Sequential Linear Quadratic Optimal Control for Nonlinear Switched Systems

Farbod Farshidian ∗ Maryam Kamgarpour ∗∗ Diego Pardo ∗
Jonas Buchli ∗

Abstract: In this contribution, we introduce an efficient method for solving the optimal control problem for an unconstrained nonlinear switched system with an arbitrary cost function. We assume that the sequence of the switching modes are given but the switching time in between consecutive modes remains to be optimized. The proposed method uses a two-stage approach as introduced by Xu and Antsaklis (2004) where the original optimal control problem is transcribed into an equivalent problem parametrized by the switching times and the optimal control policy is obtained based on the solution of a two-point boundary value differential equation. The main contribution of this paper is to use a Sequential Linear Quadratic approach to synthesize the optimal controller instead of solving a boundary value problem. The proposed method is numerically more efficient and scales very well to the high dimensional problems. In order to evaluate its performance, we use two numerical examples as benchmarks to compare against the baseline algorithm. In the third numerical example, we apply the proposed algorithm to the Center of Mass control problem in a quadruped robot locomotion task.

Keywords: Control design for hybrid systems, Switching stability and control, Optimal control of hybrid systems, Optimal control theory, Real-time control, Riccati equations, and Mobile robots.

1. INTRODUCTION

Switched systems are a subclass of a general family known as hybrid systems. Hybrid system model consists of a finite number of dynamical subsystems subjected to discrete events which cause transition between these subsystems. This transition is either triggered by an external input, or through the intersection of the continuous states trajectory to a certain manifolds known as the switching surfaces. Switched systems are usually characterized by systems that have continuous state transition during these switches. Switched system models are encountered in many practical applications such as automobiles and locomotives with different gears, DC-DC converters, manufacturing processes, biological systems, and robotics.

Our interest in switched systems originates from an application on a legged robot where we model the Center of Mass (CoM) as a switched system. The control goal is to synthesize a controller which, for a given gait, stabilizes the robot while it minimizes a cost function. To fulfill this task, the robot can manipulate the ground reaction forces at the stance feet and adjust the switching times between different stance leg configurations. For instance, assume the problem of controlling the walking gait for a quadruped robot. In this task, the gait is fixed, thus the sequence of mode switches are known. The control task is to modulate the contact forces of the stance legs and to determine the switching times between each mode.

The optimal control problem for the switched systems involves synthesizing the optimal controller for the continuous inputs and finding a mode sequence and the switching times between the modes. In general, the procedure of synthesizing an optimal control law for a switched system can be divided into three subtasks (Giua et al., 2001; Xu and Antsaklis, 2004): (1) finding the optimal sequence of the modes, (2) finding the optimal switching times between consecutive modes, (3) finding the optimal continuous control inputs. Given the switching sequence and times, the third subtask is a regular optimal control problem with a finite number of discontinuities in the system vector field. The necessary condition of optimality in the context of hybrid systems has been derived from Pontryagin’s maximum principle (Branicky et al., 1998; Sussmann, 1999; Riedinger et al., 2003) and subsequently, various computational techniques have been developed to solve this problem (Shaikh and Caines, 2007; Soler et al., 2012; Pakniyat and Caines, 2014).

Based on the Pontryagin’s maximum principle, the optimal solution should satisfy a two-point boundary value problem (BVP). However, similar to the classical control problem, the difficulties related to numerical solution of the necessary condition of optimality limits the application of this approach. In Riedinger et al. (1999), it has been shown that for a Linear-Quadratic (LQ) problem it is sufficient to solve a sequence of Riccati equations with proper transversality conditions at the switching times in order to optimize the continuous inputs but the mode switches should be still calculated based on the enumerations of all the possible switches at each time step. In order to ease the computational burden of finding the optimal switching behavior, in Bengea and DeCarlo (2005), the switched...
system is embedded in a larger family of systems defined by the convex hull of the switched subsystems. It has been shown that if the sufficient and necessary conditions for optimality in the embedded systems exist, the bang-bang optimal solution of the embedded problem is also the optimal solution of the switched system; otherwise a sub-optimal solution can be derived.

Borrelli et al. (2005) propose an off-line method to synthesize an optimal control law for a discrete linear hybrid system with linear inequality constraints. The proposed method is a combination of dynamic programming and multi-parametric quadratic programming which designs a feedback law for continuous and discrete inputs in the feasible regions. A simpler approach in Bemporad and Morari (1999) uses a mixed integer linear/quadratic program to solve the optimal control problem for mixed logical dynamical systems.

Optimizing the cost function with respect to the switching times has been studied for autonomous systems by Egerstedt et al. (2003); Johnson and Murphey (2011); Wardi et al. (2012) and for non-autonomous systems by Kamgarpour and Tomlin (2012). By using the derivative of the cost function with respect to the switching time, these methods use nonlinear programming techniques to optimize the cost function. However, in general these methods do not consider the sensitivity of the continuous inputs’ control law to the switching times.

While many of the aforementioned approaches are computationally demanding for real-time robotic applications, there is a class of efficient optimal control algorithms known as Sequential Linear Quadratic (SLQ) methods which can be applied to real-time, complex robotic applications (Neunert et al., 2016). An SLQ algorithm sequentially solves the extremal problem around the latest estimation of the optimal trajectories and improves these optimal trajectories using the extremal problem solution (Mayne, 1966; Todorov and Li, 2005; Sideris and Bобрю, 2005). Motivated by their efficiency in solving regular optimal control problems, in this paper we have extended an SLQ algorithm to solve the optimal control problem for nonlinear switched systems with predefined mode sequence. To this end, we adopt an approach introduced by Xu and Antsaklis (2004) where the primary switching problem is transcribed into an equivalent problem. We then introduce a two-stage optimization method to optimize the continuous inputs and the switching times. While Xu and Antsaklis (2004) use a computationally demanding approach based on solving a set of two-point BVPs, we propose a new SLQ algorithm to efficiently solve the optimal control problem for nonlinear switched systems.

The main contributions of this paper are: (1) it uses an efficient SLQ algorithm to synthesize the optimal control law for the continuous inputs. (2) it calculates the cost function derivative with respect to switching times using an LQ approximation of the problem. This approximation is obtained without any additional computation from the SLQ solution. (3) it introduces a new practical application of the optimal control for switched systems in the field of motion planning of legged robots.

2. PROBLEM FORMULATION

In this section, we briefly introduce the optimal control problem based on the parameterization of switching times. We assume that the switched system dynamics consist of \( I \) subsystems where the system dynamics for the \( i \)th subsystem \((i \in \{1, 2, \ldots, I\})\) is as follows

\[
\dot{x}(t) = f_i(x(t), u(t)) \quad \text{for} \quad t_{i-1} \leq t < t_i, \tag{1}
\]

where \( x(t) \in \mathbb{R}^n \) is the continuous state, \( u(t) \in \mathbb{R}^m \) is the piecewise continuous control input, and \( f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R} \) is the vector field of subsystem \( i \) which is continuously differentiable and Lipschitz up to the first order derivatives. \( t_i \) is the switching time between subsystem \( i \) and \( i + 1 \). \( t_0 \) and \( t_I \) are respectively the given initial time and the final time. The initial state is \( x_0 \) and \( x(t_f) = x(t_f^+) \) at the switching moments because of the state continuity condition. The optimal control problem for the switched system in Equation (1) is defined as

\[
\min_{t \in \mathbb{T}, u(t)} \Phi(x(t_1)) + \sum_{i=1}^{I} \int_{t_{i-1}}^{t_i} L_i(x, u) dt, \tag{2}
\]

where \( \Phi(\cdot) \) and \( L_i(\cdot, \cdot) \) are the final cost and the running cost (for subsystem \( i \)) which are continuously differentiable and Lipschitz up to the second order derivatives. \( \mathbb{T} \) is a polytope in \( \mathbb{R}^{I-1} \) defined as \( \mathbb{T} = \{ (t_1, \ldots, t_{I-1}) | t_0 \leq t_1 \leq \cdots \leq t_{I-1} \leq t_I \} \).

The optimal control problem based on the parameterization of switching times can be defined as the following two-stage optimization problem

\[
\min_{t \in \mathbb{T}} J[t, x^*, u^*] \quad \text{s.t.} \quad \{ x^*, u^* \} = \text{argmin}_{t \in \mathbb{T}} J[t, x, u], \tag{3}
\]

with

\[
J[t, x, u] = \Phi(x_1) + \sum_{i=1}^{I} (t_i - t_{i-1}) \int_{t_{i-1}}^{t_i} L_i(x(z), u(z)) dz, \tag{4}
\]

\[
\frac{dz}{z} = (t_i - t_{i-1}) f_i(x(z), u(z)) \quad \text{for} \quad i \leq 1 \leq i \leq I \tag{5}
\]

\[
t = (t_i - t_{i-1}) (z - i) + t_i, \tag{6}
\]

in which we have replaced the independent time variable \( t \) with a normalized time variable \( z \) defined by (6). With this change of variable, while the switching times are still part of the decision variables, they are fixed parameters for the bottom-level optimization in (3). Therefore, the reformulated optimal control problem does not have variable switching times and it reduces to a conventional optimal control problem parameterized over the switching times (Theorem 1 in Xu and Antsaklis (2004)). In the next section, we introduce our algorithm for calculating the optimized cost function and its gradient.

3. SOLUTION APPROACH

In this section, we use a gradient-based method to solve the top-level optimization introduced in (3) and a dynamic programming approach to synthesize the continuous inputs control law. In each iteration of the gradient-based method for finding the optimal switching time, we first solve a continuous-time optimal control problem for the system with a fixed switching times. Then, the cost function gradient with respect to the switching times is calculated in order to determine the descent direction for the switching times update. This approach is similar to Xu and Antsaklis (2004). However, instead of using the two-point BVP solver for optimizing \( J(t) \) with respect to \( u \) and calculating its gradient with respect to \( t \), we use a more efficient approach. This facilitates implementing this algorithm in real-time on complex systems such as legged robots.

Our Optimal Control for Switched Systems algorithm (OCS2 algorithm) consists of two main steps: a method which synthesizes the continuous input controller and a method which calculates the parameterized cost function derivatives with respect to the switching times. For synthesizing the continuous input controller, OCS2 uses the SLQ algorithm in Algorithm 1. As a dynamic programming approach, SLQ uses the Bellman
equation of optimality to locally estimate the value function and consequently the optimal control law. In the second step of OCS2, we use the approximated problem for calculating the value function from the first step to efficiently compute the value function gradient with respect to switching times. Using the SLQ algorithm to calculate the optimal control law for the continuous inputs has two major advantages. First, as discussed by Sideris and Rodriguez (2010), this algorithm has a linear time complexity with respect to the optimization time horizon in contrast to many standard discretization-based algorithms which scale cubically (such as the direct collocation methods). Second, since in each iteration of SLQ, an LQ problem is optimized, we can efficiently calculate the value function derivatives by obtaining derivatives of an LQ subproblem’s value function. However, in contrast to a regular time-varying LQ problem in which the system dynamics and the cost function are independent of the switching times, in this problem the cost and the system dynamics of the approximated LQ subproblem are functions of the switching times. Therefore, in order to calculate the cost function gradient, we should also consider the LQ subproblem variations to the switching times. We tackle this problem more rigorously in Theorem 1.

Here, we first briefly introduce the SLQ algorithm. We consider an intermediate iteration of the algorithm. We assume that \( \{\tilde{s}(z)\}_{z=0}^i \) and \( \{\tilde{u}(z)\}_{z=0}^i \) are the nominal state and input trajectories which are obtained by forward integrating system dynamics in (5) (performing a rollout) using the latest estimation of the optimal control law with a fixed switching time vector \( \tilde{I} \). For simplicity of notation, in the followings we have dropped the dependencies of the nominal trajectories and consequently the approximated LQ problem with respect to \( \tilde{I} \). The linearized system dynamics of the equivalent system in (5) around these nominal trajectories are defined as

\[
\frac{d(\delta x_k)}{dz} = (t_i - t_{i-1}) (A_i (z) \delta x_k + B_i (z) \delta u_k) \quad i \leq z < i + 1
\]

(7)

where \( \delta x_k(z) = x(z) - \tilde{x}(z), \delta u_k(z) = u(z) - \tilde{u}(z) \). The quadratic approximation of the cost function in (4) is

\[
J = \Phi(x_{t_f}) + \sum_{i=1}^{I} \int_{t_i-1}^{t_i} (t_i - t_{i-1}) L_i(z, x, u) dz
\]

(8)

In the above, \( q(z), q_i(z), r(z), Q(z), P(z), \) and \( R(z) \) are the coefficients of the Taylor expansion of the cost function in (4) evaluated at the nominal trajectories. The optimal control law for this LQ extremal subproblem can be derived by solving the following Riccati equations (Bryson, 1975)

\[
\begin{align*}
\frac{dS(z)}{dz} &= (t_i - t_{i-1}) W(z), \quad S(i^-) = S(i^+), \quad S(I) = Q_f \\
\frac{ds(z)}{dz} &= (t_i - t_{i-1}) w(z), \quad s(i^-) = s(i^+), \quad s(I) = q_f \\
\frac{ds(z)}{dz} &= (t_i - t_{i-1}) w(z), \quad s(i^-) = s(i^+), \quad s(I) = q_f
\end{align*}
\]

where \( S(z) \) and \( W(z) \) are in \( \mathbb{R}^{n \times n} \), \( s(z) \) and \( w(z) \) are in \( \mathbb{R}^{n} \), \( s(z) \) and \( w(z) \) are in \( \mathbb{R}^{n} \). These matrices are defined as

\[
\begin{align*}
W(z) &= Q(z) + A(z)^T S(z) + S(z) A(z) - L(z)^T R(z) L(z) \\
w(z) &= q(z) + A(z)^T s(z) - L(z)^T R(z) l(z) \\
w(z) &= q(z) - 0.05 \alpha (z - \alpha) I(z)^T R(z) I(z) \\
l(z) &= -R(z)^{-1} (r(z) + B(z)^T s(z)) \\
L(z) &= -R(z)^{-1} (P(z) + B(z)^T S(z)).
\end{align*}
\]

The updated optimal control law, the value function, and the total cost are defined as

\[
\begin{align*}
u(z, x) &= \tilde{u}(z) + \alpha l(z) + L(z) \delta x \\
V(z, x) &= s(z) + \delta x^T s(z) + s(z)^T \delta x \\
\dot{f} &= V(0, x_0) = s(0).
\end{align*}
\]

Lemma 1. \( H : \mathbb{R}^n \to \mathbb{R} \) and \( y : \mathbb{R} \to \mathbb{R}^n \) as twice continuously differentiable functions. If \( t_z - t_z < e_1 \) and \( |\delta y| < e_2 \) for small enough \( e_1 \) and \( e_2 \), then \( \tilde{y}(t_z + \delta t_z) \) can be approximated as

\[
H(\tilde{y}(t_z) + \delta t_z) \approx H(\tilde{y}) + \delta t_z V \tilde{H}(\hat{y}) + 0.5 \delta t_z^2 V^2 \tilde{H}(\hat{y})(\tilde{t} - \hat{t})^2,
\]

where \( \hat{y} = y(\tilde{t}), \delta \hat{y} = \frac{dy}{dt}, \) and \( \delta \tilde{y} = \frac{d\tilde{y}}{dt} \). Furthermore, \( V \) and \( V^2 \) are the gradient and the Hessian of \( H \) respectively.

Proof. This lemma can be easily proven using the Taylor series expansion of \( H \) and few steps of simplification.

Theorem 2. The partial derivative of the value function in (18) and the partial derivative of the total cost with respect to the switching time \( t_j \) can be derived as

\[
q_j V(z, x) = q_j s(z) + \delta x^T \partial_j s(z) + \delta x^T \partial_j P(z) \delta u + 0.5 \delta x^T \delta P(z) \delta u - 0.5 \delta x^T \delta P(z) \delta u
\]

(20)

where \( z \) is defined in (6), \( \alpha \in [0, 1] \) is the learning rate for backtracking line-search (Armijo, 1966). In each iteration of SLQ, the -search parameter controls the maximum step size to move along the feedforward component of the control law update. This parameter is chosen by starting with a full step in the direction of the update (\( \alpha = 1 \)), and then iteratively shrinking the step size until the cost associated to the updated controller rollout is lower than the current nominal trajectories cost. Before proceeding to the main theorem and its proof, we state the following lemma which will be used in Theorem 1.

Proof. This lemma can be easily proven using the Taylor series expansion of \( H \) and few steps of simplification.
Proof. Finally, the terminal condition are defined as (5). We can readily derive the differential equation which de-

\begin{align}
\frac{d}{dz}\left(\delta_{ij} - \delta_{i-1,j}\right)W(z) &= (\delta_{ij} - \delta_{i-1,j})W(z) + (t_i - t_{i-1})\partial_j W(z) \\
\frac{d}{dz}\left(\delta_{ij} - \delta_{i-1,j}\right)w(z) &= (\delta_{ij} - \delta_{i-1,j})w(z) + (t_i - t_{i-1})\partial_j w(z) \\
\frac{d}{dz}\left(\delta_{ij} - \delta_{i-1,j}\right)z(z) &= (\delta_{ij} - \delta_{i-1,j})z(z) + (t_i - t_{i-1})\partial_j z(z).
\end{align}

where we have

\begin{align}
\partial_j W(z) &= A(z)^\top \partial_j S(z) + \partial_j S(z)A(z) - \partial_j L(z)^\top R(z)\partial_j L(z) \\
\partial_j S(z) &= \partial_j q(z) + A(z)^\top \partial_j R(z)\partial_j S(z) - \partial_j L(z)^\top R(z)I(z) \\
\partial_j z(z) &= \partial_j q(z) - 0.5\alpha(2 - \alpha)(\partial_j I(z)^\top R(z))I(z) + L(z)^\top R(z)\partial_j I(z) \\
\partial_j I(z) &= -R(z)^{-1}r(z) + B(z)^\top \partial_j z(z) \\
\partial_j L(z) &= -R(z)^{-1}B(z)^\top \partial_j S(z).
\end{align}

The derivative of the cost coefficients with respect to \( t_j \) is

\begin{align}
\begin{bmatrix}
\partial_j q(z) \\
\partial_j r(z) \\
\partial_j q(z)
\end{bmatrix} &= \begin{bmatrix}
Q(z) \\
R(z) \\
q(z)^\top r(z)
\end{bmatrix} \begin{bmatrix}
\partial_j x(z) \\
\partial_j u(z)
\end{bmatrix}.
\end{align}

Finally, the terminal condition are defined as

\begin{align}
\partial_j S(I) = 0, \quad \partial_j x(I) &= Q_{j}^\top \partial_j x(I), \quad \partial_j s(I) = q_{j}^\top \partial_j x(I),
\end{align}

and the transversality conditions are

\begin{align}
\partial_j S(i^-) &= \partial_j S(i^-), \quad \partial_j x(i^-) = \partial_j x(i^-), \quad \partial_j s(i^-) = \partial_j x(i^-).
\end{align}

Proof. Equations (24-26) can be derived by directly differentiating the Riccati equations in (9-11), where based on the continuity condition, the order of differentiation with respect to \( z \) and \( t_j \) has been changed. For calculating \( \partial_j W, \partial_j w, \) and \( \partial_j w, \) we need to calculate the derivative of the linearized dynamics and the gradient approximation of the cost function with respect to switching time \( t_j \) as well. To do so, we need to examine the impact of \( t_j \) variation on the system dynamics and cost function of the approximated LQ subproblem.

In each approximated LQ subproblem, we use a quadratic approx-

\begin{align}
&= P(z)\partial_j x(z) + R(z)\partial_j u(z), \\
&= Q(z)\partial_j x(z) + P(z)^\top \partial_j u(z), \\
&= Q(z)^\top \partial_j x(z) + R(z)^\top \partial_j u(z), \\
&= Q(z)^\top \partial_j x(z) + R(z)^\top \partial_j u(z), \quad \partial_j P(z) = 0,
\end{align}

which can be written in matrix form as (27). By the same process, we can derive (28) for the final cost.

In order to find the linearized system dynamics sensitivity with respect to the switching time \( t_j \), we can use the result from

\begin{align}
&= \frac{d}{dz}\left(\delta_{ij} - \delta_{i-1,j}\right)\Phi(z, x) + (t_i - t_{i-1})\left(A_{ij}^\top \Phi(z, x) + B_{ij}\Phi(z, x)\right).
\end{align}

For simplicity in notation, we drop this dependencies on \( z \). Using the continuity condition of the state trajectory, the order of derivatives on the right hand side of the equation is changed which is resulted in (22). Furthermore, since the control input trajectory in SLQ comprises a time-varying feedbackterm and a time-varying linear state feedback, its sensitivity can be calculated as (23). Moreover, for the initial condition of this equation we have \( \delta_j \Phi(z = 0) = 0 \) due to the fixed initial state. Based on Lemma 1, the second order approximation of the intermediate cost \( L(x, u) \) around the nominal trajectories (which are in turn a function of \( t_j \)) can be written as

\begin{align}
L(x, \Phi(z) + \delta x, u(z) + \delta u) &\approx q(z) + \left[q(z)^\top r(z)\right] \left[\partial_j x(z) \partial_j u(z)\right] (t_j - t_j) \\
&+ \left[q(z)^\top r(z)\right] \left[\partial_j x(z) \partial_j u(z)\right] (t_j - t_j) \\
&+ \left[q(z)^\top r(z)\right] \left[\partial_j x(z) \partial_j u(z)\right] (t_j - t_j) \\
&+ \left[q(z)^\top r(z)\right] \left[\partial_j x(z) \partial_j u(z)\right] (t_j - t_j) \\
&+ \left[q(z)^\top r(z)\right] \left[\partial_j x(z) \partial_j u(z)\right] (t_j - t_j).
\end{align}

In the above, we have combined the like terms of the state and input increments. Based on this equation the sensitivity of the cost function components will be

\begin{align}
\partial_j r(z) &= P(z)\partial_j x(z) + R(z)\partial_j u(z), \\
\partial_j q(z) &= Q(z)\partial_j x(z) + P(z)^\top \partial_j u(z), \\
\partial_j q(z) &= Q(z)^\top \partial_j x(z) + R(z)^\top \partial_j u(z), \\
\partial_j P(z) &= 0,
\end{align}

which can be written in matrix form as (27). By the same process, we can derive (28) for the final cost.
Lemma 1 where we only keep the terms up to the first order.

\[
\begin{align*}
\frac{d}{dz} (x + \partial_1(x(t_j + \bar{\tau})) & \simeq (\bar{t}_i - \bar{t}_{i-1}) f(x, u) + A_i \partial_1 x + B_i \partial_1 u) \\
+ (t_j - t_i) (\delta_{ij} - \delta_{i-1,j}) f(x, u) & + (\bar{t}_i - \bar{t}_{i-1}) (A_i \partial_1 x + B_i \partial_1 u),
\end{align*}
\]

where for simplicity in notation, we drop the dependencies on time, z. By equating the coefficients, we get

\[
\frac{d}{dz} \delta_{ij} \simeq (\bar{t}_i - \bar{t}_{i-1}) f(x, u)
\]

which are respectively the nominal trajectory equation, the linear approximation of system dynamics, and trajectory sensitivity. Based on this approximation, the linear part is not a function of \(t_j\), so we get \(\partial_1 A_i(z) = 0\) and \(\partial_1 B_i(z) = 0\) (note that the effect of the switching time variations on the approximated system dynamics would have been appeared if we had used a second order or a higher order approximation).

4. OCS2 ALGORITHM

In Section 3, we have discussed the technical details behind the OCS2 algorithm. Here, we highlight the main steps of the algorithm for synthesizing the optimal continuous control law and the optimal switching times (refer to Algorithm 2). Each iteration of the OCS2 algorithm has three main steps namely (1) using SLQ algorithm to find the continuous inputs’ optimal control, (2) calculating the cost function gradient based on the LQ approximation of the problem which has been already calculated by the SLQ algorithm, (3) using a gradient descent method to update the switching times where we use the Frank-Wolfe method (Jaggi, 2013). An interesting aspect of our algorithm is its linear-time computational complexity with respect to the optimization time-horizon both in SLQ algorithm and calculating cost gradient. As we will show in the next section, this characteristic results in a superior performance of our algorithm in comparison to the two-point BVP approach originally introduced by Xu and Antsaklis (2004).

As discussed, OCS2 uses the SLQ algorithm for solving a nonlinear optimal control problem on the equivalent system with fixed switching times (refer to Algorithm 1). The SLQ algorithm is based on an iterative scheme where in each iteration it forward integrates the controlled system and then approximates the nonlinear optimal control problem with a local LQ problem. In general, the SLQ algorithm requires an initial stable controller for the first forward integration. For deriving the initial controller, we define a set of operating points in each switching mode (normally one point) and approximate the optimal control problem around these operating points with an LQ approximation. Then, we follow the same process described in the SLQ algorithm to design an initial controller.

A good initialization can often increase the convergence speed of the SLQ algorithm. One interesting characteristic of our algorithm is the warm starting scheme for the initial policy of SLQ. Here, we use a memorization scheme where we store the solutions of the different runs of SLQ in a solution bag and later initialize the policy of a new run of SLQ with the most similar switching time’s policy (refer to Algorithm 2). The similarity between two switching time sequences is measured as a sum of squared differences between corresponding times in two sequences.

5. CASE STUDIES

In this section, we evaluate the performance of the proposed method on three numerical examples. The first two examples are provided as illustrative cases to compare the performance of the proposed algorithm to the baseline algorithm introduced by Xu and Antsaklis (2004) and the third one is about the application of our method in motion control of legged robots. We here refer to the baseline algorithm as the BVP method since it is based on the solution of two-point BVPs.

Example 1: The first example is a nonlinear switched system with three mode switches which are defined as followings

\[
\begin{align*}
\dot{x}_1 &= x_1 + u_1 \sin x_1 \\
\dot{x}_2 &= -x_2 + u_1 \cos x_2 \\
\dot{x}_3 &= -x_3 - u_1 \sin x_1 - u_1 \cos x_1,
\end{align*}
\]

with the initial condition \(x_0 = [2, 3]^\top\). We apply both OCS2 and BVP to this system with uniformly distributed initial switching times. The optimized switching times, the optimized cost, the number of the iterations, and number of function calls (bottom-level optimization calls) for both algorithms are presented in Table 1. As it illustrates, OCS2 and BVP converge to the same solution within a comparable number of iterations and function calls. However, the consumed CPU times are significantly different. The BVP method utilizes about 235 seconds on an Intel’s core i7 2.7-GHz processor while the OCS2 algorithm uses only 31 seconds which is roughly 7.5 times faster (Figure 1).

Example 2: In order to examine the scalability of both algorithms to higher dimensions, we augment each subsystem in Example 1 with two more states and one additional control input. The augmented states have the following system dynamics

\[
\begin{align*}
\dot{x}_3 &= -x_3 + 3x_1 u_2 \\
\dot{x}_4 &= x_4 + x_1 u_2.
\end{align*}
\]

with the initial condition \(x_0 = [2, 3, 1, 1]^\top\). The cost function is the same as Example 1 with \(x_f = [1, -1, 2, 2]^\top\). As Example 1, the optimized solutions are comparable for both algorithms (refer to Table 1). However the CPU times are drastically different. For the BVP method, the CPU time is about 1500 seconds while the OCS2 algorithm uses only 76 seconds which is 19.5 times more efficient. This manifests the efficiency of the OCS2 algorithm in higher dimensional problems where computational time of the BVP algorithm prohibitively increases.

Example 3: Quadruped’s CoM motion control In this example, we apply OCS2 to the quadruped’s CoM control problem. In a quadruped robot, based on the stance legs configuration the system shows different dynamical behavior. Furthermore, at the switching instances the non-elastic nature of contacts introduces a jump in the state trajectory. Therefore, basically a legged robot have to be modeled as a nonlinear hybrid system with discontinuous state trajectory.
Fig. 1. Comparison between the CPU time consumption of the BVP algorithm (blue) and the OCS2 algorithm (orange). The evaluation is done on an Intel’s core i7 2.7-GHz processor. The BVP method run-time on Example 3 is missing since the algorithm failed to terminate.

Table 1. Comparison between the performance of the BVP and OCS2 algorithms. In the table Itr. is the number of iterations in the gradient-descent method until it converges. FC. is the number of requests for cost function and its derivative evaluation in the gradient-descent algorithm.

| Alg. | J   | s1  | s2  | Itr | FC. |
|------|-----|-----|-----|-----|-----|
| EX1  | BVP | 5.4498 | 0.2235 | 1.0198 | 9   | 13  |
| OCS2 | 5.4438 | 0.2324 | 1.0236 | 7   | 14  |
| EX2  | BVP | 10.3797 | 0.2754 | 1.6076 | 9   | 13  |
| OCS2 | 10.3888 | 0.2973 | 1.5978 | 8   | 20  |

The computational burden of solving the optimal control problem on such a nonlinear and hybrid system has encouraged researchers to use CoM dynamics instead of the complete system dynamics. In addition to optimizing a lower dimension problem, this approach has another important advantage: the CoM state trajectory is more smooth and the impact force does not cause a noticeable jump in states. Therefore, the CoM dynamics can be modeled as a switched system which facilitates solving the optimal control problem. The CoM system has in total 12 states consisting of: 3 states for orientation, θ, 3 states for position, p, and 6 states for linear and angular velocities in body frame, v and ω. The control inputs of the CoM system are an input which triggers the mode switch and the contact forces of the stance legs \((\mathbf{\lambda}^s(t))_{i=1}^4\). The simplified CoM equation can be derived from the Newton-Euler equation as

\[
\begin{align*}
\dot{\theta} &= \mathbf{R}(\theta)\omega, \\
\dot{\omega} &= \mathbf{I}^{-1}(-\omega \times \mathbf{I} \omega + \sum_{i=1}^{4} \mathbf{s}_i \mathbf{J}_i^W \mathbf{\lambda}_i), \\
\dot{p} &= \mathbf{R}(\theta)v, \\
\dot{v} &= \frac{1}{m}(mg - \sum_{i=0}^{4} \mathbf{s}_i \mathbf{J}_i^L \mathbf{\lambda}_i),
\end{align*}
\]

where \(\mathbf{R}(\theta)\) is the rotation matrix, \(\mathbf{g}\) is gravitational acceleration in body frame, \(\mathbf{I}\) and \(m\) are moment of inertia about the CoM and the total mass respectively. \(\mathbf{J}_i^W\) and \(\mathbf{J}_i^L\) are the Jacobian matrices of the \(i\)th foot with respect to CoM which depends on the foot position and CoM’s position and orientation (for technical specifications of this quadruped robot refer to Semini et al. (2011)). In this example, we study the trotting gait.

In a quadruped trotting gait, the pairwise alternating diagonal legs are the stance legs. In each phase of the motion only two opposing feet are on the ground. Since we assume that our robot has point feet, it loosens controllability in one degree of freedom, namely the rotation around the connecting line between the two stance feet. Therefore, the subsystems in each phase of the trotting gait are not controllable individually. However, the ability to switch from one diagonal pair to the other, allows the robot for gaining control over that degree of freedom. Therefore, for a successful trot, the robot requires to plan and control the contact forces as well as the switching times.

Fig. 2. Comparison between the initial (blue) and optimized (yellow) switching times sequence. After optimization mode 6 and 7 switching times are set to zero and the number of total steps reduced to 6 form the initial 8 steps.

The cost function in this example is defined as (8). The control effort weighting matrix is a diagonal matrix which penalizes all continuous control inputs (contact forces) equally. Furthermore, our cost function penalizes intermediate orientation and \(z\)-offsets during the entire trajectory. In the final cost, we penalize deviations from a target point \(p = [3, 0, 0]^T\) which lies 3 m in front of the robot. For the following evaluation, we assume that the \(x\) direction points to the front, the \(y\) direction points to the left and the \(z\) direction is orthogonal to ground. Furthermore, we penalize the linear and angular velocities to ensure the robot comes to a stop towards the end of the trajectory.

Figure 2 shows the difference between the initial and optimized switching times. The initial switching times have been chosen uniformly over the three-second period. During optimization the controller both extends and shortens different switching times. More interestingly, it set the time periods of two phases to zero. Therefore, although we initially asked the robot to take 8 steps the optimized trot has 6 steps.

Figure 3 shows the optimized contact forces of the trotting experiment. The vertical contact force \(\mathbf{\lambda}_i\) which needs to compensate for gravity is fairly equally distributed between the legs. However, we can see that the plots are not perfectly symmetric. This results from the quadruped being modeled after the hardware, which has unsymmetrical inertia. Additionally, we can see that the force profiles are non-trivial and thus would be difficult to derive manually. However, the contact forces are smooth within each step sequence which makes tracking easy.

Figure 4 shows a few snapshots of the quadruped during the optimized trotting gait. The CoM motion is controlled by the OCS2-designed controller. The swing feet’s trajectories (the ones that are not in contact) are designed based on a simple heuristic that the distance between the CoM and a swing foot at the moment of take off (breaking contact) and at the moment of touch down (establishing contact) should be the same.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an efficient method to solve the optimal control problem for nonlinear switched systems with predefined mode sequence. The proposed method is based on a two-stage optimization scheme which optimizes the continuous inputs as well as the switching times. In order to obtain an accurate estimation of the cost function derivatives, Xu and Antsaklis (2004) have introduced a set of two-point BVPs. Although there exist many numerical methods for solving two-point BVP (e.g. collocation method), many of them do not scale properly to high dimensional problems such as controlling quadruped locomotion. To tackle this issue, we have proposed the OCS2 algorithm which is based on an SLQ algorithm. In order to obtain cost function derivatives, OCS2 takes into account the sensitivity of the approximated LQ models with respect to the switching times. OCS2 obtains an approximation of the LQ
Borrelli, F., Baotic, M., Bemporad, A., and Morari, M. (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*.

Todorov, E. and Li, W. (2005). A generalized iterative lqg control of switching times in switched dynamical systems. In *42nd IEEE Conference on Decision and Control (CDC)*.

Xu, X. and Antsaklis, P.J. (2004). Optimal control of switched autonomous linear systems. In *51st IEEE Conference on Decision and Control (CDC)*.

Armijo, L. (1966). Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*.

Bemporad, A., and Morari, M. (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*.

Branicky, M.S., Borkar, V.S., and Mitter, S.K. (1998). A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control*.