Finite Set Control Transcription for Optimal Control Applications

Stuart A. Stanton
Captain, USAF
Ph.D. Candidate

Department of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin

Doctoral Dissertation Defense, 26 March 2009

The views expressed here are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Zermelo Navigation Problem

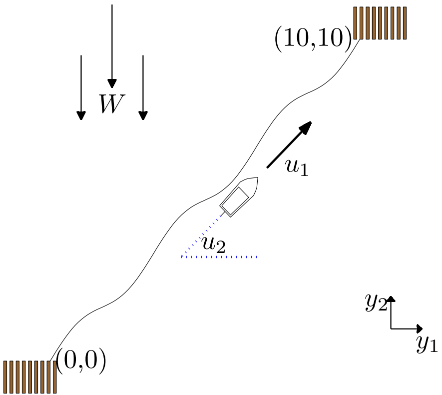


Figure: The Zermelo Navigation Problem

Zermelo Navigation Problem: 1 Segment Solution

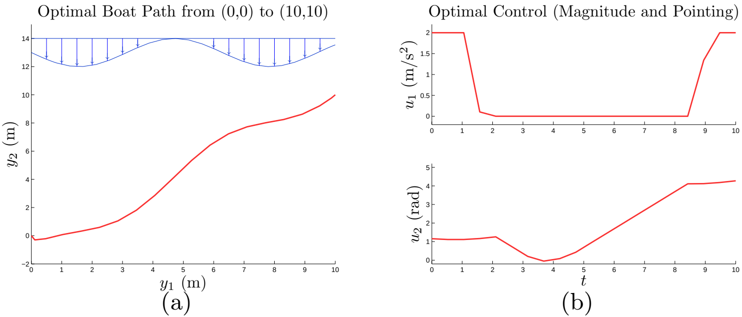


Figure: Optimal Path (a) and Control (b) for the Minimum Acceleration Zermelo Problem

Zermelo Navigation Problem: 3 Segment Solution

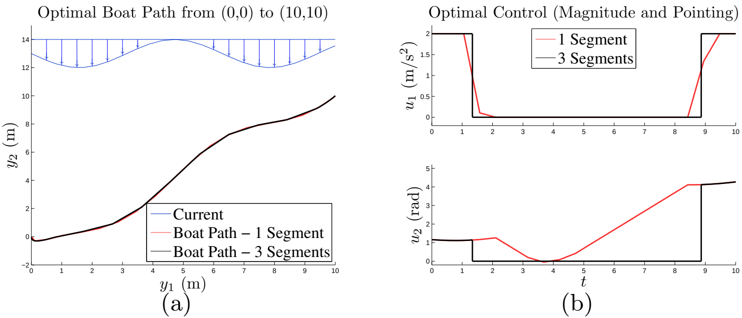


Figure: Optimal Path (a) and Control (b) for the Minimum Acceleration Zermelo Problem

Introduction

- Research Motivation
  - Accomodate realistic actuator constraints into the discovery of optimal solutions
    - Optimality vs. Implementability
  - Realize bang-bang control solutions without ambiguity
- Realm of Hybrid Systems
  - Medical diagnostics
  - Psychology
  - Education
  - Economy
  - Management
  - Sociology
  - Engineering...

System Description

- Hybrid System Dynamics

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u})$$

- Continuous States

$$\begin{aligned} \boldsymbol{y} &= \begin{bmatrix} y_1 & \cdots & y_{n_y} \end{bmatrix}^T \\ y_i &\in \mathbb{R} \end{aligned}$$

- Discrete Controls

$$\begin{aligned} \boldsymbol{u} &= \begin{bmatrix} u_1 & \cdots & u_{n_u} \end{bmatrix}^T \\ u_i &\in \mathbb{U}_i = \{\tilde{u}_{i,1}, \ldots, \tilde{u}_{i,m_i}\} \end{aligned}$$

- Examples
  - Switched Systems
  - Task Scheduling and Resource Allocation Models
  - On-Off Control Systems
  - Control Systems with Saturation Limits

Solving an Optimal Control Problem Numerically

$$
\begin{aligned}
\text{Minimize } \mathcal{J} \quad &= \quad \phi(t_0, \boldsymbol{y}_0, t_f, \boldsymbol{y}_f) + \int_{t_0}^{t_f} L(t, \boldsymbol{y}, \boldsymbol{u})\, dt \\
&\text{subject to} \\
\dot{\boldsymbol{y}} \quad &= \quad \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u}), \\
\boldsymbol{0} \quad &= \quad \boldsymbol{\psi}_0(t_0, \boldsymbol{y}_0), \\
\boldsymbol{0} \quad &= \quad \boldsymbol{\psi}_f(t_f, \boldsymbol{y}_f), \\
\boldsymbol{0} \quad &= \quad \boldsymbol{\beta}(t, \boldsymbol{y}, \boldsymbol{u})
\end{aligned}
$$

↘ ?

$$
\begin{aligned}
\text{Minimize } \mathcal{J} \quad &= \quad F(\boldsymbol{x}) \\
&\text{subject to} \\
\boldsymbol{c}(\boldsymbol{x}) \quad &= \quad \left[\boldsymbol{c}_{\dot{y}}^T(\boldsymbol{x})\ \boldsymbol{c}_{\psi_0}^T(\boldsymbol{x})\ \boldsymbol{c}_{\psi_f}^T(\boldsymbol{x})\ \boldsymbol{c}_{\beta}^T(\boldsymbol{x}) \right]^T = \boldsymbol{0}
\end{aligned}
$$

↘ ✓

NLP Solver

FSCT Method Overview

- Parameter vector consists only of states and times

$$\boldsymbol{x} = \left[\cdots \; y_{i,j,k} \; \cdots \; \cdots \; \Delta t_{i,k} \; \cdots \; t_0 \; t_f\right]^T$$

- Control history is completely defined by
  - Pre-specified control sequence
  - Control value time durations, $\Delta t_{i,k}$, between switching points
- Key parameterization factors

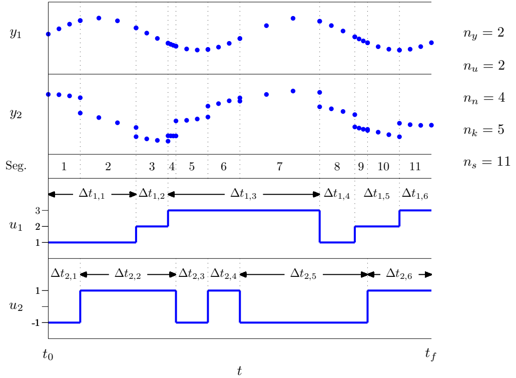
n_y Number of States
n_u Number of Controls
n_n Number of Nodes
n_k Number of Knots
n_s Number of Segments ($n_s = n_u n_k + 1$)

FSCT Method Overview

$$\boldsymbol{x} = [\cdots\ y_{i,j,k}\ \cdots\ \cdots\ \Delta t_{i,k}\ \cdots\ t_0\ t_f]^T$$

$$u_1 \in \mathbb{U}_1 = \{1,2,3\}, \quad u_2 \in \mathbb{U}_2 = \{-1,1\}.$$

$$\boldsymbol{u}^* = \begin{bmatrix} 1 & 2 & 3 & 1 & 2 & 3 \\ -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}$$



FSCT Interface

User Inputs

- System Dynamics

  $$\boldsymbol{f},\ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}},\ \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}},\ \frac{\partial \boldsymbol{f}}{\partial t}$$

- Extra Constraints/ Objectives
- Input File
  - Initial Guess
  - Control Sequence
  - User Options
  - Problem Specifics

↓

FSCT Suite

- Constraint/ Objective Libraries
- Data Manipulation Library
- SNOPT Interface Library

↓

Outputs

- Final Solution

  $$\boldsymbol{x},\ F(\boldsymbol{x}),\ \boldsymbol{c}(\boldsymbol{x}),\ \boldsymbol{\eta}$$

- Convergence Log
- Iteration History

FSCT Process

$$\boldsymbol{x} = [\cdots\ y_{i,j,k}\ \cdots\ \cdots\ \Delta t_{i,k}\ \cdots\ t_0\ t_f]^T$$

↓

Constraint Subroutines

- Initial States
- Initial Time
- Simpson Continuity
- Segment Continuity
- Time
- Final States
- Final Time
- User Defined Constraints

Objective Subroutines

- Control
- Time
- User Defined Objectives

↓

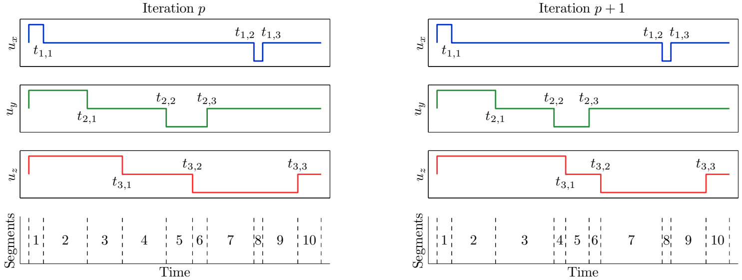
$$F(\boldsymbol{x}) \quad \boldsymbol{c}(\boldsymbol{x})$$

↓

NLP Iteration

Switching Segments and Time for Multiple Independent Controls

- Knots designate switching times in each control axis
- Segments are bounded by switches in any control
- The chronological ordering of knots changes at each iteration of the optimization



Derivative Discontinuities: An Example

- Function

$$f(x_1, x_2) = \min\{x_1, x_2\} = \begin{cases} x_1, & x_1 < x_2, \\ x_1 = x_2, & x_1 = x_2, \\ x_2, & x_1 > x_2. \end{cases}$$

- Derivative

$$\frac{\partial f}{\partial x_1} = \begin{cases} 1, & x_1 < x_2, \\ \text{undefined}, & x_1 = x_2, \\ 0, & x_1 > x_2. \end{cases}$$

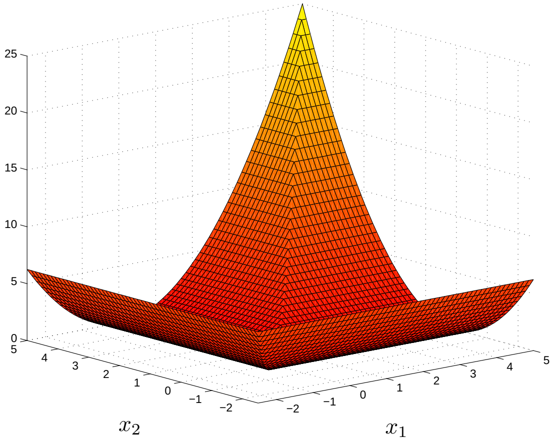
- Candidates for Numerical Implementation

$$\begin{aligned} \text{Forward:} \quad \left.\frac{\partial f}{\partial x_1}\right|_{x_1=x_2} &= \frac{f(x_1+\delta, x_2) - f(x_1, x_2)}{\delta} &= 0, \\ \text{Backward:} \quad \left.\frac{\partial f}{\partial x_1}\right|_{x_1=x_2} &= \frac{f(x_1, x_2) - f(x_1-\delta, x_2)}{\delta} &= 1, \\ \text{Central:} \quad \left.\frac{\partial f}{\partial x_1}\right|_{x_1=x_2} &= \frac{f(x_1+\delta, x_2) - f(x_1-\delta, x_2)}{2\delta} &= \frac{1}{2}. \end{aligned}$$

- Choosing an analytic expression for $\left.\frac{\partial f}{\partial x_1}\right|_{x_1=x_2}$ is equivalent to selecting a finite differencing scheme for numerically evaluated derivatives

Derivative Discontinuities: An Example

$$F(\boldsymbol{x}) = (\min\{x_1, x_2\})^2$$



Derivative Discontinuities: An Example

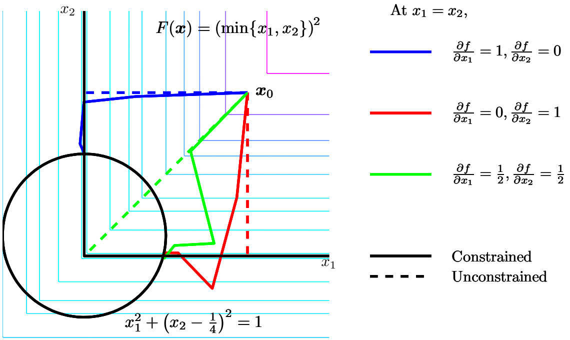


Figure: Effects of Alternative Derivative Definitions on Optimization Path

Implementation of Numerical Derivatives

- An optimizer may calculate finite differences automatically in the absence of analytic derivative expressions
- FSCT method's derivatives may not be effectively evaluated without user intervention
  - Some sophisticated optimizers perform initialization routines to more efficiently calculate numerical derivatives
    - Functions are evaluated from an initial or random point of $\boldsymbol{x}$ to determine the structure of the Jacobian matrix
    - If the initialization routine falsely identifies elements as constant (zero or nonzero), then proper derivatives are not evaluated for future iterations when the knot arrangement is different
- Overcome with user defined procedures to flag *all* potential dependencies (nonzero Jacobian elements) as varying gradients

2-Dimensional Lunar Lander

- Dynamics

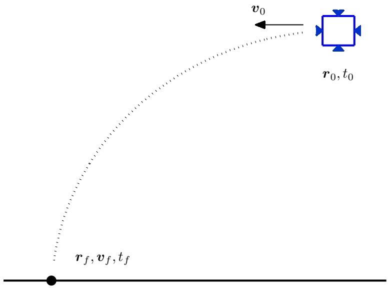
$$\dot{\boldsymbol{y}} = \begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ u_1 \\ -g + u_2 \end{bmatrix}$$

- Controls

$$\begin{aligned} u_1 &\in \{-50,\ 0,\ 50\}\ \mathrm{m/s}^2 \\ u_2 &\in \{-20,\ 0,\ 20\}\ \mathrm{m/s}^2 \end{aligned}$$

- Initial and Final Conditions

$$\begin{aligned} \boldsymbol{r}_0 &= [200\ 15]^T\ \mathrm{km} \\ \boldsymbol{v}_0 &= [-1.7\ 0]^T\ \mathrm{km/s} \\ \boldsymbol{r}_f &= \mathbf{0} \\ \boldsymbol{v}_f &= \mathbf{0} \end{aligned}$$



2-Dimensional Lunar Lander

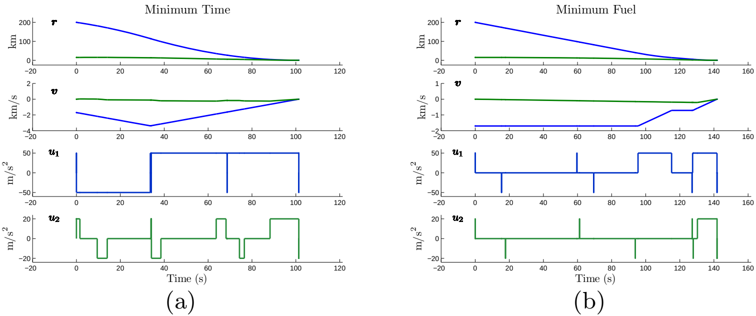
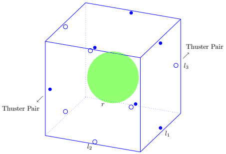


Figure: Optimal Solutions for the Minimum-Time (a) and Minimum-Fuel (b) Lunar Lander Problem

Small Spacecraft Attitude Control: Fixed Thrust

- Fixed thrust cold gas propulsion for arbitrary attitude tracking
  - Reference trajectory defined by ${}^{r}\boldsymbol{q}^{i}{}_{0}$ and ${}^{r}\boldsymbol{\omega}^{i}(t)$
- Minimize deviations between body frame and reference frame with minimum propellant mass consumption



$$\mathcal{J} = \beta_1 p_f - \beta_2 m_{p_f}$$

$$p_f - p_0 = \int_{t_0}^{t_f} \dot{p}\, dt = \int_{t_0}^{t_f} \left({}^{r}\boldsymbol{q}_v{}^{b}\right)^T \left({}^{r}\boldsymbol{q}_v{}^{b}\right) dt.$$

$$\dot{\boldsymbol{y}} = \begin{bmatrix} {}^{b}\dot{\boldsymbol{q}}^{i} \\ {}^{b}\dot{\boldsymbol{\omega}}^{i} \\ \dot{m}_p \\ {}^{r}\dot{\boldsymbol{q}}^{i} \\ \dot{p} \end{bmatrix} = \boldsymbol{f}(t, \boldsymbol{y}, \boldsymbol{u})$$

$$u_i \in \mathbb{U} = \{-1, 0, 1\}$$

- where u_i indicates for each principal axis whether the positive-thrusting pair, the negative-thrusting pair, or neither is in the on position

Small Spacecraft Attitude Control: Fixed Thrust

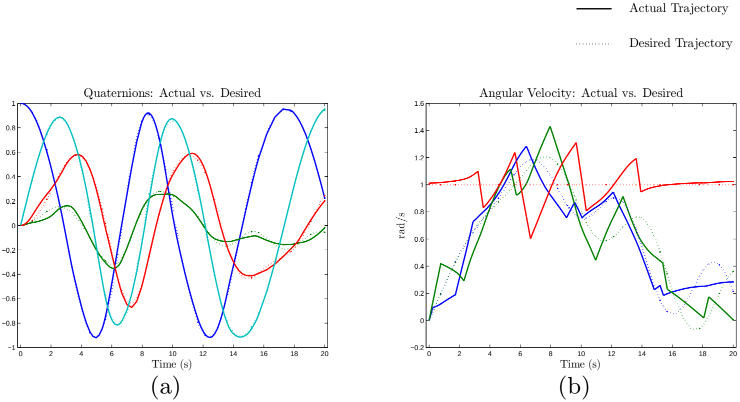
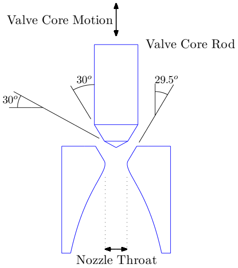


Figure: Fixed Thrust Attitude Control

Small Spacecraft Attitude Control: Variable Thrust

- Variable thrust cold gas propulsion
  - Valve rod modifies nozzle throat area
- Include *additional* states to model variable thrust
  - Resulting dynamics are still hybrid
- States and Controls



$$\boldsymbol{y} = \begin{bmatrix} {}^b\boldsymbol{q}^i \\ {}^b\boldsymbol{\omega}^i \\ m_p \\ \boldsymbol{d} \\ \boldsymbol{v} \\ {}^r\boldsymbol{q}^i \\ p \end{bmatrix} \qquad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{a} \end{bmatrix} \qquad \begin{matrix} w_i \in \{0, 1\} \\ a_i \in \{-1, 0, 1\} \end{matrix}$$

  - w_i indicates whether the i^{th} thruster pair is on or off
  - a_i indicates the acceleration of the valve core rods of the i^{th} thruster pair

Small Spacecraft Attitude Control: Variable Thrust

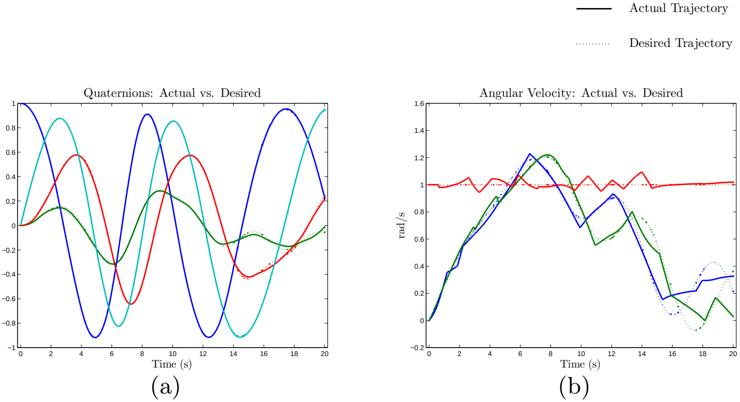


Figure: Variable Thrust Attitude Control

Libration Point Formations: Formation Limitations

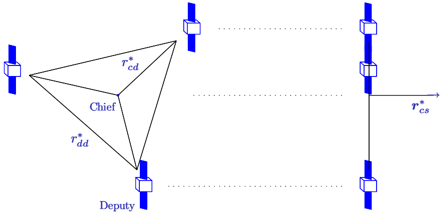


Figure: Formation Pointing

- Fixed size, shape, and orientation of the formation
- Fixed orientation of each member of the formation (deputy spacecraft)

Libration Point Formations: Dynamical Sensitivities

- Previous investigations have focused on unconstrained continuous control solutions
  - Linear and nonlinear; feasible and optimal solutions
  - Non-natural formations require extremely precise control ($<$ nm/s^2 accelerations)
- These controls are impossible to implement with existing actuator technology

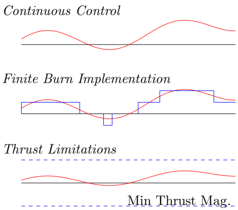


Figure: Implementing a Continuous Control Solution

- Cannot reproduce the fidelity of continuous control
- Continuous control may even be *smaller* than minimum thrust bound

Libration Point Formations: Control Limitations

- Fixed thruster location on each spacecraft body
- Specified thrust acceleration magnitude
  - Based on actuator performance capability

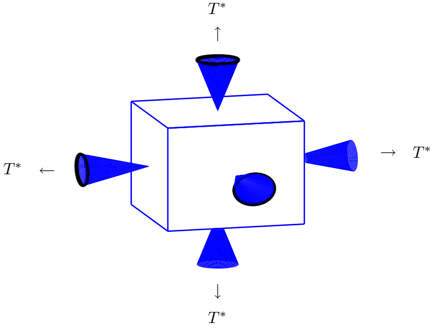


Figure: Spacecraft Body

Costs and Constraints

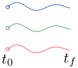
- Constraints
  - Initial time and states specified
  - Final time and formation size and plane specified
    - $r^*_{cd} = 1$ km distance between chief and deputy, $r^*_{dd} = 1.73$ km distance between deputies
    - Specified pointing $\boldsymbol{r}^{*\mathcal{I}}_{cs} = [1\ 0\ 0]$
  - State continuity (differential constraints) by segment
  - State equality across segments (at knots)
- Weighted Costs
  - Minimize thrust
  - Minimize formation size deviations along trajectory
  - Minimize formation plane deviations along trajectory

$$J = w_1 J_1 + w_2 J_2 + w_3 J_3$$
$$F(\boldsymbol{x}) = w_1 F_1(\boldsymbol{x}) + w_2 F_2(\boldsymbol{x}) + w_3 F_3(\boldsymbol{x})$$

Baseline Initial Guess

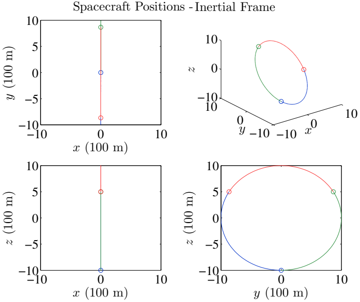
Trajectory Legend

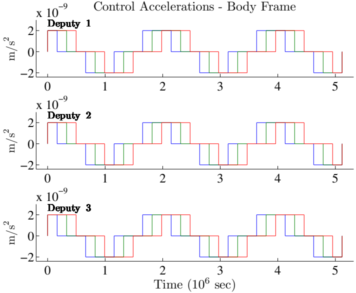
Deputy 1 Trajectory
Deputy 2 Trajectory
Deputy 3 Trajectory



Control Legend

Axis 1 Control (u_x)
Axis 2 Control (u_y)
Axis 3 Control (u_z)





Baseline Solution

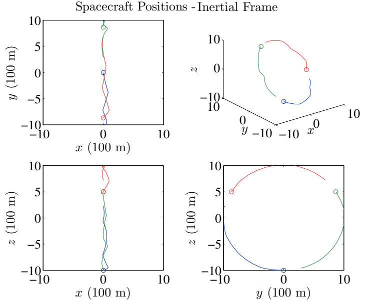
Trajectory Legend

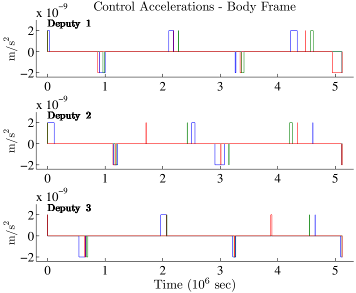
- Deputy 1 Trajectory
- Deputy 2 Trajectory
- Deputy 3 Trajectory

t_0 t_f

Control Legend

- Axis 1 Control (u_x)
- Axis 2 Control (u_y)
- Axis 3 Control (u_z)





Traffic Flow Management

- Density of traffic, ρ, in vehicles/distance unit

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial \tau} = 0,$$

- Velocity of traffic, v, in distance unit/time unit

$$v = v_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right).$$

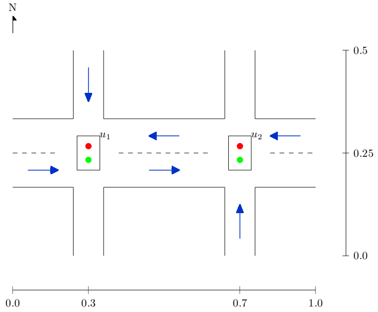
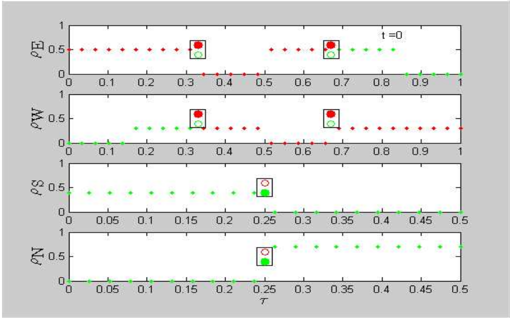


Figure: Traffic Flow Problem: Four Lanes, Two Intersections

Traffic Flow Management



Traffic Flow Management

Conclusions

- Finite Set Control Transcription: This investigation yielded a new metholology for treating *Hybrid Control Systems*
  - Initially designed for *continuous states* and *discrete controls*
  - Extensions are demonstrated for broader classes of hybrid systems
- Future Work
  - Further Development of the Methodology
    - Apply concepts to direct/indirect shooting methods
    - Investigate mesh refinement
    - Explore extended classes of hybrid systems
  - Further Application of the Methodology
    - Further depth in relevant problems, such as *Spacecraft Attitude Control* and *Traffic Management*
    - Petroleum engineering: *Smart Well Technology*
    - Others...

Action Items

- Spelling and Rewording
  - Marchand: Ch 6-7, App A (Ch 5, App B)
  - D'Souza: Ch 1-7
- Transition to Chapter 5 (Libration Point Formations)
- Figure Size and Legibility
- Formatting Considerations

... To be completed before 8 May!

Backup

The Optimization Parameters

$$\boldsymbol{x} = \left[\cdots\ y_{i,j,k}\ \cdots\ \cdots\ \Delta t_{i,k}\ \cdots\ t_0\ t_f\right]^T$$

Array	Description	Dimension	Element
$\boldsymbol{Y}_{n_y,n_n,n_s}$	States by node	$n_y \times n_n \times n_s$	$\boldsymbol{y}_{j,k}$ or $y_{i,j,k}$
$\boldsymbol{U}^*_{n_u,n_k+1}$	Pre-specified controls	$n_u \times (n_k+1)$	$u^*_{i,k}$
$\boldsymbol{U}_{n_u,n_s}$	Controls by segment	$n_u \times n_s$	$u_{i,k}$
$\boldsymbol{U}_{n_u,n_n,n_s}$	Controls by node	$n_u \times n_n \times n_s$	$\boldsymbol{u}_{j,k}$ or $u_{i,j,k}$
$\boldsymbol{T}_{n_n,n_s}$	Node times	$n_n \times n_s$	$t_{j,k}$
$\boldsymbol{T}_{n_u,n_k}$	Knot times	$n_u \times n_k$	$t_{i,k}$
$\Delta\boldsymbol{T}_{n_u,n_k+1}$	Axis durations	$n_u \times (n_k+1)$	$\Delta t_{i,k}$
$\boldsymbol{T}'_{n_s+1}$	Unordered knot times	$0 \ldots n_s$	t'_k
$\boldsymbol{T}_{n_s+1}$	Ordered knot times	$0 \ldots n_s$	t_k
n_s	Number of segments	$n_u n_k + 1$	

$\boldsymbol{x} \to \boldsymbol{Y}_{n_y,n_n,n_s},\ \Delta\boldsymbol{T}_{n_u,n_k+1},\ t_0,\ t_f$	
$t_0,\ \Delta\boldsymbol{T}_{n_u,n_k+1} \to \boldsymbol{T}_{n_u,n_k}$	$t_{i,k} = t_0 + \sum_{\kappa=1}^{k} \boxed{\Delta t_{i,\kappa}}$
$t_0,\ \boldsymbol{T}_{n_u,n_k},\ t_f \to \boldsymbol{T}'_{n_s+1}$ $\boldsymbol{T}'_{n_s+1} \to \boldsymbol{T}_{n_s+1}$	$\left[t'_0\ t'_1\ \cdots\ t'_\kappa\ \cdots\ t'_{n_s-1}\ t'_{n_s}\right] \equiv \left[t_0\ t_{1,1}\ \cdots\ t_{i,k}\ \cdots\ t_{n_u,n_k}\ t_f\right]$
$\boldsymbol{T}_{n_u,n_k},\ \boldsymbol{T}_{n_s+1},\ \boldsymbol{U}^*_{n_u,n_k+1} \to \boldsymbol{U}_{n_u,n_s}$	$u_{i,1} = u^*_{i,1}$ $u_{i,k} = u^*_{i,\kappa_i}$ $u_{i,k+1} = \begin{cases} u^*_{i,\kappa_i+1}, & t_k \equiv t_{i,\kappa_i}, \\ u^*_{i,\kappa_i}, & \text{otherwise} \end{cases}$
$\boldsymbol{U}_{n_u,n_s} \to \boldsymbol{U}_{n_u,n_n,n_s}$	$u_{i,j,k} = u_{i,k}$

Dynamical Constraints Using Simpson Integration Equations

$$\boldsymbol{c}_{\dot{y}}(\boldsymbol{x}) = \left[\boldsymbol{c}_{\dot{y}_{1,1}}^T(\boldsymbol{x}) \ \cdots \ \boldsymbol{c}_{\dot{y}_{j,k}}^T(\boldsymbol{x}) \ \cdots \ \boldsymbol{c}_{\dot{y}_{n_n-1,n_s}}^T(\boldsymbol{x})\right]^T$$

where

$$\begin{aligned}\boldsymbol{c}_{\dot{y}_{j,k}}(\boldsymbol{x}) &= \boldsymbol{y}_{j+1,k} - \boldsymbol{y}_{j,k} \\ &\quad - \frac{h}{6}\left[\boldsymbol{f}\left(t_{j,k}, \boldsymbol{y}_{j,k}, \boldsymbol{u}_{j,k}\right) + 4\boldsymbol{f}\left(t_m, \boldsymbol{y}_m, \boldsymbol{u}_m\right) + \boldsymbol{f}\left(t_{j+1,k}, \boldsymbol{y}_{j+1,k}, \boldsymbol{u}_{j+1,k}\right)\right]\end{aligned}$$

and

$$\begin{aligned} h &= \frac{t_k - t_{k-1}}{n_n - 1}, \\ t_{j,k} &= t_{k-1} + h(j-1), \\ t_m &= \frac{1}{2}\left(t_{j,k} + t_{j+1,k}\right) \end{aligned}$$

with midpoint states and controls

$$\begin{aligned} \boldsymbol{y}_m &= \frac{1}{2}\left(\boldsymbol{y}_{j,k} + \boldsymbol{y}_{j+1,k}\right) + \frac{h}{8}\left(\boldsymbol{f}_{j,k} - \boldsymbol{f}_{j+1,k}\right), \\ \boldsymbol{u}_m &= \frac{1}{2}\left(\boldsymbol{u}_{j,k} + \boldsymbol{u}_{j+1,k}\right) = \boldsymbol{u}_{j,k} = \boldsymbol{u}_{j+1,k}. \end{aligned}$$

Partial Derivatives for the Simpson Integration Equations

$$\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{y}_{j,k}} = -\boldsymbol{I} - \frac{h}{6}\left(\frac{\partial \boldsymbol{f}_{j,k}}{\partial \boldsymbol{y}_{j,k}} + 4\frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{y}_m}\frac{\partial \boldsymbol{y}_m}{\partial \boldsymbol{y}_{j,k}}\right)$$

$$\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{y}_{j+1,k}} = \boldsymbol{I} - \frac{h}{6}\left(4\frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{y}_m}\frac{\partial \boldsymbol{y}_m}{\partial \boldsymbol{y}_{j+1,k}} + \frac{\partial \boldsymbol{f}_{j+1,k}}{\partial \boldsymbol{y}_{j+1,k}}\right)$$

$$\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{u}_{j,k}} = -\frac{h}{6}\left(\frac{\partial \boldsymbol{f}_{j,k}}{\partial \boldsymbol{u}_{j,k}} + 4\frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{u}_m}\frac{\partial \boldsymbol{u}_m}{\partial \boldsymbol{u}_{j,k}}\right)$$

$$\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{u}_{j+1,k}} = -\frac{h}{6}\left(4\frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{u}_m}\frac{\partial \boldsymbol{u}_m}{\partial \boldsymbol{u}_{j+1,k}} + \frac{\partial \boldsymbol{f}_{j+1,k}}{\partial \boldsymbol{u}_{j+1,k}}\right)$$

$$\begin{aligned}\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial t_{k-1}} = &-\frac{1}{6}\left(\boldsymbol{f}_{j,k} + 4\boldsymbol{f}_m + \boldsymbol{f}_{j+1,k}\right)\frac{\partial h}{\partial t_{k-1}} \\ &- \frac{h}{6}\left[\frac{\partial \boldsymbol{f}_{j,k}}{\partial t_{j,k}}\frac{\partial t_{j,k}}{\partial t_{k-1}} + 4\left(\frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{y}_m}\frac{\partial \boldsymbol{y}_m}{\partial h}\frac{\partial h}{\partial t_{k-1}} + \frac{\partial \boldsymbol{f}_m}{\partial t_m}\frac{\partial t_m}{\partial t_{k-1}}\right) + \frac{\partial \boldsymbol{f}_{j+1,k}}{\partial t_{j+1,k}}\frac{\partial t_{j+1,k}}{\partial t_{k-1}}\right]\end{aligned}$$

$$\begin{aligned}\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial t_{k}} = &-\frac{1}{6}\left(\boldsymbol{f}_{j,k} + 4\boldsymbol{f}_m + \boldsymbol{f}_{j+1,k}\right)\frac{\partial h}{\partial t_{k}} \\ &- \frac{h}{6}\left[\frac{\partial \boldsymbol{f}_{j,k}}{\partial t_{j,k}}\frac{\partial t_{j,k}}{\partial t_{k}} + 4\left(\frac{\partial \boldsymbol{f}_m}{\partial \boldsymbol{y}_m}\frac{\partial \boldsymbol{y}_m}{\partial h}\frac{\partial h}{\partial t_{k}} + \frac{\partial \boldsymbol{f}_m}{\partial t_m}\frac{\partial t_m}{\partial t_{k}}\right) + \frac{\partial \boldsymbol{f}_{j+1,k}}{\partial t_{j+1,k}}\frac{\partial t_{j+1,k}}{\partial t_{k}}\right]\end{aligned}$$

Partial Derivatives for the Simpson Integration Equations

- Divide parameters into state and time elements: $\boldsymbol{x} = \left[\boldsymbol{x}_y^T \ \boldsymbol{x}_t^T\right]^T$
- Partial Derivatives for State-Parameters

$$\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{x}_{y_\gamma}} = \begin{cases} \dfrac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{y}_{j,k}}, & \boldsymbol{x}_{y_\gamma} \equiv \boldsymbol{y}_{j,k}, \\ \dfrac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{y}_{j+1,k}}, & \boldsymbol{x}_{y_\gamma} \equiv \boldsymbol{y}_{j+1,k}, \\ \boldsymbol{0}, & \text{otherwise.} \end{cases}$$

- Partial Derivatives for Time-Parameters

$$\frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial \boldsymbol{x}_t} = \frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial t_{k-1}} \frac{\partial t_{k-1}}{\partial \boldsymbol{x}_t} + \frac{\partial \boldsymbol{c}_{\dot{y}_{j,k}}}{\partial t_k} \frac{\partial t_k}{\partial \boldsymbol{x}_t}.$$

where

$$\frac{\partial t_k}{\partial t_0}, \quad \frac{\partial t_k}{\partial t_f}, \quad \text{and} \quad \frac{\partial t_k}{\partial \Delta t_{i,\kappa}}$$

are determined according to knot ordering

Initial States and Time

- Constraint Function

$$\boldsymbol{c}_{\psi_0}(\boldsymbol{x}) = \begin{bmatrix} y_{1,1,1} - (y_0^*)_1 \\ \vdots \\ y_{i,1,1} - (y_0^*)_i \\ \vdots \\ y_{n_y,1,1} - (y_0^*)_{n_y} \\ t_0 - t_0^* \end{bmatrix}$$

- Jacobian elements

$$\frac{\partial c_{\psi_{0_i}}}{\partial x_\gamma} = \begin{cases} 1, & x_\gamma \equiv y_{i,1,1} \Leftrightarrow \gamma = i, \\ 0, & \text{otherwise,} \end{cases}$$

$$\frac{\partial c_{\psi_{0_{n_y+1}}}}{\partial x_\gamma} = \begin{cases} 1, & x_\gamma \equiv t_0 \Leftrightarrow \gamma = n_y n_n n_s + n_u(n_k+1)+1, \\ 0, & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, n_y$

Segment Continuity Between Knots

- Constraint Function

$$\boldsymbol{c}_s(\boldsymbol{x}) = \begin{bmatrix} y_{1,1,2} & - & y_{1,n_n,1} \\ & \vdots & \\ y_{i,1,k+1} & - & y_{i,n_n,k} \\ & \vdots & \\ y_{n_y,1,n_s} & - & y_{n_y,n_n,n_s-1} \end{bmatrix}$$

- Jacobian elements

$$\frac{\partial c_{s_{n_y(k-1)+i}}}{\partial x_\gamma} = \begin{cases} 1, & x_\gamma \equiv y_{i,1,k+1}, \\ -1, & x_\gamma \equiv y_{i,n_n,k}, \\ 0, & \text{otherwise}, \end{cases}$$

for $i = 1, \ldots, n_y$ and $k = 1, \ldots, n_s - 1$

Time

- Constraint Function

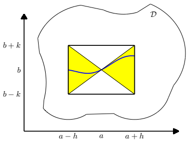
$$\boldsymbol{c}_t(\boldsymbol{x}) = \begin{bmatrix} t_f - t_0 - \sum_{\kappa=1}^{n_k+1} |\Delta t_{1,\kappa}| \\ \vdots \\ t_f - t_0 - \sum_{\kappa=1}^{n_k+1} |\Delta t_{i,\kappa}| \\ \vdots \\ t_f - t_0 - \sum_{\kappa=1}^{n_k+1} |\Delta t_{n_u,\kappa}| \end{bmatrix}$$

- Jacobian elements

$$\frac{\partial c_{t_i}}{\partial x_\gamma} = \begin{cases} 1, & x_\gamma \equiv t_f, \\ -1, & x_\gamma \equiv t_0, \\ & \\ -1, & x_\gamma \equiv \Delta t_{i,k}, \\ 0, & \text{otherwise.} \end{cases}$$

for $i = 1, \ldots, n_u$ and $k = 1, \ldots, n_k + 1$

Existence and Uniqueness



Theorem

Let $\dot{y} = f(t, y, u)$, where u is constant, and $y(a, u) = b$. Suppose that f is continuous in some closed region $\mathcal{D}$ of the t, y plane and hence is bounded. In particular, suppose that

$$|f(t, y, u)| \leq M \text{ over } \mathcal{D}$$

and also that f satisfies a Lipschitz condition in the y argument—that is,

$$|f(t, y_2, u) - f(t, y_1, u)| \leq C|y_2 - y_1|,$$

where the constant C is independent of t or u. Finally, define a rectangle

$$|t - a| \leq h \qquad |y - b| \leq k$$

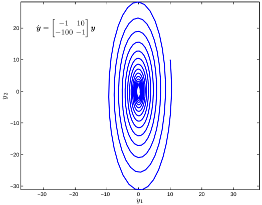
such that $Mh \leq k$. Then $\dot{y} = f(t, y, u)$ has a unique solution $y(t, u)$ in the shaded part of the rectangle.

Two Stable Linear Systems

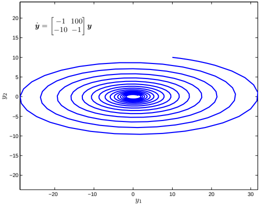
$$\begin{aligned} \dot{\boldsymbol{y}} &= \boldsymbol{f}(\boldsymbol{y}, u) = \boldsymbol{A}_u \boldsymbol{y}, \\ u &\in \{1, 2\}, \end{aligned}$$

where

$$\boldsymbol{A}_1 = \begin{bmatrix} -1 & 10 \\ -100 & -1 \end{bmatrix}, \qquad \boldsymbol{A}_2 = \begin{bmatrix} -1 & 100 \\ -10 & -1 \end{bmatrix}$$



(a) $u = 1$



(b) $u = 2$

Figure: Individually Stable Systems

Two Stable Linear Systems

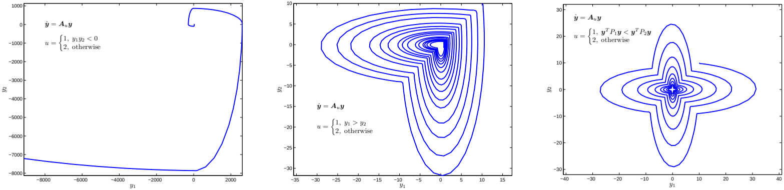
- Several switching laws

(a) Unstable $$u = \begin{cases} 1, & y_1 y_2 < 0 \\ 2, & \text{otherwise} \end{cases}$$

(b) Stable $$u = \begin{cases} 1, & y_1 > y_2 \\ 2, & \text{otherwise} \end{cases}$$

(c) Stable $$u = \begin{cases} 1, & \boldsymbol{y}^T \boldsymbol{P}_1 \boldsymbol{y} < \boldsymbol{y} \boldsymbol{P}_2 \boldsymbol{y} \\ 2, & \text{otherwise} \end{cases}$$

where $\boldsymbol{P}_u \boldsymbol{A}_u + \boldsymbol{A}_u^T \boldsymbol{P}_u = -\boldsymbol{I}$



(a) (b) (c)

Figure: Three Switching Laws

Two Stable Linear Systems

- FSCT Optimization

$$\mathcal{J} = F(\boldsymbol{x}) = t_f - t_0$$
$$\boldsymbol{y}_f^T \boldsymbol{y}_f = 1$$
$$u_k^* = \frac{3}{2} + \frac{1}{2}(-1)^k$$

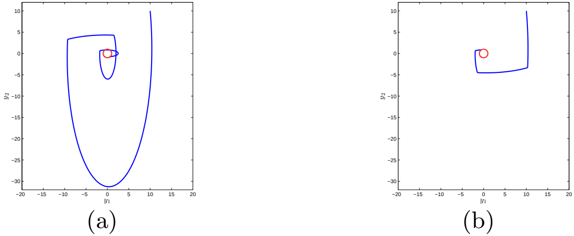


Figure: FSCT Locally Optimal Switching Trajectories

- Optimization implies the switching law

$$u = \begin{cases} 1, & -\frac{1}{m} \le \frac{y_2}{y_1} \le m \\ 2, & \text{otherwise} \end{cases}$$

1-D Lunar Lander

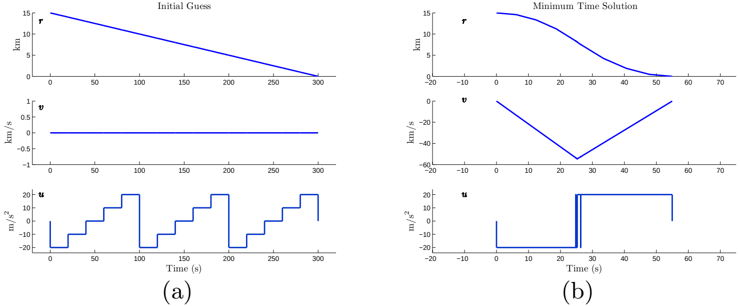


Figure: Initial Guess (a) and Minimum Time Solution (b) for the 1-D Lunar Lander Problem

Equations of Motion

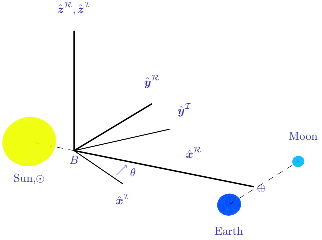


Figure: CR3BP Frame

- Circular Restricted Three-Body Problem (CR3BP) Equations
  - Rotating Frame $\mathcal{R} = \{\hat{\boldsymbol{x}}_{\mathcal{R}}, \hat{\boldsymbol{y}}_{\mathcal{R}}, \hat{\boldsymbol{z}}_{\mathcal{R}}\}$,
- Chief spacecraft lies on a natural trajectory

$$\dot{\boldsymbol{y}}_c = \tilde{\boldsymbol{f}}(\boldsymbol{y}_c, \boldsymbol{u}_c) = \tilde{\boldsymbol{f}}(\boldsymbol{y}_c, \boldsymbol{0})$$

- The lth deputy spacecraft measured relative to the chief

$$\begin{aligned} \boldsymbol{y}_{cd_l} &\equiv \boldsymbol{y}_{d_l} - \boldsymbol{y}_c \\ \dot{\boldsymbol{y}}_{cd_l} &= \boldsymbol{f}(t, \boldsymbol{y}_{cd_l}, \boldsymbol{u}_{d_l}) \end{aligned}$$

Reference Halo Orbit

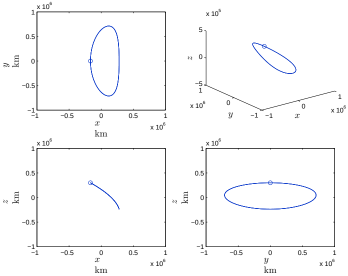


Figure: Reference Halo Orbit for Chief Spacecraft with Origin at L_1

- Chief trajectory is a Halo Orbit about L_1 (approx. 148×10^6 km in the $\hat{\boldsymbol{x}}_{\mathcal{R}}$)
  - At epoch, chief is at northern most point (300,000 km in $\hat{\boldsymbol{z}}_{\mathcal{R}}$)

Impacts of Fixed Spacecraft Orientation

- A traditional finite burn formulation specifies thrust (acceleration) magnitude, but not direction
  - Assumes spacecraft can re-orient to deliver required thrust vector
  - Control space $\mathcal{U}_1$: $\boldsymbol{u}^T\boldsymbol{u} = (T^*)^2$
- If spacecraft orientation is predetermined (according to other mission requirements)
  - Actuator configuration must provide 3-axis maneuverability
    - Assume thrusters are located on principal axes of body frame $\mathcal{B} \equiv \{\hat{\boldsymbol{x}}_\mathcal{B}, \hat{\boldsymbol{y}}_\mathcal{B}, \hat{\boldsymbol{z}}_\mathcal{B}\}$
  - Control space $\mathcal{U}_2$: $u_i(u_i - T^*)(u_i + T^*) = 0, i = \hat{\boldsymbol{x}}_\mathcal{B}, \ldots, \hat{\boldsymbol{z}}_\mathcal{B}$

Fixed spacecraft orientation leads to discrete optimization, which gradient-type NLP algorithms cannot support.

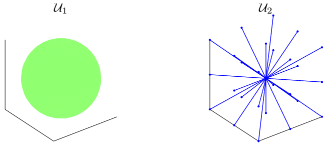
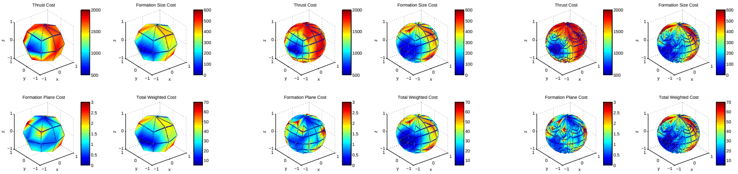


Figure: Control Spaces (a) $\mathcal{U}_1$ (Orientation Free), and (b) $\mathcal{U}_2$ (Orientation Fixed)

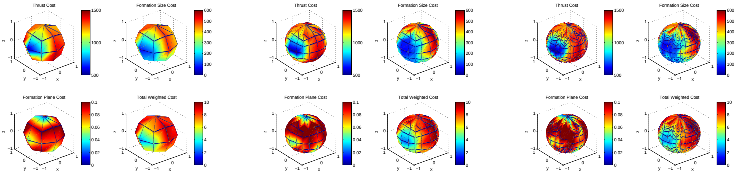
Pointing Survey: Formation Emphasis



(a) 26 Points (b) 98 Points (c) 290 Points

Figure: Formation Emphasis: Comparison of 26, 98, 290 Points

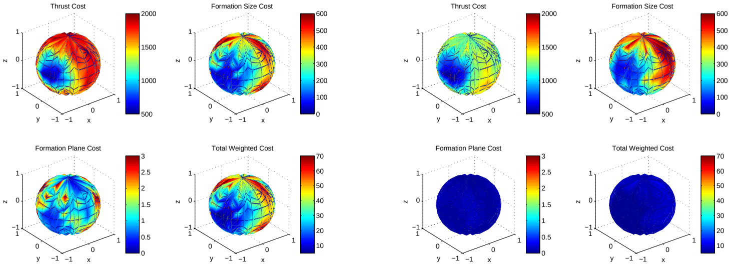
Pointing Survey: Plane Emphasis



(a) 26 Points (b) 98 Points (c) 290 Points

Figure: Plane Emphasis: Comparison of 26, 98, 290 Points

Formation vs. Plane Emphasis Comparison

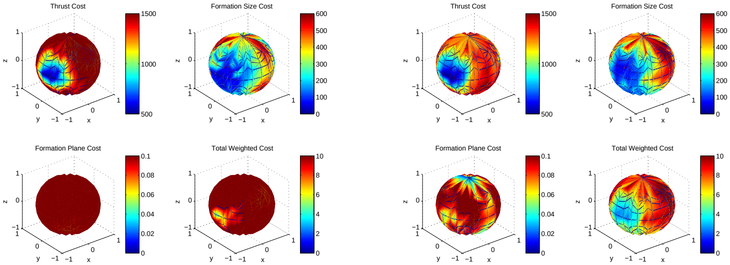


(a) Formation Emphasis

(b) Plane Emphasis

Figure: Formation vs. Plane Emphasis: Scaling for Formation Emphasis

Formation vs. Plane Emphasis Comparison



(a) Formation Emphasis

(b) Plane Emphasis

Figure: Formation vs. Plane Emphasis: Scaling for Plane Emphasis

Conclusions

- A modified collocation method with a segment-time switching algorithm leads to highly constrained control solutions
- Generalized formulation allows users to input
  - formation configuration, size, orientation, and rotation rate
  - thruster capability and placement
  - dynamic model and reference trajectory
  - initial and terminal conditions
- Suited to aid in establishing requirements and capabilities for highly constrained formations

Conclusions

- This investigation explores the range of applications of the FSCT method
  - The applicability of the method extends to all engineering disciplines
- FSCT vs. Multiple Lyapunov Functions
  - Optimal control laws may be extracted whose performance exceeds those derived using a Lyapunov argument
- Multiple independent decision inputs managed simultaneously
- Solutions derived via the FSCT method are utilized in conjunction with a hybrid system model predictive control scheme
  - Optimized control schedules can be realized in the context of potential perturbations or other unknowns
- Some continuous control input systems may be more accurately described as systems ultimately relying on discrete decision variables
  - Continuous control variables may often be extended into a set of continuous state variables and discrete inputs