Mathematics, Springer-Verlag, Vol. 680, Ed. W.A. Coppel, pp. 77-118, 1977.

[7] P. E. Moraal and J. W. Grizzle, “Observer Design for Nonlinear Systems with Discrete-Time Measurements”, *IEEE Trans. on Automatic Control*, Vol. 40, pp. 395–404, 1995.

[8] B. Piccoli, “Hybrid Systems and Optimal Control”, in *Proc. Conference on Decision and Control*, Tampa, Florida, December 1998.

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

[10] M.S. Shaikh and P. Caines, “On Trajectory Optimization for Hybrid Systems: Theory and Algorithms for Fixed Schedules”, in *Proc. IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[11] M.S. Shaikh and P.E. Caines, “ On the Optimal Control of Hybrid Systems: Optimization of Trajectories, Switching Times and Location Schedules”, in *Proceedings of the 6th International Workshop on Hybrid Systems: Computation and Control*, Prague, The Czech Republic, 2003.

[12] H.J. Sussmann, “Set-Valued Differentials and the Hybrid Maximum Principle”, in *Proc. IEEE Conference on Decision and Control*, Vol. 1, pp. 558-563, December 2000.

[13] Y. Wardi, X.D. Ding, M. Egerstedt, and S. Azuma, “On-Line Optimization of Switched-Mode Systems: Algorithms and Convergence Properties”, in *Proc. 46th CDC*, pp. 5005-5010, New Orleans, Louisiana, December 12-14, 2007.

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

[15] 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.