# An adjoint method for second-order switching time optimization

T. M. Caldwell
Department of Mechanical Engineering
Northwestern University
2145 Sheridan Road
Evanston, IL 60208, USA
caldwelt@u.northwestern.edu

T. D. Murphey
Department of Mechanical Engineering
Northwestern University
2145 Sheridan Road
Evanston, IL 60208, USA
t-murphey@northwestern.edu

***Abstract*— Switched systems evolve over a sequence of continuous modes of operation, transitioning between modes in a discrete manner. Assuming a mode sequence is known, the evolution of a switched system is dictated by the set of times for which the modes transition. This paper presents second-order optimization of these switching times and compares its convergence with first-order switching time optimization. We emphasize the importance of the second-order method 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.**

## I. Introduction

Switched systems are a class of hybrid systems where the system evolves over a sequence of continuous modes of operation, transitioning between modes in a discrete manner [2]. Therefore, a switched system is described by the pair $(\mathcal{T}, \Psi)$, where $\Psi$ is the mode sequence and $\mathcal{T}$ is the set of switching times for each mode transition in $\Psi$. As in [16], we assume the mode sequence is known ahead of time. Thus our goal is to optimize $\mathcal{T}$.

First-order descent methods (i.e. steepest descent) for switching time optimization are in [3], [5], [7], [8], [16]. As for second-order descent methods (i.e. Newton's method), [6] calculates the second-order descent direction for a bi-modal LTI system, and [15] presents on-line convergence results assuming a calculation for the second derivative is known. We present an explicit derivation of the second-order descent direction for non-linear switched systems.

We emphasize the importance of Newton's method because Newton's method exhibits quadratic convergence. This convergence is in comparison to steepest descent which only converges linearly (see [9], [10]). The distinction between steepest descent and Newton's method is the primary point of this paper because *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 with the convergence of Newton's method for a aircraft flight mode estimation example presented later in the paper.

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.

This paper is organized as follows: Section II provides two definitions of switched systems as well as presents first- and second-order switching time descent directions for steepest descent and Newton's method[1]. Section III, uses an aircraft flight mode estimation example for applying switched system optimization and compares the convergences of steepest descent and Newton's method for the example.

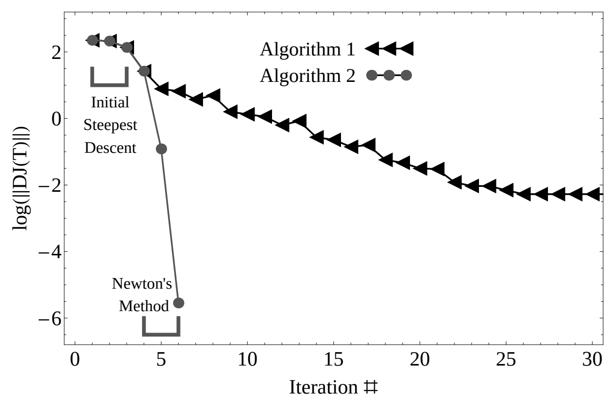


Fig. 1. Log plot of the first 30 steps comparing the convergence to the zero gradient of the cost functional, $DJ(\mathcal{T}) = 0$, for Algorithms 1 and 2. Algorithm 1 is a pure steepest descent algorithm whereas Algorithm 2 conducts a user defined number of initial steps of steepest descent followed by Newton's method until convergence (see Section II). Algorithm 1 fails to converge to the prescribed tolerance of $\|DJ(\mathcal{T})\| \leq 10^{-5}$ in the alloted 1000 steps. The results shown are from the example in Section III

## II. 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 two equivalent definitions of the state trajectories, $x(t) : \mathbb{R} \mapsto \mathbb{R}^n$.

The first is the standard definition [3], [5], [7], [11], [14], [16]:

$$\begin{aligned} \dot{x}(t) &= f\Big(x(t), \mathcal{T}, \Psi, t\Big) \\ &= \begin{cases} f_i\Big(x(t), t\Big), \text{ where } T_{i-1} \leq t < T_i \\ \text{for } i = 1, \ldots, N \end{cases} \\ &\text{subject to: } x(T_0) = x_0 \end{aligned} \tag{1}$$

[1]Section II is included in a paper we submitted to the IFAC journal, *Automatica*. The paper is titled "Switching Mode Generation and Optimal Estimation with Application to Skid-Steering." The section has not been published in any conference proceedings.

where $N$ is the number of modes in the mode sequence $\Psi$, $\mathcal{T} = \{T_1, T_2, \ldots, T_{N-1}\} \in \mathbb{R}^{N-1}$ 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 $i^{\text{th}}$ mode of operation in the sequence $\Psi$. 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:

$$\begin{aligned}\dot{x}(t) &= f\Big(x(t), \mathcal{T}, t\Big)\\ &= \Big[1(t-T_0) - 1(t-T_1^-)\Big] f_1\Big(x(t), t\Big)\\ &\quad + \Big[1(t-T_1^+) - 1(t-T_2^-)\Big] f_2\Big(x(t), t\Big)\\ &\quad + \cdots + \Big[1(t-T_{N-1}^+) - 1(t-T_N)\Big] f_N\Big(x(t), t\Big).\end{aligned} \tag{2}$$

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

The rest of Section II is concerned with optimizing the switched system with respect to switching times.

### A. *Steepest Descent and Newton's Method*

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

$$J(\mathcal{T}) = \int_{T_0}^{T_N} \ell\Big(x(\tau)\Big)\, \mathrm{d}\tau + m\Big(x(T_N)\Big). \tag{3}$$

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

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

Descent techniques are commonly employed to conduct such a minimization [9]. Descent techniques are iterative and have the following form:

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

where $k$ is the current iteration of the descent, $\gamma$ is the step size and $d$ is the step direction.

This paper investigates two descent techniques: steepest descent and Newton's method. The descent directions for the two techniques are (see [9], [10]):[2]

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

where steepest descent converges linearly and Newton's method convergences quadratically. This convergence rate distinction stems from steepest descent suffering from poor conditioning of $J$ around a minimizer while Newton's method does not. Furthermore, steepest descent requires a line search in order to ensure a sufficient decrease. However, Newton's method's descent direction, $d^2$, points in a descending direction only if $D^2J(\mathcal{T})$ is positive definite. Therefore, Newton's method may not function far from a minimizer. (see [9])

As stated in the introduction, convergence rate is important for on-line implementation. Therefore, we stress the value of the quadratic convergence of Newton's method. See [15] for further elaboration of this point. In [15], the authors place an upper bound on the convergence rate of an on-line implementation scheme for switched systems which utilizes the Newton's method descent direction.

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 (STM).[3]

The first-order result in the following subsection, II-B, is not new (see [7]). However, the proof we present is new and the technique used extends to second-order. Also, the second-order result presented later in II-C is also presented in [8] using a parametric approach. In contrast, though, we present an adjoint representation of the second derivative, where our approach instead depends on generalized functions.

### B. *Calculating $DJ(\mathcal{T})$*

*Lemma 2.1:* Provided every $f_i(\cdot,\cdot)$ in $f$ is $C^1$, the $i^{\text{th}}$ switching time derivative of $J(\mathcal{T})$ is

$$D_{T_i} J(\mathcal{T}) = \rho^T(T_i)\Big[f_i\Big(x(T_i), t\Big) - f_{i+1}\Big(x(T_i), t\Big)\Big] \tag{6}$$

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

$$\begin{aligned}\dot{\rho}(t) = -D_1 f\Big(x(t), \mathcal{T}, t\Big)^T \rho(t) - D\ell\Big(x(t)\Big)^T\\ \text{subject to: } \rho(T_N) = Dm\Big(x(T_N)\Big)^T.\end{aligned} \tag{7}$$

*Proof:*

The switching time derivative of the cost function in the

[2] The notation $D$ is the slot derivative. For a function $g$, $Dg(\cdot)$ is the partial derivative of $g$ with respect to its single argument. Similarly, $D_i g(\cdot,\cdot)$ is the partial of $g$ with respect to the $i^{\text{th}}$ argument.

[3] Consider the linear time varying (LTV) control system

$$\begin{aligned}\dot{x}(t) = A(t)x(t) + B(t)u(t)\\ \text{subject to: } x(t_0) = x_0.\end{aligned} \tag{4}$$

The solution to this system is

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)\, \mathrm{d}\tau \tag{5}$$

where $\Phi(t,\tau)$ is the state-transition matrix (STM) 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 [4].

direction of the variation $\theta \in \mathbb{R}^{N-1}$ is

$$DJ(\mathcal{T})\cdot\theta = \int_{T_0}^{T_N} D\ell\Big(x(\tau)\Big)\cdot z(\tau)\,\mathrm{d}\tau + Dm\Big(x(T_N)\Big)\cdot z(T_N) \tag{8}$$

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

$$\begin{aligned}\dot z(t) = \tfrac{\partial}{\partial\mathcal{T}}\dot x(t) = D_1 f\Big(x(t),\mathcal{T},t\Big)\cdot z(t)\\ +D_2 f\Big(x(t),\mathcal{T},t\Big)\cdot\theta\\ \text{subject to: } z(0) = \tfrac{\partial}{\partial\mathcal{T}}x(0) = 0.\end{aligned} \tag{9}$$

Referring to Eq (2), $D_1 f\Big(x(t),\mathcal{T},t\Big)$ and $D_2 f\Big(x(t),\mathcal{T},t\Big)$ become

$$\begin{aligned}D_1 f\Big(x(t),\mathcal{T},t\Big) &= \Big[1(t-T_0) - 1(t-T_1^-)\Big]D_1 f_1\Big(x(t),t\Big)\\ &+\Big[1(t-T_1^+) - 1(t-T_2^-)\Big]D_1 f_2\Big(x(t),t\Big)\\ &+\cdots+\Big[1(t-T_{N-1}^+) - 1(t-T_N)\Big]D_1 f_N\Big(x(t),t\Big)\end{aligned}$$

and

$$\begin{aligned}&D_2 f\Big(x(t),\mathcal{T},t\Big)\\ &=\Big\{\delta(t-T_k^-)f_k\Big(x(t),t\Big) - \delta(t-T_k^+)f_{k+1}\Big(x(t),t\Big)\Big\}_{k=1}^{N-1}\end{aligned} \tag{10}$$

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

Define $A(t)\triangleq D_1 f\Big(x(t),\mathcal{T},t\Big)$ and $B(t)\triangleq D_2 f\Big(x(t),\mathcal{T},t\Big)$ so that

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

This differential equation is of the same form as Eq (4) and has the solution

$$z(t) = \int_{T_0}^{t}\Phi(t,\tau)B(\tau)\cdot\theta\,\mathrm{d}\tau \tag{11}$$

where $\Phi(t,\tau)$ is the STM corresponding to $A(t)$ in Eq (10). Plugging $z(\cdot)$ into $DJ(\mathcal{T})\cdot\theta$ in Eq. (8), we see that

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

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

$$\begin{aligned}=&\int_{T_0}^{T_N}\int_{s}^{T_N} D\ell\Big(x(\tau)\Big)\Phi(\tau,s)B(s)\cdot\theta\,\mathrm{d}\tau\,\mathrm{d}s\\ &+\int_{T_0}^{T_N} Dm\Big(x(T_N)\Big)\Phi(T_N,s)B(s)\cdot\theta\,\mathrm{d}s.\end{aligned} \tag{12}$$

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

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

where $\rho$ is the solution to the backwards differential equation:

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

This result is verified by taking the time derivative of $\rho(t)$ or by comparing to Eqs. (4) and (5). $DJ(\mathcal{T})$ becomes

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

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

$$D_{T_i}J(\mathcal{T})\cdot\theta_i = \rho^T(T_i)\Big[f_i\Big(x(T_i),t\Big) - f_{i+1}\Big(x(T_i),t\Big)\Big]\cdot\theta_i$$

for $i=1,2,\ldots,N-1$, where $\theta_i$ is the $i^{\text{th}}$ index of $\theta$. This completes the proof. ∎

### C. Calculating $D^2J(\mathcal{T})$

*Lemma 2.2:* Provided every $f_i(\cdot,\cdot)$ in $f$ is $C^2$, the second-order switching time derivative of $J(\mathcal{T})$ is

$$\begin{aligned}D_{T_i,T_j}J(\mathcal{T}) = \Bigg[&\Big[f_i\Big(x(T_i),T_i\Big) - f_{i+1}\Big(x(T_i),T_i\Big)\Big]^T\lambda(T_i)\\ &+\rho(T_i)^T\Big[D_1 f_i\Big(x(T_i),T_i\Big) - D_1 f_{i+1}\Big(x(T_i),T_i\Big)\Big]\Bigg]\\ &\cdot\Phi(T_i,T_j)\Big[f_j\Big(x(T_j),T_j\Big) - f_{j+1}\Big(x(T_j),T_j\Big)\Big]\end{aligned} \tag{13}$$

when $i\neq j$ and

$$\begin{aligned}=&\Big[f_i\Big(x(T_i),T_i\Big) - f_{i+1}\Big(x(T_i),T_i\Big)\Big]^T\lambda(T_i)\\ &\cdot\Big[f_i\Big(x(T_i),T_i\Big) - f_{i+1}\Big(x(T_i),T_i\Big)\Big]\\ &+\rho(T_i)^T\Big[D_1 f_i\Big(x(T_i),T_i\Big)f_i\Big(x(T_i),T_i\Big)\\ &-2\,D_1 f_{i+1}\Big(x(T_i),T_i\Big)f_i\Big(x(T_i),T_i\Big)\\ &+D_1 f_{i+1}\Big(x(T_i),T_i\Big)f_{i+1}\Big(x(T_i),T_i\Big)\\ &+D_2 f_i\Big(x(T_i),T_i\Big) - D_2 f_{i+1}\Big(x(T_i),T_i\Big)\Big]\\ &-D\ell\Big(x(T_i)\Big)\Big[f_i\Big(x(T_i),T_i\Big) - f_{i+1}\Big(x(T_i),T_i\Big)\Big]\end{aligned} \tag{14}$$

when $i==j$, where $\rho(t)$ is the numerical solution to Eq (7) and $\lambda(t)\in\mathbb{R}^{n\times n}$ is the numerical solution to

$$\begin{aligned}\dot\lambda(t) = -A(t)^T\lambda(t) - \lambda(t)A(t) - D^2\ell\Big(x(t)\Big)\\ -\sum_{k=1}^{n}\rho_k(t)D_1^2 f^k\Big(x(t),\mathcal{T},t\Big)\\ \text{subject to: } \lambda(T_N) = D^2 m\Big(x(t)\Big).\end{aligned} \tag{15}$$

*Proof:* This proof follows from the proof for Lemma 2.1. 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 has a corresponding integral equation which makes use of STM. We switch the order of integration of $D^2J$, in order to extract an adjoint differential equation, which may be calculated separately. The rest of the proof investigates what the integrals of the $x$ and $\mathcal{T}$ derivatives of $f(\cdot,\cdot,\cdot)$ are.

Let $\theta^1$ and $\theta^2$ be different variations of $\mathcal{T}$ and $z^1(t)$ and $z^2(t)$ be the variations of $x(t)$ corresponding respectively to $\theta^1$ and $\theta^2$. 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 (8)):

$$\begin{aligned}&\frac{\partial}{\partial\mathcal{T}}\Big(DJ(\mathcal{T})\cdot\theta^1\Big)=D^2J(\mathcal{T})\cdot(\theta^1,\theta^2)+DJ(\mathcal{T})\cdot\eta\\&=\int_{T_0}^{T_N}D^2\ell\Big(x(\tau)\Big)\cdot\Big(z^1(\tau),z^2(\tau)\Big)+D\ell\Big(x(\tau)\Big)\cdot\zeta(\tau)\,\mathrm{d}\tau\\&+D^2m\Big(x(T_N)\Big)\cdot\Big(z^1(T_N),z^2(T_N)\Big)+Dm\Big(x(T_N)\Big)\cdot\zeta(T_N)\end{aligned}\tag{16}$$

where $\eta$ 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 $\dot{x}(t)$

$$\begin{aligned}\dot{\zeta}(t)&=\tfrac{\partial^2}{\partial\mathcal{T}^2}\dot{x}(t)\\&=A(t)\cdot\zeta(t)+B(t)\cdot\eta+\begin{pmatrix}z^1(t)^T & {\theta^1}^T\end{pmatrix}\\&\cdot\begin{pmatrix}D_1^2f\Big(x(t),\mathcal{T},t\Big) & D_{1,2}f\Big(x(t),\mathcal{T},t\Big)\\ D_{2,1}f\Big(x(t),\mathcal{T},t\Big) & D_2^2f\Big(x(t),\mathcal{T},t\Big)\end{pmatrix}\begin{pmatrix}z^2(t)\\ \theta^2\end{pmatrix}\\&\text{subject to: }\zeta(0)=\tfrac{\partial^2}{\partial\mathcal{T}^2}x(0)=0.\end{aligned}\tag{17}$$

Define $C(t)\triangleq\begin{pmatrix}D_1^2f\Big(x(t),\mathcal{T},t\Big) & D_{1,2}f\Big(x(t),\mathcal{T},t\Big)\\ D_{2,1}f\Big(x(t),\mathcal{T},t\Big) & D_2^2f\Big(x(t),\mathcal{T},t\Big)\end{pmatrix}$. Notice that $\dot{\zeta}(t)$ is linear with respect to $\zeta(t)$ and therefore, Eq (17) is of the same form as (4). Thus, $\dot{\zeta}(t)$ has solution

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

Plugging $\zeta(t)$ into Eq (16), we see that

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

Note that

$$\begin{aligned}DJ(\mathcal{T})\cdot\eta&=\int_{T_0}^{T_N}D\ell\Big(x(\tau)\Big)\int_{T_0}^{\tau}\Phi(\tau,s)B(s)\cdot\eta\,\mathrm{d}s\,\mathrm{d}\tau\\&+Dm\Big(x(T_N)\Big)\int_{T_0}^{T_N}\Phi(T_N,s)B(s)\cdot\eta\,\mathrm{d}s,\end{aligned}$$

which is obvious from Eq (12). Therefore,

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

Split the integral over $\mathrm{d}\tau$, move $D\ell\Big(x(\tau)\Big)$ and $Dm\Big(x(T_N)\Big)$ into their respective integrals and switch the order of integration of the double integral. This results in

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

We combine the integrals over $\mathrm{d}s$, and notice that $\rho(\tau)^T$ enters the equations. Furthermore, we switch the dummy variable $s$ to $\tau$ and put everything back under one integral:

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

Expand $C(\cdot)$ out

$$\begin{aligned}&=\int_{T_0}^{T_N}z^1(\tau)^TD^2\ell\Big(x(\tau)\Big)z^2(\tau)\\&+\rho(\tau)^T\Big[z^1(\tau)^TD_1^2f\Big(x(\tau),\mathcal{T},\tau\Big)z^2(\tau)\\&+z^1(\tau)^TD_{1,2}f\Big(x(\tau),\mathcal{T},\tau\Big)\theta^2\\&+{\theta^1}^TD_{2,1}f\Big(x(\tau),\mathcal{T},\tau\Big)z^2(\tau)+{\theta^1}^TD_2^2f\Big(x(\tau),\mathcal{T},\tau\Big)\theta^2\Big]\mathrm{d}\tau\\&+z^1(T_N)^TD^2m\Big(x(T_N)\Big)z^2(T_N).\end{aligned}$$

Switching to an index notation where $\rho_k(\cdot)$ is the $k^{\text{th}}$ component of $\rho(\cdot)$ and $f^k(\cdot,\cdot,\cdot)$ is the $k^{\text{th}}$ component of $f(\cdot,\cdot,\cdot)$, results in

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

Rearranging the terms allows $D^2J(\mathcal{T})\cdot(\theta^1,\theta^2)$ to be the summation of three parts:

$$D^2J(\mathcal{T})\cdot(\theta^1,\theta^2) = \mathcal{P}_1+\mathcal{P}_2+\mathcal{P}_3 \tag{18}$$

where

$$\mathcal{P}_1 = \int_{T_0}^{T_N} z^1(\tau)^T\Big[D^2\ell\Big(x(\tau)\Big)+\sum_{k=1}^{n}\rho_k(\tau)D_1^2 f^k\Big(x(\tau),\mathcal{T},\tau\Big)\Big] \cdot z^2(\tau)\,\mathrm{d}\tau + z^1(T_N)^T D^2 m\Big(x(T_N)\Big)z^2(T_N),$$

$$\mathcal{P}_2 = \int_{T_0}^{T_N} \theta^{2^T}\sum_{k=1}^{n}\rho_k(\tau)D_{2,1}f^k\Big(x(\tau),\mathcal{T},\tau\Big)z^1(\tau) + \theta^{1^T}\sum_{k=1}^{n}\rho_k(\tau)D_{2,1}f^k\Big(x(\tau),\mathcal{T},\tau\Big)z^2(\tau)\,\mathrm{d}\tau$$

and

$$\mathcal{P}_3 = \int_{T_0}^{T_N}\theta^{1^T}\sum_{k=1}^{n}\rho_k(\tau)D_2^2 f^k\Big(x(\tau),\mathcal{T},\tau\Big)\theta^2\,\mathrm{d}\tau.$$

First, let us examine $\mathcal{P}_1$. Let

$$g(\tau) = D^2\ell\Big(x(\tau)\Big)+\sum_{k=1}^{n}\rho_k(\tau)D_1^2 f^k\Big(x(\tau),\mathcal{T},\tau\Big).$$

Thus,

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

Plugging Eq (11) in for $z^{\cdot}(\cdot)$ results in

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

The integrals may be specified as

$$= \int_{T_0}^{T_N}\int_{T_0}^{\tau}\int_{T_0}^{\tau}\theta^{1^T}B(s)^T\Phi(\tau,s)^T g(\tau)\Phi(\tau,w)B(w)\theta^2\,\mathrm{d}s\,\mathrm{d}w\,\mathrm{d}\tau + \int_{T_0}^{T_N}\int_{T_0}^{T_N}\theta^1 B(s)^T\Phi(T_N,s)^T D^2 m\Big(x(T_N)\Big) \cdot\Phi(T_N,w)B(w)\theta^2\,\mathrm{d}s\,\mathrm{d}w.$$

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

$$= \int_{T_0}^{T_N}\int_{T_0}^{T_N}\int_{\max(s,w)}^{T_N}\theta^{1^T}B(s)^T\Phi(\tau,s)^T g(\tau)\Phi(\tau,w) \cdot B(w)\theta^2\,\mathrm{d}\tau\,\mathrm{d}s\,\mathrm{d}w + \int_{T_0}^{T_N}\int_{T_0}^{T_N}\theta^1 B(s)^T\Phi(T_N,s)^T \cdot D^2 m\Big(x(T_N)\Big)\Phi(T_N,w)B(w)\theta^2\,\mathrm{d}s\,\mathrm{d}w.$$

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

$$= \int_{T_0}^{T_N}\int_{T_0}^{T_N}B(s)^T\Big[\int_{\max(s,w)}^{T_N}\Phi(\tau,s)^T g(\tau)\Phi(\tau,w)\,\mathrm{d}\tau + \Phi(T_N,s)^T \cdot D^2 m\Big(x(T_N)\Big)\Phi(T_N,w)\Big]B(w)\,\mathrm{d}s\,\mathrm{d}w\cdot(\theta^1,\theta^2). \tag{19}$$

Let

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

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

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

Then, depending on whether $s>w$, $s<w$ or $s==w$, $\lambda(\cdot)$ enters into Eq (19) as shown:

$$\mathcal{P}_1 = \begin{cases}\displaystyle\int_{T_0}^{T_N}\int_{T_0}^{T_N}B(s)^T\lambda(s)\Phi(s,w)B(w)\,\mathrm{d}s\,\mathrm{d}w\cdot(\theta^1,\theta^2) & \text{when } s>w,\\ \displaystyle\int_{T_0}^{T_N}\int_{T_0}^{T_N}B(s)^T\Phi(w,s)^T\lambda(w)B(w)\,\mathrm{d}s\,\mathrm{d}w\cdot(\theta^1,\theta^2) & \text{when } s<w, \text{ or}\\ \displaystyle\int_{T_0}^{T_N}\int_{T_0}^{T_N}B(s)^T\lambda(s)B(w)\,\mathrm{d}s\,\mathrm{d}w\cdot(\theta^1,\theta^2) & \text{when } s==w.\end{cases} \tag{20}$$

The $s<w$ case may be rewritten as

$$\mathcal{P}_1 \overset{s\le w}{=} \int_{T_0}^{T_N}\int_{T_0}^{T_N}B(w)^T\lambda(w)\Phi(w,s)B(s)\,\mathrm{d}s\,\mathrm{d}w\cdot(\theta^2,\theta^1).$$

Use $i$ and $j$ to index $\theta^1$ and $\theta^2$ respectively, where $i,j=1,\ldots,N-1$. Integrating the $\delta$-functions in $B(s)$ and $B(w)$ will pick out times $s=T_i$ and $w=T_j$ such that

$$\mathcal{P}_{1_{ij}} = \begin{cases}\Big[f_i\Big(x(T_i),t\Big)-f_{i+1}\Big(x(T_i),t\Big)\Big]^T\lambda(T_i)\Phi(T_i,T_j) \cdot\Big[f_j\Big(x(T_j),t\Big)-f_{j+1}\Big(x(T_j),t\Big)\Big]\cdot(\theta_i^1,\theta_j^2) & \text{when } i>j,\\ \Big[f_j\Big(x(T_j),t\Big)-f_{j+1}\Big(x(T_j),t\Big)\Big]^T\lambda(T_j)\Phi(T_j,T_i) \cdot\Big[f_i\Big(x(T_i),t\Big)-f_{i+1}\Big(x(T_i),t\Big)\Big]\cdot(\theta_i^1,\theta_j^2) & \text{when } i<j, \text{ or}\\ \Big[f_i\Big(x(T_i),t\Big)-f_{i+1}\Big(x(T_i),t\Big)\Big]^T\lambda(T_i) \Big[f_i\Big(x(T_i),t\Big)-f_{i+1}\Big(x(T_i),t\Big)\Big]\cdot(\theta_i^1,\theta_j^2) & \text{when } i==j.\end{cases} \tag{21}$$

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

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

$$D_2 f\Big(x(t),\mathcal{T},t\Big) = \Big\{\delta(t-T_a^-)D_1 f_a^k\Big(x(t),t\Big)^T - \delta(t-T_a^+)D_1 f_{a+1}^k\Big(x(t),t\Big)^T\Big\}_{a=1}^{N-1}.$$

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

$$z_k^{\cdot}(t) = \int_{T_0}^{t} \Phi(t,\tau)\Big[\delta(\tau - T_k^-) f_k\Big(x(\tau), \tau\Big) \\ -\delta(\tau - T_k^+) f_{k+1}\Big(x(\tau),\tau\Big)\Big]\, \mathrm{d}\tau \theta_k^{\cdot}. \tag{22}$$

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

$$\mathcal{P}_{2_{ij}} = \int_{T_0}^{T_N} \theta_j^2 \rho(\tau)^T \Big[\delta(\tau - T_j^-) D_1 f_j\Big(x(\tau),\tau\Big) \\ -\delta(\tau - T_j^+) D_1 f_{j+1}\Big(x(\tau),\tau\Big)\Big] z_i^1(\tau) \\ +\theta_i^1 \rho(\tau)^T \Big[\delta(\tau - T_i^-) D_1 f_i\Big(x(\tau),\tau\Big) \\ -\delta(\tau - T_i^+) D_1 f_{i+1}\Big(x(\tau),\tau\Big)\Big] z_j^2(\tau)\, \mathrm{d}\tau. \tag{23}$$

Integrating over the $\delta$-functions picks out the times for which the $\delta$-functions' arguments are zero:

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

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 (22), observe that $z_i^{\cdot}(t)$ is non-zero only after time $t = T_i^-$ due to the $\delta$-functions. Consequently, $z_i^1(T_j^{\cdot}) = 0$. Therefore,

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

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

$$\overset{i \geq j}{=} \theta_i^1 \rho(T_i)^T \Big[D_1 f_i\Big(x(T_i), T_i\Big) \\ -D_1 f_{i+1}\Big(x(T_i), T_i\Big)\Big] z_j^2(T_i).$$

Plugging Eq (22) in for $z_j^2(T_i)$ results in

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

the integration over the $\delta$-function results in

$$\mathcal{P}_{2_{ij}} \overset{i \geq j}{=} \rho(T_i)^T \Big[D_1 f_i\Big(x(T_i), T_i\Big) - D_1 f_{i+1}\Big(x(T_i), T_i\Big)\Big] \\ \cdot \Phi(T_i, T_j)\Big[f_j\Big(x(T_j), T_j\Big) - f_{j+1}\Big(x(T_j), T_j\Big)\Big] \cdot (\theta_i^1, \theta_j^2). \tag{24}$$

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.

Finally, we analyze the $i == j$ case. First, note that because $i == j$, the perturbations $\theta_i^1$ and $\theta_j^2$ are equivalent and therefore, $z_i^1(t)$ are $z_j^2(t)$ are equivalent. Eq (23) for this case becomes

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

Plugging Eq (22) in for the $z_i^1(\cdot)$ terms, we see that

$$\overset{i==j}{=} 2\, \theta_i^2 \rho(T_i^-)^T D_1 f_i\Big(x(T_i^-), T_i^-\Big) \int_{T_0}^{T_i^-} \Phi(T_i^-, \tau) \\ \cdot \Big[\delta(\tau - T_i^-) f_i\Big(x(\tau),\tau\Big) - \delta(\tau - T_i^+) f_{i+1}\Big(x(\tau),\tau\Big)\Big] \mathrm{d}\tau \theta_i^1 \\ -2\, \theta_i^2 \rho(T_i^+)^T D_1 f_{i+1}\Big(x(T_i^+), T_i^+\Big) \int_{T_0}^{T_i^+} \Phi(T_i^+, \tau) \\ \cdot \Big[\delta(\tau - T_i^-) f_i\Big(x(\tau),\tau\Big) - \delta(\tau - T_i^+) f_{i+1}\Big(x(\tau),\tau\Big)\Big] \mathrm{d}\tau \theta_i^1.$$

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

$$\overset{i==j}{=} 2\, \theta_i^2 \rho(T_i^-)^T D_1 f_i\Big(x(T_i^-), T_i^-\Big) \tfrac{1}{2}\, \Phi(T_i^-, T_i^-) \\ \cdot f_i\Big(x(T_i^-), T_i^-\Big)\theta_i^1 - 2\, \theta_i^2 \rho(T_i^+)^T D_1 f_{i+1}\Big(x(T_i^+), T_i^+\Big) \\ \cdot \Big[\Phi(T_i^+, T_i^-) f_i\Big(x(T_i^-), T_i^-\Big) - \tfrac{1}{2}\, \Phi(T_i^+, T_i^+) \\ \cdot f_{i+1}\Big(x(T_i^+), T_i^+\Big)\Big]\theta_i^1.$$

We 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,

$$\mathcal{P}_{2_{ij}} \overset{i==j}{=} \rho(T_i)^T \Big[D_1 f_i\Big(x(T_i), T_i\Big) f_i\Big(x(T_i), T_i\Big) \\ -2\, D_1 f_{i+1}\Big(x(T_i), T_i\Big) f_i\Big(x(T_i), T_i\Big) \\ +D_1 f_{i+1}\Big(x(T_i), T_i\Big) f_{i+1}\Big(x(T_i), T_i\Big)\Big] \cdot (\theta_i^1, \theta_i^2). \tag{25}$$

Finally, we are left with $\mathcal{P}_3$. Again, Let us index $\theta^1$ with $i$ and $\theta^2$ with $j$ where $i, j = 1, \ldots, N-1$. Start with the $ij^{\text{th}}$ component of $D_2^2 f^k\Big(x(\tau), \mathcal{T}, \tau\Big)$

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

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

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

Then, $\mathcal{P}_3$ becomes

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

Conducting integration by parts, we see that

$$\stackrel{i==j}{=} \Bigg[\int_{T_0}^{T_N} \Big[\dot{\rho}(\tau)^T f_i\Big(x(\tau),\tau\Big) + \rho(\tau)^T D_1 f_i\Big(x(\tau),\tau\Big)\dot{x}(t)$$
$$+\rho(\tau)^T D_2 f_i\Big(x(\tau),\tau\Big)\Big]\delta(\tau - T_i^-)\,\mathrm{d}\tau$$
$$-\int_{T_0}^{T_N} \Big[\dot{\rho}(\tau)^T f_{i+1}\Big(x(\tau),\tau\Big) + \rho(\tau)^T D_1 f_{i+1}\Big(x(\tau),\tau\Big)\dot{x}(t)$$
$$+\rho(\tau)^T D_2 f_{i+1}\Big(x(\tau),\tau\Big)\Big]\delta(\tau - T_i^+)\,\mathrm{d}\tau\Bigg]\cdot(\theta_i^1,\theta_i^2).$$

Integrating over the $\delta$-functions picks out the times for which the $\delta$-functions' arguments are zero:

$$\stackrel{i==j}{=} \Bigg[\dot{\rho}(T_i^-)^T f_i\Big(x(T_i^-),T_i^-\Big) - \dot{\rho}(T_i^+)^T f_{i+1}\Big(x(T_i^+),T_i^+\Big)$$
$$+\rho(T_i^-)^T D_1 f_i\Big(x(T_i^-),T_i^-\Big)\dot{x}(T_i^-)$$
$$-\rho(T_i^+)^T D_1 f_{i+1}\Big(x(T_i^+),T_i^+\Big)\dot{x}(T_i^+)$$
$$+\rho(T_i^-)^T D_2 f_i\Big(x(T_i^-),T_i^-\Big)$$
$$-\rho(T_i^+)^T D_2 f_{i+1}\Big(x(T_i^+),T_i^+\Big)\Bigg]\cdot(\theta_i^1,\theta_i^2).$$

Plugging Eq (1) in for $\dot{x}(\cdot)$ and Eq (7) in for $\dot{\rho}(\cdot)$ and simplifying reveals

$$\mathcal{P}_{3_{ij}} \stackrel{i==j}{=} -D\ell\Big(x(T_i)\Big)\Big[f_i\Big(x(T_i),T_i\Big) - f_{i+1}\Big(x(T_i),T_i\Big)\Big]$$
$$+\rho(T_i)^T\Big[D_2 f_i\Big(x(T_i),T_i\Big) - D_2 f_{i+1}\Big(x(T_i),T_i\Big)\Big]. \tag{26}$$

Recall from Eq (18) that $D^2J(\mathcal{T})\cdot(\theta^1,\theta^2) = \mathcal{P}_1+\mathcal{P}_2+\mathcal{P}_3$ is the summation of Eqs (21) and (24) for the case when $i \neq j$ and the summation of Eqs (21), (25) and (26) for the case when $i == j$. ■

### D. Steepest Descent and Newton's Method Algorithms

With Lemmas 2.1 and 2.2, we can 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.

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 $D^2J(\mathcal{T})$, *regardless of the number of switching times*.
2) When calculating $\Phi(T_i,T_j)$, calculate $\Phi(T_k,T_{k+1})$ for $k$=1,..., $N-1$ 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.

We present the algorithm, Optimization Algorithm, which conducts a user defined number of initial steps of steepest descent, followed by Newton's method. Steepest descent's descent direction, $d^1(\mathcal{T})$, is calculated from Eq (6), where $x(t)$ is calculated from Eq (2) and $\rho(t)$ is calculated from Eq (7). Newton's method's descent direction, $d^2(\mathcal{T})$, is calculated from Eqs (6), (13) and (14), where $x(t)$ and $\rho(t)$ are calculated as before and $\lambda(t)$ is calculated from Eq (15). The algorithm has the following arguments: $\mathcal{T}_0$ are the initial switching times, $sd_{init}$ is the number of initial steps of steepest descent, $k_{max}$ is the maximum number of steps (for implementation), and $\epsilon$ is the convergence criteria tolerance.

**Optimization Algorithm:** $desc()$

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

---

$k = 0$; $\mathcal{T}_k = \mathcal{T}_0$
**while** $\|DJ(\mathcal{T}_k) > \epsilon\|$ and $k < k_{max}$ **do**
  **if** $k < sd_{init}$ **then**
    Calculate $d^1(\mathcal{T}_k)$
    Choose $\gamma$ according to Armijo line search (see [1])
    $\mathcal{T}_{k+1} = \mathcal{T}_k + \gamma d^1(\mathcal{T}_k)$
  **else**
    Calculate $d^2(\mathcal{T}_k)$
    $\mathcal{T}_{k+1} = \mathcal{T}_k + \gamma d^2(\mathcal{T}_k)$
  **end if**
  $k = k+1$
**end while**
$\mathcal{T}^* = \mathcal{T}_k$

## III. Example

Due to the increasing demand of air travel coupled with the desire for minimal flight time and fuel consumption, the idea of "free" flight was proposed [12], [13]. The concept is to lessen or remove the air traffic controller's input into flight trajectory decision making and transition the responsibility to the pilots. This responsibility includes resolving trajectory conflicts of multiple aircrafts. As Tomlin et al. state, the new control must be *provably safe* [12], [13]. Tomlin et al.'s work is concerned with finding control laws such that multiple aircrafts remain in "safe" regions relative to one another. A difficulty in this work is the estimation of the behavior of an aircraft, which is pertinent to ensuring safe distances between two aircrafts. We present an example demonstrating how switching time optimization may be used for this purpose.

Suppose an aircraft is flying at a fixed altitude and therefore, its configuration may be represented by three variables, $x = \{X, Y, \psi\}$, for its position and orientation in $\mathbb{R}^2$. The craft's motion is dictated by the following kinematic model:

$$\dot{x}(t) = f\big(x(t),\nu(t),\omega(t)\big)$$
$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \end{bmatrix}(t) = \begin{bmatrix} \nu(t)\cos\psi(t) \\ \nu(t)\sin\psi(t) \\ \omega(t) \end{bmatrix}$$

where $\nu(t)$ is its linear velocity and $\omega(t)$ is its angular velocity. The presented model is the same model as used in [12], [13]. Now, suppose the aircraft will either fly straight, bank left, or bank right at fixed velocities. This assumption may be a reasonable one if, for instance, the aircraft is performing a "roundabout" maneuver for collision avoidance, as described in [13]. The following are the three flight modes:

$$\sigma = 1: \quad \nu = \nu_c, \quad \omega = 0, \text{ (Straight)},$$
$$\sigma = 2: \quad \nu = \nu_c, \quad \omega = \omega_c, \text{ (Bank Left) and}$$
$$\sigma = 3: \quad \nu = \nu_c, \quad \omega = -\omega_c, \text{ (Bank Right)}$$

where $\nu_c = 4$ nautical miles per minute and $\omega_c = 1$ radian per minute.

We include disturbances in the simulation. The disturbances may be from pilot, sensor and model errors. The errors are represented as a random walk entering additively to $v$ and $\omega$. The perturbed velocity is $v_p = v + \mathcal{V}(0,4)$ and the perturbed rotational velocity is $\omega_p = \omega + \Omega(0,1)$, where $\mathcal{V}$ and $\Omega$ are Gaussian signals with mean of 0 and standard deviation of 4 and 1 respectively. The measured trajectory, $x_m$ is found by simulation using $v_p$ and $\omega_p$ such that the aircraft flies the maneuver described in Table I. Figure 2 shows the perturbed $X-Y$ trajectory as well as the unpertrubed $X-Y$ trajectory for comparison.

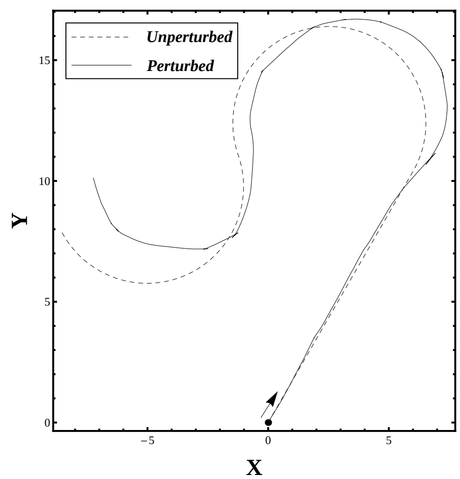


Fig. 2. Comparison between two measured $X-Y$ trajectories of an aircraft maneuver. One is with disturbances and the other is without.

| Time ($t$) | [0,3) | [3,7) | [7,10] |
|---|---|---|---|
| Mode ($\sigma$) | 1 | 2 | 3 |

TABLE I

THE TIMES AND FLIGHT MODES OF A MANEUVER FOR THE TRUE SYSTEM.

Using a quadratic performance index, Eq (3) is specified with

$$\ell\Big(x(\tau)\Big) = 1/2\Big(x(\tau) - x_m(\tau)\Big)^T Q\Big(x(\tau) - x_m(\tau)\Big)$$

and

$$m\Big(x(T_N)\Big) = 1/2\Big(x(T_N) - x_m(T_N)\Big)^T P\Big(x(T_N) - x_m(T_N)\Big)$$

where $Q$ and $P$ are symmetric positive semi-definite (i.e. $Q^T = Q \geq 0$ and $P^T = P \geq 0$). For the example, we set $Q$ and $P$ to $Q = \mathrm{diag}(1,1,1)$ and $P = \mathrm{diag}(1,1,10)$.

The initial switching times are arbitrarily chosen to be $\mathcal{T}_0 = (2,5)^T$. First, we call Optimization Algorithm with $desc(\mathcal{T}_0,\ \infty,\ 1000,\ 10^{-5})$ so that the algorithm only uses the steepest descent direction. Without Newton's method, the algorithm fails to converge to the prescribed accuracy within the alloted 1000 steps. Around the $25^{\text{th}}$ step, the algorithm stagnates because the step size that meets the sufficient decrease requirements of the Armijo line search [1] is $\gamma = 1.8 \cdot 10^{-9}$. Now we call the algorithm with $desc(\mathcal{T}_0,$ 3, 1000, $10^{-5})$ for comparison. This time, the algorithm converges to the switching times $\mathcal{T}^\star = (3.1789, 7.0696)^T$ in three steps of Newton's method after the initial three steps of steepest descent. Compare this result with Table I. Furthermore, Fig 1, presented in the introduction, compares the convergences of steepest descent and Newton's method for this example.

## IV. Conclusion

This paper presents a second-order descent direction used in Newton's method for optimizing the switching times of a switched system. Newton's method exhibits quadratic convergence. This convergence is in comparison to the linear convergence of steepest descent. For the presented example, the convergence of steepest descent and Newton's method are compared.

## References


[1] L. Armijo. Minimization of functions having lipschitz continuous first-partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.

[2] M. S. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Autmatic Control*, 43:31–45, 1998.

[3] 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.

[4] C.-T. Chen. *Linear System theory and Design*. Oxford University Press, 1999.

[5] F. C. Delmotte. *Multi-Modal Control: From Motion Description Languages to Optimal Control*. PhD thesis, Georgia Institute of Technology, 2006.

[6] M. Egerstedt, S.-I. Azuma, and Y. Wardi. Optimal timing control of switched linear systems based on partial information. *Nonlinear Analysis: Theory, Methods and Applications*, 65:1736–1750, 2006.

[7] 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.

[8] E. R. Johnson and T. D. Murphey. Second order switching time optimization for time-varying nonlinear systems. *IEEE Conference on Decision and Control*, 48:5281–5286, 2009.

[9] C. T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, 1999.

[10] D. G. Luenberger. *Linear and Nonlinear Programming*. Springer, 2nd edition, 2003.

[11] A. Schild, X. C. Ding, M. Egerstedt, and J. Lunze. Design of optimal switching surfaces for switched autonomous systems. *IEEE Conference on Decision and Control*, 48:5293–5298, 2009.

[12] C. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proc. of the IEEE*, 88:949–970, 2000.

[13] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43:509–521, 1998.

[14] E. I. Verriest. Optimal control for switched point delay systems with refractory period. *IFAC World Congress*, 2005.

[15] Y. Wardi, X. C. Ding, M. Egerstedt, and S. Azuma. On-line optimization of switched-mode systems: Algorithms and convergence properties. *IEEE Conference on Decison and Control*, 46, 2007.

[16] X. Xu and P. J. Antsaklis. Optimal control of switched autonomous systems. *IEEE Conference on Decision and Control*, 41:4401 – 4406, 2002.