Proceedings of the 17th International Symposium on Mathematical
Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006

Convergence of Gradient-Descent Algorithm for Mode-Scheduling Problems in Hybrid Systems †‡

H. Axelsson, Y. Wardi, and M. Egerstedt
School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.
{henrik,ywardi,magnus}@ece.gatech.edu

***Abstract*— In this paper we consider optimal mode-scheduling problems in hybrid dynamical systems where the design parameter has both a discrete and a continuous parameter. From an algorithmic standpoint, a number of techniques have been developed, but most of them include a systematic approach only to the continuous variable, while treating the discrete variable in ad-hoc ways. The main contribution of this paper is a solution to this problem, i.e. a systematic approach to optimizing over both the continuous and the discrete parameters.**

I. Problem Definition

Let Φ be a finite set of continuously-differentiable functions from R^n into R^n, and consider the dynamical system

$$\dot{x} \;\in\; \{f(x) : f \in \Phi\}, \tag{1}$$

defined on a given time-interval $[0,T]$ ($T > 0$), with a given initial condition $x(0) := x_0 \in R^n$. Suppose, moreover, that only a finite number of switchings between the functions $f \in \Phi$ is allowed in the interval $[0,T]$. Thus, there is an integer $N > 0$, a finite sequence $\{f_{\sigma(i)}\}_{i=1}^{N+1} \in \Phi \times \ldots \times \Phi$ (the $N+1$ set-product of Φ), and a vector $s := (\tau_1, \ldots, \tau_N)^T \in R^N$ satisfying the inequalities

$$0 \leq \tau_1 \leq \ldots, \leq \tau_N \leq T, \tag{2}$$

such that,

$$\dot{x} \;=\; f_{\sigma(i)}(x) \tag{3}$$

for all $i \in [\tau_{i-1}, \tau_i)$, and for all $i = 1, \ldots, N+1$. We assume that there exists a constant $K > 0$ such that $||f(x)|| \leq K(||x||+1)$ for every $f \in \Phi$ and for all $x \in R^n$, in order to ensure that Eq. (3) has a unique, continuous, and piecewise continuously-differentiable solution for the given initial condition x_0. We point out that the case $\tau_{i-1} = \tau_i$, permitted by (2), implies that $f_{\sigma(i)}$ plays no role in the evolution of the state variable x, but we consider this case for reasons that will be made clear in the sequel. The functions $f \in \Phi$ are called *modal functions*, the sequence $\sigma := \{\sigma(1), \ldots, \sigma(N+1)\}$ is called the *modal sequence*, and the vector $s \in R^n$ satisfying (2) is called the *switching-times vector*. Furthermore, we call the pair $\xi := (\sigma, s)$ the *modal schedule*. We also define $\tau_0 := 0$ and $\tau_{N+1} := T$ for the sake of convenience of notation.

Let $L : R^n \times [0,T] \to R$ be a cost function defined on (x,t), and let the cost functional J be defined by

$$J \;=\; \int_0^T L(x,t)dt. \tag{4}$$

Observe that J is a function of the schedule $\xi = (\sigma, s)$, and we denote the problem of minimizing J with respect to this variable by P. It is this optimization problem that we consider in this paper.

P is a scheduling problem cast in the form of an optimal control problem. Its variable parameter, $\xi = (\sigma, s)$, consists of the discrete, sequencing parameter σ, and the continuous, timing parameter s. Like many scheduling problems, P is NP-hard by dint of its discrete parameter. On the other hand, for a fixed σ, the continuous-time parameter can be handled by a nonlinear-programming algorithm (see, e.g., [8], [9]). Recently, there emerged local-descent algorithms that optimize J with respect to the discrete parameter σ [4], [2], [1]. They forego the pursuit of a global minimum and search only for local minima, thus evading the exponential-complexity issue inherent in globally optimal scheduling problems by taking on the lesser challenge of computing local solutions. The term "local" has to be carefully defined since the parameter space for the optimization problem P does not have a natural topology. In fact, the main issues addressed in this paper are to define convergence of iterative algorithms in a suitable sense, and to develop algorithms that are provably convergent in that sense. To address these issues, we first define a local direction in the space of schedules.

II. Mode Insertions

Let us fix a schedule $\xi := (\sigma, s)$, where $\sigma = \{\sigma(1), \ldots, \sigma(N+1)\}$ and $s = (\tau_1, \ldots, \tau_N)^T$, and recall that $\tau_0 = 0$ and $\tau_{N+1} = T$ by definition. Fix $\tau \in (0,T)$ and a modal function $f \in \Phi$, henceforth denoted by $f = f_{\sigma(\star)}$. For a given $\lambda > 0$, consider the insertion of the modal function $f_{\sigma(\star)}$ in the interval $[\tau - \frac{\lambda}{2}, \tau + \frac{\lambda}{2}) \cap [0,T]$, and denote the corresponding value of J as a

† This research was supported in part by the National Science Foundation under Grant #0509064.

‡ The algorithm presented in this paper has been discussed in [4], where its convergence-proof is incomplete. The proof in this paper is based on different arguments and it is much simpler and considerably shorter.

function of $\lambda \geq 0$ by $\hat{J}(\lambda)$. Let $D_{f,\tau}(\xi)$ denote the one-sided derivative of $\hat{J}$ at $\lambda = 0$, namely, $D_{f,\tau}(\xi) = \frac{d\hat{J}(0)}{d\lambda^+}$. As shown in [5], $D_{f,\tau}(\xi)$ is well defined for all $\tau \in [0, T]$, and it is continuous there in τ except at the switching times τ_i. If $D_{f,\tau}(\xi) < 0$ then an insertion of f in a "small-enough" interval centered at τ would result in a reduction in J.

Such a mode insertion modifies the modal sequence σ and the dimension of the switching-time vector s. For example, consider the case where $(\tau - \frac{1}{2}\lambda, \tau + \frac{1}{2}\lambda) \subset (\tau_{i-1}, \tau_i)$ for some $i = 1, \ldots, N+1$. Then the above insertion changes the schedule $\xi = (\sigma, s)$ to a new schedule, $\hat{\xi} = (\hat{\sigma}, \hat{s})$, defined by $\hat{\sigma} = \{\sigma(1), \ldots, \sigma(i), \sigma(\star), \sigma(i), \ldots, \sigma(N+1)\}$, and $\hat{s} = (\tau_1, \ldots, \tau_{i-1}, \tau - \frac{1}{2}\lambda, \tau + \frac{1}{2}\lambda, \tau_i, \ldots, \tau_N)^T$. Note that $\hat{s} \in R^{N+2}$.

We will be primarily interested in the case where $\lambda \to 0$, and in this case will refer to $\hat{\xi}$ as the *schedule obtained by inserting $f = f_{\sigma(\star)}$ to the schedule ξ at the time τ*. Observe that $\hat{\xi}$ has two additional switching times at the time τ, and the modal function inserted between them, f, is active on an interval of length 0.

We point out that the derivative term $D_{f,\tau}(\xi)$ has the following costate-based expression (see [5]). Define the costate $p(t)$, backwards recursively, to be the continuous function on the interval $[0, T]$ such that:

- For all $t \in [\tau_N, T]$,

$$\dot{p} \;=\; -\Big(\frac{\partial f_{\sigma(N+1)}}{\partial x}(x)\Big)^T p \;-\; \Big(\frac{\partial L}{\partial x}(x,t)\Big)^T, \quad (5)$$

  with the boundary condition $p(T) = 0$.
- For every $i \in \{N, \ldots, 1\}$, and for all $t \in [\tau_{i-1}, \tau_i]$,

$$\dot{p} \;=\; -\Big(\frac{\partial f_{\sigma(i)}}{\partial x}(x)\Big)^T p \;-\; \Big(\frac{\partial L}{\partial x}(x,t)\Big)^T, \quad (6)$$

  with the boundary condition $p(\tau_i)$ as defined by $p(t)$ in the interval $t \in [\tau_i, \tau_{i+1}]$.

Now for every $\tau \in (\tau_{i-1}, \tau_i)$, the derivative term $D_{f,\tau}(\xi)$ has the following form,

$$D_{f,\tau}(\xi) \;=\; p(\tau)^T \big(f(x(\tau)) - f_{\sigma(i)}(x(\tau))\big); \quad (7)$$

and as for the case where $\tau = \tau_i$, analogous expressions for $D_{f,\tau}(\xi)$ have been developed in [5]. It is now evident that $D_{f,\tau}(\xi)$ is generally discontinuous in τ at the switching times $\tau = \tau_i$, by the presence of the subscript $\sigma(i)$ in the RHS of Eq. (7).

III. An Algorithmic Approach

Given a fixed modal sequence $\sigma = \{\sigma(1), \ldots, \sigma(N+1)\}$, we denote by P_σ the problem of minimizing J as a function of $s \in R^N$ subject to the constraints in Eq. (2). The algorithm described below alternates between solving P_σ to the extent of computing a Kuhn-Tucker point s, and inserting a new modal sequence to the schedule $\xi = (\sigma, s)$ in a way that reduces the value of J. Such a cost-reducing insertion is possible as long as $D_{f,\tau}(\xi) < 0$ for some $(f, \tau) \in \Phi \times [0, T]$, and it is reasonable to choose f and τ that render the term $D_{f,\tau}(\xi)$ as negative as possible. Accordingly, let us define the term $D(\xi)$ by

$$D(\xi) \;:=\; \min\{D_{f,\tau}(\xi) \mid f \in \Phi,\; \tau \in [0, T]\}, \quad (8)$$

and let $(g, t) \in \Phi \times [0, T]$ be an argmin of the RHS of (8). Then, if $D(\xi) < 0$, then it is sensible to insert g at time t in order to reduce J.[1] On the other hand, the case wher $D(\xi) = 0$ does not indicate that such insertions can reduce J. The case $D(\xi) > 0$ is impossible since, for $f = f_{\sigma(i)}$ and for all $\tau \in (\tau_{i-1}, \tau_i)$, $D_{f,\tau}(\xi) = 0$, and hence, by definition (8), $D(\xi) \leq 0$ always. All of this leads us to the following definition of a stationary schedule.

Definition 1. A schedule $\xi = (\sigma, s)$ is a stationary schedule for the problem P if the following two conditions are satisfied: (i) s is a Kuhn-Tucker point for P_σ, and (ii) $D(\xi) = 0$. ■

The following algorithm aims at computing stationary points for P.

Algorithm 1.
Data: A modal sequence σ_0.
Step 0: Set $j = 0$.
Step 1: Solve the problem P_{σ_j} to the extent of computing a switching-time vector s_j that is a Kuhn-Tucker point for P_{σ_j}. Denote the resulting schedule by $\xi_j := (\sigma_j, s_j)$.
Step 2: If $D(\xi_j) = 0$, stop and exit. Otherwise, compute $g \in \Phi$ and $t \in [0, T]$ such that $D_{g,t}(\xi_j) = D(\xi_j)$.
Step 3: Define $\hat{\xi}_{j+1} := (\sigma_{j+1}, \hat{s}_{j+1})$ to be the schedule obtained by inserting g to the schedule $\xi_j = (\sigma_j, s_j)$ at time t.
Step 4: Set $j = j + 1$, and goto Step 1. ■

The algorithm stops at Step 2 if $D(\xi_j) = 0$, namely the schedule ξ_j is stationary according to Definition 1. Now if it does not stop after a finite number of iterations, namely if it computes an infinite sequence of schedules, $\{\xi_j\}_{j=1}^\infty$, then we expect this sequence to converge, in a suitable sense, to a stationary schedule. To this end, we borrow the following concept of convergence of algorithms defined on an infinite-dimensional linear space, proposed in [6].

Let $F : X \to R$ be a continuous function defined on an infinite-dimensional space X, and consider the problem of minimizing F over a given subset of X. Let Σ be a suitable optimality condition, and let $X_0 \subset X$ be the set of points where Σ is satisfied. An objective of an algorithm is to compute $x \in X_0$. Let $\theta : X \to R^-$ be a function called an *optimality function*, that is (see [7]), $\theta(x) = 0$ if and only if $x \in X_0$, and generally, $|\theta(x)|$ is a measure of an extent to which x satisfies Σ. Now following [6], *an algorithm is said to converge to* Σ if, for every sequence of points $\{x_j\}_{j=1}^\infty$ that it computes,

$$\limsup_{j \to \infty} \theta(x_j) \;=\; 0. \quad (9)$$

Arguments for the suitability of this convergence concept, and conditions under which it generalizes the established notions of algorithms' convergence, can be found in [6].

[1] Insertion of multiple functions is also possible, but we restrict the discussion to a single insertion in order to keep the discussion simple.

Now the setting in this paper is a bit different, since the parameter set for our optimization problem is not an infinite-dimensional space. In fact, the schedule ξ_j can be viewed as an element in the set Σ_{N_j} (where N_j is a positive integer), defined as follows: $\Sigma_{N_j} := \{(\sigma, s)\}$, where $\sigma = \{\sigma(1), \ldots, \sigma(N_j + 1)\}$ is a modal sequence, and $s = (s_1, \ldots, s_{N_j})^T$ is a Kuhn-Tucker point for P_σ. We note that the relation $D(\xi) = 0$ is an optimality condition for P on Σ_{N_j}, and that $D(\xi)$ generally qualifies as an optimality function for that condition. Therefore, following [6], we consider Algorithm 1 to be convergent to stationary points as long as

$$\lim_{j\to\infty} D(\xi_j) \;=\; 0 \tag{10}$$

for every sequence $\{\xi_j\}_{j=1}^{\infty}$ of schedules that it computes.

One way to ensure convergence in this sense is to establish the *sufficient-descent property* of the algorithm, defined as follows.

Definition 2. Algorithm 1 has sufficient descent if, for every $\eta > 0$ there exists $\delta > 0$ such that, for every sequence $\{\xi_j\}_{j=1}^{\infty}$ of schedules that it computes, and for every $j = 1, 2, \ldots$, if $D(\xi_j) < -\eta$ then $J(\xi_{j+1}) - J(\xi_j) < -\delta$. ■

By Eqs. (3) and (4), it is clear that there exist $K > 0$ such that $|J(\xi)| \leq K$ for every schedule ξ, and therefore, it is obvious that the sufficient-descent property of Algorithm 1 guarantees its convergence to stationary points. The next question, then, is how to guarantee that Algorithm 1 indeed has sufficient descent. A close look at the algorithm reveals the following relationship between ξ_j and ξ_{j+1}: $\xi_j = (\sigma_j, s_j)$, where s_j is the Kuhn-Tucker point for P_{σ_j} computed in Step 1. Then, $\hat{\xi}_{j+1} = (\sigma_{j+1}, \hat{s}_{j+1})$ is obtained (in Step 3) by inserting the modal function g (computed in Step 2) to ξ_j at the time t (also computed in Step 2). By the nature of this insertion (on a zero-length interval), we have that $J(\hat{\xi}_{j+1}) = J(\xi_j)$. Finally, in the next iteration of the algorithm (Step 1), s_{j+1} is computed from $\hat{s}_{j+1}$ for the modal sequence σ_{j+1}. All of this implies that if the procedure for computing s_j from $\hat{s}_j$ (likewise, s_{j+1} from $\hat{s}_{j+1}$) is comprised of a gradient descent algorithm, then $J(\xi_{j+1}) \leq J(\xi_j)$, meaning that Algorithm 1 is a descent algorithm as well. However, this does not imply that the algorithm has the sufficient descent property.

Let us denote by the *inner algorithm* the algorithm for computing s_j in Step 1. In various numerical experiments, we used a variant of a gradient-descent algorithm with Armijo step sizes for the inner algorithm. Its descent direction at each iteration is the opposite of the projection of $\nabla J(s)$ onto the feasible set (defined by Eq. (2)), and the Armijo step size gives an approximate line minimization (see [3]). This inner algorithm has been shown in [7] to have the sufficient descent property. However, this does not guarantee the sufficient descent property for Algorithm 1, due to the fact that the inner algorithm at successive steps of Algorithm 1 operates on spaces of increasing dimensions. Moreover, we have not been able to find an alternative descent direction that, with the Armijo step size, would yield the desired sufficient descent. Instead, we have discovered a *descent curve* that yields the desired property. The starting point of this curve if the point $\hat{s}_j$, and it is used only in the first step of the inner algorithm.

Let us denote the curve by $\{c_\lambda\}_{\lambda\geq 0}$, where $c_0 = \hat{s}_j$. Using the Armijo step size along this curve in the first step of the inner algorithm, let us denote the resulting point by s_A. Then, the following sufficient descent is in force.

- For every $\eta > 0$ there exists $\delta > 0$ such that, for every schedule ξ_j computed by Algorithm 1; if $D(\xi_j) < -\eta$, then $J(s_A) - J(\hat{s}_j) \leq -\delta$.

With this property of the curve $\{c_\lambda\}$, the the inner algorithm has the following form.

Algorithm 2 (Inner algorithm).

Initial step: Compute s_A by using the Armijo step size along the curve from its starting point, $\hat{s}_j$.

Subsequent steps: Starting at s_A, use any convergent feasible descent algorithm that computes a Kuhn-Tucker point for P_{σ_j}. Such a Kuhn-Tucker point will be s_j. ■

This inner algorithm guarantees sufficient descent of Algorithm 1, and hence its convergence to stationary points. The main result, to be presented in the talk, concerns the sufficient descent of the Armijo step size along the curve. We point out that this, or a similar procedure may not yield sufficient descent if it starts at any other point in the course of the inner algorithm. It has that property only at the starting point, $\hat{s}_j$ despite the fact that the inner algorithm operates on spaces of increasing dimensions, and this is due to the special structure of the optimization problem and its underlying dynamical system.

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] M. Alamir and S.A. Attia. On Solving Optimal Control Problems for Switched Nonlinear Systems by Strong Variations Algorithms, *Proc. 6th IFAC Symposium on Nonlinear Control Systems*, Stuttgart, Germany, September 2004.

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

[4] H. Axelsson, Y. Wardi, M. Egerstedt, and E.I. Verriest, A Provable Convergent Algorithm for Transition-Time Optimization in Switched Systems. To appear in *Proc. Conference on Decision and Control*, Sevilla, Spain, 2005.

[5] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-Time Optimization for Switched Systems. *IEEE Transactions on Automatic Control*, to appear.

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

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

[8] 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.

[9] 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.