

Switching Mode Generation and Optimal Estimation with Application to Skid-Steering

T. M. Caldwell, T. D. Murphey

*Mechanical Engineering
Northwestern University
Evanston IL 60208*

Abstract

Skid-steered vehicles, by design, must skid in order to maneuver. The skidding causes the vehicle to behave discontinuously during a maneuver as well as introduces complications to the observation of the vehicle's state, both of which affect a controller's performance. This paper addresses estimation of contact state by applying switched system optimization to estimate skidding properties of the skid-steered vehicle.

In order to treat the skid-steered vehicle as a switched system, the vehicle's ground interaction is modeled using Coulomb friction, thereby partitioning the system dynamics into four distinct modes, one for each combination of the forward and back wheel pairs sticking or skidding. Thus, as the vehicle maneuvers, the system propagates over some mode sequence, transitioning between modes over some set of switching times. This paper presents second-order optimization algorithms for estimating these switching times. We emphasize the importance of the second-order algorithm because it exhibits quadratic convergence and because even for relatively simple examples, first-order methods fail to converge on time scales compatible with real-time operation. Furthermore, the paper presents a technique for estimating the mode sequence by optimizing a relaxation of the switched system.

Key words: Autonomous vehicles; Autonomous mobile robots; optimization; optimal estimation; optimal control; estimation algorithms

1 Introduction

The skid-steered vehicle (SSV) is a simple vehicle composed of a body supported by four independently driven wheels. The generalized forces applied to the vehicle's left and right actuators dictates maneuvers. Consequently, the SSV is highly maneuverable, allowing for zero-point turns, which occur when the left wheel torques are equal but opposite to the right wheel torques and sufficiently large to overcome static friction. The high degree of maneuverability can be contrasted with the limited maneuverability of a car, which must translate in order to change orientation. The greater maneuverability comes at a price, however. The SSV's wheels must skid laterally against the ground in order to turn and this skidding introduces uncertainties in the vehicle's position and orientation during a maneuver due to the irregular nature of friction.

Caracciolo et al. [9] attempt to reduce the uncertainties during a maneuver by constraining the vehicle's instantaneous center of rotation (ICR). The authors explain that a SSV's ICR is not restricted to a certain region and that the ICR may even extend outside the wheelbase of the vehicle, resulting in unstable motion. Imitating the car, a vehicle with an ICR theoretically aligned with the rear axle, the authors introduce a non-holonomic constraint to pin down the longitudinal coordinate of the ICR. We argue that this limits the turning capability of an SSV, much like a car. Moreover, this abstraction is in no way based on the underlying dynamics of the physical system, making it difficult to prove properties of the algorithm.

Email addresses: caldwelt@u.northwestern.edu (T. M. Caldwell), t-murphey@northwestern.edu (T. D. Murphey).

Another approach reduces skidding uncertainties through extensive analysis of the SSV’s ground interaction. Tran et al. [28], for instance, analyze the shear stresses of the contact points of each wheel in order to ascertain the resistive forces with respect to different types of surfaces. This assumes regularity of the ground characteristics and disregards vehicle dynamics.

In contrast, Anousaki and Kyriakopoulos [3] disregard the complexities of the ground interaction by modeling the SSV with a simple unicycle model and estimating its configuration by applying well established estimation techniques. The authors fuse a dead-reckoning scheme with an inertial navigation system (INS) using a Kalman filter to estimate the configuration trajectories (position and orientation) of the SSV. The dead-reckoning scheme is designed from a kinematic model with wheel encoders (i.e. odometry) as inputs. The authors note that their approach falters when vehicle characteristics (e.g. wheel pressure, location of center of mass, ground characteristics) change slightly, requiring recalibration of the model. This sensitivity likely stems from applying a stochastic method intended for linear, or at least smooth systems, to deal with uncertainty generated from friction, a highly non-smooth dynamic system. In [27], a similar Kalman filter approach is analyzed and deemed to be inferior for the SSV.

In order to further their approach, Anousaki and Kyriakopoulos, in [4], fuse a simultaneous localization and map-building scheme (SLAM) with the dead-reckoning from above. The scheme uses a laser scan to map the vehicle’s surroundings and match with previous maps for localization. The downside of map matching is that the computational cost grows every time new objects are added. Furthermore, the authors still note that the dead-reckoning continues to falter with varying vehicle characteristics, requiring recalibration.

Consequently, due to [3,4], we expect an intelligent estimation of skidding characteristics to improve the performance of a controller without the need for expensive hardware or computationally taxing algorithms. For instance, Yi et al. [33] present an approach that uses a pseudo-static friction model to capture the SSV’s interaction with the ground. The authors present an adaptive control algorithm to estimate the coefficient of their underlying friction model in order to better track a desired trajectory. Their experimental results demonstrate that the proposed controller performs significantly better than a simple PID controller. However, the proposed estimation fails to converge near the coefficient’s true value, which questions the value of this approach’s estimation of the skidding characteristics.

Another approach that estimates the skidding characteristics of a SSV is in [27]. The authors apply a non-linear sliding mode observer (SMO) to estimate the SSV’s slip angle and the slip-ratios of the inner and outer wheels. The advantage of an SMO technique is that it accounts for the non-smooth period occurring when the ground contacts transition between slipping and sticking, a difficulty for continuous observers. For more information on sliding mode controllers and observers, an extensive analysis of them for mechanical systems is in [29].

Similar to the reason the authors in [27] use a SMO, we treat the SSV’s wheel/ground interaction as a discrete phenomenon, where the wheels discretely transition between sticking and skidding. In contrast, though, we model the SSV as a switched system and use optimization to ascertain skidding characteristics. A switched system is one that evolves over multiple distinct sub-systems by transitioning from one sub-system to another in a discrete manner [2,6–8,11,13–15,18,24–26,30,12,31,32]. In this paper, we investigate the application of switched system optimization to estimate whether the SSV’s wheels are sticking or skidding against the ground. Primarily, we are concerned with the wheels’ skid states in the lateral wheel direction. This is because skidding in the longitudinal direction induces a frictional force in the desired direction and can be modeled as uncertainty in the inputted wheel torques.

We model the wheel-ground contact of the SSV using Coulomb friction. Coulomb friction partitions the SSV into four sub-systems for the four combinations of the front and back wheel pairs skidding or sticking against the ground. Therefore, each sub-system evolves over distinct modes of operation. The four sub-systems are: 1. all wheels sticking 2. front wheels sticking, back wheels skidding 3. front wheels skidding, back wheels sticking 4. all wheels skidding.

Thus, as the SSV drives along some path, the system propagates over some mode sequence (denoted by Ψ), transitioning between modes over some set of switching times (denoted by \mathcal{T}). Our goal is to estimate the pair (\mathcal{T}, Ψ) .

To this end, the switching times are estimated using optimal techniques for switched systems (see [7,8,11,13–15,24,30–32]). In this case, the switching times are the control inputs, where the optimal switching times, \mathcal{T}^* , are the switching times that minimize some performance index of the system. Therefore, the goal is to find

$$\arg \min_{\mathcal{T}} J(\mathcal{T}, \Psi)$$

where $J(\cdot, \cdot)$ is the performance index. When observing the wheel/ground interaction of a SSV, we choose the cost function to be the time integral of some metric $\ell(\cdot)$ placed on the difference between measured configuration trajectories, $h(t)$, and simulated configuration trajectories, $x(t)$, plus some terminal cost $m(\cdot)$:

$$J(\mathcal{T}, \Psi) = \int_{t_0}^{t_f} \ell(x(\tau) - h(\tau)) \, d\tau + m(x(t_f) - h(t_f)).$$

Thus, the minimizing switching times correspond to the state trajectories that best track the measured trajectories, where “best” is defined by $\ell(\cdot)$ and $m(\cdot)$. In this paper, we investigate two minimization techniques: steepest descent and Newton’s method. Steepest descent uses a first-order model to approximate the local features of the cost function. The first order model’s descent direction is the negative gradient of the cost function, which is the steepest descent direction. Newton’s method uses a second-order model to approximate the local features. If the model has a local minimum (i.e. the second-order term of the model is positive definite), the model supplies more than a descent direction; it supplies a local optimum, which is the local minimum of the second-order model. Steepest descent exhibits linear convergence while Newton’s method exhibits quadratic convergence (see [19,21]). This distinction between steepest descent and Newton’s method is a primary point of this paper because the *rate of convergence is a paramount concern for on-line implementation*, an ultimate goal of our work. For illustration, refer to Fig 1, which compares the convergence of steepest descent and Newton’s method for an SSV estimation example presented later in the paper.

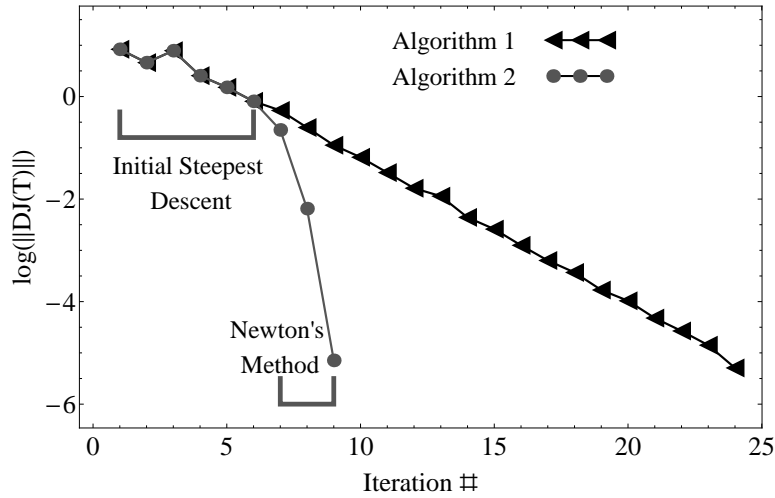


Fig. 1. The convergence to the zero gradient of the cost functional, $DJ(\mathcal{T}) = 0$, for Algorithm 1 and Algorithm 2. Algorithm 1 is a pure steepest descent algorithm whereas Algorithm 2 conducts a user defined number of initial steps of steepest descent followed by Newton’s method until convergence (see Section 3).

As for implementation, steepest descent algorithms for switching time optimization are in [11,13–15,31,32]. The gradient for steepest descent is derived in [31]. The authors in [11] and [13] use an adjoint formulation of the gradient calculation which requires the solving of only two differential equations. In other words, the computational complexity for each step of steepest does not change with regard to the number of switches. Furthermore, [14] propose a gradient-projection algorithm to deal with the inequality constraint necessitating that the switching times are monotonic.

Interestingly, a precise calculation for the second derivative of Newton’s Method does not exist as far as the authors are aware. Xu and Antsaklis [31,32] approximate the second derivative. Egerstedt et al. have calculated it for bi-modal LTI systems [15]. We present explicit adjoint equations for the second-order descent direction for non-linear switched systems as well as an algorithm for applying Newton’s method. We show that the *quadratic convergence of Newton’s method permits the convergence needed for real-time estimation of SSV contact state*.

Some of the literature assumes the mode sequence is known ahead of time [31]. Since this is not always the case, much literature deals with optimizing the mode sequence. In [26], Shaikh and Caines use combinatorial search algorithms

to locally optimize the mode sequence. They further this approach in [25] by finding optimality zones with the goal of reducing the exponential complexity of the combinatoric search. In [30], the author proposes a method for cycling the modes a specific number of times while removing modes that exist for a short time interval between steps of a descent algorithm. In [13], the authors propose a method for removing modes that exist for short time intervals as well as a method for injecting modes for a certain time interval I depending on the sign of the gradient of that interval, $D_I J(T)$. This method is formalized into an algorithm in [14] and the convergence of the approach is investigated in [7].

Each of the approaches in the previous paragraph separate the switched system optimization into two stages: switching time optimization and mode sequence optimization. In contrast, Alamir and Attia, in [2] and [6], utilize a switched system representation where the switching times and mode sequence may be optimized together. As Alamir and Attia argue in [1], the two stage approaches invariably lead to a combinatoric complexity. We propose a similar approach to that of Alamir and Attia in order to estimate the mode sequence.

In this paper, a novel approach for estimating a mode sequence is proposed and implemented for the SSV model. The approach makes use of an infinite dimensional definition of the switched system that permits relaxation. Instead of using switching times to separate the activation periods of each mode, the infinite dimensional system uses the control inputs $u(t) = (u_1, u_2, \dots, u_N)(t)$ to specify which modes are active or inactive. If the i^{th} mode is active, then $u_i = 1$, and if it is inactive, then $u_i = 0$. Furthermore, only one mode may be active at any given time. This definition of the system benefits from lacking a mode sequence, allowing the mode sequence to instead be estimated. However, the optimization of this definition of the system is a constrained infinite dimensional problem. Therefore, we relax the constraints that only one mode may be active and that $u(t)$ must be 1 or 0. Then, the optimal relaxed switched system can be projected back onto the set of feasible solutions in order to estimate a mode sequence. This infinite dimensional optimization may be accomplished using any optimal control technique; we use the projection operator approach presented in [16] and [17] because of its numerical stability.

This paper’s contribution is threefold:

- We generate a second-order descent direction for switched systems that provides quadratic convergence. This descent direction may be applied to any switched system, not just the system of focus in this paper.
- We propose a relaxed optimization routine that estimates the mode sequence for the switching time optimization algorithms. The mode sequence estimation can also be applied to any switched system.
- We provide algorithms for optimally estimating the wheel-ground contact state of a skid-steered vehicle by treating the system as a switched system.

This paper is organized as follows: Section 2 introduces the SSV and decomposes its dynamics into four skid sub-systems. Section 3 introduces switched system optimization. This section presents second-order switching time descent directions for steepest descent and Newton’s method. In this section a mode sequence is assumed, but for many systems, including the SSV, a mode sequence is not always clear. Section 4 estimates a mode sequence utilizing a relaxation of an infinite dimensional representation of the switched system. The final section, Section 5, contains a number of examples applying switched system optimization and relaxed switched system mode sequence generation for an SSV, including some initial experimental results.

2 Modeling the Skid-Steered Vehicle

The SSV is composed of four wheels supporting a body. The wheels are locked in alignment with the body and are independently driven. This vehicle design restricts the SSV to maneuvers dictated by differential torques to the left and right wheel pairs, resulting in lateral skidding. Therefore, in order to turn, the difference between the torques must be large enough to break the lateral static friction of the wheels against the ground. Our SSV is pictured in Fig 2.

Our representation of the SSV is the same as that in [22]. Considering the SSV as a set of connected rigid bodies, the vehicle’s configuration is $x = (X, Y, \theta, \phi_1, \phi_2, \phi_3, \phi_4)$. The Cartesian coordinates, X and Y specify the center of geometry relative to the world frame. The orientation, θ , is the heading of the vehicle where θ increases counter-clockwise and $\theta = 0$ only occurs when the heading is aligned with the X -axis of the world frame. The angle of rotation of each wheel respectively is $\phi_1, \phi_2, \phi_3, \phi_4$. The sub-script, 1, refers to the back right wheel and 2, 3, and 4 are ordered clockwise from wheel 1 as shown in Fig 3.

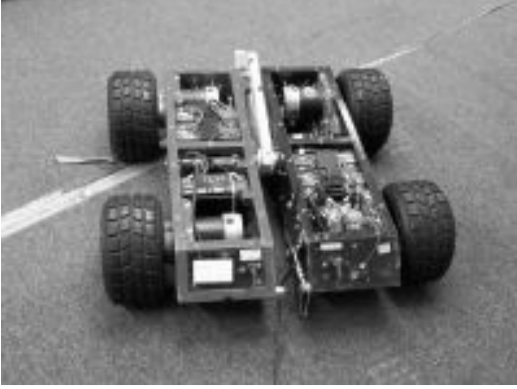


Fig. 2. The “Flexy Flyer”.

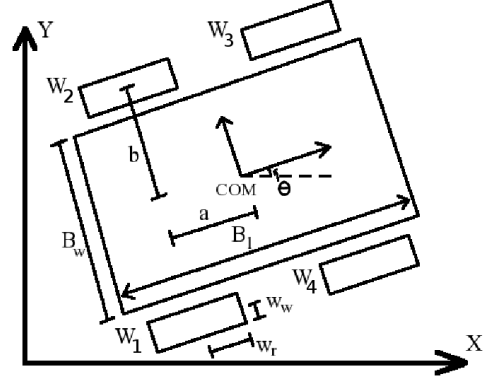


Fig. 3. Slip-steered vehicle with frame at its center of mass.

As for the wheel/ground interaction, we use a Coulomb friction model. Coulomb friction partitions the SSV into a switched system, where switched systems are described in the next section. Assuming the geometry of the wheels as seen in Fig 3, Coulomb friction generates four distinct sub-systems that are combinations of the front and back wheels sticking or skidding with respect to the ground. Sticking is enforced using non-holonomic constraints. The four sub-systems are

	front wheels	back wheels
σ =	1:	sticking laterally sticking laterally
	2:	sticking laterally skidding laterally
	3:	skidding laterally sticking laterally
	4:	skidding laterally skidding laterally

Hence, $\Sigma = \{1, 2, 3, 4\}$. The sub-systems $\sigma = 2$ and $\sigma = 3$ occur when the back or front wheels respectively are “fish tailing”. This occurs when the instantaneous center of rotation is aligned with the front wheels ($\sigma = 2$) or the back wheels ($\sigma = 3$).

As in [22], we simplify the representation of the vehicle to the configuration $x(t) = (X, Y, \theta)(t)$. We name the reduced model as the *simplified model* and the model that accounts for the wheel rotations (and associated inertias) as the *full model*. The simplified model is a reduction of the full model at the cost of losing the impact of the rotational inertias due to the spinning of the wheels. If the mass of the body is significantly greater than the mass of the wheels, this trade off of fidelity for computation is not a bad one.

Since the simplified model does not account for the wheels, the wheel torques must instead be transformed forces (i.e., using the $Ad(\cdot)$ operator [23]) to the center of mass of the vehicle. The inertia tensor for the simplified model is $G = m dx \otimes dx + m dy \otimes dy + J d\theta \otimes d\theta$, where m is the mass of the vehicle and J is the moment of inertia around the vehicle’s center of mass.

The equations of motion for each $\sigma \in \Sigma$ are found using the constrained Euler-Lagrange equations (see [23]). Depending on the value of σ , the proper non-holonomic constraints are enforced. This model is similar to the way [20] generates the dynamics of a SSV with the only dissimilarity of how friction enters into the equations.

The equations of motion for $\sigma = 1$ and $\sigma = 4$ are shown. Although we also compute the equations for $\sigma = 2, 3$, we

do not reproduce them here because of their algebraic complexity.

$$\begin{aligned}
f_{\sigma=1} &: \begin{cases} \ddot{X} = \frac{(F_1+F_2+F_3+F_4)\cos\theta(t)}{m_B+4m_w} - c_1\dot{X} \\ \ddot{Y} = \frac{(F_1+F_2+F_3+F_4)\sin\theta(t)}{m_B+4m_w} - c_1\dot{Y} \\ \ddot{\theta} = 0 \end{cases} \\
f_{\sigma=4} &: \begin{cases} \ddot{X} = \frac{(F_1+F_2+F_3+F_4)\cos\theta(t)}{m_B+4m_w} - c_4\dot{X} \\ \quad + g\mu_K \sin\theta(t)[- \sin\theta(t)\dot{X}(t) + \cos\theta(t)\dot{Y}(t)] \\ \ddot{Y} = \frac{(F_1+F_2+F_3+F_4)\sin\theta(t)}{m_B+4m_w} - c_4\dot{Y} \\ \quad - g\mu_K \cos\theta(t)[- \sin\theta(t)\dot{X}(t) + \cos\theta(t)\dot{Y}(t)] \\ \ddot{\theta} = \frac{12b(F_1-F_2-F_3+F_4)+12a^2g(m_B+4m_w)\mu_K\dot{\theta}(t)}{4m_w(12a^2+12b^2+w_w^2+3w_r^2)+m_B(B_l^2+B_w^2)} \end{cases} \quad (1)
\end{aligned}$$

where m_B is the mass of the body, m_w is the mass of a wheel, g is gravity, μ_K is the coefficient of kinetic friction, F_1 , F_2 , F_3 , and F_4 are the respective transformed wheel torques, and c_1 and c_4 are the respective damping coefficients.

One may be concerned with μ_K showing in the equations of motion for when $\sigma = 4$ (and $\sigma = 2, 3$). The coefficient of kinetic friction designates the loss of energy during a turn due to lateral friction. Therefore, μ_K may be a general estimation of the wheel-ground friction. Section 5 provides evidence that our algorithms perform well with disturbances to μ_K . Nevertheless, we plan to include on-line estimation of this parameter in future work.

Modeling the SSV as we have in this section alleviates the necessity for some complicated model of the stick/skid transitions, which would require extensive knowledge of the ground characteristics. Instead, with our model, the estimation of these transitions is based solely on an optimization over the full configuration of the vehicle. The next section provides a method to compute such an optimization.

3 First- and Second-Order Switching Time Optimization

A switched system is defined by how the system's modes of operation evolve over time. We present three equivalent definitions of the state trajectories, $x(t) : \mathbb{R} \mapsto \mathbb{R}^n$.

The first is the standard definition [8,11,13,24,30,31]:

$$\begin{aligned}
\dot{x}(t) = f(x(t), \mathcal{T}, \Psi, t) &= \begin{cases} f_1(x(t), t) & , \quad T_0 \leq t < T_1 \\ f_2(x(t), t) & , \quad T_1 \leq t < T_2 \\ \vdots & \vdots \\ f_N(x(t), t) & , \quad T_{N-1} \leq t < T_N \end{cases} \quad (2) \\
&\text{subject to: } x(T_0) = x_0
\end{aligned}$$

where N is the number of modes, $\mathcal{T} = \{T_1, T_2, \dots, T_{N-1}\} \in \mathbb{R}^{N-1}$ is a monotonically increasing set of switching times, T_0 is the initial time, T_N is the final time, x_0 is the initial state, and $f_i : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$ is the i^{th} mode of operation in the sequence Ψ . Each mode in Ψ is an element of the set of all modes, Σ . The switching times are the control inputs and a mode sequence is assumed.

The second equivalent definition of $f(\cdot, \cdot, \cdot)$ uses step functions to designate the activation periods of each mode:

$$\begin{aligned}
\dot{x}(t) = f(x(t), \mathcal{T}, t) &= \left[1(t - T_0) - 1(t - T_1^-)\right] f_1(x(t), t) + \left[1(t - T_1^+) - 1(t - T_2^-)\right] f_2(x(t), t) \\
&\quad + \dots + \left[1(t - T_{N-1}^+) - 1(t - T_N)\right] f_N(x(t), t). \quad (3)
\end{aligned}$$

The super scripts $+$ and $-$ correct for the ambiguity at T_i for $i = 1, \dots, N - 1$.¹ Once again, \mathcal{T} are the control inputs and Ψ is assumed. We prefer Eq (3) over Eq (2) because the switching times explicitly enter Eq (3). This makes apparent what the switching time derivatives of $f(x(t), \mathcal{T}, t)$ are.

The third equivalent definition is also presented in [2,6]. The third definition replaces the step functions specifying the activation periods with the constrained infinite dimensional control input $u(t)$:

$$f(x(t), u(t), t) = u_1(t)f_{\sigma=1}(x(t), t) + u_2(t)f_{\sigma=2}(x(t), t) + \dots + u_{\widehat{N}}(t)f_{\sigma=\widehat{N}}(x(t), t) \quad (4)$$

where \widehat{N} is the size of Σ and the control inputs, $u(t)$ are such that $u(t) : \mathbb{R} \mapsto \mathcal{U}$. Elements of \mathcal{U} must satisfy the following constraints:

- (1) $u_1(t), u_2(t), \dots, u_{\widehat{N}}(t) \in \{0, 1\}$
- (2) $\sum_{i=1}^{\widehat{N}} u_i(t) = 1.$

We present the third definition because it lacks a mode sequence, an assumed parameter of the first two definitions. Therefore, the mode sequence can instead be estimated. We explore this in Section 4 by relaxing the constraints on $u(t)$ and projecting the optimal relaxed solution onto the set of feasible solutions.

The rest of Section 3 is concerned with optimizing the switched system with respect to switching times, where the mode sequence, Ψ , is assumed to be known and constant.

3.1 Steepest Descent and Newton's Method

Let us choose the performance of the switched system be the integral of the Lagrangian, $\ell(\cdot, \cdot)$, plus a terminal cost $m(\cdot, \cdot)$,

$$J(\mathcal{T}) = \int_{T_0}^{T_N} \ell(x(\tau), h(\tau)) d\tau + m(x(T_N), h(T_N)). \quad (5)$$

For SSV estimation, the Lagrangian generally places a metric on the difference of the state trajectories, $x(t)$, with the measured trajectories, $h(t)$. The terminal cost places an additional cost at the terminal time. Often, we omit $h(t)$ from the notation.

The goal is to find the switching times that minimize the cost function. In other words, the goal is to find:

$$\arg \min_{\mathcal{T}} J(\mathcal{T}).$$

Suppose we have a set of points that satisfy this goal. Let us denote these minimizers as, \mathcal{T}^* . Then, $J(\mathcal{T}^*)$ has the following properties assuming $J(\mathcal{T}) \in C^2$:

$$1 \quad DJ(\mathcal{T}^*) = 0 \quad (6)$$

$$2 \quad D^2J(\mathcal{T}^*) > 0 \quad (\text{positive definite}) \quad (7)$$

Now suppose we don't know \mathcal{T}^* . One way to find \mathcal{T}^* is to solve Eq (6) directly and ensure Eq (7) is satisfied. Often, this calculation is impossible or impractical. For this reason, descent techniques are commonly employed [19]. Descent techniques are iterative and have the following form:

$$\mathcal{T}_{k+1} = \mathcal{T}_k + \gamma d_k$$

¹ For instance, if the current time is $t = T_i^-$, then the current mode is f_i and if the current time is $t = T_i^+$, then the current mode is f_{i+1} .

where k is the current iteration of the descent, γ is the step size and d is the step direction.

A descent in the cost function is ensured by choosing a γ that satisfies the *Sufficient Decrease Condition*

$$J(\mathcal{T}_k + \gamma d_k) \leq J(\mathcal{T}_k) + \alpha \gamma DJ(\mathcal{T}_k) \cdot d_k \quad (8)$$

where $\alpha \in (0, 1)$. Armijo [5] suggests choosing $\gamma = \beta^m$ and finding the smallest m that satisfies the condition where m is a positive integer and $\beta \in (0, 1)$. This is known as the *Armijo Rule*.

This paper investigates two descent techniques: steepest descent and Newton's method. The descent directions for the two techniques are (see [19,21]):

$$\begin{aligned} \text{Steepest Descent: } d &= -DJ(\mathcal{T}) \\ \text{Newton's Method: } d &= -D^2J(\mathcal{T})^{-1} \cdot DJ(\mathcal{T}) \end{aligned}$$

where steepest descent converges linearly and Newton's method converges quadratically (see [19,21]). As stated in the introduction, the convergence rate is important for on-line implementation. Therefore, we stress the value of the quadratic convergence of Newton's method. See [12] for further elaboration of this point. In [12], the authors utilize Newton's method for switched systems in an on-line implementation algorithm. The two convergences are defined as

Definition:

- (1) A sequence has *linear convergence* if the sequence approaches T^* with a rate of convergence $K \in (0, 1)$ such that

$$\|\mathcal{T}_{k+1} - \mathcal{T}^*\| \leq K \|\mathcal{T}_k - \mathcal{T}^*\|$$

- (2) A sequence has *quadratic convergence* if the sequence approaches T^* with a rate of convergence $K > 0$ such that

$$\|\mathcal{T}_{k+1} - \mathcal{T}^*\| \leq K \|\mathcal{T}_k - \mathcal{T}^*\|^2$$

In order to calculate the descent directions for steepest descent and Newton's method, we present formulas for $DJ(\mathcal{T})$ and $D^2J(\mathcal{T})$. The formula proofs rely on a basic understanding of State Transition Matrices, so a review of them is presented as a footnote.²

The following are the calculations for $DJ(\mathcal{T})$ and $D^2J(\mathcal{T})$ and their proofs:

² Consider the linear time varying (LTV) control system

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ \text{subject to: } x(t_0) &= x_0. \end{aligned} \quad (9)$$

The solution to this system is

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau) d\tau \quad (10)$$

where $\Phi(t, \tau)$ is the state-transition matrix corresponding to $A(t)$. $\Phi(t, \tau)$ satisfies the following properties: $x(t) = \Phi(t, \tau)x(\tau)$, $\frac{\partial}{\partial t}\Phi(t, \tau) = A(t)\Phi(t, \tau)$, $\frac{\partial}{\partial \tau}\Phi(t, \tau) = -\Phi(t, \tau)A(\tau)$, $\Phi(t, t) = I$, $\Phi(t, \tau) = \Phi(t, s)\Phi(s, \tau)$. In order to calculate $\Phi(t, \tau)$ given $A(t)$, solve the differential equation $\frac{d}{dt}\Phi(t, \tau) = A(t)\Phi(t, \tau)$ with initial condition $\Phi(\tau, \tau) = I$. For reference, see [10].

3.2 Calculating $DJ(\mathcal{T})$

Lemma 1 *Provided every $f_i(\cdot, \cdot)$ in f is C^1 , the i^{th} switching time derivative of $J(t)$ is³*

$$D_{T_i}J(\mathcal{T}) = \rho^T(T_i) \left[f_i(x(T_i), t) - f_{i+1}(x(T_i), t) \right] \quad (11)$$

where $\rho(\cdot)$ is the solution to the following backwards differential equation:

$$\begin{aligned} \dot{\rho}(t) &= -D_1f(x(t), \mathcal{T}, t)^T \rho(t) - D\ell(x(t))^T \\ \text{subject to: } \rho(T_N) &= Dm(x(T_N))^T. \end{aligned} \quad (12)$$

PROOF. A proof using calculus of variations is in [13]. The proof we present has an equivalent result to that in [13], but we employ a different technique that generalizes to $D^2J(\mathcal{T})$, presented later in the paper.

The switching-time derivative of the cost function in the direction of the variation θ is

$$DJ(\mathcal{T}) \cdot \theta = \int_{T_0}^{T_N} D\ell(x(\tau)) \cdot z(\tau) \, d\tau + Dm(x(T_N)) \cdot z(T_N) \quad (13)$$

where $z(t) : \mathbb{R} \mapsto \mathbb{R}^n$ is the variation of $x(t)$ due to the variation, θ , in \mathcal{T} . $z(t)$ is the solution to

$$\begin{aligned} \dot{z}(t) &= \frac{\partial}{\partial \mathcal{T}} \dot{x}(t) = D_1f(x(t), \mathcal{T}, t) \cdot z(t) + D_2f(x(t), \mathcal{T}, t) \cdot \theta \\ \text{subject to: } z(0) &= \frac{\partial}{\partial \mathcal{T}} x(0) = 0. \end{aligned} \quad (14)$$

Referring to Eq (3), $D_1f(x(t), \mathcal{T}, t)$ and $D_2f(x(t), \mathcal{T}, t)$ become

$$\begin{aligned} D_1f(x(t), \mathcal{T}, t) &= \left[1(t - T_0) - 1(t - T_1^-) \right] D_1f_1(x(t), t) + \left[1(t - T_1^+) - 1(t - T_2^-) \right] D_1f_2(x(t), t) \\ &\quad + \cdots + \left[1(t - T_{N-1}^+) - 1(t - T_N) \right] D_1f_N(x(t), t) \end{aligned}$$

and

$$D_2f(x(t), \mathcal{T}, t) = \left\{ \delta(t - T_k^-) f_k(x(t), t) - \delta(t - T_k^+) f_{k+1}(x(t), t) \right\}_{k=1}^{N-1} \quad (15)$$

where $\delta(\cdot)$ is the Dirac delta function.

Define $A(t) \triangleq D_1f(x(t), \mathcal{T}, t)$ and $B(t) \triangleq D_2f(x(t), \mathcal{T}, t)$. Therefore, \dot{z} is

$$\begin{aligned} \dot{z}(t) &= A(t) \cdot z(t) + B(t) \cdot \theta \\ \text{subject to: } z(0) &= 0. \end{aligned}$$

³ For notation, $D_{T_i}J$ is the partial derivative of J with respect to T_i . Furthermore, we use the slot derivative. Simply, $D_1J(\cdot, \cdot)$ is the partial derivative of J with respect to the argument in its first slot. Likewise, $D_2J(\cdot, \cdot)$ is with respect to the argument in its second slot and so forth. Later we will use $D_{1,2}J(\cdot, \cdot)$ to denote the second partial of J with respect to its first and second arguments.

This differential equation is of the same form as Eq (9) and has the solution

$$z(t) = \int_{T_0}^t \Phi(t, \tau) B(\tau) \cdot \theta \, d\tau \quad (16)$$

where $\Phi(t, \tau)$ is the state transition matrix corresponding to $A(t)$ in Eq (15). Plugging $z(\cdot)$ into $DJ(\mathcal{T}) \cdot \theta$ in Eq. (13), we see that

$$DJ(\mathcal{T}) \cdot \theta = \int_{T_0}^{T_N} D\ell(x(\tau)) \int_{T_0}^{\tau} \Phi(\tau, s) B(s) \cdot \theta \, ds \, d\tau + Dm(x(T_N)) \int_{T_0}^{T_N} \Phi(T_N, s) B(s) \cdot \theta \, ds.$$

Moving $D\ell(x(\tau))$ and $Dm(x(T_N))$ into their respective integrals and switching the order of integration of the first term results in

$$= \int_{T_0}^{T_N} \int_s^{T_N} D\ell(x(\tau)) \Phi(\tau, s) B(s) \cdot \theta \, d\tau \, ds + \int_{T_0}^{T_N} Dm(x(T_N)) \Phi(T_N, s) B(s) \cdot \theta \, ds. \quad (17)$$

Now, we may combine the integrals and move all terms not depending on τ outside the inner integral and denote the inner integral plus the terminal term as $\rho(s)^T$:

$$= \int_{T_0}^{T_N} \underbrace{\left[\int_s^{T_N} D\ell(x(\tau)) \Phi(\tau, s) \, d\tau + Dm(x(T_N)) \Phi(T_N, s) \right]}_{\rho(s)^T} B(s) \, ds \cdot \theta.$$

Then, $\rho(t) = \Phi(T_N, t)^T Dm(x(T_N)) + \int_t^{T_N} \Phi(\tau, t)^T D\ell(x(\tau)) \, d\tau$, where $\rho(t)$ is the solution to the backwards differential equation:

$$\begin{aligned} \dot{\rho}(t) &= -A(t)^T \rho(t) - D\ell(x(t))^T \\ &= -D_1 f(x(t), \mathcal{T}, t)^T \rho(t) - D\ell(x(t))^T \\ \text{subject to: } \rho(T_N) &= Dm(x(T_N))^T. \end{aligned} \quad (18)$$

This result is verified by taking the time derivative of $\rho(t)$ or by comparing to Eqs. (9) and (10). $DJ(\mathcal{T})$ becomes

$$DJ(\mathcal{T}) \cdot \theta = \int_{T_0}^{T_N} \rho(s)^T B(s) \, ds \cdot \theta.$$

Integrating the δ functions in $B(s)$ pick out the times for which the δ -functions in $B(s)$ are 0, such that

$$\begin{aligned} D_{T_i} J(\mathcal{T}) \cdot \theta_i &= \rho^T(T_i) \left[f_i(x(T_i), t) - f_{i+1}(x(T_i), t) \right] \cdot \theta_i \\ &\text{for } i = 1, 2, \dots, N-1 \end{aligned}$$

where θ_i is the i^{th} index of θ . This completes the proof. □

3.3 Calculating $D^2J(\mathcal{T})$

Lemma 2 *Provided every $f_i(\cdot, \cdot)$ in f is C^2 , the second-order switching time derivative of $J(t)$ is*

$$DJ_{T_i, T_j}(\mathcal{T}) = \left[\left[f_i(x(T_i), T_i) - f_{i+1}(x(T_i), T_i) \right]^T \lambda(T_i) + \rho(T_i)^T \left[D_1 f_i(x(T_i), T_i) - D_1 f_{i+1}(x(T_i), T_i) \right] \right] \cdot \Phi(T_i, T_j) \left[f_j(x(T_j), T_j) - f_{j+1}(x(T_j), T_j) \right] \quad (19)$$

when $i \neq j$ and

$$\begin{aligned} &= \left[f_i(x(T_i), T_i) - f_{i+1}(x(T_i), T_i) \right]^T \lambda(T_i) \left[f_i(x(T_i), T_i) - f_{i+1}(x(T_i), T_i) \right] \\ &\quad + \rho(T_i)^T \left[D_1 f_i(x(T_i), T_i) f_i(x(T_i), T_i) - 2 D_1 f_{i+1}(x(T_i), T_i) f_i(x(T_i), T_i) \right. \\ &\quad \left. + D_1 f_{i+1}(x(T_i), T_i) f_{i+1}(x(T_i), T_i) + D_2 f_i(x(T_i), T_i) - D_2 f_{i+1}(x(T_i), T_i) \right] \\ &\quad - D\ell(x(T_i)) \left[f_i(x(T_i), T_i) - f_{i+1}(x(T_i), T_i) \right] \end{aligned} \quad (20)$$

when $i = j$, where $\rho(t)$ is the numerical solution to Eq (12) and $\lambda(t) \in \mathbb{R}^{n \times n}$ is the numerical solution to

$$\begin{aligned} \dot{\lambda}(t) &= -A(t)^T \lambda(t) - \lambda(t) A(t) - D^2 \ell(x(t)) - \sum_{k=1}^n \rho_k(t) D_1^2 f^k(x(t), \mathcal{T}, t) \\ &\text{subject to: } \lambda(T_N) = D^2 m(x(t)). \end{aligned} \quad (21)$$

PROOF. This proof follows from the proof for Lemma 1. We find that the second derivative of $J(\mathcal{T})$, $D^2J(\mathcal{T})$, depends on the second variation of $x(t)$, which we label $\zeta(t)$. The differential equation, $\dot{\zeta}$, is affine linear and therefore may be represented in its integral form according to Eq (10). Using this form, D^2J depends on a double integral, where an adjoint differential equation may be extracted upon switching the order of the integration. The rest of the proof is concerned with the time integral of the x and \mathcal{T} partial derivatives of $f(\cdot, \cdot, \cdot)$.

Let θ^1 and θ^2 be different variations of \mathcal{T} and $z^1(t)$ and $z^2(t)$ be the variations of $x(t)$ corresponding respectively to θ^1 and θ^2 . Then, in order to find $D^2J(\mathcal{T}) \cdot (\theta^1, \theta^2)$, take the switching time derivative of $DJ(\mathcal{T})$ (i.e. Eq (13)):

$$\begin{aligned} \frac{\partial}{\partial \mathcal{T}} (DJ(\mathcal{T}) \cdot \theta^1) &= \frac{\partial}{\partial \mathcal{T}} \left(\int_{T_0}^{T_N} D\ell(x(\tau)) \cdot z^1(\tau) d\tau + Dm(x(T_N)) \cdot z^1(T_N) \right) \\ D^2J(\mathcal{T}) \cdot (\theta^1, \theta^2) + DJ(\mathcal{T}) \cdot \eta &= \int_{T_0}^{T_N} D^2 \ell(x(\tau)) \cdot (z^1(\tau), z^2(\tau)) + D\ell(x(\tau)) \cdot \zeta(\tau) d\tau \\ &\quad + D^2 m(x(T_N)) \cdot (z^1(T_N), z^2(T_N)) + Dm(x(T_N)) \cdot \zeta(T_N) \end{aligned} \quad (22)$$

where η is the second-order variation of \mathcal{T} and $\zeta(t)$ is the second-order variation of $x(t)$. $\dot{\zeta}(t)$ is found by taking the second-order switching time derivative of $\dot{x}(t)$:

$$\begin{aligned} \dot{\zeta}(t) &= \frac{\partial^2}{\partial \mathcal{T}^2} \dot{x}(t) = \frac{\partial}{\partial \mathcal{T}} \dot{z}^1(t) = \frac{\partial}{\partial \mathcal{T}} \left(D_1 f(x(t), \mathcal{T}, t) \cdot z^1(t) + D_2 f(x(t), \mathcal{T}, t) \cdot \theta^1 \right) \\ &= A(t) \cdot \zeta(t) + B(t) \cdot \eta + \left(z^1(t)^T \theta^{1T} \right) \begin{pmatrix} D_{1,2}^2 f(x(t), \mathcal{T}, t) & D_{1,2} f(x(t), \mathcal{T}, t) \\ D_{2,1} f(x(t), \mathcal{T}, t) & D_2^2 f(x(t), \mathcal{T}, t) \end{pmatrix} \begin{pmatrix} z^2(t) \\ \theta^2 \end{pmatrix} \\ &\text{subject to: } \zeta(0) = \frac{\partial^2}{\partial \mathcal{T}^2} x(0) = 0. \end{aligned} \quad (23)$$

Define $C(t) \triangleq \begin{pmatrix} D_1^2 f(x(t), \mathcal{T}, t) & D_{1,2} f(x(t), \mathcal{T}, t) \\ D_{2,1} f(x(t), \mathcal{T}, t) & D_2^2 f(x(t), \mathcal{T}, t) \end{pmatrix}$. Notice that $\dot{\zeta}(t)$ is linear with respect to $\zeta(t)$ and therefore, Eq (23) is of the same form as (9). Thus, $\dot{\zeta}(t)$ has solution

$$\zeta(t) = \int_{T_0}^t \Phi(t, \tau) \left[B(\tau) \cdot \eta + \begin{pmatrix} z^1(\tau)^T & \theta^{1T} \end{pmatrix} C(\tau) \begin{pmatrix} z^2(\tau) \\ \theta^2 \end{pmatrix} \right] d\tau.$$

Plugging $\zeta(t)$ into Eq (22), we see that

$$\begin{aligned} D^2 J(\mathcal{T}) \cdot (\theta^1, \theta^2) + DJ(\mathcal{T}) \cdot \eta &= \int_{T_0}^{T_N} \left[z^1(\tau)^T D^2 \ell(x(\tau)) z^2(\tau) + D\ell(x(\tau)) \int_{T_0}^{\tau} \Phi(\tau, s) \left[B(s) \cdot \eta \right. \right. \\ &\quad \left. \left. + \begin{pmatrix} z^1(s)^T & \theta^{1T} \end{pmatrix} C(s) \begin{pmatrix} z^2(s) \\ \theta^2 \end{pmatrix} \right] ds \right] d\tau \\ &\quad + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N) + Dm(x(T_N)) \int_{T_0}^{T_N} \Phi(T_N, s) \left[B(s) \cdot \eta + \begin{pmatrix} z^1(s)^T & \theta^{1T} \end{pmatrix} C(s) \begin{pmatrix} z^2(s) \\ \theta^2 \end{pmatrix} \right] ds. \end{aligned}$$

Note that $DJ(\mathcal{T}) \cdot \eta$ equals $\int_{T_0}^{T_N} D\ell(x(\tau)) \int_{T_0}^{\tau} \Phi(\tau, s) B(s) \cdot \eta ds d\tau + Dm(x(T_N)) \int_{T_0}^{T_N} \Phi(T_N, s) B(s) \cdot \eta ds$, which is clear from Eq (17). Therefore, this leaves

$$\begin{aligned} D^2 J(\mathcal{T}) \cdot (\theta^1, \theta^2) &= \int_{T_0}^{T_N} \left[z^1(\tau)^T D^2 \ell(x(\tau)) z^2(\tau) + D\ell(x(\tau)) \int_{T_0}^{\tau} \Phi(\tau, s) \begin{pmatrix} z^1(s)^T & \theta^{1T} \end{pmatrix} C(s) \begin{pmatrix} z^2(s) \\ \theta^2 \end{pmatrix} ds \right] d\tau \\ &\quad + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N) + Dm(x(T_N)) \int_{T_0}^{T_N} \Phi(T_N, s) \begin{pmatrix} z^1(s)^T & \theta^{1T} \end{pmatrix} C(s) \begin{pmatrix} z^2(s) \\ \theta^2 \end{pmatrix} ds. \end{aligned}$$

Split the integral over $d\tau$, move $D\ell(x(\tau))$ and $Dm(x(T_N))$ into their respective integrals and switch the order of integration of the double integral:

$$\begin{aligned} &= \int_{T_0}^{T_N} z^1(\tau)^T D^2 \ell(x(\tau)) z^2(\tau) d\tau + \int_{T_0}^{T_N} \int_s^{T_N} D\ell(x(\tau)) \Phi(\tau, s) \begin{pmatrix} z^1(s)^T & \theta^{1T} \end{pmatrix} C(s) \begin{pmatrix} z^2(s) \\ \theta^2 \end{pmatrix} d\tau ds \\ &\quad + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N) + \int_{T_0}^{T_N} Dm(x(T_N)) \Phi(T_N, s) \begin{pmatrix} z^1(s)^T & \theta^{1T} \end{pmatrix} C(s) \begin{pmatrix} z^2(s) \\ \theta^2 \end{pmatrix} ds. \end{aligned}$$

We combine the integrals over ds , and notice that $\rho(\tau)^T$ (from Eq (18)) enters the equations. Furthermore, we switch the dummy variable s to τ and put everything back under one integral:

$$\begin{aligned} &= \int_{T_0}^{T_N} z^1(\tau)^T D^2 \ell(x(\tau)) z^2(\tau) + \rho(\tau)^T \begin{pmatrix} z^1(\tau)^T & \theta^{1T} \end{pmatrix} C(\tau) \begin{pmatrix} z^2(\tau) \\ \theta^2 \end{pmatrix} d\tau \\ &\quad + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N). \end{aligned}$$

Expand $C(\cdot)$ back out,

$$\begin{aligned}
&= \int_{T_0}^{T_N} z^1(\tau)^T D^2 \ell(x(\tau)) z^2(\tau) + \rho(\tau)^T \left[z^1(\tau)^T D_1^2 f(x(\tau), \mathcal{T}, \tau) z^2(\tau) \right] \\
&\quad + \rho(\tau)^T \left[z^1(\tau)^T D_{1,2} f(x(\tau), \mathcal{T}, \tau) \theta^2 \right] + \rho(\tau)^T \left[\theta^{1T} D_{2,1} f(x(\tau), \mathcal{T}, \tau) z^2(\tau) \right] \\
&\quad + \rho(\tau)^T \left[\theta^{1T} D_2^2 f(x(\tau), \mathcal{T}, \tau) \theta^2 \right] d\tau + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N).
\end{aligned}$$

We switch to an index notation where $\rho_k(\cdot)$ is the k^{th} component of $\rho(\cdot)$ and $f^k(\cdot, \cdot, \cdot)$ is the k^{th} component of $f(\cdot, \cdot, \cdot)$. Doing so,

$$\begin{aligned}
&= \int_{T_0}^{T_N} z^1(\tau)^T D^2 \ell(x(\tau)) z^2(\tau) + z^1(\tau)^T \sum_{k=1}^n \rho_k(\tau) D_1^2 f^k(x(\tau), \mathcal{T}, \tau) z^2(\tau) \\
&\quad + z^1(\tau)^T \sum_{k=1}^n \rho_k(\tau) D_{1,2} f^k(x(\tau), \mathcal{T}, \tau) \theta^2 + \theta^{1T} \sum_{k=1}^n \rho_k(\tau) D_{2,1} f^k(x(\tau), \mathcal{T}, \tau) z^2(\tau) \\
&\quad + \theta^{1T} \sum_{k=1}^n \rho_k(\tau) D_2^2 f^k(x(\tau), \mathcal{T}, \tau) \theta^2 d\tau + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N).
\end{aligned}$$

Rearrange the terms allows $D^2 J(\mathcal{T}) \cdot (\theta^1, \theta^2)$ to be partitioned into the summation of three parts \mathcal{P}_1 , \mathcal{P}_2 and \mathcal{P}_3 as specified below

$$\begin{aligned}
&= \underbrace{\int_{T_0}^{T_N} z^1(\tau)^T \left[D^2 \ell(x(\tau)) + \sum_{k=1}^n \rho_k(\tau) D_1^2 f^k(x(\tau), \mathcal{T}, \tau) \right] z^2(\tau) d\tau + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N)}_{\mathcal{P}_1} \\
&\quad + \underbrace{\int_{T_0}^{T_N} \theta^{2T} \sum_{k=1}^n \rho_k(\tau) D_{2,1} f^k(x(\tau), \mathcal{T}, \tau) z^1(\tau) + \theta^{1T} \sum_{k=1}^n \rho_k(\tau) D_{2,1} f^k(x(\tau), \mathcal{T}, \tau) z^2(\tau) d\tau}_{\mathcal{P}_2} \\
&\quad + \underbrace{\int_{T_0}^{T_N} \theta^{1T} \sum_{k=1}^n \rho_k(\tau) D_2^2 f^k(x(\tau), \mathcal{T}, \tau) \theta^2 d\tau}_{\mathcal{P}_3}.
\end{aligned} \tag{24}$$

First, let us examine \mathcal{P}_1 . Let $g(\tau) = D^2 \ell(x(\tau)) + \sum_{k=1}^n \rho_k(\tau) D_1^2 f^k(x(\tau), \mathcal{T}, \tau)$. Then,

$$\mathcal{P}_1 = \int_{T_0}^{T_N} z^1(\tau)^T g(\tau) z^2(\tau) d\tau + z^1(T_N)^T D^2 m(x(T_N)) z^2(T_N).$$

Plugging Eq (16) in for $z(\cdot)$, results in

$$\begin{aligned}
&= \int_{T_0}^{T_N} \left[\int_{T_0}^{\tau} \Phi(\tau, s) B(s) \theta^1 ds \right]^T g(\tau) \int_{T_0}^{\tau} \Phi(\tau, w) B(w) \theta^2 dw d\tau \\
&\quad + \left[\int_{T_0}^{T_N} \Phi(T_N, s) B(s) \theta^1 ds \right]^T D^2 m(x(T_N)) \int_{T_0}^{T_N} \Phi(T_N, w) B(w) \theta^2 dw.
\end{aligned}$$

The integrals may be specified as follows:

$$\begin{aligned}
&= \int_{T_0}^{T_N} \int_{T_0}^{\tau} \int_{T_0}^{\tau} \theta^{1^T} B(s)^T \Phi(\tau, s)^T g(\tau) \Phi(\tau, w) B(w) \theta^2 \, ds \, dw \, d\tau \\
&\quad + \int_{T_0}^{T_N} \int_{T_0}^{T_N} \theta^1 B(s)^T \Phi(T_N, s)^T D^2 m(x(T_N)) \Phi(T_N, w) B(w) \theta^2 \, ds \, dw.
\end{aligned}$$

Note that the volume of the triple integral is given by $\tau = \max(s, w)$. Therefore, the order of integration may be switched to:

$$\begin{aligned}
&= \int_{T_0}^{T_N} \int_{T_0}^{T_N} \int_{\max(s, w)}^{T_N} \theta^{1^T} B(s)^T \Phi(\tau, s)^T g(\tau) \Phi(\tau, w) B(w) \theta^2 \, d\tau \, ds \, dw \\
&\quad + \int_{T_0}^{T_N} \int_{T_0}^{T_N} \theta^1 B(s)^T \Phi(T_N, s)^T D^2 m(x(T_N)) \Phi(T_N, w) B(w) \theta^2 \, ds \, dw.
\end{aligned}$$

We combine the double integral with the triple integral and rearrange the terms so that only the ones depending on τ are inside the internal integral:

$$= \int_{T_0}^{T_N} \int_{T_0}^{T_N} B(s)^T \left[\int_{\max(s, w)}^{T_N} \Phi(\tau, s)^T g(\tau) \Phi(\tau, w) \, d\tau + \Phi(T_N, s)^T D^2 m(x(T_N)) \Phi(T_N, w) \right] B(w) \, ds \, dw \cdot (\theta^1, \theta^2).$$

Let

$$\lambda(t) = \int_t^{T_N} \Phi(\tau, t)^T g(\tau) \Phi(\tau, t) \, d\tau + \Phi(T_N, t)^T D^2 m(x(T_N)) \Phi(T_N, t)$$

where $\lambda(t) \in \mathbb{R}^{n \times n}$ is the integral curve to the following differential equation

$$\begin{aligned}
\dot{\lambda}(t) &= -A(t)^T \lambda(t) - \lambda(t) A(t) - g(t) \\
&= -A(t)^T \lambda(t) - \lambda(t) A(t) - D^2 \ell(x(t)) - \sum_{k=1}^n \rho_k(t) D_1^2 f^k(x(t), \mathcal{T}, t) \\
&\quad \text{subject to: } \lambda(T_N) = D^2 m(x(T_N)).
\end{aligned}$$

Then, depending on whether $s > w$, $s < w$ or $s = w$, $\lambda(\cdot)$ enters into Eq (25) as shown:

$$\mathcal{P}_1 = \begin{cases} \int_{T_0}^{T_N} \int_{T_0}^{T_N} B(s)^T \lambda(s) \Phi(s, w) B(w) \, ds \, dw \cdot (\theta^1, \theta^2) & \text{when } s > w \\ \int_{T_0}^{T_N} \int_{T_0}^{T_N} B(s)^T \Phi(w, s)^T \lambda(w) B(w) \, ds \, dw \cdot (\theta^1, \theta^2) & \text{when } s < w \\ \int_{T_0}^{T_N} \int_{T_0}^{T_N} B(s)^T \lambda(s) B(w) \, ds \, dw \cdot (\theta^1, \theta^2) & \text{when } s = w. \end{cases} \quad (25)$$

Noting that \mathcal{P}_1 is scalar, transpose the $s < w$ case such that

$$\mathcal{P}_1 \stackrel{s < w}{=} \int_{T_0}^{T_N} \int_{T_0}^{T_N} B(w)^T \lambda(w) \Phi(w, s) B(s) \, ds \, dw \cdot (\theta^2, \theta^1).$$

Use i and j to index θ^1 and θ^2 respectively, where $i, j = 1, \dots, N-1$. Now, integrating the δ -functions in $B(s)$ and $B(w)$ will pick out times $s = T_i$ and $w = T_j$ such that

$$\mathcal{P}_{1_{ij}} = \begin{cases} \left[f_i(x(T_i), t) - f_{i+1}(x(T_i), t) \right]^T \lambda(T_i) \Phi(T_i, T_j) \left[f_j(x(T_j), t) - f_{j+1}(x(T_j), t) \right] \cdot (\theta_i^1, \theta_j^2) & \text{when } i > j \\ \left[f_j(x(T_j), t) - f_{j+1}(x(T_j), t) \right]^T \lambda(T_j) \Phi(T_j, T_i) \left[f_i(x(T_i), t) - f_{i+1}(x(T_i), t) \right] \cdot (\theta_i^1, \theta_j^2) & \text{when } i < j \\ \left[f_i(x(T_i), t) - f_{i+1}(x(T_i), t) \right]^T \lambda(T_i) \left[f_i(x(T_i), t) - f_{i+1}(x(T_i), t) \right] \cdot (\theta_i^1, \theta_j^2) & \text{when } i == j \end{cases} \quad (26)$$

Note that the commutative property holds for \mathcal{P}_1 .

Now for \mathcal{P}_2 . Start by calculating $D_{2,1}f^k(x(t), \mathcal{T}, t)$:

$$D_{2,1}f^k(x(t), \mathcal{T}, t) = \left\{ \delta(t - T_a^-) D_1 f_a^k(x(t), t)^T - \delta(t - T_a^+) D_1 f_{a+1}^k(x(t), t)^T \right\}_{a=1}^{N-1}.$$

Once again, choose the i^{th} index of θ^1 and the j^{th} index of θ^2 where $i, j = 1, \dots, N-1$. This corresponds to the i^{th} index of $z^1(t)$ and the j^{th} index of $z^2(t)$, where the k^{th} index of $z^{\cdot}(\cdot)$ is

$$z_k^i(t) = \int_{T_0}^t \Phi(t, \tau) \left[\delta(\tau - T_k^-) f_k(x(\tau), \tau) - \delta(\tau - T_k^+) f_{k+1}(x(\tau), \tau) \right] d\tau \theta_k. \quad (27)$$

Specifying these indexes allow us to revert back to matrix representation for $\rho(\cdot)$ and $f(\cdot, \cdot, \cdot)$. Thus,

$$\begin{aligned} \mathcal{P}_{2_{ij}} = & \int_{T_0}^{T_N} \theta_j^2 \rho(\tau)^T \left[\delta(\tau - T_j^-) D_1 f_j(x(\tau), \tau) - \delta(\tau - T_j^+) D_1 f_{j+1}(x(\tau), \tau) \right] z_i^1(\tau) \\ & + \theta_i^1 \rho(\tau)^T \left[\delta(\tau - T_i^-) D_1 f_i(x(\tau), \tau) - \delta(\tau - T_i^+) D_1 f_{i+1}(x(\tau), \tau) \right] z_j^2(\tau) d\tau. \end{aligned} \quad (28)$$

Integrating over the δ -functions picks out the times for which the δ -functions' arguments are zero:

$$\begin{aligned} = & \theta_j^2 \rho(T_j^-)^T D_1 f_j(x(T_j^-), T_j^-) z_i^1(T_j^-) - \theta_j^2 \rho(T_j^+)^T D_1 f_{j+1}(x(T_j^+), T_j^+) z_i^1(T_j^+) \\ & + \theta_i^1 \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) z_j^2(T_i^-) - \theta_i^1 \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) z_j^2(T_i^+). \end{aligned}$$

The indexes i and j relate in three possible ways. Either $i > j$, $i < j$, or $i == j$. Let us start with the case $i > j$:

Recall that \mathcal{T} is a set of monotonically increasing times. Therefore, if $i > j$, then $T_i > T_j$. Furthermore, by referencing Eq (27), observe that $z_i^1(t)$ is non-zero only after time $t = T_i^-$ due to the δ -functions. Consequently, $z_i^1(T_j) = 0$. Therefore,

$$\stackrel{i > j}{=} \theta_i^1 \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) z_j^2(T_i^-) - \theta_i^1 \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) z_j^2(T_i^+).$$

Omitting the $-$ and $+$ superscripts for they are no longer helpful, we see that

$$\stackrel{i > j}{=} \theta_i^1 \rho(T_i)^T \left[D_1 f_i(x(T_i), T_i) - D_1 f_{i+1}(x(T_i), T_i) \right] z_j^2(T_i).$$

Plugging Eq (27) in for $z_j^2(T_i)$ results in

$$\stackrel{i > j}{=} \theta_i^1 \rho(T_i)^T \left[D_1 f_i(x(T_i), T_i) - D_1 f_{i+1}(x(T_i), T_i) \right] \int_{T_0}^{T_i} \Phi(T_i, \tau) \delta(t - T_j) \left[f_j(x(\tau), \tau) - f_{j+1}(x(\tau), \tau) \right] d\tau \theta_j^2.$$

The integration over the δ -function results in:

$$\mathcal{P}_{2_{ij}} \stackrel{i > j}{=} \rho(T_i)^T \left[D_1 f_i(x(T_i), T_i) - D_1 f_{i+1}(x(T_i), T_i) \right] \Phi(T_i, T_j) \left[f_j(x(T_j), T_j) - f_{j+1}(x(T_j), T_j) \right] \cdot (\theta_i^1, \theta_j^2). \quad (29)$$

For the $i < j$ case, an equivalent process will reveal that the i s and j s switch places, which is to be expected due to the commutative property of mixed partials. Accordingly,

$$\mathcal{P}_{2_{ij}} \stackrel{i < j}{=} \rho(T_j)^T \left[D_1 f_j(x(T_j), T_j) - D_1 f_{j+1}(x(T_j), T_j) \right] \Phi(T_j, T_i) \left[f_i(x(T_i), T_i) - f_{i+1}(x(T_i), T_i) \right] \cdot (\theta_i^1, \theta_j^2).$$

Finally, we analyze the $i = j$ case. First, note that because $i = j$, the perturbations θ_i^1 and θ_j^2 are equivalent and therefore, $z_i^1(t)$ and $z_j^2(t)$ are equivalent. Therefore, Eq (28) for this case becomes

$$\mathcal{P}_{2_{ij}} \stackrel{i = j}{=} 2 \theta_i^2 \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) z_i^1(T_i^-) - 2 \theta_i^2 \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) z_i^1(T_i^+).$$

Plug Eq (27) in for the $z_i^1(\cdot)$ terms

$$\begin{aligned} \stackrel{i = j}{=} 2 \theta_i^2 \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) \int_{T_0}^{T_i^-} \Phi(T_i^-, \tau) \left[\delta(\tau - T_i^-) f_i(x(\tau), \tau) - \delta(\tau - T_i^+) f_{i+1}(x(\tau), \tau) \right] d\tau \theta_i^1 \\ - 2 \theta_i^2 \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) \int_{T_0}^{T_i^+} \Phi(T_i^+, \tau) \left[\delta(\tau - T_i^-) f_i(x(\tau), \tau) - \delta(\tau - T_i^+) f_{i+1}(x(\tau), \tau) \right] d\tau \theta_i^1. \end{aligned}$$

Again, we integrate over the δ -functions, but, unlike before, we must be careful because the times for which two of the δ -function's arguments are zero is at the upper bounds of their integrals. Thus,

$$\begin{aligned} \stackrel{i = j}{=} 2 \theta_i^2 \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) \frac{1}{2} \Phi(T_i^-, T_i^-) f_i(x(T_i^-), T_i^-) \theta_i^1 \\ - 2 \theta_i^2 \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) \left[\Phi(T_i^+, T_i^-) f_i(x(T_i^-), T_i^-) - \frac{1}{2} \Phi(T_i^+, T_i^+) f_{i+1}(x(T_i^+), T_i^+) \right] \theta_i^1. \end{aligned}$$

Recall that $\Phi(T_i^-, T_i^-) = \Phi(T_i^+, T_i^+) = I$ and note that $\Phi(\cdot, \cdot)$ is a continuous operator, so $\Phi(T_i^+, T_i^-) = I$. Therefore, while omitting the no longer helpful $-$ and $+$ super-scripts,

$$\begin{aligned} \mathcal{P}_{2_{ij}} \stackrel{i = j}{=} \rho(T_i)^T \left[D_1 f_i(x(T_i), T_i) f_i(x(T_i), T_i) - 2 D_1 f_{i+1}(x(T_i), T_i) f_i(x(T_i), T_i) \right. \\ \left. + D_1 f_{i+1}(x(T_i), T_i) f_{i+1}(x(T_i), T_i) \right] \cdot (\theta_i^1, \theta_i^2). \quad (30) \end{aligned}$$

Finally, we are left with \mathcal{P}_3 . Again, index θ^1 with i and θ^2 with j where $i, j = 1, \dots, N - 1$. Starting with the ij^{th} component of $D_2^2 f^k(x(\tau), \mathcal{T}, \tau)$,

$$D_2^2 f^k(x(\tau), \mathcal{T}, \tau)_{ij} = \begin{cases} \left(\frac{\partial}{\partial T_i} \delta(\tau - T_i^-) \right) f_i^k(x(\tau), \tau) - \left(\frac{\partial}{\partial T_i} \delta(\tau - T_i^+) \right) f_{i+1}^k(x(\tau), \tau) & , i = j \\ 0 & , i \neq j \end{cases}$$

Revert back to matrix representation for $\rho(\cdot)$ and $f(\cdot, \cdot)$. Clearly, when $i \neq j$, $\mathcal{P}_{3_{ij}} \stackrel{i \neq j}{=} 0$. For the $i = j$ case, conducting chain rule on $D_2^2 f^k(x(\tau), \mathcal{T}, \tau)_{ij}$ results in:

$$D_2^2 f^k(x(\tau), \mathcal{T}, \tau)_{ij} \stackrel{i = j}{=} -\dot{\delta}(\tau - T_i^-) f_i^k(x(\tau), \tau) + \dot{\delta}(\tau - T_i^+) f_{i+1}^k(x(\tau), \tau).$$

Then, \mathcal{P}_3 becomes

$$\mathcal{P}_{3_{ij}} \stackrel{i==j}{=} \int_{T_0}^{T_N} \left[-\rho(\tau)^T \delta(\tau - T_i^-) f_i(x(\tau), \tau) + \rho(\tau)^T \delta(\tau - T_i^+) f_{i+1}(x(\tau), \tau) \right] d\tau \cdot (\theta_i^1, \theta_i^2).$$

Conducting integration by parts, we see that

$$\begin{aligned} & \stackrel{i==j}{=} \left[-\rho(\tau)^T \delta(\tau - T_i^-) f_i(x(\tau), \tau) \Big|_{T_0}^{T_N} + \rho(\tau)^T \delta(\tau - T_i^+) f_{i+1}(x(\tau), \tau) \Big|_{T_0}^{T_N} \right. \\ & + \int_{T_0}^{T_N} \left[\dot{\rho}(\tau)^T f_i(x(\tau), \tau) + \rho(\tau)^T D_1 f_i(x(\tau), \tau) \dot{x}(t) + \rho(\tau)^T D_2 f_i(x(\tau), \tau) \right] \delta(\tau - T_i^-) d\tau \\ & \left. - \int_{T_0}^{T_N} \left[\dot{\rho}(\tau)^T f_{i+1}(x(\tau), \tau) + \rho(\tau)^T D_1 f_{i+1}(x(\tau), \tau) \dot{x}(t) + \rho(\tau)^T D_2 f_{i+1}(x(\tau), \tau) \right] \delta(\tau - T_i^+) d\tau \right] \cdot (\theta_i^1, \theta_i^2). \end{aligned}$$

Integrating over the δ -functions picks out the times for which the δ -functions' arguments are zero:

$$\begin{aligned} & \stackrel{i==j}{=} \left[\dot{\rho}(T_i^-)^T f_i(x(T_i^-), T_i^-) - \dot{\rho}(T_i^+)^T f_{i+1}(x(T_i^+), T_i^+) \right. \\ & \quad + \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) \dot{x}(T_i^-) - \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) \dot{x}(T_i^+) \\ & \quad \left. + \rho(T_i^-)^T D_2 f_i(x(T_i^-), T_i^-) - \rho(T_i^+)^T D_2 f_{i+1}(x(T_i^+), T_i^+) \right] \cdot (\theta_i^1, \theta_i^2). \end{aligned}$$

Plugging Eq (12) in for $\dot{\rho}(\cdot)$, results in

$$\begin{aligned} & \stackrel{i==j}{=} \left[\left[-\rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) - D\ell(x(T_i^-)) \right] f_i(x(T_i^-), T_i^-) \right. \\ & \quad - \left[-\rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) - D\ell(x(T_i^+)) \right] f_{i+1}(x(T_i^+), T_i^+) \\ & \quad + \rho(T_i^-)^T D_1 f_i(x(T_i^-), T_i^-) f_i(x(T_i^-), T_i^-) - \rho(T_i^+)^T D_1 f_{i+1}(x(T_i^+), T_i^+) f_{i+1}(x(T_i^+), T_i^+) \\ & \quad \left. + \rho(T_i^-)^T D_2 f_i(x(T_i^-), T_i^-) - \rho(T_i^+)^T D_2 f_{i+1}(x(T_i^+), T_i^+) \right] \cdot (\theta_i^1, \theta_i^2) \end{aligned}$$

Finally, simplifying reveals

$$\begin{aligned} \mathcal{P}_{3_{ij}} \stackrel{i==j}{=} & -D\ell(x(T_i)) \left[f_i(x(T_i), T_i) - f_{i+1}(x(T_i), T_i) \right] \\ & + \rho(T_i)^T \left[D_2 f_i(x(T_i), T_i) - D_2 f_{i+1}(x(T_i), T_i) \right]. \end{aligned} \tag{31}$$

Recall from Eq (24) that $D^2 J(\mathcal{T}) \cdot (\theta^1, \theta^2) = \mathcal{P}_1 + \mathcal{P}_2 + \mathcal{P}_3$, which is the summation of Eqs (26) and (29) for the case when $i \neq j$ and is the summation of Eqs (26), (29) and (31) for the case when $i = j$.

□

3.4 Steepest Descent and Newton's Method Algorithms

With Lemmas 1 and 2, we can thus compile optimization algorithms using the first- and second-order descent directions of steepest descent and Newton's method. Before doing so, however, we provide a few remarks important for implementation.

Remark: A few notes for applying steepest descent and Newton's method algorithms

- (1) One calculation of $\rho(t)$ and $\lambda(t)$ from T_N to T_0 suffices to fully specify all components of $DJ(\mathcal{T})$ and $D^2J(\mathcal{T})$.
- (2) When calculating $\Phi(T_i, T_j)$, calculate $\Phi(T_k, T_{k+1})$ for $k = 1, \dots, N - 1$ once and keep in memory. Then, $\Phi(T_i, T_j) = \Phi(T_i, T_{i-1})\Phi(T_{i-1}, T_{i-2}) \cdots \Phi(T_{j+1}, T_j)$, thus reducing the total number of calculations to $N - 1$. Moreover, each $\Phi(T_k, T_{k+1})$ is solved over a distinct time interval, where the union of these time intervals spans $[T_0, T_N]$. Therefore, the total time over which all the $\Phi(T_k, T_{k+1})$ is solved is not dependent on the number of switching modes, N .
- (3) The computational cost of $D^2J(\mathcal{T})$ is its differential equations (DEs). Complexity may be discussed with regard to the number of DEs as well as the number of full time intervals the DEs are solved over. Following from Remarks 1 and 2, \dot{x} and $\dot{\rho}$ constitute $2 \cdot (n \times 1)$ DEs, whereas $\dot{\lambda}$ and the relevant $\Phi(\cdot, \cdot)$ constitute $N \cdot (n \times n)$ DEs for a total of $N + 2$ DEs. However, since the collection of relevant $\Phi(\cdot, \cdot)$ are over one full time interval (from Remark 2), the total number of full time intervals over which all the DEs are specified is independent on the number of switching modes, N .
- (4) The second-order term for Newton's method must be positive definite (i.e. $D^2J(\mathcal{T}) > 0$) for the descent direction d to be pointing in a negative direction. This means the second-order descent direction does not provide useful information globally. Therefore, our Newton's method algorithm is designed to do an initial number of steepest descent steps before performing Newton's method and to furthermore do a step of steepest descent instead of a step of Newton's method if for that step, $D^2J(\mathcal{T})$ is not positive definite. Therefore, while this means our Newton's method algorithm is not true Newton's method, it functions globally.
- (5) Although not represented in the algorithms we present, it may be beneficial to remove a mode from Ψ if it only exists for a short time interval. This concept is explained in [13,30].

The steepest descent algorithm is in Alg 1 and the Newton's method algorithm is in Alg 2:

Algorithm 1 Steepest Descent: $sd()$

Arguments:

- \mathcal{T}_0 is initial guess
- k_{max} is maximum # of steps
- ϵ is convergence criteria tolerance

$\mathcal{T}^* = sd(\mathcal{T}_0, k_{max}, \epsilon)$:

while $DJ(\mathcal{T}_k) > \epsilon$ and $k < k_{max}$ **do**

- Calculate $x(t)$ from Eq (2)
- Calculate $\rho(t)$ from Eq (12)
- Calculate $d_k = -DJ(\mathcal{T}_k)$ using Eq (11)
- Find smallest m such that Eq (8), the Sufficient Decrease Condition is satisfied
- $\gamma = \beta^m$
- $\mathcal{T}_{k+1} = \mathcal{T}_k + \gamma d_k$
- $k = k + 1$

end while

$\mathcal{T}^* = \mathcal{T}_k$

4 Relaxed Switched System Mode Sequence Generation

The previous section provides techniques for optimizing the switching times of a switched system under the assumption that an intelligent mode sequence is known. This section removes the assumption by presenting a technique to estimate a mode sequence.

We recall the third definition of the switched system, Eq (4), presented in Section 3 and we reiterate the constraints on the control inputs $u(t) \in \mathcal{U}$:

- (1) $u_1(t), u_2(t), \dots, u_{\widehat{N}}(t) \in \{0, 1\}$
- (2) $\sum_{i=1}^{\widehat{N}} u_i(t) = 1.$

Algorithm 2 Newton's Method: $nm()$

Arguments:

\mathcal{T}_0 is initial guess
 sd_{init} is the # of initial steps of steepest descent
 k_{max} is maximum # of steps
 ϵ is convergence criteria tolerance

$\mathcal{T}^* = nm(\mathcal{T}_0, sd_{init}, k_{max}, \epsilon):$

$\mathcal{T}_k = sd(\mathcal{T}_0, sd_{init}, \epsilon)$
while $DJ(\mathcal{T}_k) > \epsilon$ and $k < k_{max}$ **do**
 Calculate $x(t)$ from Eq (2)
 Calculate $\rho(t)$ from Eq (12)
 Calculate $\lambda(t)$ from Eq (21)
 Calculate $D^2J(\mathcal{T}_k)$ using Eqs (19) and (20)
if $D^2J(\mathcal{T}_k) > 0$ **then**
 Calculate $d_k = -\left(D^2J(\mathcal{T}_k)\right)^{-1} \cdot DJ(\mathcal{T}_k)$ using Eq (11)
 $\mathcal{T}_{k+1} = \mathcal{T}_k + d_k$
else
 $\mathcal{T}_{k+1} = sd(\mathcal{T}_k, 1, \epsilon)$
end if
 $k = k + 1$
end while
 $\mathcal{T}^* = \mathcal{T}_k$

The second constraint is satisfied by setting one of the inputs to be one minus the summation of all other inputs:

$$u_{\widehat{N}}(t) = 1 - \sum_{i=1}^{\widehat{N}-1} u_i(t).$$

As for the first constraint, let us remove it, thereby relaxing the space \mathcal{U} . This is but one possible relaxation of \mathcal{U} . It was chosen because it simplifies the problem such that standard infinite dimensional optimization routines may be implemented as we shall see. Other relaxations with different properties may also be used.

With this relaxation on \mathcal{U} , instead of modes being active (i.e. $u_i = 1$) or inactive (i.e. $u_i = 0$), all modes are now active to some degree where the extent of activeness is specified by $u(\cdot)$. Let us designate the relaxed control inputs by $\widehat{u}(t)$ and the relaxed space as $\widehat{\mathcal{U}}$. Then, the relaxed switched system is specified by the evolution of the relaxed state trajectory, $\widehat{x}(t) : \mathbb{R} \mapsto \mathbb{R}^n$:

$$\begin{aligned} \dot{\widehat{x}}(t) &= \widehat{f}\left(\widehat{x}(t), \widehat{u}(t), t\right) = \widehat{u}_1(t) f_{\sigma=1}\left(\widehat{u}(t), t\right) + \cdots + \widehat{u}_{\widehat{N}-1}(t) f_{\sigma=\widehat{N}-1}\left(\widehat{x}(t), t\right) + \left(1 - \sum_{i=1}^{\widehat{N}-1} \widehat{u}_i(t)\right) f_{\sigma=\widehat{N}}\left(\widehat{x}(t), t\right) \quad (32) \\ &\text{subject to: } \widehat{x}(T_0) = x_0. \end{aligned}$$

Therefore, the optimization is over $\widehat{u}(t)$:

$$\arg \min_{\widehat{u}} J(\widehat{u}) \quad (33)$$

where $J(\widehat{u})$ is

$$J(\widehat{u}) = \int_{T_0}^{T_N} \ell\left(\widehat{x}(\tau), h(\tau), \widehat{u}(\tau), \widehat{u}_d(\tau)\right) d\tau + m\left(\widehat{x}(T_N), h(T_N)\right). \quad (34)$$

and $\widehat{u}_d(\cdot)$ are the desired relaxed control inputs. The optimization problem Eq (33) is an infinite dimensional one constrained to Eq (32). The optimization may be accomplished with the trajectory functional optimization approach using a projection operator, as seen in [16]. Alternatively, any other optimal control technique may be employed.

Once the optimal \hat{u}^* is found, a mode sequence may be generated by projecting the corresponding optimal system onto the set of feasible solutions such that $\pi : \hat{\mathcal{U}} \rightarrow \mathcal{U}$. The projection we use for any time t is

$$u(t) = \pi(\hat{u}(t)) = \begin{cases} 1 & \text{if } \hat{u}_i(t) > \hat{u}_j(t) \forall j \neq i \\ 0 & \text{else.} \end{cases}$$

Under this projection, the most active mode at any given time (i.e. the one corresponding to the greatest \hat{u}) is set as the single active mode and the rest are set as inactive, thereby satisfying the constraints on u once more. Furthermore, note that $\pi(\pi(\hat{u})) = \pi(\hat{u})$ and therefore, π is in fact a projection.

The estimated mode sequence, Ψ , is thus the sequence of most active modes. Furthermore, the collection of times for which the most active modes transition may be used as the initial switching times, \mathcal{T}_0 , such that the pair (\mathcal{T}_0, Ψ) are initial parameters for switching time optimization.

Before continuing, a few remarks must be made to clarify the proposed method.

Remark: A few notes on the relaxed switched system mode estimation:

- (1) The mode estimation routine presented in this section is generally not optimal because $u^* \neq \pi(\hat{u}^*)$. However, the solution of the routine may be described as sub-optimal, in that \hat{u}^* is a local optimum of the relaxed problem and that $\pi(\hat{u}^*)$ approximates u^* .
- (2) As future work, we propose two extensions to the sub-optimal routine that would instead find the *optimal* switching times and *optimal* mode sequence.
 - (a) The optimal switching times and mode sequence may be found by solving the problem $\arg \min_u J(u)$. An equivalent problem is

$$\arg \min_{\pi(\hat{u})} J(\pi(\hat{u})). \tag{35}$$

Applying the Pontryagin Maximum Principle to this problem results in an optimality condition that is dependent on $\pi(\hat{u}(t))$ and $D\pi(\hat{u}(t))$. Now, suppose α is a convergent sequence of approximations of $\pi(\hat{u}(t))$ and $D\pi(\hat{u}(t))$. Then, an optimization problem may be solved for each approximation, such that the initial optimization problem is the one presented in this paper and each subsequent optimization uses the results from the previous optimization as its initial conditions.

- (b) For the second extension, each descent step to the optimal solution of Eq (35) would consist of searching for a descent direction in $\hat{\mathcal{U}}$, taking a step in that direction and projecting the result back onto \mathcal{U} . The approach hinges on finding a relaxation and projection operator such that a reasonably sized descent step may be taken. Better choices for the relaxation and projection operator than the ones presented in this paper may well be necessary.
- (3) One may be concerned with the uniqueness of the relaxed optimization problem, Eq (33). Uniqueness is an important consideration when designing the cost function, Eq (34). For instance, suppose a system has two or more modes that are equivalent under certain circumstances. Then, if the cost function is designed poorly, a unique optimum may not exist. Take the SSV for example. If the SSV is moving purely forward and it has a zero wheel torque difference, then all four modes are equivalent. In this case, the designer could designate a “preferred” mode through a choice of the desired control inputs \hat{u}_d . In the examples in Section 5, we choose \hat{u}_d to be 1 for $\sigma = 1$ and 0 for the other modes, thereby making the mode choice $\sigma = 1$ cost the least.
- (4) Alamir and Attia, in [2] and [6], present an approach for optimizing the mode sequence and switching times that is closely related to the one in this section. They too begin with Eq (4) with the goal of optimizing over the control inputs $u(t)$. However, their routine to do the optimization significantly differs from ours. Where we optimize in a relaxed space, they optimize in discrete time. While our approach is flexible to integration schemes, such that the time stepping may be chosen independently of the optimization, their approach is restricted to the specified discretization scheme. The approach in this paper is an alternative to the approach presented by Alamir and Attia.

5 Examples

We present three examples simulated in Mathematica. For all examples, the SSV has dimensions $a = 0.16\text{m}$, $b = 0.28\text{m}$, $B_l = 0.8\text{m}$, and $B_w = 0.6\text{m}$, and masses $m_B = 70\text{kg}$ and $m_w = 2.5\text{kg}$ (refer to Fig 2 and Eq (1)). The coefficient of kinetic friction is assumed to have a mean of 0.8, so μ_K is set to 0.8.

The first example demonstrates how to apply switching time optimization to determine the optimal intervals of the skid sub-systems during some SSV maneuver given some mode sequence Ψ . The example also demonstrates the difference between the convergence rates of steepest descent and Newton’s method. The measured configuration trajectories are simulated assuming a perfect model of the SSV and perfect sensors.

The second example demonstrates how to optimize the relaxed switched system in order to estimate the mode sequence, Ψ , and initial switching times, \mathcal{T}_0 , such that the pair (\mathcal{T}_0, Ψ) may be used for the switching time optimization. The measured trajectories are equivalent to those in the first example.

The third example demonstrates how the algorithms perform in the presence of sensor and model disturbances. This example optimizes the relaxed switched system to find Ψ and T_0 and uses these parameters in the switching time optimization to find the optimal switching times T^* .

5.1 Example 1

The SSV is driven with the wheel torques shown in Fig 4 and initial state $x(0) = 0$. Initially, none of the wheels skid laterally (i.e. initially $\sigma = 1$). At some time shortly after $t = 5s$, the difference in torques becomes large enough to break the lateral static friction holding it in sub-system $\sigma = 1$ and the vehicle begins to turn left. Granted, during the turn, it is not clear which sub-system the SSV switches into due to the torque difference, but let us assume the vehicle switches into the all wheels skidding laterally sub-system, $\sigma = 4$. At some time after $t = 10.5s$, the vehicle returns to $\sigma = 1$ due to the 0 torque difference no longer injecting energy into the wheel’s lateral direction and because of the energy dissipation from skidding. Therefore, let us assume for this example that $\Psi = (1, 4, 1)$.

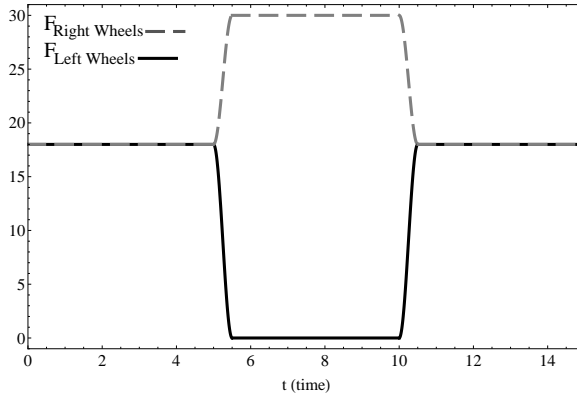


Fig. 4. The applied torques to the left and right wheels over the time interval.

Normally, the measured trajectories, $h(t)$, are measured from data collected by sensors such as a GPS. For now, we simulate $h(t)$, but in future work, we will run the algorithms presented in this paper on the SSV pictured in Fig 2 using sensors. For this example and Example 2, we assume our SSV model is perfect and that it transitions between the modes $\Psi = (1, 4, 1)$ at desired switching times $\mathcal{T}^d = (5, 11)^T$. Therefore, $h(t)$ is simulated using these parameters. Since the model is assumed perfect, steepest descent and Newton’s method will discover \mathcal{T}^d to be the optimal switching times.

We use a quadratic performance index for the cost function. Therefore, Eq (5) has $\ell(\cdot, \cdot)$ and $m(\cdot, \cdot)$ equal to $\ell(x(\tau), h(\tau)) = 1/2(x(\tau) - h(\tau))^T Q(x(\tau) - h(\tau))$ and $m(x(T_N), h(T_N)) = 1/2(x(T_N) - h(T_N))^T P(x(T_N) - h(T_N))$ where Q and P are symmetric semi-positive definite (i.e. $Q^T = Q \geq 0$ and $P^T = P \geq 0$). We let the state vector be $x(t) = (\dot{X}, X, \dot{Y}, Y, \dot{\theta}, \theta)^T(t)$ and set Q and P to $Q = \text{diag}(1, 1, 1, 1, 1, 1)$ and $P = \text{diag}(1, 10, 1, 10, 1, 1)$. The influence of the state variables on the cost function may be altered by tweaking Q and P .

Concerning the initial setup of the descent algorithms, we shall assume Ψ is already known or “guessed” from the wheel torques (Fig 4) to be $\Psi = (1, 4, 1)$. As for the initial switching times, an intelligent “guess” is $\mathcal{T}_0 = (5, 10.5)^T$, again, from the wheel torques, but for the purpose of demonstration, we arbitrarily choose initial switching times $\mathcal{T}_0 = (3, 13)^T$.

As for the descent algorithms, we use Armijo Rule parameters $\alpha = \beta = 0.4$. Calling Algorithm 1 with $sd(\mathcal{T}_0, 1000, 10^{-5})$ results in the successful convergence of the steepest descent algorithm to $\mathcal{T}^* = \mathcal{T}^d = (5, 11)^T$ in 24 steps. Now, calling Algorithm 2 with $nm(\mathcal{T}_0, 6, 1000, 10^{-5})$ this also results in the successful convergence of the algorithm to $\mathcal{T}^* = \mathcal{T}^d = (5, 11)^T$, but the benefit of quadratic convergence results in 6 initial steps of steepest descent followed by only 3 steps of Newton's method. In short, this means that after the first 6 steps of steepest descent, Algorithm 2 converges in 15 fewer steps using Newton's method than Algorithm 1 using steepest descent.

For further illustration, view Figs 1 and 5. Fig 1 depicts the convergences of steepest descent and Newton's method to $DJ(\mathcal{T}) = 0$ for this example. Fig 5 is a plot of the X-Y trajectories during the 15 second interval for each step of Algorithm 2. The sequence of plots in Fig 5 show the convergence of the simulated X-Y trajectories to the measured X-Y trajectories.

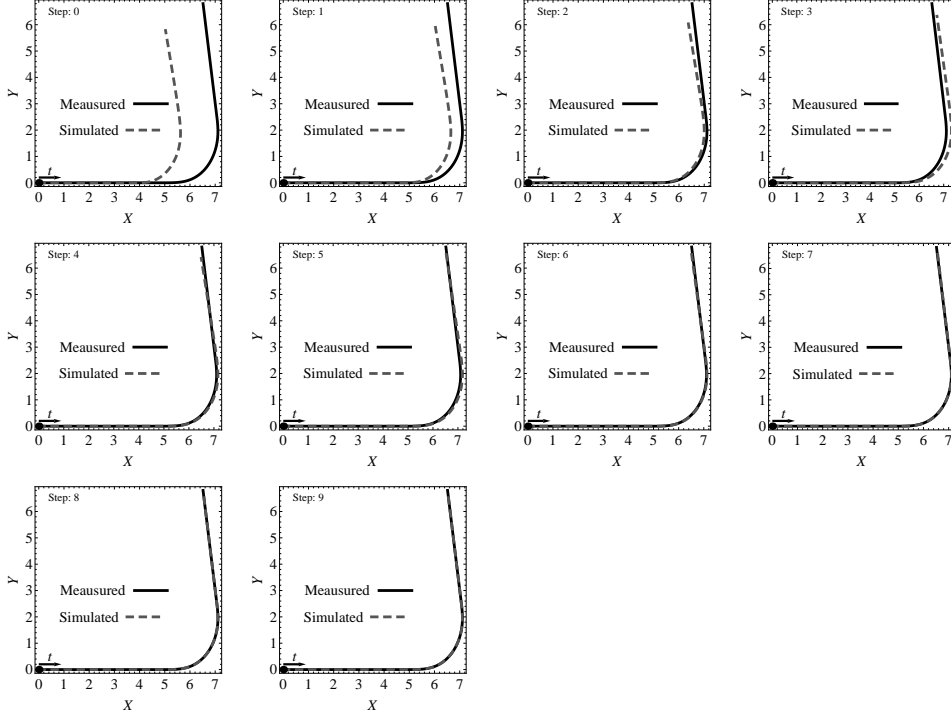


Fig. 5. Plots of the X-Y trajectories for each step of Algorithm 2. The initial position of the vehicle is (0,0) and the vehicle moves in the direction of the arrow.

5.2 Example 2

While Example 2 uses the same system, wheel torques, measured trajectories, and $\mathcal{T}^d = (5, 11)^T$ from Example 1, Example 2 assumes the mode sequence is unknown. Accordingly, the goal for this example is to estimate the mode sequence using the optimization of the relaxed switched system.

As for the cost function, we once again use a quadratic performance index. Therefore, Eq (34) has $\ell(\cdot, \cdot, \cdot, \cdot)$ and $m(\cdot, \cdot)$ equal to $\ell(\hat{x}(\tau), h(\tau), \hat{u}(\tau), \hat{u}_d(\tau)) = 1/2(\hat{x}(\tau) - h(\tau))^T Q (\hat{x}(\tau) - h(\tau)) + 1/2(\hat{u}(\tau) - \hat{u}_d(\tau))^T R (\hat{u}(\tau) - \hat{u}_d(\tau))$ and $m(\hat{x}(T_N), h(T_N)) = 1/2(\hat{x}(T_N) - h(T_N))^T P (\hat{x}(T_N) - h(T_N))$ where Q and P are symmetric semi-positive definite (i.e. $Q^T = Q \geq 0$, $P^T = P \geq 0$), R is symmetric positive definite (i.e. $R^T = R > 0$) and $\hat{x}(t)$ is the relaxed state trajectory, which is the solution to

$$\begin{aligned} \hat{\dot{x}}(t) &= \hat{f}(\hat{x}(t), \hat{u}(t), t) \\ &= \hat{u}_1(t)f_{\sigma=1}(\hat{x}(t), t) + \hat{u}_2(t)f_{\sigma=2}(\hat{x}(t), t) + \hat{u}_3(t)f_{\sigma=3}(\hat{x}(t), t) + \left(1 - \sum_{i=1}^3 \hat{u}_i(t)\right)f_{\sigma=4}(\hat{x}(t), t) \end{aligned} \quad (36)$$

subject to: $\hat{x}(T_0) = x_0$.

We choose \hat{u}_d to be $\hat{u}_d(t) = (1, 0, 0)^T$. This choice places preference on the most active sub-system being $\sigma = 1$. We did this because $\sigma = 1$ is the only sub-system that can exist without losing energy from skidding. Along this reasoning, all other sub-systems will eventually decay to $\sigma = 1$ if the vehicle is driven with zero wheel torque difference.

As for the cost function parameters, we use $Q = \text{diag}(70, 70, 70, 70, 70, 70)$, $R = \text{diag}(1, 1, 1)$ and compute P using the algebraic Riccati equation $\hat{A}^T P + P \hat{A} - P \hat{B} R^{-1} \hat{B}^T P + Q = 0$ where \hat{A} and \hat{B} are the linearizations of $\hat{f}(\hat{x}, \hat{u}, t)$ around \hat{x} and w respectively at time T_N . This is one way to make P approximately compatible with Q and R .

The cost function is minimized using the trajectory functional optimization approach shown in [16]. This results in the optimal control inputs shown in Fig 6. Clearly, \hat{u}_1 specifies $\sigma = 1$ is the most active mode initially. At $t = 4.9871\text{s}$, $\sigma = 4$ becomes the most active mode and at $t = 10.9975\text{s}$, $\sigma = 1$ returns to being the most active. Therefore, the estimated mode sequence is $\Psi = (1, 4, 1)$ with switching times $\mathcal{T} = (4.9871, 10.9975)^T$. These switching times closely approximate $\mathcal{T}^d = (5, 11)^T$.

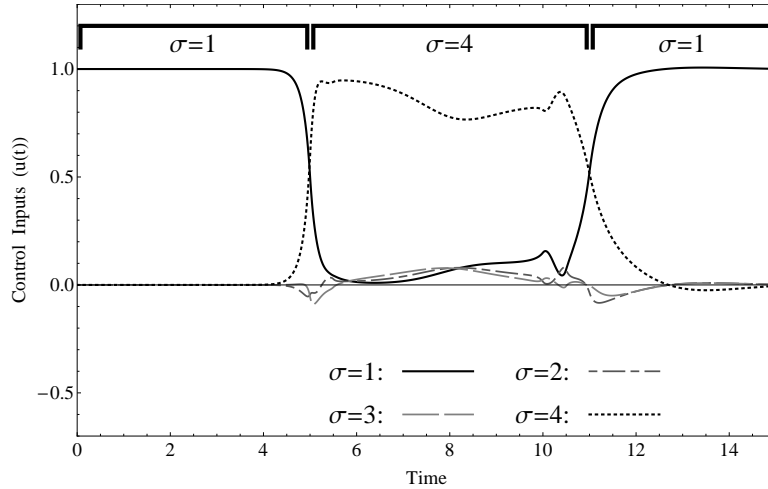


Fig. 6. The optimal relaxed control inputs for Example 2.

5.3 Example 3

In contrast to Examples 1 and 2, this example includes model and sensor disturbances, but otherwise, it uses the same system, wheel torques and $\mathcal{T}^d = (5, 11)^T$. The following is how we generate the perturbed measured trajectories, denoted $\bar{h}(t)$, due to these model and sensor disturbances.

First, we make the obvious assumption that the ground contains imperfections (egs: the ground is wet, it is deteriorated). Therefore, the coefficient of friction, μ_K , is not constant. We model the imperfections as a random walk entering additively to the coefficient of friction. Essentially, this alters $\dot{x} = f_{\sigma=i}$ for $i = 2, 3, 4$ such that μ_K becomes $\mu_K \leftarrow \mu_K + \chi$ where χ is a pseudo-Gaussian signal with a mean of 0 and a standard deviation of 0.5, but where χ differs from a pure Gaussian signal in that $\mu_K + \chi$ is not allowed to be less than 0. This disturbance results in “measured” configuration trajectories $\bar{h}_1(t)$.

Second, let us suppose we have a lone GPS sensor which, of course, imperfectly senses the configuration of the vehicle. The GPS we will use on the SSV in Fig 2 is Garmin’s GPS 16x HVS. It has a positional accuracy of < 3 meters when WAAS is enabled, a velocity accuracy of < 0.1 knots rms (i.e. $< 0.051\bar{4}$ m/s rms) and a sample rate of 1Hz. The sample rate restricts us to 16 data points. Since the positional accuracy is the absolute position from the latitude/longitude coordinate $(0,0)$, and we only need the (X, Y) offset from the initial reading, the accuracy of (X, Y) should be better than < 3 meters. Instead of “guessing” what this accuracy is and basing the disturbance model from that, we instead base the disturbance from the velocity accuracy and integrate to find the positional disturbance.

The velocity components, $(\dot{X}, \dot{Y})(t)$, of $\bar{h}_1(t)$ are sampled at 1Hz. Each sample is perturbed with a random magnitude generated from a uniform distribution of numbers from the interval $(-0.051\bar{4}, 0.051\bar{4})$ and a uniformly distributed

random angle. The samples are interpolated with splines and integrated to find $(X, Y)(t)$. The orientation and turning velocity is numerically calculated from $(X, Y)(t)$, fulfilling the composition of $\bar{h}(t)$. Figure 7 compares the perturbed measured trajectories $\bar{h}(t)$ with the unperturbed measured trajectories $h(t)$.

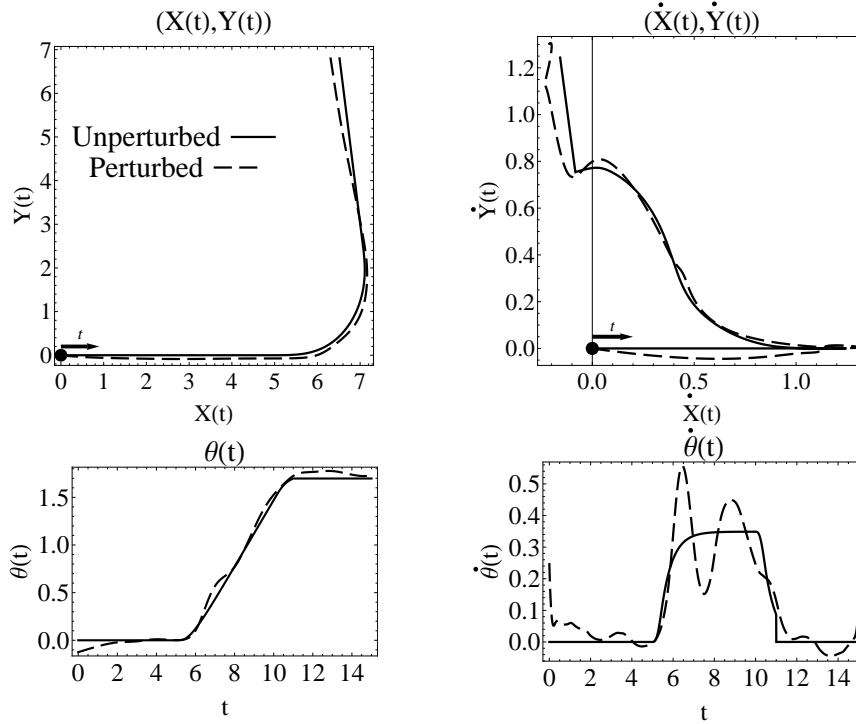


Fig. 7. Comparison of the perturbed $\bar{h}(t)$ with the unperturbed $h(t)$.

With a set of perturbed measured trajectories, the optimization techniques may now be applied to find the mode sequence and optimal switching times that correspond to $\bar{h}(t)$. Using the same cost function, with the same Q , R and P as in Example 2, we first conduct the relaxed switched system optimization. The resulting optimal relaxed control input, $\hat{u}^*(t)$, are in Fig 8. These control inputs correspond to a mode sequence of $\Psi = (1, 4, 2, 4, 1)$ and switching times $\mathcal{T}_0 = (5.0987, 8.0416, 8.7065, 11.4723)^T$, which are the initial parameters for switching time optimization.

Aside: The sub-system $\sigma = 2$ exists for only 0.6649s. Furthermore, it is only slightly the most active sub-system for its short time period. Therefore, an obvious option is to implement a heuristic with the purpose of removing modes from the mode sequence that exhibit similar behavior to $\sigma = 2$. For the purpose of demonstration though, we shall leave it in.

Using Ψ and \mathcal{T}_0 from the relaxed switched system optimization, switching time optimization shall now be conducted. First, we call Algorithm 1 with $sd(\mathcal{T}_0, 1000, 10^{-5})$. After the second step, the $\sigma = 2$ sub-system exists for less than 0.1 seconds, so we remove it, leaving $\Psi = (1, 4, 1)$. With this removal, steepest descent fails to converge to the prescribed accuracy. Now we call Algorithm 2 with $nm(\mathcal{T}_0, 4, 1000, 10^{-5})$ for comparison. With the same removal of the $\sigma = 2$ sub-system, Algorithm 2 converges to the switching times $\mathcal{T}^* = (4.8537, 10.9376)^T$ in three steps of Newton's method after the initial four steps of steepest descent. These optimal switching times are approximately $\mathcal{T}^d = (5, 11)^T$ despite the large variance of friction, the slow sample rate of the GPS and the GPS sensor error. Fig 9 compares the convergence of Algorithm 1 and Algorithm 2 to $\|DJ(\mathcal{T})\| = 0$. Fig 10 compares the X-Y trajectories computed using the optimal switching times with the measured trajectories $\bar{h}(t)$ as well as showing the performance metric $\ell(x(t))$ for the optimized system.

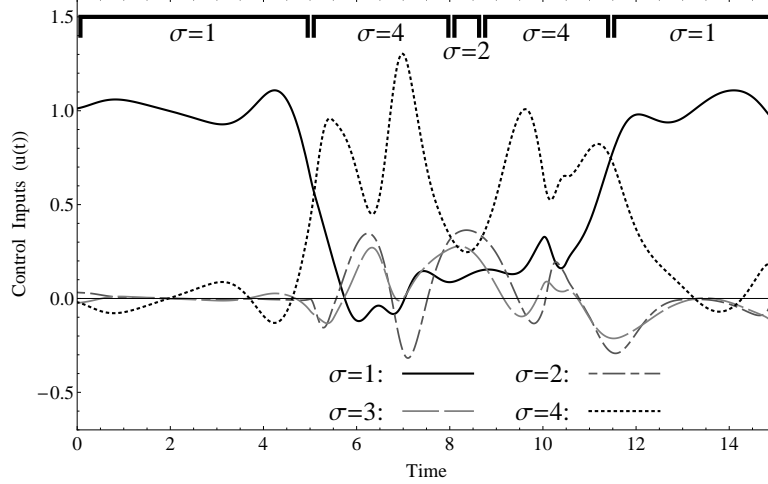


Fig. 8. The optimal relaxed control inputs for Example 3.

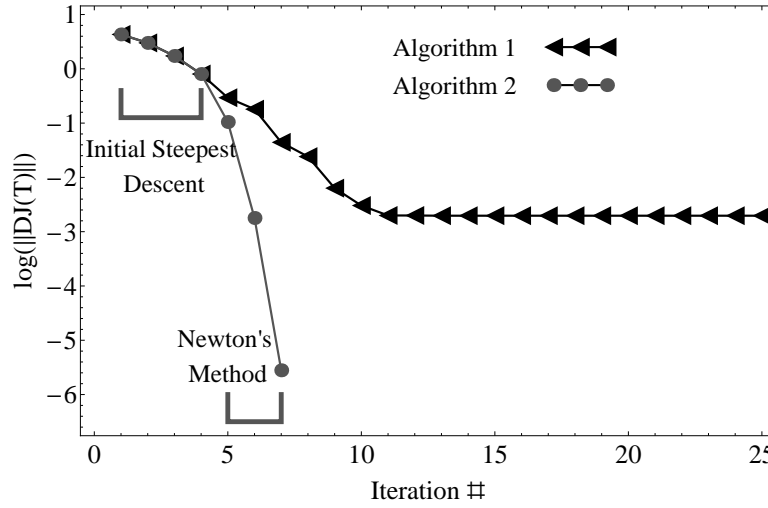


Fig. 9. Comparison of the convergence of Algorithm 1 and Algorithm 2, showing the first 25 steps of Algorithm 1. Algorithm 1 fails to converge to 10^{-5} in its allotted 1000 steps of steepest descent, while Algorithm 2 converges in 7 steps using Newton's method.

5.4 Future Work

As with all theoretical work, our ultimate goal is the implementation of the theory on a physical system. To this end, we have made an initial implementation of the switching time optimization for a maneuver conducted by the SSV shown in Fig 2. The results from this attempt are promising, but more work must be conducted for proper system identification and sensor calibration. None the less, we present the results from this initial attempt with the hope that it further indicates the usefulness of the approaches presented in this paper.

For this initial attempt, the SSV is setup as it is in the examples of Section 5, but with the initial orientation set instead to $\theta_0 = 2.22459\text{rad}$. The SSV is still driven with the same wheel torques as the examples, which is shown in Fig 4. The maneuver is captured using Garmin's GPS 16x HVS. The raw GPS data is shown in Fig 11 as points. The X-Y trajectories are interpolated from these GPS points using splines (see the black line in the figure) and the rest of $h(t)$ is numerically calculated from this interpolation. Calling $nm(\mathcal{T}_0, 6, 1000, 10^{-5})$ results in the optimal switching times $\mathcal{T}^* = (5.1478, 9.7196)^T$ after an initial 6 steps of steepest descent followed by 5 additional steps of Newton's method. While we currently lack an absolute measurement of the switching times in order to base the accuracy of \mathcal{T}^* , it is clear that \mathcal{T}^* at least approximates the correct switching values, which is clear from checking the times for which the wheel torque difference is significant. The results of this initial attempt are promising due to the poor positional accuracy and slow sample rate of the GPS data as well as the current poor system identification, which

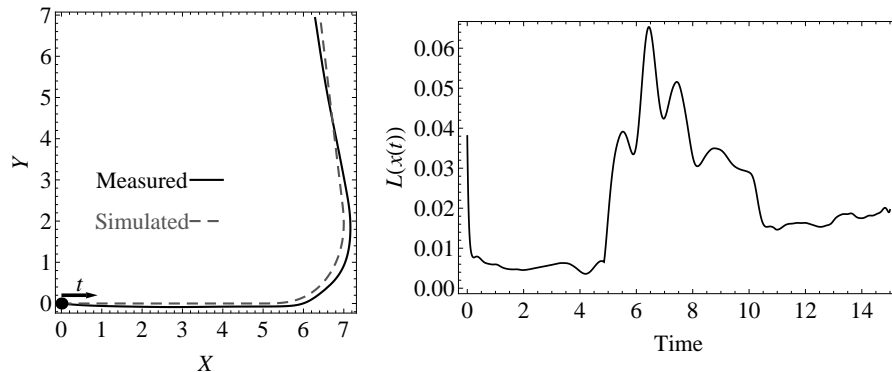


Fig. 10. (left) X-Y plot of the simulated trajectories corresponding to the optimal switching times compared with the measured trajectories $\bar{h}(t)$ and (right) plot of the performance metric $\ell(x(t))$.

is obvious from the figure.

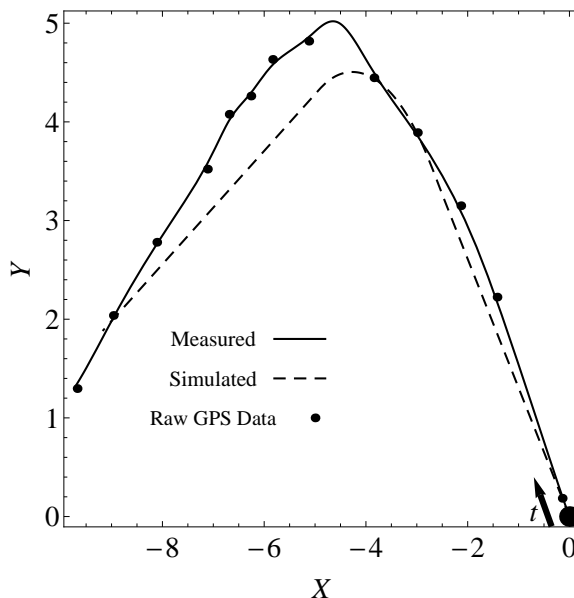


Fig. 11. X-Y plot of the simulated trajectories (dashed line) corresponding to the optimal switching times compared with the measured trajectories (solid line) calculated from GPS data. The X-Y measured trajectories is a spline interpolation of the discrete raw GPS positional data points (dots).

Other future work lies in optimally estimating parameters of the SSV. This includes optimally estimating the coefficient of friction. Furthermore, we plan to investigate optimal control of an SSV. Optimal control of an SSV has the added complication that the control inputs must force the system into the desired sub-systems.

Moreover, we plan to further the relaxed mode estimation approach presented in this paper by investigating the two approaches described in the 2nd remark of Section 5.

6 Conclusion

This paper provides algorithms for optimally estimating the skid sub-systems of a skid-steered vehicle given some maneuver. The algorithms use Coulomb friction to model the interaction of the SSV's wheels with the ground, thereby partitioning the SSV into a switched system. Doing this simplifies the representation of maneuvers of the SSV to the pair (\mathcal{T}, Ψ) , where the switching times, \mathcal{T} , are the times at which the modes from the mode sequence Ψ transition. The algorithms estimate this pair.

Two algorithms are analyzed for optimally estimating the switching times. The first uses the first-order descent direction to minimize an associated performance index. This descent direction exhibits linear convergence. In contrast, the second algorithm minimizes the performance index using the second-order descent direction, and therefore benefits from quadratic convergence. The two algorithms are compared in the example of the SSV, where the results demonstrate the advantages of quadratic convergence. In addition, the examples of the SSV demonstrate that these algorithms perform well while in the presence of model and sensor disturbances.

The switching time optimization assumes a mode sequence. This paper removes the assumption by presenting an approach for estimating a mode sequence. The approach relaxes the switched system optimization and projects back onto the set of feasible solutions in order to do this estimation. As with switching time optimization, the examples demonstrate that the mode estimation approach performs well despite model and sensor disturbances.

Acknowledgements

This material is based upon work supported by the National Science Foundation under award CCF-0907869. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- [1] M. Alamir and S. A. Attia. Discussion on the paper “an optimal control approach for hybrid systems” by P. Riefinger, C. Iung and F. Kratz. *European Journal of Control*, pages 459–460, 2003.
- [2] M. Alamir and S. A. Attia. On solving optimal control problems for switched hybrid nonlinear systems by strong variations algorithms. *In Proceedings of 6th IFAC Symposium on Nonlinear Control Systems*, pages 558–563, 2004.
- [3] Georgia Anousaki and K. J. Kyriakopoulos. A dead-reckoning scheme for skid-steered vehicles in outdoor environments. *International Conference on Robotics and Automation*, 1:580–585, 2004.
- [4] Georgia C. Anousaki and Konstantinos J. Kyriakopoulos. Simultaneous localization and map building of skid-steered robots. *IEEE Robotics and Automation Magazine*, pages 79–89, 2007.
- [5] L. Armijo. Minimization of functions having lipschitz continuous first-partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.
- [6] S. A. Attia, M. Alamir, and C. Canudas de Wit. Sub optimal control of switched nonlinear systems under location and switching constraints. *In Proceedings of the 16th IFAC World Congress*, 2005.
- [7] H. Axelsson, Y. Wardi, M. Egerstedt, and E.I. Verriest. Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *Journal of Optimization Theory and Applications*, 136:167–186, 2008.
- [8] T. M. Caldwell and T. D. Murphey. Second-order optimal estimation of slip state for a simple slip-steered vehicle. *IEEE Conference on Automation Science and Engineering*, 5:133–139, 2009.
- [9] Luca Caracciolo, Alessandro De Luca, and Stefano Iannitti. Trajectory tracking control of a four-wheel differentially driven mobile robot. *International Conference on Robotics and Automation*, 4:2632–2638, 1999.
- [10] Chi-Tsong Chen. *Linear System theory and Design*. Oxford University Press, 1999.
- [11] Florent C. Delmotte. *Multi-Modal Control: From Motion Description Languages to Optimal Control*. PhD thesis, Georgia Institute of Technology, 2006.
- [12] X.C. Ding, Y. Wardi, and M. Egerstedt. On-line optimization of switched-mode dynamical systems. *IEEE Transactions on Automatic Control*, 54:2266–2271, 2009.
- [13] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal control of switching times in switched dynamical systems. *IEEE Conference on Decision and Control*, 3:2138–2143, 2003.
- [14] Magnus Egerstedt, Shun ichi Azuma, and Henrik Axelsson. Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control*, 51:110–115, 2006.
- [15] Magnus Egerstedt, Shun ichi Azuma, and Yorai Wardi. Optimal timing control of switched linear systems based on partial information. *Nonlinear Analysis: Theory, Methods and Applications*, 65:1736–1750, 2006.
- [16] John Hauser. A projection operator approach to the optimization of trajectory functionals. *IFAC World Congress*, 2002.
- [17] John Hauser and David G. Meyer. The trajectory manifold of a nonlinear control system. *IEEE Conference on Decision and Control*, 1:1034–1039, 1998.
- [18] S. Hedlund and A. Rantzer. Convex dynamic programming for hybrid systems. *IEEE Transactions on Automatic Control*, 47:1536–1540, 2002.
- [19] C.T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, 1999.
- [20] Krzysztof Kozłowski and Dariusz Pazderski. Modeling and control of a 4-wheel skid-steering mobile robot. *Int. J. Appl. Math. Comput. Sci.*, pages 477–496, 2004.

- [21] David G. Luenberger. *Linear and Nonlinear Programming*. Springer, 2nd edition, 2003.
- [22] Todd D. Murphey. Kinematic reductions for uncertain mechanical contact. *Robotica*, 25:751–764, 2007.
- [23] R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [24] Axel Schild, Xu Chu Ding, Magnus Egerstedt, and Jan Lunze. Design of optimal switching surfaces for switched autonomous systems. *IEEE Conference on Decision and Control*, 2009. (submitted).
- [25] M. Shahid Shaikh and Peter E. Caines. On the optimal control of hybrid systems: Analysis and zonal algorithms for trajectory and schedule optimization. *IEEE Conference on Decision and Control*, 42:2144–2149, 2003.
- [26] M. Shahid Shaikh and Peter E. Caines. On the optimal control of hybrid systems: Optimization of switching times and combinatoric location schedules. *American Control Conference*, pages 2773–2778, 2003.
- [27] Zibin Song, Yahya H Zweiri, and Lakmal D Seneviratne. Non-linear observer for slip estimation of skid-steering vehicles. *International Conference on Robotics and Automation*, pages 1499–1504, 2006.
- [28] T. H. Tran, N. M. Kwok, S. Scheduling, and Q. P. Ha. Dynamic modelling of wheel-terrain interaction of a ugv. *IEEE Conference on Automation Science and Engineering*, pages 369–374, 2007.
- [29] Vadim I. Utkin and Hao-Chi Chang. Sliding mode control on electro-mechanical systems. *Mathematical Problems in Engineering*, 8:451–473, 2002.
- [30] Erik I. Verriest. Optimal control for switched point delay systems with refractory period. *IFAC World Congress*, 2005.
- [31] Xuping Xu and Panos J. Antsaklis. Optimal control of switched autonomous systems. *IEEE Conference on Decision and Control*, 41:4401 – 4406, 2002.
- [32] Xuping Xu and Panos J. Antsaklis. Optimal control of switched systems via non-linear optimization based on direct differentiations of value functions. *International Journal of Control*, 75:1406 – 1426, 2002.
- [33] Jingang Yi, Dezhen Song, Junjie Zhang, and Zane Goodwin. Adaptive trajectory tracking control of skid-steered mobile robots. *IEEE Conference on Robotics and Automation*, pages 2605–2610, 2007.