Space Mission Sustainability Leveraging Picard-Chebyshev Methods and Optimal Control

Robyn Woollands
University of Illinois Urbana-Champaign

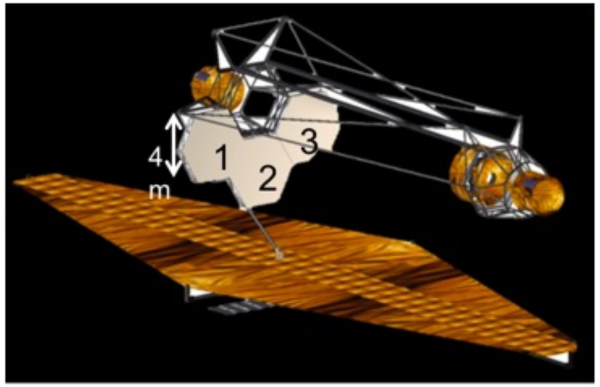
2024 International Forum on Deep Space Missions and Trajectory Optimization
12th edition of GTOC
15 Jan 2024

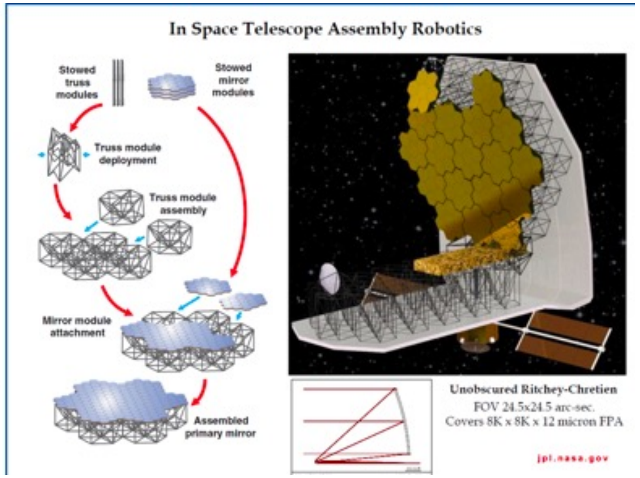
Overview

- Motivation
- Part I
  - Motivation
  - Picard-Chebyshev Methods
  - Fuel-Optimal Control Formulation
  - Gravity Approximations & Segmentation Scheme
  - Trajectory Optimization Simulations
- Part II
  - Motivation
  - Thruster Pointing Constrained Fuel-Optimal Control Formulation
  - Trajectory Optimization Simulations
- Conclusion

Motivation

- **Space Situational Awareness & Space Sustainability at Illinois** (SSASSI)
  - **Computational research:** astrodynamics, trajectory optimization, high-fidelity orbit propagation, Picard-Chebyshev methods, space mission design
  - **Goal:** Conduct research and develop tools to help facilitate and enable large-scale & routine in-space servicing and assembly missions (Earth orbit, cislunar space, SEL2)





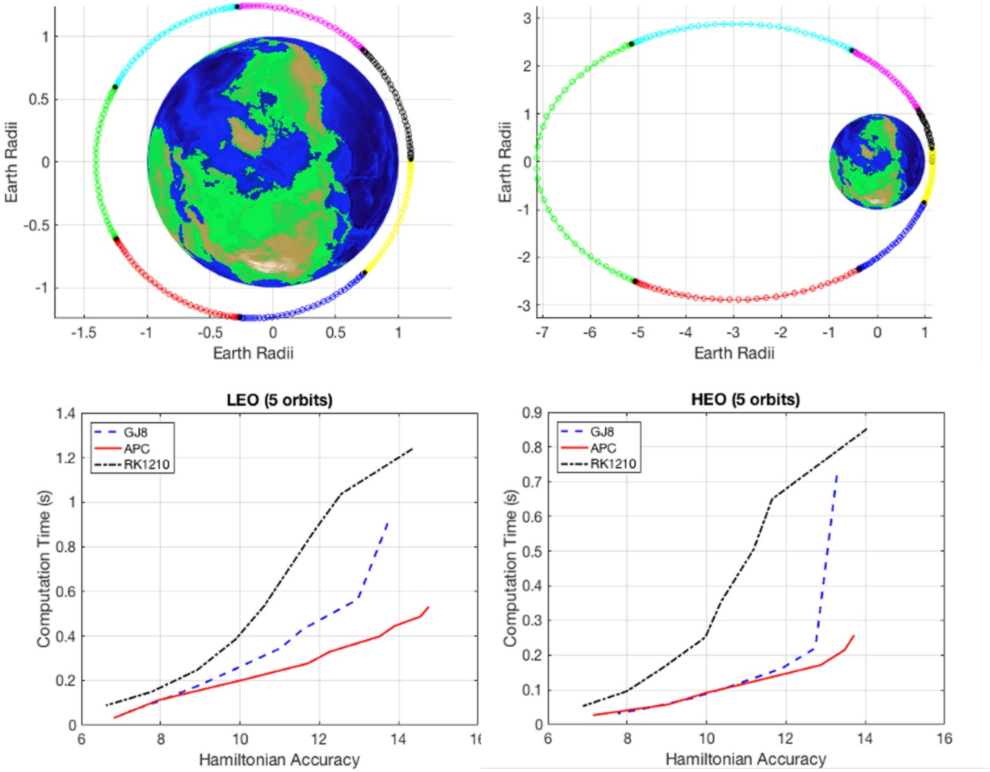
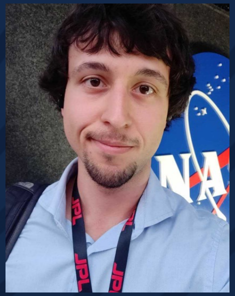


Figure Credit: https://exoplanets.nasa.gov/exep/technology/in-space-assembly/

Figure Credit: Woollands & Junkins, JGCD, 2019.



Alex Pascarella
PhD student, UIUC

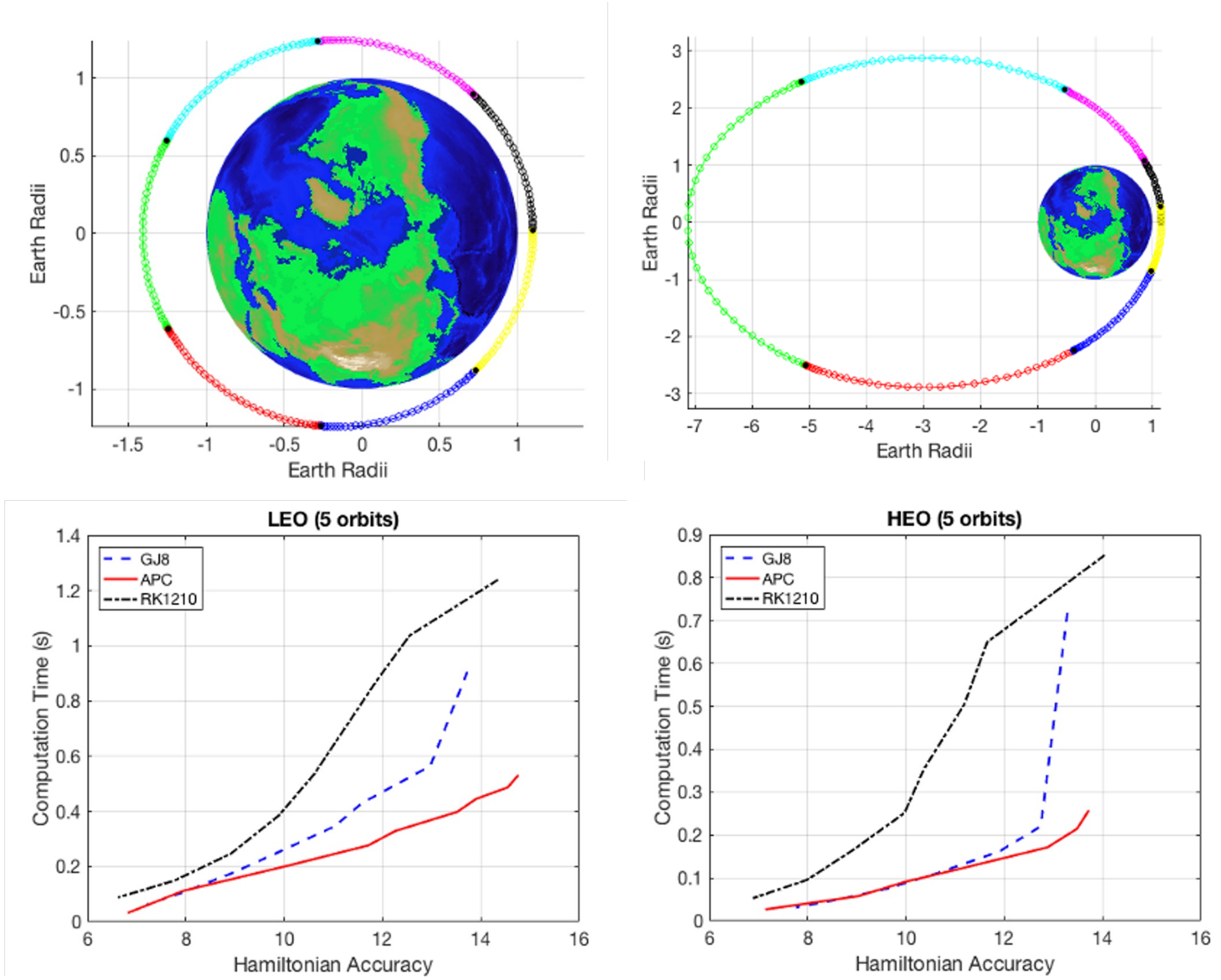
PART I

Picard-Chebyshev Methods for Bang-Bang Dynamics

PART I: Motivation

- The Adaptive Picard-Chebyshev (APC) method was shown to outperform state-of-the-practice numerical integrators (Woollands & Junkins, 2019).
- Two to three times speedup when using a high-fidelity (70x70) gravity model for Earth orbiting satellites.
- Speedup achieved due to development of a variable fidelity force model (not possible for explicit methods).

- **Can we use APC for low-thrust orbit propagation?**
- **Challenge 1**: Automatically detect and accurately fit bang-bang thruster on/off switches
- **Challenge 2**: Develop an automated segmentation scheme that accommodates bang-bang switches.



Woollands & Junkins, “Nonlinear Differential Equations Solvers via Adaptive Picard-Chebyshev Iteration: Applications in Astrodynamics”, JGCD, 2019.

Picard Iteration

Picard iteration is a successive path approximation technique for solving differential equations of the following form:

$$\frac{d\boldsymbol{x}(t)}{dt} = f\big(t, \boldsymbol{x}(t)\big), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \qquad \boldsymbol{x} \in R^{1\times n}$$

This can be rearranged to the following integral equation:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{x}(\tau))d\tau$$

A series of trajectory approximations (Picard iteration) can be generated by:

$$\boldsymbol{x}^{\boldsymbol{i}}(t) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{x}^{i-1}(\tau))d\tau\,, \qquad i = 1,2,\ldots$$

Picard Convergence Theorem

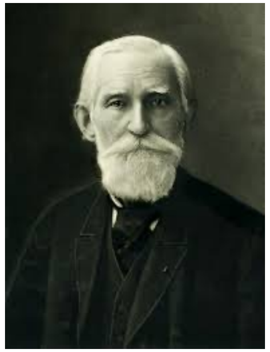
If there is a time interval $|t - t_0| < \delta$ and a starting trajectory $\boldsymbol{x}^0(t)$ satisfying $\|\boldsymbol{x}(t) - \boldsymbol{x}^0(t)\| < \Delta$, for suitable finite bounds (δ, Δ), then the Picard sequence converges.

Charles Emile Picard
(1856-1941)

Picard-Chebyshev Methods

Picard-Chebyshev integration methods combine the techniques of two great mathematicians…

Charles Emile Picard
(1856-1941)



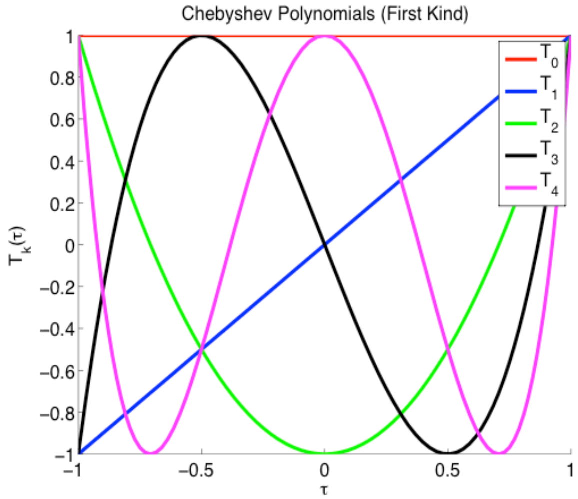
Pafnuty Chebyshev
(1821-1894)

Developed a path approximation method for solving differential equations

$$\boldsymbol{x}^{\boldsymbol{i}}(t) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} \underbrace{\boldsymbol{f}(\tau, \boldsymbol{x}^{i-1}(\tau))d\tau}, \qquad i = 1,2, \ldots$$

Chebyshev polynomials are used to ***approximate the integrand*** in the Picard iteration sequence.

Developed orthogonal Chebyshev polynomials



$$T_k(\tau) = \cos(k \cos^{-1}(\tau))$$

Fuel-Optimal Control Formulation

Objective function
Fixed time, minimum fuel trajectory

$$J = \frac{T}{c}\int_{t_0}^{t_f} \delta \, dt$$

Hamiltonian
Adjoin dynamics with costates

$$H = \frac{T}{c}\delta + \boldsymbol{\lambda}_{\boldsymbol{x}}^{T}\left[f(\boldsymbol{x}) + \frac{T}{m}\delta B\hat{\boldsymbol{u}}\right] - \lambda_m \delta \frac{T}{c}$$

Pontryagin's Minimum Principle
Necessary conditions for optimality,
control must minimize Hamiltonian

$$\hat{\boldsymbol{u}}^{*} = \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|} = -\frac{\boldsymbol{\lambda}_{\boldsymbol{v}}}{\|\boldsymbol{\lambda}_{\boldsymbol{v}}\|} \quad \text{and} \quad \dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} = -\frac{\partial H}{\partial \boldsymbol{x}} \quad \text{and} \quad \dot{\lambda}_m = -\frac{\partial H}{\partial m}$$

Switch function

$$S = \frac{c\|\lambda_v\|}{m} + \lambda_m - 1$$

Optimal engine throttle
The optimal thrust profile is a bang-bang profile

$$\delta^{*}(S) = \begin{cases} 1, & \text{if } S > 0, \\ 0 > 1, & \text{if } S = 0, \\ 0, & \text{if } S < 0, \end{cases}$$

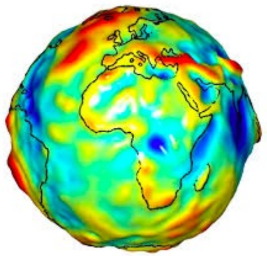
Continuation & smoothing
Hyperbolic tangent smoothing

$$\delta^{*}(S) \cong \delta^{*}(S,\rho) = \frac{1}{2}\left[1 + \tanh\left(\frac{S}{\rho}\right)\right]$$

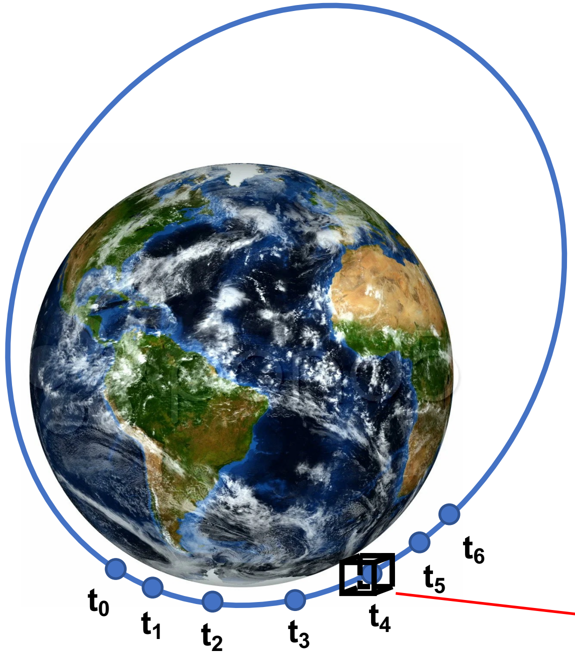
Two-point boundary value problem
Guess, propagate, update, repeat until boundary conditions are satisfied.

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{m} \\ \dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} \\ \dot{\lambda}_m \end{bmatrix} \quad \text{and} \quad \Psi = \begin{bmatrix} \boldsymbol{x}(t_f) - \boldsymbol{x}_T \\ \lambda_m(t_f) \end{bmatrix} = \boldsymbol{0}$$

Schematic: Variable Fidelity Force Model

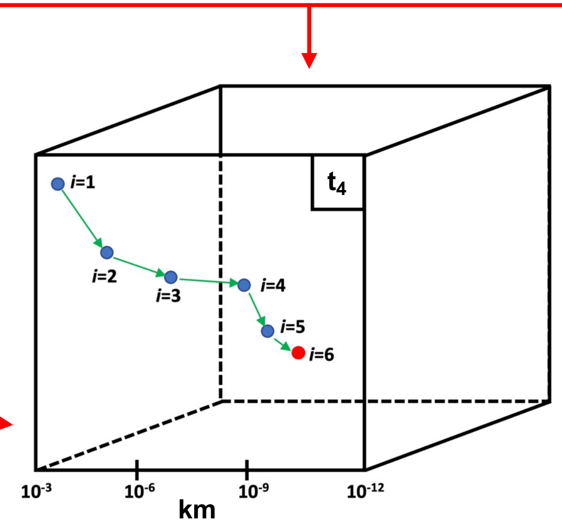


Gravitational Potential $V = \frac{\mu}{r} + \frac{\mu}{r}\sum_{n=2}^{N}\left(\frac{r_{\oplus}}{r}\right)^{n} J_n P_n(\sin\phi) + \frac{\mu}{r}\sum_{n=2}^{N}\left(\frac{r_{\oplus}}{r}\right)^{n}\sum_{m=1}^{n} P_{n,m}(\sin\phi)\big(C_{n,m}\cos(m\lambda) + S_{n,m}\sin(m\lambda)\big)$

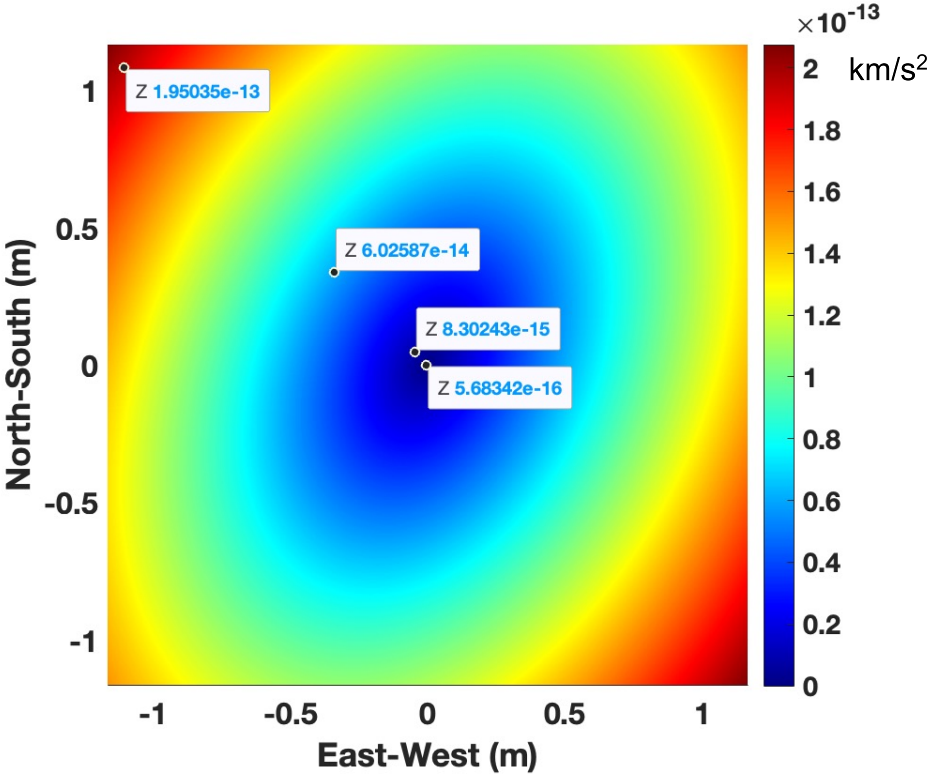


Chebyshev-Gauss-Lobatto nodes along a segment

- **Fixed point** iteration
- Small variation in **node location**
- Enables **local force approximations**
- **Reduces computation** time



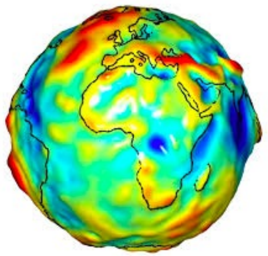
Small spatial variation



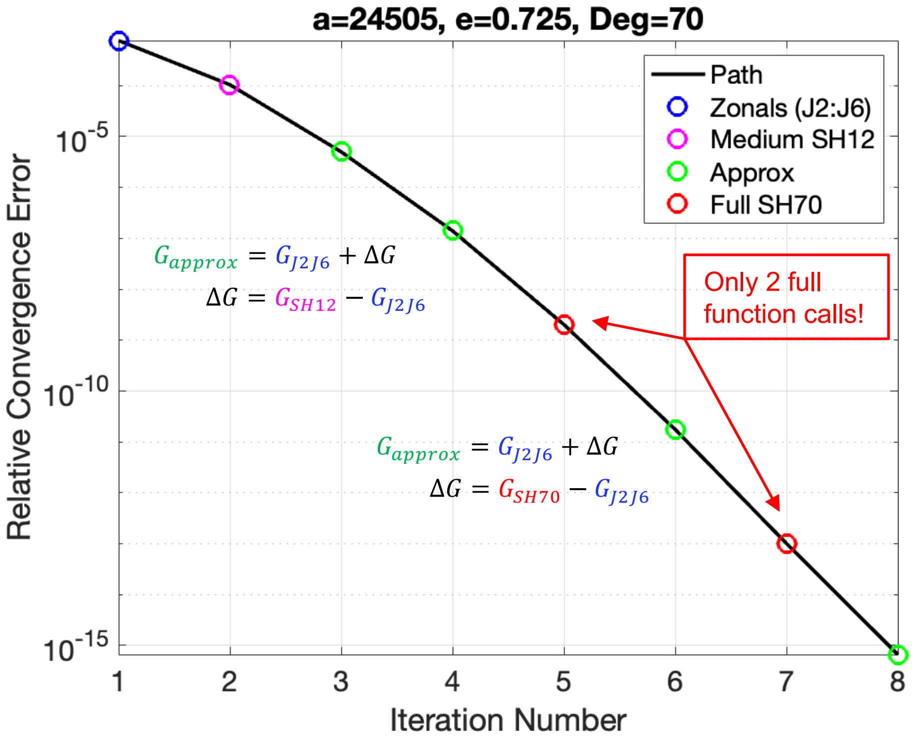
ΔG variation around a node
(ΔG = high - low)

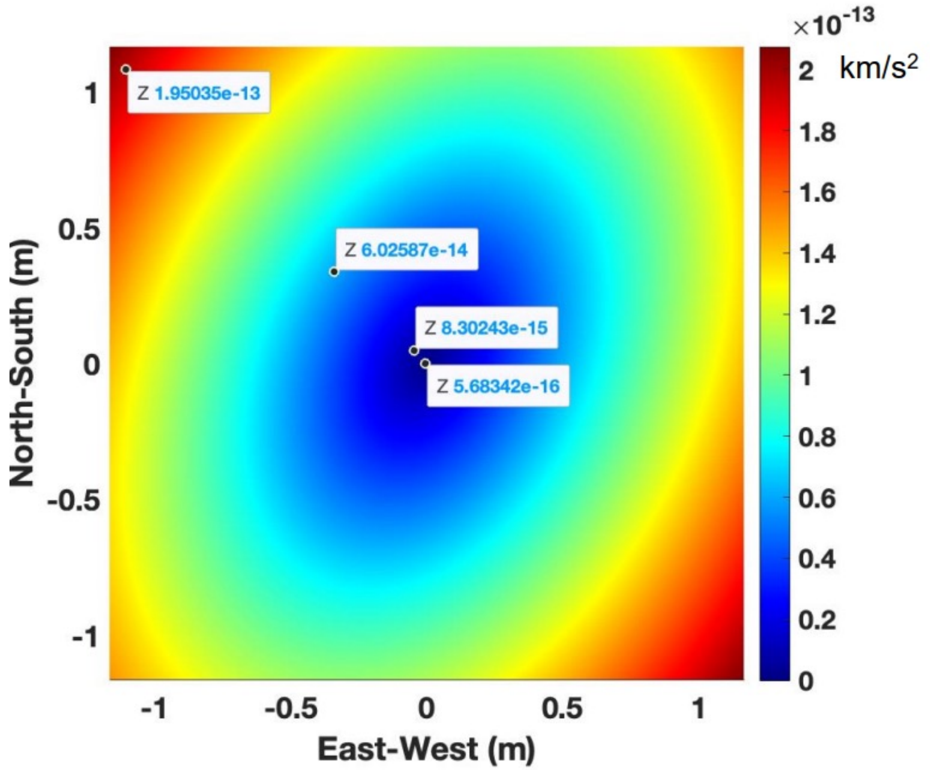
Schematic: Variable Fidelity Force Model

Gravitational Potential $$V = \frac{\mu}{r} + \frac{\mu}{r}\sum_{n=2}^{N}\left(\frac{r_{\oplus}}{r}\right)^{n} J_n P_n(\sin\phi) + \frac{\mu}{r}\sum_{n=2}^{N}\left(\frac{r_{\oplus}}{r}\right)^{n}\sum_{m=1}^{n} P_{n,m}(\sin\phi)\left(C_{n,m}\cos(m\lambda) + S_{n,m}\sin(m\lambda)\right)$$



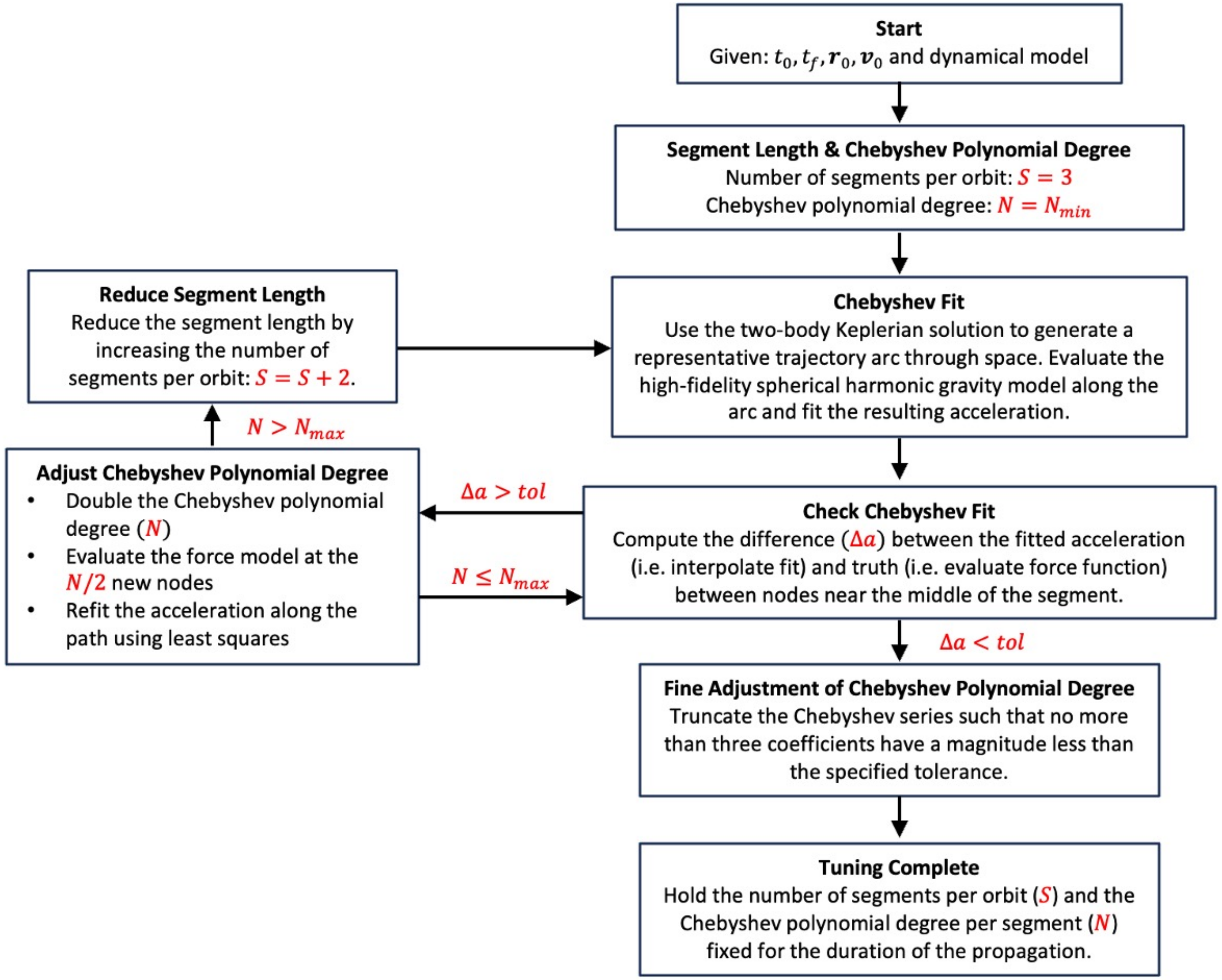
Variable Fidelity Gravity Model

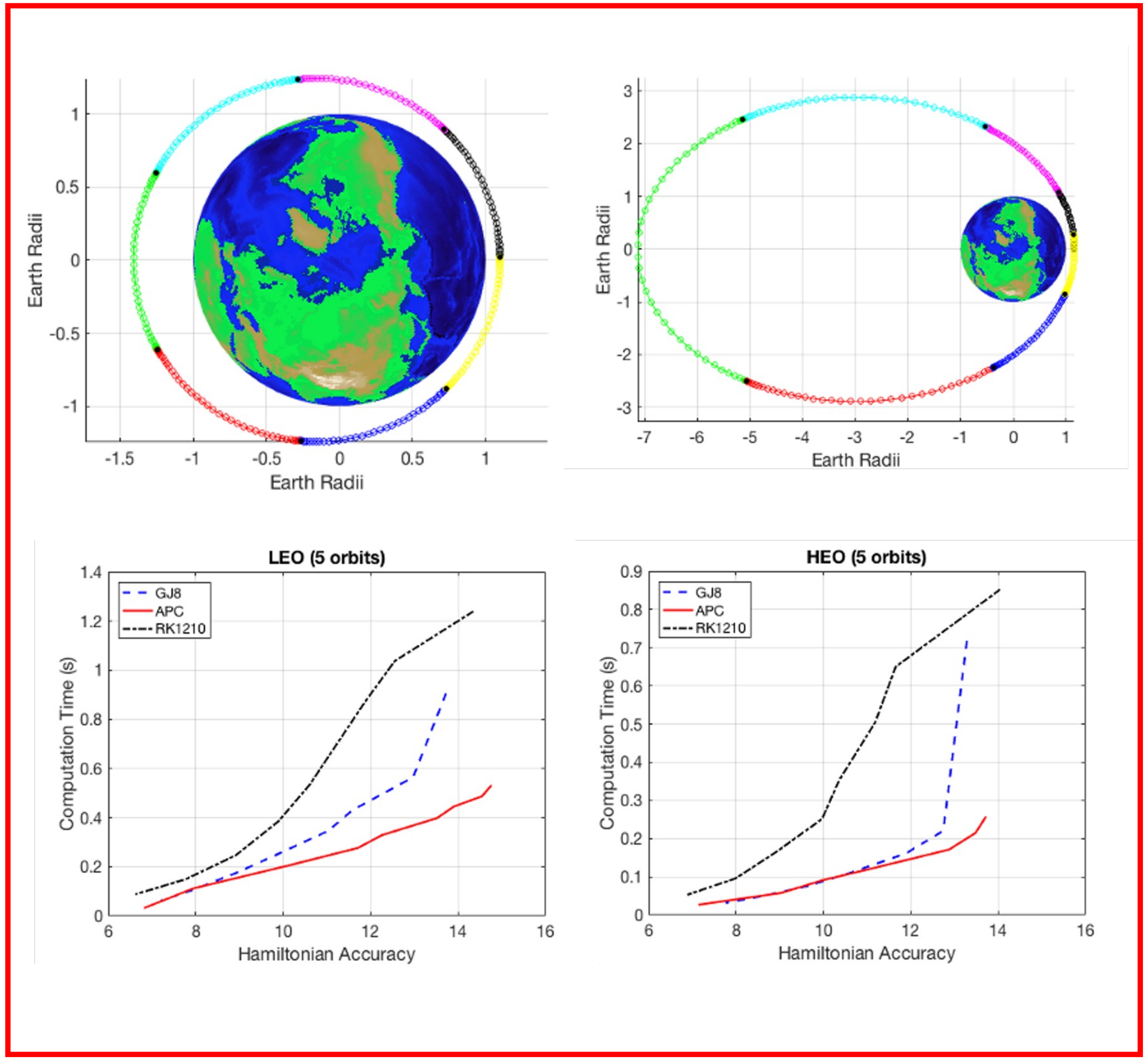




$\Delta G = G_{SH70} - G_{J2J6}$

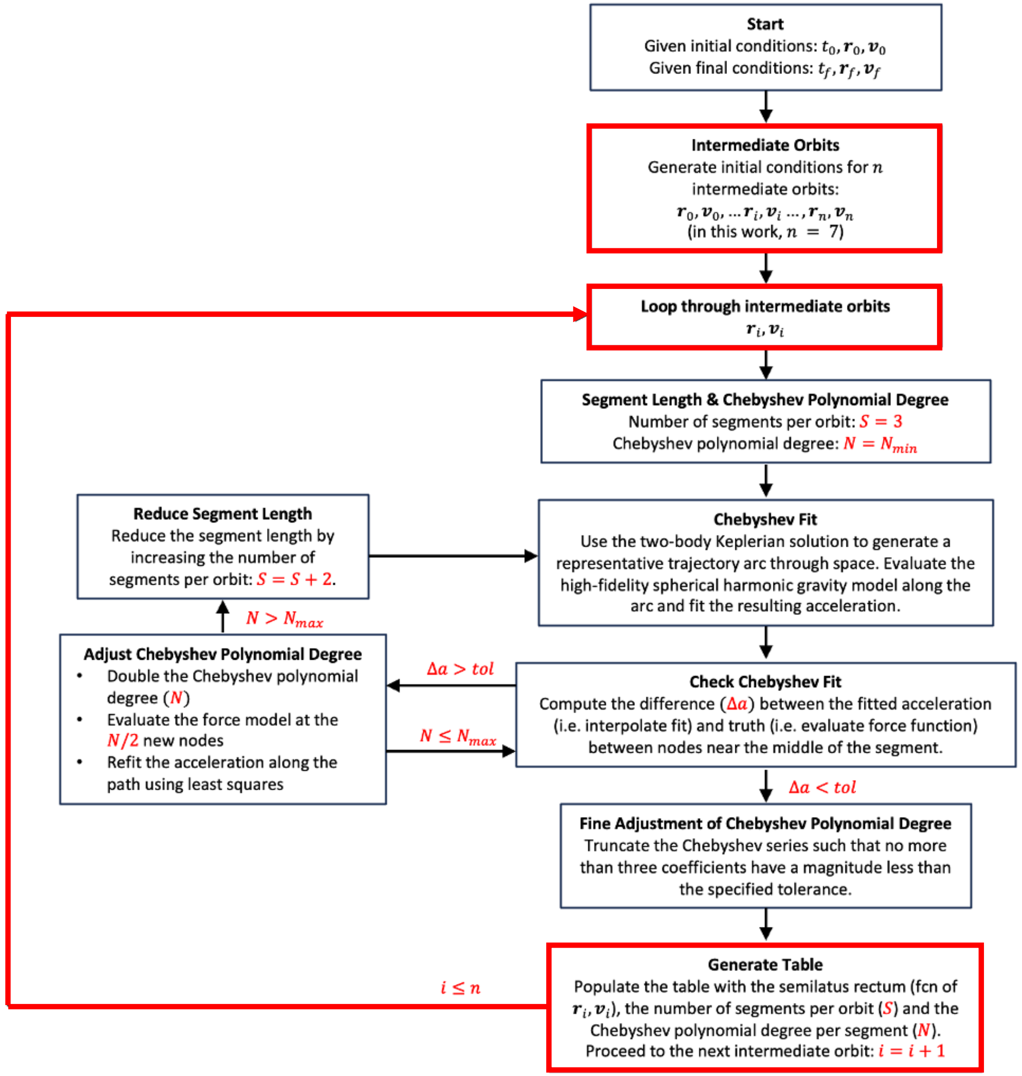
Segmentation Scheme: ***Ballistic*** Closed-Orbit Propagation



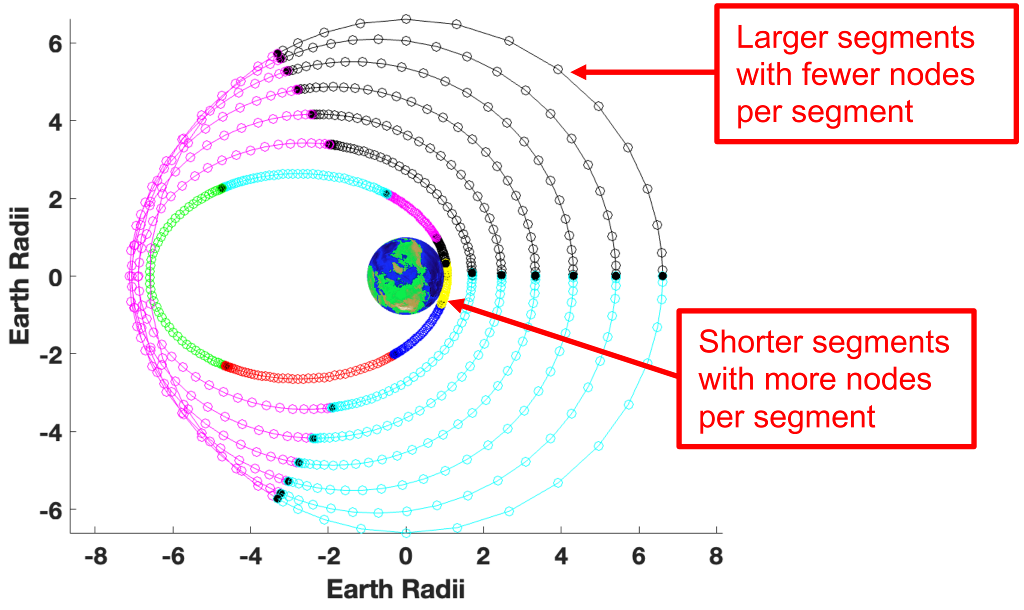


Segmentation scheme developed by Woollands & Junkins, “Nonlinear Differential Equations Solvers via Adaptive Picard-Chebyshev Iteration: Applications in Astrodynamics”, JGCD, 2019.

Segmentation Scheme: ***Bang-Bang*** Closed-Orbit Propagation



Intermediate Orbits



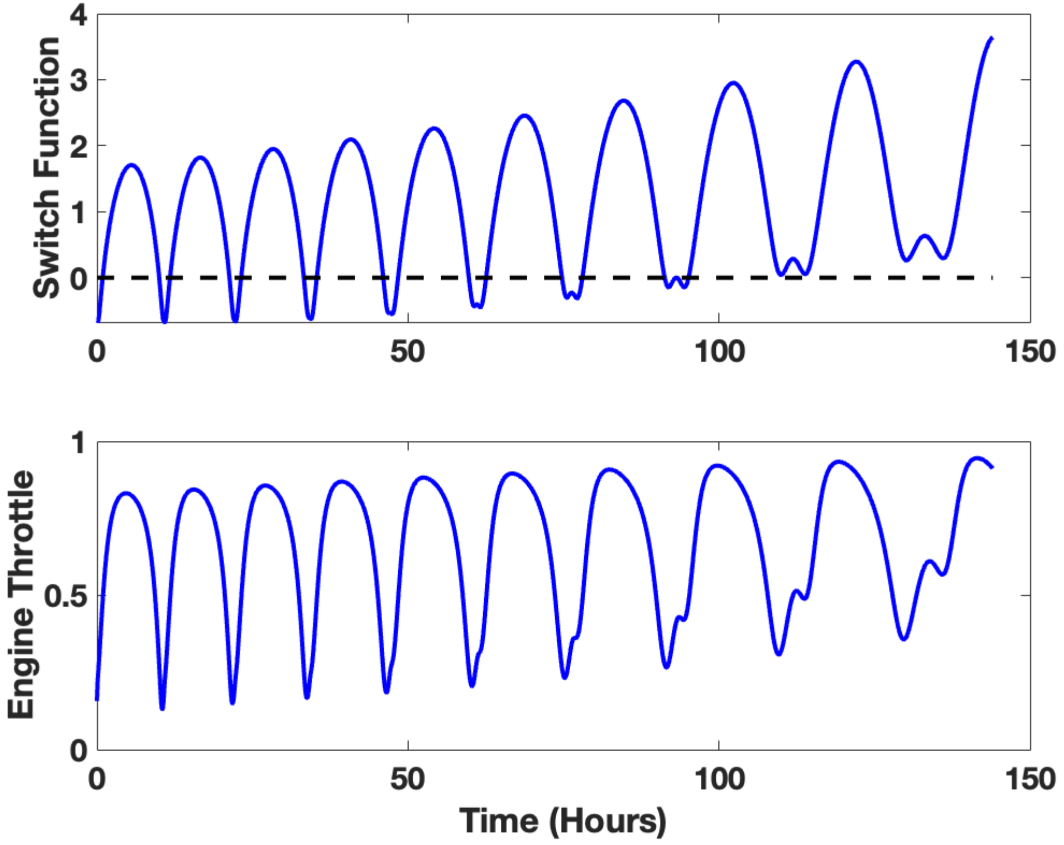
Look-up Table

Semilatus Rectum ($R_\oplus$)	Segments per orbit	Polynomial degree (N)
1.823	7	52
2.732	3	42
3.651	3	33
4.540	3	30
5.356	3	26
6.060	3	22
6.611	3	15

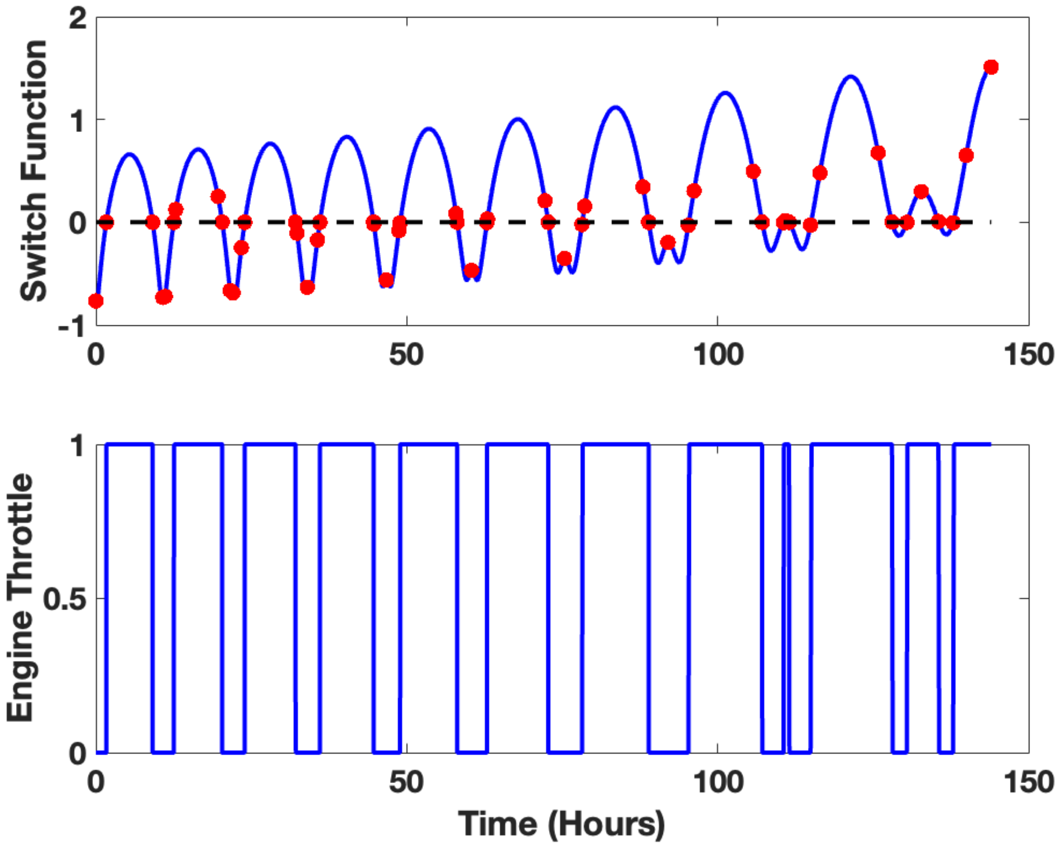
Note: 50 intermediate orbits were used for the results presented in these slides.

Anticipate Throttle Switches

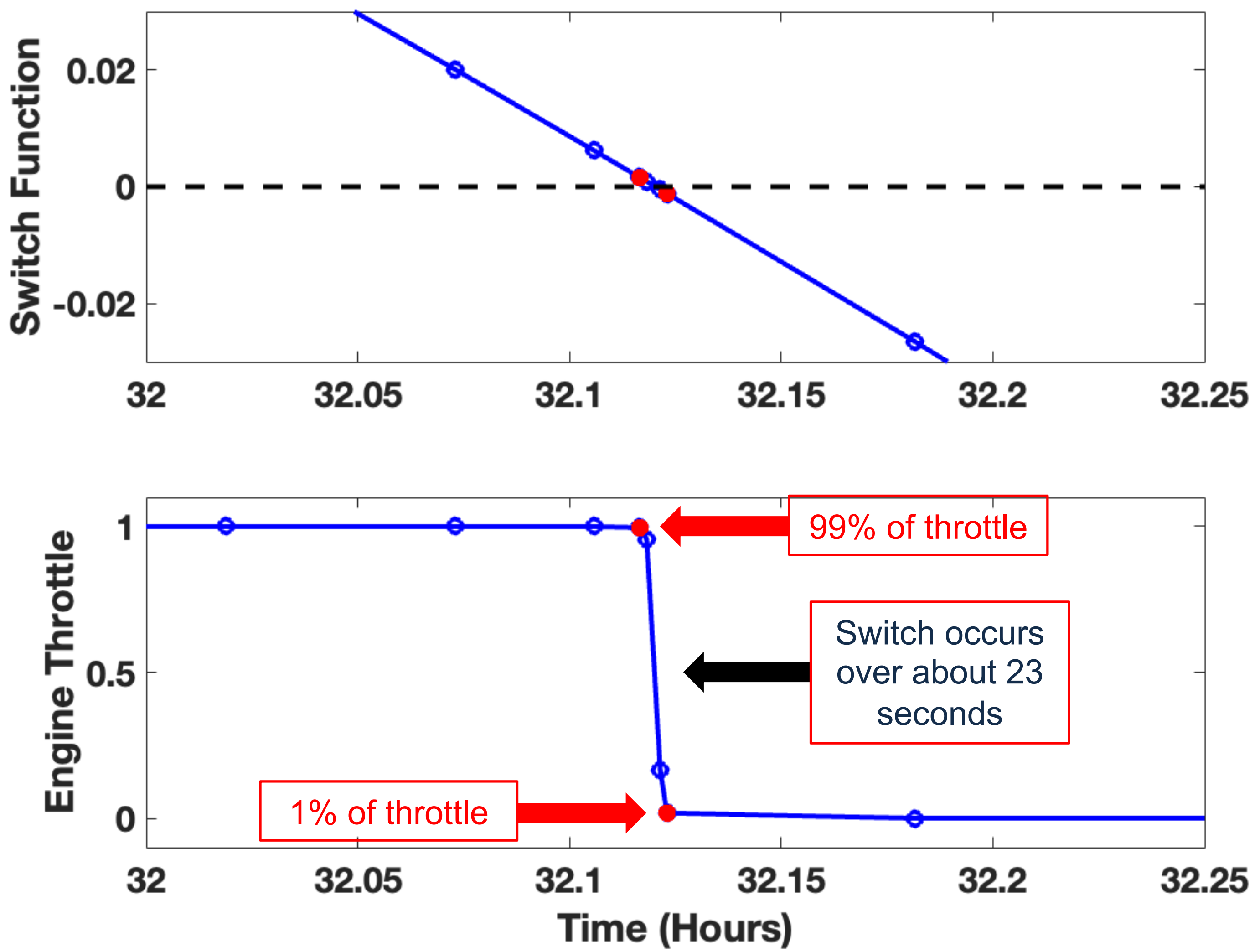
Smoothed profile - earlier iteration



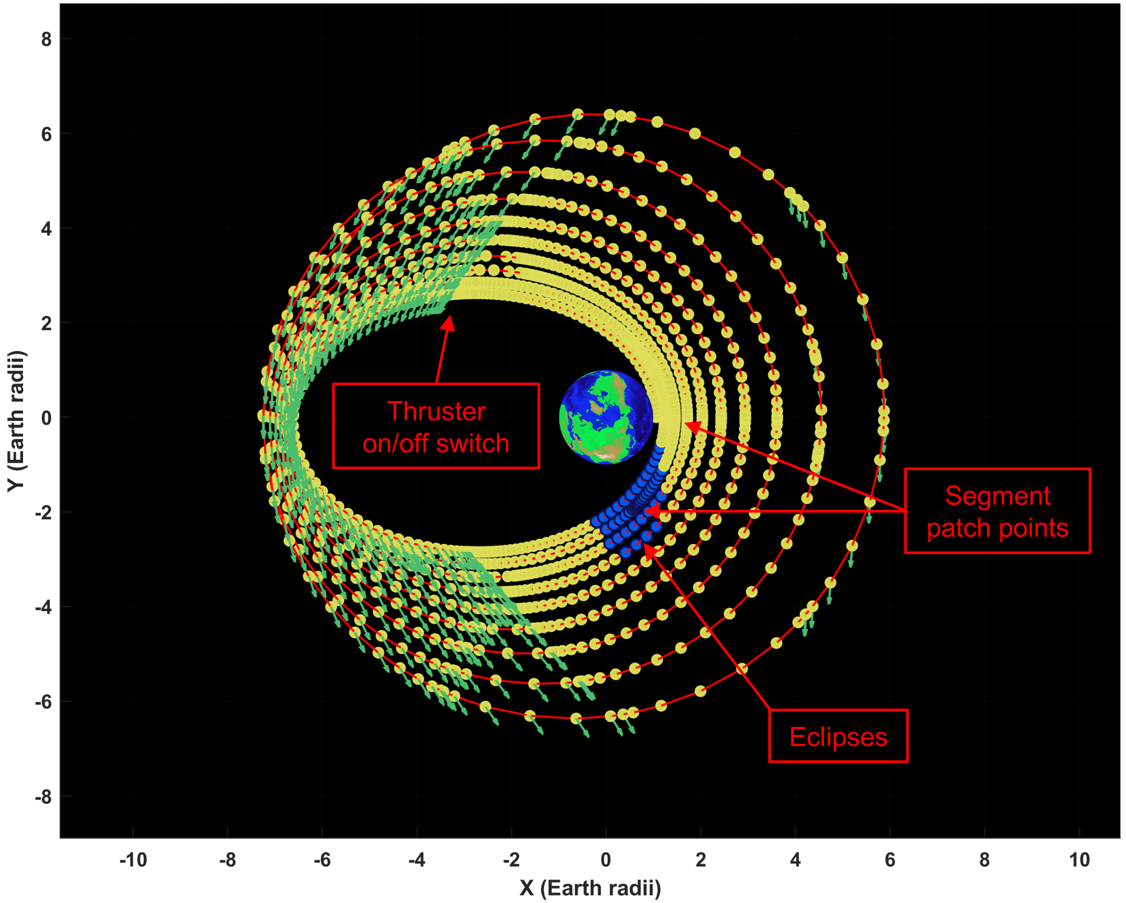
Bang-Bang profile - converged solution

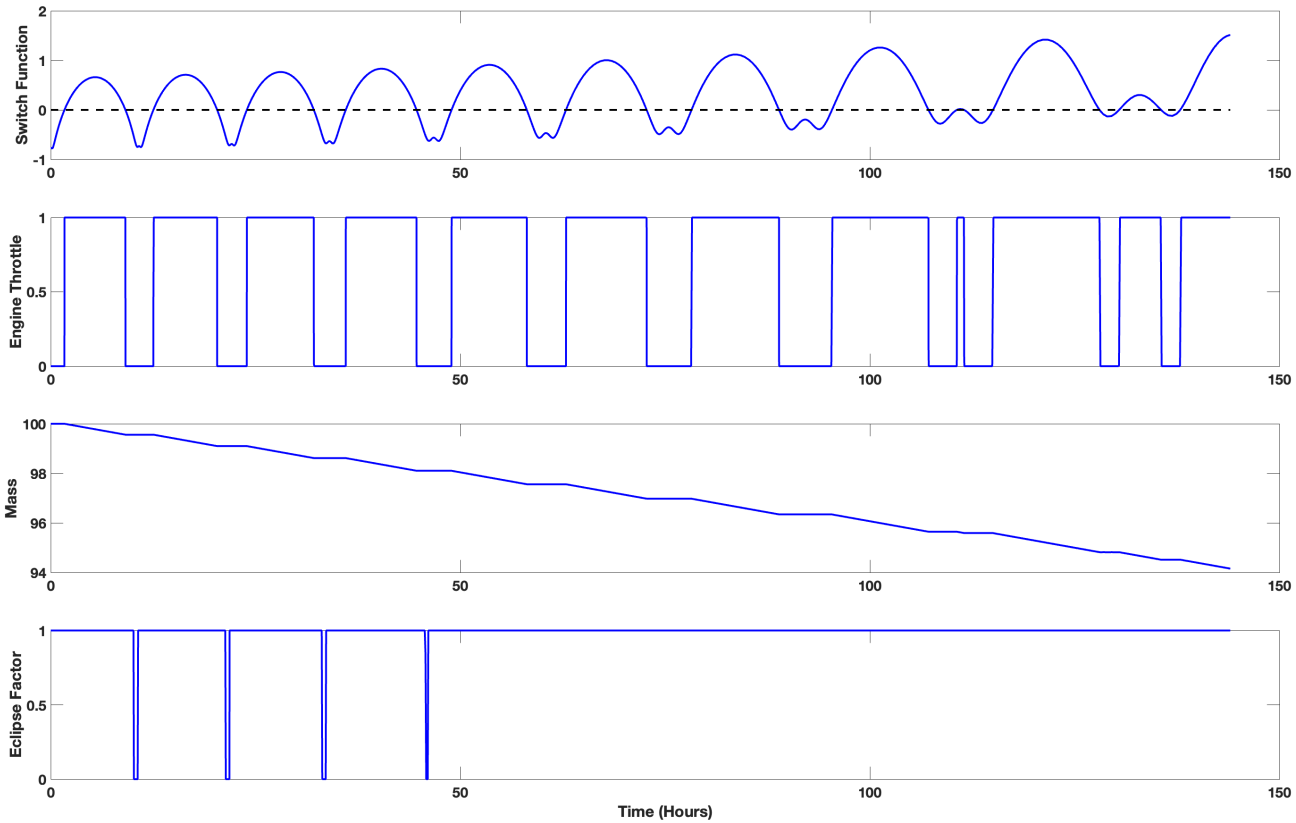


Approximate Bang-Bang Dynamics



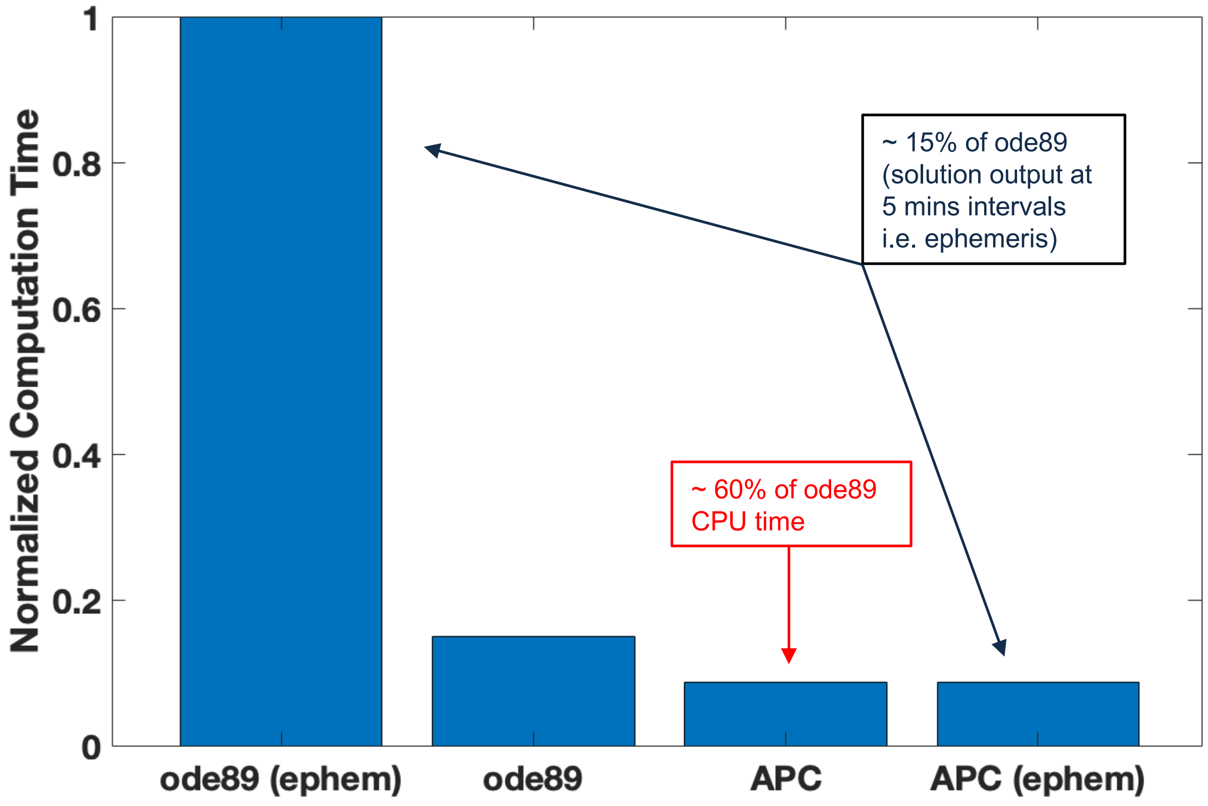
Simulation Results

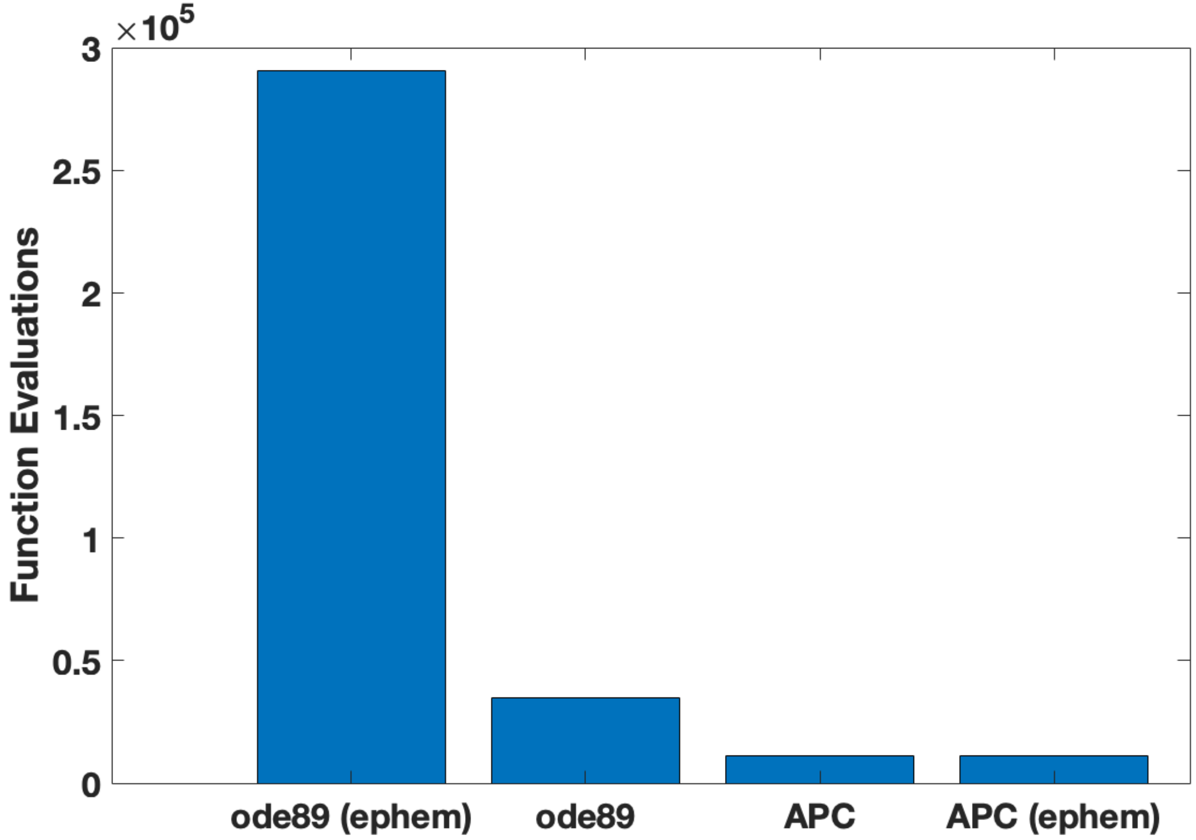




Fuel-optimal low-thrust transfer from GTO to GEO

Run-time Comparison





Why generate an ephemeris? Provides a sequence of thrust commands for mission operations.

Himmat Panag
PhD student, UIUC

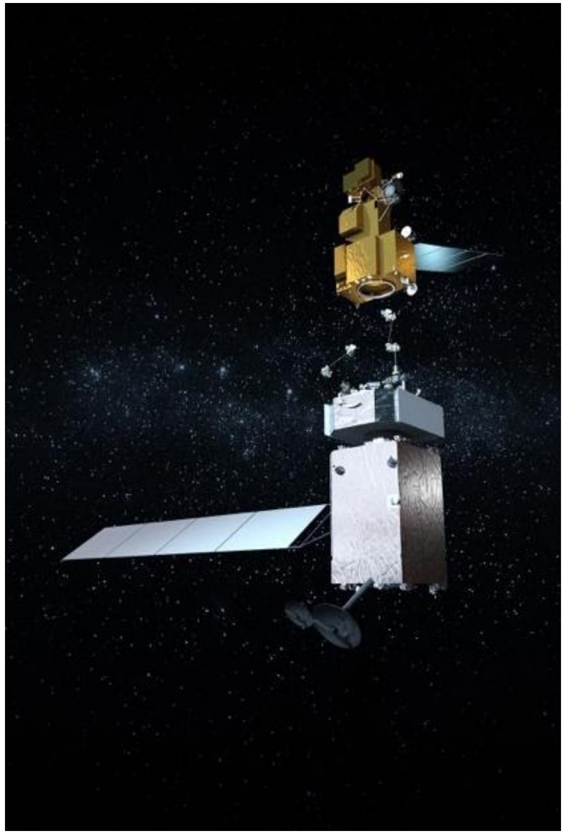
PART II

Thruster Pointing Constrained Optimal Control for Satellite Servicing using Indirect Optimization

Himmat Panag, Robyn M. Woollands, “Thruster Pointing Constrained Optimal Control for Satellite Servicing using Indirect Optimization”, AIAA SciTech Forum, Jan 7-12, 2024.

PART II: Motivation

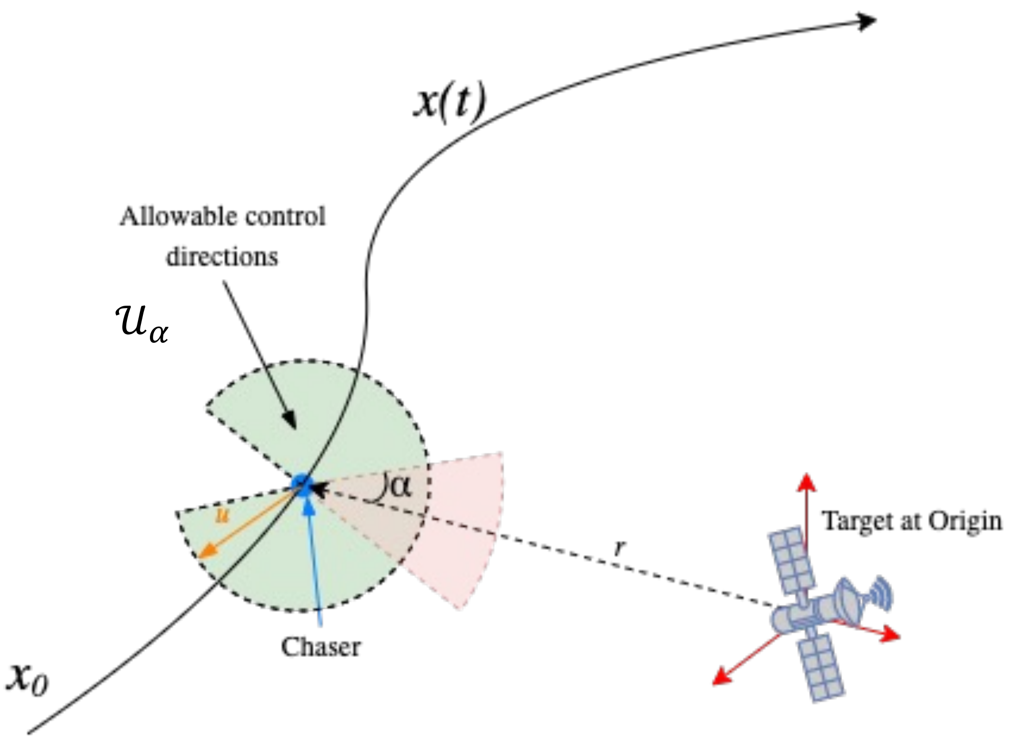
- In-space servicing
  - Rendezvous, proximity operations & docking
  - Exhaust from thrusters may contaminate delicate sensors on the target spacecraft
- Thruster pointing constraints limit on the angular range over which thrusters may operate
  - Thruster pointing constraints have received recent attention using convex optimization [Zhang, Zhu, Cheng & Li (2022); Liu & Lu (2014)].
  - Non-convex spherical pointing constraints were solved using the method of successive approximations.
- To our knowledge, problems involving thruster pointing constraints have not been solved using indirect optimization methods
  - The dependence of the control set on the chaser's state presents a challenge
- **Our contribution:**
  - Formulate the problem such that the constraint is implicitly encoded into the dynamics using trigonometric functions



Artist's concept OSAM-1

Figure Credit: NASA: https://nexis.gsfc.nasa.gov/osam-1.html.

2D Thruster Pointing Constraint Geometry



Unconstrained Problem

Optimal thruster pointing direction

$$\hat{\boldsymbol{u}}^* = \hat{\boldsymbol{u}}^*_{free} = \frac{\boldsymbol{u}}{||\boldsymbol{u}||} = -\frac{\boldsymbol{\lambda_v}}{||\boldsymbol{\lambda_v}||}$$

Constrained Problem

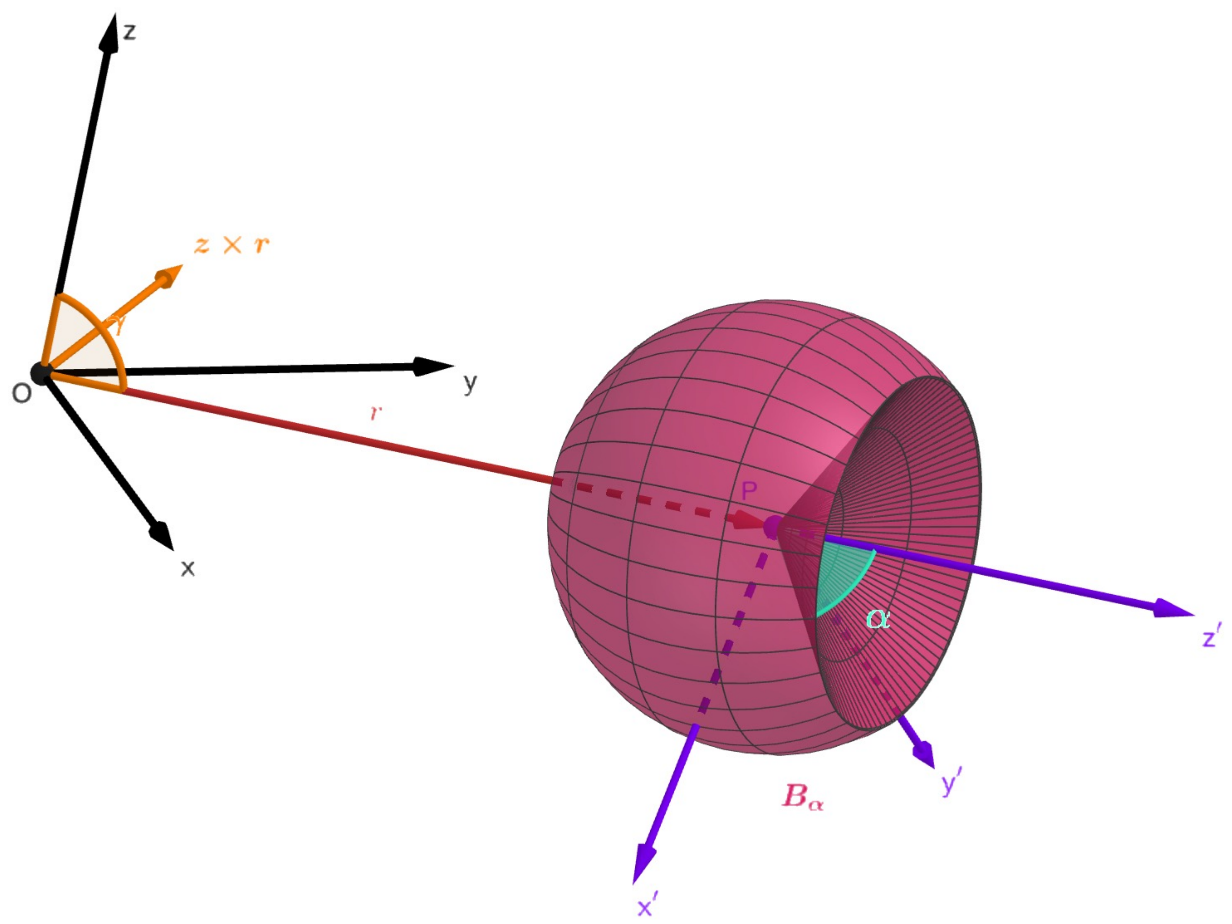
Plume is ***not*** permitted to point within the red region.

Control direction (opposite plume) is restricted to the green region.

$\mathcal{U}_\alpha$ is the set of allowable thrust pointing directions.

If $\hat{\boldsymbol{u}}^*_{free} \notin \mathcal{U}_\alpha(\boldsymbol{r})$, the vector in $\mathcal{U}_\alpha(\boldsymbol{r})$ that minimizes the Hamiltonian must be found.

3D Thruster Pointing Constraint Geometry



Black frame:
- LVLH frame
- Collocated with the target spacecraft
- Relative dynamics are derived w.r.t. this frame.

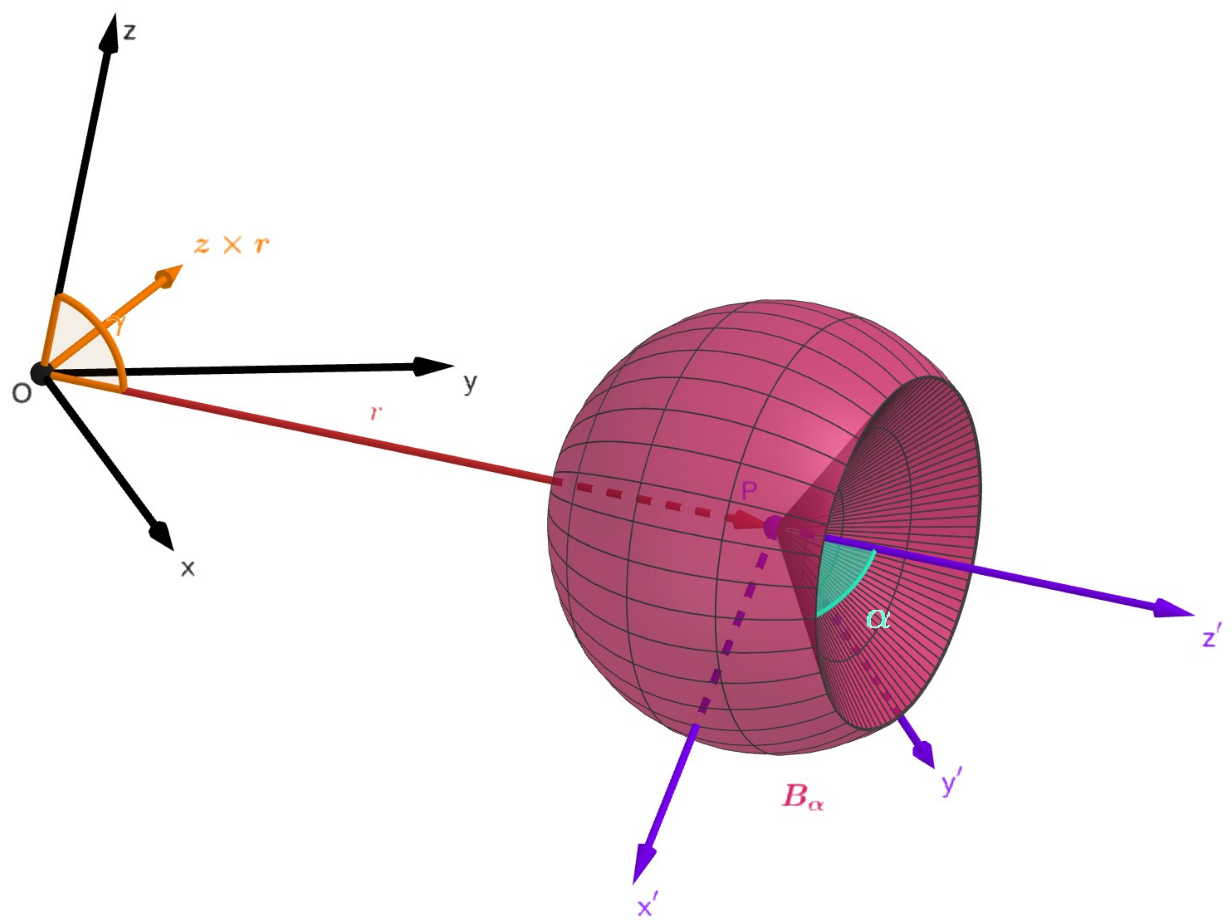
Purple frame:
- Collocated with the chaser
- Obtained by rotating the black frame through the angle $\gamma = \cos^{-1}\left(\frac{r_3}{|\boldsymbol{r}|}\right) \in [0, \pi]$ about $\boldsymbol{z}\times\boldsymbol{r}$.

Feasible control set
- Unit ball, B_α, with cone of semi-vertical angle, α, removed

Reparameterize Control
- Control direction, $\hat{\boldsymbol{u}}'$, in $x' - y' - z'$ frame can be and defined in spherical coordinates (ϕ, θ) as follows:

$$(\phi, \theta) \in B_\alpha \triangleq \left[-\frac{\pi}{2}, \frac{\pi}{2} - \alpha\right] \times [-\pi, \pi] \forall \boldsymbol{r}$$

3D Thruster Pointing Constraint Geometry



Rewrite Equations

- Use angles ϕ and θ as follows:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + \frac{T}{m}\delta B\hat{\boldsymbol{u}}$$

$$= f(\boldsymbol{x}) + \frac{T}{m}\delta B\psi(\boldsymbol{r})\hat{\boldsymbol{u}}'$$

$$= f(\boldsymbol{x}) + \frac{T}{m}\delta B\psi(\boldsymbol{r})\begin{bmatrix}\cos\phi\cos\theta \\ \cos\phi\sin\theta \\ \sin\phi\end{bmatrix}$$

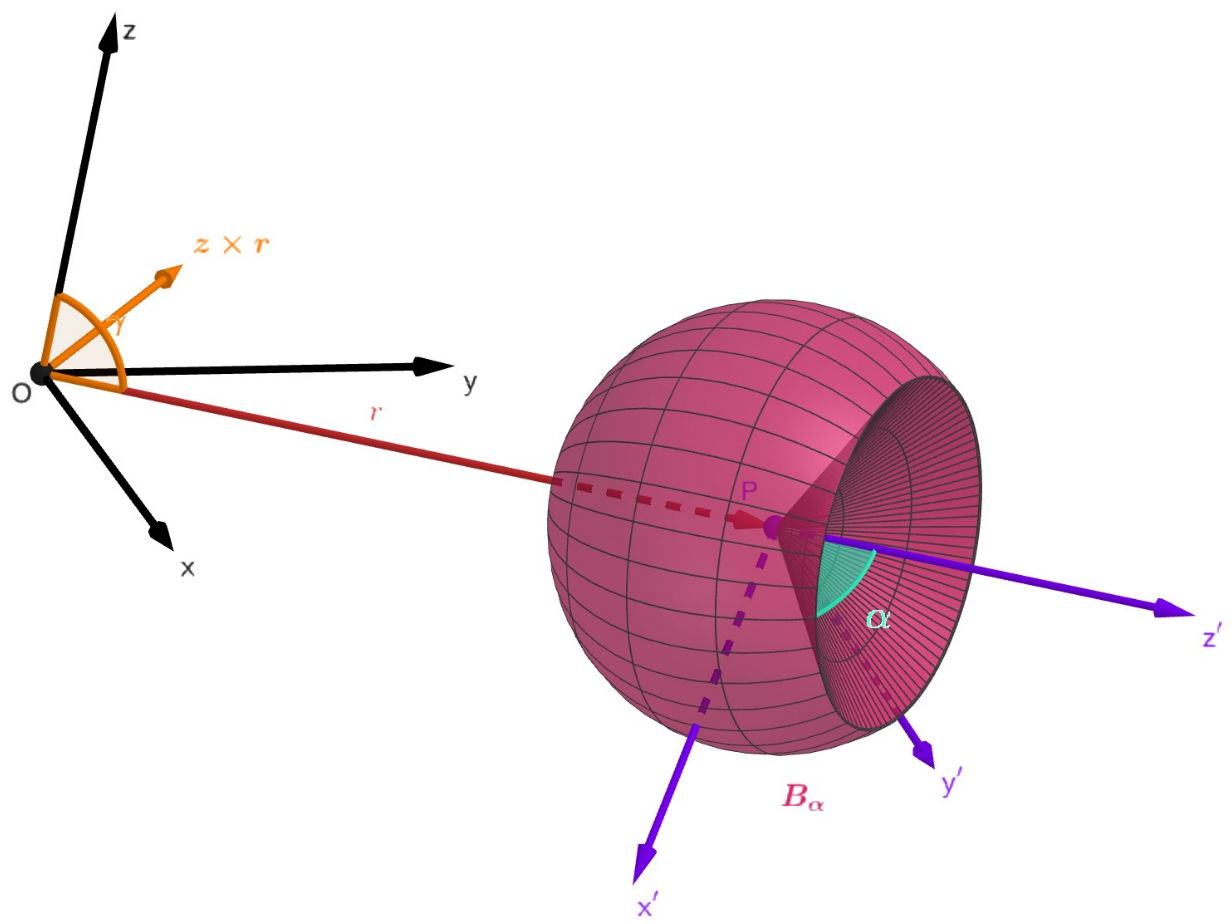
Transformation Matrix

- The matrix $\psi(\boldsymbol{r})$ transforms $\hat{\boldsymbol{u}} \in B_\alpha$ from $x' - y' - z'$ to $x - y - z$.

Hamiltonian

$$H = \frac{T}{c}\delta + \boldsymbol{\lambda}_{\boldsymbol{x}}^T\left[f(\boldsymbol{x}) + \frac{T}{m}\delta B\psi(\boldsymbol{r})\begin{bmatrix}\cos\phi\cos\theta \\ \cos\phi\sin\theta \\ \sin\phi\end{bmatrix}\right] - \lambda_m\delta\frac{T}{c}$$

3D Thruster Pointing Constraint Geometry



Hamiltonian

$$H = \frac{T}{c}\delta + \boldsymbol{\lambda}_{\boldsymbol{x}}^T\left[f(\boldsymbol{x}) + \frac{T}{m}\delta B\psi(\boldsymbol{r})\begin{bmatrix}\cos\phi\cos\theta\\ \cos\phi\sin\theta\\ \sin\phi\end{bmatrix}\right] - \lambda_m\delta\frac{T}{c}$$

Switch Function

$$S = -\frac{c}{m}\boldsymbol{\lambda}_v^T\psi(\boldsymbol{r})\begin{bmatrix}\cos\phi^*\cos\theta^*\\ \cos\phi^*\sin\theta^*\\ \sin\phi^*\end{bmatrix} + \lambda_m - 1$$

Costate Equations

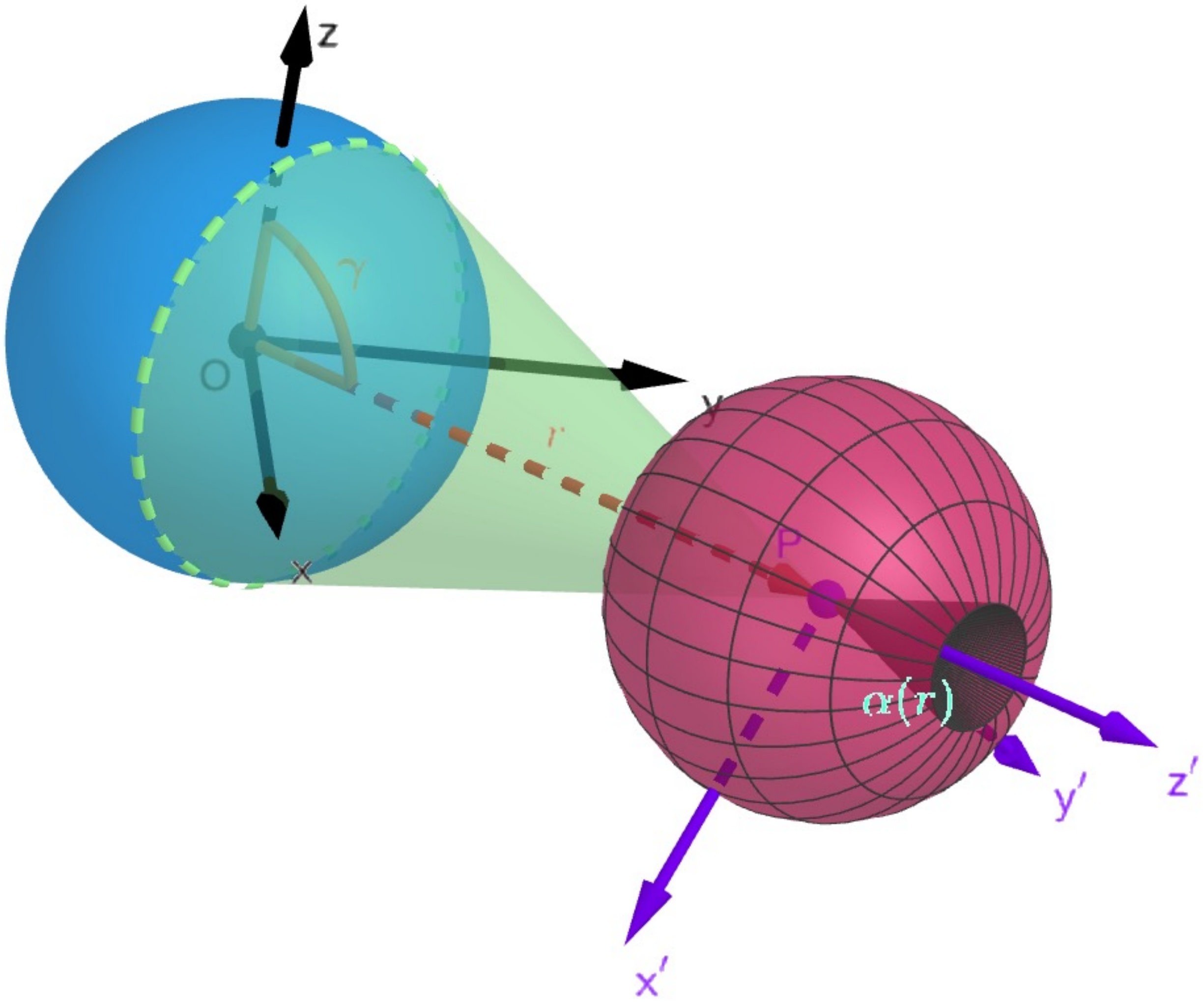
$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} = -\left.\frac{\partial H}{\partial \boldsymbol{x}}\right|_{\phi^*,\theta^*,\delta^*} \quad \text{and} \quad \dot{\lambda}_m = -\left.\frac{\partial H}{\partial m}\right|_{\phi^*,\theta^*,\delta^*}$$

Note: Derivation shown in the paper

Control

$$\frac{\partial H}{\partial \theta} = 0 \quad \text{and} \quad \frac{\partial H}{\partial \phi} = 0$$

Variation of Constraint Angle with Distance



Spherical Constraint

- Reasonable to assume that the thruster pointing constraint is ***less restrictive*** when the chaser is ***further*** from the target (e.g. spherical region around target).
- Target has a **radius**, R, and the ***constraint angle*** is defined as follows:

$$\alpha(\boldsymbol{r}) = \sin^{-1}\left(\frac{R}{|\boldsymbol{r}|}\right)$$

- Problem is not defined if $|\boldsymbol{r}| \leq R$ (i.e. chaser collides with target)

Reparameterize

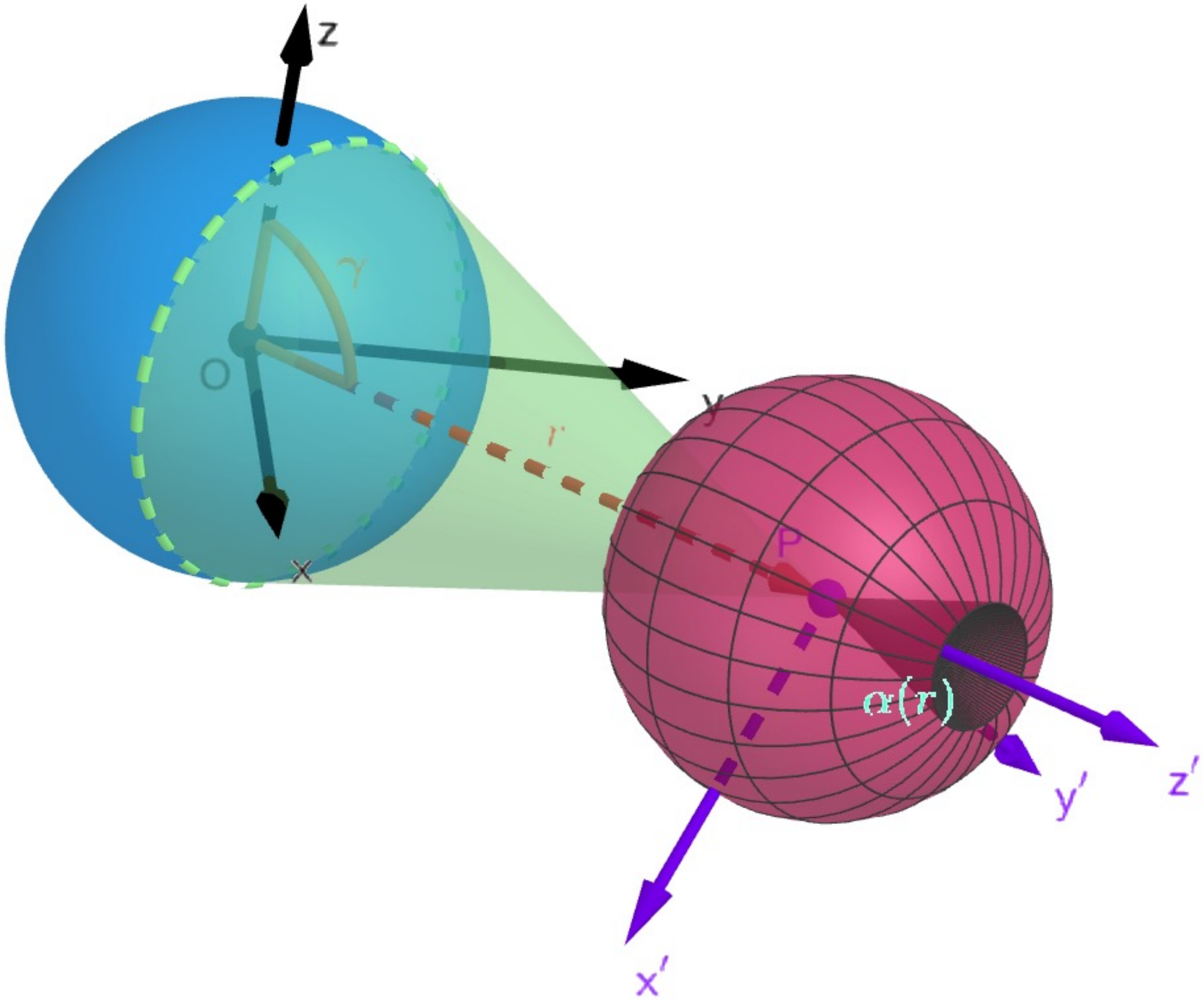
- Reparameterize the control set B_α so that it does not explicitly depend on $\boldsymbol{r}$. Introduce $\sigma \in [0,1]$ which translates ϕ linearly as follows:

$$\phi(\sigma, \boldsymbol{r}) = -\frac{\pi}{2} + (\pi - \alpha(\boldsymbol{r}))\sigma$$

$$\phi(0, \boldsymbol{r}) = -\frac{\pi}{2} \quad \text{and} \quad \phi(1, \boldsymbol{r}) = \frac{\pi}{2} - \alpha(\boldsymbol{r})$$

$\theta \in [-\pi, \pi]$ is unchanged

Variation of Constraint Angle with Distance



Hamiltonian

$$H = \frac{T}{c}\delta + \boldsymbol{\lambda}_{\boldsymbol{x}}^T \left[f(\boldsymbol{x}) + \frac{T}{m}\delta B \psi(\boldsymbol{r}) \begin{bmatrix} \cos(\phi(\sigma, \boldsymbol{r}))\cos(\theta) \\ \cos(\phi(\sigma, \boldsymbol{r}))\sin(\theta) \\ \sin(\phi(\sigma, \boldsymbol{r})) \end{bmatrix} \right] - \lambda_m \delta \frac{T}{c}$$

Switch Function

$$S = -\frac{c}{m}\boldsymbol{\lambda}_{\boldsymbol{v}}^T \psi(\boldsymbol{r}) \begin{bmatrix} \cos(\phi(\sigma^*, \boldsymbol{r}))\cos(\theta^*) \\ \cos(\phi(\sigma^*, \boldsymbol{r}))\sin(\theta^*) \\ \sin(\phi(\sigma^*, \boldsymbol{r})) \end{bmatrix} + \lambda_m - 1$$

Costate Equations

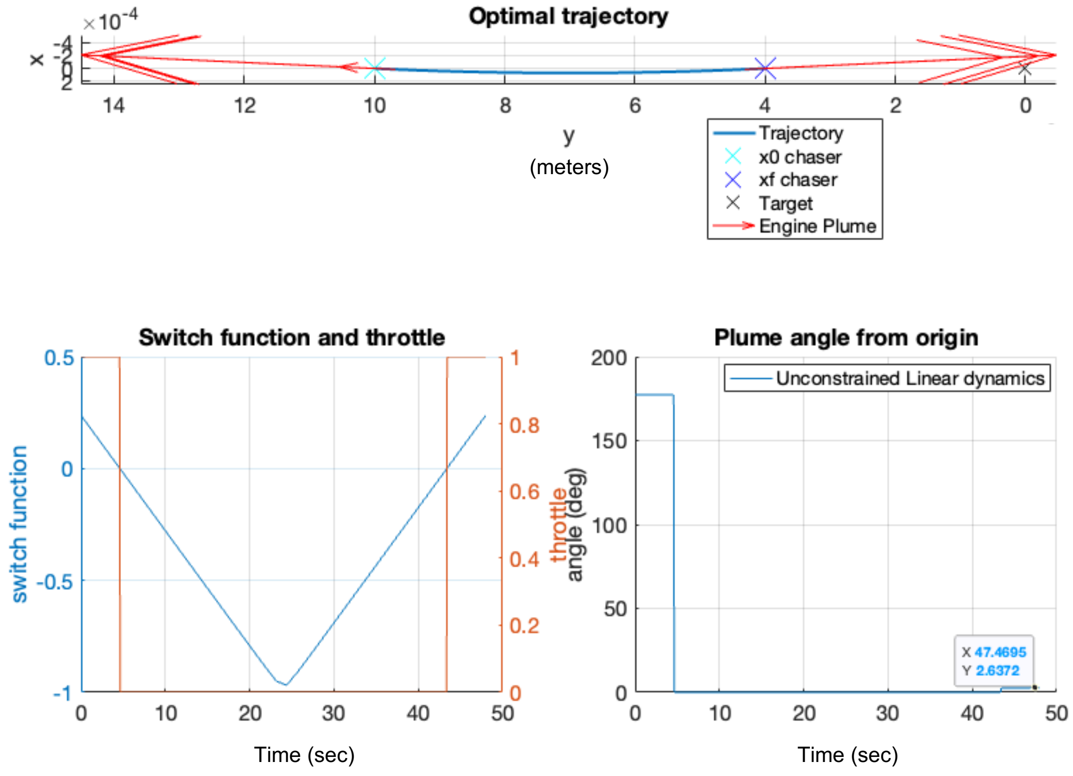
$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{x}} = -\left.\frac{\partial H}{\partial \boldsymbol{x}}\right|_{\sigma^*, \theta^*, \delta^*} \quad \text{and} \quad \dot{\lambda}_m = -\left.\frac{\partial H}{\partial m}\right|_{\sigma^*, \theta^*, \delta^*}$$

Note: Derivation shown in the paper

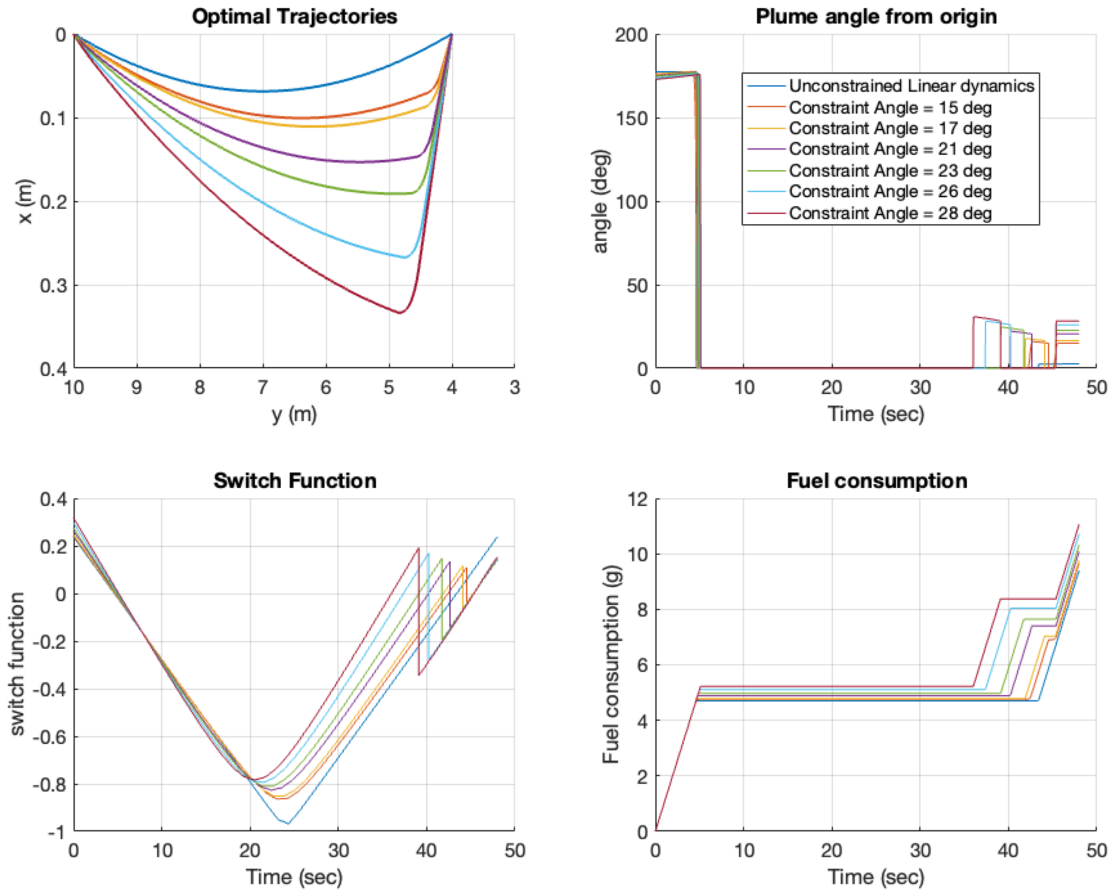
Control

$$\frac{\partial H}{\partial \theta} = 0 \quad \text{and} \quad \frac{\partial H}{\partial \sigma} = 0$$

Simulation Results – Point Target

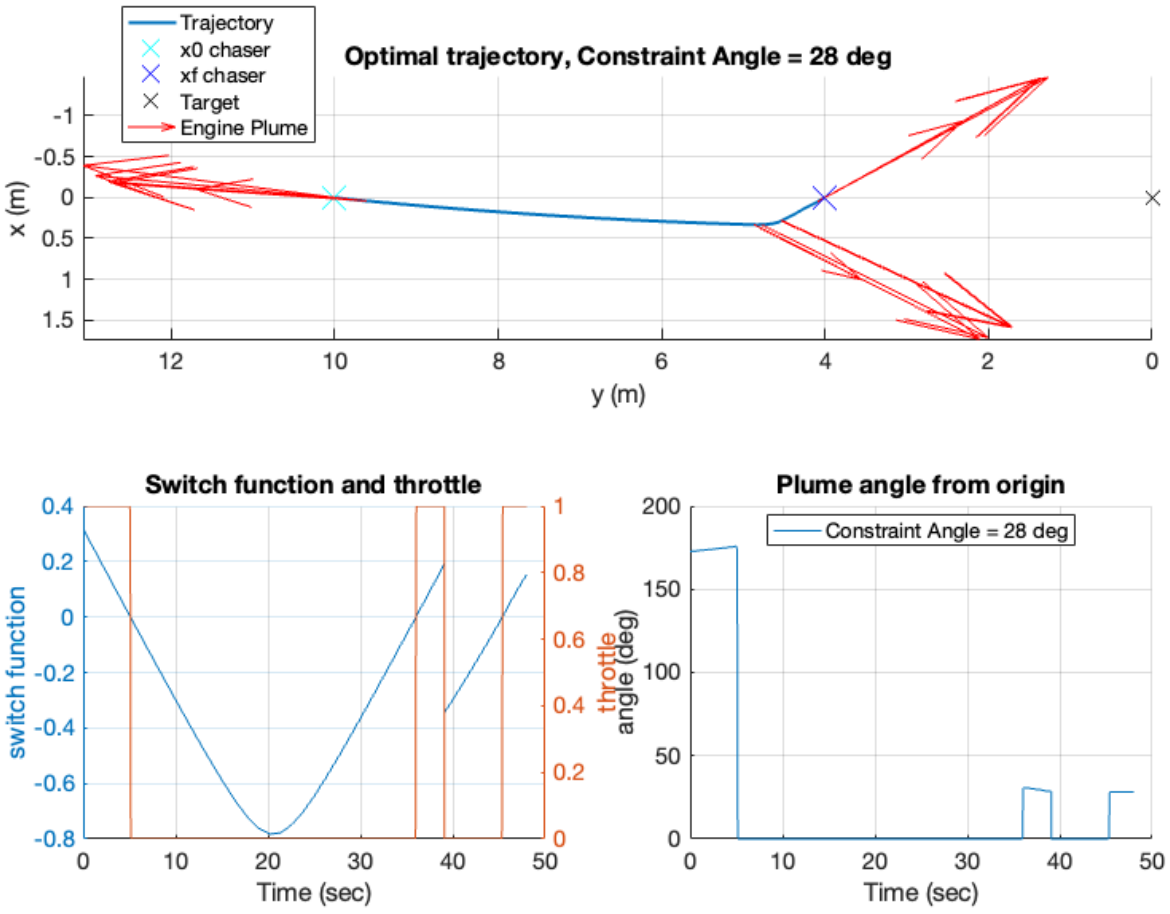


Unconstrained transfer

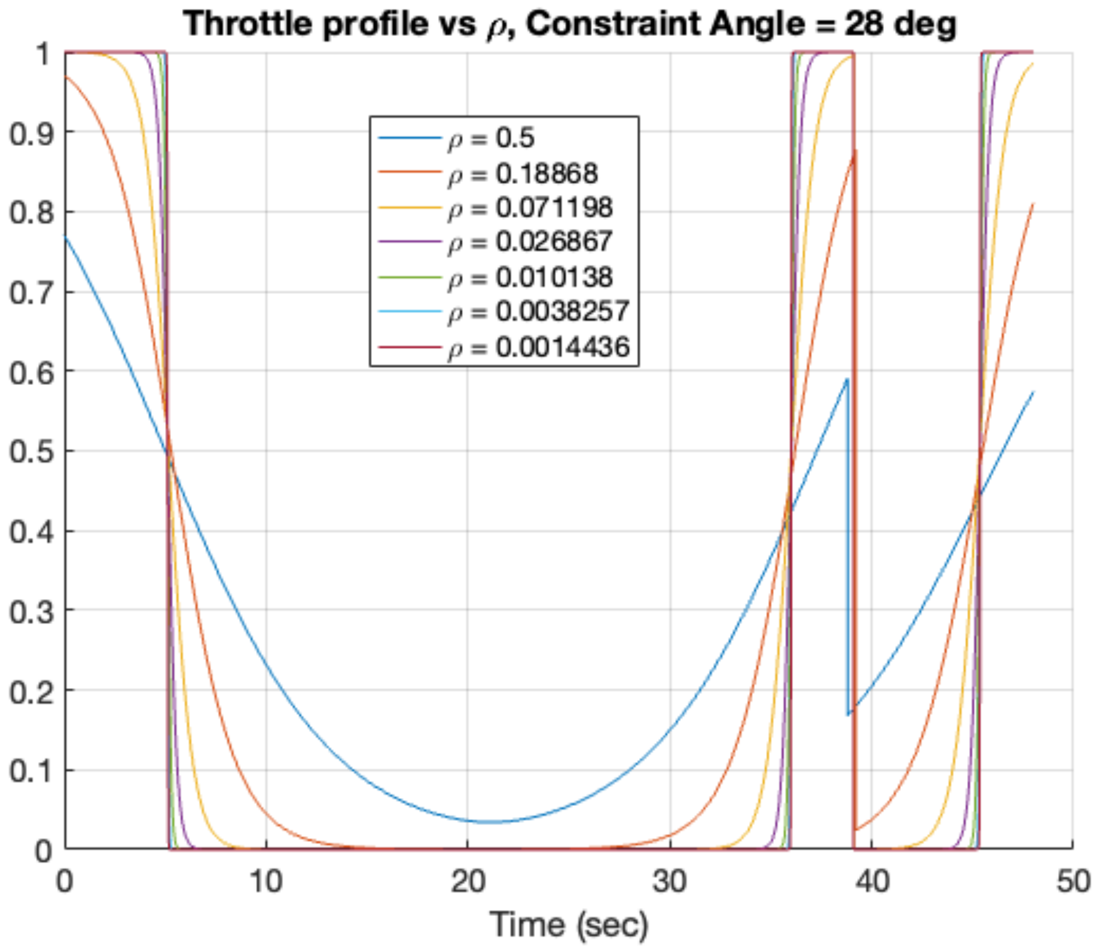


Optimal transfer for increasing thruster pointing constraint angle (α)

Simulation Results – Point Target

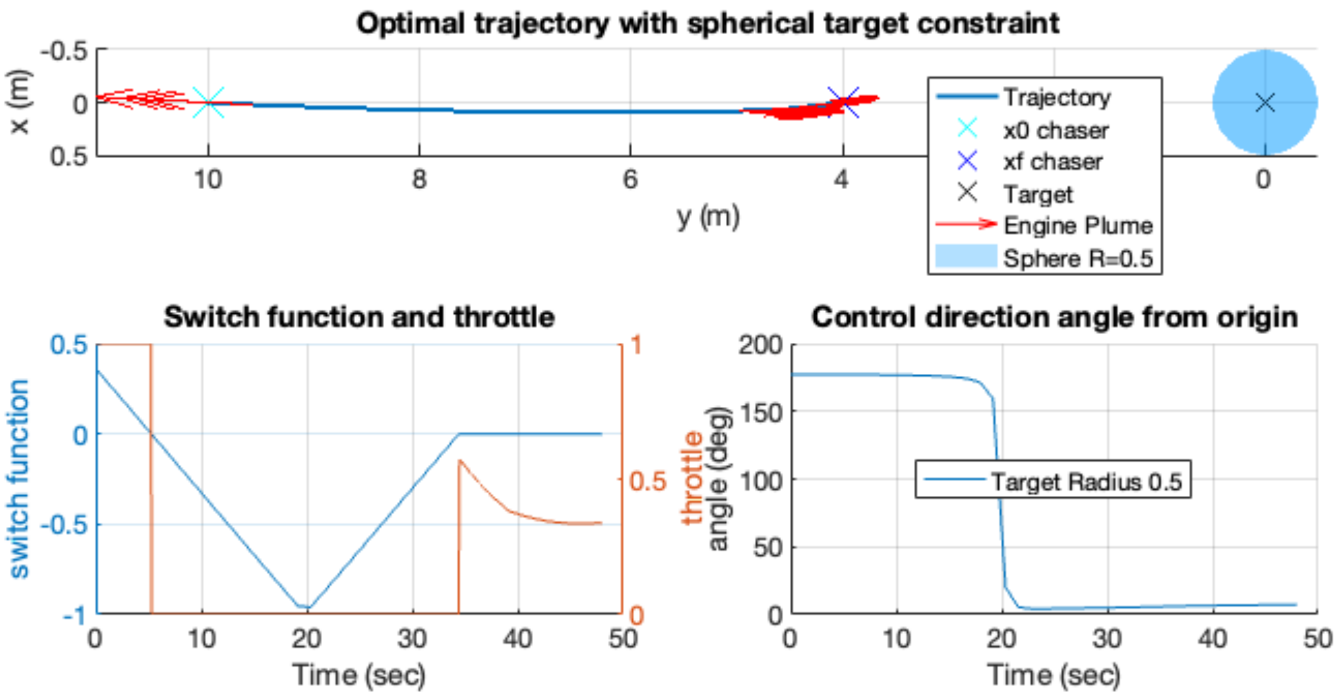


Solution with $\alpha = 28°$ pointing constraint

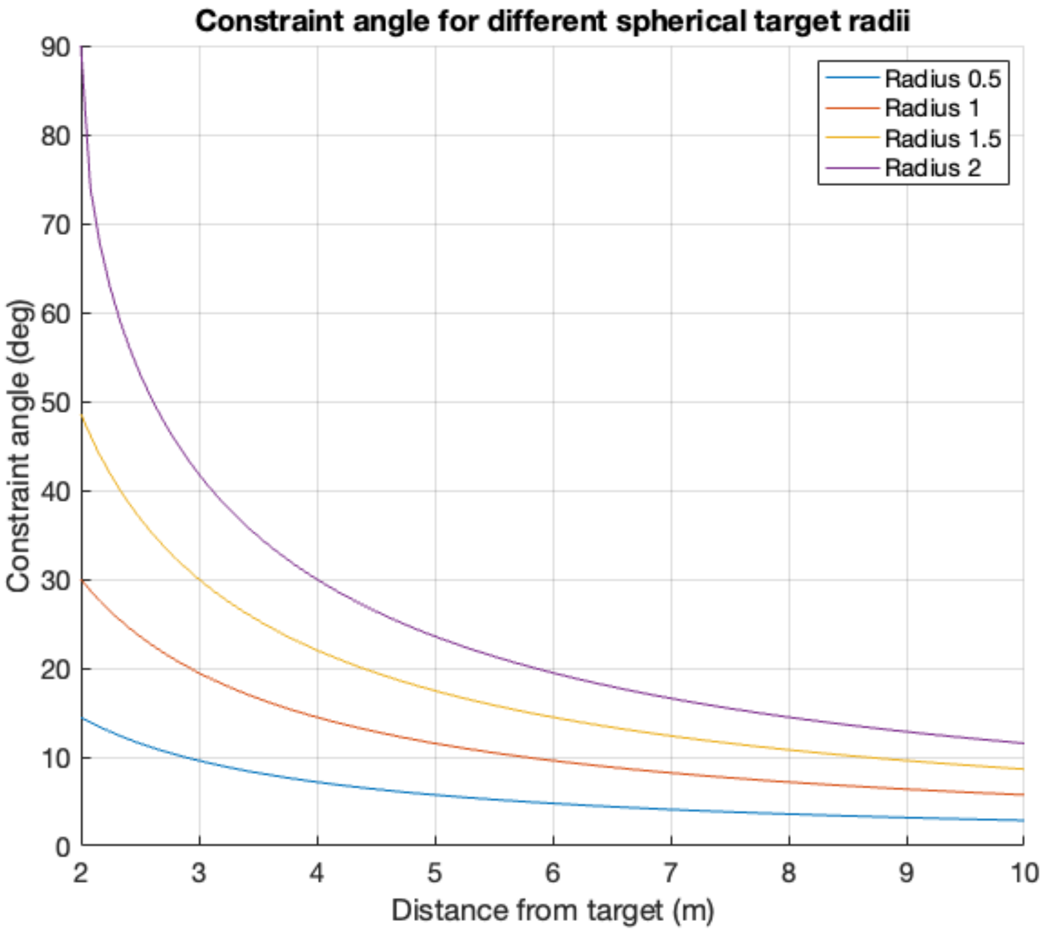


Thrust profile obtained with continuation and smoothing

Simulation Results – Spherical Target



Solution with 0.5 meter radius pointing constraint



Constraint angle as a function of distance from a spherical target

Conclusion

- Part I
  - Picard-Chebyshev methods, gravity approximations & adaptive (bang-bang conscious) segmentation scheme outperform explicit integrators
- Part II
  - Thruster pointing constrained fuel-optimal control using indirect methods shows promising initial results
- These contributions are an important step towards successful, large-scale and routine in-space sustainability operations

This work was partially supported by the Aerospace Corporation and the Australian-American Association under the Fulbright Future Program.

Contact:

Robyn M. Woollands: rmw@Illinois.edu