Applications of Numerical Optimal Control to Nonlinear Hybrid Systems

Shangming Wei\textsuperscript{a}, Kasemsak Uthaichana\textsuperscript{b}, Miloš Žefran\textsuperscript{a}, Raymond A. DeCarlo\textsuperscript{b,\ast}, Sorin Bengea\textsuperscript{c}

\textsuperscript{a}Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, Illinois, USA

\textsuperscript{b}School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, USA

\textsuperscript{c}Innovation Center, Decision and Control Group, Eaton Corporation, Eden Prairie, Minnesota

Abstract

This paper develops a technique for numerically solving hybrid optimal control problems. The theoretical foundation of the approach is a recently developed methodology by Bengea & DeCarlo [1] for solving switched optimal control problems through embedding. The methodology is extended to incorporate hybrid behavior stemming from autonomous (uncontrolled) switches that results in the plant equations with piecewise smooth vector fields. We demonstrate that when the system has no memory, the embedding technique can be used to reduce the hybrid optimal control problem for such systems to the traditional one. In particular, we show that the solution methodology does not require mixed integer programming (MIP) methods, but rather can utilize traditional nonlinear programming techniques such as sequential quadratic programming (SQP). By dramatically reducing the computational complexity over existing approaches, the proposed techniques make optimal control highly appealing for hybrid systems. This appeal is concretely demonstrated in an exhaustive application to a unicycle model that contains both autonomous and controlled switches; optimal and model predictive control solutions are given for two types of models using both a minimum energy and minimum time performance index. Controller performance is evaluated in the presence of a step frictional disturbance which demonstrates the robustness of the controllers.

Key words: Hybrid systems, optimal control, numerical optimization, model predictive control

\ast Corresponding author.

Email addresses: swei3@uic.edu (Shangming Wei), uthaicha@purdue.edu

Preprint submitted to Nonlinear Analysis TMA 28 June 2006
1 Introduction

Hybrid systems have arisen as a powerful paradigm for describing systems that are characterized by a combination of continuous-valued and discrete-valued variables. A wealth of literature is available for the modeling and control of hybrid systems and we refer the reader to the overview articles [2, 3] and the books [4, 5]. A number of authors have also considered optimal control of hybrid systems. In general, the problem is quite hard as it involves both elements of optimal control as well as combinatorial optimization. The hybrid optimal control problems of [6], [7], [8], and [9] have led to generalizations of the Maximum Principle. For certain optimization problems, there exist numerically sound algorithms for obtaining suboptimal solutions. For example, one can use a hybrid version of the Bellman inequality, as in [10], or mixed integer programming (MIP), as in [11]. Industrial experience and simulations show that the performance of the solutions obtained by means of these methods are satisfactory, although these methods in general do not scale well. An algorithm based on a dynamic programming approach is proposed in [12, 13]. For a pre-assigned switching sequence (fixed number of switches and fixed mode sequence), a method for solving an optimization problem was described in [14], and for linear autonomous systems subject to a quadratic cost in [15]. A general setup uses viscosity solution techniques: the value functions associated with each mode of operation are shown to be solutions of a system of quasivariational inequalities in a weak sense; the optimal controls are then computed using PDE methods. Such an approach is used in [16] for solving an optimization problem associated with a switched system with time-invariant autonomous subsystems. A general optimal hybrid control problem which enables a direct Maximum Principle approach was presented in [17]. However, this work discusses neither sufficient conditions for optimality nor the singular cases.

The primary focus of this paper is the development of optimal and model predictive control (MPC) [18] solutions to the hybrid optimal control problem when the plant admits both autonomous and controlled switches. As such we investigate the numerical solution of the problem assuming the autonomous switches are memoryless. Specifically we develop an extension of embedding technique of Bengea and DeCarlo [1] for the solution of the hybrid optimal (switched) control problem to the case of piecewise smooth vector fields in the plant equations which result in the presence of memoryless autonomous switches.

(Kasemsak Uthaichana), mzefran@uic.edu (Miloš Žefran), decarlo@ecn.purdue.edu (Raymond A. DeCarlo), sbengea@ieee.org (Sorin Bengea).
It is generally perceived that the best numerical methods available for hybrid optimal control problems involve mixed integer programming (MIP). While great progress has been made in recent years in improving these methods, the MIP is an NP-hard problem so scalability is problematic. However, in our context, the solution methodology does not require a mixed integer programming problem (MIP) but rather can utilize traditional nonlinear programming techniques such as sequential quadratic programming (SQP). By dramatically reducing the computational complexity over existing approaches, the proposed techniques make optimal control highly appealing for hybrid systems. This appeal is concretely demonstrated in an exhaustive application to a unicycle model that contains both autonomous and controlled switches; optimal and model predictive control solutions are given for two types of models. Two performance indexes are set forth: a minimum energy and minimum time. Controller performance is evaluated in the presence of a step frictional disturbance which demonstrates the robustness of the controllers.

The paper is organized as follows. We start by presenting our modeling framework and introducing the terminology used in the paper. Section 3 formulates the optimal control problem for systems that undergo both autonomous and controlled switches. We show how to extend the method from [1] so that hybrid optimal control problem can be embedded as a traditional smooth problem. We then turn to a set of examples to demonstrate how to develop numerical methods for the resulting smooth problem. We conclude the paper with a short discussion and the possibilities for future work.

2 Mathematical Preliminaries

The hybrid systems studied in this paper will come from the class of switched systems. In particular, we are interested in control systems that exhibit two types of switching behavior, both resulting in discontinuous jumps in the vector fields governing the evolution of the continuous state of the system. First, we are interested in systems where the vector fields undergo discontinuous jumps as a result of the state and the input entering different regions in the combined state and input space. We call such switches autonomous or uncontrolled to indicate that the switches can not be effected directly through a separate switching mechanism. An example of a system with autonomous switches is a system subject to continuous state dependent constraints, where the autonomous switches correspond to different combinations of constraints that are active in a particular continuous state. The second type of switches involves discontinuous jumps in the vector fields that can be directly controlled, thus called the controlled switches. An example of a system with controlled switches is a continuous control system that can be controlled by a finite number of different continuous controllers, where the controller to be used is
determined by a supervisory controller. In this paper we assume that the set of different operating regimes of the system defined through the autonomous and controlled switches is finite.

In order to describe the evolution of a control system subject to controlled and autonomous switches we need four quantities: (i) the discrete state $\xi(t) \in D_\xi = \{1, 2, \ldots, d_\xi\}$, (ii) the continuous state $x(t) \in \mathbb{R}^n$, (iii) the discrete control input (switching control input) $v(t) \in D_v = \{1, 2, \ldots, d_v\}$, and (iv) the continuous control input $u(t) \in \mathbb{R}^m$. The discrete state of the system describes the autonomous switches. In this paper we only consider systems for which the autonomous switches depend on the continuous state $x(t)$ and the input $u(t)$, they do not depend on the current discrete state $\xi(t)$. Such systems are usually called memoryless. Formally, the evolution of the discrete state of the memoryless system is defined by a piecewise continuous\footnote{By piecewise continuous we mean a function that is continuous everywhere except on a finite union of switching surfaces that are smooth submanifolds of $\mathbb{R}^n \times \mathbb{R}^m$ with measure 0 where it undergoes discontinuous jumps, but has well defined limits in all directions.} function $\eta : \mathbb{R}^n \times \mathbb{R}^m \rightarrow D_\xi$, that for each continuous state $x$ and continuous control input $u$ selects the discrete state $\xi$ of the system:

$$\xi^+(t) = \eta(x, u). \quad (1)$$

Let $M_i \subseteq \mathbb{R}^n \times \mathbb{R}^m, i \in D_\xi$ be the set of pairs $(x, u)$ corresponding to the discrete state $i \in D_\xi$:

$$M_i = \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^m \mid \eta(x, u) = i\},$$

and let $f_{(i,j)}, i \in D_\xi, j \in D_v$ be a collection of $C^1$ vector fields

$$f_{(i,j)} : M_i \rightarrow \mathbb{R}^n.$$

The evolution of the continuous state $x(t)$ is then described by:

$$\dot{x}(t) = f_{(\eta(x(t), u(t)), v(t))}(x(t), u(t)), \quad x(t_0) = x_0. \quad (2)$$

At each $t \geq t_0$ and for each discrete state $\xi(t) \in D_\xi$, the switching control input $v(t) \in D_v$ thus selects which of the $d_v$ vector fields governs the evolution of the continuous state. We will assume that the continuous control input $u(t)$ is constrained to the convex and compact set $\Omega \subseteq \mathbb{R}^m$ and that the switching control input $v(t)$ and the continuous control input $u(t)$ are both measurable functions. Note that in this paper we restrict our attention to time-invariant systems, but the results can be easily generalized to time-varying systems.
Given that the discrete state $\xi(t)$ is completely determined by $x(t)$ and $u(t)$ through Eq. (1), we can define for each $j \in D_v$ a piecewise $C^1$ vector field $f_j$:

$$f_j(x(t),u(t)) \triangleq f_{\eta(x(t),u(t)),j}(x(t),u(t))$$

and rewrite Eq. (2) simply as:

$$\dot{x}(t) = f_{v(t)}(x(t),u(t)), \quad x(t_0) = x_0.$$  (4)

We note that the switching behavior described by Eq. (2) has a special structure since the switching control input $v(t)$ does not affect the autonomous switches. This means the vector fields $f_j$ all have the same set of points of discontinuity. We will thus refer to the systems described by Eq. (2) as systems with decoupled switches.

We are interested in computing optimal control laws for the system described by Eq. (2) or Eq. (4). If the system only undergoes autonomous switches ($D_v = \{1\}$) only the continuous input $u(t)$ needs to be computed. This suggests that the complexity of the optimal control problem might not be any different than in the traditional case. In contrast, for systems with controlled switches we need to compute the sequence of switching times $t_1, \ldots, t_n$ (including $n$), the sequence of discrete inputs $\nu_1, \ldots, \nu_n$, as well as the continuous input $u(t)$ on each interval $[t_i, t_i+1)$ for $i = 0, \ldots, n$. It would therefore appear that for systems with controlled switches the optimal control problem has combinatorial complexity. In this paper we show that actually both these cases have the same complexity and are amenable to traditional nonlinear programming techniques such as sequential quadratic programming (SQP). This further implies that for the systems considered in this paper the optimal control problem is no more complex than the traditional smooth problem.

In general, the behavior of switched systems can be quite complex and might lead to anomalies such as Zeno behavior or deadlock states. Furthermore, the systems in Eq. (2) or Eq. (4) belong to the class of systems with discontinuous right-hand sides [19] so the questions of existence and uniqueness of solutions have to be carefully studied. While certainly important, these issues are outside the scope of this paper and we will assume that the existence and uniqueness (in the appropriate sense) are guaranteed. We refer the interested reader to conference series [20,21], special issues [22–24], and [25,26] for further reading.

Similarly as before, by piecewise $C^1$ we mean a function that is $C^1$ everywhere except on a finite union of switching surfaces that are smooth submanifolds of $\mathbb{R}^n \times \mathbb{R}^m$ with measure 0 where the function is not differentiable and undergoes discontinuous jumps, but has well defined limits in all directions.
3 Optimal Control Problem

We now formulate the optimal control problem for hybrid systems described by Eq. (4). Given that the switching control input is completely arbitrary and independent from the continuous control, the search for the optimal solution can be seen as having three stages [9]: (i) finding the optimal sequence of control modes (the sequence of values of $v$); (ii) finding the optimal switching instants; and (iii) finding the optimal value for the continuous control. Effectively, (i) and (ii) constitute the search for $v(t)$, while (iii) is the search for $u(t)$ given $v(t)$. This partition of the hybrid optimal control problem is the approach pursued in [13–15]. Clearly, (i) results in the combinatorial complexity of the optimal control problem which makes finding the solution challenging at best. In [11] and the first author’s subsequent work, it was shown that mixed integer programming (MIP) can be employed to find the optimal solution. However, despite the availability of sophisticated methods for MIP, the computational complexity of hybrid optimal control problems still severely limits the application of these techniques in practice.

In this section we show that for quite a general class of hybrid optimal control problems, the computational complexity of the problem is no greater than that of smooth optimal control problems. In other words, the combinatorial aspect of the problem can be effectively eliminated, leading to a dramatic reduction in the overall computational cost. Our approach is based on the breakthrough result in [1], where it was shown that for a system with two discrete modes, the switched optimal control problem can be embedded into a larger family of systems where the switching function takes values in the interval $[0, 1]$ as opposed to the discrete set $\{0, 1\}$. The optimal solution of the embedded system is either the solution of the original problem, or can be approximated arbitrarily closely with a trajectory of the switched system. This essentially transforms the switched optimal control problem into a smooth problem that can be solved using traditional numerical methods. We show that the approach from [1] can be readily extended to systems with an arbitrary number of modes without further increase in complexity. Finally, we demonstrate the method on a nonlinear hybrid system with decoupled switches.

3.1 Problem Formulation

Consider the system described by Eq. (4). Both $v(t)$ and $u(t)$ are control variables and for the optimal control problem we require that they are chosen on the interval $[t_0, t_f]$ so that the following initial and terminal constraints are satisfied: $(t_0, x(t_0)) \in \mathcal{T}_0 \times \mathcal{B}_0$ and $(t_f, x(t_f)) \in \mathcal{T}_f \times \mathcal{B}_f$. We will assume that the endpoint constraint set $\mathcal{B} = \mathcal{T}_0 \times \mathcal{B}_0 \times \mathcal{T}_f \times \mathcal{B}_f$ is contained in a compact
set in $\mathbb{R}^{2n+2}$. We define the optimization functional

$$J_C(t_0, x_0, u, v) = g(t_0, x_0, t_f, x_f) + \int_{t_0}^{t_f} f_{\phi(x(t), u(t)), v(t)}(x(t), u(t)) dt, \tag{5}$$

where $g$ is a real-valued $C^1$ function defined on a neighborhood of $B$, and the functions $f_{(i,j)}^0: M_i \to \mathbb{R}, i \in D_\xi, j \in D_v$ are of class $C^1$. Given that the evolution of the discrete state $\xi(t)$ is governed by Eq. (1), we can define similarly as in Eq. (3) for each $j \in D_v$ a new piecewise $C^1$ function

$$f_j^0(x(t), u(t)) \triangleq f_{\phi(x(t), u(t)), j}(x(t), u(t)), \tag{6}$$

and rewrite the cost functional as:

$$J_C(t_0, x_0, u, v) = g(t_0, x_0, t_f, x_f) + \int_{t_0}^{t_f} f_{\phi(x(t), u(t)), v(t)}^0(x(t), u(t)) dt. \tag{7}$$

We now define the hybrid optimal control problem for systems with decoupled switches (HOCD):

$$\min_{v, u} J_C(t_0, x_0, u, v)$$

subject to the constraints: (i) $x(\cdot)$ satisfies Eq. (4); (ii) $(t_0, x(t_0), t_f, x(t_f)) \in B$; (iii) for each $t \in [t_0, t_f]$, $v(t) \in D_v$ and $u(t) \in \Omega$. Except for the fact that the vector fields in question are only piecewise $C^1$, this formulation is similar to that in [12, 17, 27] where the Maximum Principle is applied directly to the formulation. The formulation can be seen as a special case of the very general formulations of [9] and [6]. If the switching sequence were fixed, then the problem formulation would be similar to the one in [14], and if in addition the systems were linear and the cost functional quadratic, to that in [15].

3.2 Embedding

We now embed system (4) into a larger set of systems. For HOCD, $v(t) \in D = \{1, 2, \ldots, d_v\}$. Introduce $d_v$ new variables $v_i \in [0, 1], i \in D_v$, that satisfy

$$\sum_{i=1}^{d_v} v_i(t) = 1. \tag{8}$$

Let $u_i$ be the control input for each vector field $f_i, i \in D_v$, in (2). Now define a new system:

$$\dot{x}(t) = \sum_{i=1}^{d_v} v_i(t) f_i(x(t), u_i(t)), \quad x(t_0) = x_0, \tag{9}$$
and the associated cost functional

\[ J_E(t_0, x_0, u, v) = g(t_0, x_0, t_f, x_f) + \int_{t_0}^{t_f} \sum_{i=1}^{d_v} v_i(t) f_i^0(x(t), u_i(t)) dt. \tag{10} \]

Equations (9) and (10) are the generalization of those in [1]. The HOCD now becomes the embedded optimal control problem (EOC): minimize the functional (10) over all functions \( v_i \) and \( u_i \) subject to the following constraints: (i) \( x(\cdot) \) satisfies Eq. (9); (ii) \((t_0, x(t_0), t_f, x(t_f)) \in B\); (iii) for each \( t \in [t_0, t_f] \) and each \( i \in D_v \), \( v_i(t) \in [0, 1] \) and \( u_i(t) \in \Omega \); and (iv) for each \( t \in [t_0, t_f] \), \( \sum_{i=1}^{d_v} v_i(t) = 1 \).

As in [1] we note that EOC is now amenable to the classical necessary and sufficient conditions of optimal control theory [28]. Furthermore, it can be easily seen that the set of trajectories of the embedded system (9) contains the trajectories of the hybrid system (2). But what is crucial for solving the HOCD problem is the following observation.

**Proposition 1.** The set of trajectories of the hybrid system (2) is dense (in the \( L^\infty \) sense) in the set of trajectories of the embedded system (9).

**Proof.** The proof is analogous to the proof in [1] and proceeds through constructing a relaxation of the system (9), for which the trajectories of (2) are also dense. In this process it can be also derived how to explicitly find the trajectory of (2) that approximates arbitrarily closely a given trajectory of (9). \( \square \)

Because of the fact that the trajectories of (2) are dense in the set of trajectories of (9) and since the functions \( f_i^0 \) in (5) and (10) are \( C^1 \), we can state the following proposition.

**Proposition 2** (Bengea & DeCarlo [1]). If one of the optimal trajectories, \( x^*_E(\cdot) \), of the EOC has the property that \( x^*_E(t_f) \in \text{Int}(B_f)^3 \), then either (a) the HOCD has a solution, \( x^*_C(\cdot) \), which is also a solution of the EOC or (b) the HOCD does not have a solution and suboptimal admissible trajectories of the system (2) can be constructed.

The proposition implies that if one first solves the EOC and obtains a solution, either the solution is of the switched type (exactly one of the \( v_i \)’s is 1 and all the others are 0), or suboptimal trajectories of the HOCD can be constructed that can approach the value of the cost for EOC arbitrarily closely and satisfy the boundary conditions within \( \epsilon \) for arbitrary \( \epsilon > 0 \). Clearly, for practical applications this is an extremely valuable observation, given that traditional numerical methods can be used to solve EOC.

---

3 For a set \( B \), \( \text{Int}(B) \) denotes its interior.
Note also that for systems that only exhibit autonomous switches \((D_v = \{1\})\), no embedding is necessary and except for the fact that the functions \(f\) and \(f^0\) are only piecewise \(C^1\), the optimal control problem in this case is no different from the traditional formulation. This fact is often overlooked in the literature.

4 Numerical Method

In order to numerically solve the embedded optimal control problem formulated in Section 3.2, any numerical method appropriate for traditional optimal control can be used. We refer the reader to [29, 30] for a discussion of such methods. In our work, we use a variation of direct collocation [31]. In this case, \(u(t)\) and \(x(t)\) are chosen from finite-dimensional spaces. Given basis functions \(\{\phi_j\}_{j=1}^N\) and \(\{\psi_j\}_{j=1}^M\),

\[
\begin{align*}
x_i &= \sum_{j=1}^N p_i^j \phi_j(t), \quad p_i^j \in \mathbb{R}, \quad i = 1, \ldots, n, \\
u_i &= \sum_{j=1}^M q_i^j \psi_j(t), \quad q_i^j \in \mathbb{R}, \quad i = 1, \ldots, m.
\end{align*}
\]

Since \(f\) is only piecewise \(C^1\) and as a result \(x(t)\) can be nonsmooth, the basis functions \(\{\phi_j\}_{j=1}^N\) are chosen to be nonsmooth. Similarly, since the control \(u(t)\) can be discontinuous, the basis functions \(\{\psi_j\}_{j=1}^M\) are chosen to be discontinuous. We start by choosing a discretization of the time interval \([t_0, t_f]\)

\[
t_0 = t_1 < t_2 < \ldots < t_N = t_f.
\]

The state trajectory is approximated by a piecewise-linear function:

\[
\dot{x}_i(t) = x_i(t_j) + \frac{t - t_j}{t_{j+1} - t_j} (x_i(t_{j+1}) - x_i(t_j)), \quad t_j \leq t < t_{j+1}, \quad i = 1, \ldots, n.
\]

This approximation corresponds to the triangular basis functions:

\[
\phi_j(t) = \begin{cases} 
\frac{t-t_{j+1}}{t_{j+1}-t_{j-1}} & t_{j-1} \leq t < t_j, \\
\frac{t_{j+1}-t}{t_{j+1}-t_j} & t_j \leq t < t_{j+1}, \\
0 & \text{otherwise}.
\end{cases}
\]

The control input is chosen to be piecewise constant so that

\[
\dot{u}_i(t) = u_i(t_j), \quad t_j \leq t < t_{j+1}, \quad i = 1, \ldots, m.
\]
This approximation corresponds to the square basis functions:

$$
\psi^j(t) = \begin{cases} 
1 & t_j \leq t < t_{j+1}, \\
0 & \text{otherwise}.
\end{cases}
$$

The system equations are enforced at the midpoints:

$$
\dot{x}(t) - f(x(t), u(t)) = 0, \quad \text{for} \quad t = \frac{t_j + t_{j+1}}{2}, \quad j = 1, \ldots, N. \quad (11)
$$

If the original problem contains additional equality and inequality constraints they can be easily added in a similar way. With the chosen representation of $x$ and $u$, approximation of the integral (10) with a finite sum (using e.g. trapezoidal rule), and together with the equality constraints represented by Eq. (9) the optimal control problem thus becomes a nonlinear programming problem in the unknowns $p_i^j$ and $q_i^j$.

We use the optimization toolbox in Matlab to solve the resulting nonlinear programming problem. The methods for solving nonlinear programming problems such as those provided by Matlab typically require smoothness of the functions $f$ and $f^0$ to guarantee convergence. However, in our experience the fact that these functions are only piecewise smooth does not present any difficulties.

5 Examples

5.1 Autonomous switches

We first demonstrate the effectiveness of the proposed approach for systems with autonomous switches. In this case no embedding is necessary and we show that approximate solutions can be computed using standard nonlinear programming methods such as sequential quadratic programming (SQP), mixed integer programming (MIP) methods are not necessary. We present two examples. The first is taken from [11] where it was solved using a MIP package. It involves discrete state switches that only depend on the continuous state. In the second example, the switches are a function of the continuous state and the input.
Example 1. Consider the following system:

\[
x(t + 1) = 0.8 \begin{bmatrix}
\cos(\alpha(t)) & -\sin(\alpha(t)) \\
\sin(\alpha(t)) & \cos(\alpha(t))
\end{bmatrix} x(t) + \begin{bmatrix}
0 \\
1
\end{bmatrix} u(t),
\]

\[
\alpha(t) = \begin{cases} 
\frac{\pi}{3} & \text{if } x_1(t) \geq 0 \\
-\frac{\pi}{3} & \text{if } x_1(t) < 0
\end{cases},
\]

\[u(t) \in [-1, 1].\]

The task is to transfer the state from \(x_0 = [-1 \ 1]^T\) to the origin so that the following cost is minimized:

\[
J_A(x_0, u) = \sum_{k=1}^{N} \left[ q \|u\|^2 + \|x\|^2 \right].
\]

Note that the system in the example is a discrete-time system so the state equation can be used directly in the discretization in place of Eq. (11). The results of optimization for \(N = 5\) and for three different values of \(q\) are shown in Figure 1. The method efficiently finds the optimal solution, on a 2GHz PC it takes only about 1s and 10 iterations to converge, starting with a random initial guess.

![State trajectory](a) State trajectory. ![Control input](b) Control input.

Fig. 1. Optimal solution for Example 1 for different values of the weight \(q\).

The next example is more demanding as it involves more variables, more complicated dynamics, and more complicated switching behavior. Furthermore, the system has continuous-time dynamics, requiring a fine grid in order to enforce the differential equations with sufficient precision.
**Example 2.** Consider the following system with 4 states and 2 inputs:

\[
\dot{x}(t) = \begin{cases} 
A_1 x(t) & 9 < x_1^2 + x_2^2 + \|u\|^2 \leq 100 \\
A_2 x(t) + B_2 u(t) & \text{otherwise}
\end{cases},
\]

where

\[
A_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\
0 \\
1 \\
0 \end{bmatrix}.
\]

The task is to bring the system to the origin in 1s so that the following cost functional is minimized:

\[
J_A(x_0, u) = \int_0^1 \|u\|^2 dt.
\]

It is worth noting that the surface on which the system switches is nonlinear and it depends both on the state and the input. The system can be interpreted as a model for a mass particle sliding on a horizontal plane and controlled by two thrusters. The particle has to pass through a repulsive region where the thrusters have no effect on the motion before approaching the origin.

For the computation, we chose to discretize the system on a uniform grid with 100 points. Due to a large number of variables the optimization took around 10 minutes to converge on a 2GHz PC. We started with a random initial guess and computed the optimal solution on a grid with 10 points. This solution was subsequently used as the initial guess for a grid with 20, 40, and finally 100 points, the solution from each step serving as the initial guess for the next.

The results of the optimization for \(x_0 = [6 \ -12 \ -1 \ 2]^T\) are shown in Figure 2. The optimal behavior of the system is not unexpected: the velocity

![State trajectory](image1.png) ![Control input](image2.png)

(a) State trajectory. (b) Control input.

**Fig. 2.** Optimal solution for Example 2 for \(x_0 = [6 \ -12 \ -1 \ 2]^T\).
of the mass particle is increased before entering the repulsive region so that
the particle can slide through it. In the repulsive region the thrusters have
no effect so they are turned off to reduce the cost. After the particle leaves
the repulsive region, the thrusters become active and bring the particle to the
origin.

5.2 Decoupled switches: Unicycle Application

To demonstrate the full power of the approach in Section 3 we will show
how to compute optimal trajectories for a system exhibiting both controlled
and autonomous switches. The example we will use is a unicycle driving on a
horizontal plane (Figure 3). The wheel of the unicycle can either roll or slide,
resulting in autonomous switches. In addition, we assume that the unicycle
has a regenerative brake that can be turned on or off. These switches are
controlled. We will consider two different models of the unicycle. The first
model is for the unicycle with a separate motor and a generator both connected
to a battery-pack; the other model assumes that the unicycle contains only
one electric drive-battery-pack that performs as either a motor or a generator.
The essential features of each model are listed in Table 1 which also indicates
the designated modes.

Table 1
Characteristics of the two models

<table>
<thead>
<tr>
<th>Quantities</th>
<th>Model A</th>
<th>Model B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of electric machines</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Mode-1 denotation</td>
<td>Brake-off</td>
<td>Propelling</td>
</tr>
<tr>
<td>Mode-2 denotation</td>
<td>Brake-on</td>
<td>Regenerative Braking</td>
</tr>
<tr>
<td>Torque $u_1$ in Mode-1</td>
<td>Propelling and Braking Torque</td>
<td>Propelling Torque Only</td>
</tr>
</tbody>
</table>

Referring to Figure 3, the forward velocity of the wheel is controlled by the
torque $u_1$, while its heading is controlled by the torque $u_2$. Our objective is to
formulate and solve two hybrid optimal control problems: (i) drive the unicycle
to the origin within an allotted time while minimizing weighted power usage
and (ii) drive the unicycle from the same initial condition as in problem (i) to
the origin in minimum time. As discussed earlier, we will use model predictive
control (MPC) [18] to compute the control inputs for the embedded system.
The generalized coordinates for the unicycle are its center of mass position \( x \) and \( y \), body orientation \( \theta \) (heading) relative to the \( x \)-axis, and the rotation of the wheel \( \phi \). Since we are not interested in \( \phi \) itself, the state variables for the system are \( z^T = [x, y, \theta, v_x, v_y, \dot{\theta}, \dot{\phi}]^T \in \mathbb{R}^7 \), where \([v_x, v_y]\) is the velocity of the center of mass of the unicycle, expressed in the body frame, \( \dot{\theta} \) is the turning velocity of the unicycle, and \( \dot{\phi} \) is the angular velocity of the wheel as it spins on its axle. For either model the equations of motion for the unicycle take the form:

\[
\begin{align*}
    \dot{x} &= v_x \cos(\theta) - v_y \sin(\theta), \\
    \dot{y} &= v_x \sin(\theta) + v_y \cos(\theta), \\
    \dot{\theta} &= \dot{\theta}, \\
    \dot{v}_x &= F_x(z) + \dot{\theta} v_y, \\
    \dot{v}_y &= F_y(z) - \dot{\theta} v_x, \\
    \ddot{\theta} &= \frac{1}{I_2} u_2, \\
    \ddot{\phi} &= \frac{F_x r}{I_1} + \frac{1}{I_1} u_1, 
\end{align*}
\]  

where (i) \( m \) is the mass of the unicycle, (ii) \( r \) is the radius of the wheel, (iii) \( I_1 \) is the moment of inertia of the wheel around its axis, (iv) \( I_2 \) is the moment of inertia of the unicycle about the vertical axis through the center of mass, and (v) \( F_x \) and \( F_y \) are the forces between the ground and the wheel in the forward and lateral directions, respectively. As discussed above, \( u_1 \) is the control torque applied to the axle of the wheel, while \( u_2 \) is the torque controlling the heading of the unicycle. The structure of \( u_1 \) will differentiate the model types and is described later.

The distinctive hybrid nature of the unicycle occurs because the forces \( F_x(z) \) and \( F_y(z) \) depend on whether the unicycle is rolling or sliding. In the rolling
mode, the relative velocity \( v_r \triangleq [v_{rx}, v_{ry}]^T = [v_x + \dot{\phi}r, v_y]^T \), between the ground and the wheel’s point of contact is zero, i.e.,

\[
\begin{align*}
v_{rx} &= v_x + \dot{\phi}r = 0, \\
v_{ry} &= v_y = 0.
\end{align*}
\]  

From Eqs. (13) and (12) we have that

\[
\begin{align*}
F_x(z) &= -\frac{mr u_1}{mr^2 + I_1} \\
F_y(z) &= -mr \dot{\theta} \dot{\phi}.
\end{align*}
\]  

Here, the forces \( F_x(z) \) and \( F_y(z) \) are ground reaction forces that prevent wheel slippage. In the sliding mode, \( F_x(z) \) and \( F_y(z) \) are frictional forces. In summary the forces \( F_x \) and \( F_y \) are given by the following expressions

\[
\begin{align*}
F_x(z) &= \begin{cases} 
-\frac{mr u_1}{mr^2 + I_1} & \text{rolling mode} \\
-\mu d \frac{v_x + \dot{\phi}r}{\|v_r\|} mg & \text{sliding mode}
\end{cases} \\
F_y(z) &= \begin{cases} 
-mr \dot{\theta} \dot{\phi} & \text{rolling mode} \\
-\mu d \frac{v_y}{\|v_r\|} mg & \text{sliding mode}
\end{cases}
\]  

where \( \mu_d \) is the coefficient of dynamic friction and \( g \) is the gravitational constant.

The autonomous switch from rolling to sliding occurs when the magnitude of the constraint force \( F \triangleq [F_x, F_y]^T \) exceeds the maximum possible magnitude of the static friction, \( \mu_s mg \), i.e.,

\[
\|F\| > \mu_s mg \Rightarrow \text{rolling} \rightarrow \text{sliding},
\]  

where \( \mu_s \) is the coefficient of static friction. On the other hand, the switch from sliding to rolling occurs when (i) \( v_r = [v_{rx}, v_{ry}]^T = 0 \), and (ii) the maximum magnitude of the frictional force exceeds that of the constraint force \( F = [F_x, F_y]^T \), i.e.,

\[
\|v_r\| = 0 \quad \text{and} \quad \|F\| \leq F_{s,\text{max}} = \mu_s mg \Rightarrow \text{sliding} \rightarrow \text{rolling}.
\]  

Therefore, the system has two kinds of switches:

1. Autonomous switches where the system switches between rolling and sliding depending on the physics of the contact.
2. Controlled switches where the regenerative brake can be switched on or off arbitrarily.

Let Model A denote the configuration of a separate motor and generator.
Mode 1 will denote the use of

\[ u_1 = u_{1A}^1 \in [-20, 20] \]  \hspace{1cm} (19)

as an actuating torque that can be either propelling or braking. Mode 2 denotes the use of regenerative braking alone in which case

\[ u_1 = u_{1A}^2 = \begin{cases} -K_b \dot{\phi}, & |\dot{\phi}| \leq 2 \\ -20 \text{sgn}(\dot{\phi}), & |\dot{\phi}| > 2 \end{cases} \]  \hspace{1cm} (20)

where \( K_b = 10 \) is a fixed regenerative braking coefficient; note that \( u_1 \) saturates at 20Nm.

Let Model B denote the configuration of a single electric drive that can operate either as a (propelling) motor or as a generator for regenerative braking. For model B, the control input

\[ u_1 = u_{1B}^1(t) \text{sgn}(\dot{\phi})u_{1\text{max}} \]  \hspace{1cm} (21)

strictly provides propelling torque in Mode 1, where \( u_{1\text{max}} = 20 \text{Nm} \), \( u_{1B}^1(t) \in [0, 1] \) modulates \( u_{1\text{max}} \), and \( \text{sgn}(\dot{\phi}) \) insures that the applied torque is always propelling. The regenerative braking term for Mode 2 coincides with that used for Model A, i.e.,

\[ u_1 = u_{1B}^2 = \begin{cases} -K_b \dot{\phi}, & |\dot{\phi}| \leq 2 \\ -20 \text{sgn}(\dot{\phi}), & |\dot{\phi}| > 2 \end{cases} \]  \hspace{1cm} (22)

Note that for each model, Modes 1 and 2 are possible in both the rolling and sliding modes. For each model the unicycle is thus an example of a system with decoupled switches given by Eq. (2).

### 5.2.2 Embedded Model Formulation

We now follow the procedure outlined in Section 3.2. For simplicity, let \( v_1 = \alpha \) and \( v_2 = 1 - \alpha \) and define for both models

\[ u_1(t) = (1 - \alpha(t)) u_{1}^1(t) + \alpha(t) u_{1}^2(t) \]  \hspace{1cm} (23)

The variable \( \alpha(t) \in \{0, 1\} \) denotes the original problem whereas \( \alpha(t) \in [0, 1] \) denotes the embedded problem which will be solved in this investigation. Thus, the embedded formulation is given by Eqs. (12) and (23) with the specific structures of Eqs. (19)-(22) from Models A and B respectively and the forces \( F_x(z) \), and \( F_y(z) \) of the unicycle motion in rolling and sliding given by Eqs. (15) and (16) respectively. Because of the autonomous switches determined by Eqs. (17) and (18), the right hand side of Eq. (12) is piecewise continuous.
provided the system does not chatter about the boundary implicitly defined by Eqs. (17) and (18).

5.2.3 The Performance Index and MPC Design

5.2.3.1 Fixed Time Control. The first objective of the control design for each model is to drive the unicycle from a given starting initial state, $z^T_0 = [x(0), y(0), \theta(0), v_x(0), v_y(0), \theta(0), \phi(0)]^T$ back to the origin while minimizing the energy usage. In addition, since sliding of the wheel is undesirable since it implies the loss of controllability, we would like to limit the sliding motion of the unicycle. Hence, the integral performance index for this study is taken as

$$J = c_0 \|z(T)\|^2 + \int_0^T \left[ c_1 (1 - \alpha) u_1^2 + c_2 u_2^2 + c_3 \|v_r\|^2 \right] dt \quad (24)$$

where the positive weights $c_i$ (for $i = 0, \ldots, 4$) are constant. The term (i) $c_0 \|z(T)\|^2$ drives the final states of the unicycle $z(T)$ toward the origin; (ii) $c_1 (1 - \alpha) u_1^2$ penalizes the actuating power usage; (iii) $c_2 u_2^2$ penalizes the heading power usage; and (iv) $c_3 \|v_r\|^2$ is to limit sliding motion. The terminal constraints are enforced through the cost functional (as soft constraints) rather than imposed as hard constraints because the system is stabilizable but not controllable in the sliding regime. Therefore, using hard constraints could make the optimal control problem unfeasible. Note also that there is no penalty for regenerative braking usage.

The first objective thus leads to minimization of the performance index in Eq. (24) subject to the embedded state dynamics given by Eqs. (12)-(23) and an initial state $z_0$, resulting in an optimal control input for the nominal dynamical system. However, when applying the computed controls to the actual model that differs from the nominal model due to the presence of disturbances or modeling uncertainties, the state trajectory might deviate from the desired trajectory, and fail to reach the desired final state within the allotted time interval. To cope with such disturbances and uncertainties, an MPC-type controller that is well-known for its robustness will be adopted to drive the unicycle from a given initial state $z_0$ to the origin at a pre-specified final time. The adopted optimal control based MPC approach can be summarized in steps as follows:

1. Given $z_0$, partition the time interval $T$ into $N$ equal subintervals of length $h = \frac{T}{N}$, for the purpose of computing a (backward) piecewise constant control sequence $\{\hat{u}_1, \ldots, \hat{u}_N\}$, where $\hat{u}_i = \left[ u_1(ih) \ u_2(ih) \right]^T$, and the state values $\{z_1, \ldots, z_N\}$.

4 An analysis of the vector fields on the switching surface shows that chattering can not occur.
(2) For \( k = 1, \ldots, N \), solve the embedded problem of the unicycle over the receding horizon \([k, N]\) by minimizing the performance index given by Eq. (24) subject to the nominal model with the initial state \( z_{k-1} \) and obtain the (look ahead) control sequence \( \hat{u}_k, \ldots, \hat{u}_N \).

(3) Apply the control input \( \hat{u}_k \) for the time interval \( t_{k-1} \leq t < t_k \) to the real model. The value of the state of the real model at the end of the interval becomes \( z_k \), the initial condition for the next iteration.

(4) Repeat steps 2 and 3 until \( k = N \).

A variation of the direct collocation method from Section 4 is used to numerically solve the embedded optimal control problem at each step of the MPC algorithm. As per development in Section 4, the discretized control input is assumed to be piecewise constant function:

\[
\hat{u}(t) = u(t_j), \quad t_j \leq t < t_{j+1}, \quad i = 0, \ldots, N - 1.
\]

However, unlike in Section 4, since the switching between sliding and rolling requires precise solution of the state equations, the state equations are integrated directly for the given control variables using Runge-Kutta ODE solver.

5.2.3.2 Time Optimal Control. The second objective of this work is the construction of a minimum time control. Given the same initial point, \( z_0^T = [x(0), y(0), v_x(0), v_y(0), \theta(0), \dot{\theta}(0), \dot{\phi}(0)]^T \), drive the state to the origin in minimum time. The performance index in this case is

\[
J = c_0 \|z(t_f)\|^2 + \int_0^{t_f} 1 \, dt,
\]

where \( c_0 = 2 \cdot 10^4 \) and \( t_f \) is the unknown final time. As before, the terminal state constraint is soft in that we only penalize the deviation from the origin.

Controller construction entails a discretization which cannot be done directly since the final time is unknown. By letting \( t = t_f s \), an equivalent version of performance index (25) is obtained as

\[
\min_{\hat{u}, \alpha, t_f} c_0 \|z(1)\|^2 + \int_0^1 t_f \, ds
\]

where \( s \in [0, 1] \) is normalized time and \( t_f \) is an unknown constant. With this formulation the same method as above can be used to solve the optimization problem.

Some variation is needed for the MPC strategy. Initially, the interval \([0, 1]\) is partitioned into \( N \) subintervals of width \( h = \frac{1}{N} \). After computing \( \hat{u}_1, \ldots, \hat{u}_N \)
over the interval \( s \in [0, 1] \) and \( t_f \), and applying \( \hat{u}_1 \) over the time interval \([0, ht_f]\) one generates a new minimum time problem for the interval \( s \in [h, 1] \) with the new partition length of \( N - 1 \). In general, in iteration \( j + 1 \) one must solve the minimum time problem over the interval \( s \in [jh, 1] \) and in each iteration a new \( t_f \) is computed. This implies that the final grid is not uniform in time, \( t_{j+1} - t_j \neq \text{const}. \)

5.2.4 Simulation Results and Discussion

This subsection contains the simulation results of the investigation. In all cases the initial condition of the unicycle was \( z_0^T = [x(0), y(0), \theta(0), v_x(0), v_y(0), \dot{\theta}(0), \dot{\phi}(0)]^T = [0, 4, 0, 0, 1, 0, 1]^T \) and the number of points for the discretization was \( N = 20 \). Nominal trajectories are for the model without any disturbances. Comparisons are made to the MPC controlled process with a frictional disturbance. In all cases the embedded problem has a bang-bang solution meaning that it is also a solution to the original hybrid optimal control problem.

The first set of plots shows the responses for Model A with the minimum energy performance index of Eq. (24). The computed cost of the nominal optimal trajectory is 972.2 versus 1036.3 for the disturbed MPC controlled trajectory.
The nominal response is for the model without disturbances whereas the MPC control is for the frictional disturbance around the origin the static coefficient of friction $\mu_s$ drops from 0.7 to 0.002, and the dynamic coefficient of friction $\mu_d$ drops from 0.6 to 0.001. Moreover, the unicycle’s nominal parameters $m_{\text{nominal}} = 1$ and $r_{\text{nominal}} = 4$ during the simulation were perturbed so that $m_{\text{actual}} = 1.05$ and $r_{\text{actual}} = 3.9$. After leaving the slippery area at about 6s, in conjunction with the corrective action of the MPC controller, the unicycle starts to roll again. Figure 4(a) and Figure 4(b) show the unicycle’s trajectories and the evolutions of two states $v_x$ and $v_y$. It clearly shows that the unicycle can still reach the origin in the required time despite disturbances and model errors. Figure 4(d) displays the control inputs which again adapt in accordance with state and model changes. The results thus confirm that the influence of disturbances on system performance is small and the MPC scheme achieves good performance and robustness. Recall that for this model, the electric motor can apply both a propelling torque and a braking torque $u_1^1$ in Mode 1, while a regenerative braking torque $u_2^1$ is applied in Mode 2. One observes that during the final 4s of the simulation, the switches of the braking torque in Mode 1 and the regenerative braking torque in Mode 2 are coordinated to reduce cost and reach the origin on time.

Figure 5 shows the response of Model B to the same initial condition and the same disturbances. The cost of the nominal optimal trajectory is 997 versus 1000 for the disturbed MPC controlled trajectory. As for Model A, around 4s the unicycle encounters the frictional disturbance and switches from rolling to sliding as indicated in Figure 5(c), locally increasing the cost. One notes that toward the end of the trajectory, there is a small deviation from the optimal undisturbed trajectory; this results because (i) the magnitude of the velocity of the unicycle leaving the slippery area is greater for the MPC controlled case, and (ii) the only mechanism for stopping is regenerative braking. Hence, there must be an appropriate path length increase to reach the origin at the desired time. Indeed, because for Model B there is only regenerative braking, it takes longer to reach the origin, from 9.5s (Model A) to about 15s (Model B). As will be seen later, the minimum time trajectory takes about 11.1s to reach the origin. Again one observes that the MPC control performs quite well relative to the nominal, disturbance free optimal trajectory.

Our final set of plots in Figure 6 deals with the minimum time performance index of Eq. (25) for Model B. The cost consists of the traveling time and a terminal penalty on the deviation of the state from the origin. The nominal cost is 16.2 versus 15.0 for the disturbed MPC controlled trajectory. While unexpected, the lower cost for the MPC controlled system is a result of a high
weight \((c_0 = 2 \cdot 10^4)\) on the terminal penalty. The actual traveling time for the nominal case is 11.1s vs. 11.2s for the MPC control. Interestingly, due to lower effective inertia, the optimal traveling time for the model and the environment used for the MPC control is 11s, less than that for the nominal system.

Figure 6(c) shows that initially, the wheel is in the sliding state. Kinetic friction associated with the sliding motion decreases the magnitude of the angular velocity, \(|\dot{\phi}|\), and simultaneously increases the forward velocity \(v_x\) of the unicycle. This decreases the relative velocity, \(v_r\), until it becomes zero, the condition for rolling that results by the end of the first control window. However, we note that by the end of the first control window, the forward velocity \(v_x\) is higher than in the above two cases because the control chooses to propel the unicycle to achieve minimum time, rather than lower the energy usage. Indeed in the first two cases, the control brakes to avoid the high penalty on sliding in the performance index.

At 3s the unicycle enters the slippery area and begins to slide. During sliding \(\dot{\phi}\) is independent of the forward velocity \(v_x\). Since the kinetic friction has an extremely small magnitude, \(v_x\) remains relatively constant, as it does in the previous two cases, while due to braking \(\dot{\phi}\) drops at a faster rate than before.
Finally, faced with the longer path in addition to the (soft) terminal constraint, the MPC makes a "no cost" abrupt change in the steering wheel during the last few control windows to achieve the terminal constraint. Indeed, the MPC control which uses a receding horizon is more sensitive to the terminal constraint during the last few control windows as seen in Eq. (26).

6 Conclusion

The paper studies applications of numerical optimal control to nonlinear hybrid systems. While the formulation of the optimal control problem for hybrid systems depends on the nature of the mode switches of the system, we show that in contrast to a widely accepted belief that such problems always lead to mixed integer programming problems (MIPs), in many important cases traditional nonlinear programming techniques such as sequential quadratic programming (SQP) can be readily applied. In particular, we demonstrate that when the switches are autonomous (uncontrolled) the optimal control problem can be solved using traditional techniques. Furthermore, while controlled switches can possibly lead to a combinatorial complexity of the numerical al-
gorithms, we show that an extension of techniques recently developed in [1] can again effectively reduce such problems to the traditional one. By dramatically reducing the computational complexity over existing approaches, the proposed techniques make optimal control highly appealing for hybrid systems, potentially leading to a much broader appeal of hybrid system models. As an application, we show in detail how the proposed techniques can be applied to a unicycle model that contains both autonomous and controlled switches.

Acknowledgment

The authors thank Dr. Michael Branicky for his comments on an earlier version of this paper. The work of S. Wei and M. Žefran has been supported in part by NSF grants IIS-0093581 and CCR-0330342.

References


