8-2024

Cislunar Optimal Robust Trajectory Design With Corrective Maneuvers

Scott J. Kelly
Utah State University

CISLUNAR OPTIMAL ROBUST TRAJECTORY DESIGN WITH CORRECTIVE MANEUVERS

by

Scott J. Kelly

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Aerospace Engineering

Approved:

David Geller, Ph.D. Major Professor	Matthew Harris, Ph.D. Committee Member
Douglas Hunsaker, Ph.D. Committee Member	Tianyi He, Ph.D. Committee Member
Jonathan Black, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

ABSTRACT

Cislunar Optimal Robust Trajectory Design with Corrective Maneuvers

by

Scott J. Kelly, Doctor of Philosophy

Utah State University, 2024

Major Professor: David Geller, Ph.D.
Department: Mechanical and Aerospace Engineering

The focus of this research is the optimization of impulsive transfer trajectories in cislunar space in the presence of stochastic effects. A deterministic optimal trajectory is developed. An equivalent robust optimal trajectory is developed by incorporating an initial position dispersion, velocity dispersion, maneuver execution error, and random distubances as process noise in the optimization problem. The cost function for the robust trajectory includes nominal impulsive maneuvers plus corrections. The robust trajectory is also subject to a terminal dispersion covariance constraint. The hypothesis is made that incorporating stochastic error sources will result in a different trajectory path and different nominal maneuvers when compared to the deterministic optimal trajectory. Specific applications include a low-Earth orbit to low lunar orbit transfer, a low-Earth orbit to Near Rectilinear Halo Orbit insertion, and a Near Rectliniear Halo Orbit rendezvous trajectory.

(182 pages)

PUBLIC ABSTRACT

Cislunar Optimal Robust Trajectory Design with Corrective Maneuvers

Scott J. Kelly

One common method for determining how to get from point A to point B is by finding the most fuel-efficient path. This is a common method for spacecraft trajectory designers also. Without any random or unexpected occurrences this path is called the deterministic optimal trajectory. Sometimes unexpected events happen that result in deviating from the most fuel-efficient path; while driving, you may encounter a detour. Naturally, we tend to return to the original fuel-efficient path and continue the journey. Similar deviations happen in space. The rocket delivering a spacecraft to orbit likely will not deliver exactly to the intended starting point of the most fuel-efficient path to the target. As a result, a mid-course correction is required to not miss the target which costs additional fuel. This research seeks to determine the most fuel-efficient path by optimizing the baseline maneuvers in addition to corrections when characterized sources of error exist. This is referred to as a robust optimal trajectory. To determine if the robust path is beneficial, this research compares the total cost of the optimal deterministic trajectory with corrections with the optimal robust trajectory and its associated corrections.

For Cody.

"That's no moon."

ACKNOWLEDGMENTS

I would like to acknowledge my wife Heather. Your patience, support, and love have made my PhD experience an enjoyable one. I am so proud of the mother you've become in the meanwhile.

I would also like to acknowledge my adviser Dr. Geller. It is clear you enjoy your work and genuinely care about your students. I am very thankful to be one of your students. Working with you has legitimately been fun! I hope we get to work together again in the future.

Scott J. Kelly

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Traveling the cosmos can be an uncertain endeavor with random error sources that are impossible to predict. There is error from the start in a spacecraft's actual and planned orbit at launch vehicle separation. At major mission events, maneuvers are rarely executed exactly according to the plan due to maneuver execution error. Throughout the mission, and especially during random venting that occurs in manned missions, spacecraft impart random impulses, disturbing the current trajectory. These errors sources have the potential to impact a spacecraft's planned mission and mission success mandates planning for these inherent errors. A common plan may be to carry sufficient fuel margin to correct dispersions along a deterministically planned trajectory. With properly characterized error sources, a Monte Carlo analysis can inform the required quantity of fuel margin but does not provide a framework for determining path choices to minimize the required propellant. This work presents a robust trajectory design method that accounts for the impact of an initial state dispersion, maneuver execution error, and process noise, simultaneously optimizing the number and location of trajectory correction maneuvers (TCM) along the trajectory.

Cislunar space, the area of space affected by the gravitational influence of both Earth and the Moon, is a burgeoning area of interest and importance. The three-body problem and cislunar repeating trajectories are of particular interest due to recent and upcoming space exploration missions. The NASA Artemis program has a large cislunar component and includes a Near-Rectilinear Halo Orbit (NRHO) as an operational orbit for the Gateway Space Station [1]. Gateway is planned to serve as a base of operations and staging point for lunar surface missions, research, and a departure point for further space exploration. The unmanned Artemis I demonstration mission utilized a three body trajectory as part of

its mission plan, a Distant Retrograde Orbit (DRO). The CAPSTONE spacecraft was the first spacecraft to inject into and operate in an NRHO (in November 2022), arriving via a ballstic lunar transfer [2].

1.2 Literature Review

1.2.1 Deterministic Trajectory Design and Optimization

Optimization of deterministic impulsive trajectories has been a topic of extensive study. Conway presents a thorough history and overview [3] of early influential works in trajectory optimization [4–8]. Indirect optimization formed a large portion of the early trajectory optimization approaches prior to the advance of significant personal computing power. Betts presents a survey of numerical methods for trajectory optimization to include nonlinear programming, optimal control problems, numerical analysis, shooting methods, transcription, dynamic programming, and genetic algorithms [9]. Trajectory optimization algorithms that connect multiple events via segments and incorporate node flexibility enable optimization trades across an entire mission [10,11]. In two works by Ocampo forming the foundation for the COPERNICUS trajectory design tool, multiple impulses are simultaneously minimized while satisfying segment connectivity constraints as well as other constraint options [12,13].

In some cases, a deterministically planned trajectory may appear to provide a minimum fuel path to a target. The same trajectory may exhibit sensitivities that result in excessively expensive corrective maneuvers if not properly planned. On the other hand, robust trajectories that take into account uncertainties may require more energy to embark upon, but reduce the cost of expensive corrections. For example, it is commonly known that the Hohmann transfer is the theoretically optimal ΔV two impulse transfer. However, even as early as Apollo, transfer angles of 130 to 160 degrees for terminal phase initiation (TPI) burns were utilized rather than "optimal" 180 degrees to make the TPI time less sensitive to orbit insertion dispersions [14]. Space shuttle rendezvous maneuvers implemented a similar transfer angle to minimize sensitivities related to ΔV dispersions that are amplified with a 180 degree transfer angle.

1.2.2 Robust Trajectory Optimization

The search for a robust trajectory has taken numerous forms. In one of the earlier examples, Nishimura and Pfeiffer utilize a dynamic programming approach to develop trajectories with stochastic error sources that constrain trajectory corrections within a magnitude threshold and minimize dispersions at a target [15]. Jin et al. embedded a linear covariance analysis tool within a genetic algorithm to identify robust trajectories for rendezvous and proximity operations (RPO) that minimize nominal ΔV plus the sum of corrections [16]. Oguri and McMahon optimize trajectories in the presence of stochastic error sources to maximize the likelihood of a spacecraft's ability to meet its mission operating parameters [17]. Geller et al. incorporate robustness into linear covariance analysis by triggering terminal RPO maneuvers with an event occurrence rather than at a specific time [18]. Jenson and Scheeres approach a robust optimization problem with maneuver execution error using indirect methods [19]. Boone and McMahon perform optimization of a nonlinear system with impulsive controls and stochastic constraints [20]. Greco et al. optimize interplanetary transfers under uncertainty by abandoning the concept of a reference trajectory and errors with respect to the reference in favor of a more general framework where each sample is a separate trajectory with probability density [21].

In this dissertation, the robust trajectory design method is formulated as a direct optimization problem with a constraint on position dispersion at specified events and a cost function comprised of the mean (nominal) ΔV plus 3σ TCM δV (the lower-case δ signifies a TCM versus a nominal maneuver). Stochastic error sources include an initial position and velocity dispersion, maneuver execution error applied to nominal maneuvers and TCMs, and process noise to account for random spacecraft impulses and errors throughout the trajectory. The design method is meant to be agnostic to dynamical system; the problem scenarios presented are impulsive transfers in the classical two-body problem and the circular restricted three-body problem but the state variational equations and the Jacobian functions can be replaced with other systems as desired.

The direct optimization approach uses nonlinear programming (NLP) with analytical gradients of a multiple segment nominal trajectory. Using multiple trajectory segments enables the incorporation of specific events throughout a trajectory [22] and also results in additional benefits for nonlinear systems as sensitivities are distributed across the entire trajectory [23]. Linear covariance analysis rapidly estimates the state dispersion covariance and TCM covariance along each nominal trajectory using propagated state transition matrices (STM). Analytical gradients are derived to include the sensitivity of TCMs with respect to the problem parameters in a manner similar to the Jacobian in a multiple differential correction scheme [24]. Validation of the optimal NLP robust solutions for simpler scenarios is peformed using a brute force mission map approach; the mission maps also enable visualization of a simplified solution space and build confidence in the gradient-based optimizer's likelihood of identifying the optimal solution.

An easier approach as far as implementation effort is concerned is to implement NLP gradients using the finite difference method. This approach is a computationally time consuming one though and one that in the end may preventing convergence. At a minimum, the cost function and each non-zero constraint equation for a perturbed trajectory must be tested against a nominal case to determine the sensitivity of each to variations in problem parameters. The number of perturbed trajectory propagations grows quickly as the number of segments and problem parameters increases. The reduction in accuracy of gradients using the finite difference method versus analytical gradients is enough to prevent convergence in some of the subsequent trajectory design scenarios. The main drawback to analytical gradients is the initial effort in deriving and implementing the analytical gradients, however the benefits are significant run-time reduction and improved convergence properties [25]. Significant effort was spent in deriving and verifying the implemented analytical gradients, however, verification details are not presented in this dissertation.

Incorporating robust trajectory design stochastic elements in a NLP increases the complexity of deriving analytical gradients but does not prevent convergence to the robust nominal trajectory for the cases tested. Effectively, linear covariance equations are em-

bedded inside the NLP where the STM along portions of the nominal trajectory informs the covariance of the TCMs. First-order sensitivities are part of the cost function; as a result, analytical gradients involve taking the partial derivative of an STM with respect to state vectors. The resulting second-order terms are referred to as state transition tensors (STT) which are also propagated along a nominal trajectory similar to STM propagation. Process noise is also incorporated by numerically integrating its time rate of change equation between trajectory events (referred to as the Q-bar matrix or QBM). A process noise covariance state sensitivity tensor and its time rate of change equations are derived and numerically integrated to produce the process noise analytical gradients (referred to as the Q-bar tensor or QBT). Derivations for the covariance-based gradients are presented in this dissertation.

1.2.3 The Three-Body Problem

Finding repeating orbits in the CR3BP can be non-trivial. Linearization of the system dynamics about a Lagrange point allows for the direct solution of a repeating orbit, however when modeled in the nonlinear system it no longer repeats. It is possible to find a member of the planar Lyapunov family, for example, by iteratively correcting the repeating linearized solution using the sensitivities in a numerically propagated state transition matrix (STM) until convergence. Grebow [26] and Zimovan [27] describe methods for finding repeating trajectories via shooting methods, assembling constraint equations satisfied by repeating orbits, and performing differential corrections, a common term for using STM elements as sensitivities for a multi-variable Newton's method to adjust initial conditions and satisfy the constraint equations. They also use continuation techniques to generate families of repeating trajectories and stability properties to identify bifurcations and additional families.

Farquhar introduced the halo orbit concept [28] and proposed it as a promising space station orbit as early as 1970 due to the family's constant line of sight to Earth and portions of the lunar surface. Howell and Breakwell [29] initially explored NRHO properties in the CR3BP. Many additional interesting and promising natural dynamical structures exist in cislunar space [24, 27, 30, 31]. Nonlinearities and varying stability properties (including

incredibly large instabilities) exist for many CR3BP trajectories. The NRHO portion of the L2 halo orbit family is an ideal target orbit for extended missions due to favorable stability properties and very minimal orbital maintenance ΔV required. The orbital period of an individual trajectory varies as the family of orbits is traversed; selecting a member of the family with a period that is an even ratio with other nearby periodic motion can lead to favorable mission characteristics. For example, phasing a spacecraft in the 9:2 synodic ratio NRHO (completes 9 orbital revolutions for every two synodic lunar periods) creates a predictable eclipse environment enabling spacecraft eclipse minimization, further increasing the trajectory's utility. The L2 Southern halo NRHO places the majority of the trajectory with line of sight to the lunar south pole, an area of high interest for numerous reasons including complex geography harboring ice.

For three body orbital transfers involving additional complexity, long propagation times, or requiring multiple events, multiple shooting methods are commonly implemented [27, 32, 33]. State transition matrix elements containing sensitivities from the beginning to end of each segment are used to correct segment initial states to meet a set of constraints. When time frames are lengthy or dynamics are highly nonlinear, first order STM sensitivities may be insufficient in creating meaningful connections between initial and final state element variations whicn can prevent convergence. Multiple shooting or a multiple segment trajectory is a common three body trajectory design method which attempts to alleviate issues with nonlinearities and longer propagation times. The trade-off is an increase in complexity and numerical computation.

In many cases where a Newton's method-like application is implemented, an initial guess that is close to convergence is required. Dynamical systems theory techniques are commonly used to develop initial three body trajectory guesses prior to correction [34]. Invariant manifold theory is commonly applied to generate initial guesses. Stable and unstable manifolds are generated where nearly free departures/arrivals along an unstable/stable eigenvector direction are propagated forward/backward in time to find intersections with other feasible trajectories. Direct observation or other tools such as a Poincare mapping

assist in identifying desired connections of state variables. When found, these nearly connected trajectories form initial guesses that a multiple shooting algorithm can easily correct to connect and meet a set of constraints [32]. However, the result is a feasible trajectory that meets a set of constraints but is not optimal, even in a local sense.

1.3 Dissertation Outline

This dissertation is organized as follows:

- Chapter 2: Mathematical Background
  - This chapter presents an overview of the theory and derivation of the two-body problem, the circular restricted three-body problem, and lays the groundwork for the first and second-order state sensitivity theory.
- Chapter 3: Deterministic Trajectory Design
  - The deterministic optimal trajectory serves as the comparison baseline for the robust trajectory results. This chapter presents numerical methods for differential correction, a demonstration of their application for finding CR3BP repeating trajectories, and introduces an optimization approach via a nonlinear program (NLP) with a similar multiple segment trajectory design approach.
- Chapter 4: Two-Body Trajectories Robut to Initial Dispersion
  - The robust trajectory NLP solution method is presented in this chapter. The only error source that is incorporated in finding two-body robust trajectories is an initial state dispersion. The derivation of the analytical gradient of linear covariance-based terms is presented, requiring the propagation of state transition tensors to implement. The two-body results closely mirror the author's published work in [35].
- Chapter 5: Stochastic Analysis Along a Nominal Trajectory
  - This chapter introduces additional error sources: nominal maneuver execution error, trajectory correction maneuver execution error, and random disturbances modeled as process noise. The accumulated state dispersion covariance from process noise, referred to as the Q-bar matrix (QBM) is introduced and incorporated

into the state dispersion at appropriate trajectory events. Analytical gradient derivation of linear covariance-based cost terms with the QBM require the QBM state sensitivity, referred to as the Q-bar tensor (QBT), which is also derived in this chapter.

  - This chapter also presents a fast TCM optimization method and a comparison of the optimal TCM set with variations in stochastic parameters.

- Chapter 6: Cislunar Robust Trajectory Design
  - Three-body robust trajectory results are presented and compared to their deterministic optimal equivalent trajectories. Specific trajectories include: low-Earth orbit to low-lunar orbit (two and three nominal impulsive maneuvers); low-Earth orbit to powered lunar flyby to NRHO insertion; two-impulse NRHO rendezvous.
- Chapter 7: Results Verification
  - This chapter presents three verification methods of robust results presented. First, a Monte Carlo analysis verifies the accuracy of modeled TCM magnitude estimates. Second, a genetic algorithm is used to verify the optimality of the TCM solution set chosen along a nominal trajectory. Third, the robust solution space for the two impulse NRHO rendezvous trajectory is observed as a series of contour plots to verify NLP convergence to the optimal solution.
- Chapter 8: Conclusions, Final Remarks, and Future Work

1.4 Summary of Contributions

- A paper similar to the contents of Chapters 3 and 4 titled "Optimal Robust Two-Body Trajectory Design with Corrective Maneuvers" was published in the Journal of Spacecraft and Rockets [35] and presented at the 45th American Astronautical Society Guidance & Control Conference in Breckenridge, CO, February 2023.
- A paper similar to the contents of Chapters 5 and 6 titled "Robust Cislunar Trajectory Optimization in the Presence of Stochastic Errors" was submitted to and is in the review process with the Journal of Astronautical Sciences. Another similar set of

results was presented at the 46th American Astronautical Society Guidance & Control Conference, February 2024.

- Unique technical contributions to the field include:
  - The simultaneous optimization of deterministic cost plus a stochastic cost estimate via a nonlinear program.
  - The use of state transition tensors in the analytical gradient of linear covariance-based stochastic cost estimates.
  - The fast TCM optimization method presented in Chapter 5.
  - The propagation and manipulation of the QBM history alongside STM for incorporation in linear covariance analysis.
  - The propagation and manipulation of the QBT history alongside the QBM for use in the derivation of the QBM analytical gradients.
  - The development of a trajectory that could save NASA up to 77.8 m/s in total upper bound maneuver requirement for a spacecraft traveling from LEO to the NRHO Gateway.

CHAPTER 2

MATHEMATICAL BACKGROUND

2.1 Dynamical Systems

2.1.1 N-Body Problem

In the problem of n-bodies, from Newton's law of gravitation, the gravitational forces on each of the n-bodies imparted by the others, modeled as point masses, are described by Equation 2.1 where $r_{ij} = \|\boldsymbol{r}_j - \boldsymbol{r}_i\|$.

$$\boldsymbol{f}_i = G\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{m_i m_j}{r_{ij}^3} (\boldsymbol{r}_j - \boldsymbol{r}_i) \tag{2.1}$$

Assuming that only gravitational forces are acting on each body, from Newton's second law, Equation 2.2 describes the motion of the ith particle with respect to an inertial frame. The result is $3n$ second-order differential equations.

$$\ddot{\boldsymbol{r}}_i = G\sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{m_j}{r_{ij}^3} (\boldsymbol{r}_j - \boldsymbol{r}_i) \tag{2.2}$$

2.1.2 Two-Body Problem

For the two-body problem the equations of motion defining the motion of bodies 1 and 2 are:

$$\begin{aligned} \ddot{\boldsymbol{r}}_1^I &= \tfrac{Gm_2\boldsymbol{r}_{12}}{r_{12}^3} \\ \ddot{\boldsymbol{r}}_2^I &= -\tfrac{Gm_1\boldsymbol{r}_{12}}{r_{12}^3} \end{aligned} \tag{2.3}$$

When modeling the motion of a spacecraft, the mass of the first body (Earth or another planetary body) is assumed to be much greater than the second (the spacecraft). Ignoring

the effect of the spacecraft on the motion of the first body results in the following 3 second-order differential equations where μ is the gravitational parameter of the first body:

$$\ddot{\boldsymbol{r}}_2 = -\frac{\mu \boldsymbol{r}_{12}}{r_{12}^3} \tag{2.4}$$

2.1.3 Three-Body Problem

For the three-body problem, Equation 2.2 results in 9 second-order differential equations taken in an inertial frame:

$$\begin{aligned} \ddot{\boldsymbol{r}}_1^I &= \tfrac{Gm_3\boldsymbol{r}_{13}}{r_{13}^3} + \tfrac{Gm_2\boldsymbol{r}_{12}}{r_{12}^3} \\ \ddot{\boldsymbol{r}}_2^I &= \tfrac{Gm_3\boldsymbol{r}_{23}}{r_{23}^3} - \tfrac{Gm_1\boldsymbol{r}_{12}}{r_{12}^3} \\ \ddot{\boldsymbol{r}}_3^I &= -\tfrac{Gm_1\boldsymbol{r}_{13}}{r_{13}^3} - \tfrac{Gm_2\boldsymbol{r}_{23}}{r_{23}^3} \end{aligned} \tag{2.5}$$

where the relative position vectors are

$$\begin{aligned} \boldsymbol{r}_{13} &= \boldsymbol{r}_3 - \boldsymbol{r}_1 \\ \boldsymbol{r}_{23} &= \boldsymbol{r}_3 - \boldsymbol{r}_2 \\ \boldsymbol{r}_{12} &= \boldsymbol{r}_2 - \boldsymbol{r}_1 \end{aligned} \tag{2.6}$$

For the case where the motion of m_3 is of interest with respect to m_1 and m_2, the relative motion equations are found as the difference of inertial derivatives:

$$\begin{aligned} \ddot{\boldsymbol{r}}_{13}^I &= \ddot{\boldsymbol{r}}_3^I - \ddot{\boldsymbol{r}}_1^I = -\tfrac{G\boldsymbol{r}_{13}}{r_{13}^3}(m_1 + m_3) - Gm_2\left(\tfrac{\boldsymbol{r}_{23}}{r_{23}^3} + \tfrac{\boldsymbol{r}_{12}}{r_{12}^3}\right) \\ \ddot{\boldsymbol{r}}_{23}^I &= \ddot{\boldsymbol{r}}_3^I - \ddot{\boldsymbol{r}}_2^I = -\tfrac{G\boldsymbol{r}_{23}}{r_{23}^3}(m_2 + m_3) - Gm_1\left(\tfrac{\boldsymbol{r}_{13}}{r_{13}^3} + \tfrac{\boldsymbol{r}_{12}}{r_{12}^3}\right) \end{aligned} \tag{2.7}$$

2.1.4 Circular Restricted Three Body Problem

The circular restricted three body problem (CR3BP) is a simplification of the gravitational attraction of three bodies (the classical three body problem) [36, 37]. The first CR3BP simplification (presented by Lagrange) relates to the mass of one of the bodies: the mass of one body is assumed to be neglibible in comparison to the two other massive

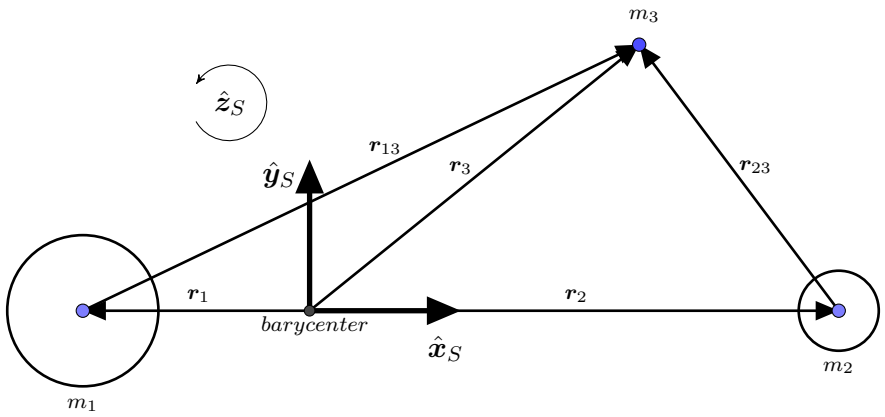


Fig. 2.1: CR3BP Synodic Rotating Coordinate Frame

bodies. This is a reasonable assumption; a nearby spacecraft does not measurably change the orbit of the Earth or the Moon.

The second CR3BP simplification is the assumption that the relative motion of the two primary bodies is circular. Circular motion means a constant distance between the two primaries and rotation at a constant angular rate ($\dot{\boldsymbol{\omega}} = 0$). The result is an autonomous system that does not require knowledge of each primary's location at a specific time. In fact, aligning a coordinate frame that rotates at the angular rate of the two primaries results in the location of each primary as a fixed point independent of time. This assumption simplifies the study of the system's nonlinear differential equations, and enables the development of families of repeating orbits as well as the application of dynamical systems theory [32, 34], among others.

In the context of trajectory design, the CR3BP is typically used as a starting point to generate an initial guess that is refined in a more complicated model. Models of increasing fidelity typically include the elliptical resricted three body problem [38, 39], the bicircular restricted four body problem [40, 41], or an ephemeris model that incorporates the time-dependent location of any number of planetary bodies [42, 43]. For cislunar applications, a common ephemeris model incorporates the locations of Earth, Sun, Moon, and Jupiter [44].

Substituting $m_3 = 0$ into Equation 2.7 results in the equations of motion for the restricted three body problem:

$$\begin{aligned}\ddot{\boldsymbol{r}}_{13}^I &= -\tfrac{G\boldsymbol{r}_{13}}{r_{13}^3}(m_1) - Gm_2\left(\tfrac{\boldsymbol{r}_{23}}{r_{23}^3} + \tfrac{\boldsymbol{r}_{12}}{r_{12}^3}\right) = -\mu_1\tfrac{\boldsymbol{r}_{13}}{r_{13}^3} - \mu_2\left(\tfrac{\boldsymbol{r}_{23}}{r_{23}^3} + \tfrac{\boldsymbol{r}_{12}}{r_{12}^3}\right) \\ \ddot{\boldsymbol{r}}_{23}^I &= -\tfrac{G\boldsymbol{r}_{23}}{r_{23}^3}(m_2) - Gm_1\left(\tfrac{\boldsymbol{r}_{13}}{r_{13}^3} + \tfrac{\boldsymbol{r}_{12}}{r_{12}^3}\right) = -\mu_2\tfrac{\boldsymbol{r}_{23}}{r_{23}^3} - \mu_1\left(\tfrac{\boldsymbol{r}_{13}}{r_{13}^3} + \tfrac{\boldsymbol{r}_{12}}{r_{12}^3}\right)\end{aligned} \tag{2.8}$$

Since the mass of the third body is negligible, the motion of m_1 and m_2 is Keplerian about their common barycenter and not necessarilty circular at this point.

A convenient coordinate frame is centered at the system barycenter and rotates at the rotation rate of m_1 and m_2, $\omega_{S/I}$. This rotating coordinate system is commonly called the synodic frame (or the S frame) [36]. The synodic frame's x-axis, $\hat{\boldsymbol{x}}_S$, points from the barycenter toward m_2. The synodic frame's z-axis, $\hat{\boldsymbol{z}}_S$, is aligned with the angular momentum vector of the Moon's orbit such that $\boldsymbol{\omega}_{S/I} = \omega_{S/I}\hat{\boldsymbol{z}}_S$. The y-axis completes the right-handed coordinate system; as a result, $\hat{\boldsymbol{x}}_S$ and $\hat{\boldsymbol{y}}_S$ lie in the Earth-Moon orbital plane. Figure 2.1 shows the $\hat{\boldsymbol{x}}_S$, $\hat{\boldsymbol{y}}_S$, and $\hat{\boldsymbol{z}}_S$ axes. In the S frame, the position of m_1 and m_2 (defined by $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$) will only have an $\hat{\boldsymbol{x}}_S$ component. While it may appear that m_3 lies in the $\hat{\boldsymbol{x}}_S$-$\hat{\boldsymbol{y}}_S$ plane, there is no $\hat{\boldsymbol{z}}_S$ component restriction on $\boldsymbol{r}_3$.

The transport theorem is required to compute the m_3 equations of motion in the rotating S frame. Equation 2.9 shows the kinematics relating the inertial derivative to the derivative in the rotating frame.

$$\dot{\boldsymbol{r}}_3^I = \dot{\boldsymbol{r}}_3^S + \boldsymbol{\omega}_{S/I} \times \boldsymbol{r}_3 \tag{2.9}$$

Equation 2.10 is the result of applying the transport theorem a second time, relating inertial acceleration to acceleration in the rotating frame.

$$\ddot{\boldsymbol{r}}_3^I = \ddot{\boldsymbol{r}}_3^S + 2\boldsymbol{\omega}_{S/I} \times \dot{\boldsymbol{r}}_3^S + \dot{\boldsymbol{\omega}}_{S/I}^S \times \dot{\boldsymbol{r}}_3 + \boldsymbol{\omega}_{S/I} \times \left(\boldsymbol{\omega}_{S/I} \times \boldsymbol{r}_3\right) \tag{2.10}$$

Combining the previous expression for $\ddot{\boldsymbol{r}}_3^I$ from Equation 2.5 with Equation 2.10 results in Equation 2.11.

$$\ddot{\boldsymbol{r}}_3^S + 2\boldsymbol{\omega}_{S/I} \times \dot{\boldsymbol{r}}_3^S + \dot{\boldsymbol{\omega}}_{S/I}^S \times \dot{\boldsymbol{r}}_3 + \boldsymbol{\omega}_{S/I} \times \left(\boldsymbol{\omega}_{S/I} \times \boldsymbol{r}_3\right) = -\frac{Gm_1\boldsymbol{r}_{13}}{r_{13}^3} - \frac{Gm_2\boldsymbol{r}_{23}}{r_{23}^3} \tag{2.11}$$

In the S frame, the position vectors of each mass are:

$$\boldsymbol{r}_1 = -r_1\hat{\boldsymbol{x}}_S$$
$$\boldsymbol{r}_2 = r_2\hat{\boldsymbol{x}}_S$$
$$\boldsymbol{r}_3 = x\hat{\boldsymbol{x}}_S + y\hat{\boldsymbol{y}}_S + z\hat{\boldsymbol{z}}_S$$

and the relative position vectors previously defined in Equation 2.6 become:

$$\boldsymbol{r}_{13} = \boldsymbol{r}_3 - \boldsymbol{r}_1 = (x + r_1)\,\hat{\boldsymbol{x}}_S + y\hat{\boldsymbol{y}}_S + z\hat{\boldsymbol{z}}_S$$
$$\boldsymbol{r}_{23} = \boldsymbol{r}_3 - \boldsymbol{r}_2 = (x - r_2)\,\hat{\boldsymbol{x}}_S + y\hat{\boldsymbol{y}}_S + z\hat{\boldsymbol{z}}_S$$

Finally, substituting the relative position vectors into Equation 2.11 results in Equation 2.12 in component form. These equations define the motion of the elliptical restricted three body problem. An important observation is that r_1 and r_2 are not constant for elliptical motion; they are time-dependent functions of the relative position of m_1 and m_2.

$$\begin{bmatrix} \ddot{x} - 2\omega_{S/I}\dot{y} - \dot{\omega}_{S/I}y - \omega_{S/I}^2 x \\ \ddot{y} + 2\omega_{S/I}\dot{x} + \dot{\omega}_{S/I}x - \omega_{S/I}^2 y \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} -\mu_1\frac{(x+r_1)}{r_{13}^3} - \mu_2\frac{(x-r_2)}{r_{23}^3} \\ -\mu_1\frac{y}{r_{13}^3} - \mu_2\frac{y}{r_{23}^3} \\ -\mu_1\frac{z}{r_{13}^3} - \mu_2\frac{z}{r_{23}^3} \end{bmatrix} \tag{2.12}$$

The next simplification to reach the CR3BP equations of motion is enforcing circular motion for masses m_1 and m_2 about their barycenter. In circular motion, the distance from each primary mass to the barycenter is constant. Additionally, the angular rate is constant

and $\dot{\omega}_{S/I} = 0$. Equation 2.13 shows the resulting CR3BP equations of motion.

$$\begin{bmatrix} \ddot{x} - 2\omega_{S/I}\dot{y} - \omega_{S/I}^2 x \\ \ddot{y} + 2\omega_{S/I}\dot{x} - \omega_{S/I}^2 y \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} -\mu_1 \frac{(x+r_1)}{r_{13}^3} - \mu_2 \frac{(x-r_2)}{r_{23}^3} \\ -\mu_1 \frac{y}{r_{13}^3} - \mu_2 \frac{y}{r_{23}^3} \\ -\mu_1 \frac{z}{r_{13}^3} - \mu_2 \frac{z}{r_{23}^3} \end{bmatrix} \tag{2.13}$$

Szebehely shows that the system in Equation 2.13 depends only on one parameter through the use of dimensionless variables [36]. Currently, the system depends on the two primary gravitational parameters (μ_1 and μ_2), the distance from each primary to the system barycenter (r_1 and r_2), and the angular rate of the system ($\omega_{S/I}$). Leveraging the fact that many of these parameters are functions of the two gravitational parameters enables the reduction in dependence.

First, defining a system gravitational parameter μ (Equation 2.14) enables the expression of each primary gravitational parameter as a function of μ. If the mass of the system is scaled such that $\mu_1 + \mu_2 = 1$, then $\mu_2 = \mu$ and $\mu_1 = 1 - \mu$.

$$\mu = \frac{\mu_2}{\mu_1 + \mu_2} \tag{2.14}$$

Next, using the equation for the center of mass (Equation 2.15) and scaling system distances such that the distance between the primaries is equal to 1 ($\|\tilde{\boldsymbol{r}}_{12}\| = \|\tilde{\boldsymbol{r}}_2 - \tilde{\boldsymbol{r}}_1\| = \tilde{r}_1 + \tilde{r}_2 = 1$) enables the expression of the position of each primary as a function of μ (Equations 2.16 and 2.17). The tilde indicates a dimensionless parameter; i.e., for distance, $\tilde{r}_1 = \frac{r_1}{r_{12}}$.

$$-\mu_1 r_1 + \mu_2 r_2 = 0 \tag{2.15}$$

$$\tilde{r}_1 = \mu \tilde{r}_{12} = \mu \tag{2.16}$$

$$\tilde{r}_2 = (1 - \mu)\tilde{r}_{12} = 1 - \mu \tag{2.17}$$

Finally, finding the mean motion n (Equation 2.18) and scaling time such that nondimensional time $\tau = tn$ enables the expression of the angular rate as a ratio of angle traveled

per nondimensional time τ.

$$n = \sqrt{\frac{\mu_1 + \mu_2}{r_{12}^3}} \tag{2.18}$$

$$\tilde{\omega}_{S/I} = \frac{\omega_{S/I}}{n} = 1 \tag{2.19}$$

The result is the non-dimensional CR3BP equations of motion (Equation 2.20). The system of equations is now only dependent on the mass ratio of the two primaries and the position of m_3.

$$\begin{bmatrix} \ddot{\tilde{x}} - 2\dot{\tilde{y}} - \tilde{x} \\ \ddot{\tilde{y}} + 2\dot{\tilde{x}} - \tilde{y} \\ \ddot{\tilde{z}} \end{bmatrix} = \begin{bmatrix} -\frac{(1-\mu)(\tilde{x}+\mu)}{\tilde{r}_{13}^3} - \frac{\mu(\tilde{x}-(1-\mu))}{\tilde{r}_{23}^3} \\ -\frac{(1-\mu)\tilde{y}}{\tilde{r}_{13}^3} - \frac{\mu\tilde{y}}{\tilde{r}_{23}^3} \\ -\frac{(1-\mu)\tilde{z}}{\tilde{r}_{13}^3} - \frac{\mu\tilde{z}}{\tilde{r}_{23}^3} \end{bmatrix} \tag{2.20}$$

where

$$\tilde{r}_{13} = \sqrt{(\tilde{x} + \mu)^2 + \tilde{y}^2 + \tilde{z}^2} \tag{2.21}$$

$$r_{23} = \sqrt{(\tilde{x} - 1 + \mu)^2 + \tilde{y}^2 + \tilde{z}^2} \tag{2.22}$$

While there is no closed-form solution to Equation 2.20, given an initial state for m_3, it is possible to numerically integrate the corresponding six first-order differential equations and determine a future state for m_3. For implementation, the initial state must be converted to nondimensional units. As an example, conversion of the m_3 state vector to nondimensional distance and nondimensional velocity occurs via the following relationship:

$$\tilde{\boldsymbol{r}}_3 = \frac{\boldsymbol{r}_3}{r_{12}} \tag{2.23}$$

$$\tilde{\boldsymbol{v}}_3 = \boldsymbol{v}_3 \frac{n}{r_{12}} \tag{2.24}$$

Rearranging the CR3BP equations of motion

$$\begin{aligned} \ddot{\tilde{x}} &= 2\dot{\tilde{y}} + \tilde{x} - \tfrac{1-\mu}{\tilde{r}_{13}^3}(\tilde{x}+\mu) - \tfrac{\mu}{\tilde{r}_{23}^3}(\tilde{x}-1+\mu) \\ \ddot{\tilde{y}} &= -2\dot{\tilde{x}} + \tilde{y} - \tfrac{1-\mu}{\tilde{r}_{13}^3}\tilde{y} - \tfrac{\mu}{\tilde{r}_{23}^3}\tilde{y} \\ \ddot{\tilde{z}} &= -\tfrac{1-\mu}{\tilde{r}_{13}^3}\tilde{z} - \tfrac{\mu}{\tilde{r}_{23}^3}\tilde{z} \end{aligned} \tag{2.25}$$

and introducing a pseudo-potential function U

$$U = \frac{\tilde{x}^2 + \tilde{y}^2}{2} + \frac{1-\mu}{\tilde{r}_{13}} + \frac{\mu}{\tilde{r}_{23}} \tag{2.26}$$

enables writing the equations of motion in the following alternate form:

$$\begin{aligned} \ddot{\tilde{x}} &= \tfrac{\partial U}{\partial \tilde{x}} + 2\dot{\tilde{y}} \\ \ddot{\tilde{y}} &= \tfrac{\partial U}{\partial \tilde{y}} - 2\dot{\tilde{x}} \\ \ddot{\tilde{z}} &= \tfrac{\partial U}{\partial \tilde{z}} \end{aligned} \tag{2.27}$$

2.2 First-order Dynamics and State Sensitivity

In this dissertation, the six-dimensional spacecraft state represents the spacecraft position and velocity:

$$\boldsymbol{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^\top \tag{2.28}$$

Error sources create a dispersion, $\delta\boldsymbol{x}$, with respect to the nominal trajectory $\boldsymbol{x}_N$. The dynamics of the dispersed state are a function of the nominal state and the dispersion:

$$\dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_N + \delta\dot{\boldsymbol{x}} = \boldsymbol{f}\left(\boldsymbol{x} + \delta\boldsymbol{x}\right) \tag{2.29}$$

A first-order Taylor series expansion (TSE) along the nominal trajectory provides an estimate for the dispersed state at a future time:

$$\dot{\boldsymbol{x}} \approx \boldsymbol{f}\left(\boldsymbol{x}_N\right) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_N} \delta\boldsymbol{x} \tag{2.30}$$

which simplifies to the linearized dynamics along the nominal reference trajectory

$$\delta\dot{\boldsymbol{x}} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_N} \delta\boldsymbol{x} \tag{2.31}$$

and has the solution

$$\delta\boldsymbol{x}\left(t\right) = \Phi\left(t, t_0\right)\delta\boldsymbol{x}\left(t_0\right) \tag{2.32}$$

where $\Phi\left(t, t_0\right)$ is the state transition matrix (STM), which contains the first-order dynamics between t_0 and t along the reference trajectory [32]:

$$\Phi(t,t_0) = \frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{x}(t_0)} = \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} & \frac{\partial x}{\partial z_0} & \frac{\partial x}{\partial \dot{x}_0} & \frac{\partial x}{\partial \dot{y}_0} & \frac{\partial x}{\partial \dot{z}_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} & \frac{\partial y}{\partial z_0} & \frac{\partial y}{\partial \dot{x}_0} & \frac{\partial y}{\partial \dot{y}_0} & \frac{\partial y}{\partial \dot{z}_0} \\ \frac{\partial z}{\partial x_0} & \frac{\partial z}{\partial y_0} & \frac{\partial z}{\partial z_0} & \frac{\partial z}{\partial \dot{x}_0} & \frac{\partial z}{\partial \dot{y}_0} & \frac{\partial z}{\partial \dot{z}_0} \\ \frac{\partial \dot{x}}{\partial x_0} & \frac{\partial \dot{x}}{\partial y_0} & \frac{\partial \dot{x}}{\partial z_0} & \frac{\partial \dot{x}}{\partial \dot{x}_0} & \frac{\partial \dot{x}}{\partial \dot{y}_0} & \frac{\partial \dot{x}}{\partial \dot{z}_0} \\ \frac{\partial \dot{y}}{\partial x_0} & \frac{\partial \dot{y}}{\partial y_0} & \frac{\partial \dot{y}}{\partial z_0} & \frac{\partial \dot{y}}{\partial \dot{x}_0} & \frac{\partial \dot{y}}{\partial \dot{y}_0} & \frac{\partial \dot{y}}{\partial \dot{z}_0} \\ \frac{\partial \dot{z}}{\partial x_0} & \frac{\partial \dot{z}}{\partial y_0} & \frac{\partial \dot{z}}{\partial z_0} & \frac{\partial \dot{z}}{\partial \dot{x}_0} & \frac{\partial \dot{z}}{\partial \dot{y}_0} & \frac{\partial \dot{z}}{\partial \dot{z}_0} \end{bmatrix} \tag{2.33}$$

Intuitively, the STM describes the sensitivity of a future state with respect to modifications in the current state.[1]

The linear differential equations for propagating the STM are derived by substituting Equation 2.33 into Equation 2.31:

$$\dot{\Phi}\left(t, t_0\right) = F\left(\boldsymbol{x}_N\right)\Phi\left(t, t_0\right) \tag{2.34}$$

where F represents the Jacobian, also sometimes referred to as the system matrix:

$$F\left(\boldsymbol{x}_N\right) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_N} \tag{2.35}$$

The initial condition for the linear dynamics between t_0 and t_0 is the identity matrix:

[1] This train of thought is useful when deriving and utilizing constraint equations in subsequent chapters; i.e., what is the sensitivity of the subject equation to variations in the state at a specific point or segment along the reference trajectory.

$$\Phi(t_0, t_0) = I_{6\times 6} \tag{2.36}$$

For the CR3BP, evaluating Equation 2.35 yields

$$F(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ U_{xx} & U_{xy} & U_{xz} & 0 & 2 & 0 \\ U_{yx} & U_{yy} & U_{yz} & -2 & 0 & 0 \\ U_{zx} & U_{zy} & U_{zz} & 0 & 0 & 0 \end{bmatrix} \tag{2.37}$$

where U_{xx} represents the second partial derivative of U with respect to x, etc. The partial derivatives of the gravitational potential are:

$$U_{xx} = 1 - \frac{1-\mu}{\tilde{r}_1^3} - \frac{\mu}{\tilde{r}_2^3} + \frac{3(1-\mu)}{\tilde{r}_1^5}(\tilde{x}+\mu)^2 + \frac{3\mu}{\tilde{r}_2^5}(\tilde{x}-1+\mu)^2 \tag{2.38}$$

$$U_{yy} = 1 - \frac{1-\mu}{\tilde{r}_1^3} - \frac{\mu}{\tilde{r}_2^3} + \frac{3(1-\mu)}{\tilde{r}_1^5}\tilde{y}^2 + \frac{3\mu}{\tilde{r}_2^5}\tilde{y}^2 \tag{2.39}$$

$$U_{zz} = -\frac{1-\mu}{\tilde{r}_1^3} - \frac{\mu}{\tilde{r}_2^3} + \frac{3(1-\mu)}{\tilde{r}_1^5}\tilde{z}^2 + \frac{3\mu}{\tilde{r}_2^5}\tilde{z}^2 \tag{2.40}$$

$$U_{xy} = \frac{3(1-\mu)}{\tilde{r}_1^5}(\tilde{x}+\mu)\tilde{y} + \frac{3\mu}{\tilde{r}_2^5}(\tilde{x}-1+\mu)\tilde{y} \tag{2.41}$$

$$U_{xz} = \frac{3(1-\mu)}{\tilde{r}_1^5}(\tilde{x}+\mu)\tilde{z} + \frac{3\mu}{\tilde{r}_2^5}(\tilde{x}-1+\mu)\tilde{z} \tag{2.42}$$

$$U_{yz} = \frac{3(1-\mu)}{\tilde{r}_1^5}\tilde{y}\tilde{z} + \frac{3\mu}{\tilde{r}_2^5}\tilde{y}\tilde{z} \tag{2.43}$$

which are applied when numerically integrating the STM.

2.3 Second-order Dynamics and State Sensitivity

The first-order linearized dynamics provide the equations form determining STMs. Higher order approximations provide the equations for what are commonly called state

transition tensors (STT). This section introduces the theory for initializing and numerically integrating the second-order STT. Two papers by Park and Scheeres are frequently cited and form the basis of the derivation in 2.3.2 [45, 46].

2.3.1 Index / Einstein Notation

A brief introduction to tensor notation and operations is required prior to the second-order STT derivation. To this point, it is assumed that vector and matrix math (one and two dimensional arrays) rules are known. Additional rules are required for greater than two dimensional arrays.

Indices are applied to operations that indicate an array index that is summed over. For traditional matrix multiplication, $C = AB$, the columns of a row of A are multiplied with the corresponding rows of a column of B, then summed. This operation is represented by the following summation operation:

$$C\left(i,j\right) = \sum_{k} A\left(i,k\right) B\left(k,j\right) \tag{2.44}$$

However, the summation convention becomes cumbersome. The same operation written in index notation (also referred to as Einstein notation) is

$$C_{ij} = A_{ik} B_{kj} \tag{2.45}$$

such that when an index appears twice in a term (index k in this example), it is implied that it is summed over. This also aligns with the fact that the inner indices must match and the outer indices are the final dimensions in matrix multiplication, i.e.: $C_{4\times3} = A_{4\times2} B_{2\times3}$

When mutltiplying a transposed matrix, the order of the indices swap. For example, $E = AB^{\top}$ is

$$E_{ij} = A_{ik} B_{jk} \tag{2.46}$$

Applying the above two concepts to linear covariance propagation

$$P(t_2) = \Phi(t_2, t_1) P(t_1) \Phi(t_2, t_1)^\top \tag{2.47}$$

is made easier by breaking the operation into multiple simpler operations (generally a helpful strategy for deriving complex index notation applications):

$$P(t_2)_{ij} = D_{ik} \Phi(t_2, t_1)_{jk} \tag{2.48}$$

where

$$D_{ik} = \Phi(t_2, t_1)_{im} P(t_1)_{mk} \tag{2.49}$$

Combining Equations 2.48 and 2.49 reveals the index notation form for covariance propagation:

$$P(t_2)_{ij} = \Phi(t_2, t_1)_{im} P(t_1)_{mk} \Phi(t_2, t_1)_{jk} \tag{2.50}$$

Another feature is that the order terms are written in does not affect the result since the summed-over indices are explicit. Equations 2.50 and 2.51 are equivalent expressions.

$$P(t_2)_{ij} = P(t_1)_{mk} \Phi(t_2, t_1)_{im} \Phi(t_2, t_1)_{jk} \tag{2.51}$$

The following examples demonstrate some rules for vector and matrix differentiation. Column vector $\boldsymbol{p}$ is $n \times 1$, column vector $\boldsymbol{q}$ is $m \times 1$. The partial derivative of $\boldsymbol{p}$ with respect to $\boldsymbol{q}$ is

$$\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial p_1}{\partial q_1} & \frac{\partial p_1}{\partial q_2} & \cdots & \frac{\partial p_1}{\partial q_m} \\ \frac{\partial p_2}{\partial q_1} & & & \\ \vdots & & & \\ \frac{\partial p_n}{\partial q_1} & & & \frac{\partial p_n}{\partial q_m} \end{bmatrix} \tag{2.52}$$

In index notation, if matrix $W = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}$

$$W_{i,j} = \frac{\partial \boldsymbol{p}_i}{\partial \boldsymbol{q}_j} \tag{2.53}$$

where indices following commas identify the index across which a partial derivative is taken.

The partial derivative of a matrix M with respect to the vector $\boldsymbol{p}$ is a 3-dimensional array:

$$\frac{\partial M}{\partial \boldsymbol{p}} = \begin{bmatrix} \frac{\partial M}{\partial p_1} & \cdots & \frac{\partial M}{\partial p_n} \end{bmatrix} \tag{2.54}$$

or, in index notation, if $R = \frac{\partial M}{\partial \boldsymbol{p}}$:

$$R_{ij,k} = \frac{\partial M_{ij}}{\partial \boldsymbol{p}_k} \tag{2.55}$$

Similarly, the partial derivative of $\boldsymbol{p}$ with respect to M, in index notation, is

$$F_{i,jk} = \frac{\partial \boldsymbol{p}_i}{\partial M_{jk}} \tag{2.56}$$

For demonstrating the chain rule, suppose two column vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$, and two matrices, V and W with appropriate dimensions exist. The following equation also exists:

$$\boldsymbol{y} = VW\boldsymbol{x} \tag{2.57}$$

and the desired quantity is $\frac{\partial y}{\partial x}$. Defining an intermediate vector $\boldsymbol{m} = W\boldsymbol{x}$ results in

$$\boldsymbol{y} = V\boldsymbol{m} \tag{2.58}$$

and being intentionally verbose when evaluating $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$ shows a simple chain rule application:

$$\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{x}_j} = \frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{m}} \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{x}_j} \tag{2.59}$$

Treating the product on the right like a matrix multiplication and applying indices, vector $\boldsymbol{m}$ extends across the row elements of the first matrix and the column elements of the second

matrix. In this way, k is the element that is summed over in the index notation matrix multiplication.

$$\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{x}_j} = \frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{m}_k}\frac{\partial \boldsymbol{m}_k}{\partial \boldsymbol{x}_j} = VW \tag{2.60}$$

2.3.2 Second-order State Transition Tensor Derivation

The second-order STT derivation begins in a similar way that the (first-order) STM does, via a TSE corresponding to the desired order. The second-order TSE of the state dispersion at a future time is

$$\delta \boldsymbol{x}_i(t) \approx \Phi_{i,k_1}\delta \boldsymbol{x}^0_{k_1} + \frac{1}{2}\Phi_{i,k_1k_2}\delta \boldsymbol{x}^0_{k_1}\delta \boldsymbol{x}^0_{k_2} \tag{2.61}$$

where i, k_1, and k_2 are index notation subscripts:

$$\Phi_{i,k_1} = \frac{\partial \boldsymbol{x}\left(t\right)_i}{\partial \boldsymbol{x}\left(t_0\right)_{k_1}} \tag{2.62}$$

$\delta \boldsymbol{x}^0_{k_1}$ and $\delta \boldsymbol{x}^0_{k_2}$ are both the same values numerically with the index summation applied to different indices. Φ_{i,k_1k_2} represents the second-order STT. The second-order TSE of the system dynamics is

$$\delta \dot{\boldsymbol{x}}_i(t) = F_{i,k_1}\delta \boldsymbol{x}_{k_1} + \frac{1}{2}F_{i,k_1k_2}\delta \boldsymbol{x}_{k_1}\delta \boldsymbol{x}_{k_2} \tag{2.63}$$

where F_{i,k_1} is equivalent to 2.35 and F_{i,k_1k_2} is the second partial derivative of the system dynamics.

$$F_{i,k_1k_2} = \frac{\partial^2 \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}^2} = \frac{\partial F_{i,k_1}(\boldsymbol{x})}{\partial \boldsymbol{x}_{k_2}} \tag{2.64}$$

Substituting Equation 2.61 into Equation 2.63 (and appropriately adjusting indices) results in

$$\begin{aligned}\delta \dot{\boldsymbol{x}}_i(t) = {} & F_{i,m}\left(\Phi_{m,k_1}\delta \boldsymbol{x}^0_{k_1} + \frac{1}{2}\Phi_{m,k_1k_2}\delta \boldsymbol{x}^0_{k_1}\delta \boldsymbol{x}^0_{k_2}\right) \\ & + \frac{1}{2}F_{i,mn}\left(\Phi_{m,k_1}\delta \boldsymbol{x}^0_{k_1} + \frac{1}{2}\Phi_{m,k_1k_2}\delta \boldsymbol{x}^0_{k_1}\delta \boldsymbol{x}^0_{k_2}\right)\left(\Phi_{n,k_1}\delta \boldsymbol{x}^0_{k_1} + \frac{1}{2}\Phi_{n,k_1k_2}\delta \boldsymbol{x}^0_{k_1}\delta \boldsymbol{x}^0_{k_2}\right)\end{aligned} \tag{2.65}$$

The next step is taking the time derivative of Equation 2.61

$$\delta\dot{\boldsymbol{x}}_i(t) = \dot{\Phi}_{i,k_1}\delta\boldsymbol{x}^0_{k_1} + \frac{1}{2}\dot{\Phi}_{i,k_1k_2}\delta\boldsymbol{x}^0_{k_1}\delta\boldsymbol{x}^0_{k_2} \tag{2.66}$$

and equating Equations 2.65 and 2.66, from which the terms of equivalent order are equated. The first-order terms are:

$$\dot{\Phi}_{i,k_1} = F_{i,m}\Phi_{m,k_1} \tag{2.67}$$

which matches Equation 2.34, the usual STM time derivative. The second-order terms are:

$$\frac{1}{2}\dot{\Phi}_{i,k_1k_2}\delta\boldsymbol{x}^0_{k_1}\delta\boldsymbol{x}^0_{k_2} = F_{i,m}\left(\frac{1}{2}\Phi_{m,k_1k_2}\delta\boldsymbol{x}^0_{k_1}\delta\boldsymbol{x}^0_{k_2}\right) + \frac{1}{2}F_{i,mn}\Phi_{m,k_1}\delta\boldsymbol{x}^0_{k_1}\Phi_{n,k_2}\delta\boldsymbol{x}^0_{k_2} \tag{2.68}$$

revealing the second-order STT time derivative when common terms are canceled [45]:

$$\dot{\Phi}_{i,ab} = F_{i,m}\Phi_{m,ab} + F_{i,mn}\Phi_{m,a}\Phi_{n,b} \tag{2.69}$$

The initial conditions for the second-order STT for numerical integration are

$$\Phi_{m,ab} = 0_{6\times6\times6} \tag{2.70}$$

Similar to STM propagation, evaluating Equation 2.64 involves taking an additional partial derivative of Equation 2.37 with respect to each state variable[2]. The partial with

[2]The implementation of the third partial derivatives of the gravitational potential U was performed symbolically in Matlab. Repeated terms exist such that $U_{xyx} = U_{yxx}$ resulting in only ten total new terms.

respect to x follows:

$$F_{i,m1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ U_{xxx} & U_{xyx} & U_{xzx} & 0 & 0 & 0 \\ U_{yxx} & U_{yyx} & U_{yzx} & 0 & 0 & 0 \\ U_{zxx} & U_{zyx} & U_{zzx} & 0 & 0 & 0 \end{bmatrix} \tag{2.71}$$

The partial with respect to $\dot{x}$, $\dot{y}$, and $\dot{z}$ are each $0_{6\times6}$.

A few shorthand notations will be used for upcoming multisegment trajectory design. Regarding STM notation, the STM from the beginning to the end of segment i is $\Phi(t_{f,i}, t_{0,i})$. A shorthand notation used here for an STM that spans a segment is the replacement of time intervals for the segment with a superscript of the segment number i:

$$\Phi^i = \Phi(t_{f,i}, t_{0,i}) \tag{2.72}$$

In cases where an STM is required that exactly spans multiple segments, the colon operator is used. For example, the STM that spans segments 2 and 3 in this notation is

$$\Phi^{3:2} = \Phi^3\Phi^2 = \Phi(t_{f,3}, t_{0,2}) \tag{2.73}$$

Second-order STT terms are also used and share the same superscript notation shorthand with an additional Roman numeral subscript corresponding to the order of the terms contained (Φ^i_{II}):

$$\Phi_{m,ab}(t_{f,2}, t_{0,2}) = \Phi^2_{II} \tag{2.74}$$

The required dimension grows by a factor of six for second-order STTs. One interpretation of a second-order STT is a quantification of how the STM varies for a segment with respect

to variations in the segment's initial state:

$$\Phi_{II}^{i} = \frac{\partial \Phi^{i}}{\partial \boldsymbol{x}_{0,i}} = \frac{\partial^2 \boldsymbol{x}_{f,i}}{\partial \boldsymbol{x}_{0,i}^2} \tag{2.75}$$

which can also be interpreted as

$$\frac{\partial \Phi^{i}}{\partial \boldsymbol{x}_{0,i}} = \left[\begin{array}{cccccc} \frac{\partial \Phi^{i}}{\partial x_{0,i}}, & \frac{\partial \Phi^{i}}{\partial y_{0,i}}, & \frac{\partial \Phi^{i}}{\partial z_{0,i}}, & \frac{\partial \Phi^{i}}{\partial \dot{x}_{0,i}}, & \frac{\partial \Phi^{i}}{\partial \dot{y}_{0,i}}, & \frac{\partial \Phi^{i}}{\partial \dot{z}_{0,i}} \end{array} \right]_{6\times 6\times 6} \tag{2.76}$$

The second-order STT shorthand $\Phi_{II}\left(t, t_0\right)$ will only be used in lieu of index notation to represent the following form of matrix-tensor multiplication (a common form that can be thought of as an array of matrix multiplications in the third dimension). The matrix-tensor product of $\Phi^{(i+1)}\Phi_{II}^{i}$ can be thought of as distributing $\Phi^{(i+1)}$ across the array of matrices in the tensor:

$$\Phi_{p,m}\left(t_{f,i+1}, t_{0,i+1}\right)\Phi_{m,jk}\left(t_{f,i}, t_{0,i}\right) = \Phi^{(i+1)}\Phi_{II}^{i} = \left[\begin{array}{ccc} \Phi^{(i+1)}\frac{\partial \Phi^{i}}{\partial x_{0,i}} & \cdots & \Phi^{(i+1)}\frac{\partial \Phi^{i}}{\partial \dot{z}_{0,i}} \end{array} \right]_{6\times 6\times 6} \tag{2.77}$$

Combining second-order STTs is a bit more involved than combining sequential STMs. Combining a second-order STT from t_0 to t_1 with an STT from t_1 to t_2 requires STMs from the same two periods, $\Phi(t_1, t_0)$ and $\Phi\left(t_2, t_1\right)$, and the corresponding STTs, $\Phi_{II}(t_1, t_0)$ and $\Phi_{II}\left(t_2, t_1\right)$. Equation 2.78 shows the operation using index notation [45]:

$$\Phi_{II}\left(t_2, t_0\right)_{i,jk} = \Phi_{II}\left(t_2, t_1\right)_{i,pq}\Phi\left(t_1, t_0\right)_{p,j}\Phi\left(t_1, t_0\right)_{q,k} + \Phi\left(t_2, t_1\right)_{i,p}\Phi_{II}\left(t_1, t_0\right)_{p,jk} \tag{2.78}$$

Inverting a second-order STT requires the forward STT and inverse STM for a corresponding time period [45]:

$$\Phi_{II}\left(t_0, t_1\right)_{i,jk} = -\Phi\left(t_0, t_1\right)_{i,p}\Phi_{II}\left(t_1, t_0\right)_{p,qm}\Phi\left(t_0, t_1\right)_{q,j}\Phi\left(t_0, t_1\right)_{m,k} \tag{2.79}$$

CHAPTER 3
DETERMINISTIC TRAJECTORY DESIGN

3.1 Finding CR3BP Repeating Trajectories

3.1.1 Newton's Method

Typically, when searching for a root using the Newton-Rhapson method, a function's slope at an initial guess can be used to approximate a new x-intercept that is typically closer to the root than the initial guess. When successful, iterations can be performed until a maximum error threshold is achieved.

$$f'(x_i) = \frac{f(x_i) - f(x_{i+1})}{x_i - x_{i+1}} \tag{3.1}$$

Where the desired root satisfies

$$f(x_d) = 0 \tag{3.2}$$

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}} \tag{3.3}$$

and with some rearranging

$$0 = f(x_i) + f'(x_i)\,(x_{i+1} - x_i) \tag{3.4}$$

Additional rearranging reveals the traditional Newton-Rhapson iterative root-finding form:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{3.5}$$

3.1.2 Single Differential Correction

The above method can be applied to a nonlinear multi-dimensional system of equations where certain problem parameters $\boldsymbol{V}$ can be modified to satisfy a set of constraint equations $\boldsymbol{F}$ given an intial guess $\boldsymbol{V}^0$ [26, 27]. The constraints are

$$\boldsymbol{F} = [F_1, F_2, \ldots, F_m]^T \tag{3.6}$$

where each F_i equals zero at the desired solution $\boldsymbol{V}^d$:

$$\boldsymbol{F}(\boldsymbol{V}^d) = \boldsymbol{0} \tag{3.7}$$

The design variable vector, $\boldsymbol{V}$, can be formed from a variety of sources (state variables, integration times, etc.)

$$\boldsymbol{V} = [V_1, V_{2,}, \ldots, V_n]^T \tag{3.8}$$

Performing a first-order Taylor series expansion

$$\boldsymbol{F}(\boldsymbol{V}) \approx \boldsymbol{F}(\boldsymbol{V}^0) + \frac{\partial \boldsymbol{F}(\boldsymbol{V}^0)}{\partial \boldsymbol{V}^0}(\boldsymbol{V} - \boldsymbol{V}^0) \tag{3.9}$$

yields a result very similar to the single-variable Newton-Rhapson method. When solving for an update to the initial guess to satisfy the constraint equations, the equation becomes

$$\boldsymbol{0} = F(\boldsymbol{V}^0) + \frac{\partial \boldsymbol{F}(\boldsymbol{V}^0)}{\partial \boldsymbol{V}^0}(\boldsymbol{V} - \boldsymbol{V}^0) \tag{3.10}$$

$$0 = F(\boldsymbol{V}^i) + D\boldsymbol{F}(\boldsymbol{V}^i)(\boldsymbol{V}^{i+1} - \boldsymbol{V}^i) \tag{3.11}$$

which is more challenging to rearrange as direct scalar division isn't possible and matrix inversion may not be possible if $D\boldsymbol{F}$ is not square. If square

$$\boldsymbol{V}^{i+1} = \boldsymbol{V}^i - D\boldsymbol{F}(\boldsymbol{V}^i)^{-1}\boldsymbol{F}(\boldsymbol{V}^i) \tag{3.12}$$

In cases where there are more design variables than constraints, there are infinitely many solutions. $D\boldsymbol{F}$ is not square in these cases. A minimum norm pseudo-inverse can be implemented:

$$\boldsymbol{V}^{i+1} = \boldsymbol{V}^i - D\boldsymbol{F}(\boldsymbol{V}^i)^T \left[D\boldsymbol{F}(\boldsymbol{V}^i)D\boldsymbol{F}(\boldsymbol{V}^i)^T\right]^{-1} \boldsymbol{F}(\boldsymbol{V}^i) \tag{3.13}$$

The $D\boldsymbol{F}$ matrix describes how the constraint equations vary with changes in the design parameters.

$$D\boldsymbol{F}(\boldsymbol{V}) = \frac{\partial(\text{constraint equations})}{\partial(\text{design variables})} \tag{3.14}$$

State transition matrix elements satisfy these individual sensitivity terms in many cases (see Equation 2.33) and numerically integrating an STM in the CR3BP is trivial given an initial guess.

Targeting a Future Position

For targeting a position at time t, the state variables that are allowed to change instantaneously at time t_0 are the velocity terms:

$$\boldsymbol{V} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{3.15}$$

and the constraint equations to target a specific position at a future time t are

$$\boldsymbol{F} = \begin{bmatrix} x(t) - x_d \\ y(t) - y_d \\ z(t) - z_d \end{bmatrix} \tag{3.16}$$

The partials in the matrix $D\boldsymbol{F}$ are

$$D\boldsymbol{F} = \frac{\partial \boldsymbol{F}(\boldsymbol{V}^0)}{\partial \boldsymbol{V}^0} = \begin{bmatrix} \frac{\partial x}{\partial \dot{x}_0} & \frac{\partial x}{\partial \dot{y}_0} & \frac{\partial x}{\partial \dot{z}_0} \\ \frac{\partial y}{\partial \dot{x}_0} & \frac{\partial y}{\partial \dot{y}_0} & \frac{\partial y}{\partial \dot{z}_0} \\ \frac{\partial z}{\partial \dot{x}_0} & \frac{\partial z}{\partial \dot{y}_0} & \frac{\partial z}{\partial \dot{z}_0} \end{bmatrix} = \Phi_{rv} \tag{3.17}$$

Substituting specifics for this problem, the targeting iterative update equation is

$$\boldsymbol{v}^{i+1} = \boldsymbol{v}^{i} - \Phi_{rv}^{-1} \begin{bmatrix} x(t) - x_d \\ y(t) - y_d \\ z(t) - z_d \end{bmatrix}_i \tag{3.18}$$

$$\boldsymbol{v}^{i+1} = \boldsymbol{v}^{i} - \Phi_{rv}^{-1} \delta \boldsymbol{x}_r(t) \tag{3.19}$$

Finding an NRHO

A single shooting method can be utilized to find repeating orbits in the CR3BP. The following example reproduces the steps for finding a member of the L2 southern halo family of repeating orbits by Grebow [26]. In addition to this, Grebow details strategies for finding many families of repeating orbits [26] as well as initial guess strategies.

The design variables used leverage the symmetry of the repeating orbit. The L2 southern halo family is symmetric about the xz plane as shown in Figure 3.1. At the perilune crossing of the xz plane, the y position component is equal to zero while the x and z position are nonzero. Additionally, the trajectory crossing occurs perpendicular to the xz plane. As a result, the velocity vector is normal to the plane with $\dot{y}$ being the only nonzero component, also shown in Figure 3.1. The state vector at perilune is then:

$$\boldsymbol{x}_{perilune} = \begin{bmatrix} x & 0 & z & 0 & \dot{y} & 0 \end{bmatrix}^\top \tag{3.20}$$

Establishing an initial state for the trajectory guess at the xz plane crossing minimizes the number of state elements to be modified in the design vector to three (in addition to a variable integration time).

$$\boldsymbol{V} = \begin{bmatrix} x(t_1) & z(t_1) & \dot{y}(t_1) & \Delta t \end{bmatrix}^\top \tag{3.21}$$

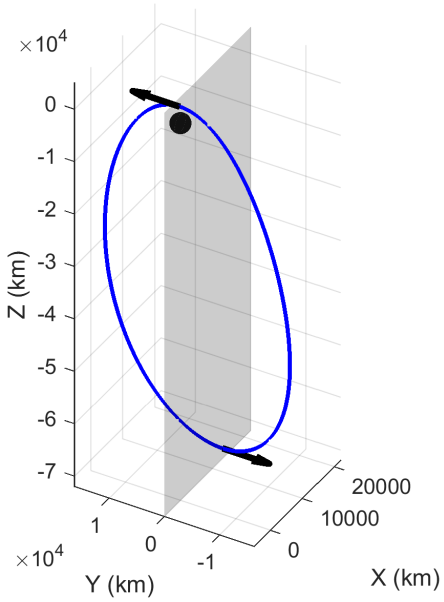


Fig. 3.1: NRHO Symmetry

Similarly, the constraint equations are comprised of the state elements that equal zero at apolune (y position, x and z velocity):

$$F(\boldsymbol{V}) = \begin{bmatrix} y(t_2) & \dot{x}(t_2) & \dot{z}(t_2) \end{bmatrix}^\top \tag{3.22}$$

In this manner, the design variables (Equation 3.21) are modified to find the subsequent perpendicular xz plane crossing Δt later. When satisfied, Δt corresponds to half of an orbital period.

The $D\boldsymbol{F}$ matrix is constructed from the appropriate STM matrix elements. If integration time or other quantities are design elements, they are appended to $D\boldsymbol{F}$.

$$
\begin{aligned}
D\boldsymbol{F}(\boldsymbol{V}) &= \begin{bmatrix} \Phi(t_2,t_1)_{2,1} & \Phi(t_2,t_1)_{2,3} & \Phi(t_2,t_1)_{2,5} & \dot{y}(t_2) \\ \Phi(t_2,t_1)_{4,1} & \Phi(t_2,t_1)_{4,3} & \Phi(t_2,t_1)_{4,5} & \ddot{x}(t_2) \\ \Phi(t_2,t_1)_{6,1} & \Phi(t_2,t_1)_{6,3} & \Phi(t_2,t_1)_{6,5} & \ddot{z}(t_2) \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial y(t_2)}{\partial x(t_1)} & \frac{\partial y(t_2)}{\partial z(t_1)} & \frac{\partial y(t_2)}{\partial \dot{y}(t_0)} & \frac{\partial y(t_2)}{\partial \Delta t} \\ \frac{\partial \dot{x}(t_2)}{\partial x(t_1)} & \frac{\partial \dot{x}(t_2)}{\partial z(t_1)} & \frac{\partial \dot{x}(t_2)}{\partial \dot{y}(t_1)} & \frac{\partial \dot{x}(t_2)}{\partial \Delta t} \\ \frac{\partial \dot{z}(t_2)}{\partial x(t_1)} & \frac{\partial \dot{z}(t_2)}{\partial z(t_1)} & \frac{\partial \dot{z}(t_2)}{\partial \dot{y}(t_1)} & \frac{\partial \dot{z}(t_2)}{\partial \Delta t} \end{bmatrix}
\end{aligned} \tag{3.23}
$$

where partial derivatives with respect to time are evaluations of the CR3BP equations of motion (Equation 2.25) at time t_2.

As a summary, the iteration steps are as follows: First, an initial guess is formulated by an initial state and coast duration ($\boldsymbol{X}(t_1)$ and Δt) which form the design variable vector $\boldsymbol{V}$. Next, the state and state transition matrix are propagated to $\boldsymbol{X}(t_2)$ and $\Phi(t_2,t_1)$. The constraint equation vector is constructed from elements at time t_2; in L2 NRHO example, $F(\boldsymbol{V})$ is constructed from the elements equal to zero at the perpendicular xz plane crossing at time t_2. $DF(\boldsymbol{X}^i)$ is constructed per Equation 3.14. $\boldsymbol{X}^1$ can then be calculated using Equation 3.13. The process is repeated until the constraint equation $F(\boldsymbol{X}^i)$ produces a sufficiently small norm.

3.1.3 Multiple Differential Correction

Multiple differential correction connects multiple single differential correction problem segments together via additional constraints. A more detailed general construction scheme is outlined in [24, 47]. This subsection shows the steps in constructing a three segment multiple differential correction scheme that allows two impulsive maneuvers and constrains an initial and target position. A design vector for this three segment scenario with a fixed

initial state is

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{x}_{0,1} \\ \Delta t_1 \\ \boldsymbol{x}_{0,2} \\ \Delta t_2 \\ \boldsymbol{x}_{0,3} \\ \Delta t_3 \end{bmatrix}_{21\times 1} \tag{3.24}$$

In this three segment example a constraint vector $\boldsymbol{F}$ can be constructed using the following guidelines:

- Segment 1: Constrained initial state $\boldsymbol{x}_{0,1}$ in initial orbit. Unconstrained propagation time Δt_1 for segment 1:

$$\boldsymbol{F}_1 = \left[\boldsymbol{x}_{0,1} - \boldsymbol{X}_{start}\right]_{6\times 1} \tag{3.25}$$

- Segment 2: Constrained position at the end of the first segment ($\boldsymbol{r}_{f,1}$) to equal the position at the beginning of the second segment ($\boldsymbol{r}_{0,2}$). The velocity between segments is left unconstrained in this example to incorporate a ΔV. Unconstrained propagation time Δt_2 for segment 2.

$$\boldsymbol{F}_2 = \left[\boldsymbol{r}_{0,2} - \boldsymbol{r}_{f,1}\right]_{3\times 1} \tag{3.26}$$

- Segment 3: Constrained position at the end of the second segment ($\boldsymbol{r}_{f,2}$) with position at the beginning of the third segment ($\boldsymbol{r}_{0,3}$) to allow a second ΔV. Constrained final state $\boldsymbol{x}_{0,3}$ with state $\boldsymbol{X}_{target}$.

$$\boldsymbol{F}_3 = \begin{bmatrix} \boldsymbol{r}_{0,3} - \boldsymbol{r}_{f,2} \\ \boldsymbol{x}_{0,3} - \boldsymbol{X}_{target} \end{bmatrix}_{9\times 1} \tag{3.27}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \end{bmatrix} \tag{3.28}$$

The construction of the derivative matrix follows the same format as Equation 3.14:

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{V}} = \begin{bmatrix} \frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{V}_1} & \frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{V}_2} & \frac{\partial \boldsymbol{F}_1}{\partial \boldsymbol{V}_3} \\ \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{V}_1} & \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{V}_2} & \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{V}_3} \\ \frac{\partial \boldsymbol{F}_3}{\partial \boldsymbol{V}_1} & \frac{\partial \boldsymbol{F}_3}{\partial \boldsymbol{V}_2} & \frac{\partial \boldsymbol{F}_3}{\partial \boldsymbol{V}_3} \end{bmatrix}_{18\times 21} \tag{3.29}$$

Individual partial derivatives are not complicated in this example with many populated by identity matrices and zeros. Two nonzero examples follow:

$$\frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{V}_1} = \begin{bmatrix} \frac{\partial(\boldsymbol{r}_{2initial} - \boldsymbol{r}_{1final})}{\partial \boldsymbol{X}_{1initial}} & \frac{\partial(\boldsymbol{r}_{2initial} - \boldsymbol{r}_{1final})}{\partial t_1} \end{bmatrix}_{3\times 7} = \begin{bmatrix} -\Phi_{1rr} & -\Phi_{1rv} & -\frac{\partial(\boldsymbol{r}_1(t_1))}{\partial t_1} \end{bmatrix} \tag{3.30}$$

$$\frac{\partial \boldsymbol{F}_3}{\partial \boldsymbol{V}_3} = \begin{bmatrix} \frac{\partial(\boldsymbol{r}_{3initial} - \boldsymbol{r}_{2final})}{\partial \boldsymbol{X}_{3initial}} & \frac{\partial(\boldsymbol{r}_{3initial} - \boldsymbol{r}_{2final})}{\partial t_3} \\ \frac{\partial(\boldsymbol{X}_{3final} - \boldsymbol{X}_{target})}{\partial \boldsymbol{X}_{3initial}} & \frac{\partial(\boldsymbol{X}_{3final} - \boldsymbol{X}_{target})}{\partial t_3} \end{bmatrix}_{9\times 7} = \begin{bmatrix} \boldsymbol{I}_{3\times 3} & \boldsymbol{0}_{3\times 4} \\ \Phi_3 & \frac{\partial(\boldsymbol{X}_3(t_3))}{\partial t_3} \end{bmatrix} \tag{3.31}$$

Once the partial derivatives are constructed, the iterative algorithm in Equation 3.13 is used until the constraints are sufficiently satisfied.

3.2 Multiple Segment Trajectory Optimization

As derived, differential correction creates a continuous trajectory that satisfies a set of constraints (i.e., a feasible trajectory). Differential correction does not optimize any cost related to the transfer scenario, however. While it is possible to create a creative constraint set that results in an optimal solution in some cases (collinear velocity and maneuver vectors to find a Hohmann transfer), as scenarios become increasingly complicated, the correct aspects of optimality may become difficult to implement via constraints. Given that differential correction is a common gradient-based method used to generate CR3BP trajectories, and the appropriate linear terms effectively inform steps toward a feasible solution, the upcoming section utilizes similar sensitivities to find a feasible solution and simultaneously optimize a deterministic objective function via nonlinear programming.

3.2.1 Problem Setup

A multiple segment trajectory is a method of modeling a trajectory as a set of discrete dynamics and constraints. Each trajectory segment formulation is a boundary value problem that is designed to satisfy its endpoint constraints. Assembling the multi-segment problem involves discretizing a trajectory into n segments separated by $n+1$ nodes that are not necessarily evenly spaced in time. Each trajectory segment is defined by the six-dimensional state at the beginning of segment i, $\boldsymbol{x}_{0,i}$ and the duration of the segment, Δt_i. The state at the end of segment i, $\boldsymbol{x}_{f,i}$ is a function of the natural motion dynamics of the system, the state at the beginning of the segment, and the duration Δt_i:

$$\boldsymbol{x}_{f,i} = \boldsymbol{f}\left(\boldsymbol{x}_{0,i}, \Delta t_i\right) \tag{3.32}$$

Each trajectory segment parameter vector $\boldsymbol{s}_i$ is defined by its initial state and duration

$$\boldsymbol{s}_i = \begin{bmatrix} \boldsymbol{r}_{0,i} \\ \boldsymbol{v}_{0,i} \\ \Delta t_i \end{bmatrix} \tag{3.33}$$

and are each assembled into a parameter vector defining the trajectory:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_n \end{bmatrix} \tag{3.34}$$

Enforcing natural motion (a coast) between segments i and $i+1$ involves constraining the final state of segment i with the initial state of segment $i+1$:

$$\boldsymbol{x}_{f,i} = \boldsymbol{x}_{0,i+1} \tag{3.35}$$

Similarly, an impulsive maneuver is allowed at the end of the first segment and beginning of the last segment by leaving the velocity elements unconstrained (or only constraining the

position elements):

$$\boldsymbol{r}_{f,1} = \boldsymbol{r}_{0,2} \tag{3.36}$$

$$\boldsymbol{r}_{0,n} = \boldsymbol{r}_{f,n-1} \tag{3.37}$$

The multiple-segment trajectory forms the basis of the following optimization problem definition. Each entry of the trajectory parameter vector $\boldsymbol{S}$ is an optimization parameter (Equation 3.38) that can be modified to minimize the total deterministic ΔV (Equation 3.39) while satisfying the constraints (Equations 3.40 through 3.45) and initial conditions (Equation 3.46).

$$\text{optimization parameters } \boldsymbol{S} \tag{3.38}$$

$$\text{minimize } J = \|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\| + \|\boldsymbol{v}_{0,n} - \boldsymbol{v}_{f,n-1}\| \tag{3.39}$$

$$\text{subject to } \boldsymbol{x}_{f,i} = \boldsymbol{f}\left(\boldsymbol{x}_{0,i}, \Delta t_i\right) \tag{3.40}$$

$$\boldsymbol{r}_{0,2} = \boldsymbol{r}_{f,1} \tag{3.41}$$

$$\boldsymbol{r}_{0,n} = \boldsymbol{r}_{f,n-1} \tag{3.42}$$

$$\boldsymbol{x}_{0,3} = \boldsymbol{x}_{f,2} \ldots \boldsymbol{x}_{0,n-1} = \boldsymbol{x}_{f,n-2} \tag{3.43}$$

$$\Delta t_i \geq 0 \tag{3.44}$$

$$\sum_i \Delta t_i = t_{total} \tag{3.45}$$

$$\text{given } \boldsymbol{x}_{0,1}, \boldsymbol{x}_{f,n}, t_{total} \tag{3.46}$$

The cost function in Equation 3.39 is the magnitude of two impulsive maneuvers. Equation 3.40 defines the system dynamics. Each segment forms an initial value problem; the final state $\boldsymbol{x}_{f,i}$ is a function of the initial state $\boldsymbol{x}_{0,i}$ and the segment duration Δt_i.

Equations 3.41 and 3.42 allow for an impulsive maneuver at the end of the first segment and the beginning of the final segment. Equation 3.43 enforces a coast (with no maneuvers) for the $n-2$ intermediate segments. Equation 3.44 ensures each segment is being propagated forward in time and Equation 3.45 enforces a fixed total time (when appropriate) for the

trajectory by constraining the sum of the segment durations to the total trajectory duration t_{total}. As the two-body problem and CR3BP are both time invariant, the absolute initial and final times are not required.

An equality constraint vector $\boldsymbol{C}_{eq}$ is formed from the equality constraint equations, and likewise for $\boldsymbol{C}_{in}$ comprised of the inequality constraint equations. As an example, consider a 3 segment problem setup. $\boldsymbol{S}$ is comprised of three segment parameter vectors, $\boldsymbol{s}_1$ through $\boldsymbol{s}_3$, for 21 total optimization parameters. $\boldsymbol{C}_{eq}$ contains 3 equations to enforce position continuity between segments 1 and 2 (Constraint Equation 3.41), 3 equations to enforce position continuity between segments 2 and 3 (Constraint Equation 3.42), 6 equations constraining the initial state, and 1 equation constraining the fixed total transfer time. The result is 21 optimization parameters, 13 of which are directly constrained. When the constraints are crossed out from the list of all optimization parameters (Equation 3.47), there are 8 of 21 free variables remaining, which correspond to two trajectory coast durations and two $\boldsymbol{\Delta V}$ vectors:

$$\boldsymbol{s}_1 = \begin{bmatrix} \cancel{\boldsymbol{r}_{0,1}} \\ \cancel{\boldsymbol{v}_{0,1}} \\ \Delta t_1 \end{bmatrix}, \quad \boldsymbol{s}_2 = \begin{bmatrix} \cancel{\boldsymbol{r}_{0,2}} \\ \boldsymbol{v}_{0,2} \\ \Delta t_2 \end{bmatrix}, \quad \boldsymbol{s}_3 = \begin{bmatrix} \cancel{\boldsymbol{r}_{0,3}} \\ \boldsymbol{v}_{0,3} \\ \cancel{\Delta t_3} \end{bmatrix} \tag{3.47}$$

Figure 3.2 shows a sample 3 segment trajectory with relevant features labeled. Subsequent trajectory figures follow this same format.

Numerical integration of the dynamics along each trajectory segment is used to obtain $\boldsymbol{x}_{f,i}$ given $\boldsymbol{x}_{0,i}$ and Δt_i. A variable-step variable-order Adams-Bashforth-Moulton predictor-corrector is implemented to solve each initial value problem [48]. It is also common to propagate the state transition matrix (STM) Φ^i terms alongside state elements for each trajectory segment. A gradient-based solver (Matlab's fmincon) utilizing the partial derivatives of the equality constraint equations, inequality constraint equations, and cost function with respect to the problem parameters ($\frac{\partial \boldsymbol{C}_{eq}}{\partial \boldsymbol{S}}$, $\frac{\partial \boldsymbol{C}_{in}}{\partial \boldsymbol{S}}$, and $\frac{\partial J}{\partial \boldsymbol{S}}$) informs the direction and size of a step toward a solution. These sensitivities can be computed numerically via the finite difference method or complex step differentiation, however the latter can be quite time consuming when the gradient matrix is large. Instead, the gradient function is de-

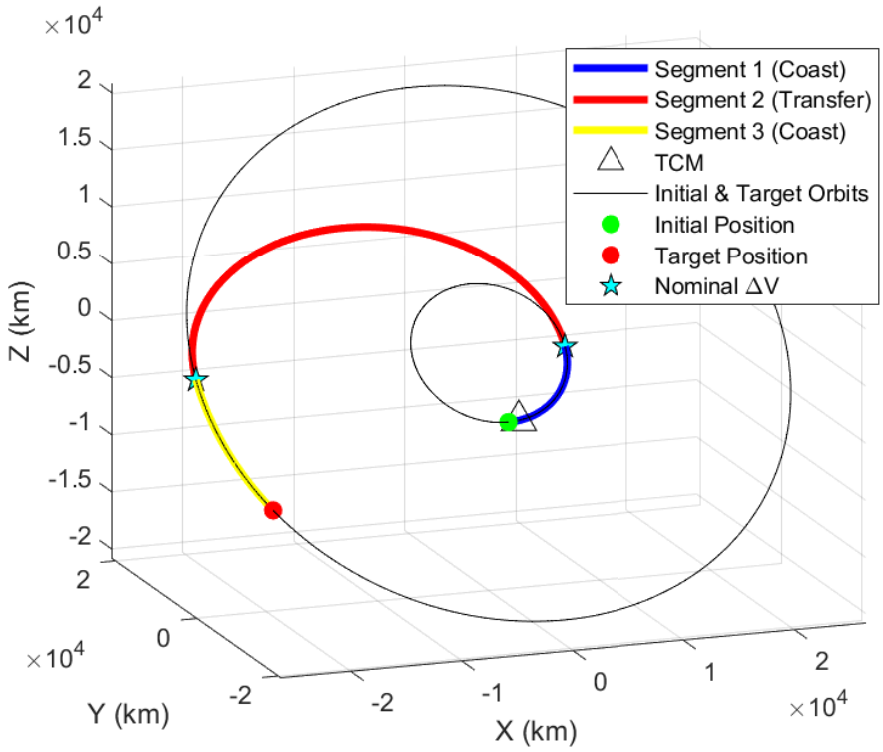


Fig. 3.2: Sample 3 Segment Trajectory

rived analytically and, in simpler cases, populated sparsely of ones, STM elements, and time derivatives. Cases with linear covariance-based stochastic cost or constraints (Chapter 4) require nontrivial derivation effort and may need to utilize second order STT terms. STM and STT terms are propagated via numerical integration alongside nonlinear state propagation.

3.2.2 Other Determinstic Constraint Options

The problem setup in Subsection 3.2.1 directly constrains initial and final elements of the state vector. While coast durations in the first and third segments enable flexibility in the specific departure and arrival orbits, effectively five of the six orbital elements are fixed with the true anomaly being eligible for optimization. This subsection explores other constraint options.

The first constraint fixes the two-body orbital energy

$$\epsilon = \frac{\left(v_{sc}^{I}\right)^{2}}{2} - \frac{\mu_{body}}{r_{sc/B}} \tag{3.48}$$

with respect to the planetary body B to be equal to a desired value. $r_{sc/B}$ is the magnitude of the position vector with respect to B, $\boldsymbol{r}_{sc/B}$, and

$$\boldsymbol{r}_{sc/B} = \boldsymbol{r}_{0,i} - \boldsymbol{r}_{B} \tag{3.49}$$

v_{sc}^{I} is the velocity of the spacecraft in a non-rotating frame. In this case, the velocity elements of the parameter vector are in the rotating frame and the rotating frame velocity needs to be included. Assuming the constraint is applied to the initial orbit:

$$\boldsymbol{v}_{sc/B}^{I} = \boldsymbol{v}_{0,1} + \omega_{S/I} \times \boldsymbol{r}_{0,1}$$

The constraint equation is

$$\epsilon_{initial} - \epsilon_{fixed} = 0 \tag{3.50}$$

where ϵ_{fixed} in this dissertation generally corresponds to an initial circular orbit of a desired altitude.

Equation 3.51 enforces a specific position vector magnitude with respect to B:

$$r_{sc/B} - r_{fixed_B} = 0 \tag{3.51}$$

To enforce orthogonal position and velocity vectors (an apse, or a circular orbit) the constraint is

$$\boldsymbol{r}_{sc/B}^{\top} \boldsymbol{v}_{sc}^{I} = 0 \tag{3.52}$$

In trajectory design problems that involve a planetary flyby, the optimal set of maneuvers may result in a trajectory that passes within the surface of a planetary body (or within a specified keep-out altitude). The inequality constraint Equation 3.53 enforces a minimum flyby distance to a specific planetary body to avoid a too-close approach. This constraint can be implemented multiple ways: First, a search of the entire trajectory for the minimum flyby distance along the trajectory can identify the location of periapse and at which point to apply the constraint. Another option is applying the constraint at the same location as an apse constraint, enforcing a specific trajectory node to be periapse with respect to a body and for it to be no closer than $d_{\text{min flyby}}$.

$$\|\boldsymbol{r}_{periapse}\| \geq d_{\text{min flyby}} \tag{3.53}$$

It is possible to constrain specific expenses of transfer trajectories such that they are required to be performed by a launch vehicle upper stage, for example. An implementation of this concept is forcing a plane change to occur with a specific nominal maneuver by constraining the orbital planes surrounding a second maneuver to be co-planar. In the following constraint formulation, a nominal maneuver is performed at the beginning of segment $i + 1$. In this manner, the unit vector orthogonal to the orbital plane during

segment i and $i+1$ are constrained to be collinear.

$$\left(\frac{(\boldsymbol{r}_{0,i} \times \boldsymbol{v}_{0,i})}{\|(\boldsymbol{r}_{0,i} \times \boldsymbol{v}_{0,i})\|} \bullet \frac{(\boldsymbol{r}_{f,i+1} \times \boldsymbol{v}_{f,i+1})}{\|(\boldsymbol{r}_{f,i+1} \times \boldsymbol{v}_{f,i+1})\|} = 1 \right) \tag{3.54}$$

It is also possible to constrain an initial or target orbit to a circular orbit of specific radius by implementing the combination of Constraint Equations 3.50, 3.51, and 3.52. In this manner, the orbit is constrained to be circular and a specific size but the orbital plane (inclination and right ascension) and true anomaly are free to be optimized.

The deterministic trajectory design methods described in this chapter are commonly utilized to develop feasible connected trajectories in the CR3BP. Depending on the initial guess, the optimization method described converges quickly to a locally optimal trajectory, offering improvement from a feasible trajectory solution. In both cases, creative implementation of constraint equations and problem segmentation benefits convergence.

CHAPTER 4

TWO-BODY TRAJECTORIES ROBUST TO INITIAL DISPERSION

4.1 Solution Method

The following solution method presents an incremental approach to developing a trajectory robust to initial state dispersion. Problem 1 optimizes two impulsive maneuvers to find a deterministic optimal trajectory. Next, stochastics are introduced via an initial state dispersion; a trajectory is sought with optimized nominal maneuvers only that constrains the final position dispersion at a target state (Problem 2). In Problem 3, two trajectory correction maneuvers (TCM) are introduced along the deterministic optimal trajectory from Problem 1; the first TCM targets zero final position dispersion and the second TCM corrects remaining velocity dispersion once the target position is achieved. Finally, the robust trajectory is found by modifying the nominal maneuvers alongside the TCMs, resulting in the lowest statistical upper bound for the total ΔV and a trajectory robust to initial state dispersion (Problem 4). Figure 4.1 shows the four problems analyzed for each scenario, progressing from deterministic optimal to robust, and their key features.[1]

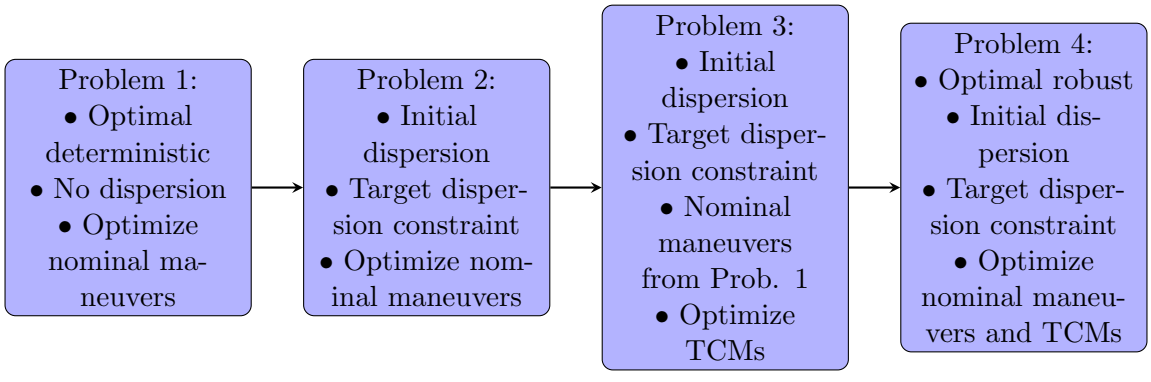


Fig. 4.1: Problem progression flowchart

[1]This chapter's content was published as [35] in the Journal of Spacecraft and Rockets.

4.1.1 Optimal Deterministic Trajectory

Problem 1 is an n segment deterministic optimization problem following the form of Equations 3.38 through 3.46. The resulting equality constraint vector is

$$\boldsymbol{C}_{eq} = \begin{bmatrix} \boldsymbol{x}_{0,1} - \boldsymbol{X}_0 \\ \boldsymbol{r}_{0,2} - \boldsymbol{r}_{f,1} \\ \boldsymbol{x}_{0,3} - \boldsymbol{x}_{f,2} \\ \vdots \\ \boldsymbol{x}_{0,n-1} - \boldsymbol{x}_{f,n-2} \\ \boldsymbol{r}_{0,n} - \boldsymbol{r}_{f,n-1} \\ \boldsymbol{x}_{f,n} - \boldsymbol{X}_t \\ \sum \Delta t_i - t_{final} \end{bmatrix} = \mathbf{0} \tag{4.1}$$

The inequality constraint vector is

$$\boldsymbol{C}_{in} = \begin{bmatrix} -\Delta t_1 \\ \vdots \\ -\Delta t_n \end{bmatrix} \leq \mathbf{0} \tag{4.2}$$

and the cost function is the sum of two nominal impulsive maneuvers

$$J = \Delta V_1 + \Delta V_2 = \|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\| + \|\boldsymbol{v}_{0,n} - \boldsymbol{v}_{f,n-1}\| \tag{4.3}$$

The equality constraint gradient matrix $\frac{\partial \boldsymbol{C}_{eq}}{\partial S}$ is not derived in its entirety in this chapter; rather, examples are presented that encompass all of the major required steps. As a first example, a single submatrix of the constraint gradient reveals that the STM directly corresponds to the sensitivity of a future state ($\boldsymbol{x}_{f,2}$) to changes in an initial state ($\boldsymbol{x}_{0,2}$). This example also demonstrates the sensitivity of a segment final state with respect to the segment duration is the direct evaluation of the system equations of motion at the final state. Depending on specific conventions and specific nonlinear programming (NLP) input

requirements, the correct gradient format may actually be the transpose of what is derived in this chapter.

$$\frac{\partial \boldsymbol{C}_{eq,3}}{\partial \boldsymbol{s}_2} = \begin{bmatrix} \frac{\partial(\boldsymbol{x}_{0,3}-\boldsymbol{x}_{f,2})}{\partial \boldsymbol{x}_{0,2}}, & \frac{\partial(\boldsymbol{x}_{0,3}-\boldsymbol{x}_{f,2})}{\partial \Delta t_2} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \boldsymbol{x}_{f,2}}{\partial \boldsymbol{x}_{0,2}}, & -\frac{\partial \boldsymbol{x}_{f,2}}{\partial \Delta t_2} \end{bmatrix} = \begin{bmatrix} -\Phi(t_{f,2}, t_{0,2}), & -\dot{\boldsymbol{x}}_{f,2} \end{bmatrix} \tag{4.4}$$

The sensitivity of the same constraint equations with respect to $\boldsymbol{x}_{0,3}$ is the identity matrix:

$$\frac{\partial \boldsymbol{C}_{eq,3}}{\partial \boldsymbol{x}_{0,3}} = \frac{\partial\left(\boldsymbol{x}_{0,3} - \boldsymbol{x}_{f,2}\right)}{\partial\left(\boldsymbol{x}_{0,3}\right)} = I_{6\times 6} \tag{4.5}$$

The inequality constraint gradient is trivially derived and populated with ones and zeros.

$$\frac{\partial\left(-\Delta t_1\right)}{\partial \boldsymbol{s}_1} = \begin{bmatrix} \frac{-\partial \Delta t_1}{\partial \boldsymbol{x}_{0,1}}, & \frac{-\partial \Delta t_1}{\partial \Delta t_1} \end{bmatrix} = \begin{bmatrix} 0_{1\times 6} & -1 \end{bmatrix} \tag{4.6}$$

The cost function gradient incorporates an identity for the partial derivative of a vector magnitude with respect to itself:

$$\frac{\partial J}{\partial \boldsymbol{s}_1} = \begin{bmatrix} \frac{\partial \|\boldsymbol{v}_{0,2}-\boldsymbol{v}_{f,1}\|}{\partial \boldsymbol{x}_{0,1}}, & \frac{\partial \|\boldsymbol{v}_{0,2}-\boldsymbol{v}_{f,1}\|}{\partial \Delta t_1} \end{bmatrix} \tag{4.7}$$

$$\begin{aligned} \frac{\partial \|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|}{\partial \boldsymbol{x}_{0,1}} &= \frac{\partial \|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|}{\partial\left(\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\right)} \frac{\partial\left(\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\right)}{\partial \boldsymbol{x}_{0,1}} \\ &= -\frac{\left(\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\right)^\top}{\|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|} \frac{\partial \boldsymbol{v}_{f,1}}{\partial \boldsymbol{x}_{0,1}} = -\frac{\left(\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\right)^\top}{\|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|} \begin{bmatrix} \Phi^1_{vr} & \Phi^1_{vv} \end{bmatrix} \end{aligned} \tag{4.8}$$

where Φ^1_{vr} and Φ^1_{vv} are the corresponding 3×3 submatrices of the STM for segment 1, Φ^1 (e.g. rows 4-6 and columns 1-3 for Φ^1_{vr}).

$$\frac{\partial \|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|}{\partial \Delta t_1} = -\frac{\left(\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\right)^\top}{\|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|} \frac{\partial \boldsymbol{v}_{f,1}}{\partial \Delta t_1} = -\frac{\left(\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\right)^\top}{\|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\|} \dot{\boldsymbol{v}}_{f,1} \tag{4.9}$$

This solution method and derivation matches a similar problem setup by Whitley and Ocampo [10]. Using these analytical gradients, the solution to Problem 1 as described by Equations 3.38 to 3.46 is achieved using an NLP.

4.1.2 Initial State Dispersion with Target Position Dispersion Constraint

The second problem introduces stochastics into the optimization problem in the form of a normally distributed initial state dispersion $\delta \boldsymbol{x}_0$.

$$\delta \boldsymbol{x}_0 = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \dot{\delta x} \\ \dot{\delta y} \\ \dot{\delta z} \end{bmatrix} \tag{4.10}$$

The initial dispersion elements are modeled as zero-mean, uncorrelated, normally distributed random variables with equal variance in each direction (position dispersion distribution $\delta r_0 \sim \mathcal{N}\left(0, \sigma_r^2\right)$ and velocity dispersion distribution $\delta v_0 \sim \mathcal{N}\left(0, \sigma_v^2\right)$). The resulting dispersion covariance at the beginning of segment 1 (P_1) is

$$P_1 = \begin{bmatrix} \sigma_r^2 I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \sigma_v^2 I_{3\times 3} \end{bmatrix} \tag{4.11}$$

The state dispersion at node i (the beginning of segment i) is represented by a dispersion covariance matrix P_i. The segment STM Φ^i linearly propagates the dispersion covariance to the beginning of the subsequent segment and preserves the Gaussian distribution [49]. The transfer trajectory is still restricted to two impulsive maneuvers but a target position dispersion inequality constraint is introduced. P_n represents the target dispersion covariance at node n (the beginning of the final segment), which is roughly equivalent to arriving to the target state early (segment n is a coast segment). Another common choice may be constraining the terminal position covariance (P_{n+1}).

With the exception of the introduction of the new target position dispersion constraint, the remainder of the optimization problem is unchanged from Problem 1 (the optimization parameters $\boldsymbol{S}$, the cost function, and the remainder of the constraints are all the same). Formally, the optimization problem is

$$\text{optimization parameters } \boldsymbol{S} \tag{4.12}$$

$$\text{minimize } J = \sum \Delta V = \|\boldsymbol{v}_{0,2} - \boldsymbol{v}_{f,1}\| + \|\boldsymbol{v}_{0,n} - \boldsymbol{v}_{f,n-1}\| \tag{4.13}$$

$$\text{subject to } \boldsymbol{x}_{f,i} = f\left(\boldsymbol{x}_{0,i}, \Delta t_i\right) \tag{4.14}$$

$$P_{i+1} = \Phi^i P_i \Phi^{i\top} \tag{4.15}$$

$$\boldsymbol{r}_{0,2} = \boldsymbol{r}_{f,1} \tag{4.16}$$

$$\boldsymbol{r}_{0,n} = \boldsymbol{r}_{f,n-1} \tag{4.17}$$

$$\boldsymbol{x}_{0,3} = \boldsymbol{x}_{f,2} \ldots \boldsymbol{x}_{0,n-1} = \boldsymbol{x}_{f,n-2} \tag{4.18}$$

$$\Delta t_i \geq 0 \tag{4.19}$$

$$\sum \Delta t_i = t_{final} \tag{4.20}$$

$$\text{tr}(M_r P_n M_r^\top) \leq \sigma_{r,max}^2 \tag{4.21}$$

$$\text{given } \boldsymbol{x}_{0,1}, \boldsymbol{x}_{f,n}, t_{final}, P_1 \tag{4.22}$$

Equation 4.15 introduces the dispersion covariance propagation (with no process noise or disturbances) and Equation 4.21 introduces the new target position dispersion covariance inequality constraint. M_r is a mapping matrix to the position covariance submatrix (the mapping matrix can be generalized for any target constraint) and tr() represents the trace operator which yields the sum of the position dispersion variances:

$$M_r = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \end{bmatrix} \tag{4.23}$$

The dispersion covariance inequality constraint increases the complexity of the analytical inequality constraint gradient, in derivation effort and in the quantity of terms propagated numerically. Deriving the gradient requires the following identity which is derived in

Appendix A.1.

$$\frac{\partial \text{tr}\left(CABA^\top C^\top\right)}{\partial \boldsymbol{x}} = 2\begin{bmatrix} \text{tr}\left[C^\top CAB\frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \text{tr}\left[C^\top CAB\frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top \tag{4.24}$$

The identity in Equation 4.24 assumes the matrix A is a function of the vector $\boldsymbol{x}$ and the matrices B and C are not. Generally, the identity applies when the matrix A is dimension $n \times m$, B is $m \times m$ and symmetric, and C is $p \times n$.

Applying the identity in Equation 4.24 yields the following where the target covariance is given by $P_n = \Phi^{(n-1):1} P_1 \Phi^{(n-1):1\top}$ and the sum of the position dispersion variances is $\text{tr}(M_r \Phi^{(n-1):1} P_1 \Phi^{(n-1):1\top} M_r^\top)$.

$$\frac{\partial \text{tr}\left(M_r\Phi^{(n-1):1}P_1\Phi^{(n-1):1\top}M_r^\top\right)}{\partial \boldsymbol{x}_{0,i}} = 2\begin{bmatrix} \text{tr}\left(M_r^\top M_r\left(\Phi^{(n-1):1}P_1\right)\frac{\partial \Phi^{(n-1):1}}{\partial x_{0,i}}^\top\right) \\ \text{tr}\left(M_r^\top M_r\left(\Phi^{(n-1):1}P_1\right)\frac{\partial \Phi^{(n-1):1}}{\partial y_{0,i}}^\top\right) \\ \text{tr}\left(M_r^\top M_r\left(\Phi^{(n-1):1}P_1\right)\frac{\partial \Phi^{(n-1):1}}{\partial z_{0,i}}^\top\right) \\ \text{tr}\left(M_r^\top M_r\left(\Phi^{(n-1):1}P_1\right)\frac{\partial \Phi^{(n-1):1}}{\partial \dot{x}_{0,i}}^\top\right) \\ \text{tr}\left(M_r^\top M_r\left(\Phi^{(n-1):1}P_1\right)\frac{\partial \Phi^{(n-1):1}}{\partial \dot{y}_{0,i}}^\top\right) \\ \text{tr}\left(M_r^\top M_r\left(\Phi^{(n-1):1}P_1\right)\frac{\partial \Phi^{(n-1):1}}{\partial \dot{z}_{0,i}}^\top\right) \end{bmatrix}^\top \tag{4.25}$$

where $\frac{\partial \Phi^{(n-1):1}}{\partial \boldsymbol{x}_{0,i}}$ is a combination of first-order STMs and second-order STTs from the beginning of segment i to the end of segment $n-1$. As each segment is a separate boundary value problem, the STMs for segments 2 through $n-1$ do not vary with changes in $\boldsymbol{x}_{0,1}$ when $i = 1$, for example. Therefore, they are treated as matrices of constants when evaluating $\frac{\partial \Phi^{(n-1):1}}{\partial \boldsymbol{x}_{0,1}}$:

$$\frac{\partial \Phi^{(n-1):1}}{\partial \boldsymbol{x}_{0,1}} = \frac{\partial}{\partial \boldsymbol{x}_{0,1}}\left(\Phi^{(n-1):2}\Phi^1\right) = \Phi^{(n-1):2}\frac{\partial \Phi^1}{\partial \boldsymbol{x}_{0,1}} = \Phi^{(n-1):2}\Phi^1_{II} \tag{4.26}$$

Partial derivatives of the STM with respect to segment duration represent the instantaneous time rate of change of the STM at the final node of a segment. Thus, they are a direct evaluation of the time rate of change equations for the state transition matrix

$\dot{\Phi} = F\Phi$ about $\boldsymbol{x}_{f,i}$ and Φ^i.

$$\frac{\partial \text{tr}\left(M_r \Phi^{(n-1):1} P_1 \Phi^{(n-1):1\top} M_r^\top\right)}{\partial \Delta t_i} = 2\text{tr}\left(M_r^\top M_r \left(\Phi^{(n-1):1} P_1\right) \frac{\partial \Phi^{(n-1):1}}{\partial \Delta t_i}^\top\right) \tag{4.27}$$

As an example, when $i = 1$, $\frac{\partial \Phi^{(n-1):1}}{\partial \Delta t_1}$ is

$$\frac{\partial \Phi^{(n-1):1}}{\partial \Delta t_1} = \Phi^{(n-1):2} \frac{\partial \Phi^1}{\partial \Delta t_1} = \Phi^{(n-1):2} F\left(\boldsymbol{x}_{f,1}\right) \Phi^1 \tag{4.28}$$

Using these analytical gradients, the solution to Problem 2 as described by Equations 4.12 to 4.22 can be solved using an NLP.

4.1.3 Optimal 3σ Trajectory Correction Maneuver Along a Nominal Trajectory

Problem 3 incorporates a TCM which attempts to minimize the position dispersion at the target state. This models a common trajectory design method: a deterministic optimal trajectory is developed and embarked upon. However, sources of error result in deviations from the optimal path and TCM(s) must be performed along the trajectory in order to successfully arrive at the desired state. The magnitude of each TCM varies based on its execution time throughout the trajectory and can be incorporated as part of a cost function. In addition to the time of the correction, the TCM is also a function of the initial state dispersion $\delta\boldsymbol{x}_0$. The quantity minimized along a single trajectory is the root sum of the squares (RSS) of the TCM covariance ($\sigma_{\delta V}$), subsequently defined in Equations 4.35 and 4.41. The RSS of a vector's covariance is independent of rotation and provides a statistical bound for the corresponding vector's magnitude, given by

$$\sigma_{RSS} = \sqrt{\text{tr}\left(P\right)} \tag{4.29}$$

Two TCMs are required to remove all six components of the dispersion at the target ($\delta\boldsymbol{x}(t_n) = 0_{6\times 1}$). The first TCM, $\delta\boldsymbol{V}_r$, performed at time t_c along a reference trajectory

targets zero position dispersion at the target state at a future time t_n ($\delta\boldsymbol{r}(t_n) = 0_{3\times1}$). The second TCM, $\delta\boldsymbol{V}_v$, corrects the remaining velocity dispersion once the target position is achieved. In terms of the multi-segment trajectory design architecture, the chosen target is node n ($\boldsymbol{x}_{0,n}$, the beginning of the final segment). The remainder of the final segment will be a coast in the target state with zero remaining dispersion. In this manner, t_c can occur anywhere along the reference trajectory prior to the target, $t_{0,1} \leq t_c < t_{0,n}$.

Position Dispersion Correction

If the dispersion magnitude at the time of the correction ($\delta\boldsymbol{x}(t_c)$) is relatively small, linear targeting techniques can be implemented to estimate the expected TCM magnitude. In general, the dispersion at time t_n given a dispersion at t_c is estimated by

$$\delta\boldsymbol{x}(t_n) = \Phi(t_n, t_c)\delta\boldsymbol{x}(t_c) = \Phi^{n_0:c}\delta\boldsymbol{x}(t_c) \tag{4.30}$$

where t_c is not required to occur at a segment intersection.

With perfect state knowledge at time t_c, the position dispersion correction impulse $\delta\boldsymbol{V}_r$ can be solved using the following equation:

$$\delta\boldsymbol{r}(t_n) = \boldsymbol{0} = \Phi_{rr}(t_{0,n}, t_c)\delta\boldsymbol{r}(t_c) + \Phi_{rv}(t_{0,n}, t_c)\left(\delta\boldsymbol{v}(t_c) + \delta\boldsymbol{V}_r\right) \tag{4.31}$$

Solving for $\delta\boldsymbol{V}_r$ to produce zero position dispersion at the target yields

$$\delta\boldsymbol{V}_r = -\Phi_{rv}^{n_0:c^{-1}}\Phi_{rr}^{n_0:c}\delta\boldsymbol{x}_r(t_c) - \delta\boldsymbol{x}_v(t_c) = \begin{bmatrix} -\Phi_{rv}^{n_0:c^{-1}}\Phi_{rr}^{n_0:c} & -I_{3\times3} \end{bmatrix}\delta\boldsymbol{x}(t_c) = T\delta\boldsymbol{x}(t_c) \tag{4.32}$$

where the intermediate matrix T is a function of $\Phi^{n_0:c} = \Phi\left(t_{0,n}, t_c\right)$:

$$T = \begin{bmatrix} -\Phi_{rv}^{n_0:c^{-1}}\Phi_{rr}^{n_0:c} & -I_{3\times3} \end{bmatrix} \tag{4.33}$$

The covariance of the position dispersion TCM, $\delta \boldsymbol{V}_r$, is

$$E\left[\delta \boldsymbol{V}_r \delta \boldsymbol{V}_r^\top\right] = E\left[\left(T\delta \boldsymbol{x}(t_c)\right)\left(T\delta \boldsymbol{x}(t_c)\right)^\top\right] = TE\left[\delta \boldsymbol{x}\delta \boldsymbol{x}^\top\right]T^\top = TP_cT^\top \tag{4.34}$$

and the RSS of the position dispersion TCM is

$$\sigma_{\delta V_r} = \sqrt{\mathrm{tr}\left(TP_iT^\top\right)} = \sqrt{\mathrm{tr}\left(T\Phi(t_c,t_1)P_1\Phi(t_c,t_1)^\top T^\top\right)} \tag{4.35}$$

While the TCM does not modify the nominal trajectory (the mean of the TCM is zero), it does modify the dispersion covariance. The post-correction state dispersion is

$$\delta \boldsymbol{x}_c^{+C} = \delta \boldsymbol{x}_c^{-C} + \begin{bmatrix} 0_{3\times 6} \\ T \end{bmatrix} \delta \boldsymbol{x}_c^{-C} \tag{4.36}$$

and the post-correction dispersion covariance is

$$P_c^{+C} = E\left[\left(\delta \boldsymbol{x}_c^{-C} + N\delta \boldsymbol{x}_c^{-C}\right)\left(\delta \boldsymbol{x}_c^{-C} + N\delta \boldsymbol{x}_c^{-C}\right)^\top\right] = \left(I+N\right)P_c^{-C}\left(I+N\right)^\top \tag{4.37}$$

where $N = \begin{bmatrix} 0_{3\times 6} \\ T \end{bmatrix}$ and I is the identity matrix.

Velocity Dispersion Correction

Upon arrival to the target state at t_n, the position dispersion $\delta \boldsymbol{r}(t_n)$ has been corrected. However, velocity dispersion $\delta \boldsymbol{v}(t_n)$ still exists that needs to be corrected with a second TCM, $\delta \boldsymbol{V}_v$. The velocity dispersion at t_n is

$$\delta \boldsymbol{v}(t_n) = \Phi_{vr}(t_n,t_c)\delta \boldsymbol{r}(t_c) + \Phi_{vv}(t_n,t_c)\left(\delta \boldsymbol{v}(t_c) + \delta \boldsymbol{V}_r\right) \tag{4.38}$$

Substituting the previous expression for $\delta \boldsymbol{V}_r$, introducing two intermediate matrices $W = \begin{bmatrix} \Phi_{vv} & -\Phi_{vr} \end{bmatrix}$, $L = \begin{bmatrix} WT^\top & 0_{3\times 3} \end{bmatrix}$, and simplifying results in

$$\delta\boldsymbol{v}(t_n) = \begin{pmatrix} \Phi_{vr}^{n_0:c} - \Phi_{vv}^{n_0:c}\Phi_{rv}^{n_0:c^{-1}}\Phi_{rr}^{n_0:c} & 0_{3\times3} \end{pmatrix} \delta\boldsymbol{x}(t_c) = \begin{bmatrix} WT^\top & 0_{3\times3} \end{bmatrix} \delta\boldsymbol{x}(t_c) = L\delta\boldsymbol{x}(t_c) \tag{4.39}$$

The variance of the final velocity dispersion $\delta\boldsymbol{v}(t_n)$ is

$$E\left[\delta\boldsymbol{v}(t_n)\delta\boldsymbol{v}(t_n)^\top\right] = E\left[L\delta\boldsymbol{x}(t_c)\delta\boldsymbol{x}(t_c)^\top L^\top\right] = LP_cL^\top \tag{4.40}$$

and the RSS of the final velocity dispersion is

$$\sigma_{\delta\boldsymbol{v}(t_n)} = \sqrt{\mathrm{tr}\left(LP_iL^\top\right)} = \sqrt{\mathrm{tr}\left(L\Phi(t_c,t_1)P_1\Phi(t_c,t_1)^\top L^\top\right)} \tag{4.41}$$

which is corrected directly via the second TCM, $\sigma_{\delta V_v} = \sigma_{\delta\boldsymbol{v}(t_n)}$.

TCM Optimization

With the intent to use a specific deterministic trajectory as the nominal trajectory there is no need to incorporate nominal ΔV as part of the cost function. Additionally, without any additional sources of error or uncertainty (only an initial state dispersion), the TCM will result in zero position dispersion at the target. Thus, the position dispersion covariance constraint becomes unnecessary for this problem.

A new optimization variable t_c is introduced which is currently the only free optimization parameter. The optimization problem becomes

$$\text{optimization parameter } t_c \tag{4.42}$$

$$\text{minimize } J = \sigma_{\delta V} = \sigma_{\delta V_r} + \sigma_{\delta V_v} \tag{4.43}$$

$$\text{subject to } P_{i+1} = \Phi^i P_i \Phi^{i\top} \tag{4.44}$$

$$P_i^{+C} = (I+N)\,P_i^{-C}\,(I+N)^\top \tag{4.45}$$

$$t_{0,1} \le t_c < t_{0,n} \tag{4.46}$$

$$\text{given } \boldsymbol{S}_{DO}, P_1 \tag{4.47}$$

where $\boldsymbol{S}_{DO}$ represents the deterministic optimal trajectory parameter vector.

Direct optimization of t_c outside of the NLP is more efficient and avoids the pitfalls of multiple local minima. Propagating and storing the STM time history along the nominal trajectory, $\Phi(t, t_0)$, allows for the rapid calculation of $\sigma_{\delta V}$ via the following steps at each time increment chosen along the trajectory (the automatically chosen nonlinear integration time increments are utilized in the current implementation):

1. Calculate $\Phi\left(t_{0,n}, t_c\right) = \Phi\left(t_{0,n}, t_1\right)\Phi\left(t_1, t_c\right)$
   (a) From $\Phi\left(t_c, t_1\right)$, invert using the symplectic unit matrix to compute $\Phi\left(t_1, t_c\right)$ [50].
2. Compute
   (a) $T = \begin{bmatrix} -\Phi_{rv}^{n_0:c^{-1}}\Phi_{rr}^{n_0:c} & -I_{3\times 3} \end{bmatrix}$
   (b) $P_{t_c} = T\Phi(t_c, t_1)P_1\Phi(t_c, t_1)^\top T^\top$
   (c) $\sigma_{\delta V_r} = \sqrt{\mathrm{tr}\left(P_{t_c}\right)}$
3. Compute
   (a) W, L, $P_{t_f} = L\Phi(t_c, t_1)P_1\Phi(t_c, t_1)^\top L^\top$
   (b) $\sigma_{\delta V_v} = \sqrt{\mathrm{tr}\ \left(P_{t_f}\right)}$
4. Compute total TCM RSS $\sigma_{\delta V} = \sigma_{\delta V_r} + \sigma_{\delta V_v}$ at each time increment

Once the TCM RSS is computed for each time increment, the solution to Problem 3 is a direct search for minimum value and the corresponding correction time t_c yields the minimum cost correction time to perform the first TCM ($\delta \boldsymbol{V}_r$) along the nominal trajectory.

4.1.4 Robust Trajectory, Optimal Nominal ΔV and 3σ TCM

The fourth problem combines various aspects of the previous problems to develop an optimal trajectory that is robust to initial state dispersion. In this problem, the impulsive nominal maneuvers are reintroduced into the optimization problem alongside the minimum 3σ TCM pair. In this manner, the nominal trajectory may be modified at the cost of increasing the nominal ΔV in order to reduce the total $\Delta V + 3\sigma_{\delta V}$, nominal plus TCM δV.

It is worth addressing the nature of the two cost function terms. $\sum \Delta V$ represents a deterministic path cost while $3\sigma_{\delta V}$ represents a statistical upper bound for the expected TCM magnitude. In reality, while a spacecraft mission's actual TCM cost may not reach the

3σ value, it is being considered as a planning factor in this analysis that directly correlates to the amount of on-board propellant a mission may carry. In general, the appropriate σ scaling factor may be flexible based on a specific mission's risk posture and is easily modified in the problem formulation should a different planning factor be appropriate.

The aspect of modifying the nominal trajectory to influence the target position dispersion is similar to the Problem 2. The multi-segment trajectory parameter vector $\boldsymbol{S}$ and the TCM execution time t_c comprise the optimization parameters. The target position dispersion constraint is unnecessary in this case as there are no random disturbances (process noise) or maneuver execution error; as a result, the TCM completely mitigates the target position dispersion.

The optimization problem is

$$\text{optimization parameters } \boldsymbol{S}, t_c \tag{4.48}$$

$$\text{minimize } J = \sum \Delta V + 3\sigma_{\delta V} \tag{4.49}$$

$$\text{subject to } \boldsymbol{x}_{f,i} = f\left(\boldsymbol{x}_{0,i}, \Delta t_i\right) \tag{4.50}$$

$$P_{i+1} = \Phi^i P_i \Phi^{i\top} \tag{4.51}$$

$$P_i^{+C} = (I+N)\, P_i^{-C}\, (I+N)^\top \tag{4.52}$$

$$\boldsymbol{r}_{0,2} = \boldsymbol{r}_{f,1} \tag{4.53}$$

$$\boldsymbol{r}_{0,n} = \boldsymbol{r}_{f,n-1} \tag{4.54}$$

$$\boldsymbol{x}_{0,3} = \boldsymbol{x}_{f,2} \ldots \boldsymbol{x}_{0,n-1} = \boldsymbol{x}_{f,n-2} \tag{4.55}$$

$$t_{0,1} \leq t_c < t_{0,n} \tag{4.56}$$

$$\Delta t_i \geq 0 \tag{4.57}$$

$$\sum \Delta t_i = t_{final} \tag{4.58}$$

$$\text{given } \boldsymbol{x}_{0,1}, \boldsymbol{x}_{f,n}, t_{final}, P_1 \tag{4.59}$$

Deriving the analytical gradients of the $3\sigma_{\delta V}$ magnitude (Equation 4.60) requires a modification to the previous identity in Equation 4.24 such that the matrix C is also a

function of $\boldsymbol{x}$ (presented in Appendix A.1). Equation 4.61 shows the updated identity, applied subsequently in Equations 4.62 and 4.63.

$$3\sigma_{\delta V} = 3\left(\sqrt{\text{tr}\left(T\Phi(t_c,t_1)P_1\Phi(t_c,t_1)^\top T^\top\right)} + \sqrt{\text{tr}\left(L\Phi(t_c,t_1)P_1\Phi(t_c,t_1)^\top L^\top\right)}\right) \tag{4.60}$$

$$\text{tr}\frac{\partial CABA^\top C^\top}{\partial \boldsymbol{x}} = 2\begin{bmatrix} \text{tr}\left[CABA^\top \frac{\partial C}{\partial x_1}^\top\right] + \text{tr}\left[C^\top CAB\frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \text{tr}\left[CABA^\top \frac{\partial C}{\partial x_n}^\top\right] + \text{tr}\left[C^\top CAB\frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top \tag{4.61}$$

An abbreviated derivation is presented for the TCM partial derivatives. Implementing the identity in Equation 4.61 results in the following partial derivative of the position dispersion TCM squared ($\sigma^2_{\delta V_r}$) with respect to the initial state of segment i, $\boldsymbol{x}_{0,i}$ and the duration of segment i, Δt_i. $\frac{\partial \sigma^2_{\delta V_v}}{\partial \boldsymbol{x}_{0,i}}$ and $\frac{\partial \sigma^2_{\delta V_v}}{\partial \Delta t_i}$ follow the same form as Equations 4.62 and 4.63 with the L matrix in place of T.

$$\frac{\partial \sigma^2_{\delta V_r}}{\partial \boldsymbol{x}_{0,i}} = 2\begin{bmatrix} \text{tr}\left[T\Phi^{c:1}P_1\Phi^{c:1\top}\frac{\partial T}{\partial x_{0,i}}^\top\right] + \text{tr}\left[T^\top T\Phi^{c:1}P_1\frac{\partial \Phi^{c:1}}{\partial x_{0,i}}^\top\right] \\ \vdots \\ \text{tr}\left[T\Phi^{c:1}P_1\Phi^{c:1\top}\frac{\partial T}{\partial \dot{z}_{0,i}}^\top\right] + \text{tr}\left[T^\top T\Phi^{c:1}P_1\frac{\partial \Phi^{c:1}}{\partial \dot{z}_{0,i}}^\top\right] \end{bmatrix}^\top \tag{4.62}$$

$$\frac{\partial \sigma^2_{\delta V_r}}{\partial \Delta t_i} = 2\begin{bmatrix} \text{tr}\left[T\Phi^{c:1}P_1\Phi^{c:1\top}\frac{\partial T}{\partial \Delta t_i}^\top\right] + \text{tr}\left[T^\top T\Phi^{c:1}P_1\frac{\partial \Phi^{c:1}}{\partial \Delta t_i}^\top\right] \end{bmatrix}^\top \tag{4.63}$$

$$\frac{\partial T}{\partial \boldsymbol{x}_{0,i}} = \begin{bmatrix} \Phi^{n_0:c^{-1}}_{rv}\left[\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}\right]_{rv}\Phi^{n_0:c^{-1}}_{rv}\Phi^{n_0:c}_{rr} - \Phi^{n_0:c^{-1}}_{rv}\left[\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}\right]_{rr} & 0_{3\times 3\times 6} \end{bmatrix} \tag{4.64}$$

$$\frac{\partial T}{\partial \Delta t_i} = \begin{bmatrix} \Phi^{n_0:c^{-1}}_{rv}\left[\frac{\partial \Phi^{n_0:c}}{\partial \Delta t_i}\right]_{rv}\Phi^{n_0:c^{-1}}_{rv}\Phi^{n_0:c}_{rr} - \Phi^{n_0:c^{-1}}_{rv}\left[\frac{\partial \Phi^{n_0:c}}{\partial \Delta t_i}\right]_{rr} & 0_{3\times 3} \end{bmatrix} \tag{4.65}$$

$$\frac{\partial L}{\partial \boldsymbol{x}_{0,i}} = \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{x}_{0,i}}\left(WT^\top\right) & 0_{3\times 3} \end{bmatrix} = \begin{bmatrix} \frac{\partial W}{\partial \boldsymbol{x}_{0,i}}T^\top + W\frac{\partial T^\top}{\partial \boldsymbol{x}_{0,i}} & 0_{3\times 3\times 6} \end{bmatrix} \tag{4.66}$$

$$\frac{\partial W}{\partial \boldsymbol{x}_{0,i}} = \frac{\partial}{\partial \boldsymbol{x}_{0,i}} \begin{bmatrix} \Phi_{vv}^{n_0:c} & -\Phi_{vr}^{n_0:c} \end{bmatrix} = \begin{bmatrix} \left[\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}\right]_{vv} & \left[-\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}\right]_{vr} \end{bmatrix} \tag{4.67}$$

where the relevance of $\frac{\partial \Phi^{c:1}}{\partial \boldsymbol{x}_{0,i}}$ and $\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}$ in Equations 4.62 through 4.67 vary by segment. For example, in cases where segment i is before the segment containing t_c, $\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}$ and $\frac{\partial \Phi^{n_0:c}}{\partial \Delta t_i}$ are populated with zeros and $\frac{\partial \Phi^{c:1}}{\partial \boldsymbol{x}_{0,i}}$ follows the form

$$\frac{\partial \Phi^{c:1}}{\partial \boldsymbol{x}_{0,i}} = \Phi^{c:(i+1)} \frac{\partial \Phi^{i}}{\partial \boldsymbol{x}_{0,i}} \Phi^{(i-1):1} = \Phi^{c:(i+1)} \Phi_{II}^{i} \Phi^{(i-1):1} \tag{4.68}$$

$$\frac{\partial \Phi^{c:1}}{\partial \Delta t_i} = \Phi^{c:(i+1)} F\left(\boldsymbol{x}_{f,i}\right) \Phi^{i:1} \tag{4.69}$$

Similarly, when segment i is after the TCM segment, $\frac{\partial \Phi^{c:1}}{\partial \boldsymbol{x}_{0,i}}$ and $\frac{\partial \Phi^{c:1}}{\partial \Delta t_i}$ are zero and $\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}$ follows the form

$$\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}} = \frac{\partial}{\partial \boldsymbol{x}_{0,i}} \left[\Phi^{n_0:(i+1)} \Phi^{i} \Phi^{(i-1):c}\right] = \Phi^{n_0:(i+1)} \Phi_{II}^{i} \Phi^{(i-1):c} \tag{4.70}$$

$$\frac{\partial \Phi^{n_0:c}}{\partial \Delta t_i} = \Phi^{n_0:(i+1)} F\left(\boldsymbol{x}_{f,i}\right) \Phi^{i:c} \tag{4.71}$$

Both ranges of STM sensitivities are nonzero when segment i contains t_c. In this case, $\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}}$ is the most challenging and requires an additional step to derive the sensitivity of $\Phi^{i_f:c}$ (the STM from the TCM at t_c to $t_{f,i}$, the end of the ith segment) with respect to $\boldsymbol{x}_{0,i}$. The main additional step is expressing $\Phi^{i_f:c}$ in terms of STM segments that have $t_{0,i}$ as an endpoint.

$$\Phi^{n_0:c} = \Phi^{n_0:(i+1)} \Phi^{i_f:c} = \Phi^{n_0:(i+1)} \Phi^{i} \Phi^{i_0:c} = \Phi^{n_0:(i+1)} \Phi^{i} \left(\Phi^{c:i_0}\right)^{-1} \tag{4.72}$$

Evaluating $\partial \left(\Phi^{c:i_0}\right)^{-1} / \partial \boldsymbol{x}_{0,i}$ involves utilizing the matrix inverse derivative property:

$$\frac{\partial \left(\Phi^{c:i_0}\right)^{-1}}{\partial \boldsymbol{x}_{0,i}} = -\Phi^{i_0:c} \Phi_{II}^{c:i_0} \Phi^{i_0:c} \tag{4.73}$$

$$\frac{\partial \Phi^{n_0:c}}{\partial \boldsymbol{x}_{0,i}} = \Phi^{n_0:(i+1)} \frac{\partial}{\partial \boldsymbol{x}_{0,i}} \left[\Phi^i \left(\Phi^{c:i_0} \right)^{-1} \right] = \Phi^{n_0:(i+1)} \left[\Phi^i_{II} \Phi^{i_0:c} - \Phi^i \Phi^{i_0:c} \Phi^{c:i_0}_{II} \Phi^{i_0:c} \right]$$

Figure 4.2 summarizes the robust trajectory algorithm steps required to solve Problem 4.

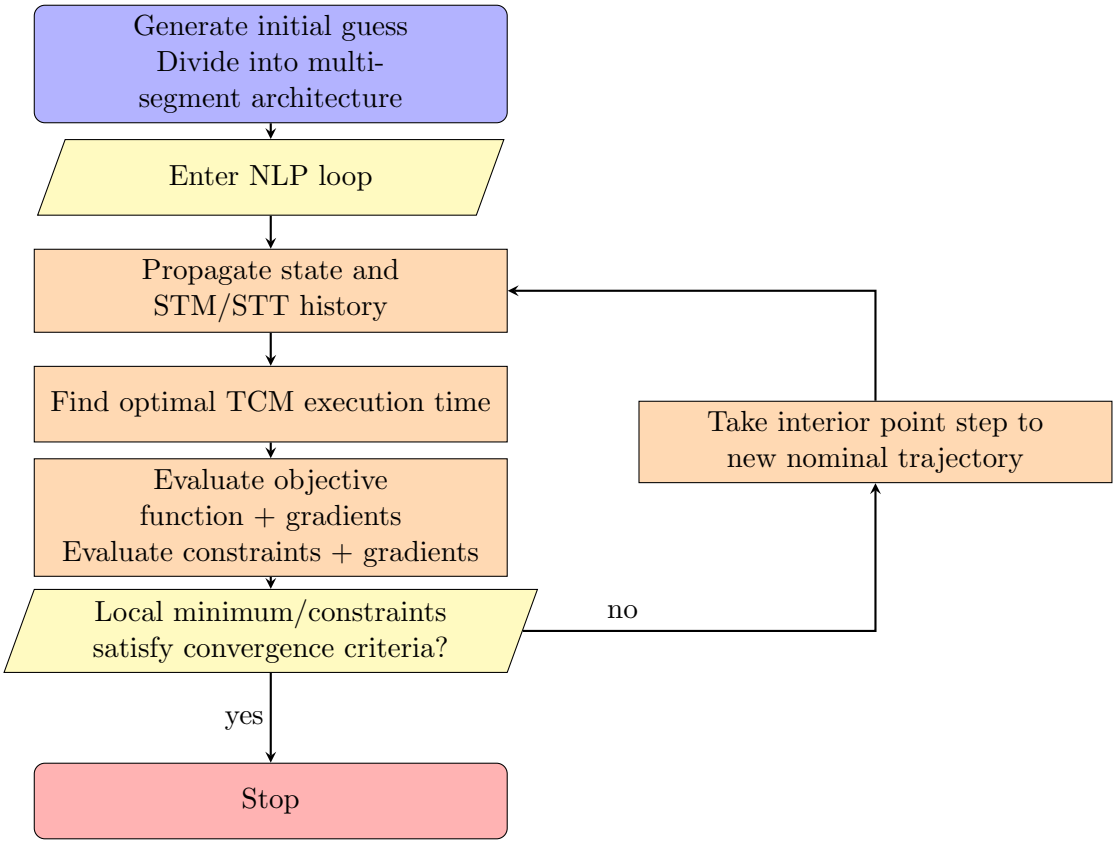


Fig. 4.2: Robust trajectory algorithm flowchart

4.2 Results

4.2.1 450 km to 1000 km Altitude, Coplanar, Bicircular, 180 Degree Offset

Scenario 1 is overly simplified but selected for a few reasons: validation of the NLP result through comparison with mission maps; the optimal deterministic solution is well

Orbital Elements	Initial State	Target State
a (km)	6828	7378
e	0	0
i	45°	45°
Ω	0	0
u (Arg. Latitude)	0°	180°

Table 4.1: Scenario 1 initial and target state orbital elements

known; in support of building intuition incrementally. Figure 4.3 depicts the scenario and the optimal deterministic solution while Table 4.1 shows the initial and target state orbital elements for a circular inclined orbit. The trajectory setup for this scenario is 3 trajectory segments: an initial coast segment, the transfer segment, and a final coast segment.

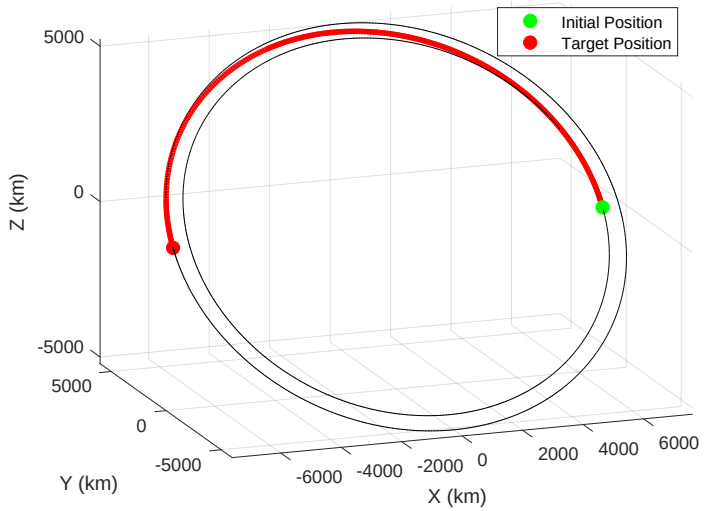


Fig. 4.3: Scenario 1 setup and deterministic optimal solution

Optimal deterministic trajectory

With this 3 segment trajectory, the problem can be reduced to two independent parameters to create a mission map of the solution space. The two parameters chosen are the coast time in Segment 1, Δt_1, and the coast time in Segment 3, Δt_3. With a fixed total

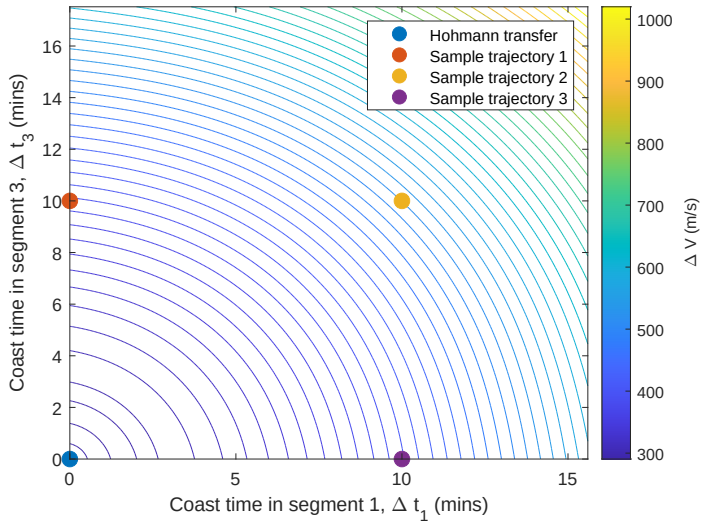


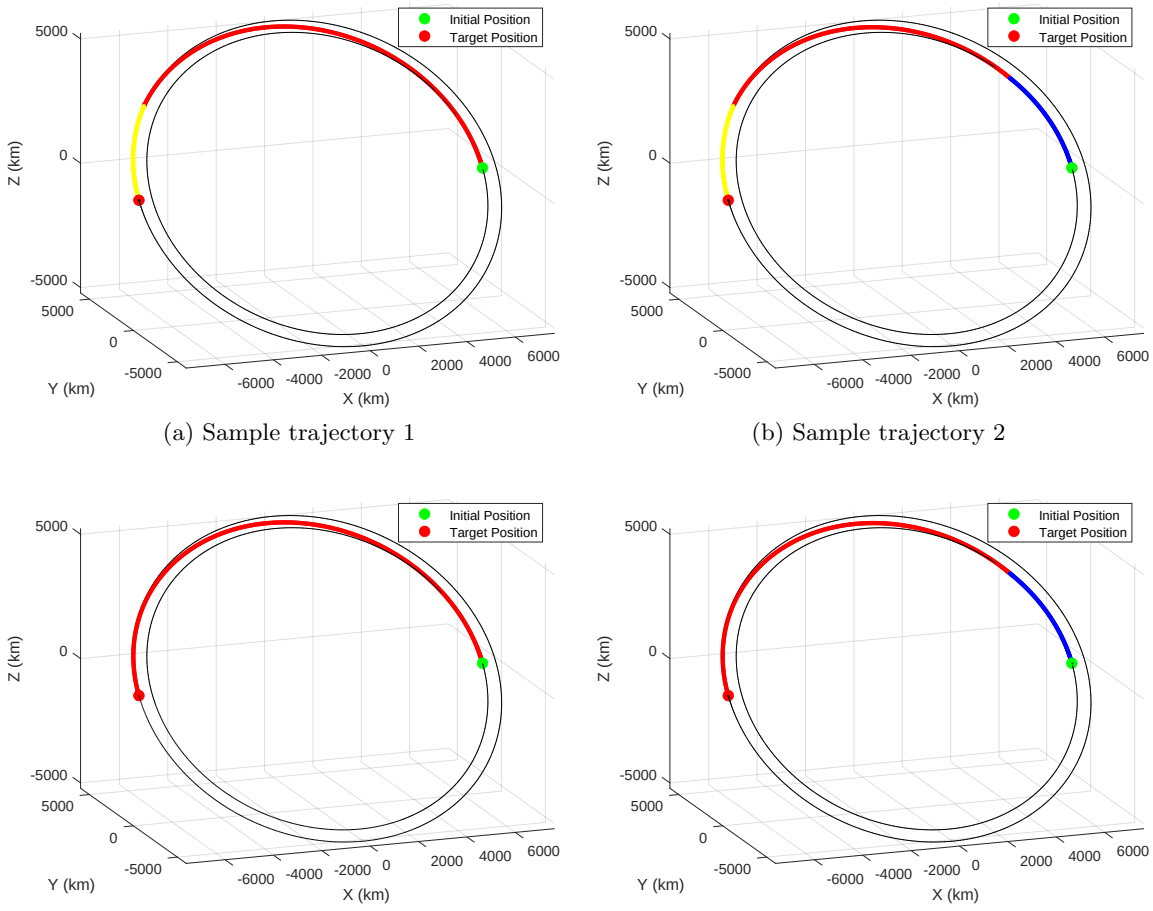
Fig. 4.4: Scenario 1 deterministic ΔV mission map

duration this determines the transfer segment duration, Δt_2. Figure 4.4 shows a contour map of the nominal ΔV magnitude as a function of possible Δt_1(x-axis) and Δt_3 (y-axis) combinations. Each individual point on the mission map is a different nominal trajectory. Four sample nominal trajectories are shown in Figure 4.5 corresponding to dots in Figure 4.4. Along each trajectory in Figure 4.5, individual trajectory segments are identified by different colors.

The solution space in Figure 4.4 is ideal in this case for gradient-based optimization. Only one local minimum value exists and the gradient of the cost function results in successful convergence toward the desired solution. The NLP solution method described in Section 4.1 results in rapid convergence as well to the Hohmann transfer with a total nominal ΔV of 290.2 m/s.

Initial state dispersion with target dispersion constraint trajectory

The next aspect of Scenario 1 is the introduction of the initial state dispersion $\delta \boldsymbol{x}_0$. The magnitude of $\delta \boldsymbol{x}_0$ modeled for this analysis is on the scale of launch vehicle orbit insertion accuracy. The initial dispersion elements are modeled as zero-mean, uncorrelated, normally distributed random variables with position variance $\sigma_r^2 = (10\text{km})^2$ per axis, and velocity



(a) Sample trajectory 1

(b) Sample trajectory 2

(c) Hohmann transfer

(d) Sample trajectory 3

Fig. 4.5: Scenario 1 sample mission map trajectories

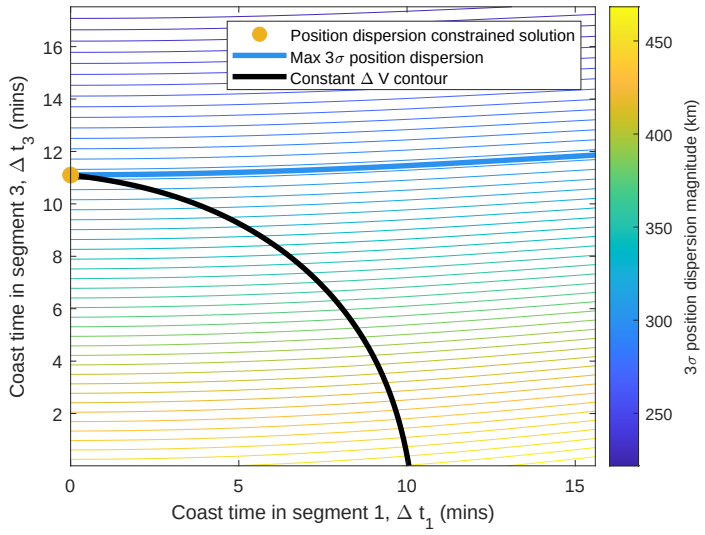


Fig. 4.6: 3σ position dispersion magnitude at target orbit arrival

variance $\sigma_v^2 = (10\text{m/s})^2$ per axis. The initial dispersion forms the dispersion covariance matrix at the beginning of segment 1, P_1.

The dispersion covariance as a function of time varies with each trajectory. An important metric to many missions is the magnitude of the position dispersion upon arrival to a target orbit. Similar to the ΔV mission map in Figure 4.4, Figure 4.6 shows the 3σ RSS position dispersion upon arrival to the target orbit (at the end of segment 2). The 3σ position dispersion is 452 km upon arrival to the target orbit for the Hohmann transfer. Suppose a mission requirement is that the 3σ position dispersion is less than or equal to 300 km upon arrival to the target orbit. Figure 4.7 shows the result of utilizing the NLP solution methodology outlined in Section 4.1.2 (Problem 2) to satisfy the position dispersion constraint. As a result, the nominal ΔV increases to 407.7 m/s. Comparing the ΔV mission map and the 3σ position dispersion mission map also enables visual manual optimization to confirm the result with no Δt_1 coast and an approximately 11.1 minute Δt_3 coast. Figure 4.6 facilitates this comparison: the thicker blue line represents the maximum target position dispersion constraint of 300 km. The lowest ΔV intersection with the position dispersion constraint occurs at the NLP result and the corresponding ΔV contour (from Figure 4.4) of 407.7 m/s is represented by the thick black line.

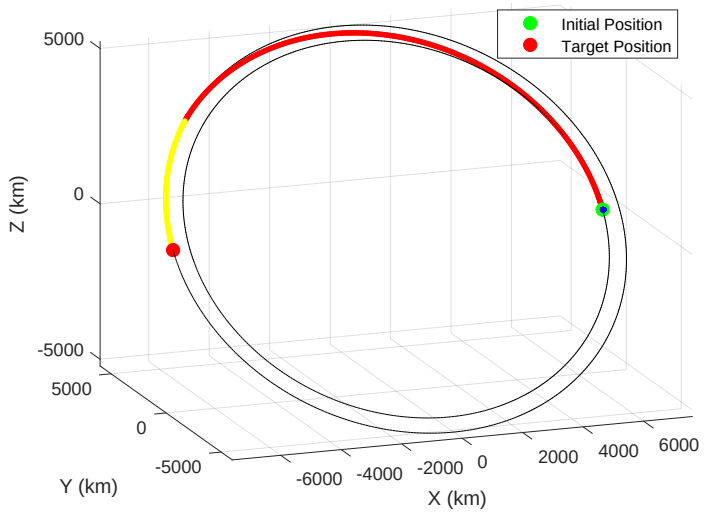


Fig. 4.7: Position dispersion constrained solution

Optimal 3σ TCM along deterministic optimal trajectory

A more effective method of minimizing position dispersion at a target is by performing a TCM. A second TCM is required to also correct the velocity dispersion upon arrival to the target state. The sum of these forms the total TCM magnitude, δV, as described in Section 4.1.3 (Problem 3). The magnitude of the TCM varies based on the time of execution along a nominal trajectory. Once a nominal trajectory and STM time history are numerically propagated along a reference trajectory, it is possible to rapidly find the minimum value and TCM execution time.

Figure 4.8 shows the TCM magnitude time history along the four sample trajectories as well as the selected minimum. Figure 4.9 shows the location of the optimal TCM along the original deterministic Hohmann transfer. The four TCM histories shown in Figure 4.8 visually assist in understanding the solution space. First, the four TCM histories show how the minimum TCM and corresponding execution time is identified. Second, they show how significantly the minimum TCM magnitude can change based on the nominal trajectory chosen. Trajectories 1 and 2 exhibit lower $3\sigma_{\delta V}$ minimums. However, when the deterministic ΔV for trajectories 1 and 2 are compared in Figure 4.4, it is clear that trajectory 1 is the lower of the two.

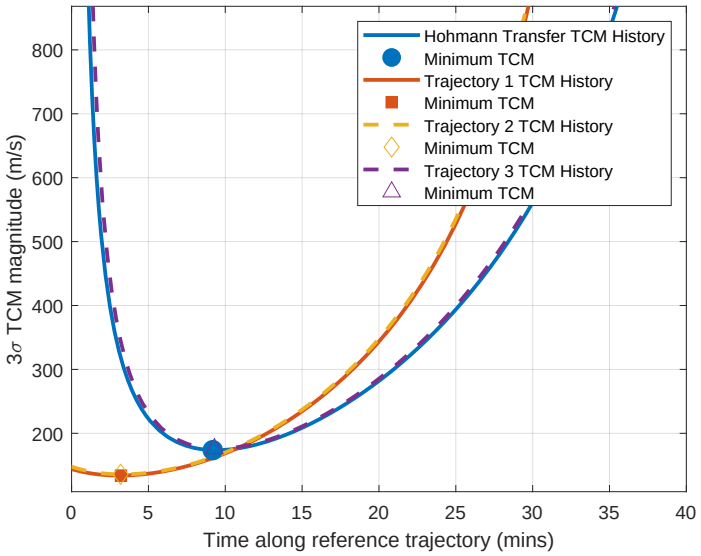


Fig. 4.8: Minimum $3\sigma_{\delta V}$ along sample trajectories

For the four current sample trajectories in Figure 4.8, another observation is the existence of the increased magnitude toward the beginning of the trajectory near a 180 degree offset. This is akin to the Lambert ridge and represents the required out of plane trajectory to correct the out of plane position dispersion. The plotted magnitudes during the initial spike are likely inaccurate however as the trajectory has departed the linear region around the nominal trajectory. The increased TCM magnitude toward the end of each trajectory is intuitive; as less time remains to correct a dispersion a higher impulse is required.

Robust trajectory, optimal nominal ΔV and 3σ TCM

Introducing the nominal ΔV and the $3\sigma_{\delta V}$ TCM magnitude into the cost function (Problem 4) provides the opportunity to find a lower ΔV statistical upper bound. Figure 4.10 shows a mission map that is the sum of the nominal ΔV and the optimal TCM along each trajectory. The minimum value is no longer the Hohmann transfer. Figure 4.11 shows the robust trajectory result from the NLP which the mission map confirms as the optimal solution.

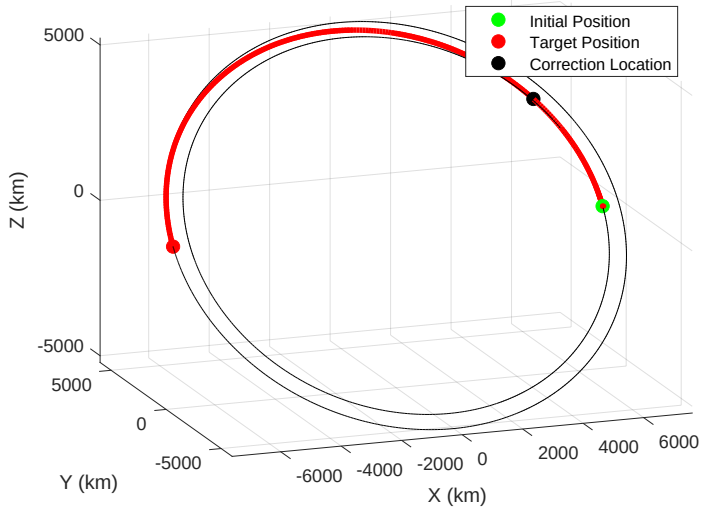


Fig. 4.9: Optimal $3\sigma_{\delta V}$ along optimal deterministic trajectory

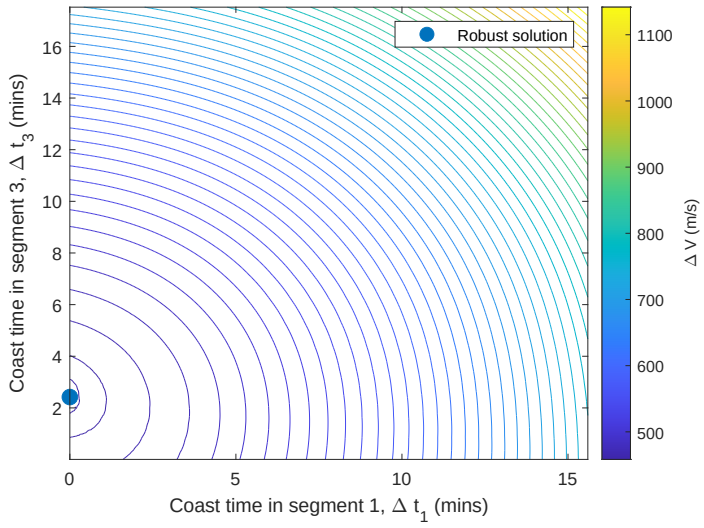


Fig. 4.10: Robust trajectory mission map: nominal ΔV plus $3\sigma_{\delta V}$

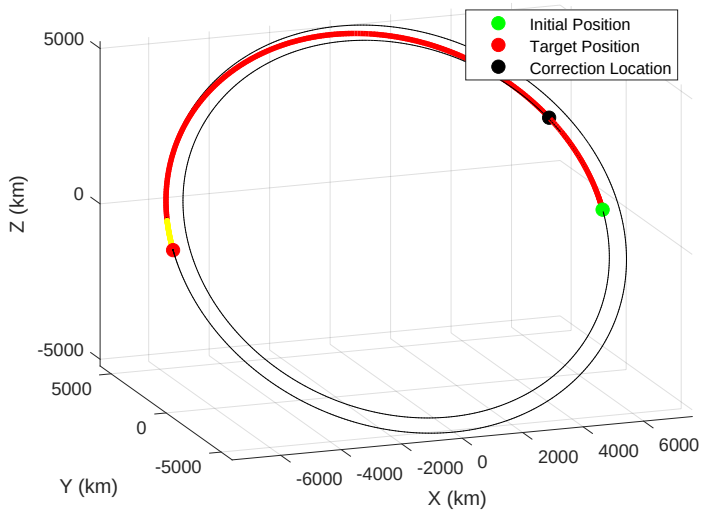


Fig. 4.11: Robust trajectory

Results comparison

Table 4.2 shows the summary of the solutions to Problems 1 through 4 for the LEO to LEO transfer (Scenario 1). These results show that the robust optimal path is no longer the Hohmann transfer. The robust path can offer a savings of up to 5.9 m/s over taking the deterministic optimal. Another important observation about this set of results is the ratio of nominal ΔV to expected $3\sigma_{\delta V}$ magnitude. The $3\sigma_{\delta V}$ is a significant percentage of the nominal ΔV magnitude, which results in a greater opportunity for TCM cost savings versus scenarios where the nominal ΔV dominates the cost function. This is one set of characteristics where a robust trajectory is more likely to provide an opportunity for savings.

4.2.2 450 km to 20,000 km, Coplanar, Bicircular

Similar to the first scenario, Scenario 2 is optimized via mission maps alongside the NLP results. Additional coast duration is incorporated prior to ΔV_1 and after ΔV_2 to enable flexibility in transfer angle, initial orbit departure, and target orbit arrival.

Optimal deterministic trajectory

With a fixed total transfer time, it is again possible to reduce the problem to a function of two variables, Δt_1 and Δt_3. The fixed duration mimics a rendezvous scenario in the target

Trajectory Description	**Nominal ΔV (m/s)**	$3\sigma_{\delta V}$ **(m/s)**	ΔV **Upper Bound (m/s)**	3σ **Target Dispersion (km)**
1. Optimal deterministic w/initial state dispersion	290.2	N/A	290.2	451.7
2. Initial state dispersion with target position dispersion constraint	407.7	N/A	407.7	300
3. Optimal 3σ TCM along deterministic optimal	290.2	173.6	463.8	0
4. Robust, optimal nominal ΔV and 3σ TCM	295.2	162.7	457.9	0

Table 4.2: Results: 450km to 1000km altitude, coplanar, 180 degree offset

orbit but the coast duration in the target orbit introduces flexibility to rendezvous with the target earlier or later than the position in the initial guess. The total fixed transfer duration is somewhat arbitrarily equal to a quarter initial orbit coast, a Hohmann transfer, and a $\frac{1}{12}$ target orbit coast. Figure 4.12 shows the nominal ΔV solution space as a mission map of combinations of nominal trajectories and the corresponding total nominal ΔV required. Six sample trajectory solutions are indicated as well which will be referenced further during the TCM analysis portion of this scenario. Figure 4.13 shows the deterministic optimal (minimum ΔV) NLP solution which corresponds to sample trajectory 2 in Figure 4.12. The nominal ΔV is 3384.45 m/s.

Initial state dispersion with target dispersion constraint trajectory

Without any TCMs and the same initial state dispersion covariance as the previous scenario, the 3σ position dispersion at the target orbit along the deterministic optimal trajectory is 1603.00 km. Figure 4.14 shows the target orbit 3σ position dispersion map.

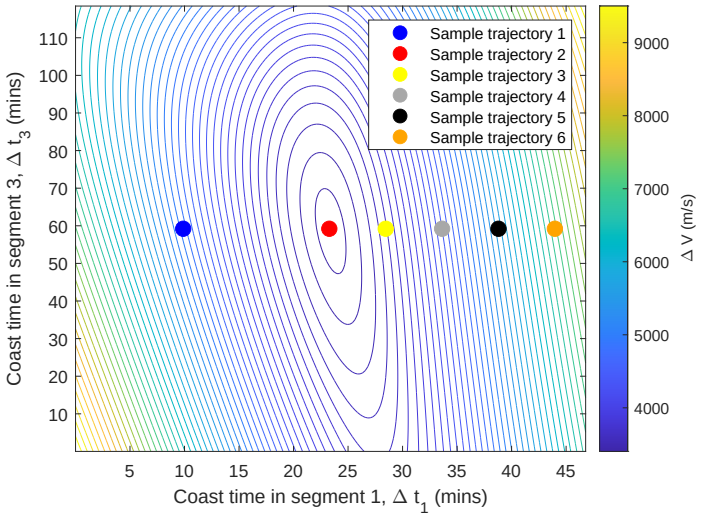


Fig. 4.12: Deterministic ΔV mission map

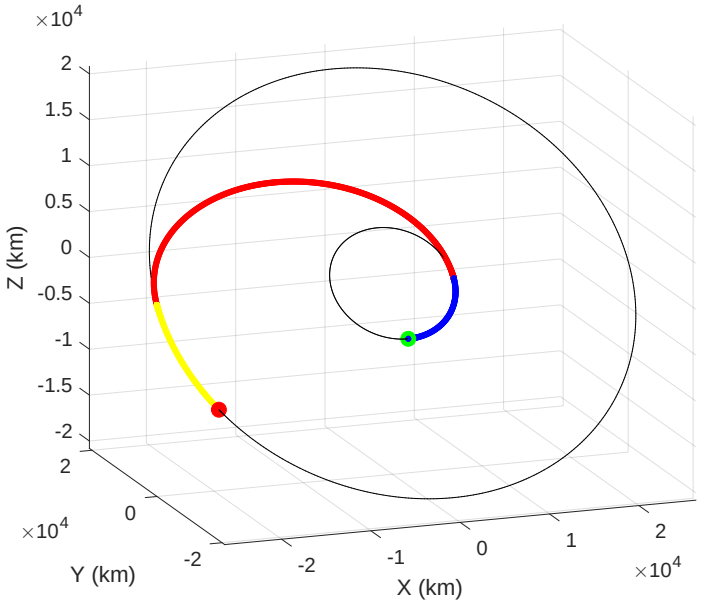


Fig. 4.13: Deterministic optimal

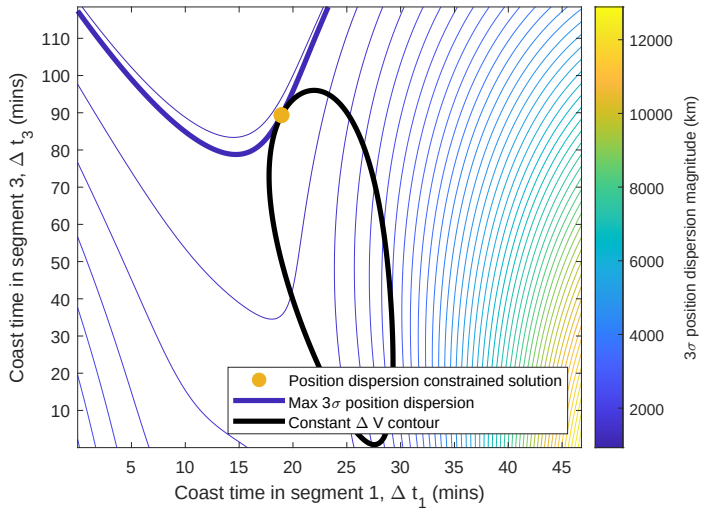


Fig. 4.14: 3σ position dispersion at target orbit

Figure 4.15 shows the minimum ΔV solution to meet a maximum 3σ position dispersion constraint at the target orbit. The result is verified on the position dispersion mission map (Figure 4.14): the thick blue line is a constant contour of 1200 km 3σ position dispersion. The lowest constant ΔV contour that is tangent to the position dispersion of interest is the thick black line (following the shape of the Figure 4.12 contours), corresponding to a nominal ΔV of 3666.46 m/s. The NLP result in Figure 4.15 corresponds to the dot at the intersection of the black and cyan contour lines.

Optimal 3σ TCM along deterministic optimal trajectory

Along the deterministic optimal trajectory, the minimum TCM magnitude is 76.51 m/s. Figure 4.19 shows the trajectory with the location of the TCM identified by the black dot.

As an aside, the shape of the mission map exhibits interesting behavior. Continuing a similar analysis to the first scenario, each point on the TCM mission map represents the minimum TCM along each reference trajectory. Figure 4.16 shows the mission map of the minimum 3σ TCM magnitude along each nominal trajectory combination. Figure 4.17a shows the minimum 3σ TCM magnitudes across the range of Δt_1 values and a single Δt_3 value. This highlights the potentially non-intuitive nature of the TCM solution space.

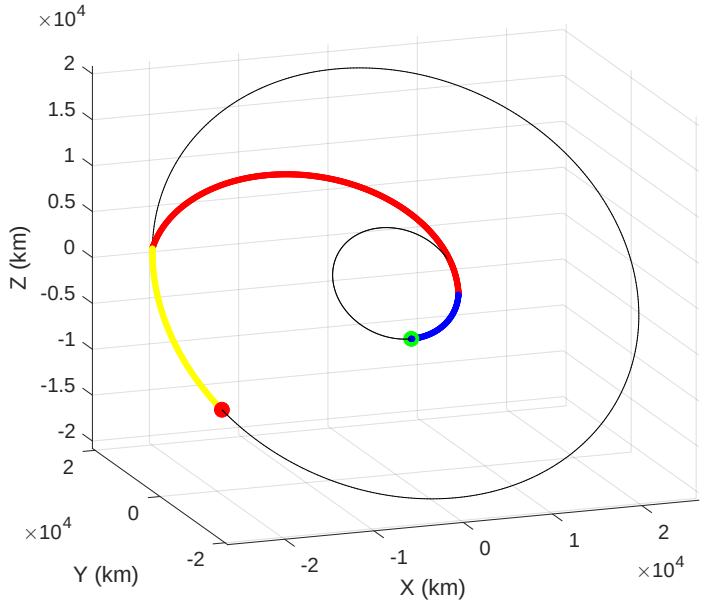


Fig. 4.15: Target position dispersion constrained

Figure 4.17b shows the TCM magnitude as a function of TCM execution time along the nominal trajectory. While some similarities exist between each TCM plot as a function of time (e.g. earlier is generally better), that that is not always the case. Sample trajectory 2 also exhibits the Lambert ridge phenomenon 180° from the target. Sample trajectories four through six exhibit multiple spikes, the location of which contribute to the spike in the minimum TCM plot in Figure 4.17a.

Robust trajectory, optimal nominal ΔV and 3σ TCM, fixed final time

There is visibly little difference in the optimal deterministic and robust trajectory in this scenario or between the corresponding mission maps in Figures 4.12 and 4.18. The resulting robust nominal ΔV is 3384.46 m/s with a 3σ TCM magnitude of 76.47 m/s for a total of 3460.93 m/s. The savings in this case are trivial as the ratio of nominal ΔV to 3σ TCM magnitude is very high. There is little opportunity to perform modifications to the nominal trajectory which save more TCM than the cost of the increased nominal ΔV. A similar result applies to Scenario 3 and is discussed further in Section 4.2.3.

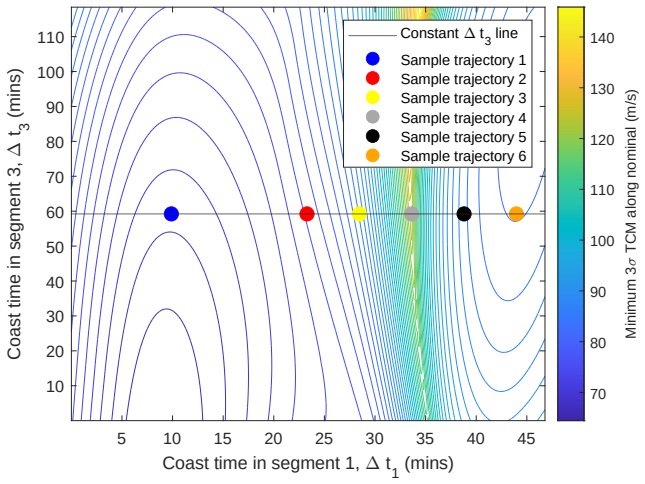
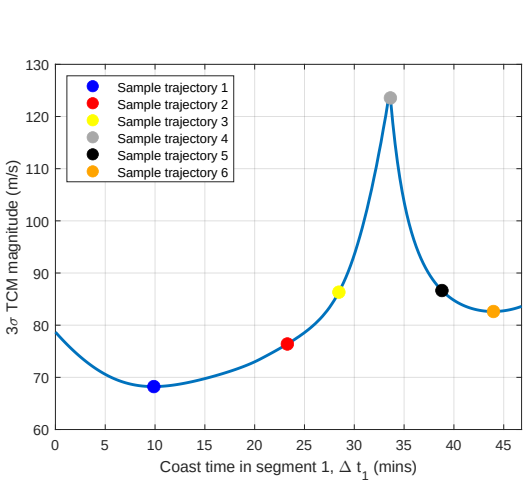
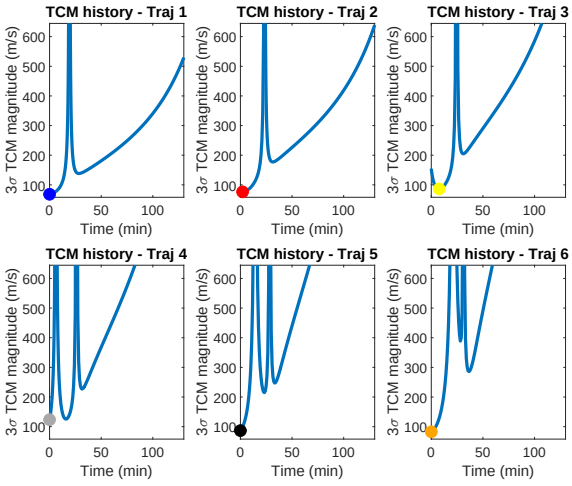


Fig. 4.16: Minimum 3σ TCM RSS along nominal trajectory mission map



(a) Constant Δt_3 line from Figure 4.16



(b) Sample trajectory TCM magnitude as a function of time

Fig. 4.17: Scenario 2, Problem 3 sample solutions

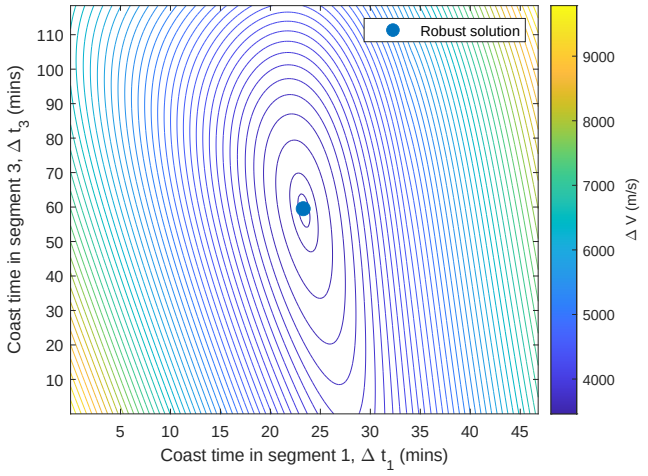


Fig. 4.18: $\Delta V + 3\sigma_{\delta V}$ mission map

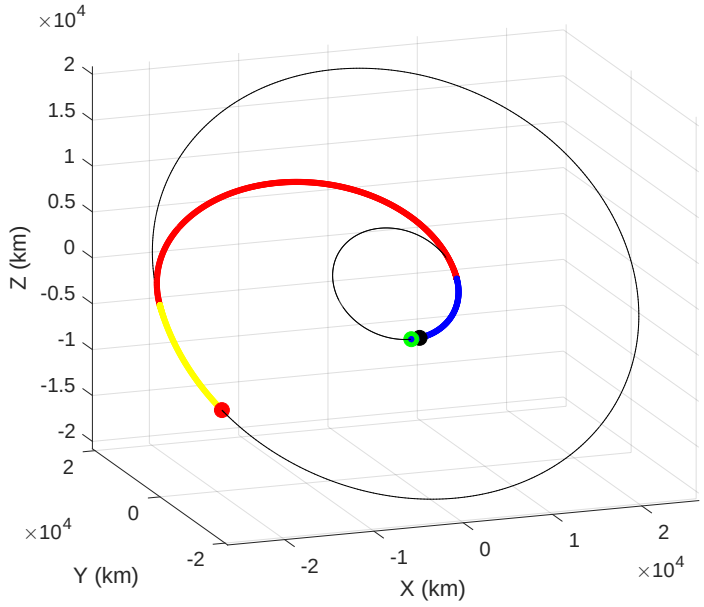


Fig. 4.19: Scenario 2, deterministic optimal w/TCM and robust, fixed final time

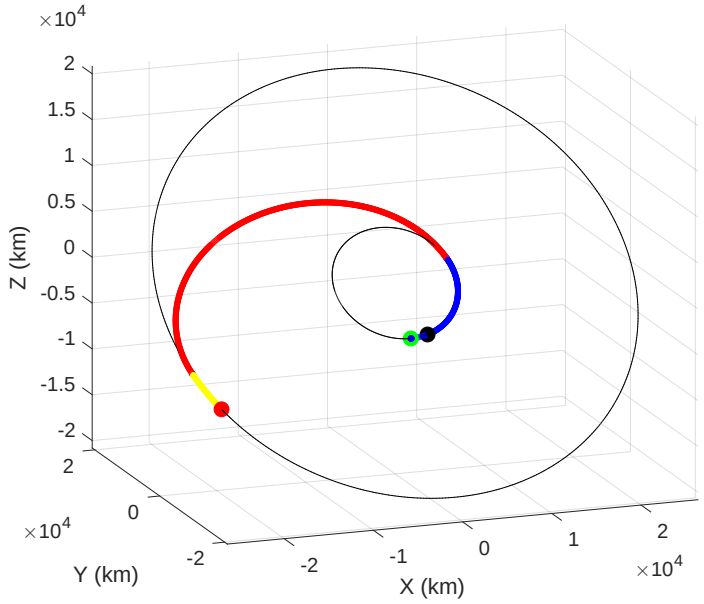


Fig. 4.20: Scenario 2, robust, free final time

Robust trajectory, optimal nominal ΔV and 3σ TCM, free final time

One additional small reduction in total ΔV can be made by removing the fixed total duration (or fixed final time) constraint. Figure 4.20 shows the resulting trajectory. The main difference with this solution is optimized coast times in Segments 1 and 3 to further minimize the cost function.

Results comparison

Table 4.3 shows the results for the four problems for the current scenario. The large nominal ΔV in this case prevents significant ΔV savings in the robust trajectory solutions, as modeled.

Trajectory Description	**Nominal ΔV (m/s)**	**$3\sigma_{\delta V}$ (m/s)**	**ΔV Upper Bound (m/s)**	**3σ Target Dispersion (km)**
1. Optimal deterministic w/initial state dispersion	3384.45	N/A	3384.45	1603.00
2. Initial state dispersion with target position dispersion constraint	3666.46	N/A	3666.46	1200
3. Optimal 3σ TCM along deterministic optimal	3384.45	76.51	3460.96	0
4. Robust, fixed final time	3384.46	76.47	3460.93	0
5. Robust, free final time	3384.45	76.20	3460.65	0

Table 4.3: Results: 450 km to 20,000 km, coplanar, bicircular

4.2.3 450 km, 28° Inclined, Circular to Geostationary

Scenario 3 represents a launch into a circular, 28° inclined, 450km altitude orbit followed by a subsequent transfer to geostationary orbit. The optimal deterministic to robust range of results are presented in Table 4.4. One new mission-specific "problem" (Problem 5) is introduced to capture an additional mission constraint where ΔV_1 and δV_r must occur

simultaneously. This may be required in the case that an upper stage has a limited number of burns, for example.

Optimal deterministic trajectory

The optimal deterministic trajectory is as expected. $\Delta V_1$and ΔV_2 occur at the intersection of the initial and target orbital planes. ΔV_1 includes 2.22° of inclination change with the remainder occurring at ΔV_2. The total nominal ΔV is 4168.27 m/s. Introducing the same magnitude initial dispersion ($\sigma_r = 10\text{km}, \sigma_v = 10\text{m/s}$) results in a 3σ position dispersion upon arrival to the target orbit of 2710.05 km.

Initial state dispersion with target dispersion constraint trajectory

In this case, a 2000 km 3σ position dispersion inequality constraint is introduced at arrival to the target orbit, although the specific value is somewhat arbitrary. The nominal ΔV is now 5312.42 m/s to meet the position dispersion constraint.

Another interesting set of solutions is the removal of the fixed total transfer time constraint. The summary for this case is the optimal dispersion constrained path results in shortening the total transfer time, mainly via shortening the duration of the geostationary transfer segment.

Optimal 3σ TCM along deterministic optimal trajectory

The optimal TCM along the deterministic optimal trajectory is not performed immediately, but rather 2.15 minutes into the first segment for a minimum 3σ magnitude of 72.08 m/s.

Robust trajectory, optimize nominal ΔV and 3σ TCM

The result in this case is relatively unremarkable as the ratio of the nominal ΔV to $3\sigma_{\delta V}$ is very high. As a result, alterations to the nominal trajectory by incorporating the TCM into the cost function are minimal. The total ΔV is reduced by 0.04 m/s.

Robust trajectory, ΔV_1 and δV_r concurrent constrained to occur concurrently

This result introduces an additional constraint whereby ΔV_1 and δV_r are required to occur simultaneously. A likely applicable scenario may be a limitation on the number of impulsive burns that a launch vehicle upper stage can perform, requiring that a single maneuver ($\Delta V_1 + \delta V_r$) perform both insertion into geostationary transfer orbit as well as minimizing the position dispersion at a future target. The angular distance traveled between $\Delta V_1$and ΔV_2 is 187°, resulting in a ΔV statistical upper bound of of 4271.15 km/s. This result optimizes the avoidance of the Lambert ridge for correcting the out of transfer plane dispersion and the deviation from the optimal deterministic transfer angle.

Results comparison

Table 4.4 shows a results summary for the current scenario. There are a few conclusions from this set of results. First, when the nominal total ΔV is significantly greater than the $3\sigma_{\delta V}$ magnitude, the difference between the optimal deterministic trajectory and the robust trajectory becomes less significant. Additionally, for this relatively long duration trajectory (when compared to Scenario 1, for example), the minimum 3σ TCM is lower as there is more time for a smaller adjustment to correct a similarly-sized dispersion. However, maneuver execution is assumed to be perfect in the current model. Increasing the model fidelity to incorporate maneuver execution error may result in the opposite effect: maneuver errors will be amplified over the course of a long trajectory and require additional TCM(s) in the optimal robust solution.

4.2.4 450km, Critically Inclined to 894 km Sun-synchronous, Bicircular

Scenario 4 models a potential ride share launch challenge. While design and situation specific, Scenario 4 highlights potential mission customization options and the relative ease of incorporating additional constraints to suit a specific set of needs without introducing problem convergence issues. The payload of interest is a ride share on a main mission payload launch with a target 450 km, circular, critically inclined (63.4°) target orbit. The target orbit for the ride share payload is an 894 km sun-synchronous orbit. The main

Trajectory Description	**Nominal ΔV (m/s)**	$3\sigma_{\delta V}$ **(m/s)**	ΔV **Upper Bound (m/s)**	3σ **Target Dispersion (km)**
1. Optimal deterministic w/initial state dispersion	4168.27	N/A	4168.27	2710.05
2. Initial state dispersion with target position dispersion constraint	5312.42	N/A	5312.42	2000
3. Optimal 3σ TCM along deterministic optimal	4168.27	72.08	4240.35	0
4. Robust, optimal nominal ΔV and 3σ TCM	4168.31	72.00	4240.31	0
5. Robust, ΔV_1 and δV_r concurrent	4203.57	67.58	4271.15	0

Table 4.4: Results: 450km, 28° inclined, circular to geostationary

mission payload does not utilize the entirety of the launch vehicle's lift capacity to low Earth orbit and as a result, the upper stage has enough propellant margin to perform an additional large burn prior to its disposal burn. One challenge is that the most efficient solution is performing the plane change at the higher altitude. This scenario introduces an additional constraint such that the plane change must occur during the ΔV_1 maneuver. The δV_r TCM will also be constrained to occur with ΔV_1 to minimize the target orbit dispersion for the ride share mission while meeting the upper stage capability constraints. The ride share payload of interest will separate following the transfer initiation burn, ΔV_1. As a result, the maneuver to circularize in the target orbit (ΔV_2) and correct the velocity dispersion upon arrival to the target orbit (δV_v) will be the responsibility of the ride share payload.

Optimal deterministic trajectory

Figure 4.21 shows the optimal deterministic trajectory without any of the mission-specific ride share constraints applied.

Initial state dispersion with target dispersion constraint trajectory

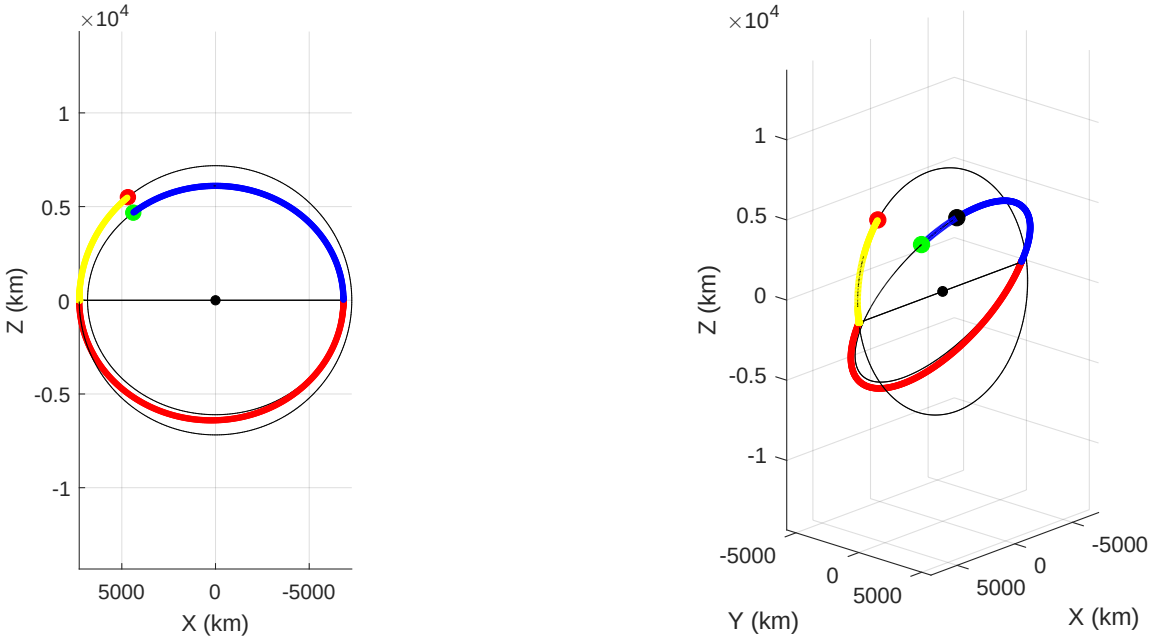
A 3σ position dispersion constraint of 400 km is introduced resulting in a total ΔV of 5472.96 m/s. The fixed total time constraint was removed in this case and for the remainder of this scenario.

Optimal 3σ TCM along deterministic optimal trajectory

Figure 4.21 shows the location of the optimal 3σ TCM along the optimal deterministic trajectory as a black dot. Table 4.5 shows the results.

Robust trajectory, optimal nominal ΔV and 3σ TCM

Due to the significantly larger magnitude again of the nominal ΔV in comparison to the $3\sigma_{\delta V}$, there is little to be gained by deviating from the nominal trajectory in order to



(a) (b)

Fig. 4.21: Scenario 4 optimal deterministic trajectory with optimal TCM identified

reduce the TCM magnitude. The resulting robust trajectory path is not visibly different and is not replotted as a result. Table 4.5 shows the resulting magnitudes.

Mission-specific robust

This trajectory solution implements the following additional constraints:

- ΔV_1 and δV_r constrained to occur concurrently: $(\Delta t_1 = t_c)$
- All inclination plane change constrained to occur with ΔV_1; equivalently, the transfer plane is coplanar with the target plane (Equation 3.54).

Figure 4.22 shows the resulting mission-specific trajectory. Table 4.5 shows the resulting maneuver magnitudes. Individual maneuver magnitudes are provided as one of the goals of this trajectory is reducing the ΔV requirement for the ride share spacecraft to circularize in the target orbit (ΔV_2) and correct the remaining velocity dispersion (δV_v). While the resulting trajectory is not the lowest TCM magnitude in the result set, there is a significant

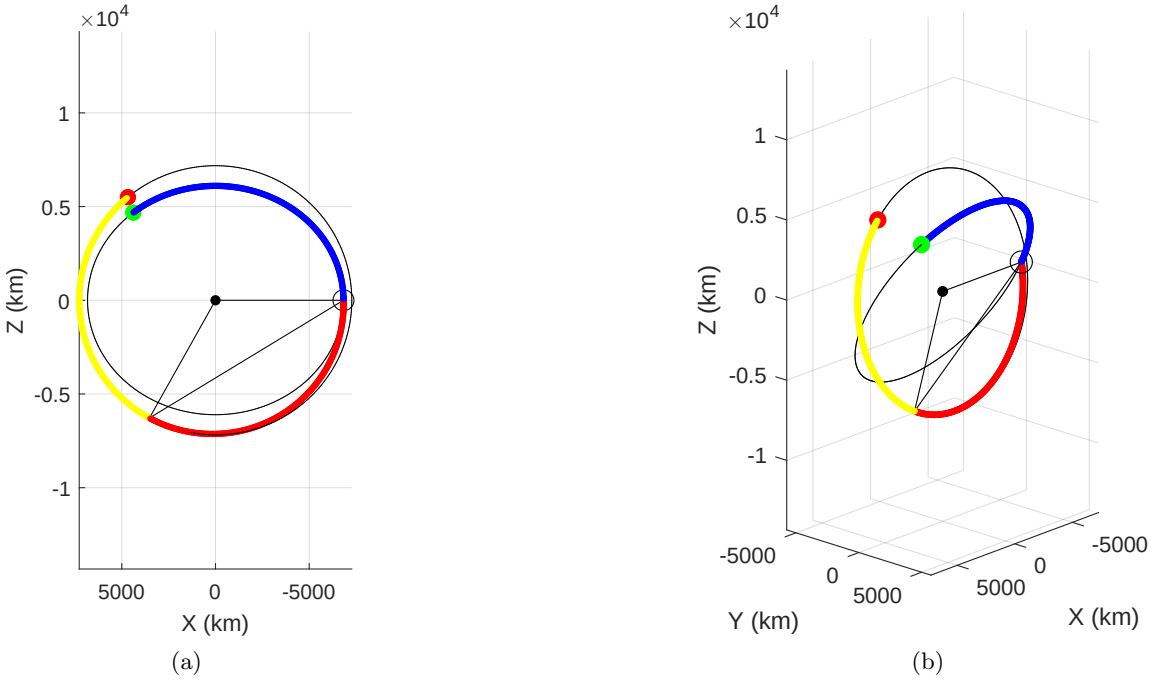


Fig. 4.22: Scenario 4 mission-specific robust trajectory

reduction in the overall ΔV requirement for the ride share spacecraft while still balancing the initial upper stage burn requirement.

Results comparison

Table 4.5 summarizes the Scenario 4 results.

4.2.5 Comments On Initial Guesses and Convergence

Two convergence examples (Figures 4.23 and 4.24) show the initial guess and intermediate steps taken toward the converged solution. Figure 4.23 shows the 82 step convergence history for the Scenario 2 robust, fixed final time trajectory. Figure 4.24 shows two views of the 114 step convergence history for the Scenario 4 mission-specific robust trajectory. Both examples show that the initial guesses are suboptimal and infeasible given the constraint set but share some of the characteristics with the converged solutions.

In general, convergence has not been an issue for this trajectory design method when provided an n segment initial guess trajectory that is in the neighborhood of the solution.

Trajectory Description	**Nom.** ΔV_1 **(m/s)**	**Nom.** ΔV_2 **(m/s)**	$3\sigma_{\delta V_r}$ **(m/s)**	$3\sigma_{\delta V_v}$ **(m/s)**	$\Delta V_2 + 3\sigma_{\delta V_v}$ **(m/s)**	ΔV **Upper Bound (m/s)**	3σ **Target Dispersion (km)**
1. Optimal deterministic w/initial state dispersion	292.42	4248.93	N/A	N/A	4248.93	4541.35	750.85
2. Initial state dispersion w/target position dispersion constraint	2027.72	3445.24	N/A	N/A	3445.24	5472.96	400
3. Optimal 3σ TCM along deterministic optimal	292.42	4248.93	66.95	49.16	4298.09	4657.46	0
4. Robust, optimal nominal ΔV and 3σ TCM	285.43	4255.99	66.94	49.03	4305.02	4657.39	0
5. Mission specific robust	4705.05	151.52	309.66	225.89	377.40	5392.11	0

Table 4.5: Results: 450km, critically inclined to 894 km sun-synchronous, bicircular

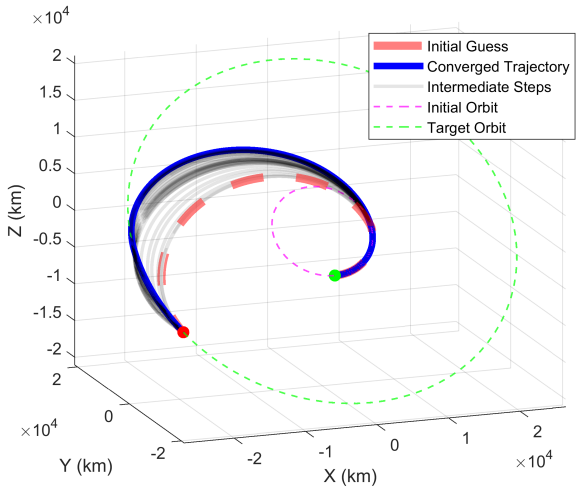
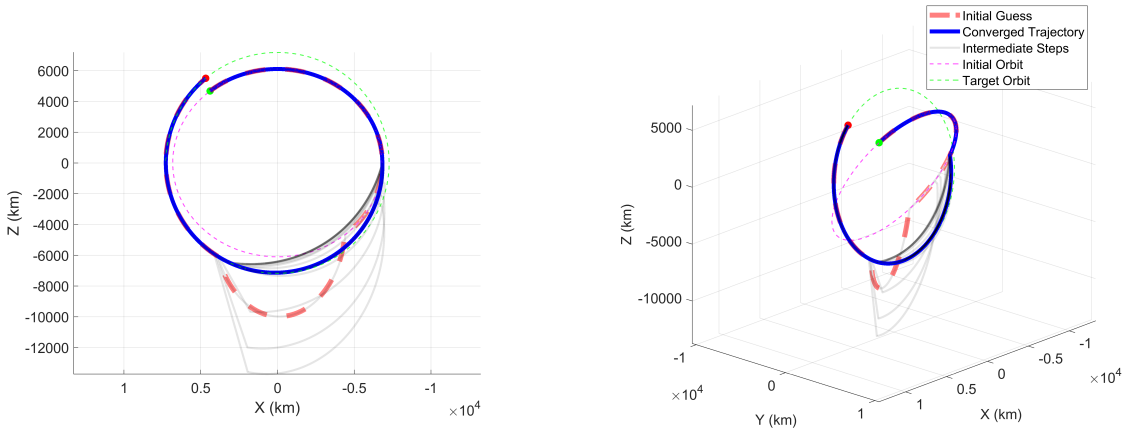


Fig. 4.23: Scenario 2 robust, fixed final time convergence steps

Lambert's Problem was used to generate connected initial trajectories with reasonable but somewhat arbitrary orbital departure/arrival positions and durations [51]. Segment connectivity is not a requirement, which presents other initial guess options such as two-sided shooting [25] for systems where a closed form solution does not exist. Another option is a continuation-like approach where the deterministic trajectory is first found and then passed as the initial guess when finding the robust solution.

Regarding the converged solution, the result is only guaranteed to be a local minimum. Large deviations from the initial guess are inherently not considered. For example, there are likely a set of LEO to LEO multiple-revolution local minima that the current algorithm, as described, will not find. Solutions with more than two nominal impulsive maneuvers are also not currently considered.

While not an extensive convergence study, Table 4.6 shows a summary of the convergence time and number of iterations for the problems presented herein. All analyses were performed on a Lenovo Thinkpad Carbon X1 Gen 9. The overall trajectory duration affects the propagation time for each NLP step and overall time to convergence. The free final time examples appear to be outliers, however large adjustments to segment durations is generally time consuming due to the small steps taken. The dispersion constrained trajectories have



(a) (b)

Fig. 4.24: Scenario 4 mission-specific robust convergence steps

a longer convergence time as well, however many small steps are taken very close to the converged solution in these cases to meet convergence thresholds.

4.3 Conclusion

This chapter introduced a trajectory design method that accounts for a normally distributed initial state dispersion and minimizes the sum of nominal impulsive ΔV plus 3σ TCM magnitude. The multiple segment approach allows for the flexible incorporation of mission events. Direct optimization was performed via nonlinear programming. Analytical gradients of state dispersion covariance and TCM δV covariance were derived to accelerate and aid convergence. Results showed that robust optimal trajectories have improved performance over deterministic optimal trajectories which are no longer optimal when stochastic effects are included.

An important conclusion from the results is the identification of transfer scenario characteristics that result in the greatest benefit based on the optimal robust trajectory versus the optimal deterministic trajectory. Short duration trajectories with a somewhat small nominal ΔV serve to benefit the most by choosing the optimal robust trajectory: TCM

Trajectory description	**Convergence time (sec)**	**Number of iterations**
Scenario 1, deterministic	10.5	37
Scenario 1, dispersion constrained	210.7	686
Scenario 1, robust	19.1	55
Scenario 2, deterministic	28.1	59
Scenario 2, dispersion constrained	744.7	1445
Scenario 2, robust, fixed total time	33.2	82
Scenario 2, robust, free total time	998.4	1887
Scenario 3, deterministic	29.8	73
Scenario 3, dispersion constrained	60.9	34
Scenario 3, robust	53.6	69
Scenario 3, mission specific robust	25.9	48
Secnario 4, deterministic	39.4	107
Scenario 4, dispersion constrained	109.4	114
Scenario 4, robust	55.8	117
Scenario 4, mission specific robust	158.0	366

Table 4.6: Convergence time and iteration count summary

δV is relatively expensive as there is less time to benefit from a correction over shorter durations; additionally, cost function trade space exists as a result of the lower magnitude nominal ΔV relative to the TCM δV. Long duration scenarios with large nominal ΔV magnitudes exhibit the opposite effect and the results show small differences between the deterministic optimal and robust solutions. The incorporation of other stochastic sources is likely to increase the benefit of taking an optimal robust path versus its deterministic counterpart.

In addition to the robust trajectory design method presented, the mission maps and the TCM optimization method presented both visually support building intuition for the impact of trajectory design choices on stochastic parameters of interest.

CHAPTER 5
STOCHASTIC ANALYSIS ALONG A NOMINAL TRAJECTORY

The trajectory parameter vector $\boldsymbol{S}$ defines a nominal trajectory. Altering any quantity in the vector $\boldsymbol{S}$ results in a new nominal trajectory that requires a new propagation and re-assessment of the deterministic cost and constraints. The state dispersion covariance defines the Gaussian statistics along the nominal trajectory, as a function of time and stochastic error sources. Unlike modifications to the nominal trajectory, modifications to the dispersion covariance analysis do not require the propagation of a new nominal trajectory. However, modifications to the dispersion covariance require repropagation of the accumulated process noise covariance, which will be discussed later. As a result, a new dispersion covariance analysis happens quickly when only testing different times to perform dispersion covariance modifying events (initial dispersion, TCM execution time, maneuver execution error values).

Chapter 5 presents a fast linear covariance analysis-based method of TCM optimization along a nominal trajectory that only requires a series of matrix multiplications versus a new trajectory propagation. Following the description of the TCM optimization method, a series of examples illustrates the dependence of the optimal TCM location and TCM δV on the magnitude of specific stochastic parameters. The stochastic parameters considered are an initial state dispersion, nominal maneuver execution error, TCM execution error, and random disturbances (also commonly known as process noise).

5.1 Modeling Error Sources

5.1.1 Initial State Dispersion

An initial state dispersion represents errors in the trajectory initial state, such as orbit insertion error. The initial state dispersion is a zero mean multivariate normal distribution

with position variance σ_r^2 and velocity variance σ_v^2 in each direction:

$$P_0 = E\left[\delta\boldsymbol{x}\left(t_0\right)\delta\boldsymbol{x}\left(t_0\right)^\top\right] = \begin{bmatrix} \sigma_r^2 I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \sigma_v^2 I_{3\times3} \end{bmatrix} \tag{5.1}$$

The dispersion along a nominal trajectory is propagated linearly from an initial time t_0 to an arbitrary time t_c using the STM

$$\delta\boldsymbol{x}(t_c) = \Phi(t_c, t_0)\delta\boldsymbol{x}(t_0) \tag{5.2}$$

Similarly, the state dispersion covariance is propagated linearly along the nominal trajectory to the next dispersion covariance modifying event:

$$P_c^- = E\left[\delta\boldsymbol{x}(t_c)\delta\boldsymbol{x}(t_c)^\top\right] = \Phi\left(t_c, t_0\right)P_0\Phi\left(t_c, t_0\right)^\top \tag{5.3}$$

5.1.2 Trajectory Correction Maneuvers without Execution Error

The purpose of a TCM is to modify the dispersion at a target event or time. In this analysis, the target is an upcoming nominal maneuver. Two types of TCMs are implemented: a position dispersion-correcting TCM (Equation 5.5) and a velocity dispersion-correcting TCM (Equation 5.10). Subsection 4.1.3 presented a more detailed derivation of the following equations.

The dispersion at t_c along a nominal trajectory is propagated linearly using the STM to the target at time t_n:

$$\delta\boldsymbol{x}(t_n) = \Phi(t_n, t_c)\delta\boldsymbol{x}(t_c) \tag{5.4}$$

A position targeting TCM, $\delta\boldsymbol{V}_r$, is designed to remove the position dispersion at a future target ($\delta\boldsymbol{r}(t_n) = 0$) via a velocity modification

$$\delta\boldsymbol{V}_r = \begin{bmatrix} -\Phi_{rv}(t_n, t_c)^{-1}\Phi_{rr}(t_n, t_c) & -I_{3\times3} \end{bmatrix}\delta\boldsymbol{x}(t_c) = T\delta\boldsymbol{x}(t_c) \tag{5.5}$$

where t_c represents TCM execution time and t_n represents the target time along a nominal trajectory. The covariance of $\delta \boldsymbol{V}_r$ is a function of the matrix T and the dispersion covariance at t_c prior to the TCM, P_c^-:

$$E\left[\delta \boldsymbol{V}_r \delta \boldsymbol{V}_r^\top\right] = TE\left[\delta \boldsymbol{x} \delta \boldsymbol{x}^\top\right] T^\top = TP_c^- T^\top \tag{5.6}$$

An upper bound for the variance of the magnitude of a TCM is the RSS of the TCM covariance:

$$\sigma_{\delta V_r} = \sqrt{\mathrm{tr}\left(TP_c^- T^\top\right)} = \sqrt{\mathrm{tr}\left(P_{TCM_r}\right)} \tag{5.7}$$

The velocity change to the state dispersion from a TCM is

$$\delta \boldsymbol{x}_c^+ = \delta \boldsymbol{x}_c^- + \begin{bmatrix} 0_{3\times 6} \\ T \end{bmatrix} \delta \boldsymbol{x}_c^- \tag{5.8}$$

When modeling the state dispersion covariance, the TCM modifies the velocity dispersion covariance rather than modifying the nominal state. The post-correction dispersion covariance follows

$$P_c^+ = E\left[\left(\delta \boldsymbol{x}_c^- + N\delta \boldsymbol{x}_c^-\right)\left(\delta \boldsymbol{x}_c^- + N\delta \boldsymbol{x}_c^-\right)^\top\right] = \left(I+N\right) P_c^- \left(I+N\right)^\top \tag{5.9}$$

where I is the identity matrix and $N = \begin{bmatrix} 0_{3\times 6} \\ T \end{bmatrix}$

A velocity dispersion-correcting TCM directly corrects remaining velocity dispersion at a final time or time of the nominal maneuver, $\delta \boldsymbol{V}_v = -M_v \delta \boldsymbol{x}\left(t_n\right)$.

$$\sigma_{\delta V_v} = \sqrt{\mathrm{tr}\left(M_v P_n^- M_v^\top\right)} = \sqrt{\mathrm{tr}\left(P_{TCM_v}\right)} \tag{5.10}$$

where M_v is a 3×6 mapping matrix that extracts the velocity dispersion covariance. Only the cost of cleaning up the final position and velocity dispersion is included in the analysis.

The actual final covariance update is not implemented as injection into the target orbit marks the end of the trajectory design.

5.1.3 Trajectory Correction Maneuvers with Execution Error

Realistically, TCMs do not perfectly mitigate a future position dispersion as they are not perfectly executed. TCM execution error is incorporated simultaneously with the TCM dispersion covariance update. The execution error model is normally distributed, zero mean, $\sigma^2_{R_{TCM}}$ variance per axis, resulting in the following TCM dispersion covariance update with maneuver execution error:

$$P_c^+ = (I + N)\, P_c^-\, (I + N)^\top + G R_{TCM} G \tag{5.11}$$

where $R_{TCM} = \sigma^2_{R_{TCM}} I_{3\times 3}$ is the TCM execution error covariance and G is a 6×3 matrix that maps the error to the velocity covariance sub-matrix. A more realistic model per a specific mission's hardware parameters can be incorporated.

The TCM execution error also contributes to the stochastic cost of the TCM:

$$\sigma_{\delta V} = \sqrt{\mathrm{tr}\,(P_{TCM} + R_{TCM})} \tag{5.12}$$

5.1.4 Nominal Maneuver Execution Error

Nominal maneuvers along the nominal trajectory are also not executed perfectly and the nominal maneuver execution error contributes to the dispersion covariance at the time of execution. The dispersion covariance update equation adds the maneuver execution error at the time of the nominal maneuver. The error model is also zero mean with a variance of $\sigma^2_{R_{\Delta V}}$ per axis where $R_{\Delta V} = \sigma^2_{R_{\Delta V}} I_{3\times 3}$ is the nominal maneuver execution error covariance. The dispersion covariance update equation for a nominal maneuver is

$$P^+_{\Delta V} = P^-_{\Delta V} + G R_{\Delta V} G^\top \tag{5.13}$$

5.1.5 Corrected Nominal Maneuvers

Another option for affecting the future state dispersion covariance is combining a TCM with a nominal maneuver. Assuming the nominal maneuver is not performed at a location where TCMs are ill-conditioned (e.g., correcting out of plane dispersion during the first maneuver in a Hohmann transfer), the effect of combining vector magnitudes of both the nominal maneuver and the TCM results in savings in many cases. When a nominal maneuver, $\Delta \boldsymbol{V}$, and a TCM, $\delta \boldsymbol{V}$, are combined in a single maneuver, the corrected nominal maneuver vector, $\Delta \boldsymbol{V}_{cr}$, is

$$\Delta \boldsymbol{V}_{cr} = \Delta \boldsymbol{V} + \delta \boldsymbol{V}_r$$

where the individual $\delta \boldsymbol{V}_r$ components are zero mean Gaussian random variables with covariance matrix P_{TCM_r}. A first-order TSE of the magnitude of the corrected position-targeting maneuver, $\|\Delta \boldsymbol{V}_{cr}\|$, about the nominal maneuver vector, helps to estimate the statistics of the correction magnitude $\|\delta \boldsymbol{V}\|$:

$$\|\Delta \boldsymbol{V}_{cr}\| \approx \|\Delta \boldsymbol{V}\| + \frac{\partial \|\Delta \boldsymbol{V}\|}{\partial \Delta \boldsymbol{V}} \delta \boldsymbol{V}_r \tag{5.14}$$

The mean of $\Delta \boldsymbol{V}_{cr}$ is $\Delta \boldsymbol{V}$ and the variance is

$$\sigma^2_{\|\Delta \boldsymbol{V}_{cr}\|} = E\left[\left(\|\Delta \boldsymbol{V}_{cr}\| - \|\Delta \boldsymbol{V}\| \right) \left(\|\Delta \boldsymbol{V}_{cr}\| - \|\Delta \boldsymbol{V}\| \right)^\top \right] \tag{5.15}$$

$$\sigma^2_{\|\Delta \boldsymbol{V}_{cr}\|} = E\left[\left(\hat{i}^\top_{\Delta \boldsymbol{V}} \delta \boldsymbol{V}_r \right) \left(\hat{i}^\top_{\Delta \boldsymbol{V}} \delta \boldsymbol{V}_r \right)^\top \right] \tag{5.16}$$

$$\sigma^2_{\|\Delta \boldsymbol{V}_{cr}\|} = \hat{i}^\top_{\Delta \boldsymbol{V}} \left(P_{TCM} + R_{\Delta V} \right) \hat{i}_{\Delta \boldsymbol{V}} \tag{5.17}$$

Similarly, the target orbit arrival $\Delta \boldsymbol{V}$ can be combined with a correction to remove remaining velocity dispersion at the target orbit insertion. The statistics of the velocity dispersion cleanup components are represented by the 3×3 velocity submatrix of the state dispersion at target orbit insertion:

$$P_{TCM_v} = M_v P_n^- M_v^\top \tag{5.18}$$

The first-order TSE produces a similar result to Equation 5.17:

$$\sigma^2_{\|\Delta \boldsymbol{V}_{cv}\|} = \hat{i}^\top_{\Delta \boldsymbol{V}} \left(P_{TCM_v} + R_{\Delta V} \right) \hat{i}_{\Delta \boldsymbol{V}} \tag{5.19}$$

One way to interpret this simplification is vector addition: the only portion of the TCM that adds cost to the combined maneuver is the portion in the direction of the nominal maneuver. This savings is only accurate in cases where the nominal maneuver is much larger than the correction magnitude. When the reverse is true, this model's perceived savings are unrealistic.

5.2 Random Disturbances / Process Noise

Incorporating process noise is a common technique for incorporating the effect of continuous stochastic errors of a process. Spacecraft venting or misaligned reaction control subsystem thrusters result in random impulses that may have a cumulative effect on the trajectory. These random impulses can also be represented by an increase to the dispersion covariance with respect to the nominal trajectory. A simple model for random disturbances is a zero mean white continuous noise process with power spectral density (PSD) Q. The effect of Q accumulates over time via the linear stochastic differential equation [49]:

$$\bar{Q}(t_f, t_0) = \int_{t_i}^{t_f} \Phi(t_f, \tau) G(\tau) Q(\tau) G(\tau)^\tau \Phi^\top(t_f, \tau) d\tau \tag{5.20}$$

where $\bar{Q}$ represents the accumulated state dispersion covariance due to process noise, and G maps Q to the velocity components of the state dispersion covariance matrix. In the current implementation, Q and G are constant so their time reference τ may be removed. $\bar{Q}$ will be referred to as the Q bar matrix, or QBM.

The upcoming TCM optimization method in Section 5.5 references propagated STM histories along a nominal trajectory many times without repropagating when testing different times to perform TCMs. Similarly, the continuous QBM history is also propagated and saved, from the beginning of each segment, alongside the STM history. Equation 5.21 [49]

is used to numerically integrate the QBM with initial condition $\bar{Q}(t_0, t_0) = 0_{6\times6}$:

$$\dot{\bar{Q}}(t, t_0) = F(t)\bar{Q}(t, t_0) + \bar{Q}(t, t_0)F(t)^\top + GQG^\top \tag{5.21}$$

With the QBM history along the trajectory, propagated from the beginning of each segment, the appropriate portion of $\bar{Q}$ is extracted from the history which contributes to the dispersion covariance at each dispersion covariance modifying event (TCMs, nominal maneuvers with execution error error). For example, at the first correction, the dispersion covariance equals the linear covariance growth via the STM plus the accumulated process noise covariance:

$$P_{c_1}^- = \Phi\left(t_{c_1}, t_0\right) P_0 \Phi\left(t_{c_1}, t_0\right)^\top + \bar{Q}_{t_{c_1}}\left(t_{c_1}, t_0\right) \tag{5.22}$$

where the STM (Φ) and the QBM ($\bar{Q}$) time-histories have been previously saved.

Some manipulation of the QBM history is required as optimizing TCMs along a nominal trajectory allows TCMs to occur independent of segment intersection, and the QBM history is propagated from the beginning to the end of each segment. The first manipulation involves combining two sequential propagated durations of $\bar{Q}$. $\bar{Q}_{t_1}\left(t_1, t_0\right)$ represents the effect of process noise covariance at t_1, accumulated from t_0 to t_1. Similarly, $\bar{Q}_{t_2}\left(t_2, t_1\right)$ represents the effect of process noise covariance at t_2, accumulated from t_1 to t_2. The goal is to combine these two sequential portions of $\bar{Q}$ to form the accumulated effect of process noise from t_0 to t_2 on the covariance at t_2, $\bar{Q}_{t_2}\left(t_2, t_0\right)$. At first, it may seem that $\bar{Q}_{t_2}\left(t_2, t_0\right)$ is merely the sum of$\bar{Q}_{t_1}\left(t_1, t_0\right)$ and $\bar{Q}_{t_2}\left(t_2, t_1\right)$, however, upon closer examination the required operation is shown in Equation 5.23:

$$\bar{Q}_{t_2}\left(t_2, t_0\right) = \bar{Q}_{t_2}\left(t_2, t_1\right) + \Phi\left(t_2, t_1\right) \bar{Q}_{t_1}\left(t_1, t_0\right) \Phi\left(t_2, t_1\right)^\top \tag{5.23}$$

The second term can be thought of as linear propagation of $\bar{Q}_{t_1}\left(t_1, t_0\right)$ to t_2:

$$\bar{Q}_{t_2}\left(t_1, t_0\right) = \Phi\left(t_2, t_1\right) \bar{Q}_{t_1}\left(t_1, t_0\right) \Phi\left(t_2, t_1\right)^\top \tag{5.24}$$

As propagation of terms of the QBM occurs from the beginning to end of each segment and TCMs are not limited to segment intersections, another QBM manipulation is required. Assuming QBM history propagation begins from t_0 through t_2 and a TCM occurs at t_1, a required quantity is $\bar{Q}_{t_2}(t_2, t_1)$. Rearranging Equation 5.23 produces the desired quantity, shown in Equation 5.25.

$$\bar{Q}_{t_2}(t_2, t_1) = \bar{Q}_{t_2}(t_2, t_0) - \Phi(t_2, t_1)\,\bar{Q}_{t_1}(t_1, t_0)\,\Phi(t_2, t_1)^{\top} \tag{5.25}$$

5.3 Stochastic Cost and Constraints

The stochastic cost represents the additional cost of managing the state dispersion along a nominal trajectory to meet mission requirements. The stochastic cost is the sum of the RSS of $n_{\delta V}$ TCMs with execution error plus the additional cost incurred by correcting $n_{\Delta V_c}$ nominal maneuvers, multiplied by a scalar that corresponds to the mission's risk tolerance related to maneuver margin (3σ is used in this analysis):

$$J_\sigma = 3\left(\sum_{k=1}^{n_{\delta V}}(\sigma_{\delta V}) + \sum_{q=1}^{n_{\Delta V_c}}\sigma_{\|\Delta \boldsymbol{V}_c\|}\right) \tag{5.26}$$

An important parameter at nominal mission events (e.g., nominal maneuvers, target orbit insertion, rendezvous) is spacecraft position relative to the nominal trajectory. The state dispersion covariance relative to the nominal trajectory represents the statistical deviation from the planned trajectory. TCMs manage the state dispersion within a reasonable level around the nominal trajectory. Said another way, TCMs ensure that the state dispersion constraints are met. The main state dispersion constraint ensures that the RSS of the position dispersion is less than a maximum value, $\sigma_{r,max}$:

$$\sqrt{\mathrm{tr}\left(M_r P M_r^{\top}\right)} \leq \sigma_{r,max} \tag{5.27}$$

In general, Constraint Equation 5.27 is applied in this analysis at all nominal maneuvers after the first nominal maneuver.

5.4 Analytical Gradients

The design method in this paper propagates, computes, and provides analytical gradients of the cost function and constraints with respect to the problem parameters, $\boldsymbol{S}$, to the NLP. Many stochastic cost terms involve STMs, which require the propagation of second-order STTs to compute the gradient analytically. Similar to propagating second-order STTs, the gradient of QBM terms requires propagating a tensor that characterizes the state sensitivity of the accumulated process noise covariance, $\frac{\partial \bar{Q}(t_1,t_0)}{\partial \boldsymbol{x}_{0,i}}$. This will be referred to as the Q bar tensor, or QBT. This section provides an overview of two gradient formulations: first, a sequential method for assembling the gradient of multiple TCMs and second, the propagation and manipulation of the QBT.

5.4.1 Multiple TCM Cost Analytical Gradient

The partial derivative of the deterministic cost, deterministic constraints, a single TCM RSS magnitude, and the target position dispersion covariance constraint are derived in Chapter 4.1. This section incorporates multiple TCMs, maneuver execution error, and process noise. The approach used to calculate the gradient of multiple TCMs follows.

The total stochastic cost, J_σ, is the summation of the cost of numerous TCMs. Correspondingly, when multiple TCMs are performed along a nominal trajectory, each at t_{c_k} with execution error covariance R_{TCM}, the total TCM RSS $\sigma_{\delta V}$ gradient requires gradients of each individual position-correcting TCM, $\sigma_{\delta V_{r,k}}$, plus the gradient of the final velocity dispersion correction $\sigma_{\delta V_v}$. However, modifications to TCMs have impacts to the dispersion covariance that affect the gradient of subsequent TCMs. The sequential nature of the following formulation maintains the sensitivity of each TCM to modifications in elements that have an impact on the preceding dispersion covariance. Said differently, each TCM affects the dispersion covariance at all future TCMs. The following sequential, nested covariance sensitivity formulation simplifies this operation.

The first TCM, $k = 1$, mirrors the single TCM scenario in Section 4.1 with the addition of R_{TCM}. The RSS of the first TCM, $\sigma_{\delta V_{r,1}}$, is

$$\sigma_{\delta V_{r,1}} = \sqrt{\text{tr}\left(T_1 P_{c_1}^- T_1^\top + R_{TCM}\right)} \tag{5.28}$$

The post-TCM covariance is defined by Equation 5.11. To simplify notation, the gradient of $\sigma_{\delta V_{r,1}}$ with respect to a segment initial state will be represented by the function $\mathcal{D}$, where $\mathcal{D}$ is a result of the application of matrix differentiation rules (product rule, exponent rule, etc.) on Equation 5.28:

$$\frac{\partial \sigma_{\delta V_{r,1}}}{\partial \boldsymbol{x}_{0,i}} = \mathcal{D}\left(T_1, P_{c_1}^-, R_{TCM}, \frac{\partial T_1}{\partial \boldsymbol{x}_{0,i}}, \frac{\partial P_{c_1}^-}{\partial \boldsymbol{x}_{0,i}}\right) \tag{5.29}$$

Evaluating $\frac{\partial T_1}{\partial \boldsymbol{x}_{0,i}}$ and $\frac{\partial P_{c_1}^-}{\partial \boldsymbol{x}_{0,i}}$ are again both applications of the product rule (and expanded in more detail in Section 4.1.4 and in [35]). The sub-terms that involve the partial derivative of STM segments $\frac{\partial \Phi}{\partial \boldsymbol{x}}$ leverage propagated STT terms. Manipulation to obtain the appropriate STT endpoints is described in Subsection 2.3.2.

The second TCM ($k = 2$) now incorporates the effects of TCM 1 to the next covariance update $P_{c_2}^-$; assuming the next covariance update is another TCM:

$$\sigma_{\delta V_{r,2}} = \sqrt{\text{tr}\left(T_2 P_{c_2}^- T_2^\top + R_{TCM}\right)} \tag{5.30}$$

$$\begin{aligned} P_{c_2}^- &= \Phi\left(t_{c_2}, t_{c_1}\right) P_{c_1}^+ \Phi\left(t_{c_2}, t_{c_1}\right)^\top \\ &= \Phi\left(t_{c_2}, t_{c_1}\right) \left(\left(I + N_1\right) P_{c_1}^- \left(I + N_1\right)^\top + G R_{TCM} G^\top\right) \Phi\left(t_{c_2}, t_{c_1}\right)^\top \end{aligned} \tag{5.31}$$

The state sensitivity of the dispersion before TCM 2, $\frac{\partial P_{c_2}^-}{\partial \boldsymbol{x}_{0,i}}$, can be expressed as another function $\mathcal{J}$, shortening another lengthy application of matrix differentiation rules:

$$\frac{\partial P_{c_2}^-}{\partial \boldsymbol{x}_{0,i}} = \mathcal{J}\left(T_1, \frac{\partial T_1}{\partial \boldsymbol{x}_{0,i}}, P_{c_1}^-, \frac{\partial P_{c_1}^-}{\partial \boldsymbol{x}_{0,i}}, \Phi\left(t_{c_2}, t_{c_1}\right), \frac{\partial \Phi\left(t_{c_2}, t_{c_1}\right)}{\partial \boldsymbol{x}_{0,i}}, R_{TCM}\right) \tag{5.32}$$

Now $\frac{\partial P_{c2}^{-}}{\partial \boldsymbol{x}_{0,i}}$ is expressed as a function of gradients at TCM 1 and the dynamics between TCMs 1 and 2. Calculating the dispersion covariance sensitivities sequentially enables subsequent TCMs and their gradients to be expressed as a function of values at the previous dispersion covariance update.

5.4.2 Accumulated Process Noise Covariance Gradient

When including the effect of process noise and evaluating $\frac{\partial P_{c_1}^{-}}{\partial \bar{x}_{0,i}}$, where $P_{c_1}^{-}$ now includes $\bar{Q}$

$$P_{c_1}^{-} = \Phi\left(t_{c_1}, t_0\right) P_0 \Phi\left(t_{c_1}, t_0\right)^{\top} + \bar{Q}_{t_{c_1}}\left(t_{c_1}, t_0\right) \tag{5.33}$$

evaluating $\frac{\partial \bar{Q}_{t_{c_1}}\left(t_{c_1}, t_0\right)}{\partial \boldsymbol{x}_{0,i}}$, the QBT, becomes a required term. These QBM state sensitivities are obtained via numerical integration of Equation 5.34 below, with a brief derivation following. The first step involves taking the partial derivative of $\dot{\bar{Q}}$ (Equation 5.21) with respect to the initial state. Applying the product rule results in Equation 5.34 which is numerically integrated to obtain the QBT, similar to an STT, with initial conditions $\frac{\partial \bar{Q}(t_0,t_0)}{\partial \boldsymbol{x}(t_0)} = 0_{6\times 6\times 6}$:

$$\begin{aligned}\left[\frac{\partial \dot{\bar{Q}}(t,t_0)}{\partial \boldsymbol{x}(t_0)}\right]_{ijk} &= \frac{\partial F(t)_{i,m}}{\partial \boldsymbol{x}(t_0)_k}\bar{Q}(t,t_0)_{mj} + F(t)_{i,m}\frac{\partial \bar{Q}(t,t_0)_{mj}}{\partial \boldsymbol{x}(t_0)_k} \\ &+ \frac{\partial F(t)_{jm}}{\partial \boldsymbol{x}(t_0)_k}\bar{Q}(t,t_0)_{im} + F(t)_{j,m}\frac{\partial \bar{Q}(t,t_0)_{im}}{\partial \boldsymbol{x}(t_0)_k}\end{aligned} \tag{5.34}$$

where $F\left(t\right)_{i,j}$ is the system Jacobian. A chain rule application utilizes the second derivative of the system dynamics with respect to a previous state, $F\left(t\right)_{i,jk}$:

$$\frac{\partial F(t)_{ij}}{\partial \boldsymbol{x}(t_0)_k} = \frac{\partial F(t)_{ij}}{\partial \boldsymbol{x}(t)_m}\frac{\partial \boldsymbol{x}(t)_m}{\partial \boldsymbol{x}(t_0)_k} = F(t)_{i,jm}\Phi(t,t_0)_{m,k} \tag{5.35}$$

Similar to the QBM, some manipulation of the QBT is also required to obtain the appropriate QBT endpoints and its effect at the appropriate time. Finding the equation for sequential QBT combination (e.g., assembling $\frac{\partial \bar{Q}_{t_2}(t_2,t_0)}{\partial \boldsymbol{x}(t_0)}$ from $\frac{\partial \bar{Q}_{t_1}(t_1,t_0)}{\partial \boldsymbol{x}(t_0)}$ and $\frac{\partial \bar{Q}_{t_2}(t_2,t_1)}{\partial \boldsymbol{x}(t_1)}$)

involves taking the partial derivative of Equation 5.23, which equals the following after two chain rule applications and term collection:

$$\frac{\partial \bar{Q}_{t_2}\left(t_2,t_0\right)}{\partial \boldsymbol{x}(t_0)} = W_{ijm}\Phi\left(t_1,t_0\right)_{m,k} + \Phi\left(t_2,t_1\right)_{i,m}\frac{\partial \bar{Q}_{t_1}\left(t_1,t_0\right)_{mn}}{\partial \boldsymbol{x}(t_0)_k}\Phi\left(t_2,t_1\right)_{j,n} \tag{5.36}$$

where

$$\begin{aligned} W_{ijm} = &\frac{\partial \bar{Q}_{t_2}\left(t_2,t_1\right)_{ij}}{\partial \boldsymbol{x}(t_1)_m} + \Phi_{II}\left(t_2,t_1\right)_{i,nm}\bar{Q}_{t_1}\left(t_1,t_0\right)_{np}\Phi\left(t_2,t_1\right)_{j,p} \\ &+ \Phi\left(t_2,t_1\right)_{i,n}\bar{Q}_{t_1}\left(t_1,t_0\right)_{np}\Phi_{II}\left(t_2,t_1\right)_{j,pm} \end{aligned} \tag{5.37}$$

Subtracting a contribution involves taking the partial derivative of Equation 5.25, performing a product rule and two chain rule applications, and collecting terms:

$$\begin{aligned} \frac{\partial \bar{Q}_{t_2}\left(t_2,t_1\right)_{ij}}{\partial \boldsymbol{x}(t_1)_k} = &D_{ijm}\Phi\left(t_0,t_1\right)_{m,k} - \Phi_{II}\left(t_2,t_1\right)_{i,nm}\bar{Q}_{t_1}\left(t_1,t_0\right)_{np}\Phi\left(t_2,t_1\right)_{j,p} \\ &- \Phi\left(t_2,t_1\right)_{i,n}\bar{Q}_{t_1}\left(t_1,t_0\right)_{np}\Phi_{II}\left(t_2,t_1\right)_{j,pm} \end{aligned} \tag{5.38}$$

where

$$D_{ijm} = \frac{\partial \bar{Q}_{t_2}\left(t_2,t_0\right)_{ij}}{\partial \boldsymbol{x}(t_0)_m} - \Phi\left(t_2,t_1\right)_{i,n}\frac{\partial \bar{Q}_{t_1}\left(t_1,t_0\right)_{np}}{\partial \boldsymbol{x}(t_0)_m}\Phi\left(t_2,t_1\right)_{j,p} \tag{5.39}$$

5.4.3 Gradient Application with Independent TCMs & Segment Intersections

The goal of this section is deriving the gradient of the accumulated process noise covariance at each TCM (or any other covariance-modifying event along a nominal trajectory), $\bar{Q}_k$, with respect to segment initial states $\boldsymbol{x}_{0,i}$, when TCMs do not occur at segment intersections (e.g., in the middle of a segment). There are four relevant time events that occur and four potential sequences of these events, each with a unique logic case. Two of the time events are the beginning and end of segment i, $t_{0,i}$ and $t_{f,i}$. The other two time events are the time the kth TCM is performed (t_k) and the time the $k-1$th TCM is performed (t_{k-1}).

The four potential sequences of events with nonzero gradients and the gradient derivation $\frac{\partial \bar{Q}_k(t_k,t_{k-1})}{\partial \boldsymbol{x}_{0,i}}$ for each sequence follows.

1. $t_{0,i} \leq t_{k-1} \leq t_{f,i} \leq t_k$

   For Case 1, TCM $k-1$ occurs in segment i and TCM k occurs after segment i. An expression for the process noise covariance at t_k, accumulated from t_{k-1} to t_k, intentionally separated into portions that are sensitive to modifications in $\boldsymbol{x}_{0,i}$ are those that are not, is

$$\bar{Q}_k\left(t_k, t_{k-1}\right) = \bar{Q}_k\left(t_k, t_{f,i}\right) + \bar{Q}_k\left(t_{f,i}, t_{k-1}\right) \tag{5.40}$$

   where manipulation is required to represent $\bar{Q}$, as propagated, from the beginning of each segment:

$$\begin{aligned}\bar{Q}_k\left(t_{f,i}, t_{k-1}\right) &= \Phi\left(t_k, t_{f,i}\right) \bar{Q}_{f,i}\left(t_{f,i}, t_{k-1}\right) \Phi\left(t_k, t_{f,i}\right)^\top \\ &= \Phi\left(t_k, t_{f,i}\right) \left(\bar{Q}_{f,i}\left(t_{f,i}, t_{0,i}\right) - \ldots \right. \\ &\left. \Phi\left(t_{f,i}, t_{k-1}\right) \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right) \Phi\left(t_{f,i}, t_{k-1}\right)^\top\right) \Phi\left(t_k, t_{f,i}\right)^\top \end{aligned} \tag{5.41}$$

   The gradient of $\bar{Q}_k\left(t_k, t_{k-1}\right)$ is

$$\frac{\partial \bar{Q}_k\left(t_k, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} = \frac{\partial \bar{Q}_k\left(t_k, t_{f,i}\right)}{\partial \boldsymbol{x}_{0,i}} + \frac{\partial \bar{Q}_k\left(t_{f,i}, t_{k-1}\right)}{\boldsymbol{x}_{0,i}} \tag{5.42}$$

   Since $\bar{Q}_k\left(t_k, t_{f,i}\right)$ and $\boldsymbol{x}_{0,i}$ are in different segments, the first term of the partial derivative in Equation 5.42 is zero.

$$\frac{\partial \bar{Q}_k\left(t_k, t_{f,i}\right)}{\partial \boldsymbol{x}_{0,i}} = 0_{6\times 6\times 6} \tag{5.43}$$

The second term of Equation 5.42 is

$$\frac{\partial \bar{Q}_k\left(t_{f,i}, t_{k-1}\right)}{\boldsymbol{x}_{0,i}} = \Phi\left(t_k, t_{f,i}\right) \frac{\partial \bar{Q}_{f,i}\left(t_{f,i}, t_{k-1}\right)}{\boldsymbol{x}_{0,i}} \Phi\left(t_k, t_{f,i}\right)^\top \tag{5.44}$$

which, when combined with the expansion in Equation 5.41, yields

$$\begin{aligned}
\frac{\partial \bar{Q}_{f,i}\left(t_{f,i}, t_{k-1}\right)}{\boldsymbol{x}_{0,i}} = & \frac{\partial \bar{Q}_{f,i}\left(t_{f,i}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} \\
& - \frac{\partial \Phi\left(t_{f,i}, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right) \Phi\left(t_{f,i}, t_{k-1}\right)^\top \\
& - \Phi\left(t_{f,i}, t_{k-1}\right) \frac{\partial \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} \Phi\left(t_{f,i}, t_{k-1}\right)^\top \\
& - \Phi\left(t_{f,i}, t_{k-1}\right) \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right) \frac{\partial \Phi\left(t_{f,i}, t_{k-1}\right)^\top}{\partial \boldsymbol{x}_{0,i}}
\end{aligned} \tag{5.45}$$

2. $t_{0,i} \leq t_{k-1} \leq t_k < t_{f,i}$

   For case 2, $k-1$ and TCM k occur in the same segment. The form of the solution changes as the portion of the trajectory from t_k to $t_{f,i}$ is not relevant to $\bar{Q}_k$. As a result

$$\bar{Q}_k\left(t_k, t_{k-1}\right) = \bar{Q}_k\left(t_k, t_{0,i}\right) - \Phi\left(t_k, t_{k-1}\right) \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right) \Phi\left(t_k, t_{k-1}\right)^\top \tag{5.46}$$

$$\begin{aligned}
\frac{\partial \bar{Q}_k\left(t_k, t_{k-1}\right)}{\boldsymbol{x}_{0,i}} = & \frac{\partial \bar{Q}_k\left(t_k, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} \\
& - \frac{\partial \Phi\left(t_k, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right) \Phi\left(t_k, t_{k-1}\right)^\top \\
& - \Phi\left(t_k, t_{k-1}\right) \frac{\partial \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} \Phi\left(t_k, t_{k-1}\right)^\top \\
& - \Phi\left(t_k, t_{k-1}\right) \bar{Q}_{k-1}\left(t_{k-1}, t_{0,i}\right) \frac{\partial \Phi\left(t_k, t_{k-1}\right)^\top}{\partial \boldsymbol{x}_{0,i}}
\end{aligned} \tag{5.47}$$

3. $t_{k-1} < t_{0,i} \leq t_{f,i} \leq t_k$

   For case 3, TCM $k-1$ occurs prior to the start of the ith segment and TCM k occurs after the end of the ith segment. The resulting total $\bar{Q}_k\left(t_k, t_{k-1}\right)$ is separated into

the pieces that are part of the ith segment and and those that aren't part of the ith segment:

$$\bar{Q}_k\left(t_k, t_{k-1}\right) = \bar{Q}_k\left(t_k, t_{f,i}\right) + \bar{Q}_k\left(t_{f,i}, t_{0,i}\right) + \bar{Q}_k\left(t_{0,i}, t_{k-1}\right) \tag{5.48}$$

The partial derivative of $\bar{Q}_k\left(t_k, t_{f,i}\right)$ with respect to $\boldsymbol{x}_{0,i}$ is zero. The partial is then

$$\frac{\partial \bar{Q}_k\left(t_k, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} = \frac{\partial \bar{Q}_k\left(t_{f,i}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} + \frac{\partial \bar{Q}_k\left(t_{0,i}, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} \tag{5.49}$$

For $\frac{\partial \bar{Q}_k\left(t_{f,i}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}}$:

$$\bar{Q}_k\left(t_{f,i}, t_{0,i}\right) = \Phi\left(t_k, t_{f,i}\right) \bar{Q}_{f,i}\left(t_{f,i}, t_{0,i}\right) \Phi\left(t_k, t_{f,i}\right)^\top \tag{5.50}$$

$$\frac{\partial \bar{Q}_k\left(t_{f,i}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} = \Phi\left(t_k, t_{f,i}\right) \frac{\partial \bar{Q}_{f,i}\left(t_{f,i}, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} \Phi\left(t_k, t_{f,i}\right)^\top \tag{5.51}$$

and for $\frac{\partial \bar{Q}_k\left(t_{0,i}, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}}$:

$$\begin{aligned} \bar{Q}_k\left(t_{0,i}, t_{k-1}\right) &= \Phi\left(t_k, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi\left(t_k, t_{0,i}\right)^\top \\ &= \Phi\left(t_k, t_{f,i}\right) \Phi\left(t_{f,i}, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi\left(t_{f,i}, t_{0,i}\right)^\top \Phi\left(t_k, t_{f,i}\right)^\top \end{aligned} \tag{5.52}$$

the partial is

$$\begin{aligned} \frac{\partial \bar{Q}_k\left(t_{0,i}, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} = \Phi\left(t_k, t_{f,i}\right) &\left(\Phi_{II}\left(t_{f,i}, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi\left(t_{f,i}, t_{0,i}\right)^\top + \ldots\right. \\ &\left. \Phi\left(t_{f,i}, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi_{II}\left(t_{f,i}, t_{0,i}\right)^\top\right) \Phi\left(t_k, t_{f,i}\right)^\top \end{aligned} \tag{5.53}$$

4. $t_{k-1} < t_{0,i} \leq t_k < t_{f,i}$

For case 4, TCM $k-1$ occurs prior to the start of the ith segment and TCM k occurs during the ith segment. Similar to case 2, the trajectory between t_k and $t_{f,i}$ does not impact $\bar{Q}_k\left(t_k, t_{k-1}\right)$:

$$\bar{Q}_k\left(t_k, t_{k-1}\right) = \bar{Q}_k\left(t_k, t_{0,i}\right) + \bar{Q}_k\left(t_{0,i}, t_{k-1}\right) \tag{5.54}$$

$$\frac{\partial \bar{Q}_k\left(t_k, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} = \frac{\partial \bar{Q}_k\left(t_k, t_{0,i}\right)}{\partial \boldsymbol{x}_{0,i}} + \frac{\partial \bar{Q}_k\left(t_{0,i}, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} \tag{5.55}$$

The first term is pulled directly from the numerically integrated $\partial\bar{Q}$ history, from the beginning of the ith segment to t_k. The second term is

$$\bar{Q}_k\left(t_{0,i}, t_{k-1}\right) = \Phi\left(t_k, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi\left(t_k, t_{0,i}\right)^\top \tag{5.56}$$

and its partial derivative is

$$\frac{\partial \bar{Q}_k\left(t_{0,i}, t_{k-1}\right)}{\partial \boldsymbol{x}_{0,i}} = \Phi_{II}\left(t_k, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi\left(t_k, t_{0,i}\right)^\top \tag{5.57}$$

$$+ \Phi\left(t_k, t_{0,i}\right) \bar{Q}_{0,i}\left(t_{0,i}, t_{k-1}\right) \Phi_{II}\left(t_k, t_{0,i}\right)^\top \tag{5.58}$$

For the remaining cases ($t_{k-1} \leq t_k \leq t_{0,i} \leq t_{f,i}$ and $t_{0,i} \leq t_{f,i} \leq t_{k-1} \leq t_k$), $\bar{Q}_k$ is independent of the ith segment, therefore the partial derivatives with respect to $\boldsymbol{x}_{0,i}$ are zero.

5.4.4 Accumulated Process Noise Covariance Sensitivity to Segment Duration

The four cases for $\bar{Q}$ state sensitivity do not specifically apply to segment duration sensitivity; the order of $t_{0,i}$ and t_{k-1} do not change the result. The important comparison is whether $t_{f,i}$ occurs after t_{k-1} and is less than or constrained to be equal to t_k: $(t_{k-1} < t_{f,i} \leq t_k)$. Consider $\bar{Q}_k\left(t_k, t_{k-1}\right)$ as the sum of $\bar{Q}$ in two parts, before and after $t_{f,i}$:

$$\bar{Q}_k\left(t_k, t_{k-1}\right) = \bar{Q}_k\left(t_k, t_{f,i}\right) + \bar{Q}_k\left(t_{f,i}, t_{k-1}\right)$$

Modifications to the duration of segment i, Δt_i, have no impact on $\bar{Q}_k\left(t_k, t_{f,i}\right)$. However, modifications to Δt_i do have an impact on $\bar{Q}_k\left(t_{f,i}, t_{k-1}\right)$:

$$\frac{\partial \bar{Q}_k\left(t_k, t_{k-1}\right)}{\partial \Delta t_i} = \Phi\left(t_k, t_{0,i+1}\right) \dot{\bar{Q}}_{f,i}\left(t_{f,i}, t_{k-1}\right) \Phi\left(t_k, t_{0,i+1}\right)^\top \tag{5.59}$$

In this case, Equations 5.21 and 5.59 apply directly. If t_k is less than $t_{f,i}$, the partial derivative with respect to Δt_i is zero. Changing the duration of a segment after the endpoint of the process noise accumulation has no impact on the QBM and the corresponding partial derivative is zero.

5.5 Trajectory Correction Maneuver Optimization

Section 5.5 explores the optimization of TCMs along a fixed nominal trajectory. Subsection 5.5.1 describes a method to optimize the number and location of multiple TCMs along the deterministic optimal trajectory to ensure a dispersion constraint is met at a single target. Subsection 5.5.1 also introduces a second target event along a nominal trajectory and connects the covariance analysis between two separate TCM target portions of the trajectory. Subsection 5.5.2 shows a series of optimal TCM examples along the same nominal trajectory with variations in error sources to highlight the sensitivity of the optimal TCM solution to variations in stochastic problem parameters/error sources. TCM optimization significantly mitigates the total cost of corrections with varying error sources when compared to a default TCM solution (referred to as the "looks about right" solution). The stochastic parameters for the analysis in this section are shown below in Figure 5.1.

5.5.1 Optimizing TCM Number and Location Along a Nominal Trajectory

At any time along a dispersed nominal trajectory between t_0 and t_{final}, it is possible to perform a TCM and affect the dispersion covariance at a future time. Figure 5.1 shows a three impulse LEO to powered lunar flyby (PLF) to NRHO insertion (NRI) trajectory in the CR3BP rotating frame. The stars represent nominal maneuvers. For the TCM optimization example, only the portion of the trajectory between the first and second nominal ΔVs is analyzed (the portion of the trajectory that is plotted with a thicker line and multiple colors in Figure 5.1). The targeted position for corrections in this portion is the position at the PLF (ΔV_2). ΔV_1 is a corrected nominal maneuver.

When an error-free correction is performed without random disturbances, the desired position is achieved. In terms of dispersion covariance, the position dispersion at the tar-

- Initial Dispersion
  - $\sigma_r = 10$ km
  - $\sigma_v = 10$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 1$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km

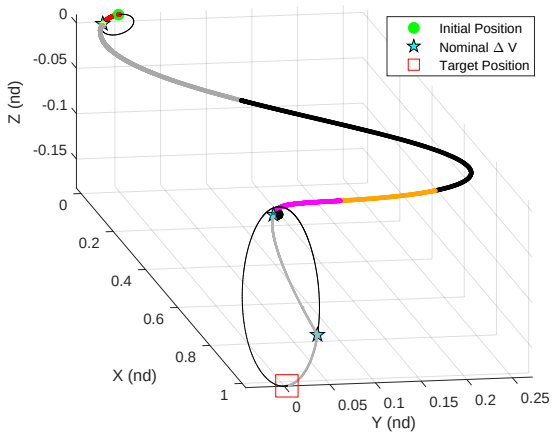
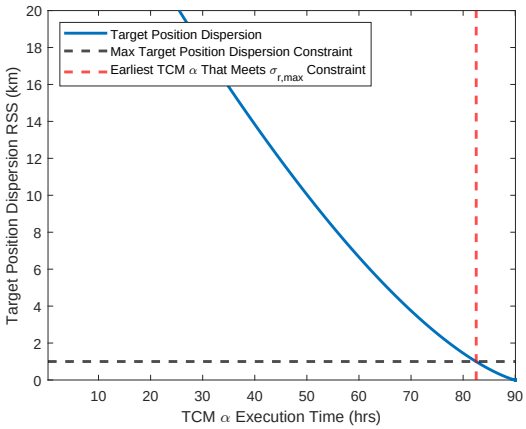


Fig. 5.1: Sample Nominal Trajectory and Error Parameters

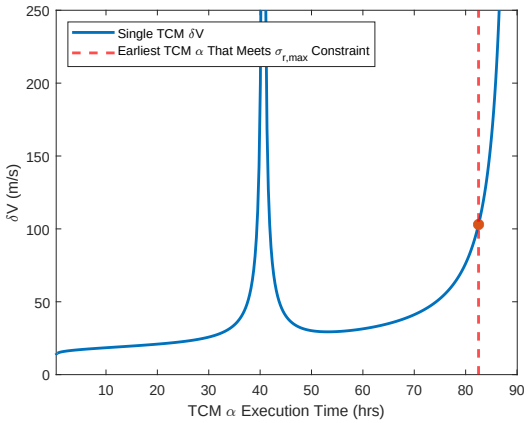
geted time along the nominal trajectory is zero. When an initial dispersion, maneuver execution error, and process noise are introduced, the position dispersion at the target becomes a function of these error sources and the TCM execution time along the nominal trajectory. The blue line in Figure 5.2a shows the target position dispersion RSS as a function of a single TCM execution time. Figure 5.2a also shows a horizontal dotted line representing a 1 km target position dispersion RSS constraint. This constraint restricts the feasible TCM execution time options to those below the dotted horizontal line. A vertical dotted red line identifies the earliest TCM option that meets the target position dispersion constraint, with all later TCM options also meeting the target dispersion constraint.

Figure 5.2b shows the TCM δV as a function of execution time for the same TCM options that produce Figure 5.2a. The red dot represents the lowest δV TCM that meets the target dispersion constraint. Generally, the earliest TCM execution time that meets the target position dispersion constraint is a function of TCM execution error and process noise. With less execution error, the TCM can be performed earlier in the trajectory at a reduced cost. Greater execution error requires a later TCM at greater δV. This TCM to meet the target dispersion constraint will be referred to as TCM α and is an expensive option in the current example performed so late in the trajectory.[1] Introducing an additional TCM (TCM

[1]Greek letters are chosen to mitigate potential confusion associated with assigning numbers to each TCM. TCM α is the first TCM applied in the optimization sequence, but will be the last TCM to be performed along the trajectory.



(a) Target Position Dispersion as a Function of TCM Execution Time

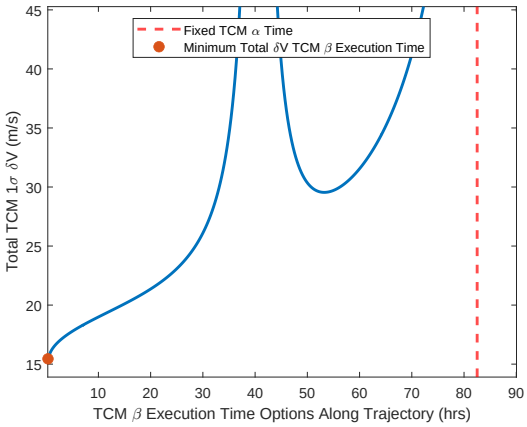


(b) Single TCM RSS as a Function of Execution Time

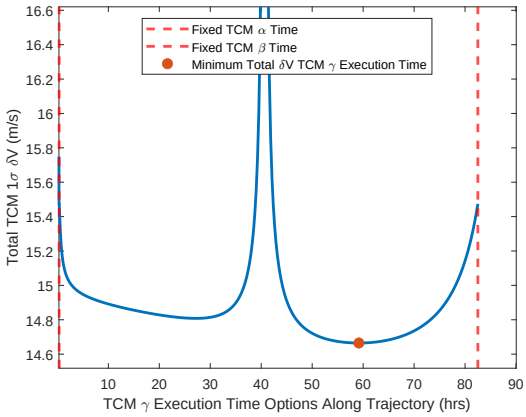
Fig. 5.2: Side-by-side Comparison of Target Position Dispersion RSS and TCM RSS as a Function of TCM Execution Time

β) has the potential to reduce the total δV when compared to the single TCM solution. By fixing the TCM α execution time, TCM β can be performed along the trajectory from t_0 to t_{c_α}. Figure 5.3a shows the two TCM total δV as a function of TCM β execution time and a fixed TCM α execution time (represented by the vertical dotted line). Selecting the TCM β execution time that minimizes the total TCM δV (the red dot in Figure 5.3a) results in the optimal two TCM solution that simultaneously satisfies the target position dispersion constraint.

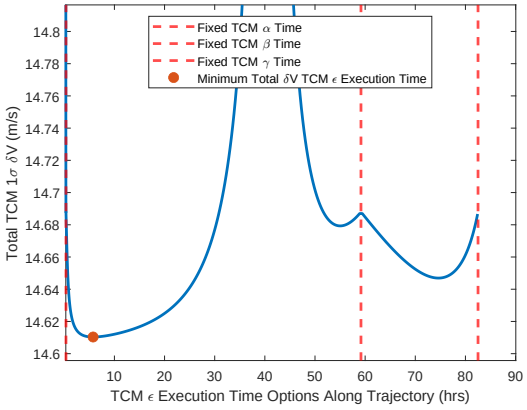
The process can be repeated as the first step in finding the optimal three TCM solution. By fixing the TCM α and β execution times and introducing a third TCM (TCM γ), a TCM γ execution time exists that again corresponds to a minimum total TCM δV. An additional step in this case is required however, as the execution time for TCMs β and γ are variable, affect one another, and affect the total TCM δV. A gradient-based search, step, and re-search until a minimum value is found is implemented. A test in each direction (one time increment earlier and later) of each TCM execution time identifies if a lower total TCM δV solution exists. If so, a gradient vector is created and all TCM execution times are modified one time index in the appropriate direction simultaneously. Figures 5.3b, 5.3c, and 5.3d respectively show the sequence of incorporating three, four, and five TCMs from



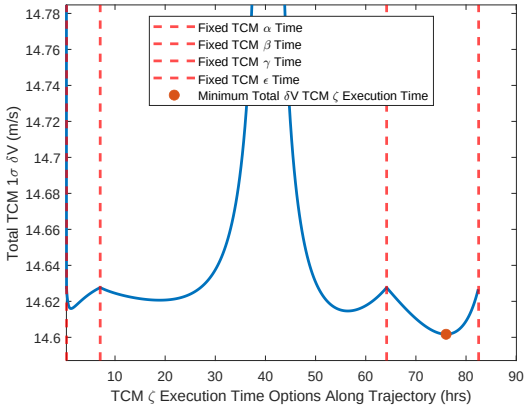
(a) Total Two TCM RSS as a Function of TCM β Execution Time



(b) Total Three TCM RSS as a Function of TCM γ Execution Time



(c) Total Four TCM RSS as a Function of TCM ϵ Execution Time



(d) Total Five TCM RSS as a Function of TCM ζ Execution Time

Fig. 5.3: Sequential Addition and Optimization of TCMs Demonstration

the previous optimal solution identified. Table 5.1 shows the gradient search steps taken from the initial placement of TCM ζ (Figure 5.3d) to the lowest five TCM total δV solution identified.

Iteration	t_β (hrs)	t_γ (hrs)	t_ϵ (hrs)	t_ζ (hrs)	t_α (hrs)	**Total** 1σ δV **(m/s)**
1	0.341831	7.046474	63.633821	76.631727	82.512024	14.600527
5	0.341831	6.693965	61.32633	75.528061	82.512024	14.595893
10	0.341831	6.253329	59.255504	74.679087	82.512024	14.59385
16	0.341831	5.724565	58.072175	74.364653	82.512024	14.593563

Table 5.1: Gradient-Based Five-TCM Optimization Steps

After a certain number of TCMs are added to the maneuver plan, the total TCM δV will start to increase rather than decreasing. This identifies the optimal number of TCMs and their optimal execution time along a nominal trajectory. The data in Table 5.2 identifies that the optimal number of TCMs corresponding to the lowest total δV is five, as increasing to 6 TCMs results in a total cost increase. However, an improvement threshold may be implemented as savings diminish prior to the cost increase. In this analysis, the improvement threshold chosen is 3 times the RSS of R_{TCM}, equating to roughly .05 m/s resulting in four TCMs being chosen despite the marginal additional savings with five TCMs.

The previous analysis only considered TCMs between ΔV_1 (departing LEO) and ΔV_2 (PLF) to introduce the optimization steps. For scenarios with multiple target dispersion constraints (multiple target portions of the trajectory), the optimization sequence is similar with an additional step. The additional first step is placing TCM(s) at the cheapest option to meet each target's position dispersion constraint. For example, in Figure 5.4, there are two TCM targets, ΔV_2 (PLF) and ΔV_3 (NRHO insertion). The first TCM is placed to optimally meet the target position dispersion constraint at ΔV_2 with a single TCM. Similarly, the second TCM is then placed to optimally meet the target position dispersion constraint at ΔV_3 with two total TCMs. The remainder of the process continues as previously described, with eligible options for the new TCMs being across all target portions of the trajectory.

# of TCMs	1	2	3	4	5	6
Optimal TCM 1σ δV **(m/s)**	103.0	15.45	14.66	14.60	14.59	14.61

Table 5.2: Optimal Total TCM RSS as a Function of Number of TCMs

TCMs are squentially added at the cheapest option throughout the entire trajectory and re-optimized via the gradient descent method until the TCM cost increases compared to the previous optimal value. Verification results for this method appear in Section 7.2.

5.5.2 Optimal TCM(s) with Variations in Stochastic Parameters

The objective of this section is to highlight the sensitivity of the optimal TCM set to variations in stochastic parameters through a series of examples. Each example shows the optimal TCM set corresponding to a change in the stochastic parameters in Figure 5.4 along the same three impulse nominal trajectory from LEO to NRI via PLF. Points of comparison include the stochastic cost as well as the number and location of TCMs along the trajectory as error sources change. Note that in Figures 5.4 through 5.11, while there is an initial coast in LEO from the green dot to the departure impulsive maneuver (not labeled to avoid congestion) and a final NRHO coast to reach the red square, these only enable flexible departure from the initial orbit and arrival to the target orbit and are not involved in the stochastic analysis. The initial dispersion is applied at ΔV_1 and the target dispersion constraint is applied at ΔV_3 with all TCMs occurring between these endpoints. The TCMs in the LEO to PLF portion of the trajectory aim to minimize position dispersion at ΔV_2 (PLF) and the TCMs in the PLF to NRI portion of the trajectory target ΔV_3 (NRI).

The first example in Figure 5.4 shows the baseline error parameters, the optimal TCM cost, and a comparison to another TCM selection method that has historically been referred to as the LAR or "looks about right" method [52]. The author selected a few TCM locations that looked about right to serve as a cost comparison to the optimal TCM set (see LAR J_σ in subsequent figures as a comparison to optimal J_σ). The LAR TCMs were selected with a few guidelines: 1) a TCM immediately follows nominal maneuvers to "clean up" nominal maneuver execution error; 2) the final TCM in each portion is placed such that it

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 1$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 18.0$ m/s
  - LAR $J_\sigma = 26.1$ m/s

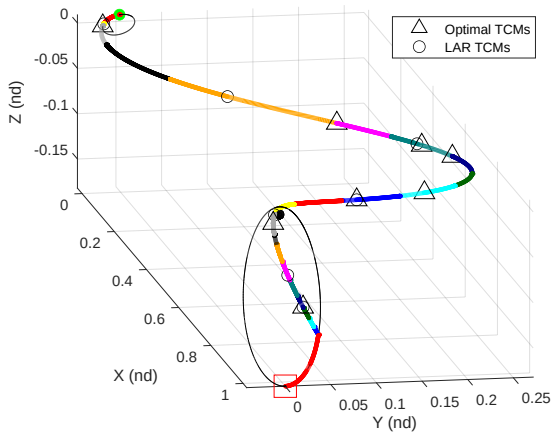


Fig. 5.4: Baseline Stochastic Parameters, Optimal and LAR TCMs Along LEO to NRHO Trajectory

meets the same target position dispersion constraint as the optimal for a fair comparison; 3) the remaining TCMs in each TCM targeted portion of the trajectory roughly visually subdivide the trajectory. The cost increase is minimal for the initial error values.

The LAR TCMs are shown as open circles in Figure 5.4, while the optimal TCMs for the baseline error sources occur at the open triangles. The second nominal maneuver at PLF is not a corrected nominal maneuver for this analysis; the magnitude comparison between the nominal maneuver and the TCM, in some cases, is such that the TCM magnitude is too large compared to the size of the nominal maneuver for the savings in Subsection 5.1.5 to apply.

The next example (Figure 5.5) explores the impact of increasing the effect of the process noise by a factor of 10. The final TCM preceding each target moved closer to the target, which makes sense; additional process noise causes the state dispersion to grow more quickly so the final TCM must be closer to the target to meet the same position dispersion constraint. Additional TCMs along the trajectory serve to reduce the cost of meeting the target position dispersion constraint as well as managing the magnitude of the trajectory dispersion along the trajectory. With increased process noise, the optimal TCM solution proactively manages dispersion growth throughout the trajectory with additional TCMs. The low TCM execution error does not penalize the addition of many TCMs in

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 1$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
  - ~~$\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$~~
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 44.4$ m/s
  - LAR $J_\sigma = 156.1$ m/s

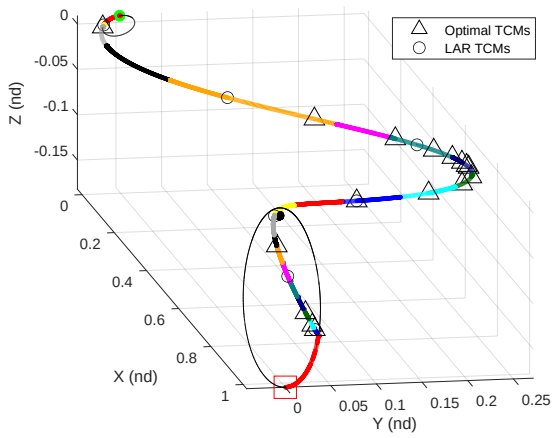


Fig. 5.5: 10x Process Noise, Optimal and LAR TCMs Along LEO to NRHO Trajectory

this case. TCMs are grouped near the lowest nominal velocity apogee-like portion of the trajectory. The optimal TCM set also minimizes the velocity dispersion at PLF, which, when not properly managed, creates a very expensive subsequent TCM (more on this in Chapter 6). The total 3σ correction cost comparison between optimal (44.4 m/s) and LAR (122.0 m/s) in this case is noteworthy. The only LAR TCM change for this example is an update to the final LAR TCM time prior to each target to meet the target position dispersion constraint.

Figure 5.6 shows the result of reducing the initial dispersion significantly. A portion of the total TCM cost is attributed to the corrected initial maneuver. With a small initial dispersion, the correction of the initial dispersion notably reduces in magnitude.

Figure 5.7 shows the impact of increasing the nominal maneuver execution error from 1 m/s to 10 m/s 1σ. The increased nominal maneuver execution error is corrected by the first TCM following each nominal ΔV. This results in a large 3σ TCM cost increase but does not significantly change the optimal TCM solution or result in significantly poor performance of the LAR TCMs when compared to the optimal set.

Figure 5.8 shows the impact of increasing the TCM execution error to 10 cm/s from 1 cm/s. The main difference in the optimal TCM set is the introduction of additional TCMs to manage the additional error being injected by each TCM into the dispersion covariance.

- Initial Dispersion
  - $\sigma_r = 10$ m ~~1 km~~
  - $\sigma_v = 1$ cm/s ~~1 m/s~~
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 1$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 13.1$ m/s
  - LAR $J_\sigma = 21.1$ m/s

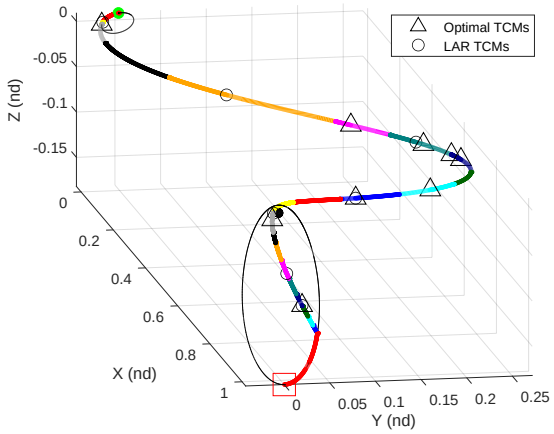


Fig. 5.6: Smaller Initial Dispersion, Optimal and LAR TCMs Along LEO to NRHO Trajectory

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
    ~~1 m/s~~
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 100.5$ m/s
  - LAR $J_\sigma = 111.6$ m/s

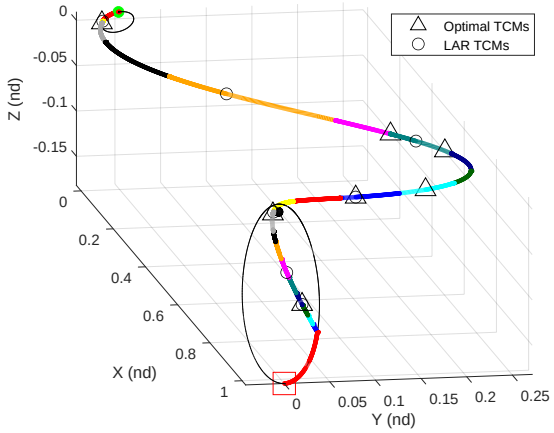


Fig. 5.7: Increased Nominal Maneuver Error, Optimal and LAR TCMs Along LEO to NRHO Trajectory

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 1$ m/s
  - $\sigma_{R_{TCM}} = 10$ cm/s
    ~~1 cm/s~~
- Process Noise
  - $\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 42.1$ m/s
  - LAR $J_\sigma = 138.0$ m/s

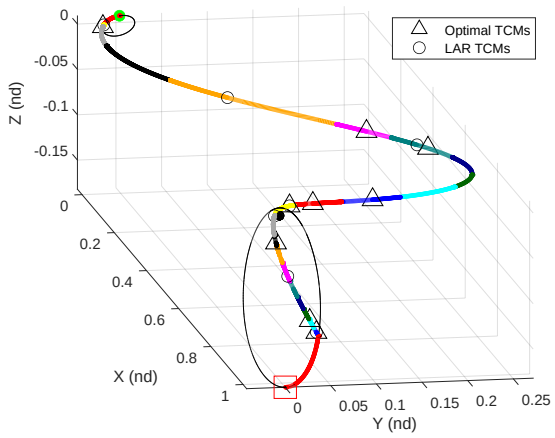


Fig. 5.8: 10 cm/s 1σ TCM Error, Optimal and LAR TCMs Along LEO to NRHO Trajectory

The TCM error is also included in the correction cost, which creates an upper bound on the number of TCMs before the cost begins to increase again. When comparing the increase of the optimal TCM cost (18.0 m/s initially to 42.1 m/s), the additional TCMs in the optimal set successfully mitigate a significant impact to the overall correction cost. On the contrary, the LAR method performs significantly worse regarding the increased TCM cost from 23.6 m/s to 114.9 m/s.

In Figure 5.9, the target position dispersion RSS constraint is reduced to 100 m from 1 km. The result is a final TCM that is closer to the target, preceded by an additional TCM to reduce the dispersion magnitude and correction magnitude for the final TCM of each targeting portion. The cost of the optimal TCM solution does not significantly increase as a result of the more stringent constraint. The performance of the LAR method is not terrible but poor by comparison as it does not benefit from additional TCMs preceding each final targeting TCM.

Figure 5.10 shows the result of increasing the magnitude of multiple error sources simultaneously, highlighting the importance of an optimized TCM set. Increasing the nominal maneuver execution error and process noise increases the TCM cost to 123.3 m/s for the optimized set, compared to 182.2 m/s for the LAR TCM set. Increasing the TCM execution error in addition to the process noise and nominal maneuver execution error (Figure 5.11

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 1$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 100$ m ~~1 km~~
- 3σ Correction Cost
  - Optimal $J_\sigma = 18.8$ m/s
  - LAR $J_\sigma = 45.1$ m/s

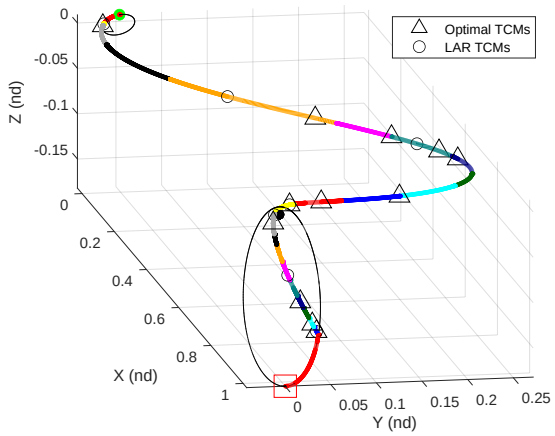


Fig. 5.9: 100 m Position Dispersion Constraint, Optimal and LAR TCMs Along LEO to NRHO Trajectory

which matches Figure 6.7) results in only a moderate increase to the TCM cost to 144.4 m/s but an additional significant increase to 332.0 m/s for the LAR TCM set. With the increase in TCM execution error, the total number of TCMs is reduced when compared to the set in Figure 5.10.

Table 5.3 summarizes the TCM scenario results shown in Figures 5.4 through 5.11. There are conclusions related to modifications in each error source. Correcting initial dispersion is expensive whether the optimal TCM set or LAR TCM set is implemented. Similarly, correcting nominal maneuver execution error is also expensive, independent of the TCM set. Optimizing TCMs provides the most benefit for error sources that contribute to the state dispersion along the trajectory, whether continuously in the form of process noise, or discretely through increased TCM execution error. Optimizing TCMs is also beneficial in a scenario that requires a more stringent target position dispersion.

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
    ~~1 m/s~~
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
    ~~$\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$~~
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 123.3$ m/s
  - LAR $J_\sigma = 213.5$ m/s

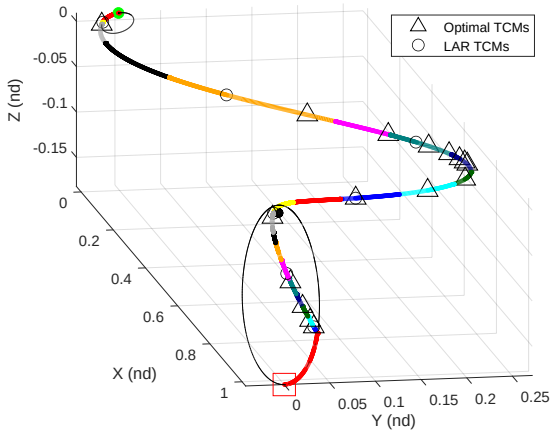


Fig. 5.10: Increased Nominal Maneuver Execution Error and Process Noise, Optimal and LAR TCMs Along LEO to NRHO Trajectory

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
    ~~1 m/s~~
  - $\sigma_{R_{TCM}} = 10$ cm/s
    ~~1 cm/s~~
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
    ~~$\sigma_Q = 0.1\text{mm/s}/\sqrt{\text{s}}$~~
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km
- 3σ Correction Cost
  - Optimal $J_\sigma = 144.4$ m/s
  - LAR $J_\sigma = 407.6$ m/s

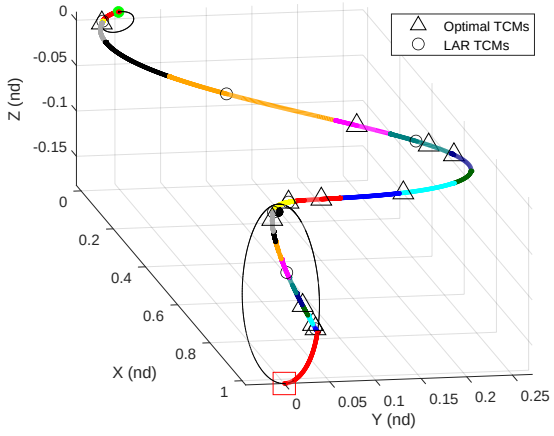


Fig. 5.11: Increased Nominal Maneuver Execution Error, TCM Execution Error, and Process Noise, Optimal and LAR TCMs Along LEO to NRHO Trajectory

σ_{r_0}	σ_{v_0}	$\sigma_{R_{\Delta V}}$	$\sigma_{R_{TCM}}$	Q	$\sigma_{r,max}$	LAR J_σ	Opt. J_σ
1 km	1 m/s	1 m/s	1 cm/s	0.1 mm/s/$\sqrt{\text{s}}$	1 km	23.6 m/s	18.0 m/s
1 km	1 m/s	1 m/s	1 cm/s	1 mm/s/$\sqrt{\text{s}}$	1 km	122.0 m/s	44.4 m/s
10 m	1 cm/s	1 m/s	1 cm/s	0.1 mm/s/$\sqrt{\text{s}}$	1 km	18.4 m/s	13.1 m/s
1 km	1 m/s	10 m/s	1 cm/s	0.1 mm/s/$\sqrt{\text{s}}$	1 km	110.8 m/s	100.5 m/s
1 km	1 m/s	1 m/s	10 cm/s	0.1 mm/s/$\sqrt{\text{s}}$	1 km	114.9 m/s	42.1 m/s
1 km	1 m/s	1 m/s	1 cm/s	0.1 mm/s/$\sqrt{\text{s}}$	100 m	37.8 m/s	18.8 m/s
1 km	1 m/s	10 m/s	1 cm/s	1 mm/s/$\sqrt{\text{s}}$	1 km	182.2 m/s	123.3 m/s
1 km	1 m/s	10 m/s	10 cm/s	1 mm/s/$\sqrt{\text{s}}$	1 km	332.0 m/s	144.4 m/s

Table 5.3: TCM Sensitivity Analysis Along Deterministic Optimal LEO to PLF to NRI Trajectory Summary. Observations to be made from the results: 1) As various error sources increase from the first row, which error sources are the most expensive to correct. 2) How the LAR TCM set compares to the optimal TCM set with changes in error sources. A major conclusion is that erorr sources that contribute to the state dispersion continuously (process noise) or discretely multiple times along the trajectory (TCM execution error) have the potential to be expensive to correct if the TCM set is not optimized. On the other hand, increased initial dispersion and nominal maneuver execution error is expensive to correct in the LAR and optimal TCM set.

CHAPTER 6

CISLUNAR ROBUST TRAJECTORY DESIGN

The robust trajectory design method incorporates linear covariance analysis along each nominal trajectory to minimize the sum of the deterministic ΔV plus an estimate for the upper bound of the optimal TCM set δV.[1] The resulting cost function represents the statistical upper bound for the total ΔV requirement, a mission planning factor that directly correlates to on-board spacecraft propellant. In addition to describing the problem setup and reviewing the robust trajectory design results for various scenarios, another goal of this section is to convey when pursuing a robust trajectory over a deterministic optimal trajectory may provide benefit versus the scenarios when there is likely little to be gained.

The solution method is nonlinear programming with analytical gradients (Matlab's fmincon and the interior point method). The nominal trajectory is divided into segments to aid convergence, particularly for nonlinear portions of the trajectory. First, an initial guess nominal trajectory is numerically integrated. Next, the number and location of TCMs are optimized along the initial guess per the method described in Section 5.5. The TCM locations form the initial segmentation of the initial guess trajectory. Subsequently, corresponding TCMs are fixed to occur at the corresponding node after each iteration, as opposed to performing the approach in Section 5.5. In some cases, successful NLP convergence requires additional segmentation of particularly long segments between TCMs. Each TCM is tied to a specific segment intersection and additional segments are incorporated for long duration trajectory spans.[2] In this manner, the TCM optimization algorithm is only

[1] A significant portion of this chapter's content was submitted to be published with the Journal of Astronautical Sciences.

[2] See the TCM identification triangles in Figure 5.8; the duration between TCMs 1 and 2 requires additional segmentation. Unsuccessful convergence steps are generally quite obvious when the issue is too few segments. When intermediate steps become disjointed and illogical, one attempt to obtain convergence should be subdividing long segments into additional segments. This work does not investigate an upper limit on the number of segments required, although the authors did not notice a drawback with erring on the side of additional segments, such as 25 or 30 total segments across the LEO-Lunar flyby-NRI trajectory.

run one time, on the initial guess trajectory.[3] This places the optimal number of TCMs in the appropriate local minima for the NLP to optimize as the trajectory changes. This also ensures TCMs times are not limited to variable step size STM histories and are instead tied to NLP problem parameters, creating a direct connection to analytical gradients with a smoother solution space.

Section 5.5 explored the optimization of TCMs along the same deterministic optimal nominal trajectory. In finding a robust trajectory, each NLP step corresponds to a new nominal trajectory. At each new nominal trajectory, a new state, STM, STT, QBM ($\bar{Q}$), and QBT ($\partial\bar{Q}$) is propagated in order to assemble the analytical gradients to provide back to the NLP for determining the next step. Section 5.4 described the steps for building the analytical gradients for each TCM and their total.

The following optimization problem applies to m nominal maneuver trajectories (indexed by the letter j) along an n segment trajectory (indexed by the letter i).

[3] If the NLP steps result in significant changes to the nominal trajectory, it may be necessary to re-run robust optimization starting with the previously converged solution as the new initial guess. This allows for an update to the number of TCMs and their associated nodes.

$$\text{optimization parameters } \boldsymbol{S} \tag{6.1}$$

$$\text{minimize } J = \sum_{j=1}^{m} \Delta V + 3\left(\sum_{k=1}^{n_{\delta V}} (\sigma_{\delta V}) + \sum_{q=1}^{n_{\Delta V_c}} \sigma_{\|\Delta \boldsymbol{V}_c\|}\right) \tag{6.2}$$

$$\text{subject to } \boldsymbol{x}_{f,i} = f\left(\boldsymbol{x}_{0,i}, \Delta t_i\right) \tag{6.3}$$

$$P_{k+1} = \Phi\left(t_{k+1}, t_k\right) P_k \Phi\left(t_{k+1}, t_k\right)^\top + \bar{Q}\left(t_{k+1}, t_k\right) \tag{6.4}$$

$$P_k^{+t_j} = P_k^{-t_j} + G R_{\Delta V} G^\top \tag{6.5}$$

$$P_k^{+t_{c_k}} = (I+N) P_k^{-t_{c_k}} (I+N)^\top + G R_{TCM} G^\top \tag{6.6}$$

$$\boldsymbol{x}_{0,i} = \boldsymbol{x}_{f,i-1}, i \notin \boldsymbol{d} \tag{6.7}$$

$$\boldsymbol{r}_{0,i} = \boldsymbol{r}_{f,i-1}, i \in \boldsymbol{d}, i > 1 \tag{6.8}$$

$$\sqrt{\text{tr}\left(P_{rr,\boldsymbol{d}}\right)} \leq \sigma_{r,max}, j > 1 \tag{6.9}$$

$$\Delta t_i \geq 0 \tag{6.10}$$

$$\text{given } \boldsymbol{x}_{0,1}, \boldsymbol{x}_{f,n}, P_1, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{c}, R_{\Delta V}, R_{TCM}, Q, \sigma_{r,max} \tag{6.11}$$

Nominal impulsive maneuvers are allowed at the trajectory nodes identified in the vector $\boldsymbol{d}$; the vector $\boldsymbol{e}$ identifies which of nominal maneuvers in $\boldsymbol{d}$ include a correction (for $n_{\Delta V_c}$ total corrected nominal maneuvers indexed by the letter q). The TCM optimization method identifies $n_{\delta V}$ total position-correcting TCMs along the initial guess nominal trajectory; the corresponding TCM execution nodes are stored in the vector $\boldsymbol{c}$ (indexed by the letter k). The cost function (Equation 6.2) is the the sum of the nominal maneuvers plus the total 3σ TCM δV. Of the constraint equations, Equation 6.3 is the system dynamics for each trajectory segment as a function of the initial state and the segment duration. Equation 6.4 represents the covariance propagation using numerically integrated STM terms along the nominal trajectory and the accumulated process noise covariance between covariance updates. Equations 6.5 and 6.6 represent the dispersion covariance update equations for a nominal maneuver with execution error and a TCM with execution error, respectively. Equations 6.7 and 6.8 constrain the trajectory segments to be connected by all six state

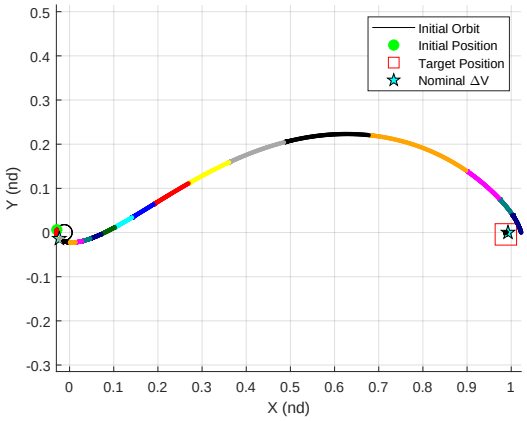
elements (a coast) or allow an impulsive ΔV between segments, as identified by the nodes in the vector $\boldsymbol{d}$. Equation 6.9 constrains the RSS of the position dispersion at nominal maneuvers after the first to be less than or equal to $\sigma_{r,max}$. The duration of each trajectory segment must go forward in time (Equation 6.10). The total trajectory duration is unconstrained. Upon convergence, the optimal TCM times for the robust trajectory are extracted from the appropriate nodes of $\boldsymbol{S}$.

6.1 Robust LEO to Low Lunar Orbit

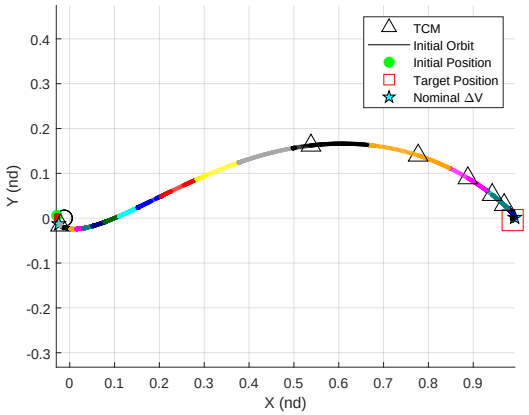
The first scenario analyzed is a nominal two impulse trajectory, a translunar injection (TLI) maneuver departing LEO followed by a lunar orbit insertion (LOI) maneuver to low lunar orbit (LLO). The initial Earth orbit is a 400 km prograde circular orbit, coplanar with the plane defined by the Moon's motion around the Earth. The target lunar orbit is 110 km retrograde circular orbit, inclined 27 degrees and LOI occurring on a descending pass. The problem formulation is such that the true anomaly is flexible in order to optimize the departure and arrival locations within the initial and target orbits. A shooting method is used to develop the disconnected initial guess trajectory shown in Figure 6.1a, which converges to the deterministic optimal trajectory shown in Figure 6.1b. Figure 6.2 shows the initial guess trajectory, convergence steps, and converged deterministic optimal trajectory in a single figure (the NLP cost function only includes nominal ΔV).

Figure 6.3 shows the error sources incorporated and the resulting robust LEO to LLO trajectory. Table 6.1 summarizes the deterministic cost with optimal corrections and the robust cost. In this case, the deterministic optimal and robust trajectories are nearly identical. The problem dynamics and constraints do not present sufficient flexibility in the solution space where spending additional nominal ΔV can result in a TCM cost savings. Additionally, a major takeaway that the authors have observed in similar scenarios is related to the cost ratio of nominal ΔV to TCM δV. For such a costly set of nominal maneuvers, there is little trade space to spend additional nominal ΔV for savings on the TCM cost.

One possibility for the lack of a difference in the deterministic and robust trajectories is that the problem is overly constrained. In an attempt to find a robust trajectory that



(a) LEO to LLO Initial Guess



(b) Deterministic Optimal LEO to LLO Trajectory with Optimal TCM Set

Fig. 6.1: Initial Guess & Converged Trajectory, Deterministic Optimal LEO to LLO

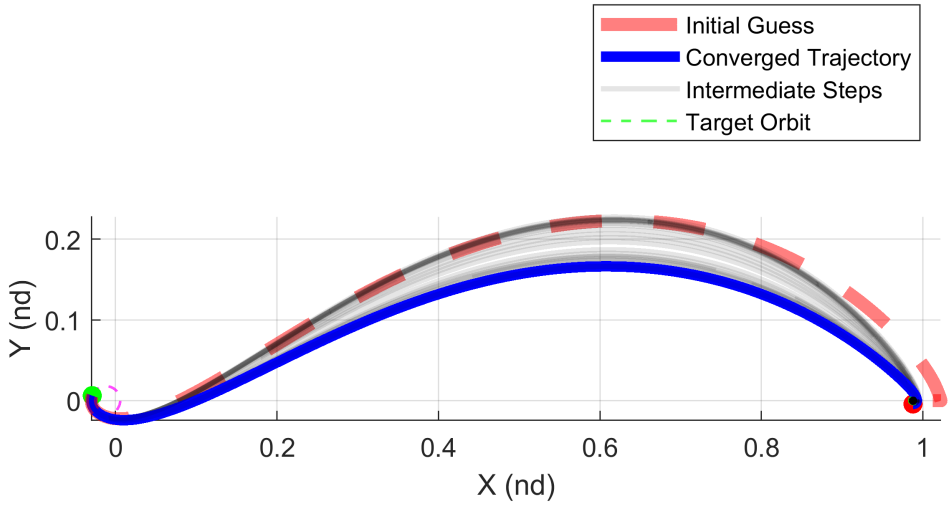


Fig. 6.2: Convergence Steps to Deterministic Optimal LEO to LLO Trajectory

- Initial Dispersion
  - $\sigma_r = 10$ m
  - $\sigma_v = 1$ cm/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km

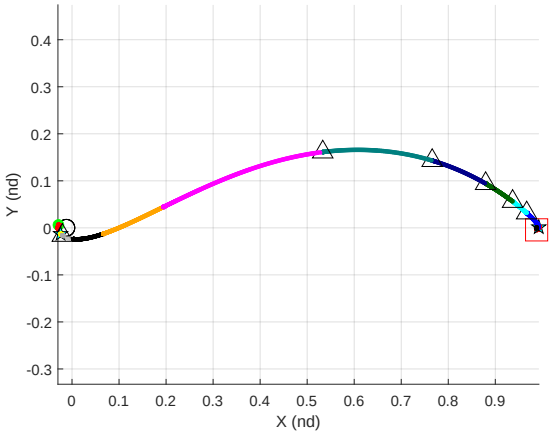
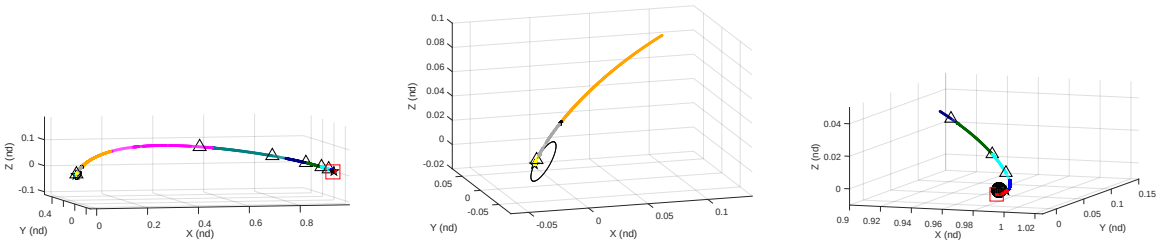


Fig. 6.3: Robust LEO to LLO Trajectory

Deterministic and Robust Cost Comparison (m/s)				
Deterministic	ΔV	4326.9	$\|\Delta V_1\|$	3015.2
	Optimal TCM 3σ δV	54.3	$\|\Delta V_2\|$	1208.0
	ΔV Total Upper Bound	4381.2		
Robust	ΔV	4326.9	$\|\Delta V_1\|$	3015.5
	Optimal TCM 3σ δV	54.3	$\|\Delta V_2\|$	1207.7
	ΔV Total Upper Bound	4381.2		

Table 6.1: Robust LEO to LLO Trajectory Cost Comparison

provides savings over the deterministic optimal, the constraints are modified in the next example. In the first LEO to LLO trajectory, the initial orbit is fully constrained with a free true anomaly. For the next example, the initial orbit is only constrained to be a 400 km circular orbit via Constraint Equations 3.50, 3.51, and 3.52, resulting in a free orbital plane and true anomaly. Figure 6.4 shows three views of the deterministic optimal trajectory. Table 6.2 shows the deterministic and robust cost comparison. While relaxing the initial orbit constraints allowed for maneuver optimization to obtain an out of plane component at TLI which in turn minimizes the plane change required at LOI, it did not create design space flexibility that enables robust savings. The robust trajectory is not shown in an additional figure as it is effectively the same trajectory as the deterministic optimal case.



(a) TLI to LOI Trajectory (b) TLI Maneuver (c) LOI Maneuver

Fig. 6.4: Deterministic Optimal TLI to LOI Trajectory, Free Initial Orbital Plane

Deterministic and Robust Cost Comparison (m/s)				
Deterministic	ΔV	4185.1	$\|\Delta V_1\|$	3060.9
	Optimal TCM 3σ δV	54.8	$\|\Delta V_2\|$	1124.2
	ΔV Total Upper Bound	4239.9		
Robust	ΔV	4185.1	$\|\Delta V_1\|$	3061.8
	Optimal TCM 3σ δV	54.7	$\|\Delta V_2\|$	1123.4
	ΔV Total Upper Bound	4239.9		

Table 6.2: Robust LEO to LLO Trajectory Cost Comparison

6.2 Robust Three Impulse LEO to Elliptical Earth Departure to Lunar Orbit

The next option allows three impulsive maneuvers to potentially enable additional flexibility in the design space from LEO to LLO. The initial guess trajectory is implemented with an additional coast of a single orbit in LEO and a relaxed constraint set to allow an additional ΔV prior to TLI. The target lunar orbit is 1000 km altitude, circular, coplanar with the Earth-Moon plane. Figure 6.5 shows the deterministic optimal trajectory: a large ΔV from LEO enters a very high apogee (near the distance of lunar orbit) elliptical Earth orbit followed by a small ΔV to intersect the lunar position and finally the LOI maneuver. The second maneuver is referred to in this dissertation as the elliptical Earth departure (EED). The total duration of this trajectory is 13.48 days.

Figure 6.6 shows the resulting robust trajectory when the cost of correcting the error sources in Table 6.3 is included in the optimization cost function. Effectively, the size of the Earth departure ellipse is optimized along with minor changes to the portion of the trajectory from the Earth to the Moon. The distance from Earth at apogee of the ellipse is 54,668 km with a new total transfer duration of 6.01 days. Table 6.4 shows the cost comparison of the deterministic optimal result with optimal corrections and the robust result. The robust trajectory results in a total cost upper bound savings of 15.7 m/s.

This scenario presents another interesting consideration regarding the correction of nominal maneuvers. The first nominal maneuver (ΔV_1) serves as the departure from LEO into an elliptical Earth orbit. ΔV_2 serves as the "Earth flyby" maneuver, performed at perigee of the elliptical orbit. The TCM target for the first correction portion is the position at ΔV_2. As a result, incorporating a combined correction with ΔV_1 results in attempting to correct out-of-plane dispersion through a 360° transfer, which is not possible, or results in being forced to use a different transfer angle. A singularity similar to correcting out-of-plane dispersion in a 180° transfer exists (the Lambert ridge for interplanetary missions). The result is a very expensive correction to ΔV_1 alongside a modification to the geometry between ΔV_1 and ΔV_2 such that the correction is not impossible. As an alternative, the first nominal maneuver can also omit the combined correction (the nominal maneuver is

1σ Error Sources					
Initial Dispersion		**ΔV Error**	**TCM Error**	**Process Noise**	**$\sigma_{r,max}$ at ΔV_2**
1 km	1 m/s	10 m/s	10 cm/s	1mm/s/$\sqrt{\text{s}}$	1 km

Table 6.3: LEO to EED to LLO Trajectory Error Sources

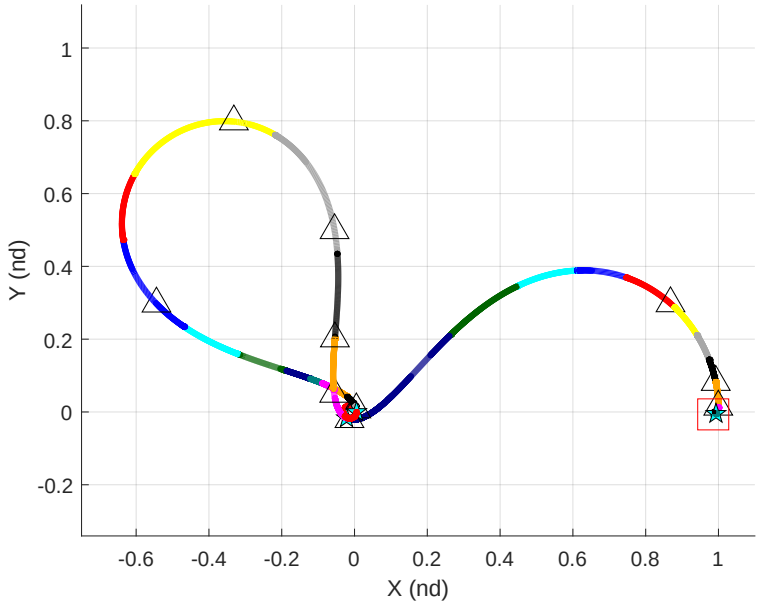


Fig. 6.5: Deterministic Optimal Three Impulse LEO to EED to LLO Trajectory

executed as planned in spite of a known state dispersion), and subsequent TCMs are then performed to manage the dispersion separately from ΔV_1. The results for this section represent an uncorrected ΔV_1 and a corrected ΔV_2 and ΔV_3. Selecting which nominal maneuvers are corrected is currently a manual step that must be identified and implemented by the trajectory designer. The very large ΔV_1 correction portion of the nominal maneuver is likely inaccurate enough to be identified during Monte Carlo verification if not identified in advance.

6.3 Robust LEO to Powered Lunar Flyby to NRHO Trajectory

This scenario starts with the same nominal trajectory used for TCM optimization in Section 5.5. The initial orbit is a 450 km circular orbit, inclined 28 degrees with respect to the Earth-Moon plane and rotated so that the right-ascension in the CR3BP rotating

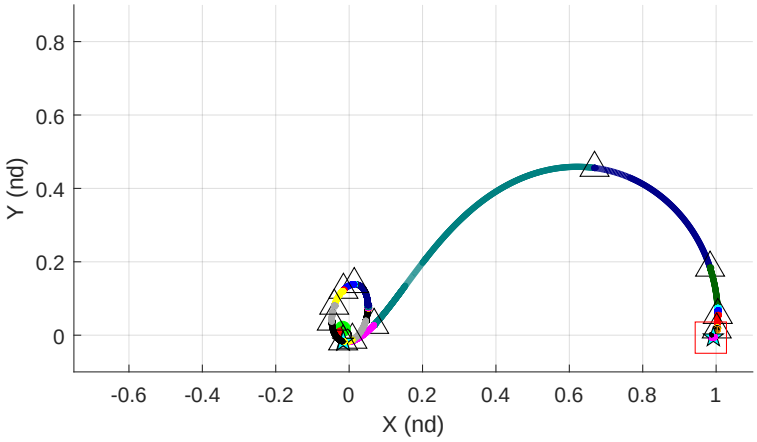


Fig. 6.6: Robust Three Impulse LEO to EED to LLO Trajectory

Deterministic and Robust Cost Comparison (m/s)				
Deterministic	ΔV	3785.1	$\|\Delta V_1\|$	3049.7
	Optimal TCM 3σ δV	133.1	$\|\Delta V_2\|$	12.3
	ΔV Total Upper Bound	3918.2	$\|\Delta V_3\|$	723.1
Robust	ΔV	3799.4	$\|\Delta V_1\|$	2544.6
	Optimal TCM 3σ δV	104.4	$\|\Delta V_2\|$	525.3
	ΔV Total Upper Bound	3903.8	$\|\Delta V_3\|$	729.4

Table 6.4: LEO to EED to LLO Trajectory Cost Comparison

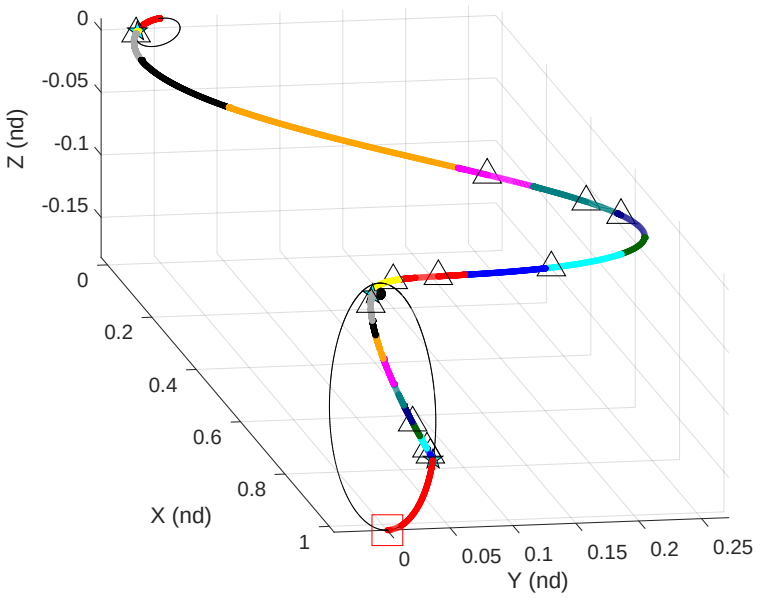


Fig. 6.7: Deterministic Optimal NRI Trajectory with Optimal TCMs

frame is 50 degrees counterclockwise from the x-axis. The target orbit is the 9:2 synodic resonant member of the L2 Southern Halo family. The nominal maneuver execution error and process noise were increased for this example from the baseline comparison set in Figure 5.4 to the error parameters shown in Figure 6.8. Table 6.5 shows a cost comparison of the deterministic optimal trajectory with an optimal set of TCMs to the robust trajectory. Tables 6.6 and 6.7 show the cost of each correction as well as the absolute time each TCM is executed along the deterministic optimal and robust trajectories, respectively.

In this scenario, the robust trajectory provides a savings of 19.8 m/s to the total cost upper bound. For missions where ΔV_1 is performed by a launch vehicle upper stage, this represents a reduction in the space vehicle's maneuver requirement from 550.1 to 530.1 m/s.[4] Similar to the message portrayed by Subsection 5.5.2, the robust trajectory result is dependent on the error sources incorporated. This scenario is ideal for pursuing a robust trajectory over the deterministic optimal. The risk of spending additional nominal ΔV is low (an additional 1.8 m/s) and the potential payoff in terms of correcting errors is meaningful.

[4]Different weighting factors of cost function values between upper stage cost and spacecraft propellant has the potential to further optimize a robust trajectory that prioritizes the spacecraft, for example.

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
  - $\sigma_{R_{TCM}} = 10$ cm/s
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km

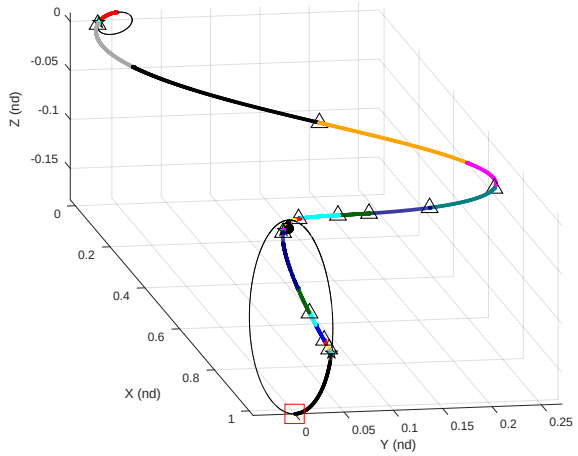


Fig. 6.8: Robust NRI Trajectory

Deterministic and Robust Cost Comparison (m/s)				
Deterministic	ΔV	3463.9	$\|\Delta V_1\|$	3058.2
	Optimal TCM 3σ δV	144.4	$\|\Delta V_2\|$	212.5
	ΔV Total Upper Bound	3608.2	$\|\Delta V_3\|$	193.2
Robust	ΔV	3465.7	$\|\Delta V_1\|$	3058.2
	Optimal TCM 3σ δV	122.6	$\|\Delta V_2\|$	213.1
	ΔV Total Upper Bound	3588.4	$\|\Delta V_3\|$	194.4

Table 6.5: Robust NRI Trajectory Cost Comparison

TCM Portion 1 Targeting ΔV_2 Pos.			**TCM Portion 2 Targeting ΔV_3 Pos.**		
	3σ δV (m/s)	Time (hrs)		3σ δV (m/s)	Time (hrs)
ΔV_1 Corr.	3.85	0.30840	ΔV_2	N/A	90.043
TCM 1	43.6	0.33967	TCM 8	72.6	90.552
TCM 2	2.84	16.483	TCM 9	3.52	108.12
TCM 3	1.48	26.842	TCM 10	2.47	115.10
TCM 4	1.28	33.218	TCM 11	1.46	116.92
TCM 5	3.64	71.453	ΔV_3 σ_v	1.82	118.36
TCM 6	3.10	84.999			
TCM 7	2.72	88.766			

Table 6.6: Deterministic NRI Trajectory TCM Set

TCM Portion 1 Targeting ΔV_2 Pos.			**TCM Portion 2 Targeting ΔV_3 Pos.**		
	3σ δV (m/s)	Time (hrs)		3σ δV (m/s)	Time (hrs)
ΔV_1 Corr.	3.85	0.30752	ΔV_2	N/A	89.871
TCM 1	43.5	0.34088	TCM 8	51.76	90.000
TCM 2	2.33	11.784	TCM 9	2.72	103.21
TCM 3	3.13	48.849	TCM 10	2.34	111.50
TCM 4	2.62	70.051	TCM 11	2.10	114.68
TCM 5	2.12	80.248	ΔV_3 σ_v	1.25	116.16
TCM 6	1.71	84.439			
TCM 7	3.23	88.666			

Table 6.7: Robust NRI Trajectory TCM Set from NLP Results Details

The majority of the savings for this trajectory comes from the TCM following ΔV_2. Correcting accumulated velocity dispersion once attaining the target position at lunar flyby is expensive; with a high velocity magnitude and a relatively low ΔV_2 magnitude there are two compounding effects. First, velocity dispersion orthogonal to the velocity vector at flyby effectively requires a plane change to correct which is more expensive with a large nominal velocity. A major optimization that occurs in the robust trajectory is the minimization of the velocity dispersion at lunar flyby (an RSS of 3.8 m/s) when compared to the deterministic optimal trajectory (RSS of 26.1 m/s). Second, since ΔV_2 is a relatively inexpensive maneuver, the corrected nominal savings from Equation 5.17 is not applicable because the correction at ΔV_2 is comparable in magnitude in some cases than the nominal maneuver, which is why the first and third nominal maneuvers are the only corrected maneuvers incorporated in this problem. When modeled with the combined maneuver savings, the nonlinear program leverages the savings in an unrealistic way, resulting in an unrealistic combined savings which fails Monte Carlo verification.

In general, robust solutions also result in an earlier arrival to the target orbit, thereby reducing the total dispersion growth that occurs from dynamics and process noise.[5] While

[5]However, there are competing stochastic interests; a TCM to mitigate an initial dispersion early in a trajectory generally benefits from having a longer duration target. Again, the robust result is very dependent upon the error sources utilized.

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
  - $\sigma_{R_{TCM}} = 10$ cm/s
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km

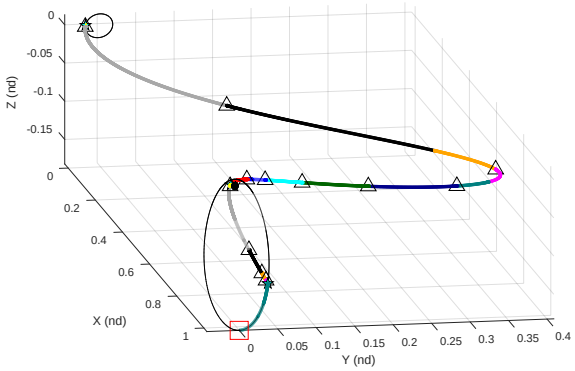


Fig. 6.9: Robust NRI Trajectory with Free Initial Plane

a difficult comparison to make visually between Figure 6.8 (robust) and Figure 6.7 (deterministic optimal), the robust NRHO insertion position occurs 2052 km earlier in the NRHO path compared to the deterministic optimal insertion. In another run of the same scenario scenario with less process noise, the robust solution resulted in an optimal NRI 1011 km earlier than the deterministic optimal NRI.

Relaxing the initial orbital constraint set to free the orbital plane by implementing Constraint Equations 3.50, 3.51, and 3.52 rather than constraining the initial position and velocity enables additional optimization in the deterministic optimal and robust NRI scenarios. Figure 6.9 shows the resulting robust NRI trajectory and Table 6.8 shows the cost. The robust trajectory provides a potential 77.8 m/s in savings compared to the deterministic optimal trajectory. Assuming ΔV_1 is performed by a launch vehicle upper stage, the space vehicle maneuver requirement is 509.7 m/s. Appendix B.2 shows the CR3BP state at mission events and coast durations required to recreate this trajectory.

Table 6.9 shows the result of a study comparing the sensitivity of multiple NRI trajectories with a free initial orbital plane to variations in stochastic parameters. The trajectories compared are: 1) the deterministic optimal trajectory in Table 6.8; 2) the robust trajectory in Table 6.8; 3) a new robust trajectory created specifically for the new stochastic parameters. The results shown for Trajectories 1 and 2 are the TCM cost along the nominal trajectory using the algorithm described in Section 5.5.1 and the total cost upper bound. The nominal ΔV is not shown again as it does not change for each test for Trajectories 1 and

Deterministic and Robust Cost Comparison (m/s)				
Deterministic	ΔV	3439.3	$\|\Delta V_1\|$	3052.5
	Optimal TCM 3σ δV	200.86	$\|\Delta V_2\|$	195.6
	ΔV Total Upper Bound	3640.2	$\|\Delta V_3\|$	191.2
Robust	ΔV	3440.9	$\|\Delta V_1\|$	3052.7
	Optimal TCM 3σ δV	121.49	$\|\Delta V_2\|$	188.9
	ΔV Total Upper Bound	3562.4	$\|\Delta V_3\|$	199.3

Table 6.8: Robust NRI Trajectory Cost Comparison

2. For Trajectory 3, each row represents a new robust trajectory with a new nominal cost and TCM cost, each the converged result from the NLP robust trajectory design method. There are numerous conclusions to be made from this table, related to the sensitivity of each trajectory to various error sources, which error source is driving the large velocity dispersion after PLF, and once a robust trajectory is found for a specific set of error sources (Trajectory 2), whether it continues to perform well to variations in the error sources or is it beneficial to find a new robust trajectory for each change in error sources.

First, observe the change in cost associated with changes to the initial dispersion. Reducing the initial 1σ position dispersion from 1 km to 100 m results in a large reduction in TCM cost along the deterministic optimal trajectory (200.86 to 139.99 m/s). This highlights that the deterministic optimal trajectory is sensitive to initial position dispersion. On the other hand, the TCM cost along the robust trajectory is only reduced from 121.49 to 120.29 m/s with the same change in initial position dispersion, which leads to the conclusion that the robust trajectory is robust to initial position dispersion. The TCM after powered lunar flyby continues to be the major cost driver due to large velocity dispersion. The robust trajectory in these cases continues to minimize the velocity dispersion at PLF and the resulting 3σ RSS of the TCM following PLF.

Second, observe the change in cost associated with changes to the nominal maneuver execution error. A reduction in 1σ nominal maneuver execution error from 10 to 1 m/s

results in a decrease along the deterministic optimal trajectory from 200.86 to 142.39 m/s. The same change to nominal maneuver execution error for the robust trajectories reduces the cost from 121.49 to 41.30 m/s, a larger difference. While the robust trajectory is less sensitive to initial position dispersion it appears to be more sensitive to nominal maneuver execution error. Both trajectories appear to exhibit similar sensitivity to changes in TCM execution error and process noise; decreases to each of these error sources results in a reduction to the cost of corrections. Even in the case with the lowest error source magnitudes, the 3σ TCM cost is reduced from 19.27 to 11.73 m/s while the nominal ΔV increases by either 0.5 m/s or 1.6 m/s, further higlighting that the robust trajectory is a worthwhile pursuit with minimal risk of an overall increase in cost.

Third, the performance of optimal TCMs along Trajectory 2 appear to perform quite comparably to a new robust convergence for each reduction in error source values. While this phenomenon has not been extensively tested to verify its universality, in this instance, identifying a robust trajectory to larger error sources appears to also be robust to reductions in smaller magnitude error sources such that there are diminishing savings realized by finding a new robust trajectory.

Problem constraints also play a significant role in finding a robust trajectory. Another three impulse lunar flyby trajectory tested was insertion into the Artemis I Distant Retrograde Orbit (DRO) via powered lunar flyby. The constraint being compared is the lunar flyby distance constraint. In the NRI lunar flyby, the optimal solution has a perilune radius of 2094 km and perilune is constrained to be greater than or equal to 1837.4 km (the moon's radius plus 100 km). As a result, in the NRI case the flyby distance constraint is not active. However, in the DRO case, the flyby constraint is active and prevents the trajectory from traversing within the surface of the moon. The active flyby distance constraint limits the design space such that the optimal deterministic and robust trajectories converge to the same trajectory shown in Figure 6.10. On the other hand, any mission related constraints on TCMs (such as crew sleep avoidance) that prevent performing the optimal number at

(km)	(m/s)	(m/s)	(cm/s)	($\frac{mm/s}{\sqrt{s}}$)	Cost Along 3439.3 m/s Det Opt (m/s)		Cost Along 3440.9 m/s Robust (m/s)		New Robust Trajectory (m/s)		
σ_r	σ_v	$\sigma_{R_{\Delta V}}$	$\sigma_{R_{TCM}}$	σ_Q	TCM δV	Upper Bound	TCM δV	Upper Bound	Nom. ΔV	TCM δV	Upper Bound
1	1	10	10	1	200.86	3640.2	121.49	3562.4	3440.9	121.49	3562.4
1	0.1	10	10	1	199.45	3638.8	120.62	3561.5	3440.9	120.08	3561.0
1	1	10	10	0.3	190.96	3630.3	116.60	3557.5	3440.9	115.40	3556.3
1	1	10	1	1	189.04	3628.4	112.28	3553.2	3440.6	112.52	3553.1
1	1	10	10	0.1	188.24	3627.6	115.05	3555.9	3440.9	115.00	3555.9
1	1	1	10	1	142.39	3581.7	43.79	3484.7	3442.0	41.30	3483.3
0.1	1	10	10	1	139.99	3579.3	121.95	3562.8	3440.2	120.29	3560.5
0.1	0.1	10	10	1	137.36	3576.7	119.32	3560.2	3440.3	117.90	3558.2
0.01	0.01	10	10	1	134.17	3573.5	118.97	3559.8	3440.2	117.26	3557.5
0.1	0.1	1	10	1	57.86	3497.2	38.73	3479.6	3441.2	38.46	3479.6
0.1	0.1	1	1	0.3	23.45	3462.8	14.70	3455.6	3439.9	14.96	3454.8
0.1	0.1	1	1	0.1	19.27	3458.6	11.45	3452.3	3439.8	11.73	3451.5

Table 6.9: Robust Trajectory Cost Comparison, NRI Trajectory with Free Initial Orbital Plane

- Initial Dispersion
  - $\sigma_r = 10$ km
  - $\sigma_v = 10$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ m/s
  - $\sigma_{R_{TCM}} = 1$ cm/s
- Process Noise
  - $\sigma_Q = 0$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 1$ km

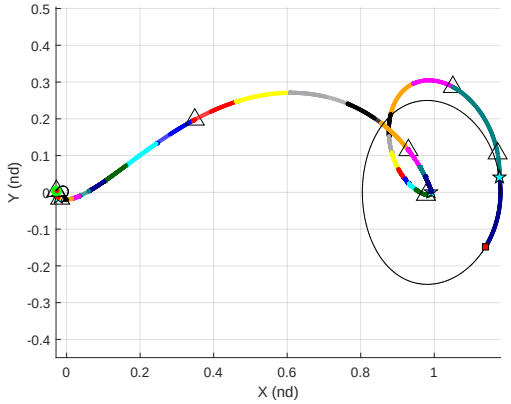


Fig. 6.10: Deterministic Optimal and Robust LEO-PLF-DRO Trajectory

the optimal execution time have the potential to increase the benefit of a robust trajectory as the TCM cost increases.

6.4 Robust NRHO Rendezvous Trajectory

The next scenario analyzes an active chaser spacecraft trailing a passive target in the same NRHO[6] by a specified amount of time (or distance, but the specific distance for a corresponding time delay depends on the true anomaly). Two nominal impulsive maneuvers are allowed by the chaser with the goal of rendezvous with a future target position. The initial and final segments are again flexible in duration resulting in the optimization of the location to perform the two nominal impulsive maneuvers. This method explores long range rendezvous and optimization of impulsive ΔV rather than terminal guidance. Effectively, two phasing maneuvers are being performed by the chaser, occurring within the same NRHO orbital period.

The deterministic optimal result for any initial and final position within the same orbital period is interesting in this case; with varying chaser trail durations, the location for the two nominal ΔVs remains nearly constant. Figure 6.11 shows five different initial conditions determined by the duration of time delay between the chaser and the target. All initial chaser positions precede the first ΔV. The nominal ΔV total varies quite significantly from 0.265 m/s in the 5 minute delay case to 38.9 m/s in the 12 hour delay case. The transfer

[6]The NRHO utilized is the 9:2 synodic resonant member of the southern L2 family.

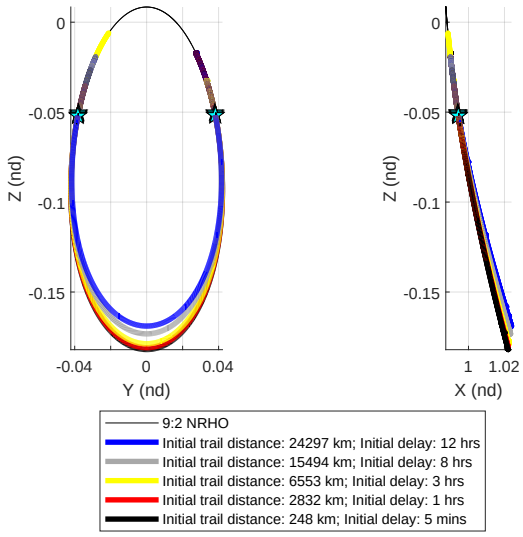


Fig. 6.11: NRHO Deterministic Optimal Rendezvous Trajectories

durations are likely unrealistic for manned chasers with durations between nominal ΔVs that vary little between cases: 5.29 days for the 12 hour delay, 5.76 days for the 5 minute delay.

As a connection between the NRI trajectory and the subject NRHO rendezvous trajectory, Figure 6.12 shows the resulting deterministic optimal trajectory with the initial state equal to the robust NRHO insertion state from Figure 6.8. The delay between chaser and target is adjusted to 6 minutes and 3.77 seconds such that the target is 100 km in the lead at NRI. The deterministic cost in this scenario is quite minimal, 0.379 m/s, with a transfer duration of 4.90 days.

Considering the minimal nominal ΔV, the 100 km separation at NRI case with only 0.379 m/s deterministic ΔV cost could potentially exhibit large variations when comparing the deterministic optimal trajectory to the robust equivalent. Two robust comparisons that apply the initial dispersion covariance at different locations are presented that answer slightly different questions. In Case 1, the initial dispersion covariance is applied at ΔV_1. This scenario finds the optimal robust combination without increasing the stochastic cost through additional coast in the first segment. In Case 2, the initial dispersion covariance is applied at the beginning of the trajectory and grows throughout the coast prior to ΔV_1.

- Initial Dispersion
  - $\sigma_r = 1$ km
  - $\sigma_v = 1$ m/s
- Maneuver Execution Error
  - $\sigma_{R_{\Delta V}} = 10$ cm/s
  - $\sigma_{R_{TCM}} = 10$ cm/s
- Process Noise
  - $\sigma_Q = 1\text{mm/s}/\sqrt{\text{s}}$
- Position Dispersion Constraint
  - $\sigma_{r,max} = 100$ m

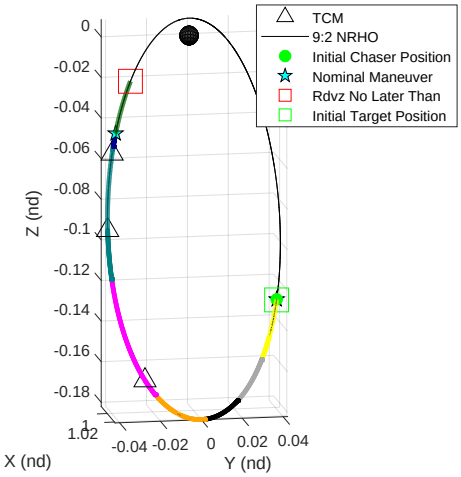


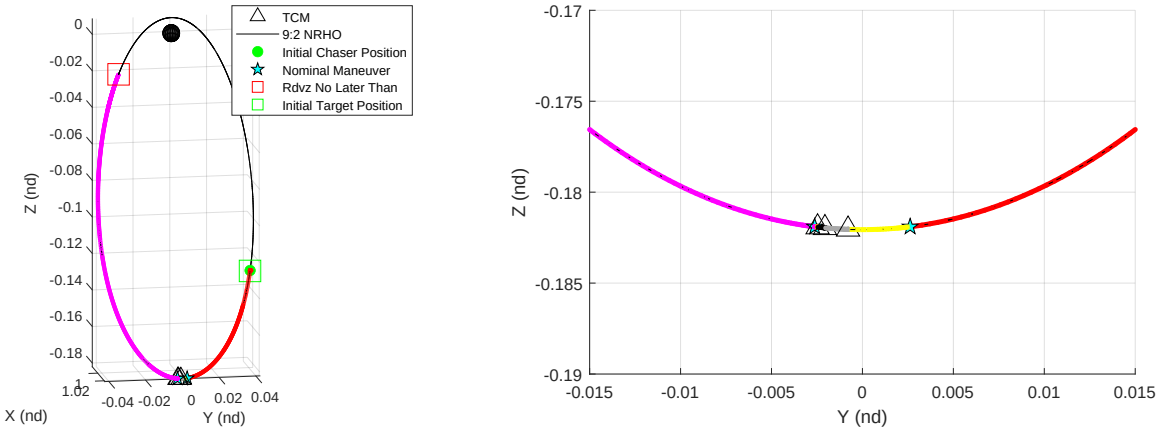
Fig. 6.12: Deterministic Optimal Rendezvous Post-NRI with Optimal TCMs

Case 2 finds the optimal robust rendezvous trajectory given the initial conditions at NRI and accounts for the stochastic cost increase of coasting in the first segment.

Figure 6.12 shows the error sources utilized in both robust trajectory cases. Due to the small nominal ΔV magnitude, the nominal ΔV error is reduced to 10 cm/s 1σ; such a small maneuver would likely be performed by a spacecraft RCS subsystem. Process noise is the larger of the values frequently used in this dissertation, simulating disturbances from a manned vehicle. The initial position dispersion matches the NRI target position dispersion constraint as a connection to the NRI trajectory development problem.

Figure 6.13 shows the robust trajectory result for Case 1 where the initial dispersion covariance is applied at ΔV_1. Figure 6.14 shows the robust trajectory result for Case 2 where the initial dispersion is applied at trajectory node 1 (the initial chaser position). Table 6.10 shows the cost comparison of the deterministic optimal with optimal TCMs, the Case 1 robust, and Case 2 robust trajectories. Transfer durations decrease from the deterministic optimal's 4.90 days to 5.20 hours for Case 1 and 9.14 hours for Case 2.

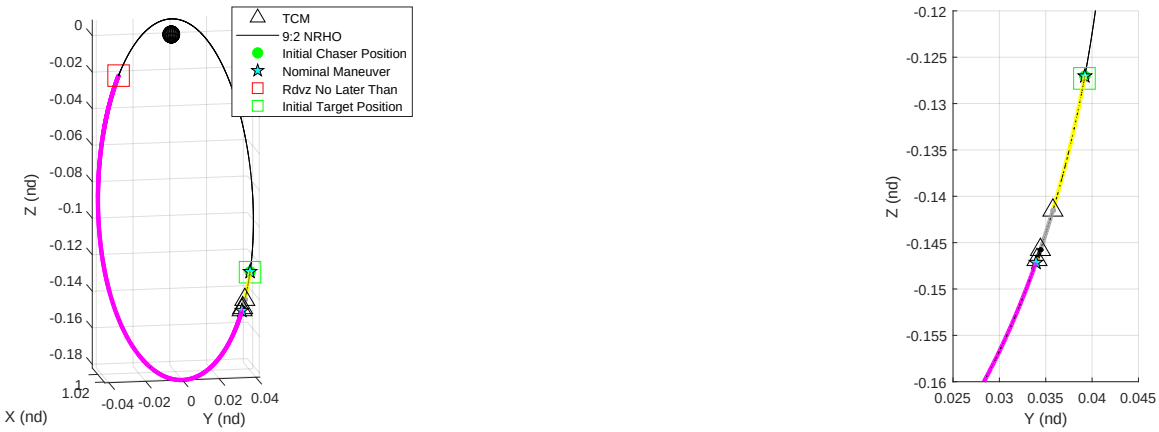
A small study was performed to compare the sensitivity of the robust trajectory to variations in process noise and target dispersion constraint. Figure 6.12 shows the baseline error sources for Case 1 and Case 2 which appear in the first and fourth rows of Table 6.11. Less process noise and a larger target position dispersion are tested to observe the



(a) Final Robust Trajectory

(b) Zoomed-in View of Rendezvous

Fig. 6.13: Robust NRHO Rendezvous Trajectory, Initial Dispersion Applied at ΔV_1



(a) Final Robust Trajectory

(b) Zoomed-in View of Rendezvous

Fig. 6.14: Robust NRHO Rendezvous Trajectory, Initial Dispersion Applied at Trajectory Node 1

Deterministic and Robust Cost Comparison (m/s)				
Deterministic	Nominal ΔV	0.379	$\|\Delta V_1\|$	0.215
	Optimal TCM 3σ δV	42.5	$\|\Delta V_2\|$	0.164
	ΔV Total Upper Bound	42.8		
Robust - Case 1	Nominal ΔV	3.81	$\|\Delta V_1\|$	1.91
	Optimal TCM 3σ δV	10.4	$\|\Delta V_2\|$	1.91
	ΔV Total Upper Bound	14.2		
Robust - Case 2	Nominal ΔV	4.46	$\|\Delta V_1\|$	1.94
	Optimal TCM 3σ δV	13.0	$\|\Delta V_2\|$	2.52
	ΔV Total Upper Bound	17.4		

Table 6.10: Robust NRHO Rendezvous Trajectory Cost Comparison

sensitivity of the robust results to error sources. The initial dispersion and maneuver execution error remain unchanged from Figure 6.12. Rather than replotting each trajectory, the NRHO true anomaly (TA) for ΔV_1 (TA1) and ΔV_2 (TA2) is shown to define the nominal trajectory. As the stochastic cost increases with more process noise or a more stringent target dispersion constraint, the optimal robust solution slides toward a more expensive (and shorter duration) nominal trajectory. The opposite is true for the case with less process noise and a less stringent target position dispersion constraint.

The ratio of nominal ΔV to TCM δV in this scenario is skewed heavily toward the TCM cost. As a result, there is a large opportunity to spend a little more nominal ΔV in exchange for significant TCM savings resulting in a drastic change to the optimal trajectory. The robust trajectory exhibits similar modifications but on a larger scale for this example: a shorter transfer duration and earlier arrival to the target state minimizes state dispersion growth through problem dynamics and random disturbances. Also, while not explicitly shown through additional figures, optimizing the number and locations of TCMs mitigates a significant increase in TCM δV with changes in error source magnitudes when compared to a fixed TCM set.

Case	1σ Error Sources		Cost (m/s)		True Anomaly	
	σ_Q	$\sigma_{r,max}$ **at** ΔV_2	Nom. ΔV	TCM 3σ δV	TA1	TA2
1	1	100 m	3.81	10.4	179.12°	180.88°
1	.1	100 m	3.00	10.2	179.12°	180.99°
1	1	1 km	2.32	7.59	178.52°	181.44°
2	1	100 m	4.46	13.0	163.13°	167.97°
2	.1	100 m	3.61	10.6	163.00°	169.94°
2	1	1 km	2.27	9.84	163.00°	168.68°

Table 6.11: Robust NRHO Rendezvous Trajectory Sensitivity to Process Noise and Target Position Dispersion Constraint

CHAPTER 7
RESULTS VERIFICATION

7.1 Stochastic Results Verification via Monte Carlo Analysis

The first comparison verifies that the modeled stochastic TCM estimates are representative of realistic magnitudes when compared to a Monte Carlo analysis. Each Monte Carlo sample represents a full pass through the trajectory with error sources incorporated as samples of random Gaussian error sources of the appropriate variance. In more detail, each dispersed trajectory in the Monte Carlo analysis begins with the initial mean position and velocity ($\bar{\boldsymbol{x}}_{0,i}$). The initial state dispersion is applied to create the dispersed initial state

$$\boldsymbol{x}\left(t_0\right) = \bar{\boldsymbol{x}}_{0,i} + \boldsymbol{e}_{P_0} \tag{7.1}$$

where $\boldsymbol{e}_{P_0}$ is a zero-mean Gaussian sampling of the initial state dispersion covariance matrix P_0 (the position and velocity dispersion of which have equal variances in all three directions as modeled). The dispersed initial state $\boldsymbol{x}\left(t_0\right)$ is numerically integrated the appropriate Δt to the next random trajectory event via the 4th-order Runge-Kutta algorithm with fixed time steps (Δt_d). Random accelerations (process noise) are incorporated discretely at each time step as an addition to the first-order velocity dynamics:

$$\dot{\boldsymbol{v}}\left(\boldsymbol{x}\right) = \boldsymbol{a}\left(\boldsymbol{x}\right) + \boldsymbol{q}_d \tag{7.2}$$

where the discrete contribution of process noise $\boldsymbol{q}_d$ is a white noise sampling scaled by the fixed numerical integration time step:

$$\boldsymbol{q}_d = \sqrt{\frac{\boldsymbol{e}_Q}{\Delta t_d}} \tag{7.3}$$

$\boldsymbol{e}_Q$ is a zero-mean Gaussian sampling of the 3×3 process noise matrix Q (also equally distributed in all three directions), which is sampled a single time and held constant for all four Runge-Kutta function calls per time step Δt_d. In this manner, uncorrelated random accelerations are discretely incorporated throughout each dispersed trajectory.

Once at the time corresponding to a nominal maneuver or TCM event, t_e, the actual ΔV that achieves the target position is iteratively calculated using Matlab's fsolve, with the nominal maneuver or nominal TCM serving as the initial guess. The calculated impulse is applied to the dispersed state along with the appropriate maneuver execution error:

$$\boldsymbol{x}\left(t_e\right)^{+} = \boldsymbol{x}\left(t_e\right)^{-} + \Delta \boldsymbol{V} + \boldsymbol{e}_R \tag{7.4}$$

where $\boldsymbol{e}_R$ is the Gaussian maneuver execution error defined by the appropriate execution error covariance matrix $R_{\Delta V}$ or R_{TCM}. The process is repeated until each trajectory event and target time have been achieved. Following the modeling of 1,000 individual dispersed trajectories, the statistics of the magnitude of each correction form the metric by which the modeled results in this dissertation will be compared. The magnitude of a vector is no longer a Gaussian random variable: all values are positive (it is not zero mean) and it is not symmetric about the mean. As such, to produce a 3σ-equivalent value for comparison, a percentile calculation is used to compute a value that contains 99.7% of the sample magnitudes. For corrected nominal maneuvers, the value used for comparison from the Monte Carlo analysis is the change in magnitude from the nominal maneuver. Appendix C presents a separate Monte Carlo analysis comparing the accumulated state dispersion covariance matrix from discrete process noise incorporation described here to the numerical integration of the QBM method.

Table 7.1 shows the results of a 1,000 run Monte Carlo analysis alongside the stochastic estimates for the robust NRHO insertion trajectory with a free initial orbital plane in Section 6.3 (Figure 6.9). The linear covariance-based estimates are conservative, however not excessively large. The conservatism of the results is expected; 3 RSS of the TCM covariance matrix bounds the worst case direction of covariance alignment [53]. These results

Correction	Monte Carlo 99.7% Magnitude (m/s)	Linear Covariance-based 3σ RSS (m/s)
ΔV_1 Corr.	5.23	3.84
TCM 1	37.4	42.8
TCM 2	1.73	1.78
TCM 3	2.17	2.88
TCM 4	1.89	2.63
TCM 5	1.57	2.30
TCM 6	1.60	2.03
TCM 7	1.47	1.79
TCM 8	1.24	1.63
TCM 9	37.5	51.1
TCM 10	2.57	2.84
TCM 11	1.78	2.44
TCM 12	1.54	2.17
ΔV_3 σ_v	1.14	1.30
Total	**99.8**	**121.5**

Table 7.1: Robust NRI Monte Carlo Verification Results

verify that the robust trajectory design method is producing results that are representative stochastic estimates given the error sources present. The magnitude of the correction to ΔV_1 appears to be an outlier in the trend of conservatism with the linear covariance-based estimates. A specific reason has not been definitively determined; hypotheses include a nonlinearity that is not being accounted for in the corrected maneuver savings model, or a potential issue with the fixed time-step Runge-Kutta integration and a loss of accuracy when compared to variable-step variable-order integration being used for the nominal trajectory.

7.2 Optimal TCM Set Verification

The purpose of this verification step is to test the optimality of the TCM set chosen using a genetic algorithm. Two sets of results are verified: first, whether the TCM opti-

Correction	Fast Opt TCM Time (hrs)	GA TCM Time (hrs)
TCM 1	0.33994	0.33994
TCM 2	25.945	25.792
TCM 3	44.974	44.757
TCM 4	55.351	55.264
TCM 5	60.846	60.784
TCM 6	63.576	63.556
Total 3σ TCM Magnitude (m/s)	**54.3026**	**54.3024**

Table 7.2: Optimal TCM Set Verification Using Genetic Algorithm Along LEO to LOI Trajectory

mization scheme along a nominal trajectory described in Section 5.5.1 finds the optimal TCM set; second, if the nonlinear program converges to the optimal set when finding the robust trajectory. The total number of TCMs is not modified in the genetic algorithm search. Each of the following genetic algorithm searches is allowed 500 generations with a population size of 100.

The first TCM set verification is along the deterministic optimal LEO to LLO trajectory in Section 6.1 (Figure 6.1b). Table 7.2 shows a comparison of the TCM time result from the fast optimization TCM method along the deterministic optimal trajectory as well as the genetic algorithm results. Table 7.2 also shows the total cost of each TCM set. While the fast TCM optimization method did not find a minimum TCM cost as low as the genetic algorithm's converged value, the genetic algorithm only reduced the cost an additional 0.0004%. The fast TCM optimization with a single target runs in 9 seconds while the genetic algorithm takes 68 seconds in this instance.

The second TCM set verification is along the robust LEO to powered lunar flyby to NRI trajectory with free initial orbital plane (Figure 6.9). The main difference in this case is the existence of two TCM targets at the second and third nominal maneuvers. Table 7.3 shows the comparison between the fast optimization method along the robust trajectory and the genetic algorithm results as well as the nonlinear program converged TCM set.

Correction	NLP Converged TCM Time (hrs)	Fast Opt TCM Time (hrs)	GA TCM Time (hrs)
TCM 1	0.02157	0.02158	0.02174
TCM 2	6.281	6.2225	5.1273
TCM 3	45.311	49.351	43.946
TCM 4	75.927	83.769	75.377
TCM 5	95.037	-	94.933
TCM 6	105.68	103.15	105.68
TCM 7	110.67	110.38	110.68
TCM 8	112.69	112.69	112.69
TCM 9	114.06	114.09	114.06
TCM 10	126.89	126.59	127.87
TCM 12	135.80	135.63	136.01
TCM 13	139.13	139.13	139.13
Total 3σ TCM Magnitude (m/s)	**121.49**	**122.05**	**121.47**

Table 7.3: Optimal TCM Set Verification Using Genetic Algorithm

The fast optimization method results in a total 3σ TCM magnitude of 122.0 m/s while the genetic algorithm produces a TCM set that costs 121.5 m/s. The fast TCM optimization in this case runs in 39 seconds without parallel processing while the genetic algorithm takes 126 seconds with parallel processing on 7 cores. All of the results are within 0.5% of each other. This instills confidence in the fast TCM optimization method and the accuracy of previous results presented.

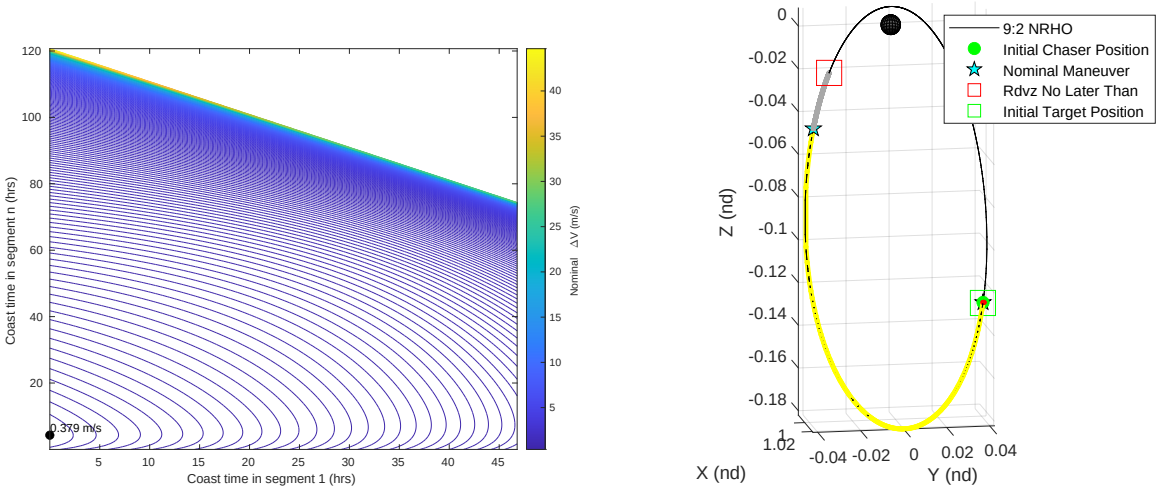
The third TCM set verification is along the deterministic NRI trajectory with a fixed initial orbital plane (Figure 6.7). Table 7.4 shows the comparison of the fast optimization TCM set and the genetic algorithm TCM set. In this instance, the genetic algorithm and the fast TCM optimization method produced results within 0.2% of each other. The genetic algorithm with 7 cores produced its results in 240 seconds while the fast TCM optimization method took 54 seconds in this instance.

Correction	Fast Opt TCM Time (hrs)	GA TCM Time (hrs)
TCM 1	0.33967	0.33967
TCM 2	16.483	16.065
TCM 3	26.842	26.337
TCM 4	33.218	32.990
TCM 5	71.453	71.453
TCM 6	84.999	84.999
TCM 7	88.766	88.768
TCM 8	90.552	90.546
TCM 9	108.12	106.05
TCM 10	115.10	113.88
TCM 11	116.92	116.92
Total 3σ TCM Magnitude (m/s)	**144.4**	**144.1**

Table 7.4: Optimal TCM Set Verification Using Genetic Algorithm

7.3 Robust Trajectory Verification

The purpose of this section is to verify a converged trajectory's optimality via a different optimization method. The trajectory chosen to verify is the two-burn NRHO rendezvous trajectory from Section 6.4, specifically "Robust - Case 2" from Table 6.10. This trajectory was chosen because it can be simplified to a problem of two independent variables with an associated cost. A mission map displaying the cost on the z-axis enables visual verification of two independent problem variables: coast duration in the first segment prior to nominal ΔV_1 and coast duration in the final segment following ΔV_2 (the rendezvous maneuver). The target begins 100 km in the lead in each scenario and follows the natural motion trajectory throughout the scenario. By prescribing the first and last segments' coast times the location of the nominal maneuvers is determined. The result is a fixed rendezvous time for each specific combination, but the transfer time and rendezvous location are variable across combinations. Each individual combination then represents a prescribed ΔV_1 and ΔV_2 position and fixed transfer duration, which has a single solution (similar to Lambert's



(a) Nominal ΔV Contour Plot

(b) Corresponding Minimum ΔV Trajectory from Contour Plot

Fig. 7.1: Nominal ΔV Optimization via Mission Map - NRHO Rendezvous

problem). The cost associated with each combination's singular solution is displayed on the mission map. The single differential correction method described in Subsection 3.1.2 is used to adjust the chaser velocity to meet the target position at the specified time.

The first step is verification of the deterministic optimal rendezvous trajectory and comparison to the NLP-converged solution in Figure 6.14. Figure 7.1 shows the contour plot for the deterministic rendezvous cost as a function of the coast time in the first and last segment and the corresponding minimum impulse trajectory. The contour represents an ideal solution space for use by a gradient-based solver with a single minimum value and a monotonic decrease toward the minimum. Validation of the optimal solution is successful as the result matches the NLP-converged deterministic optimal trajectory in Figure 6.12.

Next, the optimal TCM set cost is computed along each nominal trajectory in the mission map (Figure 7.1a). Figure 7.2 shows the contour of total 3σ TCM cost. One immediate observation is a nearly opposite cost incentive compared to nominal ΔV; an increase in the duration of coast in segment n increases the nominal ΔV cost while the opposite is true for 3σ TCM cost. Another observation is the apparent sub-structure and lack of smoothness in the TCM cost contour. This is an artifact of the optimal TCM set being limited to the

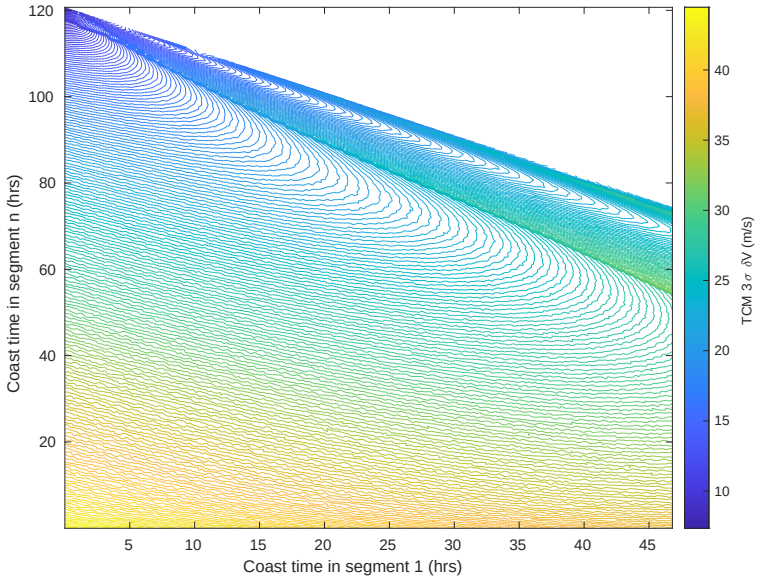
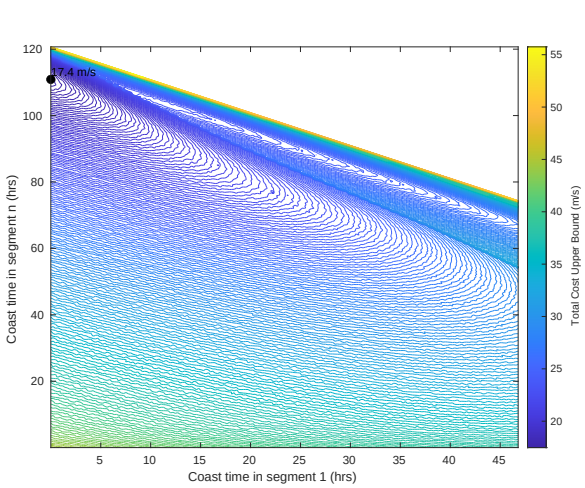


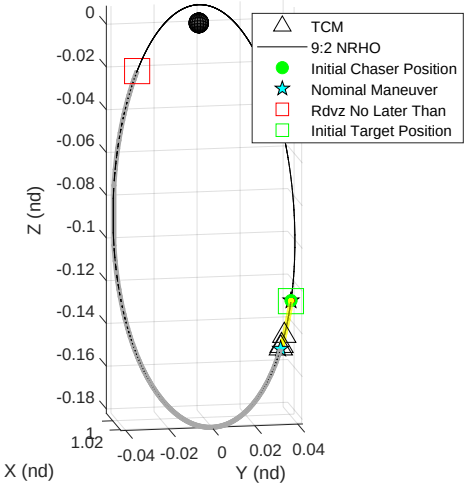
Fig. 7.2: Optimal 3σ TCM Cost Mission Map

set of discretized time increments generated during numerical integration and demonstrates the need for assigning TCMs at segment intersections during optimization. This plot was attempted to be smoothed by increasing the number of nonlinear integration time steps, however, it does not completely rectify the issue. Without the attempted smoothing as shown (which is otherwise costly in terms of additional computation required), the gradient frequently changes sign in an unpredictable way, preventing gradient-based convergence.

Finally, the robust mission map shows the combination of nominal ΔV plus 3σ TCM δV for the purpose of verifying the robust trajectory results in Figure 6.14 and Table 6.10. Figure 7.3a shows the robust mission map contour representing the total cost upper bound for each trajectory. The minimum value is identified on the contour plot and the corresponding robust trajectory is shown in Figure 7.3b. The results successfully verify the robust NRHO rendezvous trajectory identified in Section 6.4 is the two-impulse local minimum cost solution. The solution space is mostly monotonically decreasing toward the minimum value with the exception of a ridge that forms with long duration coasts in segments 1 and n (representing short duration transfers near apolune).



(a) Total Cost Upper Bound Contour Plot



(b) Corresponding Minimum Total Cost Robust Trajectory from Contour Plot

Fig. 7.3: Robust Optimization via Mission Map - NRHO Rendezvous

CHAPTER 8
CONCLUSIONS, FINAL REMARKS, AND FUTURE WORK

8.1 Conclusions

This dissertation presented a robust trajectory design method that incorporates stochastic error sources and the cost of correcting those errors into the optimization cost function alongside total deterministic ΔV. The cost of error correction is estimated using linear covariance analysis. Major mission events such as nominal maneuvers serve as the target for TCMs and an inequality constraint on the RSS of the position dispersion covariance enforces a maximum magnitude of the target position dispersion at corresponding events. The robust trajectory is found using nonlinear programming whereby the gradient of the cost function (nominal ΔV plus 3σ TCM δV) informs modifications to the nominal trajectory in the direction of a cheaper solution. Analytical gradients of the cost function and constraints are implemented and derived herein. The partial derivative of linear covariance-based cost estimates requires state transition tensors. Random disturbances are modeled as process noise and incorporated as an accumulated dispersion covariance between covariance updates (referred to as the Q-bar matrix). The analytical gradient of the Q-bar matrix requires the propagation of second-order Q-bar state sensitivities (referred to as the Q-bar tensor). A single propagation of these terms provides the opportunity to perform rapid calculation and comparison of the error and TCM estimates along a nominal trajectory.

Chapter 2 presented the mathematical background to be applied throughout the dissertation to include the CR3BP theory and second-order state transition tensor theory. Chapter 3 began investigating the trajectory design and optimization problem in the CR3BP, but only in a deterministic sense.

Chapter 4 results included two-body trajectories robust to initial state dispersion only. A useful conclusion from the results informs what scenarios are likely to benefit most by

pursuing a robust trajectory over a deterministic optimal trajectory. Short duration trajectories with relatively small nominal ΔV serve to benefit the most with a robust trajectory: with a short trajectory, there is less time to correct a large error, therefore the correction magnitude must be greater. The ratio of cost function terms is important as well. If the nominal ΔV is much greater than the cost of correcting errors, the cost function is skewed toward optimizing nominal ΔV. Long duration scenarios with large nominal ΔV magnitudes are expensive nominally and have more time to correct large errors at a cheaper cost.

Chapter 5 introduced maneuver execution error, random disturbances as process noise, multiple TCMs, and began exploring the effect of changing the magnitude of error sources on the resulting TCM set formed by the fast TCM optimization method presented. With four total error sources, a sensitivity analysis demonstrated the impact of changing each error source individually. Initial dispersion is generally expensive to correct with the first TCM but also has the potential to create large velocity dispersion at future trajectory events that can be very expensive to correct. Nominal maneuver execution error is generally corrected with a single TCM after the nominal maneuver and has a minimal impact on the remainder of the TCMs, however, there are exceptions. Since the optimal TCM solution includes numerous TCMs in many cases, TCM execution error does have a large impact on the location, number, and total cost of the TCM solution. Variations in process noise have a large impact on the resulting TCM solution as well. One conclusion is that optimizing TCMs has the potential to greatly mitigate the cost of increasing error sources, which was demonstrated by the comparison with the "looks about right" TCM solution.

Chapter 6 presented robust cislunar trajectory results and included all of the error sources and multiple TCM opportunities introduced in Chapter 5. Specific trajectories include a two-impulse LEO to LLO, a three-impulse LEO to LLO trajectory, a three-impulse LEO to powered lunar flyby to NRHO insertion trajectory, and a two-impulse NRHO rendezvous trajectory. A similar conclusion to Chapter 4 was made: the ratio of nominal ΔV to 3σ TCM δV is important. In cases where the nominal ΔV is very

expensive, the percentage savings by pursuing a robust trajectory is likely going to be small. In some cases, like the LEO to LLO trajectory, there is not a difference between the deterministic optimal and robust trajectory because the design space and dynamical connection being leveraged traversing the lunar corridor does not offer an opportunity to spend more nominal ΔV for TCM savings. Introducing additional flexibility into the design space with an additional nominal maneuver may provide that opportunity though; the three-impulse LEO to elliptical Earth departure to LLO scenario does offer savings for the robust trajectory. The three-impulse LEO to powered lunar flyby to NRHO insertion trajectory also provides flexibility to incorporate robust trajectory savings, however, the scenarios are multi-faceted regarding where the robust trajectory provides savings. The savings are found by minimizing a large velocity dispersion at the correction after powered flyby, which is very expensive to correct at that instance due to the high nominal velocity magnitude. Finally, the results from a two-impulse NRHO rendezvous scenario show that when the correction cost is now much larger than the nominal ΔV, the trajectory exhibits large changes through the shortening of the overall duration which minimizes the cost of correcting accumulated error throughout.

Chapter 7 verified robust trajectory results in three ways. First, a Monte Carlo analysis verified the linear-covariance based correction costs for a LEO to powered lunar flyby to NRHO trajectory. Second, a genetic algorithm was used to verify the optimal TCM execution times along a resulting robust trajectory. Third, in the two-impulse NRHO rendezvous, contours of the solution space were developed to verify that the nonlinear program result visually converged to the plotted local minimum value.

8.2 Future Work

There are numerous paths for next steps and future work. Major categories include incorporating navigation error into the method, incorporating this method into a trajectory design tool like COPERNICUS, improving run time, incorporating indirect optimization, further utilizing higher-order propagated terms to increase the accuracy of correction estimates, and the continued search for other useful robust trajectories.

Incorporating navigation error has the potential to further increase the benefit of taking a robust trajectory. The simplest approach would be including a standard navigation error value when performing calculations. However, a really interesting approach could be modeling the navigation problem in parallel. A robust trajectory result could then benefit from taking a path that maximizes information gained by optimizing measurement geometry and in turn minimizing the cost of corrections.

A hopeful next step is incorporating stochastics into a trajectory design tool like COPERNICUS. When finding a deterministic optimal trajectory, the tool in this dissertation is somewhat similar to a tool like COPERNICUS. Incorporating stochastics could open the aperture as far as robust trajectory discoveries go and enabling a larger group of people to search for a robust trajectory.

There are also opportunities for improving the run-time of each NLP iteration. Currently, each nominal trajectory requires numerically integrating 510 equations along the nominal trajectory each iteration. A simple improvement to run time is parallelizing the propagation of each trajectory segment since it is not required to propagate the segments sequentially. Another opportunity for reducing run time is incorporating directional state transition tensors, whereby the second-order STT's 216 equations could be reduced to as few as 6 equations by identifying and propagating the STT only in the most important one or two directions, as identified by the eigenvalues and eigenvectors of the left-hand Cauchy Green Tensor (CGT). [54] Some testing revealed that it may also be possible to propagate a directional covariance value using both the left and right-hand CGT, similarly reducing the number of equations required for the QBT.

Other improvements to the robust trajectory design method could include the incorporation of indirect optimization techniques or approximations to enable convex optimization of parts of the problem. Specific to the TCM optimization problem, it seems possible to develop an analog to primer vector theory specific to optimizing the number and location of TCMs along a nominal trajectory, or at least utilizing an indirect approach to verify a solution's optimality. Similarly, there is also still room to improve the fast TCM optimiza-

tion method along a nominal trajectory, such as using analytical gradients to take more efficient steps toward a minimum value.

Finally, linear covariance analysis utilizes first-order terms (STMs) along the trajectory to perform the stochastic estimates. There reaches a point where linearization relative to the nominal trajectory becomes inaccurate. Incorporating higher order terms into estimates has the potential to expand the region of accuracy with respect to the nominal trajectory. However, it also introduces some challenges as the solution for a correction utilizing higher order terms requires an iterative solution method as opposed to the closed-form solution using only first-order terms.

A large portion of future work is continuing the search for robust trajectories and building intuition regarding where the most potential benefit lies. The author has begun an investigation in the bicircular restricted four body problem to identify any robust benefits to the long duration ballistic lunar transfer. Investigating robust trajectories with TCM restrictions, such as crew sleep avoidance, could reveal interesting robust results as well. Another interesting problem is related to robust trajectory guidance once a trajectory has been departed upon and error sources begin to be realized. For example, after the error is realized following the initial nominal maneuver, is a tool like NASA's two-level targeter implemented or is there benefit to be gained by incorporating stochastics into a guidance and targeting algorithm. Finally, more than three nominal maneuvers could be considered to link together additional mission events. This dissertation only considered up to three nominal maneuvers, however, the method is not limited to three. The design space likely becomes more complicated with additional nominal maneuvers so additional challenges should be anticipated.

REFERENCES

[1] Lee, D. E., "White Paper: Gateway Destination Orbit Model: A Continuous 15 Year NRHO Reference Trajectory," Aug. 2019, NTRS Author Affiliations: NASA Johnson Space Center NTRS Report/Patent Number: JSC-E-DAA-TN72594 NTRS Document ID: 20190030294 NTRS Research Center: Johnson Space Center (JSC). 1.1

[2] Cheetham, B., "Cislunar Autonomous Positioning System Technology Operations and Navigation Experiment (CAPSTONE)," *ASCEND 2021*, American Institute of Aeronautics and Astronautics, _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2021-4128. 1.1

[3] Conway, B. A., *Spacecraft trajectory optimization*, Cambridge University Press, Jan. 2010. 1.2.1

[4] Rutherfobd, D. E., "Optimal Trajectories For Space Navigation. By D. F. Lawden. Pp. viii, 126. 21s. net. 1963. (Butterworth and Co.)," *The Mathematical Gazette*, Vol. 48, Dec. 1964, pp. 478–479, ISSN: 0025-5572, 2056-6328 Issue: 366 Journal Abbreviation: Math. Gaz. 1.2.1

[5] Lion, P. M. and Handelsman, M., "Primer vector on fixed-time impulsive trajectories." *AIAA Journal*, Vol. 6, No. 1, Jan. 1968, pp. 127–132. 1.2.1

[6] Jezewski, D. J., "Primer vector theory and applications," NASA Technical Report NASA-TR-R-454, Nov. 1975. 1.2.1

[7] JEZEWSKI, D. J. and ROZENDAAL, H. L., "An efficient method for calculating optimal free-space n-impulse trajectories." *AIAA Journal*, May 2012. 1.2.1

[8] Prussing, J. and Chiu, J.-H., "Optimal multiple-impulse time-fixed rendezvous between circular orbits," *Astrodynamics Conference*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, Aug. 1984. 1.2.1

[9] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, March 1998, pp. 193–207, Publisher: American Institute of Aeronautics and Astronautics. 1.2.1

[10] Whitley, R. and Ocampo, C., "Direct Multiple Shooting Optimization with Variable Problem Parameters," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, Orlando, Florida, Jan. 2009. 1.2.1, 4.1.1, D

[11] Vavrina, M., Englander, J., and Ellison, D., "Global Optimization of N-Maneuver, High-Thrust Trajectories Using Direct Multiple Shooting," Feb. 2016. 1.2.1

[12] Ocampo, C., "AN ARCHITECTURE FOR A GENERALIZED SPACECRAFT TRAJECTORY DESIGN AND OPTIMIZATION SYSTEM," *Libration Point Orbits and Applications*, WORLD SCIENTIFIC, Aiguablava, Spain, May 2003, pp. 529–571. 1.2.1

[13] Ocampo, C., "Finite Burn Maneuver Modeling for a Generalized Spacecraft Trajectory Design and Optimization System," *Annals of the New York Academy of Sciences*, Vol. 1017, No. 1, 2004, pp. 210–233. 1.2.1

[14] Goodman, J. L., "History of Space Shuttle Rendezvous," Oct. 2011, NTRS Author Affiliations: United Space Alliance NTRS Report/Patent Number: JSC-CN-24483 NTRS Document ID: 20110023479 NTRS Research Center: Johnson Space Center (JSC). 1.2.1

[15] Nishimura, T. and Pfeiffer, C. G., "A dynamic programming approach to optimal stochastic orbital transfer strategy," *Journal of Spacecraft and Rockets*, Vol. 7, No. 4, April 1970, pp. 398–404, Publisher: American Institute of Aeronautics and Astronautics. 1.2.2

[16] Jin, K., Geller, D. K., and Luo, J., "Robust Trajectory Design for Rendezvous and Proximity Operations with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, April 2020, pp. 741–753. 1.2.2

[17] Oguri, K. and McMahon, J. W., "Robust Spacecraft Guidance Around Small Bodies Under Uncertainty: Stochastic Optimal Control Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, 2021, pp. 1295–1313, Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G005426. 1.2.2

[18] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, Jan. 2009, pp. 102–111. 1.2.2

[19] Jenson, E. L. and Scheeres, D. J., "Multi-Objective Optimization of Covariance and Energy for Asteroid Transfers," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, July 2021, pp. 1253–1265. 1.2.2

[20] Boone, S. and McMahon, J., "Semi-Analytic Spacecraft Maneuver Design with Stochastic Constraints," *2022 American Control Conference (ACC)*, IEEE, Atlanta, GA, USA, June 2022, pp. 1608–1613. 1.2.2

[21] Greco, C., Di Carlo, M., Vasile, M., and Epenoy, R., "Direct multiple shooting transcription with polynomial algebra for optimal control problems under uncertainty," *Acta Astronautica*, Vol. 170, May 2020, pp. 224–234. 1.2.2

[22] Ocampo, C., Senent, J. S., and Williams, J., "Theoretical Foundation of Copernicus: A Unified System for Trajectory Design and Optimization," Madrid, May 2010, NTRS Author Affiliations: Texas Univ., Odyssey Space Research, LLC, Engineering Research and Consulting, Inc. NTRS Report/Patent Number: JSC-CN-20552 NTRS Document ID: 20100017708 NTRS Research Center: Johnson Space Center (JSC). 1.2.2

[23] Grebow, D. J., *Trajectory design in the Earth-Moon system and lunar South Pole coverage*, Ph.D. Dissertation, Purdue University, School of Aeronatics and Astronautics, Jan. 2010. 1.2.2

[24] Bosanac, N., "Leveraging Natural Dynamical Structures to Explore Multi-Body Systems," Aug. 2016. 1.2.2, 1.2.3, 3.1.3

[25] Ellison, D. H., Conway, B. A., Englander, J. A., and Ozimek, M. T., "Analytic Gradient Computation for Bounded-Impulse Trajectory Models Using Two-Sided Shooting," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 7, July 2018, pp. 1449–1462. 1.2.2, 4.2.5

[26] Grebow, D., "Generating periodic orbits in the circular restricted three-body problem with applications to lunar south pole coverage," . 1.2.3, 3.1.2, 3.1.2

[27] Zimovan-Spreen, E. M. and Howell, K. C., "DYNAMICAL STRUCTURES NEARBY NRHOS WITH APPLICATIONS IN CISLUNAR SPACE," , pp. 20. 1.2.3, 3.1.2

[28] Farquhar, R. W., "The Utilization of Halo Orbits in Advanced Lunar Operations," Tech. Rep. X-551-70-449, Dec. 1970, NTRS Author Affiliations: NASA Goddard Space Flight Center NTRS Document ID: 19710005876 NTRS Research Center: Legacy CDMS (CDMS). 1.2.3

[29] Howell, K. and Breakwell, J., "Almost rectilinear halo orbits," *20th Aerospace Sciences Meeting*, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics, Jan. 1982. 1.2.3

[30] Olikara, Z. P., "Computation of Quasi-Periodic Tori in the Circular Restricted Three-Body Problem," , pp. 80. 1.2.3

[31] Baresi, N., Olikara, Z. P., and Scheeres, D. J., "Fully Numerical Methods for Continuing Families of Quasi-Periodic Invariant Tori in Astrodynamics," *The Journal of the Astronautical Sciences*, Vol. 65, No. 2, June 2018, pp. 157–182. 1.2.3

[32] Koon, W., Lo, M., Marsden, J., and Ross, S., *Dynamical Systems, the Three-Body Problem, and Space Mission Design*, Marsden Books, 2011. 1.2.3, 2.1.4, 2.2

[33] Bosanac, N., "BOUNDED MOTIONS NEAR RESONANT ORBITS IN THE EARTH-MOON AND SUN-EARTH SYSTEMS," , pp. 20. 1.2.3

[34] Perko, L., *Differential equations and dynamical systems*, No. 7 in Texts in applied mathematics, Springer, New York, 3rd ed., 2001. 1.2.3, 2.1.4

[35] Kelly, S. and Geller, D., "Optimal Robust Two-Body Trajectory Design with Corrective Maneuvers," *Journal of Spacecraft and Rockets*, Aug. 2023, pp. 1–17, Publisher: American Institute of Aeronautics and Astronautics. 1.3, 1.4, 1, 5.4.1

[36] Szebehely, *Theory of Orbit*, Academic Press, Place of publication not identified, 1967. 2.1.4, 2.1.4, 2.1.4

[37] Poincare, H. and Magini, R., "Les methodes nouvelles de la mecanique celeste," *Il Nuovo Cimento (1895-1900)*, Vol. 10, No. 1, July 1899, pp. 128–130. 2.1.4

[38] Szebehely, V. and Giacaglia, G. E. O., "On the elliptic restricted problem of three bodies," *The Astronomical Journal*, Vol. 69, April 1964, pp. 230. 2.1.4

[39] Broucke, R., "Stability of periodic orbits in the elliptic, restricted three-body problem." *AIAA Journal*, Vol. 7, No. 6, June 1969, pp. 1003–1009. 2.1.4

[40] McCarthy, B. and Howell, K., "Ballistic Lunar Transfer Design to Access Cislunar Periodic and Quasi-Periodic Orbits Leveraging Flybys of the Moon," Oct. 2021. 2.1.4

[41] Scheuerle, S. T., "CONSTRUCTION OF BALLISTIC LUNAR TRANSFERS IN THE EARTH-MOON-SUN SYSTEM," . 2.1.4

[42] "SPICE Toolkit," . 2.1.4

[43] "Nasas Spice System Models the Solar System, International Astronomical Union Colloquium, Cambridge Core," . 2.1.4

[44] Pritchett, R. E., Zimovan, E., and Howell, K., "Impulsive and Low-Thrust Transfer Design Between Stable and Nearly-Stable Periodic Orbits in the Restricted Problem," *2018 Space Flight Mechanics Meeting*, American Institute of Aeronautics and Astronautics, Kissimmee, Florida, Jan. 2018. 2.1.4

[45] Park, R. S. and Scheeres, D. J., "Nonlinear Mapping of Gaussian Statistics: Theory and Applications to Spacecraft Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, Nov. 2006, pp. 1367–1375. 2.3, 2.3.2, 2.3.2, 2.3.2

[46] Park, R. S. and Scheeres, D. J., "Nonlinear Semi-Analytic Methods for Trajectory Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, Nov. 2007, pp. 1668–1676. 2.3

[47] Spreen, E. M. Z., "TRAJECTORY DESIGN AND TARGETING FOR APPLICATIONS TO THE EXPLORATION PROGRAM IN CISLUNAR SPACE," , pp. 248. 3.1.3

[48] Montenbruck, O. and Gill, E., *Satellite Orbits*, Springer, Berlin, Heidelberg, 2000. 3.2.1

[49] Maybeck, P. S., *Stochastic models, estimation and control*, No. v. 141 in Mathematics in science and engineering, Academic Press, New York, 1979. 4.1.2, 5.2, 5.2

[50] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, Jan. 1999. 1a

[51] Blanchard, R. C. and Lancaster, E. R., "A unified form of Lambert's theorem," Tech. Rep. NASA-TN-D-5368, Sept. 1969, NTRS Author Affiliations: NASA Goddard Space Flight Center NTRS Document ID: 19690027552 NTRS Research Center: Legacy CDMS (CDMS). 4.2.5

[52] Dannemiller, D., "Session Chair Introduction, Human Spaceflight/Deep Space Gateway Session, 44th Annual AAS Guidance, Navigation, & Control Conference, Feb 2022," Feb. 2022. 5.5.2

[53] Moon, Q. and Geller, D., "Regression Techniques for the Statistics of Correction Maneuver Magnitudes," Vol. Annual AAS Guidance & Control Conference, 24-026 (preprint), AAS/AIAA, Breckenridge, CO, 2024. 7.1

[54] Boone, S. and McMahon, J., "Directional State Transition Tensors for Capturing Dominant Nonlinear Effects in Orbital Dynamics," *Journal of Guidance, Control, and Dynamics*, Nov. 2022, Publisher: American Institute of Aeronautics and Astronautics. 8.2

[55] Morales, J. L., Nocedal, J., Waltz, R. A., Liu, G., and Goux, J.-P., "Assessing the Potential of Interior Methods for Nonlinear Optimization," *Large-Scale PDE-Constrained Optimization*, edited by L. T. Biegler, M. Heinkenschloss, O. Ghattas, and B. van Bloemen Waanders, Springer, Berlin, Heidelberg, 2003, pp. 167–183. D

[56] Nocedal, J., Wächter, A., and Waltz, R. A., "Adaptive Barrier Update Strategies for Nonlinear Interior Methods," *SIAM Journal on Optimization*, Vol. 19, No. 4, Jan. 2009, pp. 1674–1693. D

[57] Waltz, R., Morales, J., Nocedal, J., and Orban, D., "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Mathematical Programming*, Vol. 107, No. 3, July 2006, pp. 391–408. D

[58] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: AN SQP ALGORITHM FOR LARGE-SCALE CONSTRAINED OPTIMIZATION," . D

[59] Nocedal, J., Öztoprak, F., and Waltz, R. A., "An interior point method for nonlinear programming with infeasibility detection capabilities," *Optimization Methods and Software*, Vol. 29, No. 4, July 2014, pp. 837–854. D

[60] Byrd, R. H., Hribar, M. E., and Nocedal, J., "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM Journal on Optimization*, Vol. 9, No. 4, Jan. 1999, pp. 877–900, Publisher: Society for Industrial and Applied Mathematics. D

[61] "Constrained Nonlinear Optimization Algorithms - MATLAB & Simulink," . D

[62] Diana, "Color blind friendly colormap, MATLAB Central File Exchange," March 2024. E

APPENDICES

APPENDIX A
DERIVATIONS

A.1 Identity Derivation

The following concepts support the subsequent identity derivations. First, is the application of the matrix differential product rule which is applied with the same logic to the traditional product rule:

$$d(ABC) = d(A)BC + Ad(B)C + ABd(C)$$

The trace is a linear operator and follows another convenient rule set:

$$d(\mathrm{tr}\,(ABC)) = \mathrm{tr}\,(d(ABC))$$

Finally, as the trace operates on diagonal elements, interchanging the order of matrix multiplication (when the matrices are square) does not change the result:

$$\mathrm{tr}\,(AB) = \mathrm{tr}\,(BA)$$

or independent of squareness as long as the result is a square matrix

$$\mathrm{tr}\,(AB) = \mathrm{tr}\left(B^{\top}A^{\top}\right)$$

Applying these rules in evaluating the following when only matrix A is a function of $\bar{x}$ results in

$$\begin{aligned}
\frac{\partial \mathrm{tr}\left(CABA^\top C^\top\right)}{\partial \bar{x}} &= \begin{bmatrix} \mathrm{tr}\left[C\frac{\partial A}{\partial x_1}BA^\top C^\top + CAB\frac{\partial A}{\partial x_1}^\top C^\top\right] \\ \vdots \\ \mathrm{tr}\left[C\frac{\partial A}{\partial x_n}BA^\top C^\top + CAB\frac{\partial A}{\partial x_n}^\top C^\top\right] \end{bmatrix}^\top \\
&= \begin{bmatrix} \mathrm{tr}\left[\frac{\partial A}{\partial x_1}BA^\top C^\top C + C^\top CAB\frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \mathrm{tr}\left[\frac{\partial A}{\partial x_n}BA^\top C^\top C + C^\top CAB\frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top \\
&= \begin{bmatrix} \mathrm{tr}\left[C^\top CAB^\top\frac{\partial A}{\partial x_1}^\top + C^\top CAB\frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \mathrm{tr}\left[C^\top CAB^\top\frac{\partial A}{\partial x_n}^\top + C^\top CAB\frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top \\
&= \begin{bmatrix} \mathrm{tr}\left[C^\top C\left(AB^\top + AB\right)\frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \mathrm{tr}\left[C^\top C\left(AB^\top + AB\right)\frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top
\end{aligned}$$

When the matrix B is symmetric as is the case with a covariance matrix, the identity simplifies further:

$$\frac{\partial \mathrm{tr}\left(CABA^\top C^\top\right)}{\partial \bar{x}} = 2\begin{bmatrix} \mathrm{tr}\left[C^\top CAB\frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \mathrm{tr}\left[C^\top CAB\frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top$$

The case when matrices A and C are both functions of $\bar{x}$ simplifies following the same set of rules to the following expression. Two separate trace operators are applied in this

case as C is not square and the matrix multiplications result in different sized matrices.

$$\mathrm{tr}\frac{\partial CABA^\top C^\top}{\partial \bar{x}} = 2 \begin{bmatrix} \mathrm{tr}\left[CABA^\top \frac{\partial C}{\partial x_1}^\top\right] + \mathrm{tr}\left[C^\top CAB \frac{\partial A}{\partial x_1}^\top\right] \\ \vdots \\ \mathrm{tr}\left[CABA^\top \frac{\partial C}{\partial x_n}^\top\right] + \mathrm{tr}\left[C^\top CAB \frac{\partial A}{\partial x_n}^\top\right] \end{bmatrix}^\top$$

APPENDIX B

REFERENCE INFORMATION

B.1 Parameter Values Used

- Earth gravitational parameter: $\mu_E = 398600.4415$ km^3/sec^2
- Moon gravitational parameter: $\mu_M = 4902.8$ km^3/sec^2
- Circular lunar orbit radius: 384,400 km

B.2 Robust LEO to PLF to NRHO Insertion Parameters

(CR3BP positions, velocities, and time in non-dimensional distance, velocity, and time)			
Initial Pos. (ND)	-0.0195140726104922	**TLI Pos.** (ND)	-0.0195135906158413
	-0.0162187067684891		-0.0162189127918011
	-0.000332044828317289		-0.000332608222350025
Initial Vel. (ND)	4.66444762574509	**TLI Vel. After** ΔV_1 (ND)	6.51582635545753
	-1.99394107575154		-2.83385415712864
	-5.45226120271818		-7.63049481038387
LEO Coast (ND)	1.03332236237512e-07	ΔV_1 **to** ΔV_2 **Coast** (ND)	1.09343788419489
Flyby Position (ND)	0.988009871564714	**NRI Pos.** (ND)	1.00668674681398
	-0.00495904123495611		0.0374930477196215
	0.0015405279628845		-0.121392669208028
Flyby Vel. After ΔV_2 (ND)	0.229999903493435	**NRI Vel. After** ΔV_3 (ND)	0.101654785242203
	-1.24045668752913		0.149338745117038
	-1.73619605339098		-0.269420048398715
ΔV_2 **to** ΔV_3 **Coast** (ND)	0.255697143169389		

Table B.1: Robust NRI with Flexible Initial Orbital Plane Trajectory States and Durations in the CR3BP

APPENDIX C

PROCESS NOISE PROPAGATION VERIFICATION

This appendix section presents a Monte Carlo analysis for the purpose of comparing and verifying two methods of incorporating process noise. The first is the Q-bar matrix (QBM) numerical integration of accumulated state dispersion from process noise as described in Subsection 5.2 along a nominal trajectory via variable step variable order integration. The second is the discrete incorporation of process noise as random accelerations in a fourth order fixed time step Runge-Kutta (RK4) as described in Section 7.1. In the first method, a single propagation estimates the state dispersion covariance for a specific process noise power spectral density (PSD) Q relative to the nominal trajectory, which is applied continuously via Equation 5.21. In the second method, the dispersion covariance is calculated from 2,000 individual dispersed trajectories incorporating random accelerations at each discrete time step Δt_d.

For the Monte-Carlo analysis incorporating process noise discretely at fixed time steps, the following apply:

- Initial state: LEO departure from robust LEO to PLF to NRI
- Total integration time: 3.258 days (0.7502 ND time units)
- Fixed RK4 time step: 37.5 sec. ($1e-4$ ND time units)
- Process noise: $\sigma_{acc} = \sqrt{\frac{Q}{\Delta t}} = \frac{1\text{mm/s}/\sqrt{\text{s}}}{\sqrt{\Delta t}}$

The final state dispersion covariance matrix calculated from the Monte Carlo samples is

$$P_{MC} = \begin{bmatrix} 1.7559 & 0.23614 & -0.29620 & 4.1871 & -0.95950 & -0.38008 \\ 0.23614 & 0.38166 & -0.04381 & 0.96305 & 0.56841 & -0.08628 \\ -0.29620 & -0.04381 & 0.38235 & -0.70263 & 0.14672 & 0.67747 \\ 4.1871 & 0.96305 & -0.70263 & 11.026 & -1.5265 & -0.93943 \\ -0.95950 & 0.56841 & 0.14672 & -1.5265 & 2.5984 & 0.14518 \\ -0.38008 & -0.08628 & 0.67747 & -0.93943 & 0.14518 & 1.9289 \end{bmatrix} \times 10^{-7}$$

The numerically integrated QBM is

$$\bar{Q}_{t_1}(t_1, t_0) = \begin{bmatrix} 1.7489 & 0.24907 & -0.31684 & 4.3000 & -0.95456 & -0.45719 \\ 0.24907 & 0.37933 & -0.04884 & 0.99831 & 0.55612 & -0.08253 \\ -0.31684 & -0.04884 & 0.39734 & -0.75189 & 0.14696 & 0.70541 \\ 4.3000 & 0.99831 & -0.75198 & 11.441 & -1.5405 & -1.1070 \\ -0.95456 & 0.55612 & 0.14696 & -1.5405 & 2.5663 & 0.19359 \\ -0.45719 & -0.08253 & 0.70541 & -1.1070 & 0.19359 & 1.9655 \end{bmatrix} \times 10^{-7}$$

The percentage difference in diagonal terms of P_{MC} and $\bar{Q}_{t_1}(t_1, t_0)$ is

$$\%_{diff} = \begin{bmatrix} 1.6 & 0.61 & 3.7 & 3.6 & 1.3 & 1.9 \end{bmatrix}$$

with an average percent difference of 2.1%.

The total integration time used is 3.258 days in spite of the fact that powered lunar flyby occurs at 4.748 days past LEO departure. The propagated duration intentionally includes only one of the portions of the trajectory near a planetary body (Earth departure is used versus the portion arriving to lunar flyby) when coupled with the larger process noise value of $1\text{mm/s}/\sqrt{\text{s}}$. Without TCMs managing the total accumulated state dispersion from process noise, the QBM becomes large enough such that linearization no longer applies. Performing the Monte Carlo analysis from LEO to PLF results in average percent differences in Monte

Carlo dispersion covariance to QBM dispersion covariance diagonal terms in the 10-11% range. However, with a process noise of $0.1\text{mm/s}/\sqrt{\text{s}}$, the average error reduces to 1.0% encompassing the entire LEO to PLF portion of the trajectory. Based on these comparisons, the error between the Monte Carlo results and the QBM propagation appears to be related to problem nonlinearities and the error propagating the linear stochastic differential equation (Equation 5.21) versus the method utilized to incorporate process noise discretely with fixed time step RK4 integration.

APPENDIX D

NONLINEAR PROGRAMMING AND THE INTERIOR POINT METHOD

Potential optimization approaches include direct and indirect optimization. The author decided to predominantly pursue the direct optimization path via a category of numerical methods known as nonlinear programming. Many desired trajectories in the CR3BP have historically been found using differential correction, a creative implementation of Newton's method. This problem setup, however, does not enable optimization and also does not directly enable the use of inequality constraints. A literature search showed that common large-scale nonlinear programming approaches to direct optimization include software packages like SNOPT, IPOPT, KNITRO, etc [10, 55–58]. Two types of algorithms commonly employed within these optimization packages are sequential quadratic programming (SQP) and interior point (IP) methods (which incorporates aspects of direct and indirect optimization). The author pursued Matlab's nonlinear programming implementation of SQP and IP through the function fmincon (mainly due to availability) and generally observed more successful convergence with IP over SQP.

Matlab's IP method is based on the work by Nocedal et al. [59] The problem is to minimize a function $f(x)$ subject to equality constraints $h(x)$ and inequality constraints $g(x)$.

$$\min_{x} f(x), \text{ subject to } h(x) = 0 \text{ and } g(x) \leq 0$$

The IP method begins by recasting inequality constraints as equality constraints with the incorporation of a logarithmic barrier function with a slack variable, s, for each inequality constraint [56, 60, 61]. The barrier function effectively restricts the search for the optimal solution to the interior feasible space.

$$\min_{x,s} f_{\mu}(x,s) = \min_{x,s} f(x) - \mu \sum_{i} \ln(s_i), \text{ subject to } s \geq 0,\ h(x) = 0, \text{ and } g(x) + s = 0$$

Subsequently, there are two options for optimization steps. First, a direct step is the default option, where the Karush-Kuhn-Tucker (KKT) conditions of the barrier problem are evaluated, linearized, and a partial or full Newton step is taken toward the solution utilizing a line search to determine the step size until the best improvement is found. The KKT conditions are that the gradient of the Lagrangian is equal to zero at a minimum value

$$\nabla_x L\left(x, \lambda\right) = 0$$

and that the equality constraints with Lagrange multipliers are satisfied

$$\lambda_{g,i} g_i\left(x\right) = 0$$

The second step option is a conjugate gradient (CG) step, which is only taken when the direct step fails. The CG method attempts to adjust the problem parameters and slack variables to minimize a quadratic approximation of the problem and linearized equality constraints within a trust region. Convergence occurs when the KKT conditions are satisfied within a specified tolerance. The convergence parameters utilized for the interior point method within Matlab's function fmincon are as follows:

- Optimality tolerance: 1×10^{-6} (unitless)
- Step tolerance: 1×10^{-15}
  - In Matlab, this represents the lower bound on the size of a step from the previous to current parameter vector, which is a combination of nondimensional distance, velocity, and time.
- Constraint tolerance: $2 \times 10^{-10} \times m \times n$
  - m is the length of an individual segment parameter vector, equal to 7 in this analysis. n is the number of segments. Matlab compares the constraint tolerance to the sum of all constraint equations. This scales the allowable constraint violation with the number of problem parameters. This aggregate value is again a combination of nondimensional units.

APPENDIX E

ADDITIONAL CITATIONS

- Matlab colorblind colormap colors [62] were used for plotting trajectory segment colors.

CURRICULUM VITAE

Scott J. Kelly

Published Journal Articles

- Optimal Robust Two-Body Trajectory Design with Corrective Maneuvers, Scott Kelly and David Geller, *Journal of Spacecraft and Rockets*, vol. 60, number 6, August 2023.
- Robust Cislunar Trajectory Optimization in the Presence of Stochastic Errors, Scott Kelly and David Geller, *The Journal of the Astronatical Sciences*, vol. 71, number 30, June 2024.

Published Conference Papers

- Robust Cislunar Trajectory Optimization in the Presence of Stochastic Errors, Scott Kelly and David Geller, in *Proc. 46th AAS Guidance, Navigation, and Control Conf.*, Feb. 2024.
- Optimal Robust Two-Body Trajectory Design with Corrective Maneuvers, Scott Kelly and David Geller, in *Proc. 45th AAS Guidance, Navigation, and Control Conf.*, Feb. 2023.