A Controlled-Precision Algorithm for Mode-Switching Optimization

Y. Wardi, M. Egerstedt, and P. Twu$^+$

***Abstract*— This paper describes an adaptive-precision algorithm for solving a general optimal mode-scheduling problem in switched-mode dynamical systems. The problem is complicated by the fact that the controlled variable has discrete and continuous components, namely the sequence of modes and the switching times between them. Recently we developed a gradient-descent algorithm whose salient feature is that its descent at a given iteration is independent of the length (number of modes) of the schedule, hence it is suitable to situations where the schedule-lengths at successive iterations grow unboundedly. The computation of the descent direction requires grid-based approximations to solve differential equations as well as minimize certain functions on uncountable sets. However, the algorithm's convergence analysis assumes exact computations, and it breaks down when approximations are used, because the descent directions are discontinuous in the problem parameters. The purpose of the present paper is to overcome this theoretical gap and its computational implications by developing an implementable, adaptive-precision algorithm that controls the approximation levels by balancing precision with computational workloads. Its asymptotic convergence is proved, and simulation results are provided to support the theoretical developments.**

I. Introduction

Switched-mode autonomous dynamical systems can be defined by the following equation,

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

where $x \in R^n$ is the state variable, $v \in V$ is the input (or control), and the input-set V is finite. Each point $v \in V$ can be considered as a system's mode, whose associated dynamic response function is $f(\cdot,v)$. Suppose that the system evolves in a predetermined time-interval $[0,t_f]$ from a given initial state $x_0 := x(0) \in R^n$. Let $L : R^n \to R$ be a cost function defined on the state space, and define the cost-performance function J as

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

We define a *feasible control* as a function $v(t) : [0,t_f] \to V$ that is left-continuous and changes its values a finite number of times, and we denote by $\mathcal{V}$ the space of feasible controls.[1] Note that $\mathcal{V}$ is a linear space but it is not complete in any one of the L^p norms, $p \geq 1$. It will be convenient to view a control $v \in \mathcal{V}$ as a schedule of modes. We denote such a schedule by $\sigma = \big(\{v^i\}_{i=1}^{N+1};\{\tau_i\}_{i=1}^{N}\big)$, where $v^1,\ldots v^{N+1}$ are the successive values v assumes throughout the interval $[0,t_f]$, and $\tau_1,\ldots,\tau_N$ are the switching times between them. Thus, with the additional notation $\tau_0 := 0$ and $\tau_{N+1} := t_f$, we have that $v(t) = v^i \;\forall t \in [\tau_{i-1},\tau_i],\; i = 1,\ldots,N+1$. The schedule's length, $N+1$, may assume any finite value. For such a schedule σ we define v_σ to be the associated control function, and we denote by Σ the space of schedules σ such that $v_\sigma \in \mathcal{V}$. Obviously there is a one-to-one correspondence between $\sigma \in \Sigma$ and $v_\sigma \in \mathcal{V}$, but it will be convenient to consider both representations.

We make the following assumption.

Assumption 1: For every $v \in V$ the function $f(x,v)$ is twice-continuously differentiable (C^2) in x, and the function $L(x)$ is C^2.

We also assume, implicitly, that the state trajectory $\{x(t)\}$ associated with $v \in \mathcal{V}$ is continuous throughout $[0,t_f]$, and by Assumption 1, it is differentiable at all but the switching points τ_i. We consider the optimal-scheduling (or optimal-control) problem of minimizing J over all $\sigma \in \Sigma$ (or, $v \in \mathcal{V}$).

Such systems and optimization problems arise in various application areas such as mobile robotics, switching-circuit control, telecommunications, etc.; see, e.g., [15] for a survey. The optimal control problem has been formulated in the context of hybrid systems in [3], and variants of the maximum principle have been derived in [8], [11], [13], [14]. Subsequently, computational techniques have been developed in [1], [2], [4], [5], [7], [11], [12], [13], [16], [17].

The optimal mode-scheduling problem has been classified in the literature as either a *timing optimization problem* or a *sequencing optimization problem*, as follows. In timing optimization problems the mode-sequence $v^1,\ldots,v^{N+1}$ is given and the objective is to minimize J as a function of the switching times $\tau_1,\ldots,\tau_N$, whereas in sequencing optimization problems the control variable consists of the entire schedule, namely the mode-sequence as well as the switching times. Naturally the timing optimization problem is easier since it is inherently a nonlinear programming problem in the switching times, while the sequencing optimization problem has a discrete, sequencing variable in addition to the timing variable. Consequently, the early algorithmic developments considered the timing optimization problem [5], [11], [16], [17], but lately the focus has shifted to the sequencing optimization problem. A number of approaches have emerged, including geometric techniques for computing optimal mode-sequences [12], [13], algorithms based on relaxation techniques [2], [4], and gradient-descent methods [1], [7]. The algorithm considered in this paper falls into the latter category.

The algorithms published in [1], [7] alternate between the following two stages: 1). Given a sequence of modes, solve the timing optimization problem associated with them.

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

[1] The term 'v' is used to designate both a point in V and an input-control function; no confusion will arise from the context.

2). Insert into the schedule a single additional mode for a brief amount of time.[2] Reference [15] proposes an alternative approach which eliminates the need for solving the timing optimization problem. Instead, at each iteration it computes a subset of $[0,t_f]$ having a positive Lebesgue measure, where it modifies the mode-schedule. Furthermore, to achieve convergence of the algorithm, we use the Armijo step size on the Lebesgue measure of that set, thereby preserving the sufficient-descent property of gradient-descent algorithms that is associated with their convergence rate [10]. For these reasons, we argued in [15] that this algorithm may perform better than the one in [1], [7].

The convergence analysis and proofs of the former algorithm, developed in [15], are greatly complicated by the fact that the Gâteaux differential of J, associated with the needle variations via mode insertions, is discontinuous in the schedule σ, in a suitable sense defined below. Likewise, the sets where modes are changed have no continuity either. Consequently, the convergence proofs in [15] were obtained under the assumption that the various functions, gradients, and aforementioned sets are computed exactly. However, in an implementable algorithm they would have to be estimated via suitable approximations. What comes to mind is the framework of *consistent approximations*, developed by E. Polak [10], and based on adaptive balancing of precision with computational workloads. Accordingly, an algorithm starts with a crude precision, and increases it whenever it senses that an optimal point is approached. Our problem setting does not quite fit within the framework of consistent approximations due to the lack of continuity mentioned above, and hence its convergence-proof requires a new line of analysis.

An adaptive-precision algorithm was proposed in [15] without a proof of convergence, and the objective of this paper is to supply such a proof and present some simulation results. Section II formalizes the problem and provides background material, Section III carries out the analysis, Section IV presents the results of simulation experiments, and Section V concludes the paper.

II. Problem Formulation and Established Results

This section summarizes the results in [15] upon which the contributions of this paper are based.

Fix a schedule $\sigma \in \Sigma$, $s \in [0,t_f]$, and $w \in V$. For $\lambda \geq 0$, consider the change to the schedule σ obtained by swapping the mode at every time $t \in [s, s+\lambda)$, from that scheduled according to σ to the one associated with w. In other words, we replace $v_\sigma(t)$ by w for every $t \in [s, s+\lambda)$. Let us view the resulting value of J as a function of λ and s, hence denoted by $\tilde{J}_s(\lambda)$, and focus on its variable λ. We are concerned with the right-derivative $\frac{d\tilde{J}_s}{d\lambda^+}(0)$ which we denote by $D_{\sigma,s,w}$. This derivative term is the Gâteaux derivative of J at σ along the direction defined by inserting the mode w at an interval starting at s, and we call it the *insertion gradient*.

Note that if σ is a minimum point for J then it is impossible to have $D_{\sigma,s,w} < 0$ for any pair $(s,w) \in [0,t_f] \times V$, since this would indicate a direction of descent for J by inserting to the schedule σ the mode w in a small interval starting at s.[3] We can formulate the last statement as a necessary local-optimality condition on $\sigma \in \Sigma$ in the following way.

$$D_\sigma \ := \ \inf\Big\{ \min\big\{D_{\sigma,s,w} : w \in V\big\} \ : \ s \in [0,t_f]\Big\} \ \geq \ 0. \tag{3}$$

For every $s \in [0,t_f]$, define $D_{\sigma,s} := \min\{D_{\sigma,s,w} : w \in V\}$, and observe that $D_\sigma = \inf\{D_{\sigma,s} : s \in [0,t_f]\}$. We note that $D_{\sigma,s}$ cannot be positive since $D_{\sigma,s,v_\sigma(s)} = 0$ (inserting a mode onto itself would not change J), and therefore $D_{\sigma,s} \leq 0$ and hence $D_\sigma \leq 0$ as well; consequently, the necessary optimality condition is $D_\sigma = 0$.

The insertion gradient $D_{\sigma,s,w}$ can be derived by variational principles as follows; see, e.g., [5]. Define the costate variable $p(t) \in R^n$ by the differential equation

$$\dot{p} \ = \ -\Big(\frac{\partial f}{\partial x}(x, v_\sigma)\Big)^T p - \Big(\frac{dL}{dx}(x)\Big)^T \tag{4}$$

with the boundary condition $p(t_f) = 0$. Then, for every $s \in [0,t_f]$ and $w \in V$,

$$D_{\sigma,s,w} \ = \ p(s)^T\big(f(x(s),w) - f(x(s),v_\sigma(s))\big) \tag{5}$$

as shown in [5]. This equation indicates the lack of continuity of $D_{\sigma,s,w}$ with respect to "small" variations in the schedule σ. The last multiplicative term in (5), namely $f(x(s),w) - f(x(s),v_\sigma(s))$, implies that a change in a schedule σ at a single point s, by swapping $v_\sigma(s)$ by another mode, w, generally results in large changes in $D_{\sigma,s,w}$, and consequently in large changes in D_σ. This has some implications. It is tempting to regard the control functions $v \in \mathcal{V}$, namely v_σ for $\sigma \in \Sigma$, as elements in a suitable Banach space, and what comes to mind is L^p for some $p \in \{1,2,\infty\}$. However, the function $\sigma \rightarrow D_\sigma$ is not well-defined on any one of these spaces, since changing v_σ at a single point $s \in [0,t_f]$ results in the same element in L^p but in different values of D_σ. This problem is circumvented if the input control function $v(\cdot)$ is left continuous, and therefore we consider only feasible controls, namely $v \in \mathcal{V}$ having (by definition) constant values on left-closed, right-open intervals of positive lengths. Observe that the L^p norms are well-defined on $\mathcal{V}$ but this space is incomplete with respect to those norms. Also, the function D_σ is well-defined on $\mathcal{V}$ but is not continuous. All of this greatly complicates the convergence analysis of our optimization algorithms. However, it is natural to define the optimization algorithm on the space of feasible controls with the L^1 norm, and that is what we do.

We say that a schedule $\sigma \in \Sigma$ is *stationary* if $D_\sigma = 0$. Generally, the term D_σ serves as an optimality function, namely a gauge of the extent to which σ fails to be stationary. Reference [10] uses this notion to construct convergent

[2] The setting in [7] is more general than [1] since it includes constraints, multiple objectives, and a continuous-parameter control u in addition to the mode-schedule v.

[3] Please note the slight misnomer: we use the term "the mode w" rather than "the mode associated with w" when no confusion arises.

descent algorithms and to characterize their convergence rates. To explain the main idea, consider the abstract setting in which it is desirable to minimize a continuous function $J(\sigma)$ over a space Σ. Let $\theta(\sigma) \leq 0$ be an optimality function such that the set $\{\sigma \in \Sigma \,:\, \theta(\sigma) = 0\}$ is an optimality condition of interest, henceforth called stationarity.

The algorithmic framework in [10] is based on the following principle: Given $\sigma \in \Sigma$, compute the next point, σ_{next} such that $J(\sigma_{next}) - J(\sigma) \leq 0$, and furthermore, the amount of descent, $|J(\sigma_{next}) - J(\sigma)|$, is bounded from below by $K|\theta(\sigma)|$ for a given constant $K > 0$. We call this property *sufficient descent* as long as K is local to an open neighborhood of θ, and we call it *uniform sufficient descent* if K is global on Σ. The property of sufficient descent ensures the stationarity of every limit point of an iteration-sequence $\{\sigma_k\}_{k=1}^{\infty}$. However, if Σ is infinite-dimensional then bounded sequences may not have limit points, and consequently we seek an algorithm satisfying the limit $\lim_{k\to\infty} \theta(\sigma_k) = 0$ for every iteration-sequence it computes. This condition is ensured by the global-sufficient descent of an algorithm, but merely sufficient descent is not enough.

Generally, an algorithm is specified by procedures for computing its direction and step size. In the above framework, the direction typically is determined by gradients and directional derivatives so as to guarantee descent, while for the step size the Armijo procedure is often used [10]. For our optimal mode-scheduling problem, we define the descent direction by swapping the modes at sets of points s where $D_{\sigma,s} < 0$, and we use the Armijo step size on the Lebesgue measure of such sets. Furthermore, we search for such sets where $D_{\sigma,s}$ is "more negative" than at other points. To make all of this formal, fix $\eta \in (0,1)$, and consider $\sigma \in \Sigma$ such that $D_\sigma < 0$. Define the set $S_{\sigma,\eta}$ by

$$S_{\sigma,\eta} \;=\; \big\{s \in [0, t_f] \;:\; D_{\sigma,s} \leq \eta D_\sigma\big\}, \tag{6}$$

as illustrated in Figure 1. Let $\mu(\cdot)$ denote the Lebesgue measure on R. For every positive $\lambda \leq \mu(S_{\sigma,\eta})$, let $S(\lambda)$ be a subset of $S_{\sigma,\eta}$ such that $\mu(S(\lambda)) = \lambda$. We consider swapping the modes at every point $s \in S(\lambda)$; in order to ensure that the resulting schedule corresponds to a feasible control, we henceforth assume, implicitly, that such subsets consist of finite unions of left-closed and right-open intervals. For every $s \in S(\lambda)$, $D_{\sigma,s} \leq \eta D_\sigma$ (see Figure 1), and we replace the mode at s with the mode $w \in V$ such that $D_{\sigma,s,w} = D_{\sigma,s}$. Let us denote this w in a functional form as $w = w(\sigma, s)$; often it is uniquely defined, but if the set argmin$(D_{\sigma,s,w} \,:\, w \in V)$ is not a singleton, we pick $w(\sigma, s)$ in an arbitrary fashion subject to the condition that the mapping $s \to w(\sigma, s)$ is piecewise constant and left continuous. We now define $\sigma(\lambda)$ as the schedule obtained from σ by changing $v_\sigma(s)$ to $w(\sigma, s)$ for every $s \in S(\lambda)$. The remaining question is how an algorithm should choose λ in order to guarantee its convergence.

We choose the value of λ according to the Armijo step size defined as follows (see [10] for a general treatment of the Armijo step size). 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

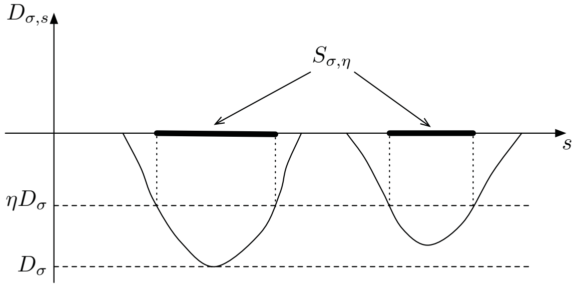

Fig. 1. Illustration of the set $S_{\sigma,\eta}$.

$$j(\sigma) \;:=\; \min\Big\{j = 0, 1, \ldots, \;:\; J(\sigma(\lambda_j)) - J(\sigma) \leq \alpha\lambda_j D_\sigma\Big\}. \tag{7}$$

Finally, define $\lambda(\sigma) := \lambda_{j(\sigma)}$. $\lambda(\sigma)$ is the Armijo step size, and starting from σ, the algorithm in [15] defines and computes the next iteration-point, σ_{next}, via $\sigma_{next} := \sigma(\lambda(\sigma))$.

The main result in [15] concerning sufficient descent of this algorithm is the following.

Fact 1 (Proposition 1 in [15]): Fix $\eta \in (0,1)$ and $\alpha \in (0, \eta)$. There exists a constant $c > 0$ such that, for every $\sigma \in \Sigma$ satisfying $D_\sigma < 0$, and for every $\lambda \in [0, \mu(S_{\sigma,\eta})]$ such that $\lambda \leq c|D_\sigma|$,

$$J(\sigma(\lambda)) - J(\sigma) \;\leq\; \alpha\lambda D_\sigma. \tag{8}$$

□

Equation (8) would provide the global sufficient-descent property if λ were bounded only by $c|D_\sigma|$, however, it is also bounded by $\mu(S_{\sigma,\eta})$. In fact, it is possible to have $|D_\sigma|$ "large" while $\mu(S_{\sigma,\eta})$ arbitrarily small, and this inhibits sufficient descent, as can be seen in the RHS of (8). These situations are not pathological and may arise when σ has a large number of modes, as happens when the iteration-sequence $\{\sigma_k\}$ approaches a solution consisting of a sliding mode. The absence of sufficient descent is due to the lack of continuity of the function D_σ in the L^1 norm on $\mathcal{V}$. Nonetheless the following result has been proved in [15].

Fact 2 (Proposition 2 in [15]): Consider an algorithm computing a sequence of schedules, $\{\sigma_k\}_{k=1}^{\infty}$, such that $\sigma_{k+1} = (\sigma_k)_{next}$ as defined above. Then the following limit holds,

$$\limsup_{k\to\infty} \theta(\sigma_k) \;=\; 0. \tag{9}$$

Furthermore, if $\sigma \in \Sigma$ is a limit point of the sequence $\{\sigma_k\}$, then $\theta(\sigma) = 0$. □

Equation (9) is the main result in Fact 2; since $\mathcal{V}$ is infinite-dimensional there are no guarantees that bounded sequences have limit points. Generally in optimization we seek a convergence result of the form $\lim_{k\to\infty} \theta(\sigma_k) = 0$, but this cannot be obtained due to the aforementioned lack of continuity of the optimality function. However, the fact that the algorithm is a descent method renders Equation (9) practically equivalent, as argued for in [15].

All of these results were derived under the assumption of exact computations of D_σ, $S_{\sigma,\eta}$, and $S(\lambda)$, and therefore the algorithm is said to be *conceptual*. Any implementation would require approximations, and hence is called *implementable* (this taxonomy was established in [10]). The approximations are due mainly to the need to solve the differential equations (1) and (4), compute the optimality function D_σ, and compute the sets $S_{\sigma,\eta}$ and $S(\lambda)$. Naturally, we would use a finite grid to do that; however, the lack of continuity, especially of the sets $S_{\sigma,s}$, impedes the sufficient-descent property of the conceptual algorithm and breaks down the proofs in [15]. The next section describes an implementable algorithm and proves its convergence in a way similar to Equations (8) and (9).

Before closing this section we mention the following extension of Fact 1. Recall that $w(\sigma,s)$ is any point $w \in V$ such that $D_{\sigma,s,w} = D_{\sigma,s}$. Fix $\rho \in (\frac{1}{2},1)$ (the restriction $\rho > \frac{1}{2}$ is needed for technical reasons, as will be seen later), and let $w_\rho(\sigma,s)$ be a point $w \in V$ such that $D_{\sigma,s,w} < \rho D_{\sigma,s}$. Such a choice of w may not be unique, but we assume a well-defined functional form of $w_\rho(\sigma,s)$ in terms of s, and that this function is piecewise constant and left continuous. Now for a given $\lambda \geq 0$, let $S_\rho(\lambda)$ be a set contained in $S_{\sigma,\eta}$ having a Lebesgue-measure λ that is comprised of a finite union of left-closed, right-open intervals, and define $\sigma_\rho(\lambda)$ to be the schedule obtained from σ by swapping $v_\sigma(s)$ with $w_\rho(\sigma,s)$ for every s in $S_\rho(\lambda)$.

Fact 3 (Proposition 3 in [15]): 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 $\rho \in (\frac{1}{2},1)$,

$$J(\sigma_\rho(\lambda)) - J(\sigma) \;\leq\; \alpha\lambda\rho D_\sigma. \tag{10}$$

□

III. Implementable Algorithm

This section presents the new results, derived in this paper, concerning an adaptive-precision algorithm.

Given $\sigma \in \Sigma$, denote by $x_\sigma(\cdot)$ and $p_\sigma(\cdot)$ the corresponding state and costate trajectories defined by Equations (1) and (4), respectively. Let $\mathcal{G} := \{t_1,\ldots,t_M\} \subset [0,t_f]$ denote a finite grid, and define the gap of $\mathcal{G}$ as $gap(\mathcal{G}) = \max\{t_\nu - t_{\nu-1} : \nu = 2,\ldots,M\}$. The numerical precision associated with a grid can be controlled, to some extent, by its gap, but more may be needed. Regarding D_σ, Equation (5) suggests that it is possible to have an arbitrarily-small $gap(\mathcal{G})$ and a large estimation error for $D_{\sigma,s,w}$ and hence for D_σ as well. This can happen when an interval $[\tau_{i-1},\tau_i)$, $i = 1,\ldots,N+1$ does not contain any point of $\mathcal{G}$.[4] We say that a grid $\mathcal{G}$ is *complete* with respect to $\sigma \in \Sigma$ if every mode in σ is represented in $\mathcal{G}$, namely, for every $i = 1,\ldots,N+1$, $[\tau_{i-1},\tau_i) \cap \mathcal{G} \neq \emptyset$.

Finite grids are required to approximate not only D_σ but also x_σ, p_σ, and $J(\sigma)$. Suppose that all numerical integrations are carried out via the Euler method; forward integration for x_σ and $J(\theta)$ in (1) and (2), and backward integration for p_σ in (4) (since the differential equation in (4) is backward, its numerical integration is performed forward in time). While cognizant of the fact that there are alternative, superior numerical integration techniques, we accept Polak's argument that in the context of an optimization algorithm Euler's method often is preferred, due to its relative simplicity, unless there are standing stability issues; see [10], pp. 534-535.

Now a small gap and completeness of a grid $\mathcal{G}$ with respect to σ may not suffice to guarantee a small error resulting from the numerical integration. What is needed, in addition, is that every switching time of σ be in the grid as well, thereby ensuring that the integration error is in the order of $gap(\mathcal{G})$ (see [6], the proof of Theorem 1.3). Without this assumption the integration error could be in the order of $(N+1)gap(\mathcal{G})$, and $N+1$, the number of modes, is not assumed to be bounded. We say that a grid $\mathcal{G}$ is *left complete* with respect to σ if every switching time of σ lies on the grid.

We assume a single grid for both numerical integration and estimation of D_σ, but generally two separate grids can be used. A left-complete grid controls the estimation errors of $J(\sigma)$ and D_σ, but not of the set $S_{\sigma,\eta}$ and other quantities that depend on it. Such uniform error bounds would be required to place the analysis in the framework of consistent approximations [10], and its absence compels us to develop a new set of arguments.

The following list of notational definitions of the various approximations we use is self-explanatory. Consider a sequence of progressively-finer grids on the interval $[0,t_f]$, denoted by $\mathcal{G}^n$, $n = 1,2,\ldots$, such that $\lim_{n\to\infty} gap(\mathcal{G}_n) \to 0$ monotonically.

- $x^n_\sigma(t)$ and $p^n_\sigma(t)$ - piecewise-constant, left-continuous functions resulting from integrations of the differential equations (1) and (4), respectively.
- $J^n(\sigma)$ - an approximation of $J(\sigma)$ resulting from a numerical integration of (2), based on $x^n_\sigma(t)$.
- $D^n_{\sigma,s,w}$ - an approximation of $D_{\sigma,s,w}$ based on (5), with $x^n_\sigma(t)$ and $p^n_\sigma(t)$; $D^n_{\sigma,s} := \min_{w\in V} D^n_{\sigma,s,w}$; and $D^n_\sigma := \min_{s\in\mathcal{G}^n} D^n_{\sigma,s}$.
- ***Given*** $s \in \mathcal{G}^n$. $\quad w^n(\sigma,s) :=$ a point $w \in V$ such that $D^n_{\sigma,s,w} = D^n_{\sigma,s}$. If such w is not unique, suppose that there exists a well-defined function yielding a particular choice of w.
- ***Given*** $\eta \in (0,1)$. $\quad S^{n,0}_{\sigma,\eta} := \{s \in \mathcal{G}^n : D^n_{\sigma,s} \leq \eta D^n_\sigma\}$. Define $S^n_{\sigma,\eta}$ to be the union, over all $s \in S^{n,0}_{\sigma,\eta}$, of the left-closed and right-open intervals whose left point is s and whose right point is the next point on the grid $\mathcal{G}^n$ following s.
- Given a mapping $S^n : [0,\mu(S^n_{\sigma,\eta})] \to 2^{S^n_{\sigma,\eta}}$ such that, $\forall\lambda \in [0,\mu(S^n_{\sigma,\eta})]$, (i) $S^n(\lambda)$ is the finite union of left-closed and right-open intervals whose left points are on $\mathcal{G}^n$, and (ii) $\mu(S^n(\lambda)) = \lambda$. $\sigma^n(\lambda)$ is the mode-schedule defined by the following two steps: (i) For every $s \in S^{n,0}_{\sigma,\eta} \cap S^n(\lambda)$, change $v_\sigma(s)$ to $w^n(\sigma,s)$. (ii) For every $t \in S^n(\lambda) \setminus \mathcal{G}^n$, with $s := \max\{\tau \leq t : \tau \in \mathcal{G}^n\}$,

[4] Recall that the switching times of σ are τ_i, $i = 1,\ldots,N$, and the mode in every interval $[\tau_{i-1},\tau_i)$ is v^i.

change $v_{\sigma}(t)$ to $w^{n}(\sigma,s)$.

- Given constants $\alpha\in(0,\eta)$ and $\beta\in(0,1)$. For the notation below, define $\lambda_{j}:=\beta^{j}\mu(S^{n}_{\sigma,\eta})$, $j=0,1,\ldots$. Define $j^{n}(\sigma):=\min\{j=0,1,\ldots,: J^{n}(\sigma^{n}(\lambda_{j}))-J^{n}(\sigma)\leq\alpha\lambda_{j}D^{n}_{\sigma}\}$, and define $\lambda^{n}(\sigma):=\lambda_{j^{n}(\sigma)}$.

The algorithm defined below will compute a sequence of mode-schedules, $\{\sigma_{n}\}$, and the grid $\mathcal{G}^{n}$ will depend on σ_{n} so as to be left complete with respect to it. The rules for controlling the grid's precision via $gap(\mathcal{G}^{n})$ will be specified later, after we establish some preliminary results concerning the approximations and sufficient descent.

Lemma 1: For every $\epsilon>0$, there exists $\delta>0$ such that, for every $\sigma\in\Sigma$, and for every finite grid $\mathcal{G}^{n}\subset[0,t_{f}]$ that is left-complete with respect to σ and such that $gap(\mathcal{G}^{n})<\delta$, the following inequalities hold: (i) $||x^{n}_{\sigma}-x_{\sigma}||_{L^{\infty}}<\epsilon$, and $||p^{n}_{\sigma}-p_{\sigma}||_{L^{\infty}}<\epsilon$. (ii). $|J^{n}(\sigma)-J(\sigma)|<\epsilon$. (iii) $\forall\ s\in[0,t_{f})$ and $\forall\ w\in V$, $|D^{n}_{\sigma,s,w}-D_{\sigma,s,w}|<\epsilon$, $|D^{n}_{\sigma,s}-D_{\sigma,s}|<\epsilon$, and $|D^{n}_{\sigma}-D_{\sigma}|<\epsilon$.

Proof: Fix $\delta>0$, and consider a grid $\mathcal{G}^{n}$ that is left-complete with respect to σ and such that $gap(\mathcal{G}^{n})<\delta$. Consider $s\in I_{\sigma,i}:[\tau_{i-1},\tau_{i})$ for some $i=1,\ldots,N+1$, and define $t_{j}:=\max\{t\in\mathcal{G}^{n}:t\leq s\}$ and $t_{j+1}:=\min\{t\in\mathcal{G}^{n}:t>s\}$. By left completeness, $t_{j}\in I_{\sigma,i}$; and since the integration method is forward Euler, it follows that

$$x^{n}_{\sigma}(t_{j+1})=x^{n}_{\sigma}(t_{j})+f(x^{n}_{\sigma}(t_{j}),v^{i})(t_{j+1}-t_{j}).\tag{11}$$

Consequently, standard arguments for deriving error bounds for the forward Euler method (see, e.g., [6], the proof of Theorem 1.3) imply part (i) of the lemma concerning x^{n}_{σ}. The statement concerning p^{n}_{σ} follows via a similar argument, and part (ii) follows immediately. Regarding part (iii), Equation (5) in conjunction with the assumption that $\mathcal{G}^{n}$ is left-complete with respect to σ imply the first stated bound, and the two other bounds follow from it immediately. ■

This lemma establishes continuity of various functions in terms of $gap(\mathcal{G}^{n})$. What is lacking is a similar continuity of the set $S^{n}_{\sigma,\eta}$ with respect to $S_{\sigma,\eta}$. Instead, we have the following result.

Lemma 2: For every $\eta_{1}>0$, $\eta>\eta_{1}$, $\eta_{2}>\eta$, and $\zeta>0$, there exist $\delta>0$ such that for all $\sigma\in\Sigma$ satisfying $D_{\sigma}<-\zeta$, and for every finite grid $\mathcal{G}^{n}\subset[0,t_{f}]$ which is left-complete with respect to σ and satisfying $gap(\mathcal{G}^{n})<\delta$,

$$S_{\sigma,\eta_{2}}\subset S^{n}_{\sigma,\eta}\subset S_{\sigma,\eta_{1}}.\tag{12}$$

Proof: Fix $\zeta>0$, and η_{1}, η, and η_{2} as specified in the statement of the lemma, and consider $\sigma\in\Sigma$ such that $D_{\sigma}<-\zeta$, and a grid $\mathcal{G}^{n}$ that is left-complete with respect to σ. We first prove the left inequality of (12).

Consider $s\in S_{\sigma,\eta_{2}}$. Let $\bar{s}:=\max\{t\in\mathcal{G}^{n}:t\leq s\}$, namely the point on the grid immediately to the left of s, possibly including s itself in the event that $s\in\mathcal{G}^{n}$. By definition $D_{\sigma,s}<\eta_{2}D_{\sigma}$. By lemma 1, aided by the assumption that $D_{\sigma}<-\zeta$, there exists $\delta>0$, independent of the specific σ and $\mathcal{G}^{n}$, such that, if $gap(\mathcal{G}^{n})<\delta$ then $D^{n}_{\sigma,\bar{s}}<\eta D^{n}_{\sigma}$. By definition, this implies that $s\in S^{n}_{\sigma,\eta}$, thereby establishing the left inequality of (12).

We next establish the right inequality of (12). Consider a point $s\in S^{n}_{\sigma,\eta}$. Let $\bar{s}:=\max\{t\in\mathcal{G}^{n}:t\leq s\}$. By definition of the set $S^{n}_{\sigma,\eta}$, $\bar{s}\in S^{n,0}_{\sigma,\eta}$, implying that $D^{n}_{\sigma,\bar{s}}\leq\eta D^{n}_{\sigma}$. Since $\mathcal{G}^{n}$ is left-complete with respect to σ, s and $\bar{s}$ are contained in the same interval $I_{\sigma,i}:=[\tau_{i-1},\tau_{i})$, $i=1,2,\ldots$; i.e., the same mode is active according to σ at both s and $\bar{s}$. Therefore, and by Lemma 1, there exists $\delta>0$, independent of the particular choice of σ or $\mathcal{G}^{n}$, such that, if $gap(\mathcal{G}^{n})<\delta$, then $D_{\sigma,s}\leq\eta_{1}D_{\sigma}$. This implies that $s\in S_{\sigma,\eta'}$, hence (12). ■

The following result provides the key argument for the proof of Proposition 1, below, by guaranteeing that certain mode-sequences qualify as $\sigma_{\rho}(\lambda)$ for the purpose of Proposition 3 in [15] (Fact 3, above).

Lemma 3: For every $\eta\in(0,1)$, $\rho\in(0,1)$, and $\zeta>0$, there exists $\delta>0$ such that, for every $\sigma\in\Sigma$ such that $D_{\sigma}<-\zeta$; for every grid $\mathcal{G}^{n}$ that is left-complete with respect to σ and satisfying $gap(\mathcal{G}^{n})<\delta$; for every $s\in S^{n,0}_{\sigma,\eta}$; for every $w\in V$ such that $D^{n}_{\sigma,s,w}=D^{n}_{\sigma,s}$; and for every $t\in S^{n}_{\sigma,\eta}$ such that $s=\max\{\tau\leq t:\tau\in\mathcal{G}^{n}\}$, we have that

$$D_{\sigma,t,w}\ <\ \rho D_{\sigma,t}.\tag{13}$$

Proof: Fix $\eta\in(0,1)$, $\rho\in(0,1)$, and $\zeta>0$. Fix $\rho_{1}\in(\rho,1)$. By Lemma 1(iii), there exists $\delta>0$ such that, for every $\sigma\in\Sigma$ such that $D_{\sigma}<-\zeta$, for every grid $\mathcal{G}^{n}$ that is left-complete with respect to σ and such that $gap(\mathcal{G}^{n})<\delta$, for every $s\in S^{n,0}_{\sigma,\eta}$, and for every $w\in V$ such that $D^{n}_{\sigma,s,w}=D^{n}_{\sigma,s}$, we have that

$$D_{\sigma,s,w}\ <\ \rho_{1}D_{\sigma,s}.\tag{14}$$

Next, consider $t\in[s,t_{f})$ such that $s=\max\{\tau\leq t:\tau\in\mathcal{G}^{n}\}$; then $v_{\sigma}(t)=v_{\sigma}(s)$, and hence, and by Equation (5) and the fact that $\rho<\rho_{1}$, and by Equation (14), we can reduce δ if necessary, by an amount that is independent of t, to ensure that Equation (13) is satisfied. ■

The following result provides us the property of uniform sufficient descent. It is an extension of Proposition 1 in [15] (Fact 1, above), but is considerably more complicated due to the fact that approximations are being used.

Proposition 1: For every $\eta>0$ and $\alpha\in(0,\eta)$ there exists $c>0$ with the following properties. For every $\zeta>0$ there exists $\delta>0$ such that, for all $\sigma\in\Sigma$ such that $D_{\sigma}<-\zeta$; for every finite grid $\mathcal{G}^{n}$ which is left-complete with respect to σ and satisfying $gap(\mathcal{G}^{n})<\delta$, and for every $\lambda\in\big[0,\min\{\mu(S^{n}_{\sigma,\eta}),c|D^{n}_{\sigma}|\}\big]$,

$$J(\sigma^{n}(\lambda))-J(\sigma)\ <\ \alpha\lambda D_{\sigma}.\tag{15}$$

Proof: Given $\eta\in(0,1)$ and $\alpha\in(0,\eta)$. Fix $\alpha_{1}\in(\alpha,\eta)$ and $\eta_{1}\in(\alpha_{1},\eta)$, so that $0<\alpha<\alpha_{1}<\eta_{1}<\eta<1$. Let c_{1} be the constant guaranteed by Fact 3 for α_{1} and η_{1}. Fix $c\in(0,c_{1})$. We next show that the statement of the proposition holds true with these η, α, and c.

Given $\zeta>0$. Fix $\rho\in(\frac{\alpha}{\alpha_{1}},1)$ such that $\rho>\frac{1}{2}$. By lemma 2 there exists $\delta>0$ such that, if $D_{\sigma}<-\zeta$, and if $\mathcal{G}^{n}$ is a left-complete (with respect to σ) grid satisfying $gap(\mathcal{G}^{n})<\delta$, then $S^{n}_{\sigma,\eta}\subset S_{\sigma,\eta_{1}}$. For such σ and $\mathcal{G}^{n}$, consider $\lambda\in[0,\min\{\mu(S^{n}_{\sigma,\eta}),c|D^{n}_{\sigma}|\})$. Since $S^{n}_{\sigma,\eta}\subset S_{\sigma,\eta_{1}}$, we have that $\mu(S^{n}_{\sigma,\eta})\leq\mu(S_{\sigma,\eta_{1}})$; moreover, by Lemma 1(iii) we can reduce δ if necessary to ensure that $c|D_\sigma^n| < c_1|D_\sigma|$; and as a result of all of this, $\lambda \le \min\{\mu(S_{\sigma,\eta_1}), c_1|D_\sigma|\}$. Next, by Lemma 3, δ can be further reduced if necessary to ensure that $\sigma^n(\lambda)$ qualifies as $\sigma_\rho(\lambda)$ for the purpose of Fact 3, and this implies that

$$J(\sigma^n(\lambda)) - J(\sigma) \;\le\; \alpha_1 \lambda \rho D_\sigma. \tag{16}$$

Since by assumption $\rho > \frac{\alpha}{\alpha_1}$, Equation (15) follows from (16). ∎

We next present our implementable algorithm, based on adaptive precision via grid selection. Given a sequence $\{\delta_n\}$ convergent to 0 monotonically, the grid's precision is controlled by δ_n via the requirement that $gap(\mathcal{G}^n) \le \delta_n$. There are three reasons to increase n: (i) D_σ^n is too small; (ii) $j^n(\sigma)$ is too large, and (iii) $\mu(S_{\sigma,\eta}^n)$ is too small. The first reason is clear because it indicates an approach to a stationary point. The second one stems from the fact that, for a given grid $\mathcal{G}^n$, it is possible for a given $\sigma \in \Sigma$ to have $D_\sigma < 0$ while $j^n(\sigma) = \infty$, and hence the grid has to be refined. The third condition requires a finer approximation in order to ensure that the grid captures every mode in σ.

Given monotone-decreasing sequences $\{\delta_n\}_{n=1}^\infty$, $\{\epsilon_{1,n}\}_{n=1}^\infty$, and $\{\epsilon_{2,n}\}_{n=1}^\infty$, all convergent to 0; and given a monotone-increasing sequence $\{M_n\}_{n=1}^\infty$, convergent to ∞.

Algorithm 1: Step 0: Start with an arbitrary schedule $\sigma_0 \in \Sigma$. Set $k = 0$ and set $n = 1$.
Step 1: Compute any finite grid $\mathcal{G}^n$ that is left-complete with respect to σ_k, such that $gap(\mathcal{G}^n) < \delta_n$.
Step 2: Compute $D_{\sigma_k}^n$. If $D_{\sigma_k}^n > -\epsilon_{1,n}$, set $n = n + 1$ and go to Step 1.
Step 3: Compute $S_{\sigma_k,\eta}^n$. If $\mu(S_{\sigma_k,\eta}^n) < \epsilon_{2,n}$, set $n = n + 1$ and go to Step 1.
Step 4: Determine whether $j^n(\sigma_k) > M_n$. If this is the case, set $n = n+1$ and go to Step 1. Otherwise, define $\lambda^n(\sigma_k) := \beta^{j^n(\sigma_k)}\mu(S_{\sigma_k,\eta}^n)$.
Step 5: Define $\sigma_{k+1} := \sigma^n(\lambda^n(\sigma_k))$. Set $k = k + 1$, and go to Step 2.

The following proposition contains the main result concerning convergence of this algorithm. Note that it extends Proposition 2 in [15] (Fact 2, above), but while the proof of the latter is straightforward, that of the present result is much more complicated by virtue of the fact that approximations are being used.

Proposition 2: (i). If Algorithm 1 jams (namely, gets "stuck") at a point $\sigma_k \in \Sigma$, then $D_{\sigma_k} = 0$. (ii). If the algorithm computes an infinite sequence $\{\sigma_k\}_{k=1}^\infty$, then

$$\limsup_{k\to\infty} D_{\sigma_k} \;=\; 0. \tag{17}$$

Proof: (i). Suppose that the algorithm jams at a point σ_k, and suppose, for the sake of contradiction, that $D_{\sigma_k} < 0$. At its final point σ_k, the algorithm returns to Step 1 infinitely often from either Steps 2, 3, or 4. Such a return from Step 2 implies that $\lim_{n\to\infty} D_{\sigma_k}^n = 0$, which is impossible by Lemma 1(iii). Next, fix $\eta_2 \in (\eta, 1)$. By Lemma 2, $S_{\sigma_k,\eta_2} \subset S_{\sigma_k,\eta}^n$ for n large enough, and hence an infinite return of the algorithm from Step 3 to Step 1 implies that $\mu(S_{\sigma_k,\eta_2}) = 0$. This, however, is impossible by the assumption that $D_{\sigma_k} < 0$ and the definition of Σ. Finally, an infinite return to Step 1 from Step 4 is impossible by Proposition 1 in conjunction with Lemma 1. This proves part (i) of the proposition.

(ii). Suppose that the algorithm computes a sequence $\{\sigma_k\}_{k=1}^\infty$, and suppose, for the sake of contradiction, that Equation (17) fails to hold. Then without loss of generality we can assume that there exists $\zeta > 0$ such that, for every $k = 1, 2, \ldots$, $D_{\sigma_k} < -\zeta$.

Let us denote by $n(k)$ the value of n with which the algorithm enters Step 5 with σ_k. We next show that $\lim_{k\to\infty} n(k) = \infty$. Suppose, for the sake of contradiction, that this is not true, so we can assume that $n(k) = n$ for a fixed $n \ge 1$ and all $k = 1, 2, \ldots$. By Step 5 and the definition of $\sigma^n(\lambda)$, it follows that $J^n(\sigma_{k+1}) - J^n(\sigma_k) \le \alpha\lambda^n(\sigma_k)D_{\sigma_k}^n$. Since $J^n(\sigma)$ is bounded from below over $\sigma \in \Sigma$, it implies that either $\limsup_{k\to\infty} D_{\sigma_k}^n = 0$, or $\liminf_{k\to\infty} \lambda^n(\sigma_k) = 0$ and hence (by the definition of $\lambda^n(\sigma)$) either $\limsup_{k\to\infty} j^n(\sigma_k) = \infty$ or $\liminf_{k\to\infty} \mu(S_{\sigma_k,\eta}^n) = 0$. In either case (see Steps 2-4), $\lim_{k\to\infty} n(k) = \infty$, a contradiction.

We thus have that $D_{\sigma_k} < -\zeta$ for all $k = 1, \ldots$, and $\lim_{k\to\infty} n(k) = \infty$. By Proposition 1 and Step 5 of the algorithm, for k large enough, $J(\sigma_{k+1}) - J(\sigma_k) < \alpha\lambda^{n(k)}(\sigma_k)D_{\sigma_k} \le -\alpha\lambda^{n(k)}(\sigma_k)\zeta$, and since $J(\sigma)$ is bounded from below over $\sigma \in \Sigma$, it follows that $\lim_{k\to\infty} \lambda^{n(k)}(\sigma_k) = 0$. By Lemma 1($iii$) and the assumption that $D_{\sigma_k} < -\zeta$, and by the definition of $\lambda^n(\sigma)$ and Proposition 1, this implies that $\lim_{k\to\infty} \mu(S_{\sigma_k,\eta}^{n(k)}) = 0$. Fix $\eta_2 \in (\eta, 1)$. By Lemma 2 (and especially the left inequality of (12)), we have that

$$\lim_{k\to\infty} \mu(S_{\sigma_k,\eta_2}) \;=\; 0. \tag{18}$$

Now by Proposition 1 and Lemma 1, there exists $k_1 \ge 1$ such that, for every $k \ge k_1$, $j^{n(k)}(\sigma_k) = 0$, namely the search for the Armijo step size ends at $j = 0$. With the aid of Lemma 3, $\liminf_{k\to\infty}(D_{\sigma_{k+1}} - \eta D_{\sigma_k}) \ge 0$. This, however, contradicts the assumption that $D_{\sigma_k} < -\zeta$ for all $k = 1, 2, \ldots$, thereby completing the proof. ∎

As in [15] for the case of exact evaluations, we suspect that the limit $\lim_{k\to\infty} D_{\sigma_k} = 0$ cannot be guaranteed in general. This is due to the lack of continuity of the sets $S_{\sigma,\eta}$ with respect to the schedules.

IV. Simulation Examples

We tested the algorithm on two examples: one concerns output tracking in a nonlinear system, and the other considers control of an unstable system. In both cases we decouple the grid used for numerical integration from the one used for computing D_σ^n, in order to focus on the effects of the latter. We set $t_f = 20$, and the integration grid consists of 2,000 equally-spaced points plus the switching times of the schedule in question.

Example 1. Consider the two-dimensional, single-input system whose state equation is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} v - \sqrt{x_1} \\ \sqrt{x_1} - \sqrt{x_2} \end{pmatrix}. \tag{19}$$

It describes a vertical arrangement of two tanks with drainage holes at the bottom, where fluid flows from an external spigot into the upper tank at the rate v, drains into the lower tank, from which it drains out of the system. x_1 and x_2 represent the fluid levels at the upper and lower tanks, and Equation (19) is based on Toricelli's law. We set the input v to have either value of 1 or 2, and the initial condition to be $x(0) = (2,2)^T$. The objective of the problem is to have the fluid level at the lower tank track a given value, $x_{2,ref} = 3$, and hence we choose the cost functional to be

$$J \ = \ 2\int_0^{20}\big(x_2-3\big)^2 dt. \qquad (20)$$

Initially the grid for computing D_σ^n (henceforth referred to as *the grid*) consists of 4 equally-spaced points, and it is later refined according to Steps 2-5 of Algorithm 2, with the following parameters: $\epsilon_{1,n} = 10/2^{n-1}$, $\epsilon_{2,n} = 10/2^{n-1}$, and $M_n = 9+n$. Each time the grid is refined, we double the number of its equally-spaced points (initially 4) and add the switching times of the schedule in question.

The algorithm was run for 71 iterations with the parameters $\alpha = 0.2$, $\eta = 0.5$, and $\beta = 0.5$, and the initial schedule $v_1(t) = \big\{1, \ \forall t \leq 10; \ 2, \ \forall t > 10\big\}$. The results are shown in Figures 2-4: Figure 2 depicts the graph of J versus the iteration count, k, and the vertical lines indicate the iterations at which the grid was refined; all of the refinements were due to $|D_\sigma^n|$ getting too small (Step 2). Initially $J(\sigma_1) = 71.03$, while at the end of the run $J(\sigma_{71}) = 4.783$, and most of the decline in the cost is obtained in about 10 iterations. Correspondingly, the approximate-optimality function, $D_{\sigma_k}^n$ increases from $D_{\sigma_1} = -12.54$ to $D_{\sigma_{71}} = -0.031$ (not shown in the figure). Extensive independent simulations at randomly-chosen 2,000 points near σ_{71} indicate that it is indeed a local minimum, or very close to one.

The final mode-schedule, $v_{\sigma_{71}}(t)$, is depicted in Figure 3, and it shows that initially $v = 2$ in order to bring x_2 close to the target level of 3, and thereafter the switching frequency increases in order to maintain the tracking. Finally, the state trajectory is shown in Figure 4, where x_2 tracks the desirable value of 3.0.

Example 2. Consider the two-dimensional linear system defined by the equation $\dot{x} = Ax$; the matrix A is switched between A_1 and A_2, defined as

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

according to whether $v = 1$ or $v = 2$. We set the initial condition to $x(0) = (3,3)^T$. Both modes are unstable, and the objective is to switch among them so as to minimize the extent to which the state runs off to ∞; hence we define the cost functional as

$$J \ = \ \frac{1}{2}\int_0^{20}||x(t)||^2 dt.$$

The grid, initially consisting of 4 equally-spaced points, is handled in the same way as in Example 1, with the following parameters: $\epsilon_{1,n} = 10{,}000/2^{n-1}$, $\epsilon_{2,n} = 2/2^{n-1}$, and $M_n = 9+n$.

The algorithm was run for 287 iterations with $\alpha = 0.2$, $\eta = 0.5$, and $\beta = 0.5$, and the initial schedule $v_1(t) = 2 \ \forall t \in [0,20]$. The results are shown in Figures 5-7: Figure 5 depicts the graph of J versus the iteration count, k, and the vertical lines indicate the iterations at which the grid was refined (mostly at Step 2). The graph shows the cost functional declining, mostly during the first 80 iterations, from its initial value of $J(\sigma_1) = 5.4\times10^{17}$ to its final value of $J(\sigma_{287}) = 1{,}043$. Correspondingly, $D_{\sigma_k}^n$ increases from $D_{\sigma_1} = -2.182\times10^{18}$ to $D_{\sigma_{287}} = -99.36$ (not shown in the figure). These numbers are not as small (in absolute value) as in Example 1, and this is due to the instability of the system. Extensive independent simulations at randomly-chosen 2,000 points near σ_{287} indicate that it is indeed a local minimum, or very close to one.

The final mode-schedule, $v_{\sigma_{287}}(t)$, is depicted in Figure 6, and it indicates a sliding-mode control that drives the state trajectory to follow a direction of least growth. Plots of the state trajectory at σ_{287} are shown in Figure 7, where fluctuations from well-defined directions, likely due to the system's instability, are apparent. However, an average direction is discernable.

V. Conclusions

This paper proposes an adaptive-precision algorithm for optimal mode-scheduling in switched-mode systems. It is based on simultaneous swapping of modes at uncountable time-sets whose Lebesgue measures are determined by the Armijo step size, and the precision is controlled by the values of optimality functions. Future research will test the algorithm on large-scale problems, where an adequate balance between precision and computational complexity may be crucial for its efficacy.

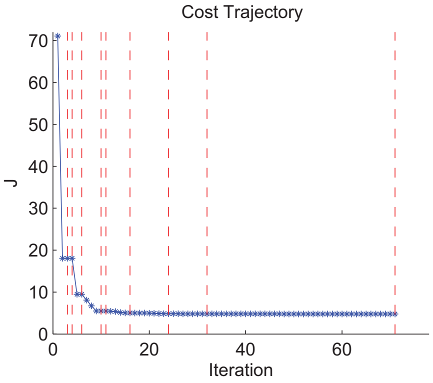

Fig. 2. Example 1: Graph of $J(\sigma_k)$.

References

[1] 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.

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

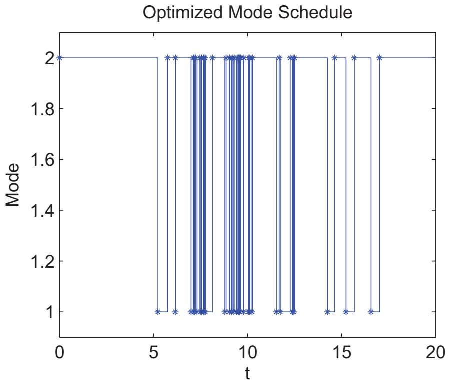

Fig. 3. Example 1: Graph of $v_{\sigma_{71}}$.

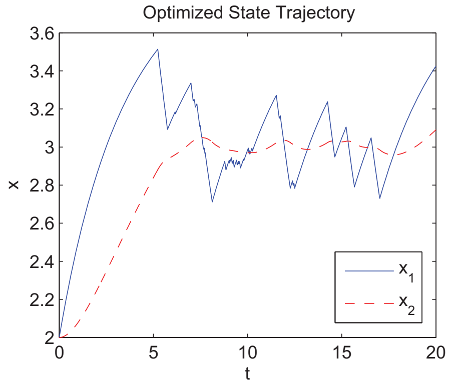

Fig. 4. Example 1: State trajectory at the last iteration.

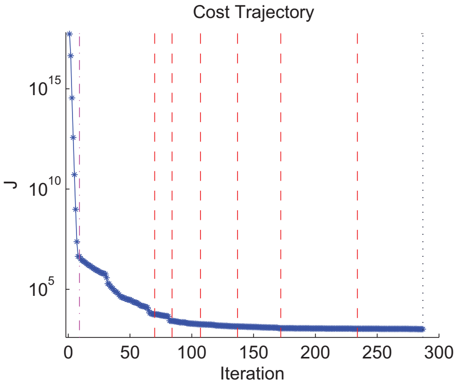

Fig. 5. Example 2: Graph of $J(\sigma_k)$.

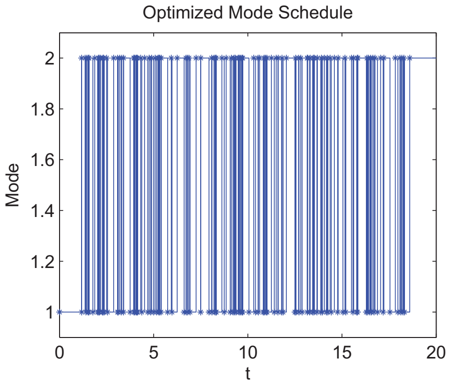

Fig. 6. Example 2: Graph of $v_{\sigma_{287}}$

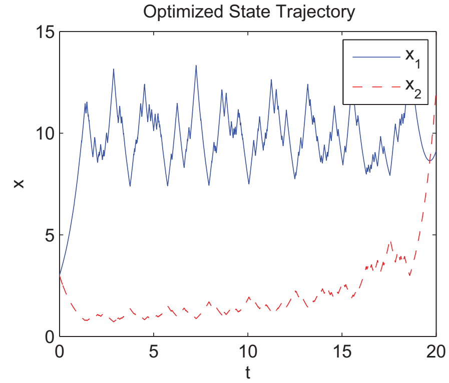

Fig. 7. Example 2: State trajectory at the last iteration.

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

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

[5] 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.

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

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

[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 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.

[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. Scheduling Optimization in Autonomous Switched-Mode Dynamical Systems. Submitted to the *IEEE Transactions on AC*, 2011.

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

[17] 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.