

## A NUMERICAL ALGORITHM FOR OPTIMAL CONTROL PROBLEMS WITH SWITCHING COSTS

DAVID E. STEWART<sup>1</sup>

(Received 15 August 1990; revised 15 January 1991 and 10 September 1991)

### Abstract

Optimal control problems with switching costs arise in a number of applications, and are particularly important when standard control theory gives “chattering controls”. A numerical method is given for finding optimal controls for linear problems (linear dynamics, linear plus switching cost). This is used to develop an algorithm for finding sub-optimal control functions for nonlinear problems with switching costs. Numerical results are presented for an implementation of this method.

### 1. Introduction

It is well known in optimal control theory that an apparently well-posed problem such as

$$\min g(x(T))$$

over  $x: [0, T] \rightarrow \mathbf{R}^n$  absolutely continuous and  $u: [0, T] \rightarrow \mathbf{R}^m$  measurable subject to

$$\begin{aligned}x'(t) &= f(x(t), u(t)) \quad \text{in } \mathbf{R}^n, \quad x(0) = x_0 \\u(t) &\in U \text{ compact } \subset \mathbf{R}^m\end{aligned}$$

may not have solutions in the ordinary sense, even if  $f$  and  $g$  are smooth. For example,

$$\min x_2(1)$$

---

<sup>1</sup> Mathematics Department, University of Queensland, Australia 4072.

© Copyright Australian Mathematical Society 1992, Serial-fee code 0334-2700/92

over  $x$  and  $u$  subject to

$$\begin{aligned}x_1' &= u(t); & x_1(0) &= 0 \\x_2' &= x_1^2; & x_2(0) &= 0 \\u(t) &\in \{-1, +1\}\end{aligned}$$

does not exist, although the infimum of the attainable values of  $x_2(1)$  is well-defined (and is, in fact, zero) under the above conditions.

To eliminate this awkwardness of the theory, a number of techniques for extending the class of admissible controls have been developed under various names such as *generalised controls* (by Gamkrelidze [4]), or as *chattering controls*. These techniques usually involve extending the class of controls to include probability measures which are the *weak* limits of rapid oscillating control functions.

While these theoretical constructs are quite adequate for dealing with questions of the existence of (generalised) optimal controls, the implementation of generalised optimal controls on a piece of physical hardware often leads to additional unanticipated costs due to the rapid changes in the control variable. This is particularly important if the control is a switching control (and  $U$  is effectively a finite set). If the *switching cost* is taken into account, the "optimal control" as obtained by the standard theory is no longer necessarily optimal. Physically, if the switching device is, say, a mechanical relay, then this switching cost appears as a reduction of the life of the device from contacts wearing out, and the cost of replacing the worn relay. In a social or economic system, rapid oscillations would incur a cost by way of confusion and social disorganisation. However, the cost of a *single* switch is often much smaller than the other costs over the time-scale of interest.

Firstly, we need to incorporate switching costs into a mathematical model of the problem. To do this the *variation* of a function is introduced:

$$\bigvee_0^T u = \sup_{\mathcal{P}} \sum_{i=0}^{n-1} \|u(t_{i+1}) - u(t_i)\|$$

where  $\mathcal{P}$  ranges over all partitions  $0 = t_0 \leq t_1 \leq \dots \leq t_{n-1} \leq t_n = T$  with  $n = 1, 2, 3, \dots$ . Any function  $u: [0, T] \rightarrow \mathbf{R}^m$  where  $\bigvee_0^T u < \infty$  is said to be of *bounded variation*; the set of functions on  $[0, T]$  of bounded variation is denoted  $BV[0, T]$ .

The following model problem is proposed: let  $D$  be a nonsingular  $m \times m$  matrix,  $f: \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}^n$  and  $g: \mathbf{R}^n \rightarrow \mathbf{R}$  be continuous,  $x_0 \in \mathbf{R}^n$  and  $U$  compact  $\subset \mathbf{R}^m$  be given. The problem, then, is to find  $x$  and  $u$  that

achieves

$$\min g(x(T)) + \int_0^T (Du) \quad (1.1a)$$

where  $x: [0, T] \rightarrow \mathbf{R}^n$  and  $u: [0, T] \rightarrow \mathbf{R}^m$  satisfy the following:  $x$  is absolutely continuous,  $u$  has bounded variation, and

$$\begin{aligned} x' &= f(x, u), & x(0) &= x_0 \\ u(t) &\in U = \{v_1, \dots, v_m\}. \end{aligned} \quad (1.1b)$$

To my knowledge, the first successful mathematical treatment of such a problem to appear in the literature is by Blatt [1] in 1976. This problem has been investigated by J. Matula [5] in relation to problems with monotone controls, and by Teo and Jennings [8]. In [5], Matula developed both the existence theory and a Pontryagin-type minimum principle, while in [8], Teo and Jennings give algorithms for computing optimal controls. However, Matula's minimum principle assumes the set  $f(x, U)$  to be convex for all  $x$ , and Teo and Jennings' algorithm assumes the set  $f(x, U)$  to be path connected for all  $x$ . On the other hand, the problems of particular interest here are problems in which  $U$  is a finite set.

The following section deals with the theoretical aspects of the general problem (1.1). Section 3 looks at the case where  $U$  is a finite set and  $f(x, u)$  is linear in  $(x, u)$  and  $g(x(T))$  is linear in  $x(T)$ . In this case a dynamic programming algorithm is developed which can find the optimal  $(x, u)$ , at least to within discretisation error. Section 4 looks at two algorithms based on that in Section 3 for dealing with the problem where  $g(x(T))$ , as a function of  $u(\cdot)$ , is convex. These algorithms give good sub-optimal results. Finally, Section 5 presents some numerical results to demonstrate the performance of the above algorithms.

## 2. General issues

Here we consider the following problem:

$$\min g(x(T)) + \int_0^T (Du) \quad (2.1a)$$

where  $x: [0, T] \rightarrow \mathbf{R}^n$  and  $u: [0, T] \rightarrow \mathbf{R}^m$  satisfy the following:  $x$  is absolutely continuous,  $u$  has bounded variation, and

$$\begin{aligned} x' &= f(x, u), & x(0) &= x_0 \\ u(t) &\in U. \end{aligned} \quad (2.1b)$$

We assume that  $f$  is Lipschitz,  $g$  is continuous and  $U$  is compact for the remainder of the paper. Note that the set

$$\mathcal{U} = \{ u \in \text{BV}[0, T] \mid u(t) \in U \text{ for all } t \}$$

is the set of admissible controls.

The question of existence of solutions to optimal controls with switching costs can be answered affirmatively under these conditions; see, for example, Matula [5, Theorem 1.1] and Noussair [6]. In [5], Matula also goes on to provide a Pontryagin-type principle in the case where  $f(x, U)$  is a convex set for all  $x$ . Teo and Jennings [8] complement the theoretical results of [5] with a numerical method based on variational techniques for the case where  $f(x, U)$  is convex.

For switching control though,  $f(x, U)$  is a finite set, and indeed there is no valid corresponding Pontryagin type principle. Variational approaches can give limited information about the nature of optimal controls. However, variational approaches give essentially no information about the sequence of control values for optimal (or even “good”) control functions. What variational arguments can help with is characterising the optimal *switching times*.

Consider, for example, a control function  $u \in \mathcal{U}$  with switching time  $\tau$  where  $u(\tau^-) = u_-$  and  $u(\tau^+) = u_+$ . Consider the perturbation  $u_{\delta\tau}$  where

$$u_{\delta\tau}(t) = \begin{cases} u(t) & \text{for } t \text{ not between } \tau \text{ and } \tau + \delta\tau, \\ u_+ & \text{for } t \text{ between } \tau \text{ and } \tau + \delta\tau. \end{cases}$$

(Note that  $\delta\tau$  may be negative.) Then for sufficiently small  $\delta\tau$ ,  $u_{\delta\tau}$  has the same control sequence as  $u$ , so that  $\int_0^T u_{\delta\tau} = \int_0^T u$ . This means that only the “smooth” part of the cost  $g(x(T))$  is perturbed. Let  $x_{\delta\tau}$  be the solution of (2.1b) with  $u$  replaced by  $u_{\delta\tau}$ . By standard variational techniques we can show that

$$\begin{aligned} g(x_{\delta\tau}(T)) - g(x(T)) &= \int_0^T \lambda(t)^\top (f(x(t), u_{\delta\tau}(t)) - f(x(t), u(t))) dt + O(\delta\tau)^2 \\ &= \lambda(\tau)^\top (f(x(\tau), u_+) - f(x(\tau), u_-))\delta\tau + O(\delta\tau)^2. \end{aligned}$$

Thus any optimal switching time  $\tau$  must satisfy

$$\lambda(\tau)^\top f(x(\tau), u_-) = \lambda(\tau)^\top f(x(\tau), u_+).$$

While the problem (1.1) can be solved by looking at every plausible sequence of control values, the number of possible sequences suffers a combinatorial explosion as the switching cost becomes small. For example, there are  $|U|(|U| - 1)^K$  possible sequences with  $K$  switches. An alternative idea is to use an initial guess for the control functions which has a large number of switches and all possible control values represented many times, and

then to optimise with respect to the switching times, dropping switches when switching times coalesce. However, as noted above, these variational techniques when applied to switching control do not use any information about the switching costs except the fact that they are positive.

On the other hand, there has been some recent success in using continuous-time dynamic programming combined with the theory of viscosity solutions [2] for control with switching cost [3]. To numerically solve an optimal control problem by these techniques requires finding the viscosity solutions to  $|U|$  coupled 1st-order PDEs in  $n + 1$  independent variables. This approach suffers from Bellman’s “curse of dimensionality”: any numerical method would have to store  $\approx |U|N^n$  numbers where  $N$  is the number of grid points on each axis.

Here we are looking for practical methods, and so we are willing to sacrifice global optimality, at least for the general nonlinear problem. However, some sort of “local” optimality should be guaranteed, along with “good quality” sub-optimal results.

### 3. Algorithm for the linear case

We now consider the problem (1.1) in the case where the smooth part of the cost is linear; that is,  $g(x(T))$  is a linear function of  $u \in BV[0, T]$ . This is the case if, for example, the differential equations are linear in  $(x, u)$  ( $f(x, u) = Ax + Bu$ ) and  $g(x(T)) = c^T x(T)$  for some  $c \in \mathbf{R}^n$ . In this case an algorithm can be formulated which gives the optimal discretised control from a discrete family of control functions.

Firstly, we show that (1.1) is equivalent to

$$\min_u \int_0^T p(t)^T u(t) dt + \bigvee_0^T(Du) \tag{3.1}$$

where the optimisation is over  $u \in \mathcal{U}$ , for some  $p(\cdot)$ . This means that the dynamic programming algorithm referred to in the previous section can be used with  $n = 0$ ; that is, there is no need to have a state vector for dealing with linear problems.

Also, the function  $p(\cdot)$  can be obtained from the “co-state” function  $\lambda(\cdot)$  defined by

$$(\lambda^T)' = -\lambda^T \nabla_x f(x, u), \quad \lambda(T) = \nabla g(x(T)) \tag{3.2}$$

for a given pair  $(x(\cdot), u(\cdot))$  satisfying the differential equation

$$x' = f(x, u), \quad x(0) = x_0. \tag{3.3}$$

These two equations are of course the differential equations that arise in Pontryagin's maximum principle. In the argument that follows it we understand  $x(\cdot)$  to be implicitly defined by  $u(\cdot)$  through (3.3).

Note that we can assume without loss of generality that  $f(x, u)$  is linear in  $u$  if  $U$  is a finite set  $\{v_1, \dots, v_m\}$ . To do this, put  $f^j(x) = f(x, v_j)$  and  $F(x) = [f^1(x), \dots, f^m(x)]$ . Clearly  $f(x, v_j) = F(x)e_j$  where  $e_j$  is the  $j$ th standard unit vector in  $\mathbf{R}^m$ , so we can replace  $f(x, u(t))$  with  $u(t) \in U$  by  $F(x)\tilde{u}(t)$  where  $\tilde{u}(t) \in \tilde{U} = \{e_1, \dots, e_m\}$ . The  $D$  matrix would then be replaced by  $\tilde{D} = D[v_1, \dots, v_m]$ . This assumption will also be used in the following section.

We define the smooth part of the cost functional to be

$$S(u) = g(x(T))$$

where  $x(T)$  is defined through (3.3).

Note that  $S$  is, in fact, defined over all of  $BV[0, T]$  (indeed, over all of  $L^1(0, T)$ ), even though our interest is in a disconnected set of admissible controls. Thus we can take derivatives of  $S$ . The gradient of  $S(u)$ , denoted  $\nabla S(u)$ , is the functional defined by

$$\nabla S(u)(\delta u) = \int_0^T \lambda(t)^\top \nabla_u f(x(t), u(t)) \delta u(t) dt. \tag{3.4}$$

Assuming  $f(x, u)$  linear in  $u$ , then the linearisation is valid if  $\delta u$  is small in  $L^1$ :

$$S(u + \delta u) - S(u) = \nabla S(u)(\delta u) + o(\|\delta u\|_1).$$

In the case considered in this section, the  $S(\cdot)$  functional is linear in  $u$ , so  $\nabla S(u)$  and  $\lambda(t)^\top \nabla_u f(x(t), u(t))$  are independent of  $u$ . For any such  $u$  we set

$$p(t)^\top = \lambda(t)^\top \nabla_u f(x(t), u(t)) \quad \text{for all } t \in [0, T]. \tag{3.5}$$

In the case where  $f(x, u) = Ax + Bu$  and  $g(x(T)) = c^\top x(T)$  explicit formulae can be developed for  $p(t)$  though it is not necessary here. Note also that the above formula for  $\nabla S(u)$  holds even if the linearity of  $S(\cdot)$  fails.

The dynamic programming approach can be motivated by noting that for any  $\tau \in (0, T)$ , and  $u \in BV[0, T]$  normalised as noted in Section 2, the

cost functional can be split into

$$\begin{aligned} \int_0^T p(t)^\top u(t) dt + \bigvee_0^T(Du) &= \int_0^\tau p(t)^\top u(t) dt + \bigvee_0^{\tau^-}(Du) \\ &+ \int_\tau^T p(t)^\top u(t) dt + \bigvee_\tau^T(Du) \\ &+ \|D(u(\tau^-) - u(\tau))\|, \end{aligned}$$

where  $\bigvee_0^{\tau^-}(Du) = \sup_{t < \tau} \bigvee_0^t(Du)$ . The dynamic programming approach works by splitting the interval  $[0, T)$  into pieces  $[0, \tau)$  and  $[\tau, T)$  with  $|T - \tau| \ll \tau$  and finding optimal costs  $C_i(\tau)$  on  $[0, \tau)$  where  $u(\tau^-) = v_i$  for  $i = 1, \dots, m$ . Assuming that the optimal control is constant on  $[\tau, T)$ , we compute for each  $j$ ,

$$\begin{aligned} C_j(T) &= \min_u \int_0^T p(t)^\top u(t) dt + \bigvee_0^T(Du) \\ &= \min_i C_i(\tau) + \|D(v_i - v_j)\| + \int_\tau^T p(t)^\top v_j dt. \end{aligned}$$

Note that  $C_i(\tau)$  can be computed in an analogous fashion to  $C_j(T)$ .

Instead of assuming that there are no switches of  $u$  on  $[\tau, T)$ , we can consider the problem of finding the minimum of (3.1) over all *piecewise constant* controls  $u$ . Let  $0 = t_0 < t_1 < \dots < t_{N-1} < t_N = T$  be a partition of  $[0, T]$  and set  $u(t) = u_k$  on  $[t_k, t_{k+1})$ , and  $u(T) = u_N$ . Let

$$p_k = \int_{t_k}^{t_{k+1}} p(t) dt \quad \text{and} \quad d_{ij} = \|D(v_i - v_j)\|. \tag{3.6}$$

Then we have the following algorithm:

**ALGORITHM 1.** Given  $\mathbf{p} = [p_k \mid k = 0, \dots, N-1]$  and  $d_{ij}$ ,  $i, j = 1, \dots, m$ , find  $\mathbf{u} = [u_k \mid k = 0, \dots, N]$  that minimises

$$\sum_{k=0}^{N-1} (p_k^\top u_k + \|D(u_{k+1} - u_k)\|).$$

**Variables:**  $C_{ik}$ , minimum cost with  $u_k = v_i$  and  $N$  replaced by  $k$ .  
 $hist_{ik}$ , used to construct optimal control.

**Step 1:** Set  $C_{i0} \leftarrow 0$  for  $i = 1, \dots, m$ .

**Step 2:** For  $k = 0, \dots, N - 1$  do

For  $j = 1, \dots, m$  do

$hist_{jk} = \arg \min_i C_{ik} + d_{ij}$

$C_{j,k+1} = \min_i C_{ik} + d_{ij} + p_k^\top v_j$

**Step 3:**  $i^* \leftarrow \arg \min_i C_{iN}$

$u_N \leftarrow v_{i^*}$

The optimal cost is  $C_{i^*N}$ .

**Step 4:** For  $k = N - 1, \dots, 0$  do

$i^* \leftarrow hist_{i^*k}$

$u_k \leftarrow v_{i^*}$ .

A formal proof that this algorithm finds the minimising control amongst the class of discretised controls will not be given. However, below a convergence result is given as  $N \rightarrow \infty$ .

**THEOREM 3.1.** *Let  $u^{(N)}$  be the computed optimal control given the value of  $N$ , and let  $S^*$  be the optimal value of the cost functional. Then  $\lim_{N \rightarrow \infty} S(u^{(N)}) + \int_0^T (Du^{(N)}) = S^*$ , and every convergent subsequence of  $(u^{(N)})$  (of which there must be at least one) converges to a minimising  $u^* \in \text{BV}[0, T]$ .*

**PROOF.** By Matula [5] there must be a minimising  $u^*: [0, T] \rightarrow U$ . As  $U$  is finite, and  $u^* \in \text{BV}[0, T]$ ,  $u^*$  has at most finitely many switching points. We can assume without loss of generality that  $u^*$  is normalised to be right-continuous. Let the switching times be  $0 < t_1 < \dots < t_k < T$ .

Define  $\tilde{u}^{(N)}: [0, T] \rightarrow U$  by  $u^{(N)}(t) = \tilde{u}_i$  whenever  $t \in [iT/N, (i+1)T/N)$ , and  $\tilde{u}_i$  is the value of  $u^*(iT/N)$ . Then  $\tilde{u}^{(N)}(t) = u^*(t)$  except for  $t$  lying in no more than  $k$  intervals of the form  $[iT/N, (i+1)T/N)$ . Furthermore,  $\int_0^T \tilde{u}^{(N)} \leq \int_0^T u^*$ . Thus

$$\limsup_{N \rightarrow \infty} S(u^{(N)}) + \int_0^T (Du^{(N)}) \leq \limsup_{N \rightarrow \infty} S(\tilde{u}^{(N)}) + \int_0^T (D\tilde{u}^{(N)}) \leq S(u^*) + \int_0^T (Du^*).$$

As  $S(u^*)$  is the minimum value of  $S$ , it follows that

$$\limsup_{N \rightarrow \infty} S(u^{(N)}) + \int_0^T (Du^{(N)}) = S(u^*) + \int_0^T (Du^*).$$

Finally, as  $U$  is compact, and the variation of the computed  $u^{(N)}$  is bounded, there must be subsequence of the  $u^{(N)}$  which converges pointwise to some  $\hat{u} \in \text{BV}[0, T]$ , by Matula [4, Lemma 1.3]. From the above results,  $S(\hat{u}) + \bigvee_0^T(D\hat{u}) = S(u^*) + \bigvee_0^T(Du^*) = S^*$ , for any limit  $\hat{u}$  of a convergent subsequence.

#### 4. Minimising nonlinear functionals

Regarding practical applications, the main use of the algorithm for the linear problem is as a stepping-stone to develop an algorithm for a more general nonlinear problem. It seems that finding the global minimiser is a combinatorially explosive problem, so here a heuristic strategy is proposed to find good sub-optimal solutions for the nonlinear problem.

Another use of the linear dynamic programming algorithm is that when the functional is convex, a lower bound on the achievable values of the functional can be obtained.

In order to develop an algorithm for (1.1), we need to incorporate some restriction parameter into the problem. This also has to be incorporated into the dynamic programming algorithm.

Given a control sequence  $u_0 = [u_{0,0}, u_{0,1}, \dots, u_{0,N}]$ , and a nonnegative integer  $r$ , we restrict attention to control sequences  $u = [u_0, u_1, \dots, u_N]$  where  $u_{0,i} \neq u_i$  for no more than  $r$  values of  $i$ . This  $r$  can be used to restrict the changes that are allowed on each iteration of, say, a trust region like method. Alternatively,  $r$  can be used as a parameter for performing a one-dimensional search; this is analogous to line-search methods in conventional nonlinear programming.

A dynamic programming algorithm which incorporates this restriction will now be given, and then an algorithm for finding a sub-optimal solution of (1.1) will be given.

**ALGORITHM 2.** *Set up hist and C data structures to compute the optimum value and control for*

$$\sum_{k=0}^{N-1} (p_k^\top u_k + \|D(u_{k+1} - u_k)\|)$$

*subject to  $u_k \in U$  and  $|\{k \mid u_k \neq u_{0,k}\}| = r$  for  $0 \leq r \leq r_{max}$ .*

**Given :**  $\mathbf{p} = [p_0, \dots, p_{N-1}]$ ,  $d_{ij}$  and  $r_{max}$ .

**Output:**  $C_{irk}$ , minimum cost with  $u_k = v_i$  and  $N$  replaced by  $k$ ,  
and  $|\{l \mid u_l \neq u_{0,l}, 0 \leq l \leq k\}| = r$ .

$hist_{irk}$ , history array.

**Step 1:** \*\*\* Initialisation \*\*\*

Set  $C_{i00} \leftarrow 0$  for  $i = 1, \dots, m$

Set  $C_{ir0} \leftarrow +\infty$  for  $i = 1, \dots, m, r = 1, \dots, r_{max}$

**Step 2:** For  $k = 0, \dots, N - 1$  do steps 3-4

**Step 3:**  $minim \leftarrow C_{q0k}, q_{min} \leftarrow q$  where  $u_{0,k} = v_q$ .

**Step 4:** For  $r = 1, \dots, r_{max}$  do

For  $i = 1, \dots, m$  do

$minim \leftarrow \min_{q, v_q \neq u_{0,k}} C_{q, r-1, k-1} + d_{iq} + p_k^T v_q$   
and set  $q_{min}$  to achieve this minimum.

$r_{min} \leftarrow r - 1$ .

Set  $q$  so that  $u_{0,k} = v_q$ .

If  $C_{q, r, k-1} + d_{iq} + p_k^T v_q < minim$  then

$minim \leftarrow C_{q, r, k-1} + d_{iq} + p_k^T v_q, q_{min} \leftarrow q$  and

$r_{min} \leftarrow r$

$C_{irk} \leftarrow minim$ , and  $hist_{irk} \leftarrow (q_{min}, r_{min})$ .

The optimal control can then be found by “unravelling” the *hist* array, much as in the first algorithm. This can be done in the following algorithm:

**ALGORITHM 3.** Given the *hist* data structure as computed by Algorithm 2 and  $0 \leq r^* \leq r_{max}$ , construct the control sequence  $\mathbf{u}$  that minimises

$$\sum_{k=0}^{N-1} (p_k^T u_k + \|D(u_{k+1} - u_k)\|)$$

subject to  $u_k \in U$  for  $k = 0, \dots, N$  and  $|\{l \mid u_l \neq u_{0,l}, 0 \leq l \leq k\}| = r^*$ .

**Given :**  $C$  and *hist* arrays, and  $r^*$

**Output:**  $\mathbf{u} = [u_0, \dots, u_N]$

**Step 1:**  $i^* \leftarrow \arg \min_i C_{irN}, u_N \leftarrow v_{i^*}$ .

**Step 2:** For  $k = N - 1, \dots, 0$  do

$(i^*, r^*) \leftarrow hist_{irk}$

$u_k \leftarrow v_{i^*}$

Keeping the  $C$  and *hist* data structures provides considerable flexibility in developing algorithms for the full nonlinear problem, as we will often wish to obtain optimal controls for the linearised problem for several different values of  $r^*$ . This flexibility is used by the following algorithm for the full nonlinear problem:

**ALGORITHM 4.** *Given an initial control sequence  $\mathbf{u}_0$ , find a sub-optimal solution  $\mathbf{u}$  for the problem (1.1)*

**Given :** functions  $f, g, \nabla f, \nabla g$ , initial value  $x_0$ , initial control sequence  $\mathbf{u}_0$ , and  $r_{max}$

**Step 1:** Do steps 2–6 while  $r_{min} > 0$

**Step 2:** Solve differential equation for  $x$  (1.1b) given  $u(t) = u_0(t)$ , where  $u_0$  is the control function defined by  $\mathbf{u}_0$ .

**Step 3:** Compute  $\mathbf{p} = [p_0, \dots, p_{N-1}]$  by (3.2), (3.5) and (3.6)

**Step 4:** Compute  $C$  and *hist* data structures by Algorithm 2.

**Step 5:**  $minim \leftarrow 0, r_{min} \leftarrow 0$

For  $r = r_{max}, \dots, 1$  do

    Compute  $\mathbf{u}$  by Algorithm 3.

    Solve differential equation for  $x$  (1.1b) where  $u$  is the control function defined by  $\mathbf{u}$ .

    If  $g(x(T)) < minim$  then

$minim \leftarrow g(x(T)), r_{min} \leftarrow r$

**Step 6:** Compute  $\mathbf{u}$  by Algorithm 3 for  $r = r_{min}$ .

For discussing termination criteria, we call the value of  $g(x(T))$  at the start of any given loop in the above algorithm, “ $g(\text{new } x(T))$ ”, and the value at the start of the previous loop “ $g(\text{old } x(T))$ ”. Instead of using  $r_{min} = 0$  as a termination criterion, we could also use “ $g(\text{new } x(T)) \geq g(\text{old } x(T)) - \varepsilon$ ” where  $\varepsilon \geq 0$  is a suitable tolerance; then if  $r_{min} = 0$  we get no change in the control function, and so  $g(\text{new } x(T)) = g(\text{old } x(T))$  and we would also get termination for this criterion. This algorithm, with the “ $g(\text{new } x(T)) \geq g(\text{old } x(T)) - \varepsilon$ ” termination criterion has been implemented, and some numerical results obtained. These numerical results are given in the next section.

While convergence to the global minimum cannot be proved for this algorithm as for the linear case (even for convex functionals of  $u$ ), a local convergence result can be proven.

**THEOREM 4.1.** *Let  $u^{(N)} \in \text{BV}[0, T]$  be the result of Algorithm 4 and  $\hat{u}$  is a limit point of the  $u^{(N)}$  with respect to the topology of pointwise convergence as  $N \rightarrow \infty$ . Then  $\hat{u}$  is an extremum with respect to variations in the switching times.*

**PROOF.** Note that  $\hat{u}$  is the limit of  $u^{(N)}$  as  $N \rightarrow \infty$  for  $n \in \mathcal{N}$ , where  $\mathcal{N}$  is an infinite subset of the positive integers. Restricting our attention to this subsequence, the corresponding solutions of the differential equations  $x^{(N)}$

and the adjoint differential equations  $\lambda^{(N)}$  converge uniformly to solutions  $\hat{x}$  and  $\hat{\lambda}$  of the respective differential equations for  $\hat{u}$ .

Let  $\mathcal{F}^{(N)}$  be the set of switching times of  $u^{(N)}$ , and  $\mathcal{F}$  is the set of switching times of  $\hat{u}$ . Note that if  $\tau \in \mathcal{F}$ , then for every  $\varepsilon > 0$  and  $N \in \mathcal{N}$  sufficiently large, there is a  $\tau^{(N)} \in \mathcal{F}^{(N)}$  such that  $|\tau - \tau^{(N)}| < \varepsilon$ . To prove this, note that for  $\varepsilon \leq \min\{|s-t| \mid s, t \in \mathcal{F}\}$ , the limits of  $u^{(N)}(\tau - \varepsilon/2)$  and  $u^{(N)}(\tau + \varepsilon/2)$  as  $N \rightarrow \infty$ ,  $N \in \mathcal{N}$  are  $\hat{u}(\tau^-)$  and  $\hat{u}(\tau^+)$  respectively. Thus there must be a switching time of  $u^{(N)}$  in  $[\tau - \varepsilon/2, \tau + \varepsilon/2]$  for sufficiently large  $N \in \mathcal{N}$  as desired. Note, however, that there may be more than one switching time of  $u^{(N)}$  in this interval.

Suppose that  $\hat{u}$  is not an extremum with respect to variations in the switching times. Then there is a  $\tau \in \mathcal{F}$  such that  $\hat{\lambda}(\tau)^T f(\hat{x}(\tau), \hat{u}(\tau^-)) - \hat{\lambda}(\tau)^T f(\hat{x}(\tau), \hat{u}(\tau^+)) \neq 0$ . Put  $u^- = \hat{u}(\tau^-)$  and  $u^+ = \hat{u}(\tau^+)$ . Then by the uniform convergence of  $x^{(N)}$  and the  $\lambda^{(N)}$  for  $N \in \mathcal{N}$ ,  $\lambda^{(N)}(t)^T f(x^{(N)}(t), u^-) - \lambda^{(N)}(t)^T f(x^{(N)}(t), u^+)$  is bounded away from zero for sufficiently large  $N \in \mathcal{N}$ , and  $t$  sufficiently close to  $\tau$ . However, this implies that there a discrete control  $\tilde{u}^{(N)}$  that gives a smaller cost than  $u^{(N)}$ , which differs from  $u^{(N)}$  in just one component, for sufficiently large  $N \in \mathcal{N}$ . This is impossible if  $u^{(N)}$  is the control generated by Algorithm 4. Thus, we conclude that  $\hat{u}$  must be extremal with respect to variations of the switching times.

### 5. Numerical results

Algorithm 4 has been implemented with a “ $g(\text{new } x(T)) \geq g(\text{old } x(T)) - \varepsilon$ ” termination criterion, and some numerical results obtained. Firstly, numerical results were obtained for the very simple problem

$$\min \int_0^T x(t)^2 dt + \varepsilon \bigvee_0^T u$$

$$x' = u, \quad x(0) = 1, \quad u \in \{-1, +1\}.$$

The parameters used were  $T = 16$ ,  $\varepsilon = 0.025$  (i.e.  $d_{12} = d_{21} = 0.05$ ),  $r_{max} = 20$  and  $N$  ranging from 50 to 3200. The stopping tolerance used was  $10^{-5}$ . The initial “guess” for the control function was  $u(t) \equiv +1$ . The minimum value can be computed analytically in this case to be  $\approx 1.9889$ .

TABLE 1

$N$	Total cost	# switches	# iterations
50	8.75188	15	11
100	4.04887	19	18
200	4.37684	15	15
300	2.84085	17	13
400	2.42132	19	19
600	2.08735	25	28
800	2.39125	37	29
1200	2.10124	25	48
1600	2.02610	23	61
2400	2.05308	21	80
3200	2.18738	19	103

The trajectory  $x(t)$  is shown below for  $N = 2400$ .

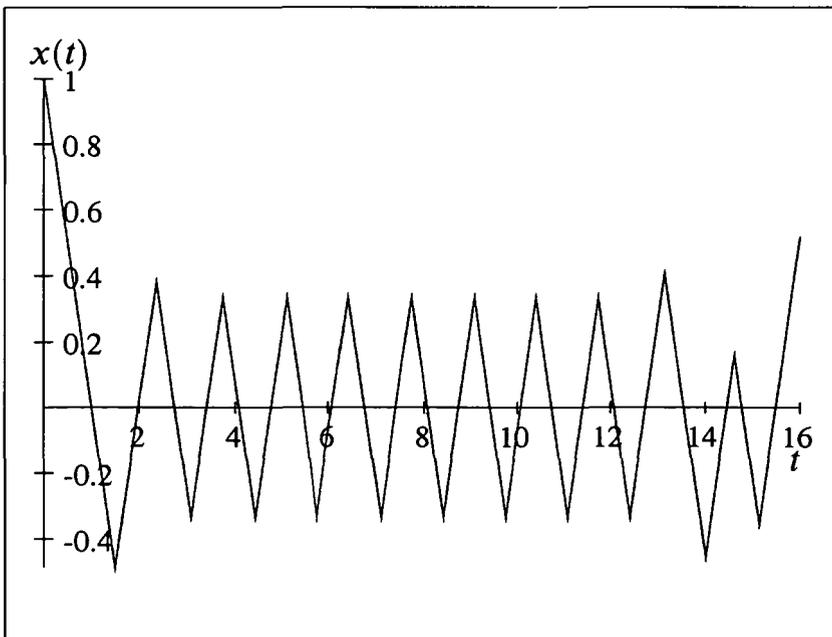


FIGURE 1. Trajectory  $x(t)$

Another test problem that was used arises in relation to switched amplifier design. The output stage of such an amplifier is modelled as an  $RLC$  circuit as shown below, where the  $v_i$  voltage is the control, and the  $v_o$  voltage is the output, which we want to track a sine wave. Using the mean square error in  $v_o$  as a measure of distortion, we wish to minimise the total distortion plus a measure of cost due to switching. During a switch, heat must be dissipated in the transistor that does the switching, and we seek to add a penalty term in the total cost proportional to the total heat dissipated in the transistor(s).

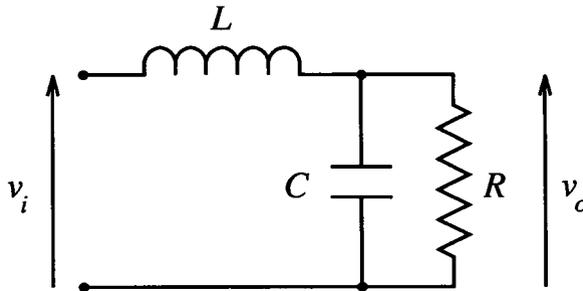


FIGURE 2. Switched amplifier output circuit

The differential equation for this circuit is

$$LC \frac{d^2 v_o}{dt^2} + \frac{L}{R} \frac{dv_o}{dt} + v_o = v_i(t).$$

The following examples were considered with  $L = C = 1.0$  and  $R = 2.0$ ; the unforced ( $v_i = 0$ ) system has a centre, rather than a node. The eigenvalues are  $(-1 \pm \sqrt{15} i)/4$ . The first problem looked at in relation to this system was the problem of maintaining  $v_o \approx 0$  where the control  $v_i$  could only be  $\pm 1$ . The cost criterion was

$$\min \int_0^T v_o(t)^2 dt + \epsilon \int_0^T v_i.$$

Below is a phase portrait of the sub-optimal trajectory obtained by the algorithm for  $T = 16$ ,  $\varepsilon = 0.025$  (switching cost = 0.05) and initial conditions  $v_o(0) = 1$ ,  $v'_o(0) = 1$ . The initial "guess" for the control function was  $v_i(t) \equiv +1$ . The total cost was 2.2896, of which 0.65 was the total switching cost. This was obtained in 186 outer iterations.

If the switching cost were zero, then there would be one switch near where the first switch above is, the trajectory would then go to the origin, where the trajectory would "chatter" about the origin.

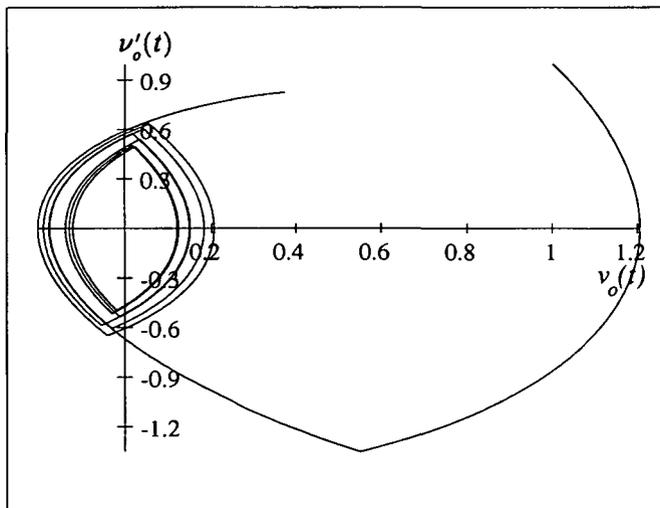


FIGURE 3

The second problem related to the switching amplifier example is in tracking a sine wave  $\sin(\omega t)$  where there are three admissible values of  $v_i$ : 0 and  $\pm 1$ . This is an important example in that it shows that the algorithm can give good sub-optimal controls where the number of possible switching sequences grows exponentially quickly in the number of switches.

In this problem the switching costs between 0 and  $\pm 1$  are both 0.03, and the switching cost between  $+1$  and  $-1$  is 0.05. Also,  $T = 40$ ,  $\omega = 0.7$ ,  $v_o(0) = 0$ ,  $v'_o(0) = 1$ , and  $N = 1600$ ,  $r_{max} = 20$ . The initial "guess" for the control function was  $v_i(t) \equiv 0$ . The control function obtained by the algorithm gave a total cost of 0.810619, of which the switching cost was 0.54. This was obtained in 59 outer iterations. The control *sequence* obtained was 0,  $+1$ , 0,  $-1$ , 0,  $+1$ , ..., 0,  $+1$ , 0 with a total of 18 switches. (Note that there are  $3 \times 2^{18} \approx 7.8 \times 10^5$  control sequences with this many switches.)

Below is a graph of  $v_o(t)$  against  $t$ ; also plotted is the sine wave that is being tracked. (The solid line is the trajectory of  $v_o(t)$ .)

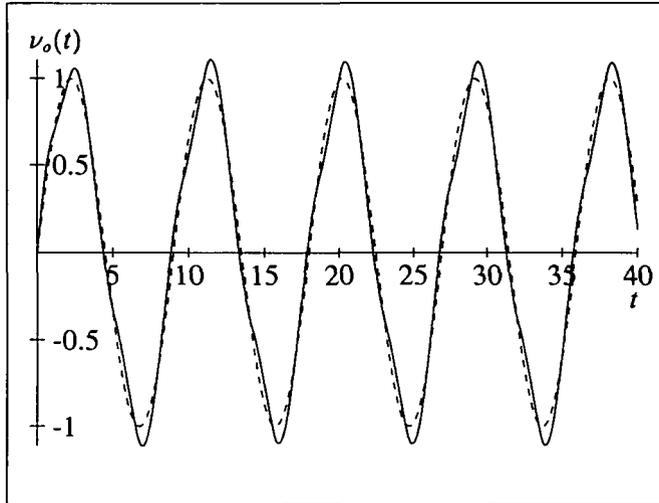


FIGURE 4. Plot of  $v_o(t)$

Finally, below is a phase-plane portrait of the trajectory of the switched controller obtained.

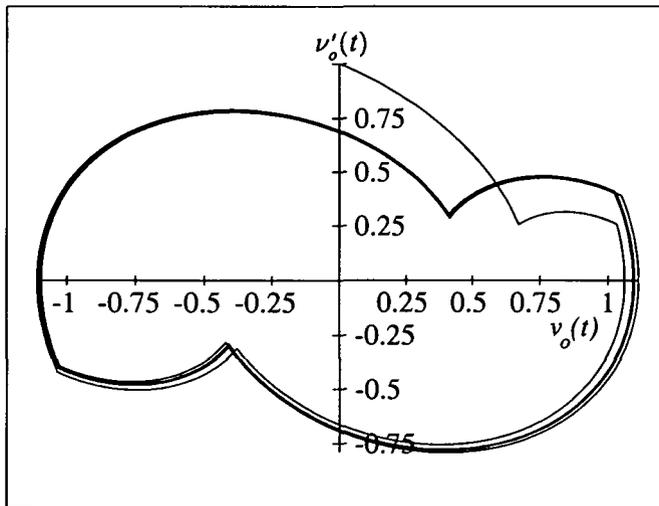


FIGURE 5. Phase plane portrait

## 6. Conclusions

An algorithm for finding optimal controls for linear optimal control problems with switching cost is given, and this is extended to find sub-optimal controls for nonlinear optimal control problems with switching cost. The results seem to be fairly good, although there are often “edge effects” near  $t = 0$  and  $t = T$ . Also the amount of computer time is larger than desired; the example problems took up to an hour of CPU time on the Mathematics Department’s Pyramid 9810.

Further work is being done to investigate how treating the switching times as variables (once the control *sequence* is fixed) can be incorporated into the algorithm for nonlinear problems. Such an approach would probably draw on the work of Wong, Clements and Teo [9].

## Acknowledgement

I would like to thank Dr. Gerry Ledwich of the University of Queensland Electrical Engineering Department for his interest and the application to switched amplifier design. I would also like to thank the referees for their comments.

## References

- [1] J.M. Blatt, “Optimal control with a cost of switching control,” *J. Austral. Math. Soc. Ser. B* **19** (1976) 316–332.
- [2] M.G. Crandall and P.L. Lions, “Viscosity solutions of Hamilton–Jacobi equations,” *Trans. Amer. Math. Soc.* **227** (1983) 1–42.
- [3] I.C. Dolcetta and L.C. Evans, “Optimal switching for ordinary differential equations,” *SIAM J. Control and Optim.* **22** (1984) 143–161.
- [4] R.V. Gamkrelidze, *Principles of Optimal Control Theory*, Orig. in Russian, (Plenum Press, London, New York 1975).
- [5] J. Matula, “On an extremum principle,” *J. Austral. Math. Soc. Ser. B* **28** (1987) 376–392.
- [6] E.S. Noussair, “On the existence of piecewise continuous optimal controls,” *J. Austral. Math. Soc. Ser. B* **20** (1977) 31–37.
- [7] K. Stromberg, *Introduction to Classical Real Analysis* (Wadsworth, Inc., Belmont, CA 1981).
- [8] K.L. Teo and L.S. Jennings, “Optimal control with a cost on changing control,” *J. Optim. Theory and Appl.* **68** 2 (1991).
- [9] K.H. Wong, D.J. Clements and K.L. Teo, “Optimal control computation for nonlinear time-lag systems,” *J. Optim. Theory and Appl.* **47** (1985) 91–107.