Optimal Control of Autonomous Switched-Mode Systems: Gradient-Descent Algorithms with Armijo Step Sizes

Y. Wardi, M. Egerstedt, and M. Hale

Research supported in part by NSF under Grant CNS-1239225.

Abstract

This paper concerns optimal mode-scheduling in autonomous switched-mode hybrid dynamical systems, where the objective is to minimize a cost-performance functional defined on the state trajectory as a function of the schedule of modes. The controlled variable, namely the modes' schedule, consists of the sequence of modes and the switchover times between them. We propose a gradient-descent algorithm that adjusts a given mode-schedule by changing multiple modes over time-sets of positive Lebesgue measures, thereby avoiding the inefficiencies inherent in existing techniques that change the modes one at a time. The algorithm is based on steepest descent with Armijo step sizes along Gâteaux differentials of the performance functional with respect to schedule-variations, which yields effective descent at each iteration. Since the space of mode-schedules is infinite dimensional and incomplete, the algorithm's convergence is proved in the sense of Polak's framework of optimality functions and minimizing sequences. Simulation results are presented, and possible extensions to problems with dwell-time lower-bound constraints are discussed.

Keywords. Switched-mode systems, optimal control, optimization theory, Armijo step size, optimality functions.

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332. Email:{ywardi,magnus,mhale30}@ece.gatech.edu

I. Introduction

This paper concerns dynamical systems described via the following equation,

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

where $x \in R^n$ is the state variable, $v \in V$ for a given finite set V, and $f : R^n \times V \rightarrow R^n$ is a suitable function. Suppose that the system evolves on a horizon-interval $[0,T]$ for some fixed $T > 0$, and that the initial state $x(0) = x_0$ is given for some $x_0 \in R^n$. The input control to this system, $v(t) : [0,T] \rightarrow V$, will be denoted by v for brevity, and the context will ensure that no confusion arises from the use of v as a control signal as well as an element of V. Every point $v \in V$ corresponds to a particular mode of the system and hence the control function $v : [0,T] \rightarrow V$ represents the schedule of modes. Let $L : R^n \rightarrow R$ be a function, and defining the cost-functional (performance criterion) J to be

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

we consider the problem of minimizing J over a class of control functions v.

A more-general class of systems has a continuous control $u \in R^k$ in addition to the discrete-valued control v, and such systems are called *controlled* in contrast to the systems defined by (1) which are called *autonomous*. We focus on autonomous systems only since they capture the salient features of switched-mode optimization, and will point out natural extensions of the results, derived in the sequel, to the case of controlled systems.

Such systems, autonomous and controlled, and their related optimization problems have been investigated in the past several years due to their relevance in control applications such as mobile robotics [14], automotive powertrain control [34], switching circuits [13]; [1] and references therein, telecommunications [26], [20], and situations where a controller has to switch its attention among multiple subsystems or data sources [21], [7]. The optimal control (optimal mode-scheduling) problem was defined in a general framework of nonlinear switched-mode systems in [6], several variants of the maximum principle were derived for it in [23], [31], [32], [27], [30], and subsequently a number of provably-convergent algorithms were developed in [37], [38], [27], [28], [15], [8], [29], [30], [3], [5], [4], [19], [9], [10], [11], [33], [16]. References [18], [12] developed an optimal control framework for systems whose modes are determined by their respective preceding events. For a recent, comprehensive survey please see [39].

Early algorithms for the optimal control problem considered the case where the sequence of modes, namely successive values of v, is given and the variable parameter consists of the switching times [37], [38], [27], [28], [15]. For the general scheduling problem, where the variable parameter consists of both the mode-sequence and the switching times, a number of approaches recently emerged, including zoning and location algorithms that use the geometric structure of the problem to iterate on the mode-sequences [8], [29], [30], relaxation algorithms that use averaging techniques [5], [9], [10], [33], [22], [11], and needle-variations methods [3], [4], [19], [36]. Our algorithm falls in the latter category.

The starting point for our investigation is in the mode-insertion algorithms developed in [4], [19]. These algorithms alternate between the following two steps: (1). Given a sequence of modes, compute the optimal switching times. (2). Update the mode-sequence by replacing a single mode by another mode during a certain time-interval. This approach may have the potential drawback of requiring an infinite-loop algorithm each time step 1 is entered, and its effectiveness at the mode-insertion step may be limited by the requirement of changing only a single mode at step 2. This can become problematic if such a mode is inserted in a short interval, which can be the case when an optimal schedule is being approached by the algorithms. It is these two points that motivated us to explore algorithms that iterate directly in the mode-schedule space without a need for optimizing mode-schedules for given mode-sequences; in other words, we develop a provably-convergent algorithm that eliminates step 1 while extending step 2 to include the switching of multiple modes across large sets in the horizon interval $[0,T]$.

The main idea underscoring our algorithm is to identify sets of time-points where needle variations yield lower values of the cost functional J, parameterize them according to their Lebesgue measures, and compute a set where, switching the modes in all of its points, results in a steep decline in J. The step size, namely the Lebesgue measure of the above set, is computed according to the Armijo procedure, having an essential quality of descent that yields effective algorithms and guarantees their convergence theoretically.

The theory of nonlinear programming contains various results regarding convergence of descent algorithms with Armijo step size [25]. However, these typically were derived in the setting of finite-dimensional optimization, and they do not apply to our scheduling optimization problem whose parameter-space is not only infinite-dimensional but also lacks a natural topology. Therefore a new framework for convergence analysis is being developed, and it is based on

Polak's notions of optimality functions and minimizing sequences, developed in [24] for infinite-dimensional optimization problems. Furthermore, our algorithm is of a *sufficient descent* type, and hence (see [25]) it yields considerable descent at mode schedules that are far from minimum (in a suitable sense); this point will be explained in detail in the sequel.

As mentioned earlier, one of the current approaches to the optimal control problem is based on relaxation and averaging. The gist of this approach is to consider a relaxed control comprised of the convex hull of the mode-functions, solve the resultant continuous-control problem by current techniques, and represent its solution point by a switched-mode control [5]. One of the appealing features of this approach is that the relaxed problem is convex in the case of autonomous systems as well as in a class of controlled systems, and it is especially suitable to problems whose solution points have infinite switching frequency. In fact, Reference [22] compared the algorithm in [5] to the one presented here, and found it to yield a lower cost for a particular two-dimensional problem (4.74 as compared to 4.78). However, our approach can have advantages in the following situations:

1) When there are lower-bound constraints on the dwelling times, i.e., periods during which modes remain fixed. Our method is suitable for this case (as will be explained in the sequel) while we do not see a direct extension of relaxation techniques.
2) When the solution point of the relaxed problem consists of extreme points, namely a switching-mode control; in some such cases our algorithm may converge quite fast.
3) Due to its sufficient-descent property, our algorithm can get a substantial amount of descent in the first few iterations. This is not untypical of gradient-descent algorithms with Armijo step sizes, and later examples indicate over 95% of the total descent in only about 5 iterations.

These points will be discussed in detail in the sequel.

The rest of the paper is organized as follows. Section II formulates the problem and recounts some existing results. Section III proposes our algorithm and establishes its sufficient-descent property. Section IV presents simulation results and discusses ways to extend the scope of the algorithm, and Section V concludes the paper.

We mention that the algorithm and its analysis were presented without proofs in the 2012 ACC [36], and the proofs were supplied in an unpublished technical memorandum [35].

II. Problem Formulation and Preliminary Results

Consider the system defined by Equation (1) with the cost functional defined by (2). Recall that the set V is finite, and suppose that the initial state and final time for the optimal control problem are given and fixed. The following assumption will be made throughout the paper.

Assumption 1: For every $v \in V$, the function $f(x,v) : R^n \rightarrow R$ is twice-continuously differentiable (C^2), and there exists a constant $K > 0$ such that, for every $x \in R^n$ and $v \in V$, $||\frac{d^2 f}{dx^2}(x,v)|| \leq K$.

We point out that the boundedness of the second derivative $\frac{d^2 f}{dx^2}(x,v)$ on compact sets in R^n follows from the C^2 property, and that is all that we need for the forthcoming analysis. However, formally assuming the bound over all of R^n will simplify the presentation.

The set V acts as an index set for the modes of the system, but we assume that it is a subset of R. The reason for this assumption is that the later discussion will involve topological concepts and properties of functions $v : [0,T] \rightarrow V$. For example, saying that such a function v is piecewise continuous means that it is piecewise constant, and speaking of the L^1 or L^2 norms of such functions requires a distance-measure on the set V.

We define a *feasible control* to be a function $v(t) : [0,T] \rightarrow V$ which is left continuous and changes its values a finite number of times in the interval $[0,T]$, and we denote by $\mathcal{V}$ the space of feasible controls. The condition of left continuity simplifies the analysis without detracting from its scope, and the condition of finite number of changes in v is certainly realistic. For every $v \in \mathcal{V}$ let v^i, $i = 1,\ldots,N+1$ (for some $N \geq 0$) denote the successive values of $v(t)$, $t \in [0,T]$, and let τ_i denote the switching time between v^i and v^{i+1}, $i = 1,\ldots,N$. We refer to the switching times by the vector notation $\bar{\tau} := (\tau_1,\ldots,\tau_N)^T \in R^N$, and we further define $\tau_0 := 0$ and $\tau_{N+1} := T$. Thus, the state equation (1) assumes the following form,

$$\dot{x} \;=\; f(x,v^i), \qquad \forall t \in [\tau_{i-1},\tau_i), \quad i = 1,\ldots,N+1, \tag{3}$$

and to simplify the notation in the sequel we denote the Right-hand Side (RHS) of (3) by $F(x,t)$, so that $\dot{x} = F(x,t)$. Furthermore, we will use the notation $f_i(x) := f(x,v^i)$ when no confusion arises, as shown in Figure 1. The optimal control (scheduling) problem is to minimize the performance function J, defined in Equation (2), over the space of feasible controls, $\mathcal{V}$.

It is convenient to use an alternative notation for representing the input control v as a schedule of modes of the system. The term “schedule” means sequencing as well as timing: the sequence

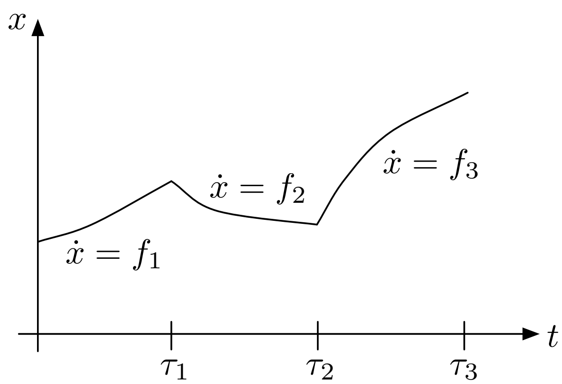

Fig. 1. The system is switching among different modes.

is that of the successive values of v, namely $\{v^i\}_{i=1}^{N+1}$, and the timing variable is comprised of the switching-times vector $\bar{\tau} := (\tau_1, \ldots, \tau_N)^T \in R^N$. Denote by $\mathcal{Q}$ the set of all finite sequences (finite strings) of elements from V, denote by q a typical element in $\mathcal{Q}$, and define $\ell(q)$ to be the length of (number of symbols in) q. Furthermore, for every $N = 1, 2, \ldots$, define $\Lambda_N := \{\bar{\tau} \in R^N : 0 \leq \tau_1 \leq \ldots \leq \tau_N \leq T\}$. Finally, define

$$\Sigma := \{(q, \bar{\tau}) \ : \ q \in \mathcal{Q}, \ \bar{\tau} \in \Lambda_{\ell(q)-1}\}. \tag{4}$$

We denote a typical element in Σ by $\sigma = (q, \bar{\tau})$, and the associated input control by $v_\sigma(\cdot) \in \mathcal{V}$. The mode associated with σ at a specific time t will be denoted for brevity by $v_\sigma(t)$. More generally, we will denote the mode associated with a point $w \in V$ by w instead of calling it "the mode associated with w".

Given $\sigma \in \Sigma$, define the costate $p(t) \in R^n$ by the following differential equation,

$$\dot{p} \ = \ -\Big(\frac{\partial F}{\partial x}(x,t)\Big)^T p - \Big(\frac{dL}{dx}(x)\Big)^T \tag{5}$$

with the boundary condition $p(T) = 0$ (recall that $F(x,t)$ is the RHS of (3)). The costate trajectory also can be used to compute the cost-sensitivity associated with needle variations of modes. To clarify this point, consider a schedule $\sigma := (q, \bar{\tau}) \in \Sigma$, a time $s \in [0, T)$, and an element $w \in V$. For a given $\lambda > 0$, consider inserting the mode (associated with) w to the schedule σ during the λ-long interval $[s, s+\lambda)$; that is, we modify the control v_σ by changing the values of $v_\sigma(t)$ to w for every $t \in [s, s+\lambda)$. Let us view the cost functional J as a function of

$\lambda \geq 0$, and denote it by $J_{\sigma,s,w}(\lambda)$. Under broad assumptions the right-derivative of this function at $\lambda = 0$ exists (see [25]), and denoting it by $D_{\sigma,s,w}$, it has the following form (e.g., [15]),

$$D_{\sigma,s,w} \ := \ \frac{dJ_{\sigma,s,w}}{d\lambda^+}(0) \ = \ p(s)^T\big(f(x(s),w) - f(x(s),v_\sigma(s))\big). \tag{6}$$

We call this one-sided derivative the *insertion gradient*, and we note that its computation requires the costate trajectory as defined by (5).

Now if $\sigma = (q,\bar{\tau}) \in \Sigma$ is an optimal schedule for J then for every $s \in [0,T)$ and $w \in V$, $D_{\sigma,s,w} \geq 0$. This is due to the fact (proved in Proposition 1, below) that if $D_{\sigma,s,w} < 0$ then inserting the mode w at a λ-long time-interval centered at s, for a sufficiently-small $\lambda > 0$, will result in a reduction in J. We can phrase this condition in the following, more-compact way,

$$\inf\Big\{\min\{D_{\sigma,s,w} : w \in V\} \ : \ s \in [0,T]\Big\} \ \geq \ 0. \tag{7}$$

Let us define, for every $s \in [0,T]$, $D_{\sigma,s} := \min\{D_{\sigma,s,w} : w \in V\}$; and define $D_\sigma := \inf\{D_{\sigma,s} : s \in [0,T]\}$, where we recognize D_σ as the Left-Hand Side (LHS) of (7). Now D_σ cannot be positive because for every $s \in [0,T]$, $D_{\sigma,s,v_\sigma(s)} = 0$ (inserting a mode onto itself would not change J) and hence $D_{\sigma,s} \leq 0$. Therefore, the necessary optimality condition for a schedule $\sigma = (q,\bar{\tau})$ is that $D_\sigma = 0$.

This optimality condition stimulated the development of the algorithm, proposed in [4] which, at each iteration, inserts a single mode to a given schedule as described in the Introduction. The algorithm in this paper pursues a different approach, in that at each iteration it considers, simultaneously, several modes for modification. In fact, the time-set where such modes are considered need not be an interval but can be a disconnected set, and it may have a large Lebesgue measure thereby yielding a large descent in J. This set is determined according to an Armijo-like procedure, albeit in a nonstandard setting. From a theoretical standpoint, the main hurdle we faced was in extending the Armijo step size from single intervals and single mode-switchings (as in [4], [19]) to general time-sets and multiple modes, and as we shall later see, this challenge was by-no-means trivial. Before discussing these issues, we recount the main results concerning the Armijo algorithm.

A. Descent Algorithms with Armijo Step Sizes

Consider the problem of minimizing a continuously-differentiable function $f : R^n \to R$ over $x \in R^n$. Steepest-descent techniques are iterative algorithms that move from a point $x \in R^n$

in the direction of $-\nabla f(x)$. Denoting by $\gamma(x) \geq 0$ the step size, the resulting (next) iteration point, denoted by x_{next}, is

$$x_{next} \;=\; x - \gamma(x)\nabla f(x). \tag{8}$$

The Armijo step size procedure defines $\gamma(x)$ by an approximate line minimization in the following way (see [25]): Given constants $\alpha \in (0,1)$ and $\beta \in (0,1)$ independent of $x \in R^n$. Define the integer $j(x)$ by

$$j(x) \;:=\; \min\Big\{ j = 0, 1, \ldots, \;:\; f(x - \beta^j \nabla f(x)) - f(x) \leq -\alpha\beta^j ||\nabla f(x)||^2 \Big\}, \tag{9}$$

and define

$$\gamma(x) \;=\; \beta^{j(x)}. \tag{10}$$

The *steepest descent algorithm with Armijo step size* computes a sequence of iteration points x_k, $k = 1, 2, \ldots,$ by the formula $x_{k+1} = x_k - \gamma(x_k)\nabla f(x_k)$, where $\gamma(x_k)$ is computed via (10) with $j(x_k)$ defined in (9). This algorithm is globally convergent (see [25]) to stationary points (typically local minima) and its convergence rate is linear. It is evidently a descent algorithm in the sense that $f(x_{next}) \leq f(x)$. In fact, (9) and (10) mean that $f(x_{next}) - f(x) \leq -\alpha\gamma(x)||\nabla f(x)||^2$, indicating that larger step sizes generally result in greater descent for a given gradient's magnitude $||\nabla f(x)||$. Reference [25] provides several practical suggestions for its implementation, including the following values of α and β, $\alpha = \beta = 0.5$.

The algorithm's convergence to stationary points has been proved under the assumption that $f(x)$ is continuously differentiable, and weaker assumptions require modifications of the algorithm. If $f(x)$ is twice-continuously differentiable then the step size is bounded from below according to the following result, proved in [25]. Let $H(x)$ denote the Hessian of f, namely $H(x) := \frac{d^2 f}{dx^2}(x)$, and let $\langle \cdot, \cdot \rangle$ denote the inner product in R^n.

Lemma 1: Suppose that $f(x)$ is C^2, and that there exists a constant $K > 0$ such that, for every $x \in R^n$, $||H(x)|| \leq K$. Then the following two statements are true: (1). For every $x \in R^n$, and for every $\gamma \in [0, \frac{2}{K}(1-\alpha)]$,

$$f(x - \gamma\nabla f(x)) - f(x) \;\leq\; -\alpha\gamma||\nabla f(x)||^2. \tag{11}$$

(2). For every $x \in R^n$,

$$\gamma(x) \;\geq\; \frac{2}{K}\beta(1-\alpha). \tag{12}$$

Proof: Please see the proofs of Equations (8a) and (8b) in [25], pp. 60-61. ■

We observe that the Right-hand Side (RHS) of (12) depends on the function $f(x)$ only via the upper bound on $||H(x)||$, K. Therefore, defining $\bar{\gamma} := \frac{2}{K}\beta(1-\alpha)$, we have that $\gamma(x) \geq \bar{\gamma}$, and hence, by (8) and (11),

$$f(x_{next}) - f(x) \ \leq \ -\alpha\bar{\gamma}||\nabla f(x)||^2. \tag{13}$$

According to this formula, the descent in f is at least by a quantity proportional to $||\nabla f(x)||^2$.

A slightly alternative view of the Armijo step size and Lemma 1 is obtained by scaling the search direction and the step size by $||\nabla f(x)||^{-1}$ and $||\nabla f(x)||$, respectively. Thus, defining $h(x) := \frac{\nabla f(x)}{||\nabla f(x)||}$ and $\lambda(x) := \gamma(x)||\nabla f(x)||$, Equation (8) becomes $x_{next} = x - \lambda(x)h(x)$. Furthermore, $\lambda(x)$ can be computed as follows: defining $j(x)$ by

$$j(x) \ : \ \min\Big\{j = 0, 1, \ldots, \ : \ f(x - \beta^j||\nabla f(x)||h(x)) - f(x) \leq -\alpha\beta^j||\nabla f(x)||^2\Big\}, \tag{14}$$

then it can be seen that

$$\lambda(x) \ = \ \beta^{j(x)}||\nabla f(x)||. \tag{15}$$

It is also evident that the steepest-descent algorithm with Armijo step size computes x_k, $k = 1, 2, \ldots$, by the formula $x_{k+1} = x_k - \lambda(x_k)h(x_k)$, and Lemma 1 is equivalent to the following assertion:

Corollary 1: Suppose that $f(x)$ is C^2, and that there exists a constant $K > 0$ such that, for every $x \in R^n$, $||H(x)|| \leq K$. Then the following two statements are true: (1). For every $x \in R^n$ and for every $\lambda \in [0, \frac{2}{K}(1-\alpha)||\nabla f(x)||]$,

$$f(x - \lambda h(x)) - f(x) \ \leq \ -\alpha\lambda||\nabla f(x)||. \tag{16}$$

(2). For every $x \in R^n$,

$$\lambda(x) \ \geq \ \frac{2}{K}\beta(1-\alpha)||\nabla f(x)||. \tag{17}$$

Proof: Immediate. ■

Suppose now that the steepest-descent algorithm with Armijo step sizes computes a sequence of iteration points x_k, $k = 1, 2, \ldots$, where $x_{k+1} = x_k - \lambda(x_k)h(x_k)$. Equations (16) and (17), with $c := \frac{2}{K}\beta\alpha(1-\alpha)$, yields the inequality $f(x_{k+1}) - f(x_k) \leq -c||\nabla f(x_k)||^2$. This implies that if $\hat{x} \in R^n$ is an accumulation point of the sequence $\{x_k\}_{k=1}^{\infty}$, then $\hat{x}$ satisfies the stationarity optimality condition $\nabla f(\hat{x}) = 0$. This reasoning has been extended to a general setting of continuous-parameter optimization which includes problems with constraints, nondifferentiable functions, and infinite-dimensional parameter spaces. The next subsection reviews the elements of this abstraction that are relevant to this paper.

B. Optimality Functions and Minimizing Sequences

The material surveyed below can be found in [25] (optimality functions) and [24] (minimizing sequences).

Let $\mathcal{X}$ be a normed linear space, and consider the problem of minimizing a function $f : \mathcal{X} \rightarrow R$. Given an appropriate optimality condition, let $\Delta \subset \mathcal{X}$ be the set where the optimality condition is satisfied, namely $x \in \Delta$ if and only if x satisfies the optimality condition. Furthermore, let $\theta : \mathcal{X} \rightarrow R^{-}$ be a non-positive valued function such that $\{x \in \mathcal{X} : \theta(x) = 0\} = \Delta$, and at every $x \in \mathcal{X}$, $|\theta(x)|$ indicates the extent to which x fails to satisfy the optimality condition. $\theta(\cdot)$ is called an *optimality function*.

An algorithm for solving the optimization problem typically computes a sequence of points $\{x_k\}_{k=1}^{\infty} \subset \mathcal{X}$. In nonlinear programming, where $\mathcal{X} = R^n$, a common requirement of an algorithm is that if $\hat{x}$ is an accumulation point of the sequence $\{x_k\}_{k=1}^{\infty}$, then is satisfies the optimality condition $\theta(\hat{x}) = 0$. Optimality functions often are not continuous but upper-semi continuous, namely, if $\lim_{m\rightarrow\infty} x_m = x$ then $\limsup_{m\rightarrow\infty} \theta(x_m) \leq \theta(x)$. This implies that, if the computed sequence $\{x_k\}_{k=1}^{\infty}$ satisfies the limit $\lim_{k\rightarrow\infty} \theta(x_k) = 0$, then each one of its accumulation points satisfies the optimality condition $\theta(\hat{x}) = 0$.

One way to ensure that $\lim_{k\rightarrow\infty} \theta(x_k) = 0$ is to have the algorithm have the following property, called *sufficient descent*: For every $\bar{x}$ such that $\theta(\bar{x}) < 0$ there exists $\delta > 0$ and $\eta > 0$ such that, if $||x_k - \bar{x}|| < \delta$, then $f(x_{k+1}) - f(x_k) \leq -\eta$. This guarantees, under mild assumptions, that if the sequence $\{x_k\}_{k=1}^{\infty}$ is bounded then $\lim_{k\rightarrow\infty} \theta(x_k) = 0$ and hence that each one of its accumulation points satisfies the optimality condition.

The latter result holds true regardless of whether $\mathcal{X}$ is finite dimensional or infinite dimensional. However, if $\dim(\mathcal{X}) = \infty$, a bounded sequence $\{x_k\}_{k=1}^{\infty}$ might not have any accumulation points, and in that case the result is vacuous. For this reason the convergence of algorithms has to be characterized by means not involving accumulation points. Reference [24] proposed a framework where an algorithm aims at computing not a minimum point, but rather a sequence $\{x_k\}_{k=1}^{\infty}$ such that the limit $\limsup_{k\rightarrow\infty} f(x_k)$ has a minimum value, and hence the existence of accumulation points is immaterial. The optimality condition analogous to stationarity is that

$$\lim_{k\to\infty} \theta(x_k) = 0 \tag{18}$$

or alternatively, if the algorithm is of a descent type, $\limsup_{k\to\infty} \theta(x_k) = 0$. This cannot be obtained by the property of sufficient descent mentioned above, but it follows (under certain conditions) from the following stronger condition: For every $C > 0$ there exists $\eta > 0$ such that, for every point $x_k \in \mathcal{X}$, computed by the algorithm, if $\theta(x_k) \leq -C$, then $f(x_{k+1}) - f(x_k) \leq -\eta$. This condition is called a *uniform sufficient descent*. In fact, the following result is immediate (see Section 1.2 of [25]): Suppose that $\inf\{f(x) : x \in \mathcal{X}\} > -\infty$. If a descent algorithm for minimizing $f(x)$ has the sufficient descent property, then for every iteration-sequence $\{x_k\}_{k=1}^{\infty}$ computed by it, Equation (18) is satisfied.

The next section presents our algorithm and derives its sufficient-descent property, the main result of the paper. This result is somewhat surprising since the insertion gradient is not continuous at boundary points of modes, and continuity of the gradient is essential for proving convergence of the Armijo procedure without resorting to (often cumbersome) techniques from nondifferentiable calculus.

III. Sufficient-Descent Algorithm for the Optimal Switched-Mode Problem

Consider the system defined by Equation (1) and the optimal control problem of minimizing J, defined via (2), as a function of the discrete control variable $v \in \mathcal{V}$. Let $\sigma \in \Sigma$ be the corresponding mode-schedule so that $v = v_\sigma$. Recall that, by definition, every schedule $\sigma \in \Sigma$ has a finite string, but the string-size of the mode-sequence $\{\sigma_k\}_{k=1}^{\infty}$ may be unbounded. The algorithm described below replaces a sufficiently-large set of modes at a given schedule so as to guarantee the uniform sufficient-decent property, and this is achieved by using the Armijo step size on the time-set where the modes are being replaced.

Consider a mode-schedule $\sigma \in \Sigma$ that does not satisfy the necessary optimality condition, namely $D_\sigma < 0$. Define the set $S_{\sigma,0}$ as $S_{\sigma,0} := \{s \in [0,T] \ : \ D_{\sigma,s} < 0\}$, and note that $S_{\sigma,0} \neq \emptyset$. For every $s \in S_{\sigma,0}$, consider a point $w \in V$ such that $D_{\sigma,s,\omega} = D_{\sigma,s}$, namely, $w \in \text{argmin}\{D_{\sigma,s,v} : v \in V\}$. Such w may not be unique but we assume a systematic way to assign a specific, single point w, which we denote by $w(\sigma,s)$ in order to highlight its dependence on σ nd s. For example, $w(\sigma,s) := v^i \in \ \text{argmin}\{D_{\sigma,s,v} : v \in V\}$ having the smallest index i.

Since $D_\sigma < 0$, for every $s \in S_{\sigma,0}$, $D_{\sigma,s,w(\sigma,s)} < 0$. This implies that, an insertion of the mode $w(\sigma, s)$ to the schedule σ at a small-enough interval beginning at s, would result in a decrease in J (a proof of this intuitive statement follows directly from Proposition 1, below). Our goal is to switch the modes in this fashion in a large subset of $S_{\sigma,0}$ so as to reduce J by a substantial amount, where by the term "substantial" we mean a decrease by at least aD_σ^2 for some constant $a > 0$. This uniform sufficient descent in J is akin to the descent property of the Armijo step size as reflected in Equation (13).

This sufficient-descent property cannot be guaranteed by changing the mode at every time-point $s \in S_{\sigma,0}$; not even any descent in J can be guaranteed. Instead, we search for a subset of $S_{\sigma,0}$ where, changing the mode at every s in that subset would guarantee a uniform sufficient descent. This subset will consist of points s where $D_{\sigma,s}$ is "more negative" than at typical points $s \in S_{\sigma,0}$. Fix $\eta \in (0, 1)$ and define the set $S_{\sigma,\eta}$ by

$$S_{\sigma,\eta} \ = \ \big\{s \in [0, T] \ : \ D_{\sigma,s} \le \eta D_\sigma\big\}, \tag{19}$$

as illustrated in Figure 2. Obviously $S_{\sigma,\eta} \neq \emptyset$ since $D_\sigma < 0$. Let $\mu(S_{\sigma,\eta})$ denote the Lebesgue measure of $S_{\sigma,\eta}$, and more generally, let $\mu(\cdot)$ denote the Lebesgue measure on R. For every subset $S \subset S_{\sigma,\eta}$, consider modifying σ by changing the mode from $v_\sigma(s)$ to $w(\sigma, s)$ at every point $s \in S$, and denote by $\boldsymbol{\sigma}(S)$ the resulting mode-schedule. Note the boldface notation $\boldsymbol{\sigma}$ which indicates that the designated mode schedule is a function of S; similar boldface notation will be used in the sequel to indicate functional notation. In the forthcoming we will search for a set $S \subset S_{\sigma,\eta}$ that will give us the desired sufficient descent.

Fix $\eta \in (0, 1)$. Consider a mapping $\mathbf{S} : [0, \mu(S_{\sigma,\eta})] \to 2^{S_{\sigma,\eta}}$ (the latter object is the set of subsets of $S_{\sigma,\eta}$) having the following two properties: (i) $\forall \lambda \in [0, \mu(S_{\sigma,\eta})]$, $\mathbf{S}(\lambda)$ is the finite union of intervals; and (ii) $\forall \lambda \in [0, \mu(S_{\sigma,\eta})]$, $\mu(\mathbf{S}(\lambda)) = \lambda$. Note that $\boldsymbol{\sigma}(\mathbf{S}(\lambda))$ is the mode-schedule obtained from σ by changing the mode at every time-point $s \in \mathbf{S}(\lambda)$ from $v_\sigma(s)$ to $w(\sigma, s)$. For example, $\forall \lambda \in [0, \mu(S_{\sigma,\eta})]$ define $s_\lambda := \inf\{s \in S_{\sigma,\eta} : \mu\big([0, s] \cap S_{\sigma,\eta}\big) = \lambda\}$, and define $\mathbf{S}(\lambda) := [0, s_\lambda] \cap S_{\sigma,\eta}$. Then $\boldsymbol{\sigma}(\mathbf{S}(\lambda))$ is the schedule obtained from σ by changing the modes lying in the leftmost subset of $S_{\sigma,\eta}$ having Lebesgue-measure λ, and it is the finite union of intervals if so is $S_{\sigma,\eta}$.

We next use such a mapping $\mathbf{S}(\lambda)$ to define an Armijo step-size procedure for computing a schedule σ_{next} from σ. Given constants $\alpha \in (0, 1)$ and $\beta \in (0, 1)$, in addition to $\eta \in (0, 1)$.

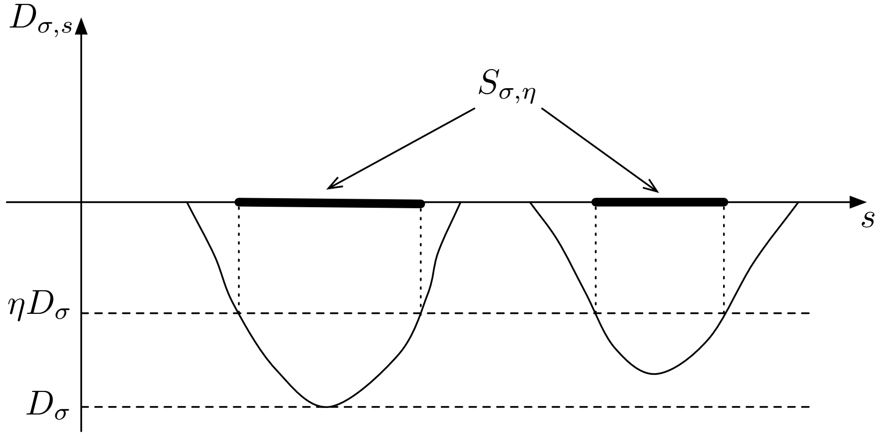

Fig. 2. Illustration of the set $S_{\sigma,\eta}$.

Consider a given $\sigma \in \Sigma$ such that $D_\sigma < 0$. For every $j = 0, 1, \ldots$, define $\lambda_j := \beta^j \mu(S_{\sigma,\eta})$, and define $j(\sigma)$ by

$$j(\sigma) \;:=\; \min\Big\{j = 0, 1, \ldots, \;:\; J(\boldsymbol{\sigma}(\mathbf{S}(\lambda_j))) - J(\sigma) \le \alpha\lambda_j D_\sigma\Big\}. \tag{20}$$

Finally, define $\boldsymbol{\lambda}(\sigma) := \lambda_{j(\sigma)}$, and set $\sigma_{next} := \boldsymbol{\sigma}(\mathbf{S}(\boldsymbol{\lambda}(\sigma)))$.

Now the algorithm we propose has the following form. Given constants $\alpha \in (0,1)$, $\beta \in (0,1)$, and $\eta \in (0,1)$. Suppose that for every $\sigma \in \Sigma$ such that $D_\sigma < 0$ there exists a mapping $\mathbf{S} : [0, \mu(S_{\sigma,\eta})] \to 2^{S_{\sigma,\eta}}$ with the aforementioned properties.

Algorithm 1: Step 0: Start with an arbitrary schedule $\sigma_0 \in \Sigma$. Set $k = 0$.

Step 1: Compute D_{σ_k}. If $D_{\sigma_k} = 0$, stop and exit; otherwise, continue.

Step 2: Compute $S_{\sigma_k,\eta}$ as defined in (19), namely $S_{\sigma_k,\eta} \;=\; \{s \in [0,T] \;:\; D_{\sigma_k,s} \le \eta D_\sigma\}$.

Step 3: Compute $j(\sigma_k)$ as defined by (20), namely

$$j(\sigma_k) \;=\; \min\Big\{j = 0, 1, \ldots, \;:\; J(\boldsymbol{\sigma}(\mathbf{S}(\lambda_j))) - J(\sigma_k) \le \alpha\lambda_j D_{\sigma_k}\Big\}, \tag{21}$$

and set $\boldsymbol{\lambda}(\sigma_k) := \lambda_{j(\sigma_k)}$.

Step 4: Define $\sigma_{k+1} := \boldsymbol{\sigma}(\mathbf{S}(\boldsymbol{\lambda}(\sigma_k)))$, namely the schedule obtained from σ_k by changing the mode at every time-point $s \in \mathbf{S}(\boldsymbol{\lambda}(\sigma_k))$ from $v_{\sigma_k}(s)$ to $w(\sigma_k, s)$. Set $k = k+1$, and go to Step 1. $\square$

Remark 1: E. Polak coined the phrases *conceptual algorithm* and *implementable algorithm*, and makes the point of distinguishing between them in the context of infinite-dimensional optimization [25]. Conceptual algorithms assume infinite computational precision and are used in analysis, while implementable algorithms are based on finite precision. Algorithm 1 is conceptual and it does not specify finite-precision approximations to $J(\sigma_k)$, D_{σ_k}, and $S_{\sigma_k,\eta}$. In fact, the entire discussion and analysis in the paper are carried out in the setting of conceptual algorithms since an extension to the implementable setting would require a longer paper and complicate the presentation without adding scope to the derived results.

Generally there are two principal approaches to implementable versions of conceptual algorithms: one discretizes the problem and then develops an algorithm for the resulting finite-dimensional problem, and the other discretizes the computation of the original, infinite-dimensional problem. The former approach underscores most of the developments in [25], whereas we implicitly adopt the latter approach. Therefore, while carrying out the entire analysis in the conceptual domain, we have in mind (but do not specify) a high-degree of grid-based approximations to the various quantities mentioned in the algorithm's statement.

The following discussion will be carried out under Assumption 1 and the following assumption.

Assumption 2: For every $v \in \mathcal{V}$, the function $w(\sigma, s)$ in the variable s is piecewise constant, left continuous, and has a finite number of switching points in the interval $s \in [0, T]$.

Remark 2: Although Assumption 2 cannot be proven from general properties of the vector fields $f_i(x)$, it is justified by the following argument that it is satisfied except under pathological situations. Consider a mode-schedule $\sigma \in \Sigma$, and recall that it has a finite sequence of mode-switchings. Consider a point $t \in [0, T)$ that is not the timing of a mode-switching. Then there exists $\delta > 0$ and $\bar{w} \in V$ such that for every $s \in [t, t+\delta)$, $v_\sigma(s) = \bar{w}$, namely $\bar{w}$ is the mode of σ throughout $s \in [t, t+\delta)$. By definition, $w(\sigma, s) \in \text{argmin}\{D_{\sigma,s,w} : w \in V\}$. By Assumption 1 and Equation (6), $D_{\sigma,s,w}$ is continuous in $s \in [s, s+\delta]$. Therefore, if $\text{argmin}\{D_{\sigma,s,w} : w \in V\}$ is a singleton then there exists $\delta_1 > 0$ such that for every $s \in [t, t+\delta_1)$, $\text{argmin}\{D_{\sigma,s,w} : w \in V\}$, hence having a constant value. On the other hand, consider the case where the set $\text{argmin}\{D_{\sigma,s,w} : w \in V\}$ consists of multiple points, and suppose without loss of generality that it has only two points, w_1 and w_2. If, for some $\delta_1 > 0$, $\text{argmin}\{D_{\sigma,s,w} : w \in V\} = \{w_1, w_2\}$ for all $s \in [t, t+\delta_1)$, then $w(\sigma, s)$ can we can choose $w(\sigma, s) = w_1$ $\forall s \in [t, t+\delta_1)$, a constant. Otherwise, by (6), the only way the statement of the assumption fails to be satisfied is if the function $p(s)^\top(f(x(s), w_1) - f(x(s), w_2))$ changes signs at an infinite sequence of points $\{s_j\}_{j=1}^\infty$ convergent to t from above. Since this function is differentiable, this situation is pathological. A similar situation arises if $w(\sigma, s)$ cannot be chosen to have a finite number of switching times in the interval $s \in [0, T]$.

The forthcoming analysis of the algorithm will be carried out in terms of optimality functions. The optimality condition that we consider is $D_\sigma = 0$, and hence it is natural to adopt the term D_σ as the optimality function. Regarding its upper-semi continuity, the question is which topology on the space of Lebesgue-measurable functions $v : [0, T] \to V$ is to be considered. What comes to mind is the topology induced by the L^1 norm or the L^2 norm since these norms are commonly used in the theory of optimal control. However, care must be taken when considering this optimality function on the Banach spaces $L^1\big([0, T], R\big)$ or $L^2\big([0, T], R\big)$, since it is not well defined there. To see this point, consider two functions $v : [0, T] \to V$ and $w : [0, T] \to V$ that have different values at a single fixed point $s \in [0, T)$ but identical values at all other points $t \in [0, T] \setminus \{s\}$. Denote the corresponding schedules by σ_v and σ_w, respectively. Certainly v and w are identical when viewed as elements in the Banach spaces $L^1\big([0, T], R\big)$ or $L^2\big([0, T], R\big)$, and $J(v) = J(w)$. On the other hand, the difference term in the RHS of (6) implies that $D_{\sigma_w, s, w(s)} = 0$ while there is no reason to expect that $D_{\sigma_v, s, w(s)} = 0$, and hence that $D_{\sigma_w} = D_{\sigma_v}$. This problem arises when a schedule is modified by inserting to it a new mode at a single point: The respective representations of the two schedules in $L^1\big([0, T], R\big)$ are identical and their respective state and costate trajectories are identical as well, but $D_{\sigma_v} \neq D_{\sigma_w}$ due to the fact that the last multiplicative term in the RHS of (6) depends on the function f at the particular point $s \in [0, T)$.

The above problem is circumvented when we restrict σ to Σ, or v to $\mathcal{V}$, since this requires $v_\sigma(t)$ to be left continuous and hence to have each one of its values on a positive-length interval. We will use the L^1 topology on the subspace $\mathcal{V} \subset L^1\big([0, T], R\big)$ even though it is not a complete space, but it serves our purposes concerning the algorithm's analysis.

We next establish the convergence of Algorithm 1. Our analysis requires a few preliminary results from the theory of perturbations of differential equations, whose proofs are based on several propositions made in [25], and hence are relegated to the appendix.

Given $\sigma \in \Sigma$, consider an interval $I := [s_1, s_2) \subset [0, T]$ such that σ has the same mode throughout I, namely for every $s \in I$, $v_\sigma(s) = v_\sigma(s_1)$. Given $w \in V$, denote by $\sigma_{s_1,w}(\gamma)$ the mode-sequence obtained from σ by changing the mode at every time $s \in [s_1, s_1 + \gamma]$ from $v_\sigma(s)$ to w, and consider the resulting cost function $J(\sigma_{s_1,w}(\gamma))$ as a function of $\gamma \in [0, s_2 - s_1]$.

Lemma 2: There exists a constant $K > 0$ such that, for every $\sigma \in \Sigma$, for every interval

$I = [s_1, s_2]$ as above, and for every $w \in V$, the function $J(\sigma_{s_1,w}(\cdot))$ is twice-continuously differentiable (C^2) on the interval $\gamma \in [0, s_2 - s_1]$; and for every $\gamma \in [0, s_2 - s_1]$, $|J(\sigma_{s_1,w}(\gamma))''| \leq K$ ("prime" indicates derivative with respect to γ).

Proof: Please see the appendix. ∎

We remark that the C^2 property of $J(\sigma_{s_1,w}(\cdot))$ is in force only as long as $v_\sigma(s) = v_\sigma(s_1)$ $\forall s \in [s_1, s_2]$. The second assertion of the lemma does not quite follow from the first one; the bound K holds for every such interval $[s_1, s_2]$, for every $\sigma \in \Sigma$, and for every $w \in V$.

Lemma 2 in conjunction with Lemma 1 can yield sufficient descent only in a local sense, as long as the same mode is scheduled according to σ. However, at mode-switching times $D_{\sigma,s}$ is no longer continuous in s, and hence Lemma 2 cannot be extended to intervals where $v(\cdot)$ does not have a constant value. Nonetheless the algorithm has a uniform sufficient descent, as will be proved with the aid of the following lemma.

Given a set $S \subset [0, T)$ we say that two schedules $\sigma_1 \in \Sigma$ and $\sigma_2 \in \Sigma$ are identical on $[0, T) \setminus S$ if $\forall\ \tau \in [0, T) \setminus S$, $v_{\sigma_1}(\tau) = v_{\sigma_2}(\tau)$.

Lemma 3: There exists $K > 0$ such that, for every subset $S \subset [0, T)$ comprised of a finite number of intervals, for every pair of schedules $\sigma_1 \in \Sigma$ and $\sigma_2 \in \Sigma$ that are identical on $[0, T) \setminus S$, for every $s \in [0, T) \setminus S$, and for every $w \in V$,

$$|D_{\sigma_1,s,w} - D_{\sigma_2,s,w}| \ \leq \ K\mu(S). \tag{22}$$

Proof: Please see the appendix. ∎

Note that Equation (22) does not hold true for $s \in S$, as can be seen from the difference term in the RHS of (6).

We now state and prove the paper's main result, namely the sufficient-descent property of the algorithm.

Proposition 1: Fix $\eta \in (0, 1)$ and $\alpha \in (0, \eta)$. There exists a constant $c > 0$ such that, for every $\sigma \in \Sigma$ satisfying $D_\sigma < 0$, for every $\lambda \in [0, \mu(S_{\sigma,\eta})]$ such that $\lambda \leq c|D_\sigma|$, and for every set $S \subset 2^{S_{\sigma,\eta}}$ comprised of a finite union of disjoint intervals such that $\mu(S) = \lambda$,

$$J(\boldsymbol{\sigma}(S)) - J(\sigma) \ \leq \ \alpha\lambda D_\sigma. \tag{23}$$

Proof: Consider $\sigma \in \Sigma$ and an interval $I := [s_1, s_2)$ such that $\forall s \in I$, $v_\sigma(s) = v_\sigma(s_1)$ and $w(\sigma, s) = w(\sigma, s_1)$. By Lemma 2, for every $w \in V$, the function $J(\sigma_{s_1,w}(\gamma))$ is C^2 in

$\gamma \in [0, s_2 - s_1)$, and by the first equality in the RHS of (6), $J(\sigma_{s_1,w}(0))' = D_{\sigma,s_1,w}$. Fix $a \in (\frac{\alpha}{\eta}, 1)$. Consider $w \in V$ such that $D_{\sigma,s_1,w} < 0$. By Lemma 2 and Corollary 1 (Equation (16)) there exists $\xi > 0$ such that, for every $\gamma \geq 0$ such that $\gamma \leq \min\{-\xi D_{\sigma,s_1,w}, s_2 - s_1\}$,

$$J(\sigma_{s_1,w}(\gamma)) - J(\sigma) = J(\sigma_{s_1,w}(\gamma)) - J(\sigma_{s_1,w}(0)) \leq -\alpha\gamma|J(\sigma_{s_1,w}(0))'| = a\gamma D_{\sigma,s_1,w}. \quad (24)$$

Furthermore, ξ does not depend on the mode-schedule σ, the interval $I = [s_1, s_2)$, or the mode $w \in V$ as long as $D_{\sigma,s_1,w} < 0$.

Next, by Lemma 3 there exists a constant $K > 0$ such that, for every set $S \subset [0,T]$ consisting of the finite union of intervals, for every pair of schedules $\sigma_1 \in \Sigma$ and $\sigma_2 \in \Sigma$ that are identical on $[0,T) \setminus S$, for every point $s \in [0,T) \setminus S$, and for every $w \in V$,

$$|D_{\sigma_1,s,w} - D_{\sigma_2,s,w}| \leq K\mu(S). \quad (25)$$

Fix $c > 0$ such that

$$c < \min\Big\{\frac{2}{aK}(a\eta - \alpha), \frac{\eta}{2K}\Big\}; \quad (26)$$

we next prove the assertion of the proposition for this c. Fix $\sigma \in \Sigma$ such that $D_\sigma < 0$, fix $\lambda \in [0, \mu(S_{\sigma,\eta})]$, and consider a set $S \subset S_{\sigma,\eta}$ consisting of the finite union of disjoint intervals such that $\mu(S) = \lambda$. Denote the intervals whose union comprises S by I_j, $j = 1, \ldots, m$, in increasing order, and let $I_j = [s_{1,j}, s_{2,j})$ for some $s_{1,j} \in [0,T)$ and $s_{2,j} \in (s_{1,j}, T)$. Denote the length of I_j by $\gamma_j := s_{2,j} - s_{1,j}$, then $\lambda = \mu(S) = \sum_{j=1}^m \gamma_j$. By subdividing the intervals I_j, $j = 1, \ldots, m$, if necessary, we can ensure $\forall\ j = 1, \ldots, m$, that: (i) $\gamma_j < -\frac{1}{2}\xi\eta D_\sigma$, and (ii) $\forall\ s \in I_j$, $v_\sigma(s) = v_\sigma(s_{1,j})$ and $w(\sigma, s) = w(\sigma, s_{1,j})$. Observe that $s_{1,j} \in S_{\sigma,\eta}$ and hence (by definition of the latter set) $D_{\sigma,s_{1,j}} \leq \eta D_\sigma$.

Next, recall that $\mu(S) = \lambda$, and that the mode-schedule $\boldsymbol{\sigma}(S)$ is obtained from σ by changing the mode $v_\sigma(s)$ to the mode $w(\sigma, s)$, for every $s \in S$. Let us define a sequence of intermediate mode-schedules, σ^j, $j = 0, 1, \ldots, M$, in the following way. $\sigma^0 = \sigma$, and for every $j = 1, \ldots, m$, $\sigma^j = \sigma^{j-1}_{s_{1,j},w(\sigma,s_{1,j})}(\gamma_j)$; in words, σ^j is obtained from σ^{j-1} by replacing $v_{\sigma^{j-1}}(s)$ by $w(\sigma, s_{1,j})$ $\forall\ s \in I_j$. Recall that $w(\sigma, s) = w(\sigma, s_{1,j})$ $\forall s \in I_j$, and hence we observe that σ^j is obtained from σ via changing the mode $v_\sigma(s)$ by $w(\sigma, s)$ $\forall\ s \in \cup_{i=1}^j I_i$. In particular, $\sigma^m = \boldsymbol{\sigma}(S)$.

Suppose that $\lambda \leq -cD_\sigma$; we next establish Equation (23), and this will complete the proof. To prove (23) we first show that for every $j = 1, \ldots, m$,

$$J(\sigma^j) - J(\sigma^{j-1}) \leq a\gamma_j\eta D_\sigma + aK\gamma_j \sum_{i=1}^{j-1} \gamma_i. \quad (27)$$

The case $j=1$ follows from Equation (24) with $I=I_1=[s_{1,1},s_{1,2})$, $w=w(\sigma,s_{1,1})$, and $\gamma=\gamma_1$. To see this recall that by definition $D_{\sigma,s_{1,1},w(\sigma,s_{1,1})}=D_{\sigma,s_{1,1}}$, by assumption $s_{1,1}\in S_{\sigma,\eta}$ and hence $D_{\sigma,s_{1,1}}<\eta D_\sigma$, and by assumption $\gamma_1<-\frac{1}{2}\xi\eta D_\sigma$; consequently $\gamma_1\leq-\frac{1}{2}\xi\eta D_\sigma<-\frac{1}{2}\xi D_{\sigma,s_{1,1}}=-\frac{1}{2}\xi D_{\sigma,s_{1,1},w(\sigma,s_{1,1})}$, and hence the conditions for (24) are satisfied. Applying (24) and recalling that $\sigma_{s_{1,1},w(\sigma,s_{1,1})}=\sigma^1$ and $D_{\sigma,s_{1,1},w(\sigma,s_{1,1})}=D_{\sigma,s_{1,1}}\leq\eta D_\sigma$, we obtain that $J(\sigma^1)-J(\sigma)\leq a\gamma_1\eta D_\sigma$. But this is (27) with $j=1$.

Consider next $j=2,\ldots,m$. Recall that σ^j is obtained from σ^{j-1} by replacing the mode $v_{\sigma^{j-1}}(s)$ by $w(\sigma,s)$ $\forall\ s\in I_j=[s_{1,j},s_{2,j})$. We next apply Equation (24) with σ^{j-1}, $I=I_j$, and $w=w(\sigma,s_{1,j})$. The conditions required for (24) are: (i) $D_{\sigma^{j-1},s_{1,j},w(\sigma,s_{1,j})}<0$, and (ii) $\gamma_j\leq-\xi D_{\sigma^{j-1},s_{1,j},w(\sigma,s_{1,j})}$; we now ascertain them. Recall that σ^{j-1} is obtained from σ by replacing $v_\sigma(s)$ by $w(\sigma,s)$ $\forall\ s\in\cup_{i=1}^{j-1}I_i$. Since $s_{1,j}\notin\cup_{i=1}^{j-1}I_i$, Equation (25) can be applied to yield the following inequality,

$$|D_{\sigma^{j-1},s_{1,j},w(\sigma,s_{1,j})}-D_{\sigma,s_{1,j},w(\sigma,s_{1,j})}|\ \leq\ K\mu(\cup_{i=1}^{j-1}I_i)\ =\ K\sum_{i=1}^{j-1}\gamma_i. \tag{28}$$

Recall that $\sum_{i=1}^{j}\gamma_i\leq\sum_{i=1}^{m}\gamma_i=\lambda$; and that $\lambda\leq-cD_\sigma$ by assumption; $c<\frac{\eta}{2K}$ by (26); $D_{\sigma,s_{1,j},w(\sigma,s_{1,j})}=D_{\sigma,s_{1,j}}$ by definition; and $s_{1,j}\in S_{\sigma,\eta}$ (by assumption) and hence $D_{\sigma,s_{1,j}}\leq\eta D_\sigma$. All of these inequalities, together with (28), imply that

$$D_{\sigma^{j-1},s_{1,j},w(\sigma,s_{1,j})}\ <\ \frac{1}{2}\eta D_\sigma\ <\ 0. \tag{29}$$

Equation (29) ascertains condition (i) for (24). As for condition (ii), by definition of the intervals I_j we have that $\gamma_j<-\frac{1}{2}\xi\eta D_\sigma$, and by (29), $-\frac{1}{2}\xi\eta D_\sigma\leq-\xi D_{\sigma^{j-1},s_{1,j},w(\sigma,s_{1,j})}$; this is condition (ii) for (24).

Equation (24) now can be applied with σ^{j-1}, $I=I_j=[s_{1,j},s_{2,j})$, and $w=w(\sigma,s_{1,j})$; it yields the following inequality,

$$J(\sigma^{j-1}_{s_{1,j},w(\sigma,s_{1,j})}(\gamma_j))-J(\sigma^{j-1})\ \leq\ a\gamma_jD_{\sigma^{j-1},s_{1,j},w(\sigma,s_{1,j})}. \tag{30}$$

Recall that $\sigma^{j-1}_{s_{1,j},w(\sigma,s_{1,j})}(\gamma_j)=\sigma^j$; now an application of (30), (28), and the fact that $D_{\sigma,s_{1,j},w(\sigma,s_{1,j})}=D_{\sigma,s_{1,j}}$, yield (27) after some straightforward algebra.

Finally, recall that $\sigma^m=\boldsymbol{\sigma}(S)$ and $\mu(S)=\lambda=\sum_{j=1}^{m}\gamma_j$. Sum up (27) over $j=1,\ldots,m$, to obtain,

$$J(\boldsymbol{\sigma}(S))-J(\sigma)\ \leq\ a\eta D_\sigma\lambda+aK\sum_{j=1}^{m}\big(\gamma_j\sum_{i=1}^{j-1}\gamma_i\big). \tag{31}$$

But

$$\sum_{j=1}^{m}\big(\gamma_j\sum_{i=1}^{j-1}\gamma_i\big) = \sum_{i,j=1,i\neq j}^{m}\gamma_i\gamma_j \leq \frac{1}{2}(\sum_{j=1}^{m}\gamma_j)^2 = \frac{1}{2}\lambda^2,$$

and hence, and by (31),

$$J(\boldsymbol{\sigma}(S)) - J(\sigma) \ \leq \ a\eta D_\sigma\lambda + \frac{1}{2}aK\lambda^2. \tag{32}$$

By assumption $\lambda \leq -cD_\sigma$, by (26) $c < \frac{2}{aK}(a\eta-\alpha)$, and by assumption $\alpha - a\eta < 0$; all of this, together with (32), yield the inequality in (23) and hence completes the proof. ■

Note that the the choice of α in Proposition 1 is restricted to the interval $(0,\eta)$, and not to the interval $(0,1)$ which is standard in Armijo-based algorithms. η can be anywhere in the interval $(0,1)$, and its choice reflects on the following balance. On one hand, larger η permits a larger value of α, with the possible result of greater decrease in J in the iterations of the algorithm. On the other hand, larger η would limit the sets $S_{\sigma,\eta}$ thereby potentially restricting the step size and hence the descent in J. The choice of η and α has to be done in an ad-hoc way.

As a result of Proposition 1, the algorithm converges in the following way.

Proposition 2: Suppose that Algorithm 1 computes a sequence of mode-schedules $\sigma_k \in \Sigma$, $k = 1, 2, \ldots$.

1) The following limit holds,

$$\limsup_{k\to\infty} D_{\sigma_k} \ = \ 0. \tag{33}$$

2) Suppose that $v \in \mathcal{V}$ is an accumulation point, in the L^1 norm, of the sequence $\{v_{\sigma_k}\}_{k=1}^{\infty}$, and let $\sigma \in \Sigma$ be its corresponding schedule so that $v = v_\sigma$. Then

$$D_\sigma \ = \ 0. \tag{34}$$

Proof: Let $\{\sigma_k\}_{k=1}^{\infty}$ be a sequence of mode-schedules computed by the algorithm. By Step 4 of the algorithm, $\sigma_{k+1} := \boldsymbol{\sigma}(\mathbf{S}(\boldsymbol{\lambda}(\sigma_k)))$, where $\mathbf{S} : [0, \mu(S_{\sigma_k,\eta})] \rightarrow 2^{S_{\sigma_k,\eta}}$ is the point-to-set mapping underscoring Step 3 and Step 4. By Equation (23),

$$J(\sigma_{k+1}) - J(\sigma_k) \ \leq \ \alpha\boldsymbol{\lambda}(\sigma_k)D_{\sigma_k}. \tag{35}$$

Since $D_{\sigma_k} \leq 0$, this implies that the algorithm is of a descent type, namely, $J(\sigma_{k+1}) \leq J(\sigma_k)$ for all $k = 1, 2, \ldots$. By an application of the Bellman-Gronwall inequality it is readily seen that there exists a constant $E \in R$ such that for every $\sigma \in \Sigma$, $J(\sigma) \geq E$; hence, and by (35) and the fact that the algorithm always yields descent, it follows that

$$\lim_{k\to\infty} \boldsymbol{\lambda}(\sigma_k) D_{\sigma_k} \;=\; 0. \tag{36}$$

We now prove Equation (33). Suppose, for the sake of contradiction, that (33) is not true. Then there exists $\epsilon > 0$ and a positive integer k_1 such that, for every $k \geq k_1$,

$$D_{\sigma_k} \;\leq\; -\epsilon. \tag{37}$$

By (36), this implies that

$$\lim_{k\to\infty} \boldsymbol{\lambda}(\sigma_k) \;=\; 0. \tag{38}$$

Recall the definition of λ_j made just before Equation (20): $\lambda_j = \beta^j \mu(S_{\sigma,\eta})$, and the set $\{\lambda_j : j = 0, 1, \ldots\}$, is the set of candidates for the Armijo step size as defined by (20). Similarly in (21), where the Armijo step size is defined via $\boldsymbol{\lambda}(\sigma_k) := \lambda_{j(\sigma_k)}$. The search for this Armijo step sizes starts at its largest-possible value, which is $\mu(S_{\sigma_k,\eta})$. Therefore, and by Proposition 1 and Equations (37) and (38), there exists $k_2 \geq k_1$ such that, for every $k \geq k_2$,

$$\boldsymbol{\lambda}(\sigma_k) \;=\; \mu(S_{\sigma_k,\eta}). \tag{39}$$

This, in conjunction with Step 4, imply that σ_{k+1} is obtained from σ_k by changing the mode, at every $s \in S_{\sigma_k,\eta}$, from $v_{\sigma_k}(s)$ to $w(\sigma_k, s)$, while leaving intact all the other modes $v_{\sigma_k}(s)$ for every $s \in [0,T) \setminus S_{\sigma_k,\eta}$. This results in eliminating the more negative part of the graph of $D_{\sigma_k,s}$ as a function of s, as illustrated in Figure 2. We next use this argument to prove that

$$\liminf_{k\to\infty} \big(D_{\sigma_{k+1}} - \eta D_{\sigma_k}\big) \;\geq\; 0. \tag{40}$$

Let $C^0\big([0,T]; R^n\big)$ denote the space of continuous functions $x : [0,T] \to R^n$, and consider the mapping $\Gamma_1 : \mathcal{V} \to C^0\big([0,T]; R^n\big)$, taking $v \in \mathcal{V}$ to $x \in C^0\big([0,T]; R^n\big)$ via Equation (1). By lemma 5.6.7 in [25], this mapping is uniformly continuous on $\mathcal{V}$ considering the L^1 topology on $\mathcal{V}$ and the L^∞ topology on $C^0\big([0,T]; R^n\big)$. In a similar way, the mapping $\Gamma_2 : \mathcal{V} \to C^0\big([0,T]; R^n\big)$, taking $v \in \mathcal{V}$ to $p \in C^0\big([0,T]; R^n\big)$ via Equations (1) and (6), is uniformly continuous on $\mathcal{V}$ considering the L^1 topology on $\mathcal{V}$ and the L^∞ topology on $C^0\big([0,T]; R^n\big)$. Let $x_k(\cdot)$ and $p_k(\cdot)$ denote the state trajectory and costate trajectory associated with v_{σ_k}, $k = 1, 2, \ldots$, via Equations (1) and (6), respectively. By (38) we have that $\lim_{k\to\infty} ||v_{\sigma_{k+1}} - v_{\sigma_k}||_{L^1} = 0$, and therefore,

$$\lim_{k\to\infty} ||x_{k+1} - x_k||_{L^\infty} \;=\; 0, \qquad \text{and} \quad \lim_{k\to\infty} ||p_{k+1} - p_k||_{L^\infty} \;=\; 0. \tag{41}$$

Consequently, for every $w \in V$,

$$\lim_{k\to\infty} \max_{s\in[0,T]} |p_{k+1}(s)^T f(x_{k+1}(s), w) - p_k(s)^T f(x_k(s), w)| \;=\; 0. \tag{42}$$

Recall that, by Step 4 of the algorithm, σ_{k+1} is obtained from σ_k by changing $v_{\sigma_k}(s)$ to $w(\sigma_k, s)$ $\forall s \in S_{\sigma_k,\eta}$, while keeping intact $v_{\sigma_k}(s)$ $\forall s \in [0,T)\backslash S_{\sigma_k,\eta}$. Consider now the relationship between $D_{\sigma_{k+1},s}$ and $D_{\sigma_k,s}$ for every $s \in [0,T)$. First, consider $s \in [0,T) \setminus S_{\sigma_k,\eta}$. By (19), $D_{\sigma_k,s} \geq \eta D_{\sigma_k}$; by Step 4, $v_{\sigma_{k+1}}(s) = v_{\sigma_k}(s)$; and by (7) and (43), we have that

$$\liminf_{k\to\infty} \inf_{s\in[0,T)\backslash S_{\sigma_k,\eta}} \left(D_{\sigma_{k+1},s} - \eta D_{\sigma_k}\right) \;\geq\; 0. \tag{43}$$

Next, consider $s \in S_{\sigma_k,\eta}$. By Step 4 of the algorithm, $v_{\sigma_{k+1}}(s) = w(\sigma_k, s)$, while by definition, $w(\sigma_k, s)$ minimizes the term $D_{\sigma_k,s,w}$ over all $w \in V$. Therefore, and by (6) and (42), we have that

$$\liminf_{k\to\infty} \inf_{s\in S_{\sigma_k,\eta}} D_{\sigma_{k+1},s} \;=\; 0. \tag{44}$$

Now Equation (40) follows from (43) and (44).

Since $\eta \in (0,1)$ and $D_{\sigma_k} \leq 0$ for all $k = 1,2,\ldots$, (40) implies that $\lim_{k\to\infty} D_{\sigma_k} = 0$. However, this contradicts Equation (37) and hence yields the proof of Equation (33).

Consider next the second part of the proposition. Let $\sigma_k \in \Sigma$, $k = 1,2,\ldots$, be a mode-sequence computed by the algorithm, and let $v_{\sigma_k} \in \mathcal{V}$ denote the corresponding input-control functions. Let $v \in \mathcal{V}$ be an accumulation point of the sequence $\{v_{\sigma_k}\}_{k=1}^{\infty}$ in the L^1 norm, and suppose that

$$\lim_{j\to\infty} ||v_{\sigma_{k(j)}} - v||_{L^1} \;=\; 0 \tag{45}$$

for some subsequence $\{v_{\sigma_{k(j)}}\}_{j=1}^{\infty}$. Let $\sigma \in \Sigma$ be the mode-sequence corresponding to v, such that $v = v_\sigma$.

Our objective is to prove Equation (34). Suppose the contrary, namely that $D_\sigma < 0$, for the sake of contradiction. We first establish the following equation:

$$\limsup_{j\to\infty} D_{\sigma_{k(j)}} \;\leq\; D_\sigma. \tag{46}$$

Since $\sigma \in \Sigma$, the function $v_\sigma : [0,T] \to V$ is piecewise constant, and it has its values changed a finite number of times (say, N) in the interval $(0,T)$. Denoting by τ_i, $i = 1,\ldots,N$ the switching times in increasing order, and further defining $\tau_0 = 0$ and $\tau_{N+1} = T$, we have, for every $i = 1,\ldots,N+1$, that (i) $\tau_i - \tau_{i-1} > 0$, and (ii) $v(s) = v^i$ for some $v^i \in V$

throughout the interval $[\tau_{i-1},\tau_i)$. Now for every $j=1,2,\ldots$, define the set $\Lambda_j\in[0,T)$ by $\Lambda_j=\{s\in[0,T): v_{\sigma_{k(j)}}(s)\neq v_\sigma(s)\}$. By (45), $\forall\ \epsilon>0\ \exists\ j_1>0$ such that for every $j\geq j_1$, $\mu(\Lambda_j)<\epsilon$. If $\epsilon<\min_{i=1,\ldots,N+1}\{\tau_i-\tau_{i-1}\}$, then for every $i=1,\ldots,N+1$, there exists $s\in[\tau_{i-1},\tau_i)$ such that

$$v_{\sigma_{k(j)}}(s)=v_\sigma(s). \tag{47}$$

This means that, for every $j\geq j_1$, every mode scheduled according to σ at a time s is also scheduled according to $\sigma_{k(j)}$ at some time $\bar{s}\in(s-\epsilon,s+\epsilon)$. Let $x_{k(j)}(\cdot)$ and $x(\cdot)$ denote the state trajectories associated with $v_{k(j)}(\cdot)$ and $v(\cdot)$, respectively, via Equation (1), and let $p_{k(j)}(\cdot)$ and $p(\cdot)$ denote the costate trajectories associated with $v_{k(j)}$ and v, respectively, via (1) and (6). By (45) and Lemma 5.6.7 in [25], we have that

$$\lim_{j\to\infty}||x_{k(j)}-x||_{L^\infty}=0, \qquad \text{and} \qquad \lim_{j\to\infty}||p_{k(j)}-p||_{L^\infty}=0. \tag{48}$$

Now by (6), (47), and (48),

$$\lim_{j\to\infty}\sup_{s\in[0,T]}\Big|p_{k(j)}(s)^Tf(x_{k(j)}(s),v_\sigma(s))-p(s)^Tf(x(s),v_\sigma(s))\Big| \ = \ 0. \tag{49}$$

By the definition of D_σ via the LHS of Equation (7), this implies (46).[1]

Proposition 1, together with (46) and the assumption that $D_\sigma<0$, imply the following three statements in the same way Equations (38) - (40) were proved: (i) the following limit holds,

$$\lim_{j\to\infty}\boldsymbol{\lambda}(\sigma_{k(j)}) \ = \ 0; \tag{50}$$

(ii) there exists $j_3>0$ such that for every $j\geq j_3$,

$$\boldsymbol{\lambda}(\sigma_{k(j)}) \ = \ \mu(S_{\sigma_{k(j)},\eta}); \tag{51}$$

and

$$\liminf_{j\to\infty}\big(D_{\sigma_{k(j)+1}}-\eta D_{\sigma_{k(j)}}\big) \ \geq \ 0. \tag{52}$$

Now (50) and (45) imply that

$$\lim_{j\to\infty}||v_{\sigma_{k(j)+1}}-v||_{L^1} \ = \ 0. \tag{53}$$

[1]Note that the argument for proving (46) requires the assumption that $v\in\mathcal{V}$, and may break down without it. Also, observe that it is possible that, for arbitrarily-large j, a mode present in $v_{k(j)}$ at a time s may not be present in v_σ at any time near s; consequently a sharp inequality in (46) is possible, namely $\lim_{j\to\infty}D_{\sigma_{k(j)}}<D_\sigma$.

Consequently, Equations (46) and (50) - (53) apply to $k(j)+1$ instead of $k(j)$. Repeating the argument, given $m \geq 1$, (46) and (50)-(53) hold true for every $k(j)+i$, $i=1,\ldots,m$, instead of $k(j)$. In particular, these extensions of (46) and (52) imply that

$$\limsup_{j\to\infty} D_{\sigma_{j+m}} \;\leq\; D_{\sigma} \tag{54}$$

and

$$\liminf_{j\to\infty} \left(D_{\sigma_{k(j)+m}} - \eta^m D_{\sigma_{k(j)}}\right) \;\geq\; 0. \tag{55}$$

But the insertion gradient D_σ is uniformly bounded from below, namely there exists a constant $E_1 < 0$ such that $\forall\ \sigma \in \Sigma$, $D_\sigma \geq E_1$. Since $\eta \in (0,1)$, Equations (55) for every positive integer m is not compatible with (54). This contradiction yields the proof of (34) and hence completes the proof of the proposition. ■

Remark 3: The assertion in part 2 of Proposition 2 may be vacuous since there are no guarantees that a bounded sequence of control functions $v_{\sigma_k} \in \mathcal{V}$ have an accumulation point in the L^1 norm. If it does, however, then the optimality condition is guaranteed via Equation (36). Generally, Equation (35) is the most we can say about the optimality function of such a sequence. The stronger condition of $\lim_{k\to\infty} D_{\sigma_k} = 0$ does not necessarily hold. This is due to the nature of the optimality function, which is not upper-semi continuous and may not be well defined on functions $v \in L^1([0,T];V)\setminus\mathcal{V})$. Alternative optimality functions such as $\int_0^T D_{\sigma,s}ds$ would not have this problem while having the same zero-set as D_σ, but we chose to develop our theory with the former one since it is more natural. From a practical standpoint, the sequence $\{J(\sigma_k)\}_{k=1}^{\infty}$ is monotone non-increasing, and hence, if the iteration sequence $\{\sigma_k\}_{k=1}^{\infty}$ has a subsequence convergent to a minimum point (schedule), the entire sequence $\{J(\sigma_k)\}_{k=1}^{\infty}$ will converge to the minimal value. All of this suggests that the result stated in part 1 of Proposition 2 adequately characterizes the asymptotic convergence of Algorithm 1.

IV. Simulation Examples

We tested the algorithm on the double-tank system shown in Figure 3. The input to the system, v, is the inflow rate to the upper tank, controlled by the valve and having two possible values, $v_1 = 1$ and $v_2 = 2$. x_1 and x_2 are the fluid levels at the upper tank and lower tank, respectively,

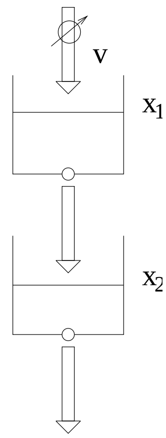

Fig. 3. Two-tank system

as shown in the figure. According to Toricelli's law, the state equation is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} v - \sqrt{x_1} \\ \sqrt{x_1} - \sqrt{x_2} \end{pmatrix}, \tag{56}$$

with the (chosen) initial condition $x_1(0) = x_2(0) = 2.0$. Notice that both x_1 and x_2 must satisfy the inequalities $1 \leq x_i \leq 4$, and if $v = 1$ indefinitely then $\lim_{t\to\infty} x_i = 1$, while if $v = 2$ indefinitely then $\lim_{t\to\infty} x_i(t) = 4$, $i = 1, 2$.

The objective of the optimization problem is to have the fluid level in the lower tank track the given value of 3.0, and hence we chose the performance criterion to be

$$J = 2\int_0^T \left(x_2 - 3\right)^2 dt, \tag{57}$$

for the final-time $T = 20$. The various integrations were computed by the forward-Euler method with $\Delta t = 0.01$. For the algorithm we chose the parameter-values $\alpha = \beta = 0.5$ and $\eta = 0.6$, and we ran it from the initial mode-schedule associated with the control input $v(t) = 1 \;\forall\; t \in [0, 10]$ and $v(t) = 2 \;\forall\; t \in (10, 20]$.

Results of a typical run, consisting of 100 iterations of the algorithm, are shown in Figures 4-6. Figure 4 shows the control computed after 100 iterations, namely the input control v associated with σ_{100}. The graph is not surprising, since we expect the optimal control initially to consist of $v = 2$ so that x_2 can rise to a value close to 3, and then to enter a sliding mode in order for x_2 to maintain its proximity to 3. Figure 5 shows the corresponding state trajectories $x_1(t)$ and $x_2(t)$, $t \in [0, T]$, associated with σ_{100}, where the jagged curve is of x_1 while the smoother

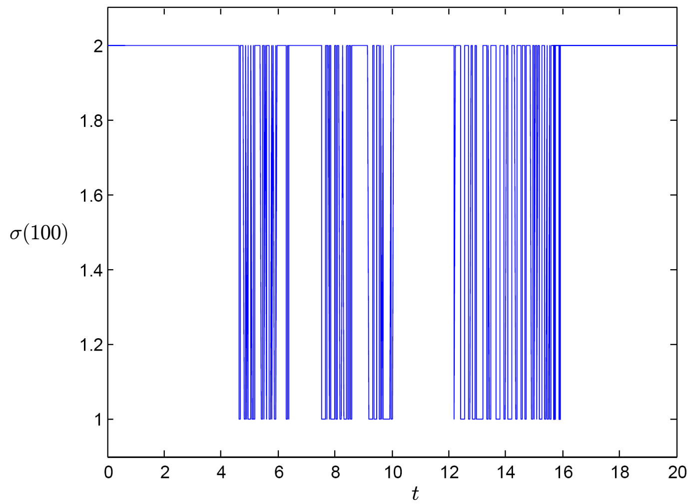

Fig. 4. Control (schedule) obtained after 100 iterations

curve is of x_2. Figure 6 depicts the graph of $J(\sigma_k)$ as a function of the iteration count k, and we discern a rapid descent at the early stage of the algorithm run.

It could be argued that Figure 4 indicates a rather slow buildup of the sliding mode, and consequently the tracking shown in Figure 5 is not very tight. As a matter of fact, [22] obtained tighter tracking by solving the relaxed problem via an alternative technique. We will address such a potential critique with the following three arguments: First, the objective of our algorithm is to solve an optimization problem and not a tracking problem, and the final cost that it obtains is barely distinguishable from that computed by [22]. Second, the most salient feature of descent algorithms with Armijo step sizes is not in their asymptotic convergence rate but rather in their initial descent rate, and this is clearly demonstrated in Figure 6. Third, in contrast with techniques for the relaxed problem, our algorithm has natural extensions to problems with minimum dwell-time constraints. We next present these arguments in detail.

The potential significant merit of our algorithm is evident from Figure 6. Depicting the graph of the cost criterion $J(\sigma_k)$ as a function of the iteration count k, it exhibits a rapid descent of the algorithm in a few iterations at the early stage of its execution. The initial schedule, σ_1, is far away from the minimum and its cost is $J(\sigma_1) = 70.90$, but $J(\sigma_k)$ declines to below 7.0 ($J(\sigma_3) = 6.34$) in just two iterations, thereafter remaining flat until $J(\sigma_{100}) = 4.85$ at the final

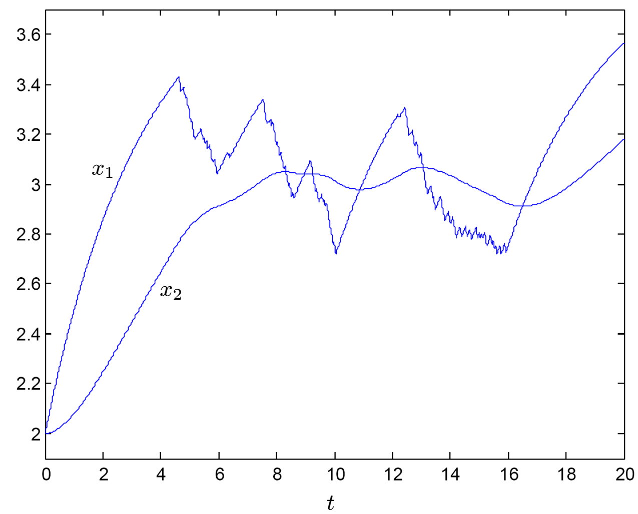

Fig. 5. x_1 and x_2 at σ_{100}

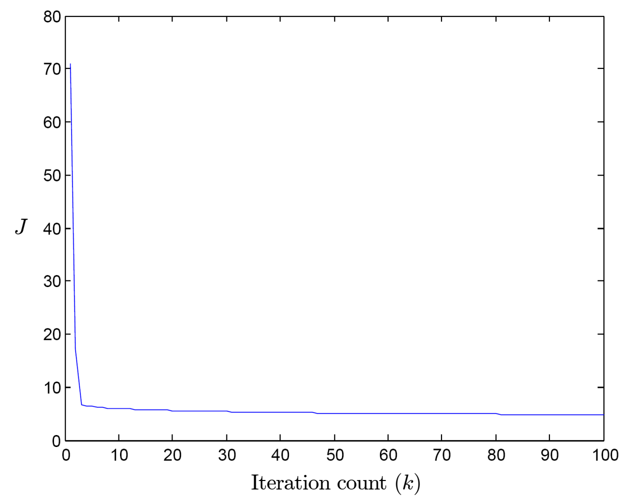

Fig. 6. Cost criterion vs. iteration count

point. An extension of the run to 200 iterations (not shown in the graph) yielded $J(\sigma_{200}) = 4.78$. With respect to the optimal cost computed in [22], $J_{opt} = 4.74$, our algorithm achieved over 97.5$\%$ of the total descent in those two iterations, which required cpu time of 0.1 seconds to execute on a MacBookPro 7.1 running at 2.4 GHz on an Intel Core 2 Duo. The entire run of 100 iterations took cpu time of 5.16 seconds, and the 200-iteration run took 10.99 seconds. As mentioned earlier, this run was made with $\Delta t = 0.01$, and hence a $2,000$-point discretization grid of the horizon interval $[0, T]$. To reduce the computing times of the algorithm, we increased Δt to 0.1 for the first 10 iterations. It took 3 iterations and 0.0091-seconds cpu time to obtain a decline from $J(\sigma_1) = 70.90$ to below 7.0 ($J(\sigma_4) = 6.78$), and 6 iterations and 0.018-seconds cpu time to obtain a value under 6.0 ($J(\sigma_7) = 5.95$). This suggests that adaptive-precision may play a role in reducing computing times in large problems.

All of this makes it clear that the salient feature of our algorithm is in its rapid initial approach towards the minimum points.

Concerning Figure 5 and the apparent oscillations of $x_2(t)$ about its target value of 3.0, most of the cost at σ_{100} is incurred during the period when $x_2(t)$ climbs to that value and not during the period when it oscillates around it (see Figure 5). To wit, we calculated the first time $x_2(t) = 3.0$ at $t = t_1 = 7.36$; the part of the cost in the interval $[0, t_1]$ is $2\int_0^{t_1}(x_2(t) - 3)^2 dt = 4.77$, while the remaining part is $2\int_{t_1}^{T}(x_2(t) - 3)^2 dt = 0.08$. Thus, over 98.3% of the total cost is incurred during the initial interval $[0, t_1]$, while the oscillatory behavior of x_2 about 3.0, noted only in the remaining interval $[t_1, T]$, plays a minor role in the total cost.

The presence of a singularity in the solution point of the relaxed problem may slow down the asymptotic convergence of our algorithm at its later stages, because it has to track the construction of the sliding modes. As a matter of fact we observed that after two iterations, at σ_3, the contribution of the interval $[0, t_1]$ (just before the sliding mode starts) to the total cost is $2\int_0^{t_1}(x_2(t) - 3)^2 dt = 4.79$ which is quite close to J_{opt}, while in subsequent iterations the main efforts of the algorithm are in constructing the sliding mode. This suggests that the algorithm would converge much faster for (a class of) problems whose solutions do not contain singularities or sliding modes.

To test this point we modified the problem to track a time-dependent curve, $r(t)$, defined as

$$r(t) \;=\; \begin{cases} 0.5, & t \in [0, 0.25T) \cup [0.5T, 0.75T) \\ 4.5, & i \in [0.25T, 0.5T) \cup [0.75T, T], \end{cases}$$

where we chose $T = 200$ to ensure that x_2 has enough time to reach its extreme values of 1.0 and 4.0. The algorithm was identical to the one described for the previous system except that $\eta = 0.75$. After 20 iterations the control is shown in Figure 7 and its corresponding graph of $x_2(t)$ is plotted in Figure 8. We believe that this is the optimal solution (or very close to it) since the tracking in Figure 8 seems to be as tight as possible. The optimality function, which started

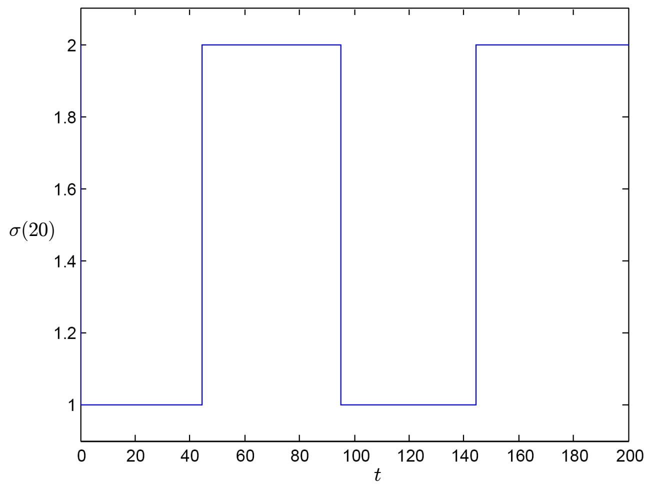

Fig. 7. Tracking $r(t)$: Control (schedule) obtained after 20 iterations

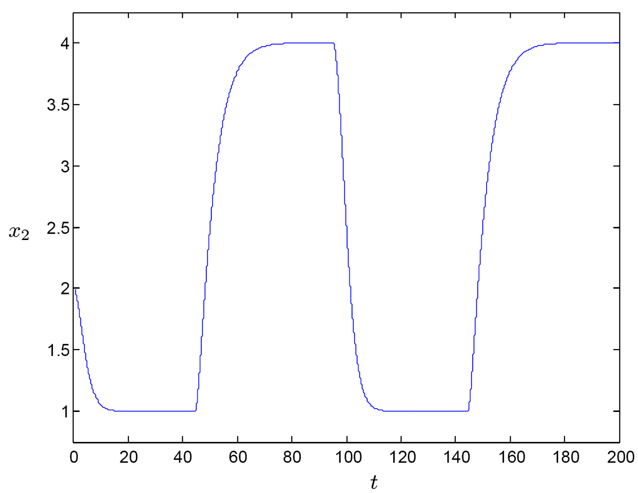

Fig. 8. Tracking $r(t)$: x_2 at σ_{20}

at $D_{\sigma_1} = -55.13$, ends at $D_{\sigma_{20}} = -0.062$, further affirming that σ_{20} indeed is very close to the minimum point. The graph of $J(\sigma_k)$ as a function of k is shown in Figure 9 and it exhibits rapid descent at the initial stage of the algorithm. As a matter of fact, with $J(\sigma_1) = 2,298.6$, $J(\sigma_5) = 343.3$, $J(\sigma_7) = 280.5$, and $J(\sigma_{20}) = 231.7$, over 94.6% of the total decrease is obtained in just 4 iterations, and 97.6% in 6 iterations.

The switching frequency often is limited for practical reasons, and the optimal switched-

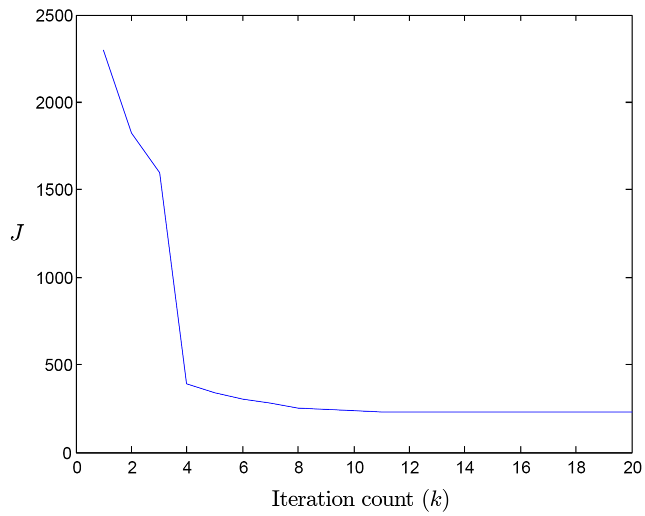

Fig. 9. Tracking $r(t)$: Cost criterion vs. iteration count

mode problem can reflect that either by adding a cost-penalty term to each switching, or by imposing lower-bound constraints on the dwell times of the modes. These formulations do not quite fall within the optimization framework discussed above, yet they cannot be ignored due to their relevance in applications. In the coming paragraphs we describe a heuristic approach to problems with minimum dwell times; the ideas are preliminary and only half-baked, and a more comprehensive analysis will be presented elsewhere.

Consider the problem of minimizing J as defined in (2) subject to the constraint that the dwell-time of each mode must not be less than a given constant $t_\delta > 0$. Recall that τ_i denotes the switching time between the ith mode and the $i+1st$ mode of a given schedule, then the dwell-time constraint has the form $\tau_{i+1} - \tau_i \geq t_\delta$, $i = 1, 2, \ldots$. Given a schedule $\sigma \in \Sigma$, the following procedure modifies it to satisfy the lower-bound constraint on the dwell times. Throughout its formal description we use the notation $\bar{\sigma}$ to refer to the running schedule-variable as it is being modified. Its final value is the output of the procedure.

Dwell-time constraint-compliance procedure.

Initialize: Set $\ell = 1$. Set $\bar{\sigma} = \sigma$.

1. If $\bar{\sigma}$ satisfies the dwell-time constraint then stop and exit. Otherwise, continue.

2. Compute $j := \min\{i \geq \ell : \tau_{i+1} - \tau_i < t_\delta\}$. Cancel all the switching times of $\bar{\sigma}$, τ_i,

$i = j+1, \ldots$, that are between τ_j and $\tau_j + t_\delta$. Set $\tau_{j+1} = \tau_j + t_\delta$.

3. Compute

$$w \ := \ \arg\min\{\int_{t_j}^{t_{j+1}} D_{\sigma,t,v}dt \ : \ v \in V\}. \tag{58}$$

Set to w the mode of $\bar{\sigma}$ throughout the interval $[t_j, t_{j+1})$.

4. Keep the rest of the schedule $\bar{\sigma}$ unchanged except for, if need be, renumbering the switching times as $\tau_j, \tau_{j+1}, \ldots$ in order to have them be consecutive.

5. Set $\ell := j+1$, and go to Step 1. $\square$

A few remarks are due.

1) Throughout the procedure we use σ (and not $\bar{\sigma}$) in Equation (58), to indicate the insertion gradient at the schedule with which we entered the procedure.
2) The use of Equation (58) to determine the mode in the interval $[\tau_j, \tau_{j+1})$ is underscored by the same principle as Algorithm 1, namely an attempt to minimize the optimality gap. In this regard, the use of the integral term is due to the minimum dwell-time constraint. Of course this would be effective only if t_δ is small enough. On the other hand, large values of t_δ may result in few switching times and hence trivialize the problem.
3) It is certainly possible to refine the result of the final run of the above procedure with a gradient-descent algorithm for adjusting the switching times while maintaining the dwell-time constraints and the sequence of modes.

It is reasonable to first solve the mode-switching problem without regard to the dwell-time constraint, and then use the procedure to guarantee that those constraints are satisfied. Alternatively, it is possible to embed the procedure in the cycles of an algorithm such as Algorithm 1; for example, run it after every given number of iterations. This approach may have the advantage of limiting the number of switching times and hence avoiding the sliding modes, which generally slows down Algorithm 1. We tested this idea on the problem of minimizing J as defined by (57), with the same initial point (schedule) as for the unconstrained problem. For this simulation experiment we chose $t_\delta = 0.25$, and we used the dwell-time procedure after every run of 10 iterations of Algorithm 1. After 100 iterations we obtained $J(\sigma_{100}) = 4.83$, essentially the same as for the unconstrained problem, where $J(\sigma_{100}) = 4.85$. The graph of σ_{100} is shown in Figure 10, and not surprisingly it does not have the chatter of the analogous graph in Figure 4. The

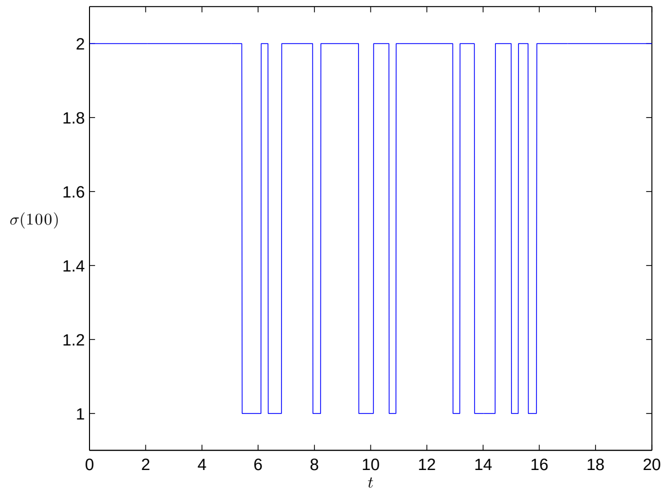

Fig. 10. Minimum dwell time: Control obtained after 100 iterations

associated state trajectories as well as the graph of $J(\sigma_k)$ (as a function of k) were quite similar to those in Figures 5 and 6 (respectively) for the unconstrained problem, and hence are not shown here.

V. Appendix

Proof of Lemma 2. The proof follows from Proposition 5.6.8 and Proposition 5.6.10 in [25], which are stated in a more general context. For detailed arguments, consider the following setting. Let $f_1 : R^n \to R^n$, $f_2 : R^n \to R^n$, and $L : R^n \to R$ be C^2 functions, and fix $T > 0$ and $x_0 \in R^n$. Given $\gamma \in [0, T]$, define the vector field

$$F(x,t;\gamma) := \begin{cases} f_1(x(t)), & \text{if } t \in [0,\gamma] \\ f_2(x(t)), & \text{if } t \in (\gamma, T], \end{cases}$$

and consider the differential equation $\dot{x}(t)) = F(x(t),t;\gamma)$ on the interval $t \in [0,T]$, with the initial condition $x(0) = x_0$. Since the equation depends on γ, we denote its solution by $x(t;\gamma)$. Define the performance function $J(\gamma)$ by $J(\gamma) := \int_0^T L(x(t;\gamma)dt$. Define the costate variable $p(t;\gamma)$ by the equation $\dot{p}(t;\gamma) = -\big(\frac{\partial F}{\partial x}(x,t;\gamma)\big)^\top p(t;\gamma) - \big(\frac{\partial L}{\partial t}(x;\gamma)\big)^\top$, *dot* denoting derivative with respect to t, with the boundary condition $p(T;\gamma) = 0$.

In the context of this paper, Lemma 2 amounts to the assertion that $J(\gamma)$ is C^2. Reference [15] proved (in a more general context) that $J(\gamma)$ is C^1 and the first derivative is given by

$$J'(\gamma) = p(\gamma;\gamma)^\top \big(f_1(x(\gamma;\gamma) - f_2(x(\gamma;\gamma))\big) \tag{59}$$

(see Proposition 2 2 and Proposition 3.1 there). Now the partial derivatives $\frac{\partial x}{\partial t}(\gamma;\gamma)$ and $\frac{\partial x}{\partial \gamma}(\gamma;\gamma)$ generally do not exist, but the total derivative $\frac{dx}{d\gamma}(\gamma;\gamma)$ exists and it is continuous. To see this note that for all $t \in [0,\gamma]$ $x(t;\gamma)$ satisfies the equation $\dot{z}(t) = f_1(z(t))$, and hence $\frac{dx}{d\gamma}(\gamma;\gamma) = \dot{z}(\gamma) = f_1(x(\gamma;\gamma)$, and the latter term is continuous by the assumption that $f_1(x)$ is C^2. In a similar way, the total derivative $\frac{dp}{d\gamma}(\gamma;\gamma)$ exists and it is continuous. To see this, [25] (Corollary 5.6.9) proves that for every $t \in (\gamma, t]$ the derivative term $\frac{\partial x}{\partial t}x(t;\gamma)$ is continuously differentiable in γ. Furthermore, by the costate equation the evolution of $p(t;\gamma)$ backwards in time depends only on the vector field f_2 but not on f_1, and therefore $\frac{dp}{d\gamma}(\gamma;\gamma) = \Big(-\big(\frac{\partial F}{\partial x}(x,\gamma;\gamma)\big)^\top p(\gamma;\gamma) - \big(\frac{\partial L}{\partial t}(x;\gamma)\big)^\top\Big)'$, "prime" denoting derivative with respect to γ; continuity follows by standard variational arguments on differentiability of differential equations (e.g., Corollary 5.6.9) and the C^2 assumptions on f_1, f_2, and L. All of this implies that term in the the RHS of Equation (59) is continuously differentiable thereby ascertaining that $J(\gamma)$ is twice continuously differentiable. $\blacksquare$

Proof of Lemma 3. Consider a set $S \subset [0,T)$ and schedules σ_1 and σ_2 as in the statement of the lemma. For $i = 1,2$, let $x_i(\cdot)$ and $p_i(\cdot)$ denote the state trajectory and costate trajectory, respectively, associated with v_{σ_i}. By Equations (1) and (5), and by Lemma 5.6.7 in [25] concerning Lipschitz continuity of solutions of differential equations, there exists $K_1 > 0$ such that, for all $S \subset [0,T)$, $\sigma_1 \in \Sigma$, and $\sigma_2 \in \Sigma$ as above,

$$||x_1 - x_2||_{L^\infty} \ \leq \ K_1||v_{\sigma_1} - v_{\sigma_2}||_{L^2}, \tag{60}$$

and

$$||p_1 - p_2||_{L^\infty} \ \leq \ K_1||v_{\sigma_1} - v_{\sigma_2}||_{L^2}. \tag{61}$$

For every $s \in [0,T) \setminus S$, $v_{\sigma_1}(s) = v_{\sigma_2}(s)$, and therefore, and by (6), for every $w \in V$,

$$\begin{aligned} & D_{\sigma_1,s,w} - D_{\sigma_2,s,w} \\ = \ & p_1(s)^T\Big(f(x_1(s),w) - f(x_1(s),v_{\sigma_1}(s))\Big) - p_2(s)^T\Big(f(x_2(s),w) - f(x_2(s),v_{\sigma_1}(s))\Big). \end{aligned} \tag{62}$$

Consequently, and by (60) and (61), there exists $K_2 > 0$ such that, for every $S \subset [0,T)$, $\sigma_1 \in \Sigma$, and $\sigma_2 \in \Sigma$ as above, and for every $w \in V$,

$$|D_{\sigma_1,s,w} - D_{\sigma_2,s,w}| \ \leq \ K_2||v_{\sigma_1} - v_{\sigma_2}||_{L^2}. \tag{63}$$

Since V is a finite set, and since $v_{\sigma_1}(\tau) = v_{\sigma_2}(\tau)\ \forall\tau \in [0,T) \setminus S$, there exists $K_3 > 0$ such that for all $S \subset [0,T)$ and $\sigma_1 \in \Sigma$ and $\sigma_2 \in \Sigma$ as above, $||v_{\sigma_1} - v_{\sigma_2}||_{L^2} \leq K_3\mu(S)$. This, together with (63), implies Equation (22) with $K := K_2K_3$. ■

REFERENCES

[1] S. Almér, S. Mariéthoz, and M. Morari. Sampled Data Model Predictive Control of a Voltage Source Inverter for Reduced Harmonic Distortion. *IEEE Control Systems Technology*, to appear (also, IEEE Early Access).

[2] S. Almér, S. Mariéthoz, and M. Morari. Robust tracking control of pulse-width modulated systems through multi-frequency averaging. *Proc. ACC*, Montreal, Canada, June 27-29, 2012.

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

[4] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest. A Gradient Descent Approach to Optimal Mode Scheduling in Hybrid Dynamical Systems. *Journal of Optimization Theory and Applications*, Vol. 136, pp. 167-186, 2008.

[5] S.C. Bengea and R. A. DeCarlo. Optimal control of switching systems. *Automatica*, Vol. 41, pp. 11-27, 2005.

[6] 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.

[7] R. Brockett. Stabilization of Motor Networks. *IEEE Conference on Decision and Control*, pp. 1484–1488, 1995.

[8] 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.

[9] T.M. Caldwell and T.D. Murphey. An Adjoint Method for Second-Order Switching Time Optimization. *Proc. 49th CDC*, Atlanta, Georgia, December 15-17, 2010.

[10] T.M. Caldwell and T.D. Murphey. Switching mode generation and optimal estimation with application to skid-steering. *Automatica*, Vol. 47, pp. 50-64, 2011.

[11] T. M. Caldwell and T. D. Murphey. Projection-Based Switched System Optimization: Absolute Continuity of the Line Search. —it Proc. 51st CDC, Maui, Hawaii, December 10-13, 3012.

[12] C.G. Cassandras, D.L. Pepyne, and Y. Wardi. Optimal Control for a Class of Hybrid Systems. *IEEE Transactions on Automatic Control*, Vol. 46, No. 3, pp. 398-415, 2001.

[13] X.C. Ding, Y. Wardi, D. Taylor, and M. Egerstedt. Optimization of Switched-Mode Systems with Switching Costs. *Proc. American Control Conference*, Seattle, Washington, June 11-13, 2008.

[14] M. Egerstedt. Behavior Based Robotics Using Hybrid Automata. *Lecture Notes in Computer Science: Hybrid Systems III: Computation and Control*, Springer Verlag, pp. 103-116, Pittsburgh, PA, March 2000.

[15] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-Time Optimization for Switched Systems. *IEEE Transactions on Automatic Control*, Vol. AC-51, No. 1, pp. 110-115, 2006.

[16] K. Flaßkamp, T. Murphey, and S. Ober-Blöbaum. Switching Time Optimization in Discretized Hybrid Dynamical Systems. *Proc. 51st CDC*, Maui, Hawaii, December 10-13, 2012.

[17] C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice Hall Series in Automatic Computation, Englewood Cliffs, New Jersey, 1971.

[18] K. Gokbayrak and C.G. Cassandras. Hybrid Controllers for Hierarchically Decomposed Systems. *Proc. 2000 Hybrid Systems Control Conf.*, pp. 117-129, 2000.

[19] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S.S. Sastry, R. Bajcsy, and C. Tomlin. A Numerical Method for the Optimal Control of Switched Systems. *Proc. 49th CDC*, Atlanta, Georgia, pp. 7519-7526, December 15-17, 2010.

[20] D. Hristu-Varsakelis. Feedback Control Systems as Users of Shared Network: Communication Sequences that Guarantee Stability. *IEEE Conference on Decision and Control*, pp. 3631–3636, Orlando, FL, 2001.

[21] B. Lincoln and A. Rantzer. Optimizing Linear Systems Switching. *IEEE Conference on Decision and Control*, pp. 2063–2068, Orlando, FL, 2001.

[22] R.T. Meyer, M.Zefran, and R.A. Decarlo. Comparison of Multi-Parametric Programming, Mixed-Integer Programming, Gradient Descent Based, Hybrid Minimum Principle, and the Embedding Approach on Six Published Hybrid Optimal Control Examples. Private Communication, 2012.

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

[24] E. Polak and Y. Wardi. A Study of Minimizing Sequences. *SIAM Journal on Control and Optimization*, Vol. 22, No. 4, pp. 599-609, 1984.

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

[26] H. Rehbinder and M. Sanfirdson. Scheduling of a Limited Communication Channel for Optimal Control. *IEEE Conference on Decision and Control*, Sidney, Australia, Dec. 2000.

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

[28] 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*, pp. 466-481, Prague, The Czech Republic, 2003.

[29] M.S. Shaikh and P.E. Caines. Optimality Zone Algorithms for Hybrid Systems Computation and Control: From Exponential to Linear Complexity. *Proc. IEEE Conference on Decision and Control/European Control Conference*, pp. 1403-1408, Seville, Spain, December 2005.

[30] M.S. Shaikh and P.E. Caines. On the Hybrid Optimal Control Problem: Theory and Algorithms. *IEEE Trans. Automatic Control*, Vol. 52, pp. 1587-1603, 2007.

[31] H.J. Sussmann. A Maximum Principle for Hybrid Optimal Control Problems. *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 425-430, Phoenix, AZ, Dec. 1999.

[32] H.J. Sussmann. Set-Valued Differentials and the Hybrid Maximum Principle. *IEEE Conference on Decision and Control*, Vol. 1, pp. 558 -563, Dec. 2000.

[33] R. Vasudevan, H. Gonzalez, R. Bajcsy, and S.S. Sastry. Consistent Approximations for the Optimal Control of Constrained Switched Systems. Private communication, August 2012.

[34] L.Y. Wang, A. Beydoun, J. Cook, J. Sun, and I. Kolmanovsky. Optimal Hybrid Control with Applications to Automotive Powertrain Systems. In *Control Using Logic-Based Switching*, Vol. 222 of LNCIS, pp. 190-200, Springer-Verlag, 1997.

[35] Y. Wardi and M. Egerstedt. Algorithm for Optimal Mode Scheduling in Switched Systems. Technical Memorandum, www.ece.gatech.edu/∼magnus/OptSwitchTechnReport.pdf, Georgia Institute of Technology, March 2011.

[36] Y. Wardi and M. Egerstedt. Algorithm for Optimal Mode Scheduling in Switched Systems. *Proc. ACC*, Montreal, Canada, June 27-29, 2012.

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

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

[39] F. Zhu and P.J. Antsaklis. Optimal Control of Switched Hybrid Systems: A Brief Survey. Technical Report of the ISIS Group at the University of Notre Dame, ISIS-2011-003, July 2011.