Algorithm for Switching-Time Optimization in Hybrid Dynamical Systems

H. Axelsson, M. Egerstedt*, Y. Wardi†, and G. Vachtsevanos

{henrik,magnus,ywardi,gjv}@ece.gatech.edu
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

***Abstract*— We consider the problem of minimizing a cost functional defined on the state trajectory of a switched-mode dynamical system with respect to the switching times. Following the derivation, in recent years, of various results concerning the gradient of the cost functional, we present a suitable algorithm, based on gradient projection, for computing local minima. Utilizing the problem's special structure, we prove a convenient formula for the direction of descent, and apply the Armijo procedure for computing the step size. A potential extension to the optimal mode-insertion problem is discussed, and numerical examples are provided.**

Keywords. Hybrid dynamical systems, switched-mode systems, optimal control, optimization algorithms.

I. Introduction

This paper concerns switched-mode systems whose dynamic response changes among various modes according to a prescribed supervisory-control law. Let $\{x(t)\}_{t=0}^{T}$ denote the state trajectory of the system, and suppose that it evolves according to the equation $\dot{x} = f(x,t)$, where the dynamic-response function $f : R^n \times [0,T] \rightarrow R^n$ is comprised of a sequential assignment of functions $f_i : R^n \rightarrow R^n$, $i = 1, 2, \ldots,$. Let us fix $T > 0$ and suppose that the dynamic response changes N times in the interval $[0,T]$. Denoting the switching times by τ_i, $i = 1, \ldots, N$, in increasing order, and defining $\tau_0 := 0$ and $\tau_{N+1} := T$, we have that

$$0 = \tau_0 \leq \tau_1 \leq \ldots, \leq \tau_N \leq \tau_{N+1} = T. \quad (1)$$

Furthermore, suppose that $f(x,t) = f_i(x)$ for every $t \in [\tau_{i-1}, \tau_i)$ and for every $i = 1, \ldots, N+1$. We then have the following differential equation defining the system's dynamics,

$$\dot{x}(t) = f_i(x(t)), \qquad t \in [\tau_{i-1}, \tau_i), \qquad i \in \{1, \ldots, N+1\}. \quad (2)$$

We assume throughout that the initial condition $x(0)$ is given and fixed. A more general setting includes an exogenous input $u(t)$ in the right-hand side of (2), but in this paper we concern only the autonomous case where such an input is absent.

*Research partly supported by the National Science Foundation under Grant # 0237971 ECS NSF-CAREER (Grant # 0237971).

†Corresponding author.

Such systems arise in a variety of applications, including situations where a control module has to switch its attention among a number of subsystems [13], [18], [22], or collect data sequentially from a number of sensory sources [5], [7], [12]. One of the problems of interest is how to determine the switching schedule in order to optimize a cost criterion of the form

$$J = \int_0^T L(x)dt, \quad (3)$$

for a given cost function $L : R^n \rightarrow R$. The term "schedule" used above means two kinds of parameters: the sequencing of the modal functions f_i in the right-hand side of (2), and the switching times τ_i for a given sequence of functions. The problem of determining the optimal switching is known to be NP hard like many scheduling problems. In contrast, the problem of determining the optimal switching times for a given sequence of modal functions can be viewed as an optimal control problem and be solved numerically by nonlinear programming algorithms.

Recently there has been a mounting interest in the optimal control problem in the broader, non-autonomous setting, where the control variable consists of a proper switching law as well as the input function $u(t)$ (see [4], [6], [10], [11], [14], [19], [20], [21], [23], [25]). Ref. [4] establishes a framework for optimal control, and [19], [20], [21] present suitable variants of the maximum principle to the setting of hybrid systems. Moreover, Refs. [2], [3], [10], [17] consider the case of piecewise-linear or affine systems. The general nonlinear case of autonomous systems, where the term $u(t)$ is absent and the control variable consists solely of the switching times, is considered in [8], [12], [24], [25], [26]. In Ref. [8] we develop a simple formula for the partial derivatives $\frac{dJ}{d\tau_i}$ and subsequently propose a gradient-projection algorithm for computing optimal switching times. This paper derives a simple formula for computing the projection of the gradient onto the feasible set, which provides the descent direction for the algorithm. We also discussed in [8] a potential way of extending the algorithm to the sequencing problem, and here we elaborate on it and provide a numerical example of such an extension.

The rest of the paper is organized as follows. Section II concerns the above optimal control problem, and Section III presents a numerical example. Section IV concludes the paper.

II. Gradient-Projection Algorithm for the Optimal Control Problem

Consider the systems defined by Eq. (2) and the cost functional defined in (3). We denote by σ the sequence $\{f_1,\ldots,f_{N+1}\}$, and call it the *modal sequence*. For the rest of the discussion in this section, the modal sequence $\{f_1,\ldots,f_{N+1}\}$ is assumed to be given and fixed. Now the functional J can be viewed as a function of the switching times, $\tau_1 \leq \tau_2 \leq \ldots, \leq \tau_N$, via Es. (2) and (3). Collectively, these switching times constitute a vector in R^N, henceforth denoted by $\bar{\tau} := (\tau_1,\ldots,\tau_N)^T$, and the dependence of J upon it is indicated in the notation $J(\bar{\tau})$. The optimal control problem that we consider, denoted by P_σ, is defined as follows.

- P_σ: Minimize $J(\bar{\tau})$ subject to the constraints in Eq. (1).

The following assumption will be made.

Assumption 2.1. (i). The functions f_i and L are twice continuously differentiable on R^n.
(ii) There exists a constant $K_0 > 0$ such that, for every $x \in R^n$, and for all $i \in \{1,\ldots,N+1\}$,

$$||f_i(x)|| \ \leq \ K_0(||x||+1). \tag{4}$$

Ref. [8] derived a formula for the derivatives $\frac{dJ}{d\tau_i}$, $i = 1,\ldots,N$ (and hence the gradient $\nabla J(\bar{\tau})$) and proposed a gradient-descent algorithm. These results are next surveyed.

Define the costate $p(t) \in R^n$ by the following differential equation,

$$\dot{p} \ = \ -\big(\frac{\partial f_{i+1}}{\partial x}(x)\big)^T p - \big(\frac{\partial L}{\partial x}(x)\big)^T, \quad t \in (\tau_i,\tau_{i+1}], \quad i = N, N-1,\ldots,0, \tag{5}$$

with the boundary condition $P(T) = 0$. Then (see [8], Proposition 2.2),

$$\frac{dJ}{d\tau_i} \ = \ p(\tau_i)^T\big(f_i(x(\tau_i)) - f_{i+1}(x(\tau_i))\big). \tag{6}$$

Next, let us denote the feasible set defined by Eq. (1) by Φ, namely,

$$\Phi := \{\bar{\tau} \in R^N \ : \ 0 \leq \tau_1 \leq \ldots, \leq \tau_N \leq T\}. \tag{7}$$

Based on Assumption 2.1, Ref. [8] proved that the function $J(\bar{\tau})$ is continuously differentiable on Φ, and hence the Lagrange multiplier rule provides a necessary optimality condition. We henceforth dub this condition *the standard local optimality condition for* P_σ. In order to solve the problem P_σ to the extent of computing a point satisfying the above optimality condition, Ref. [8] proposed to use a gradient-projection algorithm with Armijo step sizes. This algorithm is next described. Given a point $\bar{\tau} \in \Phi$, let $\Psi(\bar{\tau})$ denote the set of feasible directions from the point $\bar{\tau}$, namely,

$$\Psi(\bar{\tau}) \ := \ \{\bar{h} \in \mathbb{R}^N \ \ | \ \ \text{for some } \tilde{\zeta} > 0, \ \text{and for all } \zeta \in [0,\tilde{\zeta}), \quad \bar{\tau} + \zeta\bar{h} \in \Phi\}. \tag{8}$$

Then the algorithm uses as the direction of descent, denoted by $\bar{h}(\bar{\tau})$, the projection of the vector $-\nabla J(\bar{\tau})$ onto the set $\Psi(\bar{\tau})$. The step size, denoted by ζ, is computed according to the Armijo procedure defined as follows (see [1]). Given constants $\alpha \in (0,1)$, $\beta \in (0,1)$, and an integer k_0, define ζ by

$$\zeta \ = \ \text{argmax}\{z = \beta^k; \ k \geq k_0 \ \ | \ \ \bar{\tau} + z\bar{h}(\bar{\tau}) \in \Phi, \ \text{and} \ \ J(\bar{\tau} + z\bar{h}(\bar{\tau})) - J(\bar{\tau}) \ \leq \alpha z < \bar{h}(\bar{\tau}), \nabla J(\bar{\tau}) >\}. \tag{9}$$

Putting it all together, we have the following algorithm.

Algorithm 2.1. Gradient-Projection Algorithm with Armijo Step Sizes.
Given: Constants $\alpha \in (0,1)$, $\beta \in (0,1)$, and an integer k_0.
Initialize: Choose an initial point $\bar{\tau}_0 \in \Phi$. Set $i = 0$.
Step 1: Compute the feasible descent direction $\bar{h}(\bar{\tau}_i)$, the projection of $-\nabla J(\bar{\tau}_i)$ onto the set $\Psi(\bar{\tau}_i)$.
Step 2. Compute the Armijo step size ζ_i by (9).
Step 3: Set $\bar{\tau}_{i+1} := \bar{\tau}_i + \zeta_i\bar{h}(\bar{\tau}_i)$, set $i = i+1$, and go to Step 1. ■

This algorithm is known to solve problems like P_σ (see [16]) in the sense that, any accumulation point of a sequence $\{\bar{\tau}_i\}_{i=1}^\infty$ that it computes, satisfies the standard local optimality condition for P_σ. In order to deploy this algorithm we need an expression for the projected gradient, $\bar{h}(\bar{\tau})$. Such an expression is next derived.

Fix $\bar{\tau} := (\tau_1,\ldots,\tau_N)^T \in \Phi$. Then (see (7)) $0 \leq \tau_1 \leq \ldots, \leq \tau_N \leq T$. Let us define a *block* to be a contiguous integer-set $\{k,\ldots,n\} \subset \{1,\ldots,N\}$ such that $\tau_n = \tau_k$ (and hence $\tau_i = \tau_k$ for all $i \in \{k,\ldots,n\}$). Observe that every set of contiguous integers that is a subset of a block is also a block. Furthermore, we say that a block is *maximal* if no superset thereof is a block. Obviously, the set $\{1,\ldots,N\}$ is partitioned into disjoint maximal blocks in a way that depends on $\bar{\tau}$. Moreover, for each maximal block $\{k,\ldots,n\}$, $\tau_{k-1} < \tau_k$ unless $k = 1$ and $\tau_1 = 0$, and $\tau_n < \tau_{n+1}$ unless $n = N$ and $\tau_N = T$.

We next describe an algorithm for computing a certain vector denoted by $\bar{h} := (h_1,\ldots,h_N)^T$, and then we will show that $\bar{h} = \bar{h}(\bar{\tau})$. The coordinates h_i will be computed according to the various maximal blocks associated with the given point $\bar{\tau}$. Let us fix such a maximal block, denoted by $\{k,\ldots,n\}$. For every $\ell \in \{k,\ldots,n\}$, and for every $i \in \{\ell,\ldots,n\}$, we define $r_{\ell,i}$ by

$$r_{\ell,i} \ = \ \frac{1}{i-\ell+1}\sum_{j=\ell}^{i}\frac{dJ(\bar{\tau})}{d\tau_j}. \tag{10}$$

Algorithm 2.2. Procedure for computing h_i, $i = k,\ldots,n$.
Step 0. Set $\ell = k$.

Step 1. Compute r_{max} defined by

$$r_{max} \ := \ \max\{r_{\ell,i} \ \mid \ i=\ell,\ldots,n\}. \qquad (11)$$

Set $m := \max\{i=\ell,\ldots,n \ : \ r_{\ell,i}=r_{max}\}$.
Step 2. For all $i \in \{\ell,\ldots,m\}$, define $h_i(\bar{\tau})$ by $h_i(\bar{\tau}) = -r_{max}$ unless either (i) $\tau_m = 0$ and $r_{max} > 0$, or (ii) $\tau_m = T$ and $r_{max} < 0$. In either case (i) or (ii), set $h_i(\bar{\tau}) = 0$.
Step 3. If $m = n$, exit. If $m < n$, set $\ell := m+1$ and go to Step 1. ∎

Proposition 2.1. The vector $\bar{h}$ computed by Algorithm 2.2 for every maximal block is indeed the vector $\bar{h}(\bar{\tau})$, namely the projection of $-\nabla J(\bar{\tau})$ onto the feasible set $\Psi(\bar{\tau})$.

Proof. Let us denote by ℓ_q the value of ℓ when Algorithm 2.2 enters its Step 1 in the q^{th} time, $q=1,2,\ldots$,. Likewise, let m_q denote the value of m computed in Step 1 at the q^{th} iteration. Thus, $\ell_1 = k$, m_1 is computed by Step 1 in the first iteration, $\ell_2 = m_1 + 1$ unless $m_1 = n$, etc. Correspondingly, we denote by $r_{max,q}$ the value of r_{max} computed via (11) in the q^{th} iteration of the algorithm.

Let $\bar{h} = (h_1,\ldots,h_N)^T$ be the vector computed by Algorithm 2.2 as applied to every maximal block. We first ascertain that $\bar{h} \in \Psi(\bar{\tau})$. Observe that $\bar{h} \in \Psi(\bar{\tau})$ if and only if, for every maximal block such as $\{k,\ldots,n\}$, the following three equations are in force.

$$h_k \geq 0 \text{ if } \tau_k = 0. \qquad (12)$$

$$h_i \leq h_{i+1} \text{ for all } i \in \{k,\ldots,n-1\}. \qquad (13)$$

$$h_n \leq 0 \text{ if } \tau_n = T. \qquad (14)$$

We next prove that these three conditions are indeed in force.

Suppose first that $0 < \tau_k$ and $\tau_n < T$; the cases where $\tau_k = 0$ or $\tau_k = T$ will be considered later. Then, only Eq. (13) has to be ascertained. For all $q = 1,2,\ldots$, and for all $i \in \{\ell_q,\ldots,m_q\}$, $h_i = -r_{\ell_q,m_q}$ by Step 1. Therefore, (13) will be established once we show that

$$r_{\ell_q,m_q} \ > \ r_{\ell_{q+1},m_{q+1}}. \qquad (15)$$

This is what we next do. By Step 1, $r_{\ell_q,m_q} > r_{\ell_q,m_{q+1}}$. This means, by (10), that

$$\frac{1}{m_q-\ell_q+1}\sum_{j=\ell_q}^{m_q}\frac{dJ(\bar{\tau})}{d\tau_j} \ > \ \frac{1}{m_{q+1}-\ell_q+1}\sum_{j=\ell_q}^{m_{q+1}}\frac{dJ(\bar{\tau})}{d\tau_j}. \qquad (16)$$

Consequently, and after some algebra, we obtain that

$$\frac{1}{m_{q+1}-m_q}\sum_{j=m_q+1}^{m_{q+1}}\frac{dJ(\bar{\tau})}{d\tau_j} \ < \ \frac{1}{m_q-\ell_q+1}\sum_{j=\ell_q}^{m_q}\frac{dJ(\bar{\tau})}{d\tau_j}. \qquad (17)$$

Applying the fact that $\ell_{q+1} = m_q + 1$ (by definition) to the left-hand side of (17), Eq. (15) follows by (10).

Next, consider the case where $\tau_{m_q} = 0$. By Step 2, we see that, for all $i \in \{\ell_q,\ldots,m_q\}$, $h_i = \max\{-r_{m_q},0\}$, and this ascertains that both Eqs. (12) and (13) are in force. Finally, it follows in a similar way that, if $\tau_n = T$, then Eqs. (13) and (14) hold true. This establishes that $\bar{h} \in \Psi(\bar{\tau})$.

We next prove that $\bar{h}$ indeed is the projection of $-\nabla J(\bar{\tau})$ onto $\Psi(\bar{\tau})$. To be the projection, $\bar{h}$ has to solve the following quadratic program, denoted by Q.

$$Q \ : \ \min\{\frac{1}{2}||\tilde{h}+\nabla J(\bar{\tau})||^2 \ : \ \tilde{h} \in \Psi(\bar{\tau})\}. \qquad (18)$$

Associated with the vector $\bar{h}$, let us define the vector $\bar{h}_{k,n} \in R^{n-k+1}$ by $\bar{h}_{k,n} = (h_k,\ldots,h_n)^T$. Similarly, we define the vector $\nabla_{k,n}J(\bar{\tau}) \in R^{n-k+1}$ by $\nabla_{k,n}J(\bar{\tau}) = (\frac{dJ(\bar{\tau})}{d\tau_k},\ldots,\frac{dJ(\bar{\tau})}{d\tau_n})^T$. Now the condition $\bar{h} \in \Psi(\bar{\tau})$ is equivalent to the condition that, for every maximal block like $\{k,\ldots,n\}$, Eqs. (12)-(14) are in force. Therefore, $\bar{h}$ solves the quadratic program Q if and only if, for every maximal block like $\{k,\ldots,n\}$, $\bar{h}_{k,n}$ solves the following quadratic program, denoted by $Q_{k,n}$.

$$Q_{k,n}: \quad \min\{\frac{1}{2}||\tilde{h}_{k,n}+\nabla_{k,n}J(\bar{\tau})||^2 \ : \ \text{Eqs. } (12)-(14) \text{ are satisfied}\}. \qquad (19)$$

This is a quadratic program that has a unique solution point, which is also the only point satisfying the Lagrange multiplier rule for $Q_{k,n}$. We now show that this point is $\bar{h}_{k,n}$.

We observe that for every $q = 1,2,\ldots$, and for every $i \in \{\ell_q,\ldots,m_q-1\}$, $h_i - h_{i+1} = 0$, i.e., the constraint $h_i - h_{i+1} \leq 0$ is active. Moreover, by Step 2 and (15), $h_{m_q} < h_{m_q+1}$, and hence the constraint $h_{m_q} - h_{m_q+1} \leq 0$ is not active. Therefore, to satisfy the Lagrange multiplier rule for $Q_{k,n}$, there must exist a multiplier $\lambda_i \geq 0$ associated with the inequality constraint $h_i - h_{i+1} \leq 0$, $i = \ell_q,\ldots,m_q-1$; in the case where $q = 1$ and hence $\ell_q = k$, if $\tau_k = 0$ and $h_k = 0$ then there exists a multiplier $\mu \geq 0$ associated with the inequality $-h_k \leq 0$; and in the case where $m_q = n$, if $\tau_n = T$ and $h_n = 0$ then there exists a multiplier $\nu \geq 0$ associated with the constraint $h_n \leq 0$. The latter two situations cannot arise simultaneously since $\tau_k = \tau_n$ (by dint of the fact that the set $\{k,\ldots,n\}$ constitutes a maximal block), therefore we can assume, without loss of generality, that $\tau_1 > 0$ and consider only the possible case where $\tau_n = T$.

Consider first the case where either $\tau_n < T$ or $h_n < 0$. Then, the above Lagrange multiplier rule means that the following three equations are in force,

$$h_{\ell_q} + \frac{dJ(\bar{\tau})}{d\tau_{\ell_q}} + \lambda_{\ell_q} \ = \ 0, \qquad (20)$$

$$h_i + \frac{dJ(\bar{\tau})}{d\tau_i} + \lambda_i - \lambda_{i-1} \ = \ 0 \quad \text{for all } i = \ell_q+1,\ldots,m_q-1, \qquad (21)$$

$$h_{m_q} + \frac{dJ(\bar{\tau})}{d\tau_{m_q}} - \lambda_{m_q-1} \ = \ 0. \qquad (22)$$

For every $j \in \{\ell_q+1,\ldots,m_q-1\}$, summing up (20) with (21) for all $i = \ell_q+1,\ldots,j$, we obtain,

$$\sum_{i=\ell_q}^{j} h_i + \sum_{i=\ell_q}^{j} \frac{dJ(\bar{\tau})}{d\tau_i} + \lambda_j \;=\; 0. \tag{23}$$

Likewise, summing up over all $i \in \{\ell_q + 1, \ldots, m_q - 1\}$ and adding (20) and (22), we get that

$$\sum_{i=\ell_q}^{m_q} h_i + \sum_{i=\ell_q}^{m_q} \frac{dJ(\bar{\tau})}{d\tau_i} \;=\; 0. \tag{24}$$

In fact, it is readily seen that the condition defined by Eqs. (20)-(22) is equivalent to the condition defined by (20), (23) for all $j \in \{\ell_q + 1, \ldots, m_q - 1\}$, and (24). Since $\lambda_i \geq 0$, the latter condition amounts to the condition defined by the following two equations.

$$\sum_{i=\ell_q}^{j} h_i + \sum_{i=\ell_q}^{j} \frac{dJ(\bar{\tau})}{d\tau_i} \;\leq\; 0, \qquad j = \ell_q, \ldots, m_q - 1; \tag{25}$$

$$\sum_{i=\ell_q}^{m_q} h_j + \sum_{i=\ell_q}^{m_q} \frac{dJ(\bar{\tau})}{d\tau_i} \;=\; 0. \tag{26}$$

By (10) and Step 2, (25) amounts to

$$-(j-\ell_q+1)r_{\ell_q,m_q} - \ell_q, m_q + (j-\ell_q+1)r_{\ell_q,j} \;\leq\; 0, \tag{27}$$

and (26) means that

$$-(m_q - \ell_q + 1)r_{\ell_q,m_q} + (m_q - \ell_q + 1)r_{\ell_q,m_q} \;=\; 0. \tag{28}$$

(27) is satisfied by the maximality of m_q (Eq. (11)), and (28) is certainly true. This shows that $\bar{h}_{k,n}$ satisfies the Lagrange multiplier rule for $Q_{k,n}$, and hence it is the solution point for that quadratic program.

Finally, consider the case where $m_q = n$, $\tau_n = T$, and $h_n = 0$. Then we have the additional Lagrange multiplier $\nu \geq 0$ associated with the active inequality constraint $\tau_n - T \leq 0$. Consequently, the optimality condition amounts to Eq. (25) for all $i = \ell_q, \ldots, n-1$, but not (26); instead, we have the condition $h_n = 0$. By Step 2, the latter condition means that $h_i = 0$ for all $i = \ell_q, \ldots, n$, and, moreover, this can occur if and only if $r_{\ell_q,n} \leq 0$. In this case it is readily seen that Eq. (25) is in force, since $h_i = 0$ and (by (10))

$$\sum_{i=\ell_q}^{j} \frac{dJ(\bar{\tau})}{d\tau_i} \;\leq\; (j-\ell_q+1)r_{\ell_q,j} \;\leq\; (j-\ell_q+1)r_{\ell_q,n} \;\leq 0$$

for all $j = \ell_q, \ldots, n$. This shows that $\bar{h}_{k,n}$ indeed solves the quadratic program $Q_{k,n}$, which completes the proof. ■

III. Example

Consider the problem of controlling the fluid level at a tank by adjusting the input flow rate at an auxiliary tank, as shown in figure 1. This problem was considered in [15], and it serves to illustrate the utility of our algorithm. As can be seen in the figure, fluid from the upper tank flows to the lower tank through a valve, and fluid from the lower tank flows out through another valve. Fluid is discharged from either tank at a rate that is proportional to the square-root of the fluid level at the tank. Let x_1 and x_2 denote the fluid levels at the upper tank and the lower tank, respectively. Then, there exist constants $\alpha_1 > 0$ and $\alpha_2 > 0$ such that, the flow rate between the two tanks is $\alpha_1\sqrt{x_1}$, and the flow rate out of the lower tank is $\alpha_2\sqrt{x_2}$. Suppose that the control parameter is the inflow rate to the upper tank, denoted by u. Then, the state $x(t)$, defined by $x(t) = (x_1(t), x_2(t))^T$, evolves according to the following differential equation,

$$\dot{x} = f(x,u) = \begin{bmatrix} -\alpha_1\sqrt{(x_1)} + u \\ \alpha_1\sqrt{(x_1)} - \alpha_2\sqrt{(x_2)} \end{bmatrix}. \tag{29}$$

Suppose that the input valve can be in either one of three states: closed, half-open, or fully open. Correspondingly, the input flow rate $u(t)$ can have one of the following three values, 0, $0.5u_{max}$, and u_{max}, for some given $u_{max} > 0$. Corresponding to the three values of $u(t)$ we denote the right-hand side of (29) by $f^0(x)$ when $u = 0$, by $f^1(x)$ when $u = 0.5u_{max}$, and by $f^2(x)$ when $u = u_{max}$. Given an initial state $x(0)$ and a final time $T > 0$, the objective of a switching-control strategy is to have the fluid level in the lower tank track a given value, denoted by x_r. Thus, the performance functional that we minimize is

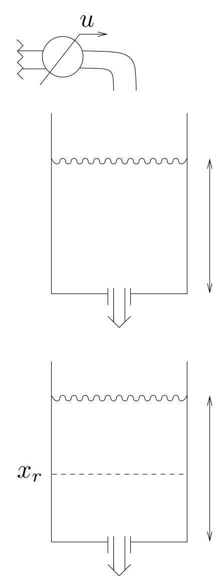

Fig. 1. The double tank process.

$$J = K\int_0^T (x_2(t) - x_r)^2 dt, \tag{30}$$

for a suitable constant $K > 0$.

Given a modal sequence $\sigma = \{f_1, \ldots, f_{N+1}\}$, the optimization problem P_σ amounts to computing the switching-time vector $\bar{\tau} = (\tau_1, \ldots, \tau_N)^T$ that minimizes the cost functional J. To this end we deploy the gradient-descent algorithm with Armijo step sizes [1]. This algorithm guarantees that, asymptotically, the projected gradients of the iterate sequence converge to 0 (see [16]), but practically we stop its execution once the magnitude of the projected gradient falls below a prescribed quantity, $\epsilon > 0$.

At this point we modify the modal sequence by adding new modal functions. To see how this is done, we recall the following result from [8]. Let us fix $\bar{\tau} = (\tau_1, \ldots, \tau_N)^T \in \Phi$, and let $\{x(t)\}$ and $\{p(t)\}$ be the associated state trajectory and costate trajectory defined, respectively, by Eqs. (2) and (5). Let g be a modal function satisfying Assumption 2.1. Consider modifying the modal sequence by inserting the modal function g at a time-point $\tau \in [\tau_i, \tau_{i+1})$ for some $i \in \{1, \ldots, N\}$, for the duration of λ seconds, for $\lambda > 0$ (assume that $\tau + \lambda < T$). Let $J(\lambda)$ denote the value of the cost functional J as a function of λ. Then (see [8], Proposition 4.1), the one-sided derivative $\frac{dJ}{d\lambda^+}(0)$ has the following form,

$$\frac{dJ}{d\lambda^+}(0) \;=\; p(\tau)^T\big(g(x(\tau)) - f_i(x(\tau))\big). \qquad (31)$$

Of course the above one-sided derivative is a function of both $\tau \in [\tau_i, \tau_{i+1})$ and the modal function g, and hence we denote it by $\frac{dJ(\tau;g)}{d\lambda^+}(0)$. If $\frac{dJ(\tau;g)}{d\lambda^+}(0) < 0$ then such an insertion for a "small enough" λ would result in a decrease in the cost J.

Suppose that there is a prescribed rule allowing g to be picked only from a given finite set of modal functions, denoted by G_i, which may depend on f_i. Let us define m_i by

$$m_i \;=\; \min\{\frac{dJ(\tau;g)}{d\lambda^+}(0)\;:\;\tau \in [\tau_i, \tau_{i+1}];\;\; g \in G_i\}, \quad (32)$$

and pick $\tau^i \in [\tau_i, \tau_{i+1}\}$ and $g^i \in G_i$ such that

$$\frac{dJ(\tau^i;g^i)}{d\lambda^+}(0) \;=\; m_i. \qquad (33)$$

If $m_i < 0$ then we insert the modal function g^i at the point τ^i. In fact, τ^i constitutes a double switching point for this insertion, and the domain of the modal function g^i is the empty set $[\tau^i, \tau^i)$. However, as we next continue with the gradient-descent algorithm, the projected gradient will separate the two insertion points and enlarge the domain of g^i. All of this is formalized in the following algorithm.

Algorithm 3.1. At a given iteration, we have a modal sequence $\sigma = \{f_i\}_{i=1}^{N+1} \in [0, T]$.
Step 1. Use Algorithm 2.1 to solve the optimization problem P_σ to the extent of computing a parameter $\bar{\tau} = (\tau_1, \ldots, \tau_N)^T$ such that $||\bar{h}(\bar{\tau})|| \leq \epsilon$.
Step 2. For every $i = 0, \ldots, N$, compute m_i by (32). If $m_i < 0$, then compute τ^i and g^i by (33), and modify the modal sequence σ in the interval $[\tau_i, \tau_{i+1}]$ by adding a double switching times at the point τ^i, and the modal function g^i between them. ■

This algorithm goes back from Step 2 to Step 1 unless $m_i \geq 0$ for all $i = 0, \ldots, N$, in which case it is terminated.

We implemented the algorithm to solve the optimal control problem with the cost function (30) and the following parameters: $K = 10.0$, $x_r = 0.5$, $T = 5.0$, $\epsilon = 0.1$, $\alpha_1 = \alpha_2 = 1.0$, $u_{max} = 1.0$, and $x(0) = (0.8, 0.2)^T$. We started the algorithm by having a single modal function, f_1, throughout the interval $[0, T)$, and hence the algorithm goes directly to Step 2. There, it inserts the function f_2 at the time $t = 1.80$, resulting in the modal sequence $\{f_1, f_2, f_1\}$ and the corresponding switching vector $\bar{\tau} = (1.80, 1.80)^T$. Then the algorithm returns to Step 1, where it computes the switching-times vector $\bar{\tau} = (1.26, 2.40)^T$ for the above modal sequence. Returning to Step 2, it computes the modal sequence $\{f_1, f_2, f_1, f_2, f_0, f_2, f_1, f_2, f_1\}$ with the corresponding switching-time vector $\bar{\tau} = (0.01, 0.01, 1.26, 1.63, 1.63, 2.40, 3.58, 3.58)^T$. Next, in Step 1, it computes its final vector $\bar{\tau} = (0, 0.40, 1.30, 1.49, 1.76, 2.46, 3.33, 3.84)^T$, at which point it is terminated.

The results are illustrated in Figure 2, Figure 3, and Table 1. Figure 2 shows the state trajectory associated with the final point, and it can be seen that, indeed, x_2 tracks x_r. It took the algorithm 10 iterations of the gradient-projection procedure (all computed in Step 1) to converge, where after iteration 4 it adds three double switching times. The cost function and the magnitude of the projected gradient are plotted, as functions of the iteration counts, in Figure 2, and Table 1 summarizes the results.

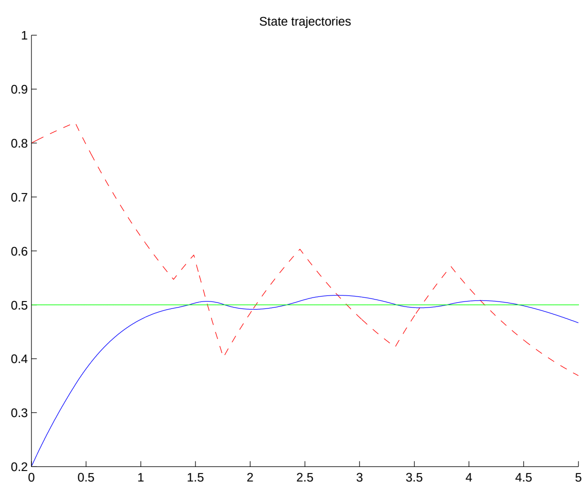

Fig. 2. State trajectory at the final point ($x_1(t)$ dotted, and $x_2(t)$ solid).

IV. Conclusions

This paper concerns an optimal control problem defined on switched-mode dynamical systems, whose variational parameter consists of the switching times. It first considers a fixed sequence of modal functions, and it derives a formula for the projected gradient of the cost functional. It then considers a way to reduce the cost by inserting additional modal functions to the schedule. A numerical example illustrates the details of the algorithm and points out its

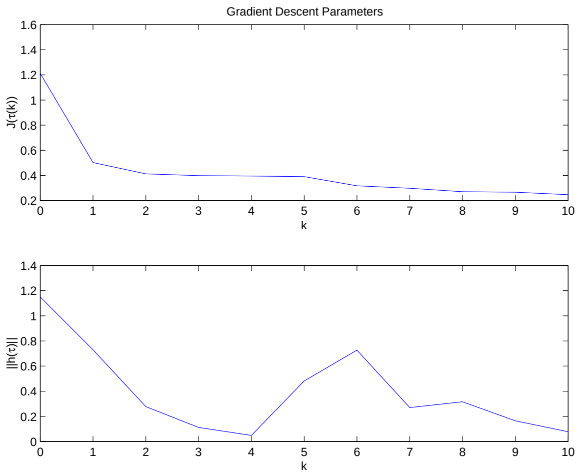

Fig. 3. Cost (top) and the norm of the projected gradient (bottom) for every iteration.

potential use in scheduling problems, where the modal sequence constitutes part of the variable parameter.

Acknowledgments

The work of the Magnus Egerstedt was sponsored in part by ECS NSF-CAREER award (grant # 0237971). The work by Yorai Wardi Henrik Axelsson was sponsored in part through a grant from the Georgia Institute of Technology Manufacturing Research Center.

REFERENCES

[1] L. Armijo. Minimization of Functions Having Lipschitz Continuous First-Partial Derivatives. *Pacific Journal of Mathematics*, Vol. 16, pp. 1-3, 1966.

[2] A. Bemporad, F. Borrelli, and M. Morari. Piecewise Linear Optimal Controllers for Hybrid Systems. In *Proceedings of the American Control Conference*, pp. 1190-1194, Chicago, Illinois, June 2000.

[3] A. Bemporad, F. Borrelli, and M. Morari. On the Optimal Control Law for Linear Discrete Time Hybrid Systems. In *Hybrid Systems: Computation and Control*, M. Greenstreet and C. Tomlin, Editors, Springer-Verlag Lecture Notes in Computer Science, Number 2289, pp. 105-119, 2002.

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

[6] J. Chudoung and C. Beck. The Minimum Principle for Deterministic Impulsive Control Systems. *IEEE Conference on Decision and Control*, Vol. 4, pp. 3569 -3574, Dec. 2001.

[7] M. Egerstedt and Y. Wardi. Multi-Process Control Using Queuing Theory. *IEEE Conference on Decision and Control*, Las Vegas, NV, Dec. 2002.

[8] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-Time Optimization for Switched-Mode Dynamical Systems. Submitted to the *IEEE Trans. on Automatic Control*, 2003.

[9] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal Control of Switching Times in Switched Dynamical Systems. To appear in the *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, Dec. 2003.

[10] A. Guia, C. Seatzu, and C. Van der Mee. Optimal Control of Switched Autonomous Linear Systems. In *Proceedings of the 40th Conference on Decision and Control*, pp. 1816-1821, Phoenix, Arizona, December 1999.

[11] S. Hedlund and A. Rantzer. Optimal Control of Hybrid Systems. *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 3972-3977, 1999.

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

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

[14] B. Lincoln and A. Rantzer. Relaxed Optimal Control of Piecewise Linear Systems. In *Proceedings of IFAC Conference on Analysis and Design of Hybrid Systems*, St. Malo, France, 2003.

[15] J. Malmborg and J. Eker, "Hybrid Control of a Double Tank System", *IEEE Conference on Control Application*, Hartford, Connecticut, 1997.

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

[17] A. Rantzer and M. Johansson. Piecewise Linear Quadratic Optimal Control. *IEEE Transactions on Automatic Control*, Vol. 54, pp. 629-637, 2000.

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

[19] 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.

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

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

[22] G. Walsh, H. Ye, and L. Bushnell. Stability Analysis of Networked Control Systems. *American Control Conference*, pp. 2876–2880, 1999.

[23] 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.

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

[25] X. Xu and P. Antsaklis. An Approach to Switched Systems Optimal Control Based on Parameterization of the Switching Instants. In *Proceedings of the IFAC World Congress*, Barcelona, Spain, 2002.

[26] 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.

Table 1. Numerical results: (*) and (**) correspond to the values before Step 1 is entered. Initially the modal sequence is $\{f_1\}$; up to Iteration 4 it is $\{f_1, f_2, f_1\}$; and in Iterations 5-10 it is $\{f_1, f_2, f_1, f_2, f_0, f_2, f_1, f_2, f_1\}$.

Iteration (k)	$\bar{\tau}(k)$	$J(\bar{\tau}(k))$	$\|\|\bar{h}(\bar{\tau})\|\|$
(*)		1.21	-
0	(1.80, 1.80)	1.21	1.10
1	(1.45, 2.17)	0.50	0.73
2	(1.32, 2.30)	0.41	0.28
3	(1.28, 2.35)	0.40	0.11
4	(1.26, 2.40)	0.40	0.05
(**)	(0.01, 0.01, 1.26, 1.63, 1.63, 2.40, 3.58, 3.58)	0.40	1.10
5	(0, 0.01, 1.26, 1.63, 1.63, 2.40, 3.58, 3.59)	0.39	0.48
6	(0, 0.17, 1.29, 1.54, 1.71, 2.41, 3.49, 3.68)	0.32	0.73
7	(0, 0.19, 1.28, 1.56, 1.69, 2.42, 3.47, 3.69)	0.30	0.27
8	(0, 0.29, 1.28, 1.54, 1.71, 2.44, 3.41, 3.76)	0.27	0.32
9	(0, 0.29, 1.28, 1.53, 1.72, 2.44, 3.40, 3.76)	0.27	0.16
10	(0, 0.40, 1.30, 1.49, 1.76, 2.46, 3.33, 3.84)	0.25	0.08