Algorithm for Optimal Mode Scheduling in Switched Systems

Y. Wardi and M. Egerstedt

***Abstract*— This paper considers the problem of computing the schedule of modes in an autonomous switched dynamical system, that minimizes a cost functional defined on the trajectory of the system's continuous state variable. It proposes an algorithm that modifies a finite but unbounded number of modes at each iteration, whose computational workload at the various iterations appears to be independent on the number of modes being changed. The algorithm is based on descent directions defined by Gâteaux differentials of the performance function with respect to variations in mode-sequences, and its convergence to (local) minima is established in the framework of optimality functions and minimizing sequences, devised by Polak for infinite-dimensional optimization problems.**

I. Introduction

Autonomous switched-mode hybrid dynamical systems often are characterized by the following equation,

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

where $x \in R^n$ is the state variable, $v \in V$ with V being a given finite set, and $f : R^n \times V \to R^n$ is a suitable function. Suppose that the system evolves on a horizon-interval $[0, T]$ for some $T > 0$, and that the initial state $x(0) = x_0$ is given for some $x_0 \in R^n$. The input control of this system, $v(t)$, is discrete since V is a finite set, and we assume that the function $v(t)$ changes its values a finite number of times during the horizon interval $[0, T]$.

Such systems have been investigated in the past several years due to their relevance in various control applications; see, e.g., [3], [6] for surveys. Of a particular interest in such applications is an optimal control problem where it is desirable to minimize a cost functional (criterion) of the form

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

for a given $T > 0$, where $L : R^n \to R$ is a cost function defined on the state trajectory.

This general nonlinear optimal-control problem was formulated in [4], where the particular values of $v \in V$ are associated with the various modes of the system.[1] Several variants of the maximum principle were derived for this problem in [14], [8], [11], and subsequently provably-convergent optimization algorithms were developed in [16], [11], [13], [6], [1]. We point out that two kinds of problems were considered: those where the sequence of modes is fixed and the controlled variable consists of the switching times between them, and those where the controlled variable is comprised of the sequence of modes as well as the switching times between them. We call the former problem the *timing optimization problem*, and the latter, the *scheduling optimization problem*.

The timing optimization problem essentially is a nonlinear programming problem with a continuous variable, the switching times. In contrast, the variable of the scheduling optimization problem has discrete and continuous elements, namely the mode-sequence and the switching times between them, and hence generally it is more difficult than the timing optimization problem. Thus, while the algorithms that were proposed early focused on the timing optimization problem, several different (and apparently complementary) approaches to the scheduling-optimization problem have emerged as well. Zoning algorithms that compute (iteratively) the mode sequences based on geometric properties of the problem have been developed in [12], needle-variations techniques were presented in [2], [7], and relaxation methods were proposed in [3], [5]. These different approaches are still being developed and conclusive comparisons between them have to wait for extensive simulation experiments on realistic applications.

Our stating point is the algorithm we developed in [2] which alternates between the following two stages: (1). Given a sequence of modes, compute the switching times between them that minimize the functional J. (2). Update the mode-sequence by inserting to it a single mode at a (computed) time that would lead to the greatest-possible reduction rate in J. Then repeat Step 1, etc.

The second stage deserves some explanation. Fix a time $t \in [0, T]$, and let us denote the system's mode at that time by M_α. Now suppose that we replace this mode by another mode, denoted by M_β, over the time-interval $[t, t + \lambda]$ for some given $\lambda > 0$, and denote by $\tilde{J}(\lambda)$ the cost functional J defined by (2) as a function of λ. We call the one-sided derivative $\frac{d\tilde{J}}{d\lambda^+}(0)$ the *insertion gradient*, and we note that if $\frac{d\tilde{J}}{d\lambda^+}(0) < 0$ then inserting M_β for a brief amount of time at time t would result in a decrease in J, while if $\frac{d\tilde{J}}{d\lambda^+}(0) > 0$ then such an insertion would result in an increase in J. Now the second stage of the algorithm computes the time $t \in [0, T]$ and mode M_β that minimize the insertion gradient, and it performs the insertion accordingly. We mention that if the insertion gradient is non-negative for every mode M_β and time $t \in [0, T]$ then the schedule in question satisfies a necessary optimality condition and no insertion is performed.

This algorithm and its setting have been extended in [7] in the following three ways: (i) the system includes a

[1]The setting in [4] is more general since it involves a continuous-time control $u \in R^k$ as well as a discrete control v. Furthermore, most of the references cited in this paper except for ours' treat this more general formulation, but we focus only on the discrete control since it captures the salient features of switched-mode hybrid dynamical systems.

continuous control u in addition to the discrete control v, (ii) the optimal control problem has inequality constraints as well as multiple objectives, and (iii) the cost functional includes penalty terms on the number of switchings. However, on the issue of optimal mode-switching the two algorithms share the aforementioned two-stage approach. In fact, for the same choice of descent direction in the setting of [2] the two algorithms are identical.

The above two-stage approach may have an inherent inefficiency for the following reason. The requirement of solving a timing optimization problem following each mode-insertion involves, theoretically, an infinite-loop procedure at each step of the algorithm. Furthermore, the approach consists of inserting a single mode at each iteration, and the inefficiencies can become especially pronounced when each mode in a given schedule is active for only a brief amount of time. In contrast, this paper develops an alternative algorithm that changes any finite (but possibly unbounded) number of modes at each iteration, and its computational workload appears to be independent of the number of modes that are being changed. Moreover, it computes directly in the space of mode-schedules and does not solve any timing optimization problems. From a theoretical standpoint, our convergence-proof requires a new line of analysis since the arguments of [2], [7] break down. From a practical standpoint, it is premature to draw sweeping comparisons between these two algorithmic approaches before extensive testing on realistic problems; the objective of this paper is but to introduce a new player to the field and test it on an example.

The rest of the paper is organized as follows. Section II sets the mathematical formulation of the problem and recounts some established results. Section III presents the algorithm and its convergence properties, Section IV provides a simulation example, and Section V concludes the paper. Due to space constraints we relegate all of the proofs to a technical memorandum that is posted on the web site of the second author [15].

II. Problem Formulation and Survey of Relevant Results

Consider the state equation (1) and recall that the initial state x_0 and the final time $T > 0$ are given. We make the following assumption regarding the vector field $f(x, v)$ and the state trajectory $\{x(t)\}$.

Assumption 1: (i). For every $v \in V$, the function $f(x, v)$ is twice-continuously differentiable (C^2) throughout R^n. (ii). The state trajectory $x(t)$ is continuous at all $t \in [0, T]$.

Every mode-schedule is associated with an input control function $v : [0, T] \to V$, and we define an admissible mode schedule to be a schedule whose associated control function $v(\cdot)$ changes its values a finite number of times throughout the interval $t \in [0, T]$. We denote the space of admissible schedules by Σ, and a typical admissible schedule by $\sigma \in \Sigma$. Given $\sigma \in \Sigma$, we define the length of σ as the number of different consecutive values of v on the horizon interval $[0, T]$, and denote it by $\ell(\sigma)$. Furthermore, we denote the ith successive value of v in σ by v^i, $i = 1, \ldots, \ell(\sigma)$, and the switching time between v^i and v^{i+1} will be denoted by τ_i. Further defining $\tau_0 := 0$ and $\tau_{\ell(\sigma)} = T$, we observe that the input control function is defined by $v(t) = v^i \; \forall i \in [\tau_{i-1}, \tau_i)$, $i = 1, \ldots, \ell(\sigma)$. We require that $\ell(\sigma) < \infty$ but impose no upper bound on $\ell(\sigma)$.

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

$$\dot{p} \;=\; -\Big(\frac{\partial f}{\partial x}(x, v)\Big)^T p - \Big(\frac{dL}{dx}(x)\Big)^T \tag{3}$$

with the boundary condition $p(T) = 0$. Fix time $s \in [0, T)$, $w \in V$, and $\lambda > 0$, and consider replacing the value of $v(t)$ by w for every $t \in [s, s + \lambda)$. This amounts to changing the mode-sequence σ by inserting the mode associated with w throughout the interval $[s, s + \lambda)$. Denoting by $\tilde{J}(\lambda)$ the value of the cost functional resulting from this insertion, the insertion gradient is defined by $\frac{d\tilde{J}}{d\lambda^+}(0)$. Of course this insertion gradient depends on the mode-schedule σ, the inserted mode associated with $w \in V$, and the insertion time s, and hence we denote it by $D_{\sigma,s,w}$. We have the following result (e.g., [6]):

$$D_{\sigma,s,w} \;=\; p(s)^T \big(f(x(s), w) - f(x(s), v(s))\big). \tag{4}$$

As mentioned earlier, if $D_{\sigma,s,w} < 0$ then inserting to σ the mode associated with w on a small interval starting at time s would reduce the cost functional. On the other hand, if $D_{\sigma,s,w} \geq 0$ for all $w \in V$ and $s \in [0, T]$ then we can think of σ as satisfying a local optimality condition. Formally, define $D_{\sigma,s} := \min\{D_{\sigma,s,w} : w \in V\}$, and define $D_\sigma := \inf\{D_{\sigma,s} : s \in [0, T]\}$. Observe that $D_{\sigma,s,v(s)} = 0$ since $v(s)$ is associated with the same mode at time s and hence σ is not modified, and consequently, by definition, $D_{\sigma,s} \leq 0$ and $D_\sigma \leq 0$ as well. The condition $D_\sigma = 0$ is a natural first-order necessary optimality condition, and the purpose of the algorithm described below is to compute a mode-schedule σ that satisfies it.

Our algorithm is a descent method based on the principle of the Armijo step size. Given a schedule $\sigma \in \Sigma$, it computes the next schedule, σ_{next}, by changing the modes associated with points $s \in [0, T]$ where $D_{\sigma,s} < 0$. The Lebesgue measure of this set where the mode-sequence is modified acts as the parameter for the Armijo procedure. We point out that the algorithm in [7] also uses the Armijo step size to compute the length of an interval where a new mode is to be inserted, but our algorithm is radically different for the following reasons. First, the insertion at each iteration can be of several modes, and second, multiple modes can be swapped. More specifically, the search for the set where the modes are changed in a given iteration is not restricted by a single mode at either the given schedule or the modified schedule. This allows us to take large step sizes, thereby avoiding the need to solve timing optimization problems. *The point where the analyses in [2], [7] breaks down is in the fact that the insertion gradient $D_{\sigma,s,w}$ generally is discontinuous in s at the mode-switching times in a given schedule σ.*

Now one of the basic requirements of algorithms in the general setting of nonlinear programming is that every accumulation point of a computed sequence of iteration points satisfies a certain optimality condition, like stationarity or the Kuhn-Tucker condition. However, in our case such a convergence property is meaningless since the schedule-space Σ is neither finite dimensional nor complete. Consequently convergence of our algorithm has to be characterized by other means, and to this end we use Polak's concept of minimizing sequences [9]. Accordingly, the quantity D_σ acts as an *optimality function* [10], namely the optimality condition in question is $D_\sigma = 0$, while $|D_\sigma|$ indicates an extent to which σ fails to satisfy that optimality condition. Convergence of an algorithm means that, if it computes a sequence of schedules $\{\sigma_k\}_{k=1}^{\infty}$ then,

$$\limsup_{k\to\infty} D_{\sigma_k} \;=\; 0; \tag{5}$$

in some cases the stronger condition $\lim_{k\to\infty} D_{\sigma_k} = 0$ applies. In either case, for every $\epsilon > 0$ the algorithm yields an admissible mode-schedule $\sigma \in \Sigma$ satisfying the inequality $D_\sigma > -\epsilon$. Our analysis (see [15]) will yield Equation (5) by proving a uniformly-linear convergence rate of the algorithm.[2]

Since the Armijo step-size technique will play a key role in our algorithm, we conclude this section with a recount of its main features. Consider the general setting of nonlinear programming where it is desirable to minimize a C^2 function $f : R^n \to R$, and suppose that the Hessian $\frac{d^2 f}{dx^2}(x)$ is bounded on R^n. Given $x \in R^n$, a steepest descent from x is any vector in the direction $-\nabla f(x)$; we normalize the gradient by defining $h(x) := \frac{\nabla f(x)}{||\nabla h(x)||}$, and call $-h(x)$ the steepest-descent direction. Let $\lambda(x) \geq 0$ denote the step size so that the next point computed by the algorithm, denoted by x_{next}, is defined as

$$x_{next} \;=\; x - \lambda(x)h(x). \tag{6}$$

The Armijo step size procedure defines $\lambda(x)$ by an approximate line minimization in the following way (see [10]): Given constants $\alpha \in (0,1)$ and $\beta \in (0,1)$, define the integer $j(x)$ by

$$\begin{aligned} j(x) \;:\; \min\Big\{ j = 0,1,\ldots, \;: \\ f(x-\beta^j \nabla f(x)) - f(x) \leq -\alpha\beta^j ||\nabla f(x)||^2 \Big\}, \end{aligned} \tag{7}$$

and define

$$\lambda(x) \;=\; \beta^{j(x)} ||\nabla f(x)||. \tag{8}$$

Now the *steepest descent algorithm with Armijo step size* computes a sequence of iteration points x_k, $k = 1, 2, \ldots$, by the formula $x_{k+1} = x_k - \lambda(x_k)h(x_k)$; $\lambda(x_k)$ is called the Armijo step size at x_k. The main convergence property of this algorithm [10] is that every accumulation point $\hat{x}$ of a computed sequence $\{x_k\}_{k=1}^{\infty}$ satisfies the stationarity condition $\nabla f(\hat{x}) = 0$. Several results concerning convergence rate have been derived as well, and the one of interest to us is given by Proposition 1 below, whose proof can be found in [10], Equation (8b).

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

$$f(x - \lambda h(x)) - f(x) \;\leq\; -\alpha\lambda ||\nabla f(x)||. \tag{9}$$

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

$$\lambda(x) \;\geq\; \frac{2}{L}\beta(1-\alpha)||\nabla f(x)||. \tag{10}$$

■

This implies the following convergence result, proved in [10]:

Corollary 1: (1). There exists $c > 0$ such that $\forall x \in R^n$,

$$f(x_{next}) - f(x) \;\leq\; -c||\nabla f(x)||^2. \tag{11}$$

(2). If the algorithm computes a bounded sequence $\{x_k\}_{k=1}^{\infty}$ then

$$\lim_{k\to\infty} \nabla f(x_k) \;=\; 0. \tag{12}$$

■

III. Algorithm for the Mode-Scheduling Problem

To simplify the notation and discussion we assume first that the set V consists only of two elements, namely the system is bi-modal. This assumption incurs no significant loss of generality, and at the end of this section we will point out an extension to the general case where V consists of an arbitrary finite number of points. Let us denote the two elements of V by v_1 and v_2. A mode-schedule σ alternates between these two points, and we denote by $\{v^1, \ldots, v^{\ell(\sigma)}\}$ the sequence of values of v associated with the mode-sequence comprising σ. Denoting by v^c the complement of v, we have that $v^{i+1} = (v^i)^c$ for all $i = 1, \ldots, \ell(\theta) - 1$.

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 $S_{\sigma,0} \neq \emptyset$. Recall that $v(s)$ denotes the value of v at the time s. Then for every $s \in S_{\sigma,0}$ which is not a switching time, an insertion of the complementary mode $v(s)^c$ at s for a small-enough period would result in a decrease of J. Our goal is to flip the modes (namely, to switch them to their complementary ones) in a large subset of $S_{\sigma,0}$ that would result in a substantial decrease in J, where by the term "substantial decrease" we mean a decrease by at least aD_σ^2 for some constant $a > 0$. This "sufficient descent" in J is akin to the descent property of the Armijo step size as reflected in Equation (11).

This sufficient-descent property cannot be guaranteed by flipping the mode at every time $s \in S_{\sigma,0}$. Instead, we search

[2]The reason for the "limsup" in (5) instead of the stronger form of convergence (with "lim" instead of "limsup") is due to technical peculiarities of the optimality function D_σ that will be discussed later. We argue in [15] that the stronger form of convergence applies except for pathological situations. Furthermore, we will define an alternative optimality function and prove (in [15]) the stronger form of convergence for it. The choice of the most-suitable optimality function is largely theoretical and will not be addressed in this paper.

for a subset of $S_{\sigma,0}$ where, flipping the mode at every s in that subset would guarantee a 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

$$S_{\sigma,\eta} \;=\; \big\{s \in [0,T] \;:\; D_{\sigma,s} \leq \eta D_\sigma\big\}. \tag{13}$$

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 flipping the mode at every point $s \in S$, and denote by $\sigma(S)$ the resulting mode-schedule. In the forthcoming we will search for a set $S \subset S_{\sigma,\eta}$ that will give us the desired sufficient descent.

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

We next use such a mapping $S(\lambda)$ 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 $\eta \in (0,1)$. 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

$$j(\sigma) := \min\Big\{j = 0,1,\ldots, \;:\; J(\sigma(\lambda_j)) - J(\sigma) \leq \alpha\lambda_j D_\sigma\Big\}. \tag{14}$$

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

Observe that the Armijo step-size procedure is applied here not to the steepest descent (which is not defined in our problem setting) but to a descent direction defined by a Gâteaux derivative of J with respect to a subset of the interval $[0,T]$ where the modes are to be flipped. Generally this Gâteaux derivative is not necessarily continuous in λ and hence the standard arguments for sufficient descent do not apply.[3] However, the problem has a special structure guaranteeing sufficient descent and the algorithm's convergence in the sense of minimizing sequences. Furthermore, the sufficient descent property depends on $\mu(S_{\sigma,\eta})$ but is independent of both the string size $\ell(\sigma)$ and the particular choice of the mapping $S : [0, \mu(S_{\sigma,\eta})] \to 2^{S_{\sigma,\eta}}$. This guarantees that the convergence rate of the algorithm is not reduced when the string lengths of the schedules computed in successive iterations grow unboundedly.

[3] In [7] this Gâteaux derivative is continuous in λ, because λ is restricted to the extent of switching a single mode. As mentioned earlier, this restriction requires a run of the timing-optimization algorithm which is not needed here.

We next present the algorithm formally. 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 $S : [0, \mu(S_{\sigma,\eta})] \to 2^{S_{\sigma,\eta}}$ 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 (13), namely $S_{\sigma_k,\eta} = \{s \in [0,T] \;:\; D_{\sigma_k,s} \leq \eta D_{\sigma_k}\}$.

Step 3: Compute $j(\sigma_k)$ as defined by (14), namely

$$j(\sigma_k) \;=\; \min\Big\{j = 0,1,\ldots, \;:\; J(\sigma_k(\lambda_j)) - J(\sigma_k) \leq \alpha\lambda_j D_{\sigma_k}\Big\} \tag{15}$$

with $\lambda_j := \beta^j \mu(S_{\sigma_k,\eta})$, and set $\lambda(\sigma_k) := \lambda_{j(\sigma_k)}$.

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

It must be mentioned that the computation of the set $S_{\sigma_k,\eta}$ at Step 2 typically requires an adequate approximation. This paper analyzes the algorithm under the assumption of an exact computation of $S_{\sigma_k,\eta}$, while the case involving adaptive precision will be treated in a later, more comprehensive publication.

The main result of the paper concerns the sufficient descent of Algorithm 1, which yields its asymptotic convergence to schedules satisfying the optimality condition. We outline the main arguments of the analysis, while the proofs can be found in [15].

Given $\sigma \in \Sigma$, consider an interval $I := [s_1, s_2] \subset [0,T]$ of a positive length, such that the modes associated with all $s \in I$ are the same, i.e., $v(s) = v(s_1)$ $\forall s \in I$. Denote by $\sigma_{s_1}(\gamma)$ the mode-sequence obtained from σ by flipping the modes at every time $s \in [s_1, s_1 + \gamma]$, and consider the resulting cost function $J(\sigma_{s_1}(\gamma))$ as a function of $\gamma \in [0, s_2 - s_1]$. The following two preliminary results follow from the perturbation theory of differential equations; see [10].

Lemma 1: There exists a constant $K > 0$ such that, for every $\sigma \in \Sigma$, and for every interval $I = [s_1, s_2]$ as above, the function $J(\sigma_{s_1}(\cdot))$ is twice-continuously differentiable (C^2) on the interval $\gamma \in [0, s_2 - s_1]$; and for every $\gamma \in [0, s_2 - s_1]$, $|J(\sigma_{s_1}(\gamma))''| \leq K$ ("prime" indicates derivative with respect to γ). ■

Lemma 1 in conjunction with Corollary 1 (above) can yield sufficient descent only in a local sense, as in [7], where the same mode is scheduled according to σ. At mode-switching times, $D_{\sigma,s}$ is no longer continuous in s, and hence Lemma 1 cannot be extended to intervals where $v(\cdot)$ does not have a constant value. The following result provides an upper bound on the sensitivity across different modes.

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 (i.e., such that σ has the same mode throughout I), for every $\gamma \in [0, s_2 - s_1)$, and for every $s \geq s_2$,

$$|D_{\sigma_{s_1}(\gamma),s} - D_{\sigma,s}| \;\leq\; K\gamma. \tag{16}$$

■

Lemma 2 concerns a schedule $\sigma_{s_1}(\gamma)$ obtained from σ by flipping the modes at every point in the interval $[s_1, s_1 + \gamma)$; and it establishes a uniform Lipschitz continuity of the insertion gradient at future points s, with respect to the length of the interval where the modes are being flipped, γ. This extends the local sufficient descent, implied by Lemma 1, to intervals having any number of modes, and yields global sufficient descent in the sense of the following result.

Proposition 2: Fix $\eta \in (0,1)$, $\beta \in (0,1)$, and $\alpha \in (0,\eta)$. There exists a constant $c > 0$ such that, for every $\sigma \in \Sigma$ satisfying $D_\sigma < 0$, and for every $\lambda \in [0, \mu(S_{\sigma,\eta})]$ such that $\lambda \le c|D_\sigma|$,

$$J(\sigma(\lambda)) - J(\sigma) \;\le\; \alpha\lambda D_\sigma. \tag{17}$$

■

General results concerning sufficient descent, analogous to Proposition 2, provide key arguments in proving asymptotic convergence of nonlinear-programming algorithms (see, e.g., [10]). In our case, the optimality function has the peculiar property that it is discontinuous in the Lebesgue measure of the set where a mode is flipped. To see this, recall that $D_{\sigma,s,v(s)^c} = p(s)^T\big(f(x(s), v(s)^c) - f(x(s), v(s))\big)$ (see Equation (4)), and hence a change of the mode at time s would flip the sign of $D_{\sigma,s,v(s)^c}$. This can result in situations where $|D_\sigma|$ is "large" while $S_{\sigma,\eta}$ is "small", and for this reason, convergence of Algorithm 1 is characterize by Equation (5) with the $limsup$ rather than with the stronger assertion with lim. This is the subject of the following result.

Corollary 2: Suppose that Algorithm 1 computes a sequence of schedules, $\{\sigma_k\}_{k=1}^{\infty}$. Then Equation (5) is in force, namely $\limsup_{k\to\infty} D_{\sigma_k} = 0$. ■

Alternative optimality functions can be considered as well, like the term $D_\sigma\mu(S_{\sigma,\eta})$, where it is shown in [15] that $\lim_{k\to\infty} D_{\sigma_k}\mu(S_{\sigma_k,\eta}) = 0$. The choice of the "most appropriate" optimality function is an interesting theoretical question that will be addressed elsewhere, while here we consider the simplest and (in our opinion) most intuitive optimality function D_σ, despite its technical peculiarities.

Finally, a word must be said about the general case where the set V consists of more than two points. The algorithm and much of its analysis remain unchanged, except that for a given $\sigma \in \Sigma$, at a time s, the mode associated with $v(s)$ should be switched to the mode associated with the point $w \in V$ that minimizes the term $D_{\sigma,s,w}$.

IV. Numerical Example

We tested the algorithm on the double-tank system shown in Figure 1. 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, as shown in the figure. According to Toricelli's law, the state equation is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} v - \sqrt{x_1} \\ \sqrt{x_1} - \sqrt{x_2} \end{pmatrix}, \tag{18}$$

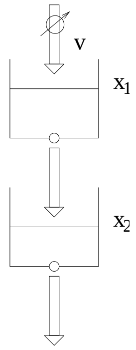

Fig. 1. Two-tank system

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 $1 \le x_i \le 4$, and if $v = 1$ indefinitely than $\lim_{t\to\infty} x_i = 1$, 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

$$J \;=\; 2\int_0^T \big(x_2 - 3\big)^2 dt, \tag{19}$$

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 2-5. Figure 2 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. This is evident from Figure 2, where the sliding mode has begun to be constructed. Figure 3 shows the resulting state trajectories $x_1(t)$ and $x_2(t)$, $t \in [0, T]$, associated with the last-computed schedule σ_{100}. The jagged curve is of x_1 while the smoother curve is of x_2. It is evident that x_2 climbs towards 3 initially and tends to stay there thereafter. Figure 4 shows the graph of the cost criterion $J(\sigma_k)$ as a function of the iteration count $k = 1, \ldots, 100$. The initial schedule, σ_1, is far away from the minimum and its associated cost is $J(\sigma_1) = 70.90$, and the cost of the last-computed schedule is $J(\sigma_{100}) = 4.87$. Note that $J(\sigma_k)$ goes down to under 8 after 3 iterations. Figure 5 shows the optimality function D_{σ_k} as a function of the iteration count k. Initially $D_{\sigma_1} = -14.92$ while at the last-computed schedule $D_{\sigma_{100}} = -0.23$, and it is seen that D_{σ_k} makes significant climbs towards 0 in few iterations. We also ran the algorithm for 200 iterations from the same initial schedule σ_1, in order to verify that $J(\sigma_k)$ and D_{σ_k} stabilize. Indeed they do, and J declined from $J(\sigma_{100}) = 4.87$ to $J(\sigma_{200}) = 4.78$, while the optimality

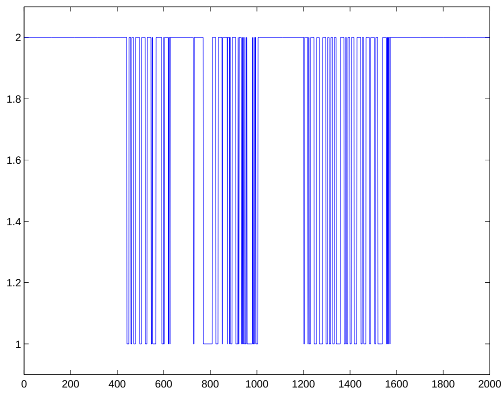

Fig. 2. Control (schedule) obtained after 100 iterations

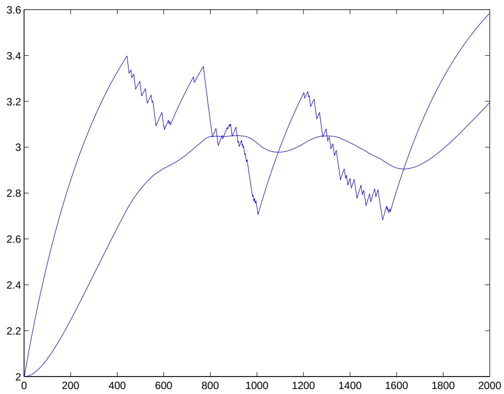

Fig. 3. x_1 and x_2 vs. t

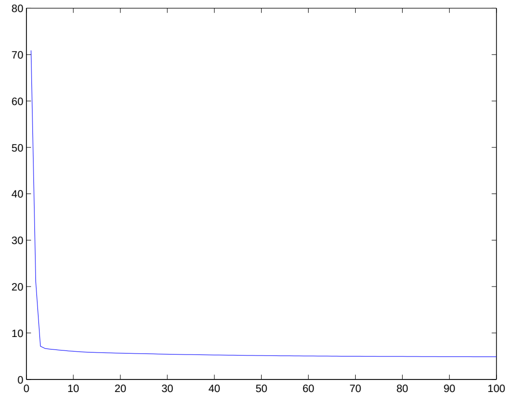

Fig. 4. Cost criterion vs. iteration count

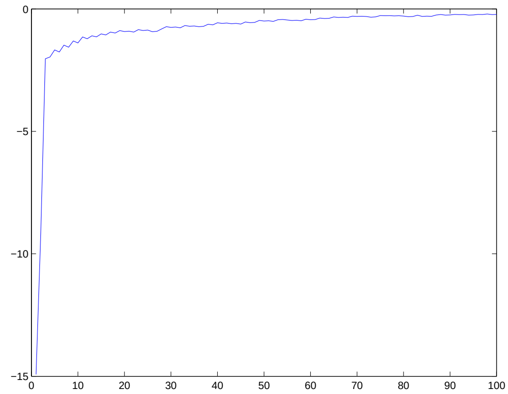

Fig. 5. Optimality function vs. iteration count

functions continues to rise towards 0, from $D_{\sigma_{100}} = -0.23$ to $D_{\sigma_{200}} = -0.062$.

V. Conclusions

This paper proposes an algorithm for the optimal mode-scheduling problem, where it is desirable to minimize an integral-cost criterion defined on the system's state trajectory as a function of the modes' schedule. Unlike extant techniques in the same vein, the algorithm here changes an unspecified number of modes per iteration, and it does not have to solve timing optimization problems. Asymptotic convergence is proved in the sense of minimizing sequences, and simulation results 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. *Proc. 16th IFAC World Congress*, Prague, the Czech Republic, July 3-8, 2005.

[2] 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.

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

[4] 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.

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

[6] 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.

[7] 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, December 15-17, 2010, pp. 7519-7526.

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

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

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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