Abstract

This paper proposes an algorithmic technique for a class of optimal control problems where it is easy to compute a pointwise minimizer of the Hamiltonian associated with every applied control. The algorithm operates in the space of relaxed controls and projects the final result into the space of ordinary controls. It is based on the descent direction from a given relaxed control towards a pointwise minimizer of the Hamiltonian. This direction comprises a form of gradient projection and for some systems, is argued to have computational advantages over direct gradient directions. The algorithm is shown to be applicable to a class of hybrid optimal control problems. The theoretical results, concerning convergence of the algorithm, are corroborated by simulation examples on switched-mode hybrid systems as well as on a problem of balancing transmission- and motion energy in a mobile robotic system.

Keywords: Optimal control, relaxed controls, optimization algorithms, switched-mode systems.

1. Introduction

Consider dynamical systems described by the differential equation

\[ \dot{x} = f(x, u), \tag{1} \]

where \( x(t) \in \mathbb{R}^n \) is its state variable and \( u(t) \in U \subseteq \mathbb{R}^k \) is the input control. The set \( U \subseteq \mathbb{R}^k \) is assumed to be compact. Suppose that the initial time is \( t_0 = 0 \), and the initial state \( x_0 := x(0) \in \mathbb{R}^n \) and the final time \( t_f > 0 \) are given and fixed. A control \( \{u(t), \ t \in [0,t_f]\} \) is said to be ordinary if the function \( u : [0,t_f] \to \mathbb{R}^k \) is Lebesgue measurable, and we call a control admissible if it is ordinary and \( u(t) \in U \) for every \( t \in [0,t_f] \). Let \( L : \mathbb{R}^n \times U \to \mathbb{R} \) be an

\[ \text{Research supported in part by the NSF under Grant CNS-1239225.} \]
absolutely-integrable cost function, and let

\[
J := \int_{t_0}^{t_f} L(x(t), u(t)) dt
\]

be its related performance functional. The optimal control problem that we consider is to minimize \( J \) over the space of admissible controls.

The following assumption will be made throughout the paper:

**Assumption 1.1.** (1). The function \( f(x, u) \) is twice-continuously differentiable in \( x \in \mathbb{R}^n \) for every \( u \in U \); the functions \( f(x, u), \frac{\partial f}{\partial x}(x, u), \) and \( \frac{\partial^2 f}{\partial x^2}(x, u) \) are locally-Lipschitz continuous in \( (x, u) \in \mathbb{R}^n \times U \); and there exists \( K > 0 \) such that, for every \( x \in \mathbb{R}^n \) and for every \( u \in U \), \( ||f(x, u)|| \leq K(||x|| + 1) \).

(2). The function \( L(x, u) \) is continuously differentiable in \( x \in \mathbb{R}^n \) for every \( u \in U \); and the functions \( L(x, u) \) and \( \frac{\partial L}{\partial x}(x, u) \) are locally-Lipschitz continuous in \( (x, u) \in \mathbb{R}^n \times U \).

Observe that part (1) of the assumption guarantees the existence of a unique absolutely-continuous solution of Equation (1) in its integral form for every admissible control and \( x_0 \in \mathbb{R}^n \), while part (2) implies that the Lebesgue integral in Equation (2) is well defined.

This paper has been motivated by two kinds of problems: one concerns switched-mode hybrid systems, and the other concerns optimal balancing of the energy required for transmission and motion in mobile robotic networks. We propose an algorithm defined in the setting of relaxed controls, and analyze its convergence using Polak’s framework of optimality functions. The theoretical results are first derived in the abstract framework of Eqs. (1) and (2), and then applied to problems in the two aforementioned areas of interest.

A standard basic requirement of algorithms for nonlinear-programming (finite-dimensional optimization) problems, and especially gradient-descent methods, is that every accumulation point of an iterate-sequence they compute must satisfy an optimality condition for a local minimum, such as the Kuhn-Tucker condition (e.g., [1]). Thus, if a bounded sequence of iteration points is computed, it has at least one accumulation point which, therefore, must satisfy that condition. However, in infinite-dimensional optimization this requirement can be vacuous since bounded sequences need not have accumulation points. This issue is not only theoretical but also has practical implications. An infinite-dimensional problem may not have a solution or a local minimum on a closed and bounded set, and even if a local minimum exists, it is not guaranteed that it can be approximated by the solution points of the problem’s finite-dimensional discretizations at any level of precision. To get around these issues, E. Polak developed a comprehensive framework for design and analysis of algorithms for infinite-dimensional optimization that gives the user considerable discretion in choosing the discretization levels in an adaptive fashion [1]. A survey of the framework will be carried out in the next section, and we mention that recently it has been used in the context of switched-mode hybrid systems in Refs. [2, 3, 4, 5, 6, 7, 8, 9].
A relaxed control is a mapping \(\mu(t)\) from the time interval \([0, t_f]\) into the space of probability measures on the set \(U\). Relaxed controls provide a useful framework for optimal control problems for the following reasons. First, the space of relaxed controls is convex even though the input-constraint set \(U\) may not be convex. Second, the space of relaxed controls is compact in a suitable topology, namely the weak-star topology, and hence it contains solutions to optimal control problems lacking solutions in a functional space of admissible controls like \(L^1(U)\) (a more detailed survey of these points is provided in Section 2). However, a relaxed control is a more abstract object than an ordinary control, which can make it problematic for an algorithm to handle an iterative sequence of relaxed controls. This point will be discussed in the sequel.

This paper combines the frameworks of optimality functions and relaxed controls to define a new algorithm for the optimal control problem. Its main innovation is in the choice of the search direction from a given relaxed control, which is based on a pointwise minimizer of the Hamiltonian (defined below) at each time \(t \in [0, t_f]\). Its step size is determined by the Armijo procedure [12. 1]. To our knowledge it is the first such general algorithm which is defined entirely in the space of relaxed controls without projecting the results of each iteration into the space of ordinary controls. The following results will be established.

- The aforementioned search direction yields descent of the cost functional (2) even though it is not defined in explicit gradient terms. It is a form of gradient projection. For a class of problems (including autonomous switched-mode systems) its computation is straightforward and simpler than that of direct gradients.

- The algorithm is stable in the sense that it yields descent in the performance integral regardless of the initial guess. Furthermore, the simulation results presented in Section 4 indicate its rapid descent at the early stages of its runs. By this we do not claim a high rate of asymptotic convergence since it is a first-order algorithm, but rather that most of its descent is obtained in few early iterations requiring meagre computing (CPU) times.

- The combination of concepts and techniques from the settings of optimality functions and relaxed controls yields an analysis framework that is based on simple arguments. This point will become evident from the simplicity and brevity of the forthcoming proofs.

Whereas the area of numerical techniques for optimal control has had a long history (e.g., [1] and references therein), recently there has been a con-
siderable interest in optimal control problems defined on switched-mode hybrid systems. In this context the problem was formulated in [13, 14], and variants of the Maximum Principle were derived for it in [14, 15, 16, 17, 18, 19, 20, 21]. New and emerging algorithmic approaches include first- and second-order techniques [22, 23, 16, 8, 3, 7], zoning algorithms based on the geometric properties of the underlying systems [26, 27, 28, 19], projection-based algorithms [2, 4, 24, 25, 7, 5], methods based on dynamic programming and convex optimization [29], and algorithms based on needle variations [30, 8, 9, 6, 21].

Concerning the relaxed hybrid problem, Ref. [31] developed generalized-linear programming techniques and convex-programming algorithms, [18] derived optimality conditions (both necessary and sufficient) for a class of hybrid optimal control problems, and [32] applied to them the MATLAB fmincon nonlinear programming solver. Comprehensive recent surveys can be found in [33, 34].

The forthcoming algorithm will be presented and analyzed in the abstract problem formulation of Eqs. (1) and (2) and their extensions to the relaxed-control setting. However, for implementation, we restrict the class of problems in the following two ways: 1). The pointwise minimizer of the Hamiltonian can be computed or adequately estimated by a simple formula. 2). For every \( x \in \mathbb{R}^n \), the dynamic response function \( f(x,u) \) in Eq. (1) is affine in \( u \), and the cost function \( L(x,u) \) in Eq. (2) is convex in \( u \). Many problems of theoretical and practical interest in optimal control satisfy these restrictions. These include problems defined on autonomous switched-mode systems and other hybrid systems, which will be shown to admit efficient implementations of the algorithm. Other kinds of hybrid systems are not yet included, and these will be mentioned in the sequel as a subject of current research.

The rest of the paper is organized as follows. Section 2 presents brief surveys of the frameworks of relaxed controls and optimality functions. Section 3 describes the algorithm and derives related theoretical results, while Section 4 presents simulation results. Finally, Section 5 concludes the paper and points out directions for future research.

**Notation**: The term control refers to the function \( \{ u : [0,t_f] \rightarrow U \} \) and is denoted by the boldface symbol \( \mathbf{u} \) to distinguish it from a point in the set \( U \) which is denoted by the lower-case \( u \) or \( u(t) \). Similarly, boldface notation refers to a function of \( t \in [0,t_f] \) as in \( \mathbf{x} := \{ x : [0,t_f] \rightarrow \mathbb{R}^n \} \) for the state trajectory, \( \mathbf{p} := \{ p : [0,t_f] \rightarrow \mathbb{R}^n \} \) for the costate (adjoint) trajectory, \( \mathbf{\mu} \) for a relaxed control defined as a function from \( t \in [0,t_f] \) into the space of probability measures on \( U \), etc.

2. Review of Established Results

This section recounts the basic framework of relaxed controls and some fundamental notions of algorithms’ convergence in infinite-dimensional spaces.

2.1. Relaxed Control

Comprehensive presentations of the theory of relaxed controls and their role in optimal control can be found in [10, 11, 35, 36, 37]; also see [35] for a re-
cent survey. In the following paragraphs we summarize its main points that are relevant to our discussion. Let $M$ denote the space of Borel probability measures on the set $U$, and denote by $\mu$ a particular point in $M$. A relaxed control associated with the system (1) is a mapping $\mu : [0, t_f] \to M$ which is measurable in the following sense: For every continuous function $\phi : U \to R$, the function $\int_0^t \phi(u) d\mu(t)$ is Lebesgue measurable in $t$. We denote the space of relaxed controls by $M$, and in accordance with previous notation we denote a relaxed control $\{\mu(t) : t \in [0, t_f]\}$ by $\mu$.

Recall that an ordinary control $u$ is admissible if the function $u : [0, t_f] \to U$ is (Lebesgue) measurable. Note that the space of ordinary controls is embedded in the space of relaxed controls by associating with $u(t)$ the Dirac probability measure at $u(t) \forall t \in [0, t_f]$. In this case we say, with a slight abuse of notation, that $\mu$ is an ordinary control, and indicate this by the notation $\mu \sim u$. Furthermore, the space of ordinary controls is dense in the space of relaxed controls in the weak-star topology, namely in the following sense: For every relaxed control $\mu$ there exists a sequence $\{u_k\}_{k=1}^\infty$ of ordinary controls such that, for every function $\psi \in L^1([0, t_f]; C(U))$,

$$\lim_{k \to \infty} \int_0^{t_f} \psi(t, u_k(t)) dt = \int_0^{t_f} \int_U \psi(t, u) d\mu(t) dt.$$  

Furthermore, the space of relaxed controls is compact in the weak-star topology.

An extension of the system defined by Equations (1) and (2) to the setting of relaxed controls is provided by the state equation

$$\dot{x}(t) = \int_U \int U f(x(t), u) d\mu(t)$$

with the same boundary condition $x_0 = x(0)$ as for (1), and the related cost functional defined as

$$J(\mu) = \int_0^{t_f} \int_U L(x(t), u) d\mu(t) dt.$$  

The relaxed optimal control problem is to minimize $J(\mu)$ over $\mu \in M$.

There are two noteworthy special cases. First, if $\mu \sim u$, then Equations (3) and (4) are reduced to Equations (1) and (2), respectively. Second, in the case where $U := \{u_1, \ldots, u_m\}$ is a finite set, (3) and (4) have the following respective forms: $\dot{x}(t) = \sum_{i=1}^m \mu^i(t) f(x(t), u_i)$ and $J = \int_0^{t_f} \sum_{i=1}^m \mu^i(t) L(x(t), u_i) dt$, with $\mu^i(t) \geq 0$ and $\sum_{i=1}^m \mu^i(t) = 1 \forall t \in [0, t_f]$, and this corresponds to the case of autonomous switched-mode systems. The space of relaxed controls generally is convex even though the set $U$ need not be convex.

Essential parts of the theory of optimal control, including the Maximum Principle [37], apply to the relaxed-control problem (see also [39] [10] [11] [36] [37]).

---

2$L^1([0, t_f]; C(U))$ is the space of functions $\psi : [0, t_f] \times U \to R$ that are measurable and absolutely integrable in $t$ on $[0, t_f]$ for every $u \in U$, and continuous on $U$ for every $t \in [0, t_f]$.

---
Thus, defining the adjoint (costate) variable $p(t) \in \mathbb{R}^n$ by the equation
\[
\dot{p}(t) = -\int_U \left( \frac{\partial f}{\partial x} (x(t), u)^\top p(t) + \frac{\partial L}{\partial x} (x(t), u)^\top \right) d\mu(t)
\] (5)
with the boundary condition $p(t_f) = 0$, the Hamiltonian has the form
\[
H(x(t), \mu(t), p(t)) = \int_U \left( p(t)^\top f(x(t), u) + L(x(t), u) \right) d\mu(t),
\] (6)
and the Maximum Principle states that if $\mu \in M$ is a minimum for the relaxed optimal control problem then $\mu(t)$ minimizes the Hamiltonian at almost every time-point $t \in [0, t_f]$.

2.2. Infinite-dimensional Optimization

It is a common practice to characterize convergence of algorithms for infinite-dimensional optimization problems in terms of optimality functions \cite{1}. Consider the abstract optimization problem of minimizing a function $\phi : \Gamma \to \mathbb{R}$ where $\Gamma$ is a topological space, and consider an optimality condition (necessary or sufficient) associated with this optimization problem. An optimality function is a function $\theta : \Gamma \to \mathbb{R}^\nu$ having the property that $\theta(v) = 0$ if and only if $v$ satisfies the optimality condition. The optimality-function concept is useful if $|\theta(v)|$ is a meaningful heuristic measure of the extent to which $v \in \Gamma$ fails to satisfy the optimality condition. For example, if $\phi$ is a continuously Frechet-differentiable functional defined on a Hilbert space $H$, then an optimality condition is $d\phi/dv(v) = 0$ ($d\phi/dv$ meaning the Frechet derivative), and a meaningful associated optimality function is $\theta(v) := -\|d\phi/dv(v)\|$, where the indicated norm is in $H$.

Reference \cite{1} devised a framework for analysis of algorithms in this abstract setting, where convergence is defined as follows: If an algorithm computes a sequence $v_k, k = 1, 2, \ldots$, of points in $\Gamma$ then
\[
\lim_{k \to \infty} \theta(v_k) = 0.
\] (7)
In this abstract setting such a sequence $\{v_k\}_{k=1}^\infty$ need not have an accumulation point even if it is a bounded sequence in a metric space (unless it is isomorphic to a Euclidean space), and therefore a characterization of an algorithm’s convergence in terms of such accumulation points could be vacuous. The use of optimality functions via Equation (7) serves to resolve this conceptual issue. Note that it is a form of weak convergence.

\footnote{Note that the integrand of (6) is the usual Hamiltonian $H(x, u, p) := p^\top f(x, u) + L(x, u)$, while the term in the Left-hand Side of (6), namely $H(x, \mu, p)$, refers to the relaxed Hamiltonian. These two notations are distinguished by their second variable, $u$ vs. $\mu$, which suffices to render the usage of the functional term $H(x, \cdot, p)$ unambiguous in the sequel.}
Consider an algorithm that computes from a given \( v \in \Gamma \) the next iteration point, denoted by \( v_{\text{next}} \), and suppose that its repetitive application computes a sequence of iteration points \( \{v_k\}_{k=1}^\infty \subset \Gamma \), where \( v_{k+1} = v_{k,\text{next}} \). The algorithm is said to have the property of sufficient descent with respect to \( \theta(\cdot) \) if the following two conditions hold: (i) \( \phi(v_{\text{next}}) - \phi(v) \leq 0 \) for every \( v \in \Gamma \), and (ii) for every \( \eta > 0 \) there exists \( \delta > 0 \) such that for every \( v \in \Gamma \), if \( \theta(v) < -\eta \) then \( \phi(v_{\text{next}}) - \phi(v) < -\delta \). For such an algorithm, the following result is a straightforward corollary of Theorem 1.2.8 in [1] and hence its proof is omitted.

**Proposition 2.1.** Suppose that \( |\phi(v)| \) is bounded over \( \Gamma \). If an algorithm is of sufficient descent then it is convergent in the sense of (7).

Refs. [2, 3, 4, 5, 6, 7, 8] used the framework of optimality functions and sufficient descent to define and analyze their respective algorithms for the switched-mode optimal control problem, where \( \Gamma \) is the space of admissible controls \( u \), and \( \theta(u) \) typically is related to the magnitude of the steepest feasible descent-direction vector. In contrast, the optimality function defined in this paper is based on the Hamiltonian rather than the steepest descent or any explicit form of a derivative.

Consider the relaxed control problem defined by Equations (3) and (4). Given a relaxed control \( \mu \in M \), let \( x \) and \( p \) be the associated state trajectory and costate trajectory as defined by (3) and (5), respectively. We use the following optimality function, \( \theta(\mu) \):

\[
\theta(\mu) = \min_{\nu \in M} \int_0^{t_f} (H(x, \nu, p) - H(x, \mu, p)) dt,
\]

(8)

where the Hamiltonians in the integrand of (8) were defined in (6). Recall that \( x \) and \( p \) in (8) are associated with \( \mu \) and hence are independent of \( \nu \); therefore, the compactness of the space of relaxed controls implies that the minimum (not only inf) in (8) exists. Let \( \mu^* \in M \) denote an argmin, then (8) becomes

\[
\theta(\mu) = \int_0^{t_f} (H(x, \mu^*, p) - H(x, \mu, p)) dt.
\]

(9)

We observe that this optimality function satisfies the aforementioned properties with respect to the Maximum Principle: Obviously \( \theta(\mu) \leq 0 \) for every \( \mu \in M \); \( \theta(\mu) = 0 \) if and only if \( \mu \) minimizes the Hamiltonian at almost every \( t \in [0, t_f] \) and hence satisfies the Maximum Principle; and \( |\theta(\mu)| \) arguably indicates the extent to which the Maximum Principle is not satisfied at \( \mu \).

**3. Hamiltonian-based Algorithm**

The analysis in this section is carried out under Assumption 1.1.

---

4In this and later equations we drop the explicit notational dependence of various integrand-terms on \( t \) when no confusion arises.
Given a relaxed control $\mu \in \mathcal{M}$, let $x$ and $p$ denote the related state trajectory and costate trajectory defined by (3) and (5), respectively. For these state and costate, and for every $t \in [0, t_f]$, consider the Hamiltonian $H(x(t), p(t))$, defined by (6), as a function of its second variable. Fix another relaxed control $\nu \in \mathcal{M}$. Now for every $\lambda \in [0, 1]$, $\lambda \nu + (1 - \lambda)\mu$ is also a relaxed control, and we denote it by $\mu_\lambda$. Furthermore, let $\{x_\lambda(t) : t \in [0, t_f]\}$ denote the state trajectory associated with $\mu_\lambda$ as defined by (3), namely,

$$
\dot{x}_\lambda(t) = \lambda \int_U f(x_\lambda(t), u) d\nu(t) + (1 - \lambda) \int_U f(x_\lambda(t), u) d\mu(t),
$$

(10)

and define $\tilde{J}(\lambda) := J(\mu_\lambda)$, where $J$ is defined by (4), namely

$$
\tilde{J}(\lambda) = \int_0^{t_f} \left( \lambda \int_U L(x_\lambda(t), u) d\nu(t) + (1 - \lambda) \int_U L(x_\lambda(t), u) d\mu(t) \right) dt.
$$

(11)

The algorithm described in this section is based on moving from $\mu \in \mathcal{M}$ in the direction of $\nu$ by choosing a step size $\lambda \in [0, 1]$, and therefore, we next characterize those $\nu \in \mathcal{M}$ that provide a direction of descent.

**Proposition 3.1.** The one-sided derivative $\frac{d\tilde{J}}{d\lambda^+}(0)$ exists and has the following form,

$$
\frac{d\tilde{J}}{d\lambda^+}(0) = \int_0^{t_f} (H(x, \nu, p) - H(x, \mu, p)) dt,
$$

(12)

where all the terms in the integrand in the Right-Hand Side (RHS) of (12) are functions of time.

**Proof.** Consider the Right-Hand Sides (RHS) of Equations (10) and (11) as functions of $x = x_\lambda(t)$ and $\lambda \in [0, 1]$, for given relaxed controls $\mu$ and $\nu$. By Assumption 1.1 these functions are twice-continuously differentiable ($C^2$) in $x$, and by (10) and (11) they are linear in $\lambda$ and hence $C^2$ as well. Therefore, standard variational techniques show that the function $\tilde{J}(\lambda)$ is differentiable on $\lambda \in [0, 1]$ and its derivative has the following form,

$$
\frac{d\tilde{J}}{d\lambda}(\lambda) = \int_0^{t_f} \left( p_\lambda(t)^T \int_U f(x_\lambda(t), u) (d\nu(t) - d\mu(t)) \right) dt + \int_U L(x_\lambda(t), u) (d\nu(t) - d\mu(t)) dt,
$$

(13)

where $p_\lambda(t)$ is the costate associated with this derivative. Furthermore, by (10), it is seen that $p_\lambda$ is given by Equation (5) with $\mu_\lambda$ instead of $\mu$. 

8
Next, for \( \lambda = 0 \), we have that \( \mu_0 = \mu \), \( x_0 = x \), and \( p_0 = p \), and therefore, with \( \lambda = 0 \) in (13) we obtain,

\[
\frac{d\tilde{J}}{d\lambda^+}(0) = \int_0^{t_f} \left( p(t)^\top \int_U f(x(t), u)(d\nu(t) - d\mu(t)) \\
+ \int_U L(x(t), u)(d\nu(t) - d\mu(t)) \right) dt.
\]

(14)

By Equation (6) the RHS of (14) is identical to the RHS of (12).

Equation (12) implies that \( \nu \) is a descent direction from \( \mu \) if the RHS of (12) is negative. This is the case if \( \nu(t) \) is a pointwise minimizer of the Hamiltonian over \( M \) at each time \( t \in [0, t_f] \), unless \( H(x(t), \nu(t), p(t)) = H(x(t), \mu(t), p(t)) \) for almost every \( t \in [0, t_f] \). In this case, let us use the notation \( \mu = \nu \) for a pointwise minimizer of the Hamiltonian. The following result implies that the pointwise search for such a minimizer can be confined to \( U \) and need not be extended to \( M \), the space of Borel probability measures on \( U \).

**Proposition 3.2.** Fix \( x \in \mathbb{R}^n \) and \( p \in \mathbb{R}^n \). Let \( u^* \in \text{argmin}\{ H(x, u, p) : u \in U \} \). Then, for every probability measure \( \nu \in M \),

\[
H(x, u^*, p) \leq H(x, \nu, p).
\]

(15)

Note that the Left-hand side (LHS) of (15) is the usual Hamiltonian while its RHS is the relaxed Hamiltonian defined by (6).

**Proof.** By (6) and the fact that \( u^* \) minimizes the ordinary Hamiltonian over \( u \in U \), we have, for every \( \nu \in M \),

\[
H(x, \nu, p) = \int_U H(x, u, p) d\nu \\
\geq \int_U H(x, u^*, p) d\nu = H(x, u^*, p).
\]

(16)

We point out that the point \( u^* \) in the statement of Proposition 3.2 is the pointwise minimizer of the Hamiltonian over \( U \), for given \( x \in \mathbb{R}^n \) and \( p \in \mathbb{R}^n \). Such a minimizer exists and the minimum is finite since, by assumption the set \( U \) is compact, and by Assumption 1.1 the function \( H(x, u, p) \) is continuous in \( u \in U \).

Let \( \mu \) be a relaxed control, and let \( x \) and \( p \) be the associated state trajectory and costate trajectory as defined by Equations (3) and (5), respectively. For every \( t \in [0, t_f] \), let \( u^*(t) \in \text{argmin}\{ H(x(t), u, p(t)) : u \in U \} \). It does not mean that the function \( \{ u^*(t), \ t \in [0, t_f] \} \), is an admissible control since it might not be Lebesgue measurable. On the other hand, we have seen that there exists a relaxed control \( \mu^* \) that minimizes the RHS of (8) over \( \nu \in M \), and hence \( \mu^*(t) \) minimizes \( H(x(t), \nu, p(t)) \) over \( \nu \in M \) (the space of Borel probability measures on \( U \)) for almost every \( t \in [0, t_f] \).

Ideally we would like to choose such \( \mu^* \) as the descent direction of the algorithm from \( \mu \), but its computation may be fraught with difficulties for the
following two reasons: (i) for a given \( t \), \( \mu^*(t) \) may not be a Dirac measure at a point in \( U \), and (ii) The pointwise minimizer \( \mu^*(t) \) has to be computed for every \( t \) in the infinite set \([0, t_f] \). Therefore we choose as descent direction a relaxed control \( \nu \in M \) having the following two properties: (i) \( \nu \sim v \) where \( v \) is a piecewise-constant ordinary control, and (ii) \( \nu \in M \) “almost” minimizes the Hamiltonian in the following sense: For a given a constant \( \eta \in (0, 1) \) which we fix throughout the algorithm (below),

\[
\int_{0}^{t_f} \left( H(x, \nu, p) - H(x, \mu, p) \right) dt \leq \eta \theta(\mu); \tag{17}
\]

\( x, \nu, p, \) and \( \mu \) are all functions of time. We label such \( \nu \) an \( \eta \)-minimizer of the Hamiltonian. It will be seen that it is always possible to choose a relaxed control \( \nu \) with these two properties as long as \( \theta(\mu) < 0 \). With this direction of descent, the algorithm uses the Armijo step size \([12, 1]\). It has the following form.

**Algorithm 3.3.** Given \( \mu \in M \), compute \( \mu_{next} \in M \) by the following steps.

- **Step 0:** If \( \theta(\mu) = 0 \), set \( \mu_{next} = \mu \), then exit.
- **Step 1:** Compute the state and costate trajectories, \( x \) and \( p \), associated with \( \mu \), by using Equations (3) and (5), respectively.
- **Step 2:** Compute a relaxed control \( \nu \in M \) which is an \( \eta \)-minimizer of the Hamiltonian, namely it satisfies Equation (17).
- **Step 3:** Compute the integer \( \ell_{\mu} \) defined as follows,

\[
\ell_{\mu} = \min \left\{ \ell = 0, 1, \ldots : J(\mu + \beta^\ell(v - \mu)) - J(\mu) \leq \alpha \beta^\ell \eta \theta(\mu) \right\}. \tag{18}
\]

Define \( \lambda_{\mu} := \beta^\ell_{\mu} \).
- **Step 4:** Set \( \mu_{next} = \mu + \lambda_{\mu}(v - \mu) \).

1. The algorithm is meant to be run iteratively and compute a sequence \( \{\mu_k\}_{k \geq 1} \) such that \( \mu_{k+1} = \mu_{k, next} \), as long as it does not exit in Step 0.
2. The algorithm does not attempt to solve a two-point boundary value problem. In Step 1 it first integrates the differential equation (3) forward from the initial condition \( x_0 \), and then the differential equation (5) backwards from the specified terminal condition \( p(t_f) = 0 \). We do not specify the particular numerical integration technique that should be used, but say more about it in the sequel.
3. We do not specify the choice of \( \nu \) in Step 2 but rather leave it to the user’s discretion. However, we point out that such an \( \eta \)-minimizer of the Hamiltonian always exists in the form of \( \nu \sim v \) for a piecewise-constant ordinary control \( v \), unless \( \theta(\mu) = 0 \). The reason is that the space of ordinary controls is dense in the space of relaxed controls in the weak star topology on \( L^1([0, t_f]; C(U)) \),
and the space of piecewise-constant ordinary controls is dense in the space of ordinary controls in the $L^1$ norm and hence in the weak-star topology.

4). In Step 4, $\mu_{\text{next}}(t)$ is a convex combination of $\mu$ and $\nu$ in the sense of measures, meaning that for every continuous function $g : U \to R$,

$$\int_U g(u) d\mu_{\text{next}}(t) = \int_U g(u)((1 - \lambda \mu)d\mu(t) + \lambda \mu d\nu(t)).$$

In the event that $\mu \sim u$ and $\nu \sim v$, this means that

$$\int_U g(u) d\mu_{\text{next}}(t) = (1 - \lambda \mu)g(u(t)) + \lambda \mu g(v(t)),$$

which does not necessarily imply that $\mu_{\text{next}} \sim u + \lambda \mu(v - u)$.

5). The Armijo step size, computed in Step 3, is commonly used in nonlinear programming as well as in infinite-dimensional optimization. Reference [1] contains analyses of several algorithms using it and practical guidelines for its implementation.

6). It will be proven that the integer $\ell_\mu$ defined in Step 3 is finite as long as $\theta(\mu) < 0$, hence the algorithm cannot jam at a point (relaxed control) that does not satisfy the Maximum Principle, namely the condition $\theta(\mu) = 0$.

We point out that the idea of a descent direction comprised of a pointwise minimizer of the Hamiltonian has its origin in [40]. The algorithm proposed in that reference is defined in the setting of ordinary controls, its descent direction is all the way to a minimizer of the Hamiltonian at a subset of the time-horizon $[0, t_f]$, and its main convergence result is stated in terms of accumulation points of computed iterate-sequences. The algorithm in this paper is quite different in that it is defined in the setting of relaxed controls, it moves part of the way towards a minimizer of the Hamiltonian throughout the entire time horizon, and its analysis is carried out in the context of optimality functions.

We next establish convergence of Algorithm 3.3.

**Proposition 3.4.** Suppose that Assumption 1.1 is satisfied. Let $\{\mu_k\}_{k=1}^{\infty}$ be a sequence of relaxed controls computed by Algorithm 3.3, such that for every $k = 1, 2, \ldots$, $\mu_{k+1} = \mu_{k,\text{next}}$. Then,

$$\lim_{k \to \infty} \theta(\mu_k) = 0. \quad (20)$$

The proof is based on the following lemma, variants of which have been proved in [11] (e.g., Theorem 1.3.7). We supply the proof in order to complete the presentation.

**Lemma 3.5.** Let $g(\lambda) : R \to R$ be a twice-continuously differentiable ($C^2$) function. Suppose that $g'(0) \leq 0$, and there exists $K > 0$ such that $|g''(\lambda)| \leq K$ for every $\lambda \in R$. Fix $\alpha \in (0, 1)$, and define $\gamma := 2(1 - \alpha)/K$. Then for every positive $\lambda \leq \gamma|g'(0)|$,

$$g(\lambda) - g(0) \leq \alpha \lambda g'(0). \quad (21)$$
Proof. Recall (see \[1\], Eq. (18b), p.660) the following exact second-order expansion of $C^2$ functions,

$$g(\lambda) = g(0) + \lambda g'(0) + \lambda^2 \int_0^1 (1-s)g''(s\lambda)ds. \quad (22)$$

Using this and the assumption that $|g''(\cdot)| \leq K$, we obtain that

$$g(\lambda) - g(0) - \alpha \lambda g'(0) = (1 - \alpha) \lambda g'(0) + \lambda^2 \int_0^1 (1-s)g''(s\lambda)ds \leq \lambda((1 - \alpha)g'(0) + \lambda K/2). \quad (23)$$

For every positive $\lambda \leq \gamma|g'(0)|$, $(1 - \alpha)g'(0) + \lambda K/2 \leq 0$, and hence, and by (23), Equation (21) follows. \[\Box\]

Proof of Proposition 3.4. Let $\mu \in \mathbf{M}$ and $\nu \in \mathbf{M}$ be any two relaxed controls, and for $\lambda \in [0,1]$, consider $\tilde{J}(\lambda)$ as defined by Equation (11). We next show that $\tilde{J}$ is a twice-continuously differentiable function of $\lambda$, and there exists $K > 0$ such that, for all $\mu \in \mathbf{M}$, $\nu \in \mathbf{M}$, and $\lambda \in [0,1]$,

$$\left| \frac{d^2 \tilde{J}}{d\lambda^2}(\lambda) \right| \leq K. \quad (24)$$

This follows from variational arguments developed and summarized in \[1\] as follows. First, consider two ordinary controls, $u$ and $w$, and define $u_\lambda := \lambda w + (1-\lambda)u$ for every $\lambda \in [0,1]$. Denote by $x_\lambda$ the state trajectory associated with $u_\lambda$ as defined by (1), and let $\tilde{J}(\lambda) := J(u_\lambda)$ be the cost functional, defined by (2), as a function of $\lambda$. By Assumption 1.1, an application of Corollary 5.6.9 and Proposition 5.6.10 in \[1\] to variations in $\lambda$ yields that $\tilde{J}(\lambda)$ is continuously differentiable in $\lambda$, and the term $|d\tilde{J}/d\lambda(\lambda)|$ is bounded from above over all ordinary admissible controls $u$, $w$, and $\lambda \in [0,1]$. A second application of these arguments to the derivative $d\tilde{J}/d\lambda$, supported by the $C^2$ assumption (Assumption 1.1), yields that $\tilde{J}(\lambda)$ is twice continuously differentiable and its second derivative also is bounded from above over all ordinary admissible controls $u$, $w$, and $\lambda \in [0,1]$.

By the Lebesgue Dominated Convergence Theorem, the same result holds true when $u$ and $w$ are replaced by two respective relaxed controls, $\mu$ and $\nu$, and the state equation and cost function are defined by Equations (3) and (4), respectively. This shows that $\tilde{J}$ is $C^2$ in $\lambda$, and there exists $K > 0$, independent of $\mu \in \mathbf{M}$, $\nu \in \mathbf{M}$, and $\lambda \in [0,1]$, such that Equation (24) is satisfied.

Let us apply this result to $\mu$ and $\nu$ where $\mu \in \mathbf{M}$ is a given relaxed control and $\nu$ is defined in Step 2 of Algorithm 3.3. Recall the constant $\alpha \in (0,1)$ that is used by the algorithm, and define $\gamma := 2(1 - \alpha)/K$. By Lemma 3.5, for every positive $\lambda < \gamma|d\tilde{J}/d\lambda(0)|$,

$$\tilde{J}(\lambda) - \tilde{J}(0) \leq \alpha \lambda \frac{d\tilde{J}}{d\lambda}(0). \quad (25)$$
Recall that $\tilde{J}(\lambda) := J(\mu_\lambda)$, and therefore, according to the notation in the RHS of (18), $J(\mu + \beta^t (\mu^* - \mu)) - J(\mu) = \tilde{J}(\beta^t) - \tilde{J}(0)$; consequently, if $\beta^t < \gamma \frac{d\tilde{J}}{d\lambda^+}(0)$, then (25) is satisfied with $\lambda = \beta^t$, namely,

$$J(\mu + \beta^t (\mu^* - \mu)) - J(\mu) \leq \alpha \beta^t \frac{d\tilde{J}}{d\lambda^+}(0).$$

(26)

By Proposition 3.1 (Equation (12)) and the choice of $\nu$ in Step 2 (Equation (17)), the RHS of (26) implies that

$$J(\mu + \beta^t (\mu^* - \mu)) - J(\mu) \leq \alpha \beta^t \eta \theta(\mu)$$

(27)

as long as $\beta^t < \gamma \frac{d\tilde{J}}{d\lambda^+}(0)$. But (12), (17), and the fact that $\frac{d\tilde{J}}{d\lambda^+}(0) \leq 0$ imply that $|\frac{d\tilde{J}}{d\lambda^+}(0)| \geq \eta \theta(\mu)$ and hence $\gamma |\frac{d\tilde{J}}{d\lambda^+}(0)| \geq \eta \gamma \theta(\mu)$; consequently (27) is satisfied as long as $\beta^t < \eta \gamma \theta(\mu)$. Therefore, and by Equation (18), the step size $\lambda_\mu$ defined in Step 3 satisfies the inequality

$$\lambda_\mu := \beta^t \geq \beta \eta \gamma \theta(\mu).$$

(28)

Next, Equations (18) and (19) imply that

$$J(\mu_{next}) - J(\mu) \leq \alpha \lambda_\mu \eta \theta(\mu),$$

(29)

and hence, and by (28) and the fact that $\theta(\mu) \leq 0$, we have that

$$J(\mu_{next}) - J(\mu) \leq -\alpha \beta \gamma \eta \theta(\mu)^2.$$  

(30)

But $\gamma$ is independent of $\mu$ or $\nu$, and hence Equation (30) implies that the algorithm is of a sufficient descent.

Finally, the set $U$ is compact by assumption, and therefore standard applications of the Bellman-Gronwall inequality and Equation (4) yield that $|J(\mu)|$ is upper-bounded over all $\mu \in M$. Consequently Proposition 2.1 implies the validity of Equation (20).

A note on implementation. Implementations of Algorithm 3.3 generally require numerical integration methods for computing (or approximating) $x$ and $p$ in Step 1, and a computation of $\nu$ in Step 2 that is based on the pointwise minimization of the Hamiltonian at a finite number of points in the time-horizon $[0, t_f]$. Both require finite grids on the time horizon, which may be different and can vary from one iteration to the next. The choice of the grid sizes generally comprises a balance between precision and computing times. A rule-of-thumb proposed in [1] is to adjust the grid adaptively by tightening it whenever it is sensed that a local optimum is approached. This adaptive-precision technique underscores Polak’s algorithmic framework of consistent approximation for infinite-dimensional optimization while guaranteeing convergence in the sense of Eq. (20).

In this paper we are not concerned with the formal rules for adjusting the grids (and hence precision levels). Instead, we run the algorithm several times
per problem, each with a fixed grid, to see how far it can minimize the cost functional. The goal of this experiment is to test the tradeoff between precision and computing times. The results, presented in the next section, indicate rapid descent from the initial guess regardless of how far it is from the optimum. In fact, the key argument in the proof of convergence is the sufficient descent property of the algorithm, captured in Equation (30), which implies large descents until an iteration-sequence approaches a local minimum. This suggests that the main utility of the algorithm is not in the asymptotic convergence close to a minimum (where higher-order methods can be advantageous), but rather in its approach to such points. For a detailed discussion of this point and some comparative results please see Section 4.

Another point related to implementation concerns the algorithm’s having to compute relaxed controls, which are more complicated objects than ordinary controls. To address this concern we next examine a class of systems where the state equation \( f(x, u) \) is affine in \( u \) and the cost function \( L(x, u) \) is convex in \( u \).

Consider the case where

\[
f(x, u) = \phi_f(x) + \Psi_f(x)u,
\]

where the functions \( \phi_f : \mathbb{R}^n \to \mathbb{R}^n \) and \( \Psi_f : \mathbb{R}^n \to \mathbb{R}^{n \times k} \) (the latter being the space of \( n \times k \) matrices) satisfy Assumption 1.1. By the linearity of the integration operator and the fact that \( \mu(t) \) is a probability measure, Equation (3) assumes the form

\[
\dot{x}(t) = f\left(x(t), \int_U u d\mu(t)\right),
\]

meaning that, for the purpose of computing the state trajectory, the convexification of the vector field inherent in (3) yields the same results as a convexification of the control. Defining \( \bar{u}(t) := \int_U u d\mu(t) \), the state equation becomes \( \dot{x} = f(x, \bar{u}) \), and we can view \( \bar{u} \) as a control function from \([0, T]\) into \( \text{conv}(U) \). Likewise, if \( L(x, u) = \phi_L(x) + \psi_L(x)u \) with \( \phi_L : \mathbb{R}^n \to \mathbb{R} \) and \( \psi_L : \mathbb{R}^n \to \mathbb{R}^{1 \times k} \), then (4) becomes

\[
J(\mu) = \int_0^T L(x, \bar{u}) dt.
\]

In this case the relaxed optimal control problem is cast as an ordinary optimal control problem with the input constraints \( \bar{u}(t) \in \text{conv}(U) \). Moreover, if Algorithm 3.3 starts at an ordinary control on \( \text{conv}(U) \) then it could compute only such ordinary controls. The reason is that \( \nu \) in Step 2 can always be an ordinary control (as earlier said), and the convexification of measures in Eq. (19) can be carried out by the convexification of ordinary controls. Therefore, if in Eq. (19), \( \mu \sim u \) and \( \nu \sim v \) for ordinary admissible controls (on \( \text{conv}(U) \)), then (19) yields

\[
\mu_{\text{next}} \sim u + \lambda \mu(v - u),
\]

which is an ordinary admissible control on \( \text{conv}(U) \). This is the case of autonomous switched-mode systems, as will be demonstrated in Section 4.
Consider next the case where \( f(x,u) \) is affine in \( u \) as in Equation (31), and \( L(x,u) \) is convex in \( u \) for every \( x \in \mathbb{R}^n \). Then Equation (32) is true but (33) and hence (34) are not true. Therefore, supposing that \( \mu \sim u \) and \( \nu \sim v \) for ordinary controls (on \( \text{conv}(U) \)), Equation (19) yields that \( \mu_{\text{next}} \) is a relaxed control but not an ordinary control on \( \text{conv}(U) \). However, the convexity of \( L(x,u) \) in \( u \) in conjunction with (32) imply that, for every \( \lambda \in [0,1] \)

\[
J(u + \lambda(v - u)) \leq J(u) + \lambda \mu(J(v) - J(u)) = J(\mu) + \lambda \mu(J(\nu) - J(\mu)).
\]

(35)

In this setting, Algorithm 3.3 uses the convexified cost in Step 3 (Eq. (18)), but by (35), we would get a lower value by convexifying the control. Consequently, the inequality in (18) would imply a similar inequality with the convexified control, as in Eq. (36), below. Modifying Algorithm 3.3 accordingly, the following algorithm results.

**Algorithm 3.6.** Given \( \mu \sim u \) such that \( u \) is an admissible control on \( \text{conv}(U) \), compute \( \mu_{\text{next}} \in \mathbb{M} \) by the following steps.

**Step 0:** If \( \theta(\mu) = 0 \), set \( \mu_{\text{next}} = \mu \), then exit.

**Step 1:** Compute the state and costate trajectories, \( x \) and \( p \), associated with \( \mu \), by using Equations (3) and (5), respectively.

**Step 2:** Compute an \( \eta \)-minimizer of the Hamiltonian, \( \nu \sim v \), such that \( v \) is an ordinary control on \( \text{conv}(U) \).

**Step 3:** Compute the integer \( \ell_{\mu} \) defined as follows,

\[
\ell_{\mu} = \min \{ \ell = 0, 1, \ldots : J(u + \beta^\ell (v - u)) - J(u) \leq \alpha \beta^\ell \eta \theta(\mu) \}.
\]

(36)

Define \( \lambda_{\mu} := \beta^{\ell_{\mu}} \).

**Step 4:** Set

\[
\mu_{\text{next}} \sim u + \lambda_{\mu}(v - u).
\]

(37)

Note that, by (34) and (37), an iterative application of Algorithm 3.6 would compute only ordinary controls that are admissible on \( \text{conv}(U) \). Furthermore, as discussed earlier, by Equation (35), if the Armijo test in Equation (18) were to be satisfied for a given \( \ell \) then it would be satisfied in (36) as well and result in a lower descent in \( J(\mu_{\text{next}}) - J(\mu) \). Since this descent in (18) yields the key condition of uniform descent for Algorithm 3.3, it also holds for Algorithm 3.6 thereby guaranteeing its convergence via a verbatim application of Proposition 3.4.

### 4. Simulation Results

This section reports on applications of Algorithm 3.3 and Algorithm 3.6 to three problems: an autonomous switched-mode problem, a controlled switched-mode problem, and a problem of balancing motion energy with transmission...
energy in a mobile network. In the first problem the state equation and the cost function are affine in $u$ and hence Algorithm 3.3 is identical to Algorithm 3.6; in the second problem the cost function is not affine in $u$ and hence we use Algorithm 3.6; and in the third problem the state equation is affine in $u$ and the cost function in convex in $u$ and hence we can use Algorithm 3.6. The first two problems were considered in [7], and we use its reported results as benchmarks for our algorithms. The third problem was addressed in [41], but we choose here an initial guess that is farther from the optimum. As stated earlier the efficiency of the algorithm depends on the ease with which the pointwise minimizer of the Hamiltonian can be computed, and for all three problems it will be shown to be computable via a simple formula. Since the resulting function $u^*$ is an ordinary control, we use $\nu \sim u$ in Step 2.

4.1. Double Tank System

Consider a fluid-storage tank with a constant horizontal cross section, where fluid enters from the top and discharged through a hole at the bottom. Let $v(t)$ denote the fluid inflow rate from the top, and let $x(t)$ be the fluid level in the tank. According to Toricelli’s law, the state equation of the system is $\dot{x}(t) = v(t) - \sqrt{x(t)}$. The system considered in this subsection is comprised of two such tanks, one on top of the other, where the fluid input to the upper tank is from a valve-controlled hose at the top, and the input to the lower tank consists of the outflow process from the upper tank. Denoting by $u(t)$ the inflow rate to the upper tank, and by $x(t) := (x_1(t), x_2(t))^\top$ the fluid levels in the upper tank and lower tank, respectively, the state equation of the system is

$$\dot{x} = \begin{pmatrix} u - \sqrt{x_1} \\ \sqrt{x_1} - \sqrt{x_2} \end{pmatrix};$$

we assume the initial condition $x(0) = (2.0, 2.0)^\top$. The control input $u(t)$ is assumed to be constrained to the two-point set $U := \{1.0, 2.0\}$, and hence the system can be viewed as an autonomous switched-mode system whose modes correspond to the two possible values of $u$. The considered problem is to have the fluid level at the lower tank track the value of 3.0, and accordingly we choose the cost functional $J$ to be

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

As in [7], the final time is $t_f = 10.0$.

By (38), $f(x, u)$ is affine in $u$, and by (39), $L(x, u)$ does not depend on $u$ and hence can be considered affine. Therefore Algorithm 3.6 can be run as a special case of Algorithm 3.3, where $\text{conv}(U) = [1, 2]$. Moreover, with the costate $p := (p_1, p_2)^\top \in \mathbb{R}^2$, the Hamiltonian has the form

$$H(x, u, p) = p_1 u - p_1 \sqrt{x_1} + p_2 (\sqrt{x_1} - \sqrt{x_2}) + 2(x_2 - 3)^2,$$

The code was written in MATLAB and executed on a laptop computer with an Intel i7 quad-core processor, clock frequency of 2.1 GHz, and 8GB of RAM.
whose pointwise minimizer is

\[ u^* = \begin{cases} 
1, & \text{if } p_1 \geq 0, \\
2, & \text{if } p_1 < 0; 
\end{cases} \]

if \( p_1 = 0 \) then \( u^* \) can be any point in the interval \([1, 2]\).

We ran Algorithm 3.6 starting from the initial control \( u_1(t) = 1 \ \forall t \in [0, t_f] \), having the cost \( J(u_1) = 50.5457 \). All numerical integrations were performed by the forward Euler method with \( \Delta t = 0.01 \), and we approximate \( u^* \) by its zero-order hold with the sample values \( u^*(i\Delta t), i = 0, 1, \ldots \). We benchmark the results against the reported run of the algorithm in [7] which, starting from the same initial control, obtained the final cost of 4.829; we reached a similar final cost. In fact, 100 iterations of our algorithm reduced the cost from \( J(\mu_1) = 50.5457 \) to \( J(\mu_{100}) = 4.7440 \) in 2.6700 seconds of CPU time.

Figure 1 depicts the graph of \( J(\mu_k) \) vs. the iteration count \( k = 1, \ldots, 100 \). The graph indicates a rapid reduction in the cost from its initial value until it stabilizes after about 18 iterations. The L-shaped graph is not atypical in applications of descent algorithms with Armijo step sizes, whose strength lies in its global stability and large strides towards local solution points at the initial stages of its runs. As a matter of fact, similar L shaped graphs were obtained from all of the algorithm runs reported on in this section.

The final relaxed control computed by Algorithm 3.6, \( \mu_{100} :\sim u_{100} \), was projected onto the space of admissible (switched-mode) controls by Pulse-Width Modulation (PWM) with cycle time of 0.5 seconds. The resulting cost value is \( J(u_{\text{fin}}) = 4.7446 \), and the combined runs of the algorithm and the projection took 2.6825 seconds of CPU time. We point out that the projection had to be performed only once, after Algorithm 3.6 had completed its run.
Returning to the run of Algorithm 3.6, the L-shaped graph in Figure 1 suggests that a reduction in CPU times can be attained, if necessary, by computing fewer iterations. Moreover, further reduction can be obtained by taking larger integration steps without significant changes in the final cost. To test this point we ran the algorithm from the same initial control for 50 iterations with $\Delta t = 0.05$, and it reduced the cost-value from $J(\mu_1) = 50.5282$ to $J(\mu_{50}) = 4.8078$ in 0.2939 seconds of CPU time; including the projection onto the space of switched-mode controls it reached $J(u_{fin}) = 4.8139$ in a total time of 0.3043 seconds. With a larger integration step, $\Delta t = 0.1$, the algorithm yielded a cost-reduction from $J(\mu_1) = 50.5069$ to $J(\mu_{50}) = 4.8816$ in 0.1566 seconds of CPU time, and $J(u_{fin}) = 4.8915$ in a total time of 0.1655 seconds. These results are summarized in Table 1.

| $\Delta t$ | $k$ | $J(\mu_k)$ | CPU | $J(u_{fin})$ | CPU |
|-----------|-----|------------|-----|-------------|-----|
| 0.01; 100 | 4.7440 | 2.6700 | 4.7446 | 2.6825 |
| 0.05; 50  | 4.8078 | 0.2939 | 4.8139 | 0.3043 |
| 0.1; 50   | 4.8816 | 0.1566 | 4.8915 | 0.1655 |

Table 1: Double-tank problem, $J(\mu_1) = 50.546$

The CPU times indicated in Table 1 are less than the run-time reported in [7] for solving the same problem (32.38 seconds). However, these numbers should not be considered as a sole basis for comparing the two techniques since the respective algorithms were implemented on different hardware and software platforms. Furthermore, the algorithm in [7] has a broader scope than ours, while our code is specific for the problem in question. The only conclusion we draw from Table 1 is that Algorithm 3.6 may have merit and deserves further investigation. We believe, however, that our choice of the descent direction, namely the pointwise minimizer of the Hamiltonian, plays a role in the fast run times as well as simplicity of the code as compared with explicit-gradient techniques.

We close this discussion with a comment on convergence of the algorithm in the control space. Algorithm 3.6 (as well as Algorithm 3.3) is defined in the space of relaxed controls where its convergence is established in the weak star topology. Therefore, there is no reason to expect the sequence of computed controls, $\{u_k\}$, to converge in any (strong) functional norm such as $L^1$. In fact, the graphs of $u_k(t)$ for $k = 1, 20, 100$, are depicted in Figure 2, where no strong convergence is discerned. However, the weak convergence proved in Section 3 suggests that the cost-sequence $\{J(u_k)\}$ would converge to the minimal cost, and this indeed is evident from Figure 1. Furthermore, the associated sequence

---

6Ref. [7] refers to a software package for its algorithm, but we were unable to run it because apparently it is linked to a proprietary code. Therefore we were unable to conduct a direct comparison between the two algorithms.
Figure 2: Two-tank system: graph of $u_1(t)$ (dashed), $u_{20}(t)$ (dash-dotted), and $u_{100}(t)$ (solid) of state trajectories are expected to converge in a strong sense ($L^\infty$ norm) to the state trajectory of the optimal control, and this is indicated by Figure 3 depicting the graphs of $x_{k,2}(t)$ (the fluid levels at the lower tank) for $k = 1, 20, 100$.

4.2. Hybrid LQR

This problem was considered in [7] as well. Consider the switched-linear system $\dot{x} = Ax + bv$, where $x \in \mathbb{R}^3$, $A = \begin{pmatrix} 1.0979 & -0.0105 & 0.0167 \\ -0.0105 & 1.0481 & 0.0825 \\ 0.0167 & 0.0825 & 1.1540 \end{pmatrix}$, $b \in \mathbb{R}^3$, and $v \in \mathbb{R}$ is a continuum-valued input. We assume the initial condition to be $x_0 := x(0) = (0, 0, 0)^T$. The set $B$ consists of three points, namely $B = \{b_1, b_2, b_3\}$, with $b_1 = (0.9801, -0.1987, 0)^T$, $b_2 = (0.1743, 0.8601, -0.4794)^T$, and $b_3 = (0.0952, 0.4699, 0.8776)^T$. The continuum-valued control $v$ is constrained to $|v| \leq 20.0$.

All three modes of the system are unstable since the eigenvalues of $A$ are in the right-half plane, and the problem is to switch among the vectors $b \in B$ and choose $\{v(t)\}$ in a way that brings the state close to the target state $x_f := (1, 1, 1)^T$ in a given amount of time and in a way that minimizes the input energy. The corresponding cost functional is

$$J = 0.01 \int_0^{t_f} v^2 \, dt + \|x(t_f) - x_f\|^2$$

(40)

with $t_f = 2.0$, and the control variable is $u = (b, v) \in B \times [-20, 20]$. Reference [7] solved this problem from the initial control of $v_1(t) = 0$ and $b = \ldots$
(0.9801, -0.1987, 0)\T for all $t \in [t_0, t_f]$, and attained the final cost of $1.23 \cdot 10^{-3}$ in (reported) 9.827 seconds of CPU time.

Since $f(x, u)$ is not affine in $u$ (due to the term $bv$) Algorithm 3.6 may not be applicable and hence we used Algorithm 3.3. We chose the same initial guess $u_1$ as in [7] and ran the algorithm for 20 iterations. The results indicate a similar L-shaped graph of $J(\mu_k)$ to the one shown in Figure 1, and attained a cost-value reduction from $J(u_1) = 3.000$ to $J(\mu_{20}) = 2.768 \cdot 10^{-3}$ in 0.761 seconds of CPU time. All integrations were performed by the forward Euler method with $\Delta t = 0.01$, and the boundary condition of Equation (5) was $p(t_f) = 2(x(t_f) - (1, 1, 1)\T)$ due to the cost on the final state.7 The Hamiltonian at a control $u = (b, v)$ has the form $H(x, u, p) = p\T(Ax + bv) + 0.01v^2$, and hence its minimizer over $U$, $u^* = (b^*, v^*)$, is computable as follows: For every $b_i \in B_i$, $i = 1, 2, 3$, define $v_i$ according to the following three contingencies: (i) if $|p\T b_i/0.02| \leq 20$, then $v_i = -p\T b_i/0.02$; (ii) if $p\T b_i/0.02 > 20$, then $v_i = -20$; and (iii) if $p\T b_i/0.02 < -20$, then $v_i = 20$. It is readily seen that $u^* = \arg\min \{H(x, (b_i, v_i), p) : i = 1, 2, 3\}$ is a minimizer of the Hamiltonian.8

A typical relaxed control can be represented as $\mu(t) \sim \sum_{i=1}^{3} \alpha_i(t) b_i v_i(t)$, with $\alpha_i(t) \in [0, 1], i = 1, 2, 3; \sum_{i=1}^{3} \alpha_i(t) = 1$; and $|v_i(t)| \leq 20.0, i = 1, 2, 3$ (this is an embedded control as defined in [18]). It can be seen, after some

---

7The cost functional in (40) is not quite in the form of (2) due to the addition of the final-state cost term $\phi(x(t_f)) := ||x(t_f) - x_f||^2$. However, using the standard transformation of a Bolza optimal control problem (as in (40)) to a Lagrange problem (as in (2)) and applying the algorithm to the latter, requires only one change, namely setting the boundary condition of the costate to $p(t_f) = \nabla \phi(x_f)$; all else remains the same.

8By Proposition 3.2, a pointwise minimizer of the Hamiltonian always can be found in $U$ and does not have to be in $M$. 

---

Figure 3: Two-tank system: graph of $x_{2,1}(t)$ (dashed), $x_{2,20}(t)$ (dash-dotted), and $x_{2,100}(t)$ (solid)
algebra, that this can be represented as
\[ \mu(t) \sim \left( \sum_{i=1}^{3} \gamma_i(t) \bar{b}_i(t) \right) w(t), \]
with \( \gamma_i(t) \in [0,1], \ i = 1, 2, 3; \sum_{i=1}^{3} \gamma_i(t) = 1; \ b_i(t) \in \{ b_i, -b_i \}, \ i = 1, 2, 3; \) and \( |w(t)| \leq 20.0. \) Define \( \bar{w}_i(t) \) as \( \bar{w}_i(t) = w \) if \( b_i(t) = b_i \), and \( \bar{w}_i(t) = -w \) if \( b_i(t) = -b_i \). The projection \( \mu(t) \) onto the space of ordinary controls was done by PWM as described in the previous subsection, with \( u \) having the successive values \( (b_1, \bar{w}_1(t)), (b_2, \bar{w}_2(t)), (b_3, \bar{w}_3(t)) \) in each cycle according to the coefficients \( \gamma_i(t), \ i = 1, 2, 3. \) The cycle time was \( 12\Delta t. \) Twenty iterations of Algorithm 3.3 followed by the projection of \( \mu_{20} \) onto the space of switched-mode controls required a total CPU time of 0.803 seconds and yielded a final cost of \( J(u_{\text{fin}}) = 2.956 \cdot 10^{-3}; \) the results are summarized in Table 2.

| \( \Delta t \cdot k \) | \( J(\mu_k) \)  | CPU  | \( J(u_{\text{fin}}) \)  | CPU  |
|-----------------|----------------|------|-----------------|------|
| 0.01; 20        | 2.768 - 10^{-3} | 0.761| 2.956 - 10^{-3} | 0.803|

Table 2: Hybrid-LQR problem, \( J(\mu_1) = 3.00 \)

Now consider the problem of minimizing the cost functional \( 0.01 \int_{0}^{t_f} u^2 dt \) subject to the constraint \( x(t_f) = x_f := (1.0, 1.0, 1.0)^T. \) The definition of \( J \) in Eq. (40) appears to address this problem with the penalty function \( ||x(t_f) - x_f||^2. \) As a matter of fact, the final state \( x(t_f) \) obtained form the run of the algorithm is \( x(t_f) = (0.9994, 0.9991, 0.9998)^T, \) and after projecting the relaxed control \( u_{20} \) onto the space of ordinary controls, the corresponding final state is \( (0.9965, 1.0020, 0.9875)^T. \)

While Algorithm 3.3 could be applied to the current problem, its scope does not include some embedded optimal control problems, comprising a class or relaxed-control problems defined in [18]. However, we believe that a possible extension of Algorithm 3.6 can close this gap, and is currently under investigation.

4.3. Balancing Mobility with Transmission Energy in Mobile Sensor Networks

In Reference [41] we considered a path-planning problem for mobile communication-relay networks, whose objective is to optimize a weighted sum of transmission energy and fuel consumption. We used there Algorithm 3.6 to solve it. Next we present simulation results for the same problem, but with a different initial control, \( u_1 \), chosen farther from the optimum in order to highlight the drastic cost-reduction of the algorithm’s run at its initial phases.

Consider a scenario where a given number \( (N) \) of mobile sensors (agents) are placed in a terrain, and at time \( t = 0 \) they are tasked with forming a point-to-point relay network for communications between a stationary object and a stationary controller. Upon issuance of the command the agents start moving while transmitting. Given the final time \( t_f \), the problem is to compute the agents’ paths in a way that minimizes a weighted sum of their fuel consumption and transmission energy over the time-interval \( t \in [0, t_f] \). Of course the optimal paths depend on the positions of the object and controller, as well as on the initial positions of the agents.

21
A detailed description of the problem and justification of the assumptions made can be found in [41]. As in [41], we assume that the agents’ positions $x_i$, $i = 1, \ldots, N$, are confined to a line-segment $[0, d]$ for a given $d > 0$. Denote by $x = (x_1, \ldots, x_N)^\top \in R^N$ the vector of the agents’ positions, and by $u = (u_1, \ldots, u_N)^\top \in R^N$, the vector of their corresponding velocities. Viewing $x$ as the state of the system and $u$ as its control input, the state equation is

$$\dot{x} = u.$$  \hfill (41)

The power required to transmit a signal over a $z$-long channel can be considered as proportional to $z^2$ (see [41]), and the fuel rate required to move an agent is proportional to the speed of motion. Consequently, and defining $x_0 := 0$ and $x_{N+1} := d$, the considered cost-performance functional is

$$J = \sum_{i=1}^{N+1} \int_0^{t_f} (x_i - x_{i-1})^2 dt + C \sum_{i=1}^{N} \int_0^{t_f} |u_i| dt$$  \hfill (42)

for a given $C > 0$. The optimal control problem is to minimize $J$ for a given initial condition $x(0)$, subject to the pointwise input constraints $|u_i| \leq \bar{u}$ for a given $\bar{u} > 0$.

By Equations (41)-(42) it can be seen that the costate $p := (p_1, \ldots, p_N)^\top$ is defined by the equation

$$p_i = 2(x_{i-1} + x_{i+1} - 2x_i),$$  \hfill (43)

$i = 1, \ldots, N$, with the boundary condition $p_i(t_f) = 0$. Therefore, given a control $u \in R^N$ and its associated state $x \in R^N$ and costate $p = (p_1, \ldots, p_N)^\top \in R^N$, the Hamiltonian has the form $H(x, u, p) = \sum_{i=1}^{N} p_i u_i + J$ with $J$ defined in (42), and its minimizer, $u^* = (u_1^*, \ldots, u_N^*)^\top$, is computable as follows,

$$u_i^* = \begin{cases} -\text{sgn}(p_i)\bar{u}, & \text{if } |p_i| > C \\ 0, & \text{if } |p_i| \leq C \end{cases}$$  \hfill (44)

(e.g., [41]).

In our simulation experiments we considered an example with $N = 6$ (six agents), $d = 20$ (hence the agents move in the interval $[0, 20]$), $t_f = 20$, $C = 7$, $\bar{u} = 1$, and the initial state is $x(0) = (1, 2, 7, 9, 12, 19)^\top$. The numerical integrations were performed by the forward Euler method with the step size $\Delta t = 0.01$. The algorithm started with the following initial control, $u_1 := (u_{1,1}, \ldots, u_{1,6})^\top$: $u_1(t) = 1.0$; $u_2(t) = \sin(\pi t/4)$; $u_3(t) = 3u_2(t)$; $u_4(t) = 2u_3(t)$; $u_5(t) = 2u_4(t)$; and $u_6(t) = u_5(t) - 4.3$, with the initial cost of $J(u_1) = 81,883.4$

A 200-iteration run took 11.3647 seconds of CPU time and yielded the final cost of $J(u_{200}) = 1,253.4$ (extensive simulations in [41] suggested that this is about the global minimum). The graph of $J(u_k)$ vs. $k$ has a similar L shape to Figure 1, and it took only 4 iterations (9 iterations, resp.) to achieve 98% (99%, resp.) of the total cost reduction to $J(u_5) = 2,701.6$ ($J(u_{10}) = 2,037.6$, resp). To reduce the CPU times we can take fewer iterations. For example, 20
iterations take 0.8653 seconds of CPU times to obtain $J(u_{20}) = 1,455.5$, and 100 iterations took 5.2379 seconds to yield $J(u_{100}) = 1,256.7$. Further speedup can be achieved by increasing the integration step size: with $\Delta t = 0.1$, 100 iterations took 0.5645 seconds of CPU time to obtain $J(u_{100}) = 1,260.4$. Note that this is quite close to the aforementioned apparent minimum of 1,253.4. These results are summarized in Table 3.

| $\Delta t$ | $k$  | $J(\mu_k)$ | CPU       |
|----------|-----|------------|-----------|
| 0.01     | 200 | 1,253.4    | 11.3647   |
| 0.01     | 100 | 1,256.7    | 5.2379    |
| 0.01     | 20  | 1,455.5    | 0.8653    |
| 0.1      | 100 | 1,260.4    | 0.5645    |

Table 3: Path planning for power-aware mobile networks, $J(u_1) = 81,883.4$

5. Conclusions

This paper presents an iterative algorithm for solving a class of optimal control problems. The algorithm operates in the space of relaxed controls and the obtained result is projected onto the space of ordinary controls. The computation of the descent direction is based on pointwise minimization of the Hamiltonian at each iteration instead of explicit gradient calculations. Simulation examples indicate fast convergence for a number of test problems.

References

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

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

[3] T. Caldwell and T. D. Murphey. Switching mode generation and optimal estimation with application to skid-steering, *Automatica*, vol. 47, no. 1, pp. 5064, 2011.

[4] T. Caldwell and T. D. Murphey. Single integration optimization of linear time-varying switched systems. *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 15921597, 2012.

[5] T. Caldwell and T. Murphey. Projection-Based Iterative Mode Scheduling for Switched Systems, *Nonlinear Analysis: Hybrid Systems*, to appear, 2016.
[6] 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.

[7] R. Vasudevan, H. Gonzalez, R. Bajcsy, and S.S. Sastry. Consistent Approximations for the Optimal Control of Constrained Switched Systems - Part 1: A Conceptual Algorithm, and Part 2: An Implementable Algorithm. SIAM Journal on Control and Optimization, Vol. 51, pp. 4663-4483 (Part 1) and pp. 4484-4503 (Part 2), 2013.

[8] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-Time Optimization for Switched Systems. IEEE Transactions on Automatic Control, Vol. 51, No. 1, pp. 110-115, 2006.

[9] 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.

[10] E.J. McShane. Ralexed Controls and Variational Problems. SIAM Journal on Control, Vol. 5, pp. 438-485, 1967.

[11] J. Warga, Optimal Control of Differential and Functional Equations, Academic Press, 1972.

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

[13] 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.

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

[15] 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.

[16] 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.

[17] M. Garavello and B. Piccoli. Hybrid Necessary Principle. SIAM J. Control and Optimization, Vol. 43, pp. 1867-1887, 2005.

[18] S.C. Bengea and R. A. DeCarlo. Optimal control of switching systems. Automatica, Vol. 41, pp. 11-27, 2005.
[19] 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.

[20] F. Taringoo and P.E. Caines. On the optimal control of hybrid systems on Lie groups and the exponential gradient HMP algorithm. In Proc. 52nd IEEE Conf. on Decision and Control, Florence, Italy, December 10-13, 2013.

[21] F. Taringoo and P.E. Caines. On the Optimal Control of Impulsive Hybrid Systems on Riemannian Manifolds. SIAM Journal on Control and Optimization, Vol. 51, Issue 4, pp. 3127 - 3153, 2013.

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

[23] 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.

[24] T. M. Caldwell and T. D. Murphey. Projection-based Switched System Optimization. Proc. American Control Conference, Montreal, Canada, June 2012.

[25] T. M. Caldwell and T. D. Murphey. Projection-Based Switched System Optimization: Absolute Continuity of the Line Search. Proc. 51st CDC, Maui, Hawaii, December 10-13, 2012.

[26] 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, pp. 466-481, Prague, The Czech Republic, 2003.

[27] P. Caines and M.S. Shaikh. Optimality Zone Algorithms for Hybrid Systems Computation and Control: Exponential to Linear Complexity. Proc. 13th Mediterranean Conference on Control and Automation, Limassol, Cyprus, pp. 1292-1297, June 27-29, 2005.

[28] 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.

[29] S. Hedlund and A. Rantzer. Optimal control for hybrid systems. Proc. 38th CDC, Phoenix, Arizona, December 7-10, 1999.

[30] 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.
[31] X. Ge, W. Kohn, A. Nerode, and J.B. Remmel. Hybrid systems: Chattering approximation to relaxed controls. *Hybrid Systems III: Lecture Notes in Computer Science*, R. Alur, T. Henzinger, E. Sontag, eds., Springer Verlag, Vol. 1066, pp. 76-100, 1996.

[32] R.T. Meyer, M. Zefran, and R.A. Decarlo. Comparison of the Embedding Method to Multi-Parametric Programming, Mixed-Integer Programming, Gradient Descent, and Hybrid Minimum Principle Based Methods. *IEEE Transactions on Control Systems Technology*, Vol. 22, no. 5, pp. 1784-1800, 2014.

[33] F. Zhu and P.J. Antsaklis. Optimal Control of Switched Hybrid Systems: A Brief Survey. Technical Report of the ISIS Group at the University of Notre Dame. ISIS-2011-003, July 2011.

[34] H. Lin and P. J. Antsaklis. Hybrid Dynamical Systems: An Introduction to Control and Verification. *Foundations and Trends in Systems and Control*, Vol. 1, no. 1, pp. 1-172, March 2014.

[35] L. C. Young. *Lectures on the calculus of variations and optimal control theory*. Foreword by Wendell H. Fleming. W. B. Saunders Co., Philadelphia, 1969.

[36] R. Gamkrelidze. *Principle of Optimal Control Theory*. Plenum, New York, 1978.

[37] L.D. Berkovitz and N.G. Medhin. *Nonlinear Optimal Control Theory*, Chapman & Hall, CRC Press, Boca Raton, Florida, 2013.

[38] H. Lou. Analysis of the Optimal Relaxed Control to an Optimal Control Problem. *Applied Mathematics and Optimization*, Vol. 59, pp. 75-97, 2009.

[39] R. Vinter. *Optimal Control*, Birkhauser, Boston, Massachusetts, 2000.

[40] D.Q. Mayne and E. Polak. First-order Strong Variation Algorithms for Optimal Control. *J. Optimization Theory and Applications*, Vol. 16, pp. 277-301, 1975.

[41] H. Jaleel, Y. Wardi, and M. Egerstedt. Minimizing Mobility and Communication Energy in Robotic Networks: an Optimal Control Approach. *Proc. 2014 American Control Conference*, Portland, Oregon, June 4-6, 2014.