

MISSION-MAPS FOR OUTBOUND CISLUNAR TRANSFER TRAJECTORIES

Carter Van Hammond*, David Woffinden[†], Luke Sauter[‡], Nathan Collins[§]

This study quantifies the robustness and sensitivity of an outbound cislunar trajectory for a lunar lander in the form of mission-maps, or topological maps that allows either a computer program or mission designer to intuitively *optimize* the placement of critical outbound correction burns from the derived sensitivity data. The non-linear multi-body dynamics are applied to generate an outbound cislunar reference profile used by a linear covariance analysis (LinCov) tool to compute the expected Δv and trajectory dispersions due to the initial state uncertainty, sensor errors, maneuver execution errors, and disturbance accelerations along the outbound cislunar profile. The rapid performance analysis capabilities of LinCov are complimented with parallel processing techniques to evaluate hundreds and thousands of different translational burn locations, placements, and targeting constraints to identify the combination that minimizes the total Δv usage (nominal plus $3\sigma \Delta v$) and trajectory dispersions at lunar orbit insertion. This study utilizes a generalized reference targeting algorithm to quickly assess the integrated closed-loop GN&C system performance due to different targeting configurations and constraints. The resulting mission maps provide an intuitive insight to ascertain each trajectory correction maneuver's (TCM) sensitivity to different burn times along an outbound cislunar trajectory and quickly identify desirable engineering trade-offs when performing analysis on the number and placement of these burns that nominally zero. Multiple mission maps are generated for a variety of different performance parameters that allow engineers to visually identify optimal solutions for trajectory correction maneuver placements, the number of correction burns, and the targeting constraints for each burn.

INTRODUCTION

As interest in exploring the lunar surface rises, space agencies around the world are planning and executing missions to cislunar space with unprecedented frequency, complexity, and scale. The flight regime shown in Figure 1 exemplifies a generic trans-lunar outbound trajectory that strives to lower overall propellant requirements, increase flexibility, reduce burn magnitudes, and increase lunar orbit capture feasibility. In order to follow an outbound cislunar trajectory like the trajectory described by Figure 1, robust mission planning will be essential for mission success and crew safety. When designing trajectories in this new environment, mission planners must account for uncertainties in the guidance, navigation, and control system (GN&C). Ideally, the selection of an outbound cislunar trajectory would account for expected and worst-case integrated GN&C performance in the presence of initial state dispersions, measurement errors, maneuver execution errors, and uncertainties in the navigation and environmental dynamics model.⁵ Furthermore, it is desirable that these missions and trajectories are optimized so that they may use minimal fuel while maximizing targeting accuracy. In this text, *robust* trajectory design refers to techniques that account for uncertainties in GN&C performance. A trajectory's *optimization* refers to using techniques to minimize a parameter such as fuel use and state dispersions. Thus, the *robust trajectory optimization* process in this study generates

*Cadet, Department of Astronautics, United States Air Force Academy, USAF Academy, CO, 80840

[†]Aerospace Engineer, GN&C Autonomous Flight Systems Branch, NASA Johnson Space Center, Houston TX, 77058

[‡]Department Head, Department of Astronautics, United States Air Force Academy, USAF Academy, CO, 80840

[§]Department Faculty, Department of Astronautics, United States Air Force Academy, USAF Academy, CO, 80840

The views expressed in this article, book, or presentation are those of the author and do not necessarily reflect the official policy or position of the United States Air Force Academy, the Air Force, the Department of Defense, or the U.S. Government.

maneuver profiles that minimize fuel usage or state dispersions while meeting specified mission objectives and constraints to a high probability (corresponding to 3σ from the expected value in this analysis).

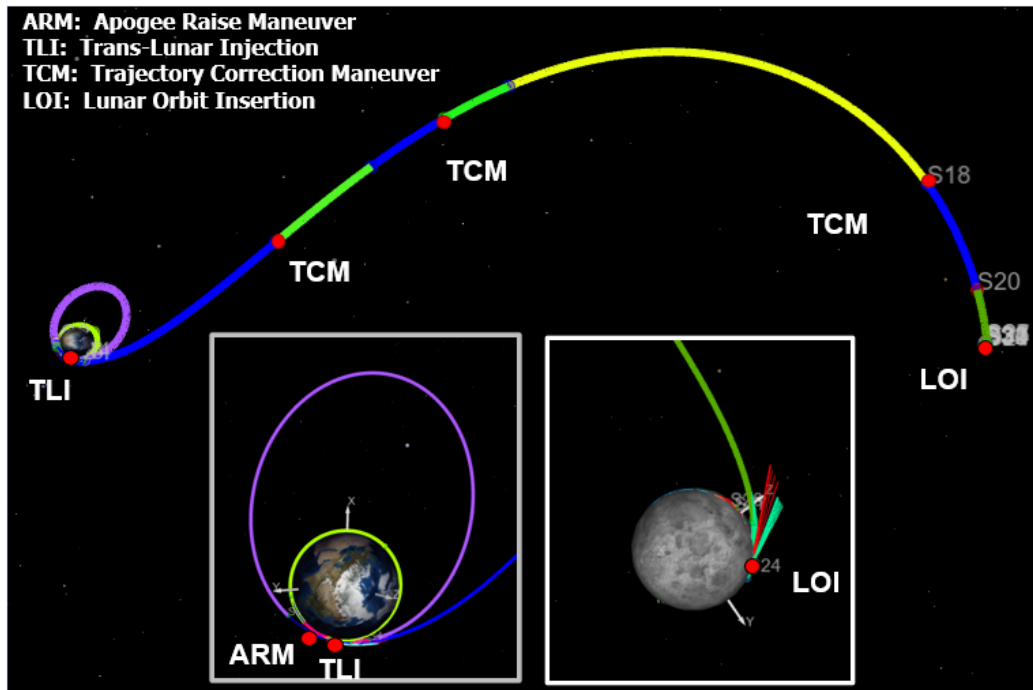


Figure 1: Outbound Cislunar Trajectory into Low Lunar Orbit

Traditionally, in the presence of uncertainties, robust mission planning involves using Monte Carlo analysis to find the distribution of Δv or state dispersions for a given reference/nominal trajectory.⁶ Monte Carlo simulations run a specific reference trajectory hundreds or thousands of times with random variables and initial conditions to generate statistical information (3σ dispersions), making Monte Carlo simulations computationally intensive with lengthy run times. Therefore, a Monte Carlo simulation and evaluation of a single reference trajectory might take on the order of minutes or hours. While this is feasible for reference trajectory validation, Monte Carlo analysis is infeasible for the numerical optimization process, which could require analyzing tens or hundreds of reference trajectory candidates to find the global minimum solution.

While the evaluation and validation of reference trajectories using Monte Carlo analysis can take minutes or hours, linear covariance analysis (LinCov) can accomplish equivalent statistical GN&C performance evaluations in just seconds. LinCov determines the relevant metrics by directly propagating (dynamics), updating (measurements), and correcting (making impulsive translational maneuvers) the covariance of the state dispersions and navigation error in time, all with a single simulation run. This is achieved by linearizing the dispersion dynamics, the navigation update equation, and the state impulsive correction about the nominal (reference) trajectory at each time step.³ The nominal trajectory is still propagated using non-linear dynamics. While this study does not directly use LinCov in a traditional optimization tool, like a particle swarm or genetic algorithm, it utilizes the computational speed of LinCov to generate *mission maps*, which enable the user to visually optimize a reference trajectory.

When optimizing an outbound cislunar trajectory, an optimization tool may require hundreds or thousands of reference trajectories to be tested. Therefore, when using Monte Carlo as the trajectory's propagation tool, the optimization of an outbound cislunar trajectory can take days or weeks of run time to generate meaningful solutions. When using LinCov as the trajectory's propagation tool, the optimization of an outbound cislunar trajectory can still take on the order of hours to days of run time. However, mission maps provide an optimization alternative that can be generated in just minutes. A *mission map*, described in Figure 2, resembles a

topological map and provides mission planners with the ability to visually optimize burn placements within their trajectory. This is achieved by choosing two nominally zero burns (traditionally trajectory correction maneuvers) to vary along the trajectory, specifying the time interval along which the burns should be varied, then recording 3σ Δv usage and position dispersions at lunar orbit insertion (LOI) through a cost function. The value assigned to this variation of the trajectory by the cost function describes the “height” of that point on the mission map. A mission map then displays this “height” data in a color-coded format, ultimately providing mission planners with the capability to perform rapid visual trajectory analysis and optimization by looking at the minimum and maximum points on a set of mission maps.

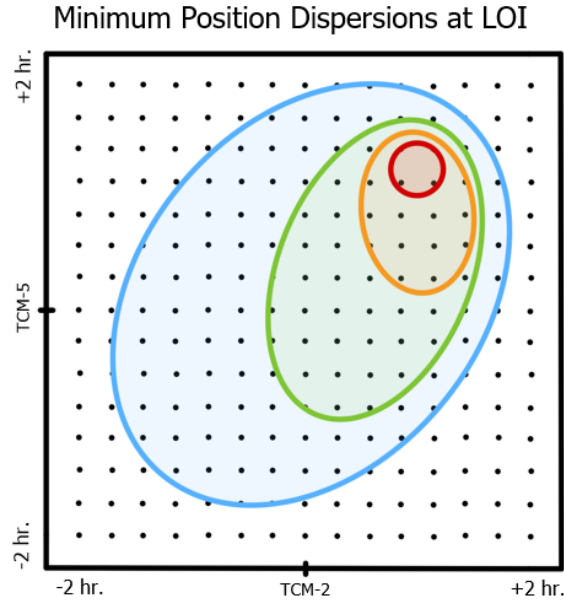


Figure 2: Mission Map Concept Drawing

This paper begins by describing linear covariance analysis, generalized reference targeting, and the targeting constraints matrix. It also describes the ground tracking model used to simulate the navigational environment of the spacecraft. Then, the cislunar trajectory scenarios are designed in order to test a variety of targeting constraints within well-defined and over-defined targeting matrices. Finally, these trajectory scenarios are used to generate sets of mission maps, which describe how the timing and placement of trajectory correction maneuvers along the outbound leg of a reference cislunar trajectory 3σ Δv usage and position dispersions at lunar orbit insertion.

ANALYSIS APPROACH

Performance Metrics

To determine the optimal performance for utilizing an angles-only navigation strategy for NRHO rendezvous, several key performance metrics must be defined to objectively quantify what is acceptable and what constitutes an integrated system failure. These metrics are derived from four states illustrated in Figure 3. The first is the true state $\mathbf{x}$ which is an n -dimensional vector that represents the *real world* environment or actual state. The nominal state $\bar{\mathbf{x}}$ is another n -dimensional vector that represents the desired or reference state. The navigation state $\hat{\mathbf{x}}$ is an $\hat{n}$ -dimensional vector ($\hat{n} < n$) that represents the filter’s estimated state. The design state $\mathbf{x}$, often assumed to be the true state, is used to design the onboard navigation filter.

The variation between these states constitute the actual performance metrics considered in this analysis, also depicted in Figure 3. These include the true trajectory dispersions $\delta\mathbf{x}$, the navigation dispersions $\delta\hat{\mathbf{x}}$, the true navigation error $\delta\mathbf{e}$, and the onboard navigation error $\delta\hat{\mathbf{e}}$. The true dispersions $\delta\mathbf{x}$ are defined as the

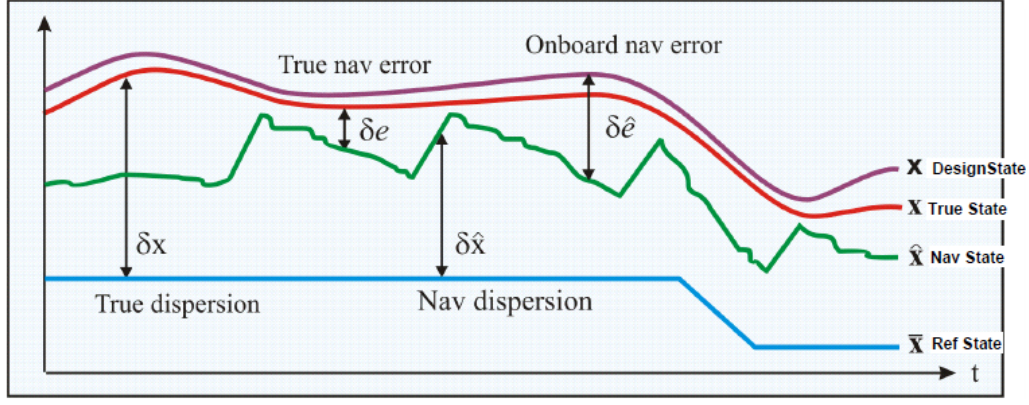


Figure 3: GN&C performance metric variables

difference between the true state $\mathbf{x}$ and the nominal state $\bar{\mathbf{x}}$. The covariance matrix of the true dispersions, $\mathbf{D}$, indicates how precisely a GN&C system can follow a desired trajectory.

$$\delta \mathbf{x} \triangleq \mathbf{x} - \bar{\mathbf{x}} \quad \mathbf{D} = E [\delta \mathbf{x} \delta \mathbf{x}^T] \quad (1)$$

The navigation dispersions $\delta \hat{\mathbf{x}}$ are defined as the difference between the navigation state $\hat{\mathbf{x}}$ and the nominal state $\bar{\mathbf{x}}$. The covariance of the navigation dispersions, $\hat{\mathbf{D}}$, reflects how precisely the onboard system thinks it can follow a prescribed reference trajectory.

$$\delta \hat{\mathbf{x}} \triangleq \hat{\mathbf{x}} - \mathbf{M}_x \bar{\mathbf{x}} \quad \hat{\mathbf{D}} = E [\delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T] \quad (2)$$

The matrix $\mathbf{M}_x$ is an $(\hat{n} \times n)$ matrix that maps the estimated state in terms of the true and nominal state. This matrix is necessary to account for the differences in vector length for $\hat{\mathbf{x}}$ and $\bar{\mathbf{x}}$.

The true navigation error $\delta \mathbf{e}$ is the difference between the true state and navigation state. It can also be calculated using the difference between the true dispersions and navigation dispersions. The covariance of the true navigation error, $\mathbf{P}$, quantifies how precisely the onboard navigation system can estimate the true state.

$$\delta \mathbf{e} \triangleq \mathbf{M}_x \mathbf{x} - \hat{\mathbf{x}} = \mathbf{M}_x \delta \mathbf{x} - \delta \hat{\mathbf{x}} \quad \mathbf{P} = E [\delta \mathbf{e} \delta \mathbf{e}^T] \quad (3)$$

The onboard navigation error $\delta \hat{\mathbf{e}}$ itself is never computed, but it is used to develop the onboard navigation filter equations. It is defined as the difference between the design state, $\mathbf{x}$, and the navigation state $\hat{\mathbf{x}}$. The covariance of the onboard navigation error, $\hat{\mathbf{P}}$, quantifies how precisely the onboard navigation system expects it can determine the actual state. The performance of the onboard navigation system is determined by comparing $\hat{\mathbf{P}}$ to the actual navigation performance $\mathbf{P}$.

$$\delta \hat{\mathbf{e}} \triangleq \mathbf{x} - \hat{\mathbf{x}} \quad \hat{\mathbf{P}} = E [\delta \hat{\mathbf{e}} \delta \hat{\mathbf{e}}^T] \quad (4)$$

If the *true* states and the *design* states are assumed to be the same (such as in this paper), then the true navigation covariance will equal the onboard navigation covariance.

The covariance matrices of the true dispersions, navigation dispersions, true navigation error, and the onboard navigation error are ultimately used to analyze and assess the performance of a proposed GN&C system. A common approach to obtain these performance metrics is to use a Monte Carlo simulation outlined in Figure 4, where the sample statistics of hundreds or thousands of runs, N , are used to numerically compute the desired covariance matrices.

$$\mathbf{D} = \frac{1}{N-1} \sum \delta \mathbf{x} \delta \mathbf{x}^T \quad \hat{\mathbf{D}} = \frac{1}{N-1} \sum \delta \hat{\mathbf{x}} \delta \hat{\mathbf{x}}^T \quad \mathbf{P} = \frac{1}{N-1} \sum \delta \mathbf{e} \delta \mathbf{e}^T \quad (5)$$

The onboard navigation error covariance $\hat{\mathbf{P}}$ is the navigation filter covariance for each run.

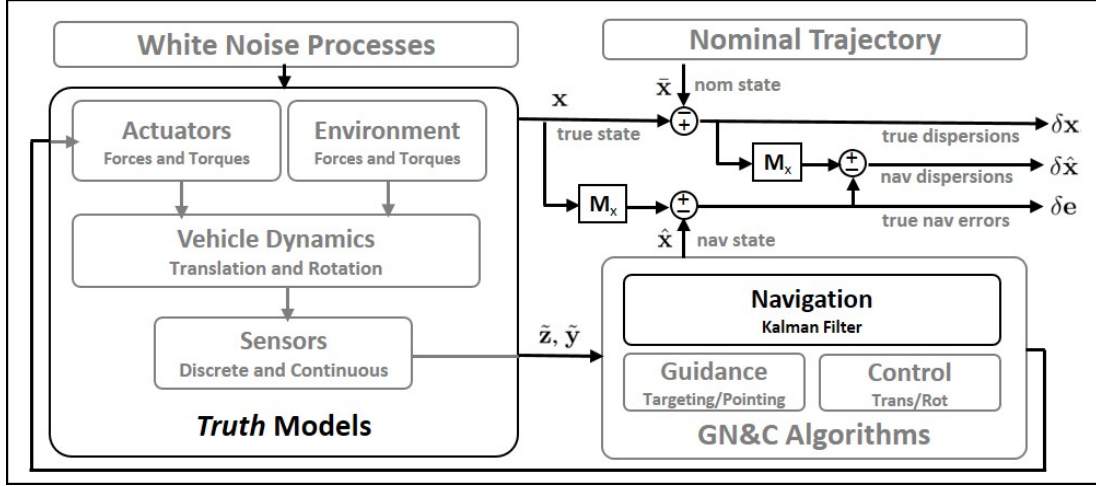


Figure 4: Extracting GN&C performance metrics using Monte Carlo techniques

This same statistical information can be obtained using linear covariance analysis techniques. Linear covariance analysis incorporates the non-linear system dynamics models and GN&C algorithms to generate a nominal reference trajectory $\bar{\mathbf{x}}$ which is then used to propagate, update, and correct an onboard navigation covariance matrix $\hat{\mathbf{P}}$ and an augmented state covariance matrix $\mathbf{C}$,

$$\mathbf{C} = E [\delta \mathbf{X} \delta \mathbf{X}^T] \quad (6)$$

where the augmented state $\delta \mathbf{X}^T = [\delta \mathbf{x}^T \ \delta \hat{\mathbf{x}}^T]$ consists of the true dispersions and the navigation dispersions. Pre- and post-multiplying the augmented state covariance matrix by the following mapping matrices, the covariance matrices of the trajectory dispersions, navigation dispersions, and the navigation error can be obtained with a single simulation run.

$$\begin{aligned} \mathbf{D} &= [\mathbf{I}_{n \times n}, \mathbf{0}_{n \times \hat{n}}] \mathbf{C} [\mathbf{I}_{n \times n}, \mathbf{0}_{n \times \hat{n}}]^T \\ \hat{\mathbf{D}} &= [\mathbf{0}_{\hat{n} \times n}, \mathbf{I}_{\hat{n} \times \hat{n}}] \mathbf{C} [\mathbf{0}_{\hat{n} \times n}, \mathbf{I}_{\hat{n} \times \hat{n}}]^T \\ \mathbf{P} &= [\mathbf{I}_{\hat{n} \times n}, -\mathbf{I}_{\hat{n} \times \hat{n}}] \mathbf{C} [\mathbf{I}_{\hat{n} \times n}, -\mathbf{I}_{\hat{n} \times \hat{n}}]^T \end{aligned} \quad (7)$$

Linear Covariance Analysis

The linear covariance (LinCov) analysis equations are used to propagate, update, and correct the augmented state covariance matrix and the onboard navigation covariance matrix, and they will be summarized here. For a deeper analysis of the construction, development, and application of the LinCov simulation, see the following references.^{2,3,7-10}

As outlined by Goulet,⁴ the discrete-time propagation equations for the onboard navigation covariance matrix $\hat{\mathbf{P}}$ and the augmented state covariance matrix $\mathbf{P}_A$ are summarized using Equation 8 and Equation 9,

$$\hat{\mathbf{P}}(t_{k+1}) = \hat{\Phi}(t_{k+1}, t_k) \hat{\mathbf{P}}(t_k) \hat{\Phi}^T(t_{k+1}, t_k) + \hat{W} \hat{S}_\omega \hat{W}^T \quad (8)$$

$$\mathbf{P}_A(t_{k+1}) = \Phi(t_{k+1}, t_k) \mathbf{P}_A(t_k) \Phi^T(t_{k+1}, t_k) + W S_\omega W^T \quad (9)$$

where Φ and $\hat{\Phi}$ are the augmented and onboard state transition matrices for the linearized perturbation dynamics about the reference trajectory. The weighting matrices, W and $\hat{W}$, incorporate the environmental and navigation noise described by S_ω and $\hat{S}_\omega$ into $\mathbf{P}_A$ and $\hat{\mathbf{P}}$.

Measurement updates are incorporated into the augmented and navigation state covariance matrices using the measurement update equations for a given time t_i ,

$$\hat{P}^+(t_i) = [\hat{I} - \hat{K}^j(t_i)\hat{H}^j]\hat{P}^-(t_i)[\hat{I} - \hat{K}^j(t_i)\hat{H}^j]^T + \hat{K}^j(t_i)\hat{R}^j(t_i)\hat{K}^j(t_i)^T \quad (10)$$

$$P_A^+(t_i) = MP_A^-(t_i)M^T + NR^j(t_i)N^T \quad (11)$$

where the matrices M and N weigh the effects of measurements and their inherent noise into the navigation state dispersion, and where the superscript 'j' denotes the jth measurement type.

The Kalman gain matrix, $\hat{K}$, adjusts the corrections applied to the predicted state based on the measurement data, and is written as

$$\hat{K}^j(t_i) = \hat{P}(t_i)(\hat{H}^j)^T[\hat{H}^j\hat{P}^-(t_i)(\hat{H}^j)^T + \hat{R}^j(t_i)]^{-1} \quad (12)$$

where the matrices $\hat{H}$ and $\hat{R}$ describe the measurement sensitivity and measurement noise matrices respectively.

The correction equations for P_A and $\hat{P}$ at a maneuver time t_m are described by

$$\hat{P}^+(t_m) = \hat{P}^-(t_m) + \hat{V}\hat{S}_\nu(t_m)\hat{V}^T \quad (13)$$

$$P_A^+(t_m) = DP_A^-(t_m)D^T + VS_\nu(t_m)V^T \quad (14)$$

where the matrix D holds the control partials correlated with a linearized two-impulse targeting algorithm. The matrices $\hat{V}$ and V weigh the effects of any actuator noise, described by the matrices $\hat{S}_\nu$ and S_ν , into P_A and $\hat{P}$.

Generalized Reference Targeting

The Generalized Reference Targeting (GRT) algorithm is a method used to mimic many of the properties of the commonly used two-level targeter (TLT). However, the TLT's complexity and iterative nature requires a substantial amount of time, making it difficult to incorporate into trajectory analysis when speed is essential. However, due to its linearization, GRT is exceptionally fast and easy to implement into a linear covariance (LinCov) analysis simulation. GRT can handle single or multiple impulsive maneuvers, and can account for multiple sets of constraints at multiple fixed or variable times. The original form of the GRT equation for all derivations can be seen in Equation 15. More an in-depth analysis of GRT, its derivation, and its application to a variety of example trajectories, please see the following references.¹

$$\mathcal{M}\delta u = \mathcal{N}\delta x_0 \quad (15)$$

where

$$M = G \{ M_{xfu} + \dot{x}_f^* P_{\delta t_f, \delta u} \} \quad (16)$$

$$N = -GM_{xfx_0} \quad (17)$$

Well-Defined, Over-Determined, and Under-Determined Problems When targeting a single impulsive maneuver at a single fixed or variable final time, the targeting constraints can be described as well-defined, over-determined, and under-determined. In these scenarios, the relationship between the n number of controls and the m number of linear equations for parameters being targeted describes which problem type is being executed. In this paper, the n number of controls is 3, which includes the engine ignition time, the direction and rotation rate of the thrust vector, and the thrust cutoff time.

When $m = n$, the problem is well-defined, which will lend itself to one solution which can be written simply as

$$\delta u = \mathcal{M}^{-1}\mathcal{N}\delta x_0 \quad (18)$$

When $m > n$, this represents an over-determined problem, and a solution for the minimum normalized constraint error is desired

$$\delta u = (\mathcal{M}^T W_\psi \mathcal{M})^{-1} \mathcal{M}^T W_\psi \mathcal{N} \delta x_0 \quad (19)$$

where the constraints are weighted by a diagonal weighting matrix W_ψ .

When $m < n$, the problem is under-determined, which potentially offers an infinite number of solutions. In this scenario, a solution for the minimum normalized control is desired

$$\delta u = W_c \mathcal{M}^T (\mathcal{M} W_c \mathcal{M}^T)^{-1} \mathcal{N} \delta x_0 \quad (20)$$

where the controls are weighted by a diagonal weighting matrix W_c .

The Targeting Constraints Matrix

The targeting matrix, G , is defined as the derivative of the targeting constraints with respect to the spacecraft's state vector in a specified reference frame, where the spacecraft's state vector is described by its position and velocity vectors.

$$G = \begin{bmatrix} \frac{\partial m_1}{\partial r_x} & \frac{\partial m_1}{\partial r_y} & \frac{\partial m_1}{\partial r_z} & \frac{\partial m_1}{\partial v_x} & \frac{\partial m_1}{\partial v_y} & \frac{\partial m_1}{\partial v_z} \\ \frac{\partial m_2}{\partial r_x} & \frac{\partial m_2}{\partial r_y} & \frac{\partial m_2}{\partial r_z} & \frac{\partial m_2}{\partial v_x} & \frac{\partial m_2}{\partial v_y} & \frac{\partial m_2}{\partial v_z} \\ \frac{\partial m_3}{\partial r_x} & \frac{\partial m_3}{\partial r_y} & \frac{\partial m_3}{\partial r_z} & \frac{\partial m_3}{\partial v_x} & \frac{\partial m_3}{\partial v_y} & \frac{\partial m_3}{\partial v_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial m_n}{\partial r_x} & \frac{\partial m_n}{\partial r_y} & \frac{\partial m_n}{\partial r_z} & \frac{\partial m_n}{\partial v_x} & \frac{\partial m_n}{\partial v_y} & \frac{\partial m_n}{\partial v_z} \end{bmatrix}$$

The size of the G matrix is defined by the number of targeting constraints, m , and the size of the spacecraft's state vector, which will always contain 6 elements. The number of rows within the G matrix corresponds to the m constraints being targeted and the columns of the G matrix correspond to the spacecraft's 6 state vector elements, making G an $m \times 6$ matrix. Therefore, when applying Generalized Reference Targeting, a 1×6 or 2×6 G matrix describes an under-determined problem, a 3×6 G matrix describes a well-defined problem, and a 4×6 or larger G matrix describes an over-determined problem.

The G matrix offers a handful of advantages as a targeting method. Compared to other historically utilized targeting algorithms, such as CW targeting, GRT and the targeting constraint matrix G allow the user to readily see the trajectory's sensitivity to targeting parameters. Furthermore, the G matrix can target any type of targeting constraint used to define an orbit or state beyond just position. Finally, the G matrix is not limited to "ideal" or well-defined targeting scenarios. Therefore, GRT's targeting constraints matrix provides flexibility and sensitivity that is not present in more traditional targeting algorithms like CW targeting.

Within this paper, three separate G matrices will be used to test and optimize the spacecraft's lunar trajectory. These 3 G matrices will correspond with the 3 "Scenarios" when generating the trajectory's mission maps. For scenario 1, the G matrix is defined by just targeting the position vector. This constitutes a well-defined targeting problem, and produces the following G matrix at any point along the trajectory:

$$G_1 = \begin{bmatrix} \frac{\partial r_x}{\partial r_x} & \frac{\partial r_x}{\partial r_y} & \frac{\partial r_x}{\partial r_z} & \frac{\partial r_x}{\partial v_x} & \frac{\partial r_x}{\partial v_y} & \frac{\partial r_x}{\partial v_z} \\ \frac{\partial r_y}{\partial r_x} & \frac{\partial r_y}{\partial r_y} & \frac{\partial r_y}{\partial r_z} & \frac{\partial r_y}{\partial v_x} & \frac{\partial r_y}{\partial v_y} & \frac{\partial r_y}{\partial v_z} \\ \frac{\partial r_z}{\partial r_x} & \frac{\partial r_z}{\partial r_y} & \frac{\partial r_z}{\partial r_z} & \frac{\partial r_z}{\partial v_x} & \frac{\partial r_z}{\partial v_y} & \frac{\partial r_z}{\partial v_z} \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & Z_{3 \times 3} \end{bmatrix}$$

For scenario 2, the G matrix is defined by targeting both the position vector and the velocity vector. This constitutes an over-defined targeting problem, and produces the following G matrix at any point along the

trajectory:

$$G_2 = \begin{bmatrix} \frac{\partial r_x}{\partial r_x} & \frac{\partial r_x}{\partial r_y} & \frac{\partial r_x}{\partial r_z} & \frac{\partial r_x}{\partial v_x} & \frac{\partial r_x}{\partial v_y} & \frac{\partial r_x}{\partial v_z} \\ \frac{\partial r_y}{\partial r_x} & \frac{\partial r_y}{\partial r_y} & \frac{\partial r_y}{\partial r_z} & \frac{\partial r_y}{\partial v_x} & \frac{\partial r_y}{\partial v_y} & \frac{\partial r_y}{\partial v_z} \\ \frac{\partial r_z}{\partial r_x} & \frac{\partial r_z}{\partial r_y} & \frac{\partial r_z}{\partial r_z} & \frac{\partial r_z}{\partial v_x} & \frac{\partial r_z}{\partial v_y} & \frac{\partial r_z}{\partial v_z} \\ \frac{\partial v_x}{\partial r_x} & \frac{\partial v_x}{\partial r_y} & \frac{\partial v_x}{\partial r_z} & \frac{\partial v_x}{\partial v_x} & \frac{\partial v_x}{\partial v_y} & \frac{\partial v_x}{\partial v_z} \\ \frac{\partial v_y}{\partial r_x} & \frac{\partial v_y}{\partial r_y} & \frac{\partial v_y}{\partial r_z} & \frac{\partial v_y}{\partial v_x} & \frac{\partial v_y}{\partial v_y} & \frac{\partial v_y}{\partial v_z} \\ \frac{\partial v_z}{\partial r_x} & \frac{\partial v_z}{\partial r_y} & \frac{\partial v_z}{\partial r_z} & \frac{\partial v_z}{\partial v_x} & \frac{\partial v_z}{\partial v_y} & \frac{\partial v_z}{\partial v_z} \end{bmatrix} = \begin{bmatrix} I_{6 \times 6} \end{bmatrix}$$

For scenario 3, the G matrix is defined by targeting the spacecraft's flight path angle, altitude, and velocity magnitude. This constitutes a well-defined targeting problem, and produces the following G matrix:

$$G_3 = \begin{bmatrix} \frac{\partial \gamma}{\partial r_x} & \frac{\partial \gamma}{\partial r_y} & \frac{\partial \gamma}{\partial r_z} & \frac{\partial \gamma}{\partial v_x} & \frac{\partial \gamma}{\partial v_y} & \frac{\partial \gamma}{\partial v_z} \\ \frac{\partial h}{\partial r_x} & \frac{\partial h}{\partial r_y} & \frac{\partial h}{\partial r_z} & \frac{\partial h}{\partial v_x} & \frac{\partial h}{\partial v_y} & \frac{\partial h}{\partial v_z} \\ \frac{\partial |v|}{\partial r_x} & \frac{\partial |v|}{\partial r_y} & \frac{\partial |v|}{\partial r_z} & \frac{\partial |v|}{\partial v_x} & \frac{\partial |v|}{\partial v_y} & \frac{\partial |v|}{\partial v_z} \end{bmatrix} = \begin{bmatrix} \frac{v_{rel}}{|r||v|} - \frac{S_g r_{rel}}{|r|^2} & \frac{r_{rel}}{|r||v|} - \frac{S_g v_{rel}}{|v|^2} \\ \frac{r_{rel}}{|r|} & Z_{1 \times 3} \\ Z_{1 \times 3} & \frac{v_{rel}}{|v|} \end{bmatrix}, \text{ where } S_g = \frac{r_{rel} \cdot v_{rel}}{|r||v|}$$

The Ground Tracking Model

In order to accurately capture the spacecraft's environment, the spacecraft's initial dispersions, navigation errors, process noise, maneuver execution errors, and ground station availability must be accounted for along the trajectory. It is important to note that the values used below approximate industry standard values as opposed to flight-specific data. It is assumed that the spacecraft experiences initial position dispersions of $(\frac{5}{3})^2$ km per axis and initial velocity dispersions of $(\frac{0.2}{3})^2$ m/s per axis. For navigation errors, it is assumed that the spacecraft initially experiences position errors of $(\frac{2.5}{3})^2$ km per axis and velocity errors of $(\frac{0.1}{3})^2$ m/s per axis. For process noise, it is assumed that the spacecraft experiences $1.5 \times 10^{-8} \text{ m}^2/\text{s}^3$ of translation process noise dispersion per axis and $1.5 \times 10^{-8} \text{ m}^2/\text{s}^3$ of translation process noise error per axis. For maneuver execution error, it is assumed that the spacecraft experiences $(\frac{1}{3})^2$ cm/s of maneuver noise dispersion per axis and $(\frac{1}{3})^2$ cm/s of maneuver noise error per axis.

Within the ground tracking model, 3 ground stations are simulated to provide realistic ground tracking updates and windows. The three ground stations used in this simulation include DS65 in Madrid, Spain, DS45 in Canberra, Australia, and DS15 in Goldstone, California. It is assumed that all three of these ground stations can provide the spacecraft with range and range rate/doppler measurements. Furthermore, the data collected from the ground stations' range and range-rate measurements is used to build the windows for the spacecraft's batch updates. The windows for batch updates exist from an hour before each burn up to the burn, and from a half hour after the burn until an hour after the burn. It is also assumed that the three ground stations can provide a valid measurement when the spacecraft is greater than 5 degrees above the horizon. These ground stations are assumed to have no minimum range and no maximum range, and can provide measurements every 5 minutes given that the spacecraft is within the station's line-of-sight. These measurements have a standard deviation of 53 meters for range and 0.52 cm/s for range-rate. For the outbound cislunar trajectory used in this paper, the ground station logic map can be seen below in Figure 5.

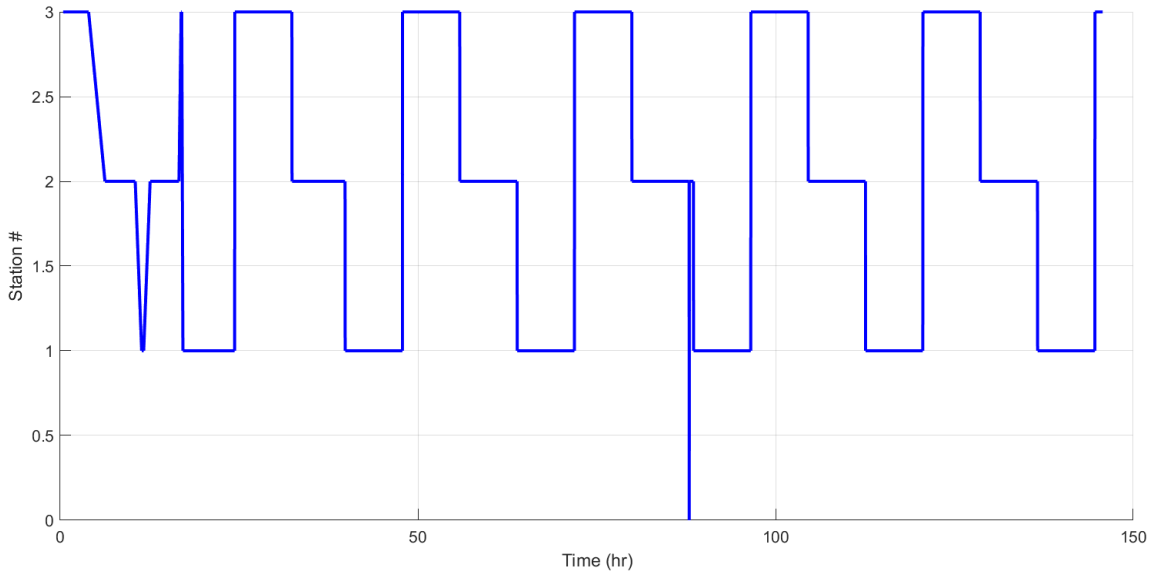


Figure 5: Ground Station Logic Map for Nominal Cislunar Trajectory

ANALYSIS RESULTS

Cislunar Trajectory Scenarios, Runs, and Cases

Nominally, the outbound cislunar trajectory followed in this paper begins 20 minutes before the spacecraft's apogee raise maneuver (ARM). After the ARM burn is executed, the ARM adjunct (ARM-ADJ) is then executed followed by trajectory correction maneuver 0 (TCM-0). Next, the trans-lunar insertion burn (TLI) is executed, which is also followed by the TLI adjunct burn (TLI-ADJ). After the TLI-ADJ burn, the spacecraft executes 6 TCMs while drifting out to the Moon before performing the lunar orbit insertion (LOI) and LOI adjunct (LOI-ADJ) burns. The simulation ends before a de-orbit insertion (DOI) burn is executed. A generalized outline of this trajectory can be seen in Figure 6.

During the outbound drift towards the Moon before LOI, spacecraft executes 6 TCMs. Because each TCM is nominally zero regardless of placement within the trajectory, the placement of TCMs along the trajectory is essentially arbitrary. In this outbound cislunar trajectory simulation, TCM-1 was chosen to be executed 12 hours after TLI, TCM-2 is executed 20 hours after TLI, TCM-3 is executed 28 hours after TLI, TCM-4 is executed 36 hours after TLI, TCM-5 is executed 24 hours before LOI, and TCM-6 is executed 6 hours before LOI.

Furthermore, because TCM ΔV usage is nominally zero, the number of TCMs used within an outbound cislunar trajectory is also essentially arbitrary. However, due to the existence of environmental dispersions, errors, and noise, the spacecraft's estimated 3σ ΔV usage and position dispersions at LOI can vary based on the number and placement of TCMs along the outbound trajectory.

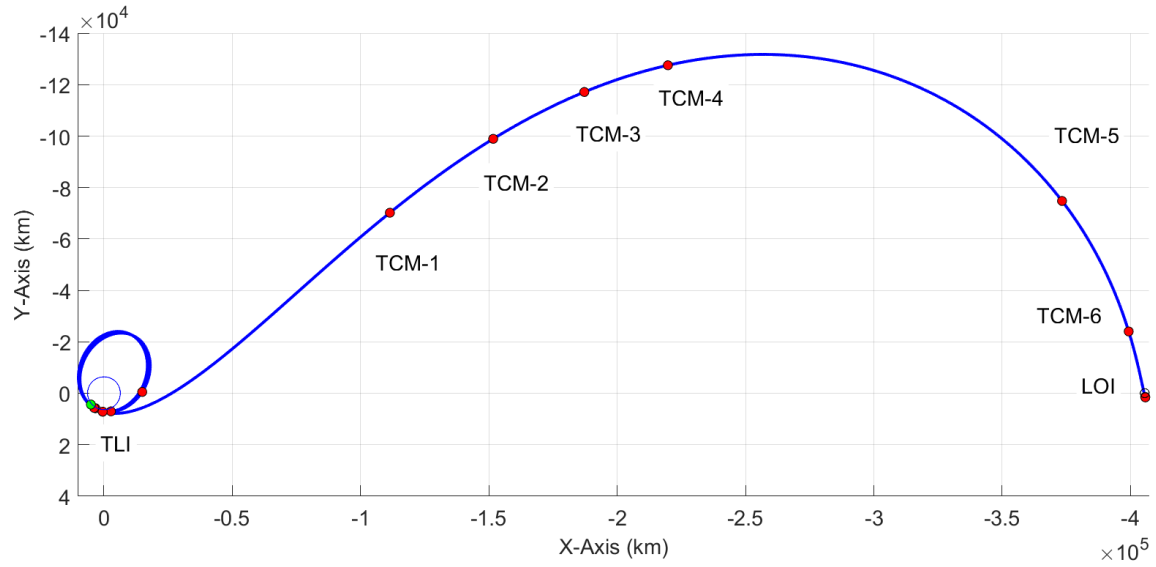


Figure 6: Simulated Trajectory into Low Lunar Orbit

This offers an interesting avenue for optimization. By varying the simulation’s placement of TCMs and targeting constraints at LOI, the spacecraft’s 3σ ΔV usage and position dispersions at LOI can be minimized. In order to effectively describe the placement of TCMs and the targeting constraints used in any given optimization problem within this paper, a key has been created below in Table 1. In this key, *Scenarios* describe the targeting constraint matrix used during the trajectory, and *Cases* describe the TCMs being altered from their nominal placement.

Table 1: Key for Scenarios and Cases

Simulation Type	Description
Scenario 1	The constraint matrix, G , is targeting the position vector at LOI
Scenario 2	The constraint matrix, G , is targeting the position vector and the velocity vector at LOI
Scenario 3	The constraint matrix, G , is targeting the spacecraft’s flight path angle, altitude, and velocity magnitude at LOI
Case 1	TCMs 2 and 5 are varied from nominal
Case 2	TCMs 2 and 6 are varied from nominal
Case 3	TCMs 3 and 5 are varied from nominal
Case 4	TCMs 3 and 6 are varied from nominal
Case 5	TCMs 4 and 5 are varied from nominal
Case 6	TCMs 4 and 6 are varied from nominal

A Visualization of Optimization: Mission Maps

Traditionally, other cislunar trajectory problems employ some form of optimization tool in order to find nominal solutions within a complex trade space. Historically, these have included particle swarm and genetic algorithm tools, which iterate through solution spaces by simulating collective behaviors or evolutionary processes to optimize complex, multidimensional problems. However, these tools can take hours or days to produce meaningful results, preventing rapid real-time analysis.

To address this limitation, “mission maps” have been developed, enabling users to quickly and visually optimize problems. Mission maps closely resemble topological maps, and are generated by performing

several runs within a parallel processing tool while moving 2 specified burns around by a predetermined time value. For any given set of targeting constraints or burn alterations, a set of mission maps takes only minutes to generate. This ultimately allows the user to work as the optimizer by visually identifying high or low points on the mission map, corresponding to the maximization or minimization of values like $3\sigma \Delta V$ usage and position dispersions at LOI.

In this paper, mission maps are created by moving around the nominally zero ΔV TCMs between TLI and LOI. Before a mission map is generated, the two TCMs being shifted and the targeting matrix (G) are specified. Each TCM is altered ± 2 hours from its original location along the nominal trajectory, and each TCM is shifted through 5 different placements to create a trade space of 25 runs on the mission map. Once all 25 runs have been executed, mission maps are generated to display $3\sigma \Delta V$ usage and position dispersions at LOI, where minimum values are desired for both scenarios. An example mission map for Scenario 1 Case 1 $3\sigma \Delta V$ usage is shown below in Figure 7.

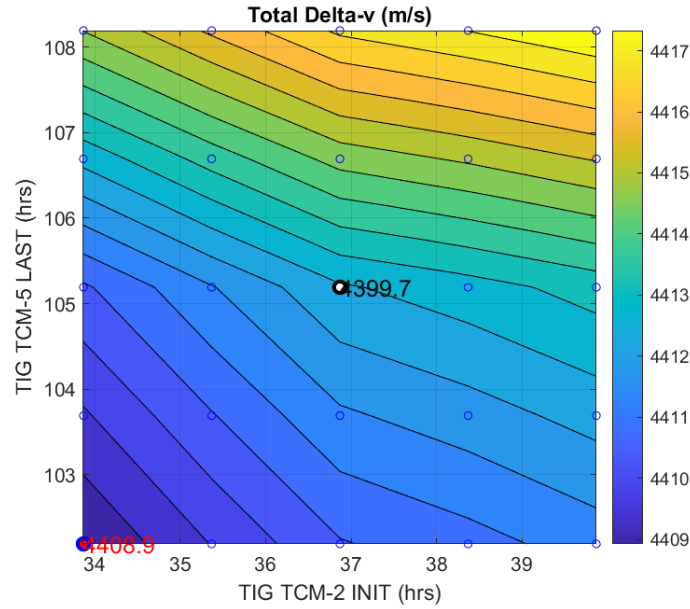


Figure 7: $3\sigma \Delta V$ Usage Mission Map for Scenario 1 Case 1

In the next pages, Figures 8-15 will include the mission summaries and mission maps for all combinations of Scenarios and Cases. Each new mission summary will correspond with a particular Scenario, followed by the mission maps for both $3\sigma \Delta V$ usage and position dispersions at LOI in their own figures.

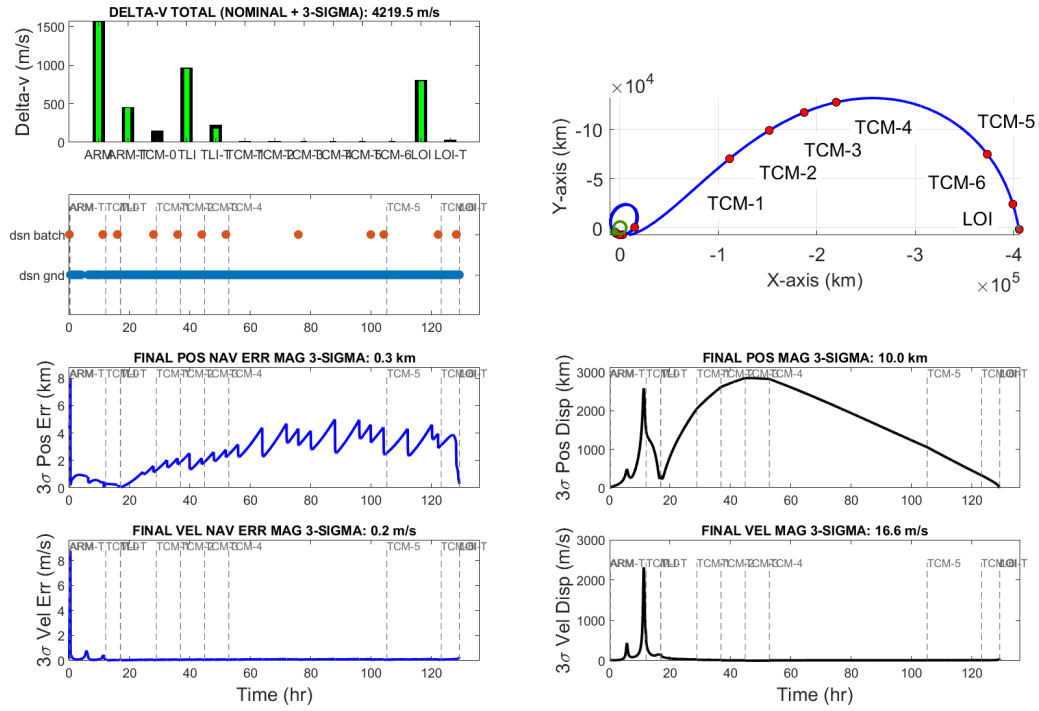


Figure 8: Scenario 1 Overview

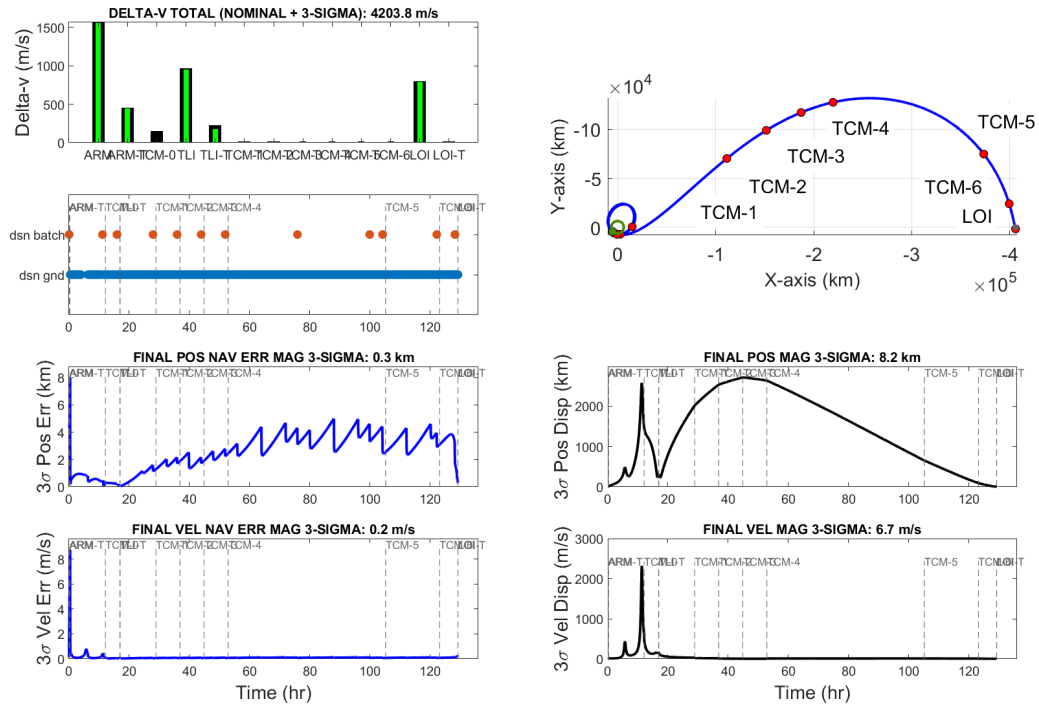


Figure 9: Scenario 2 Overview

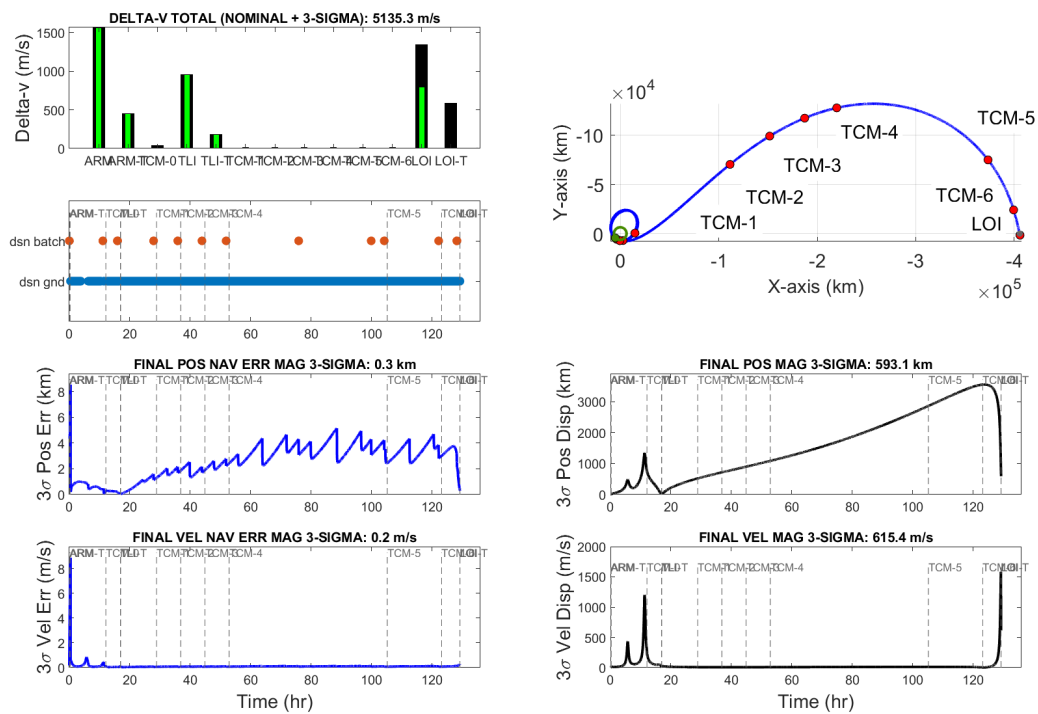


Figure 10: Scenario 3 Overview

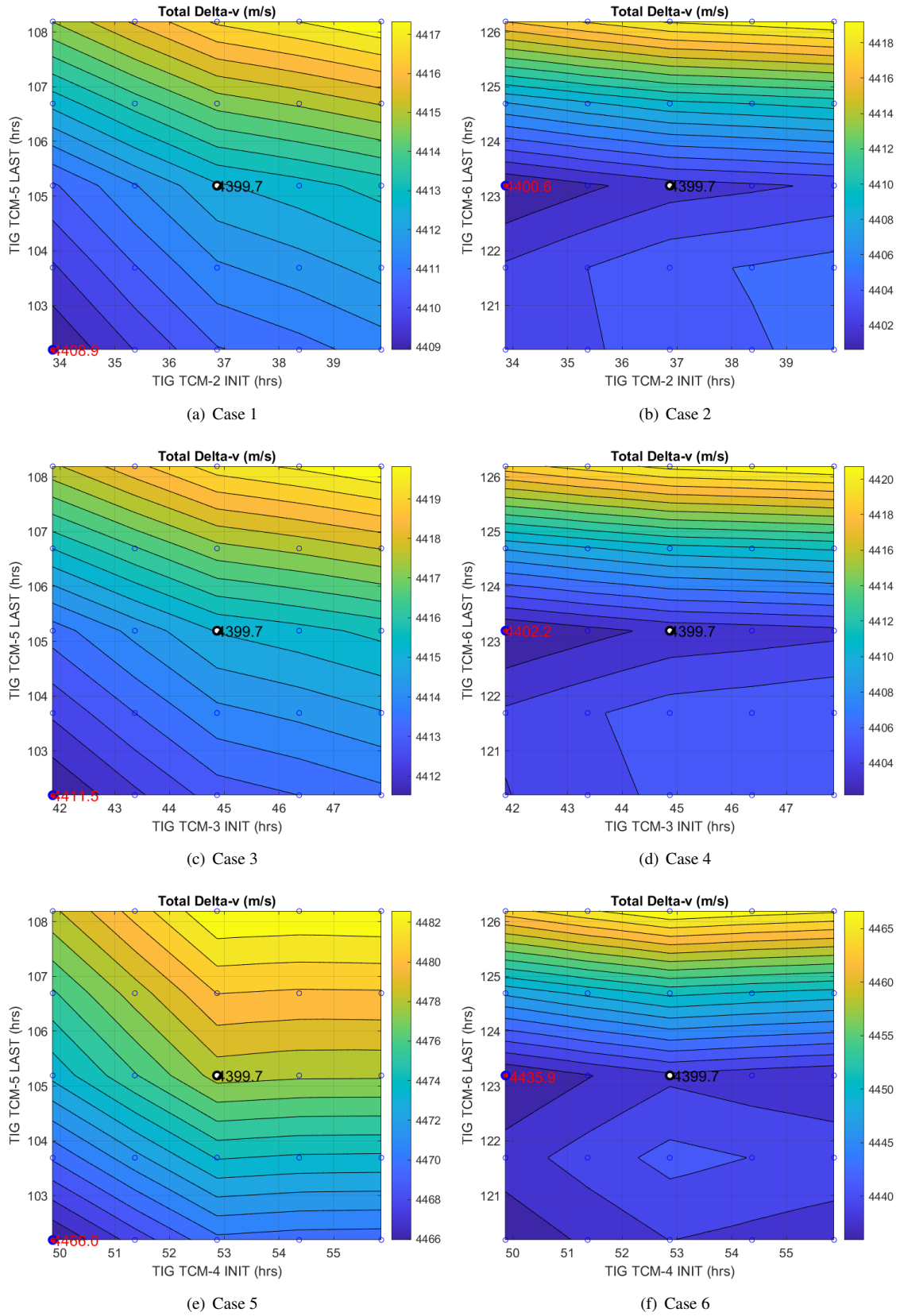
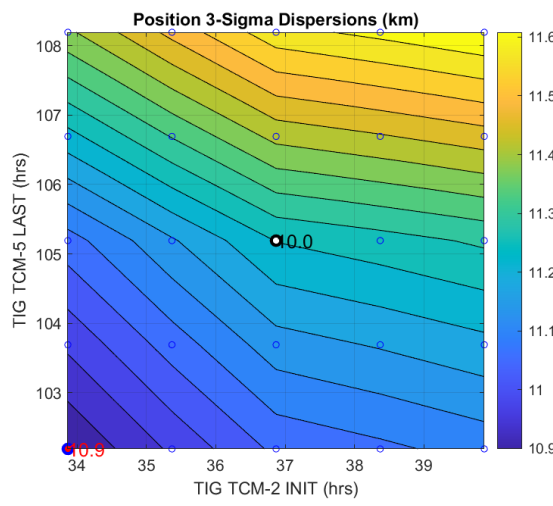
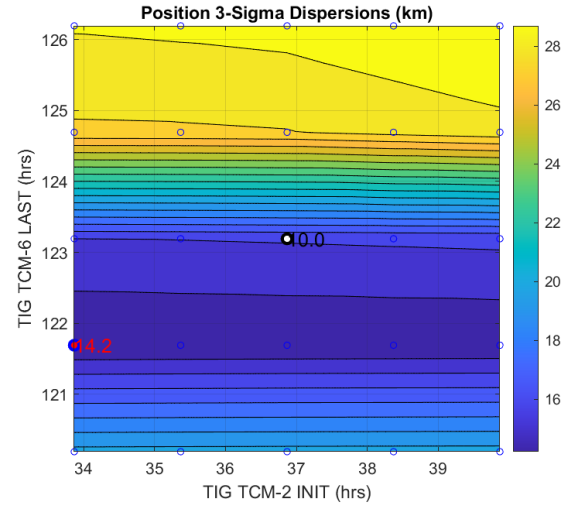


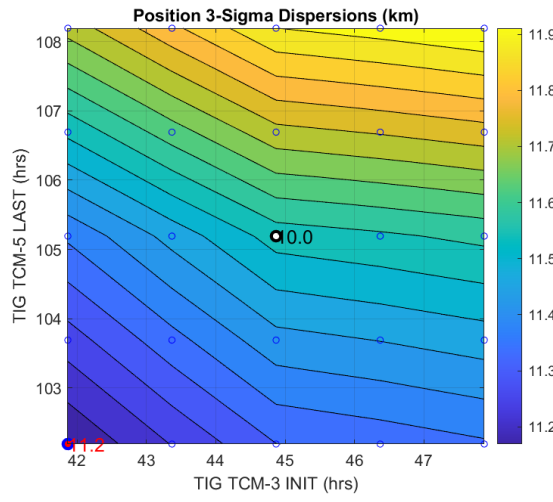
Figure 11: Summary and Mission Maps for Scenario 1 (Delta-v)



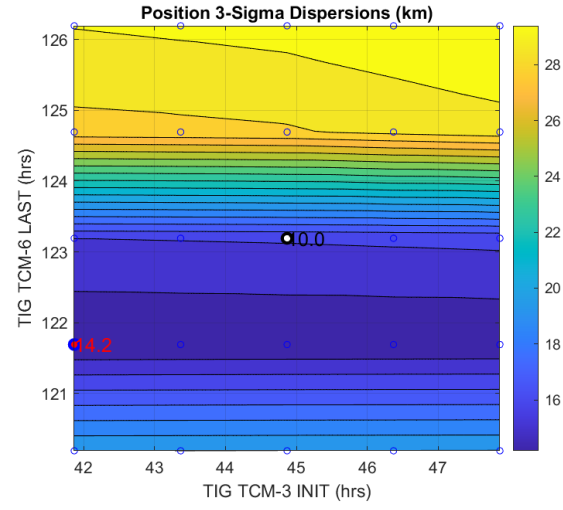
(a) Case 1



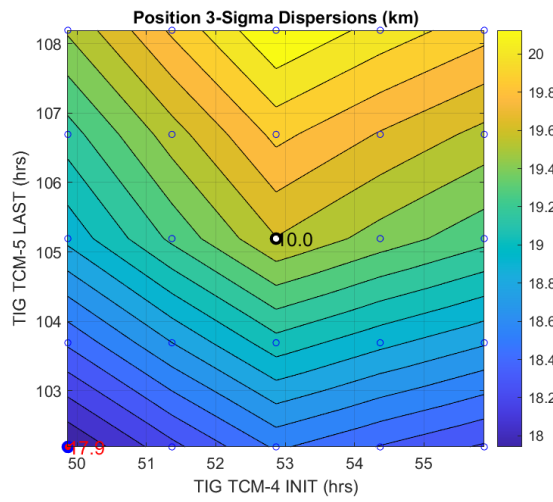
(b) Case 2



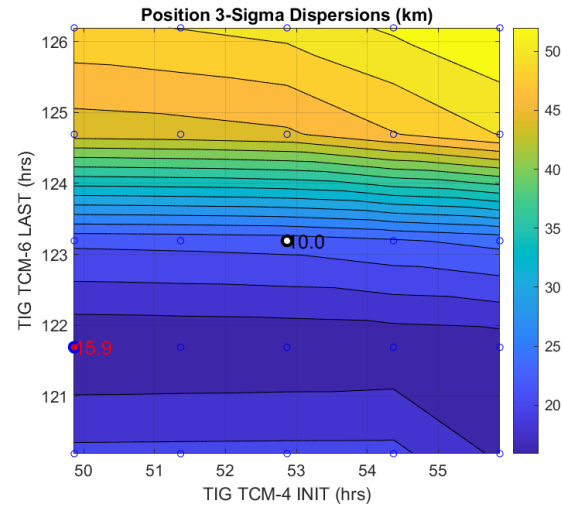
(c) Case 3



(d) Case 4

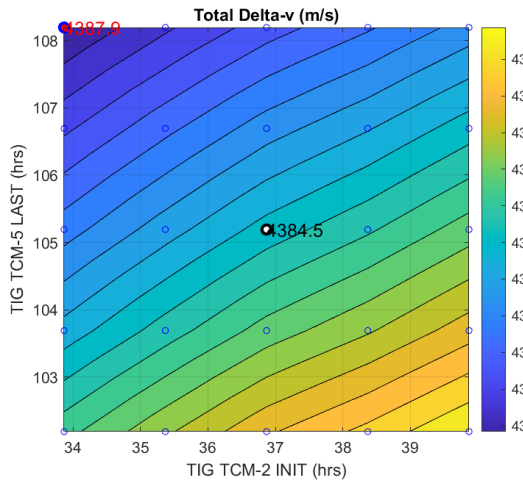


(e) Case 5

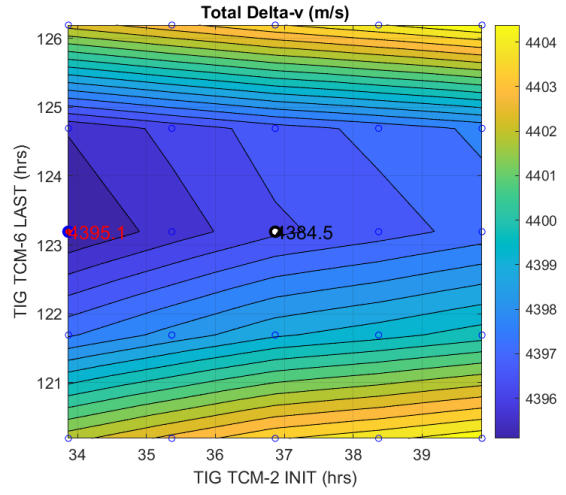


(f) Case 6

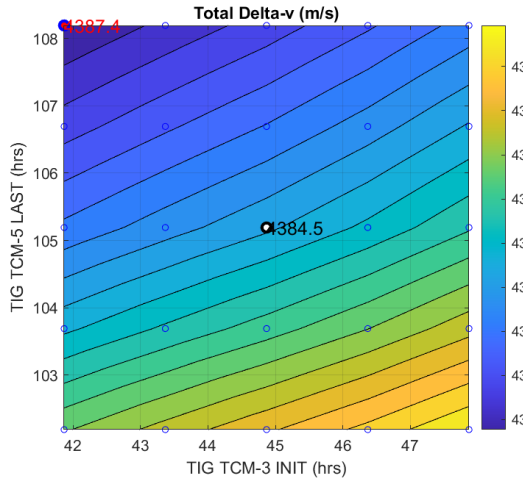
Figure 11: Summary and Mission Maps for Scenario 1 (Position)



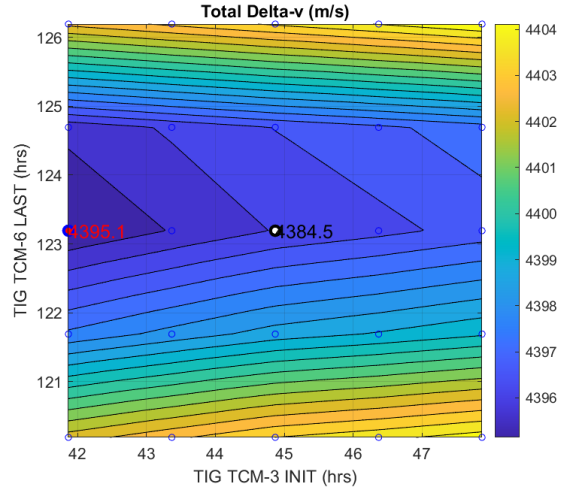
(g) Case 1



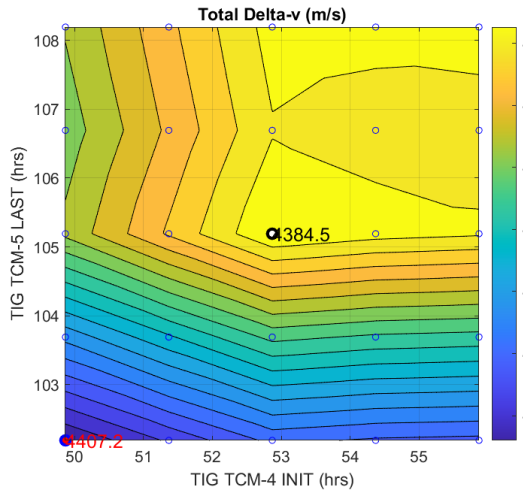
(h) Case 2



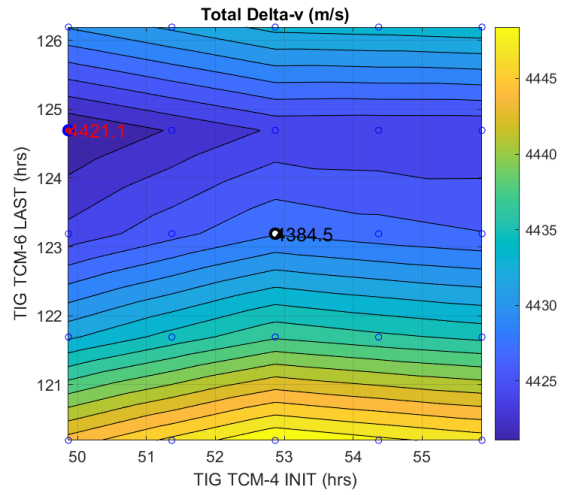
(i) Case 3



(j) Case 4



(k) Case 5



(l) Case 6

Figure 12: Summary and Mission Maps for Scenario 2 (Delta-v)

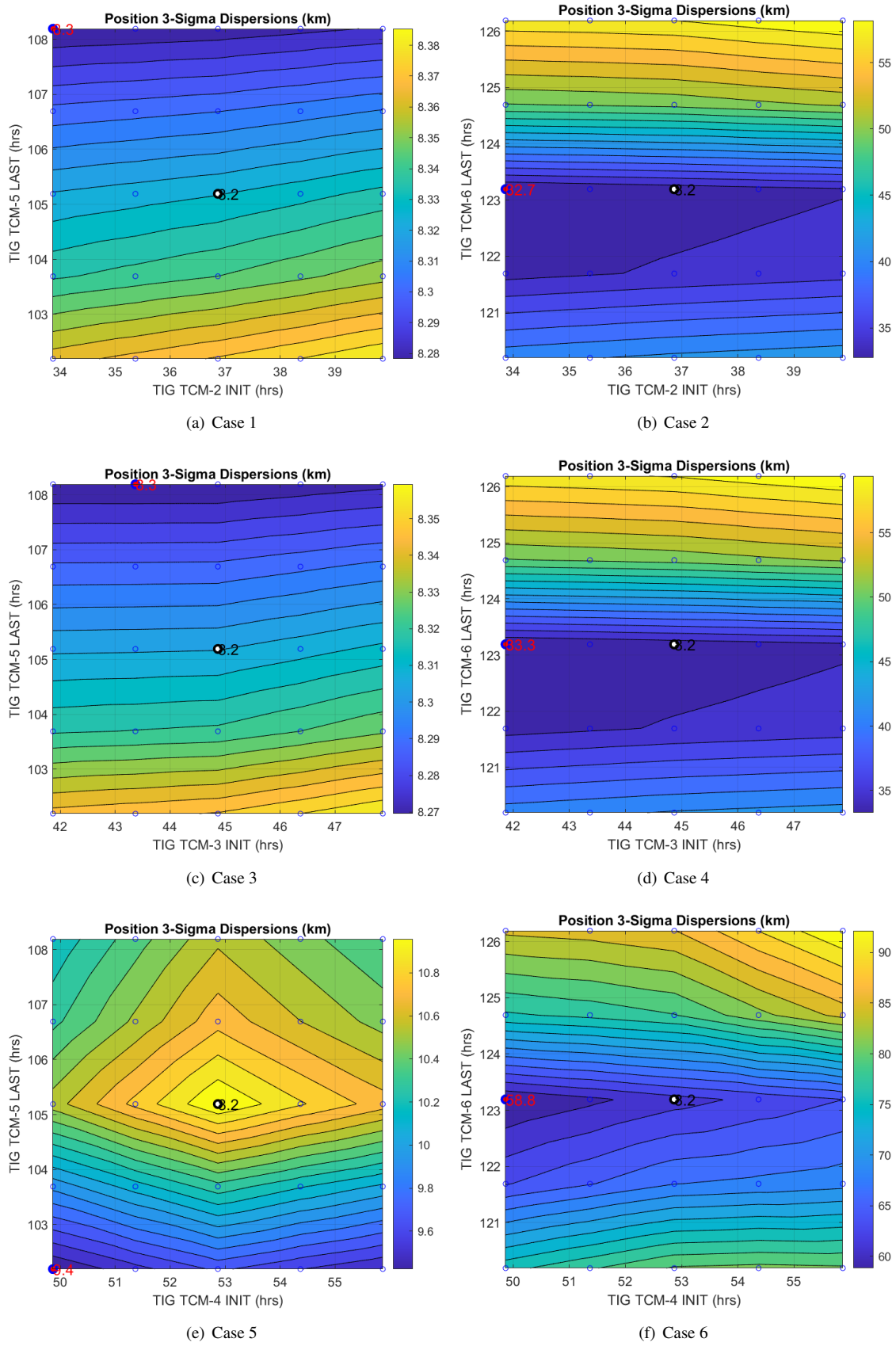


Figure 13: Summary and Mission Maps for Scenario 2 (Position)

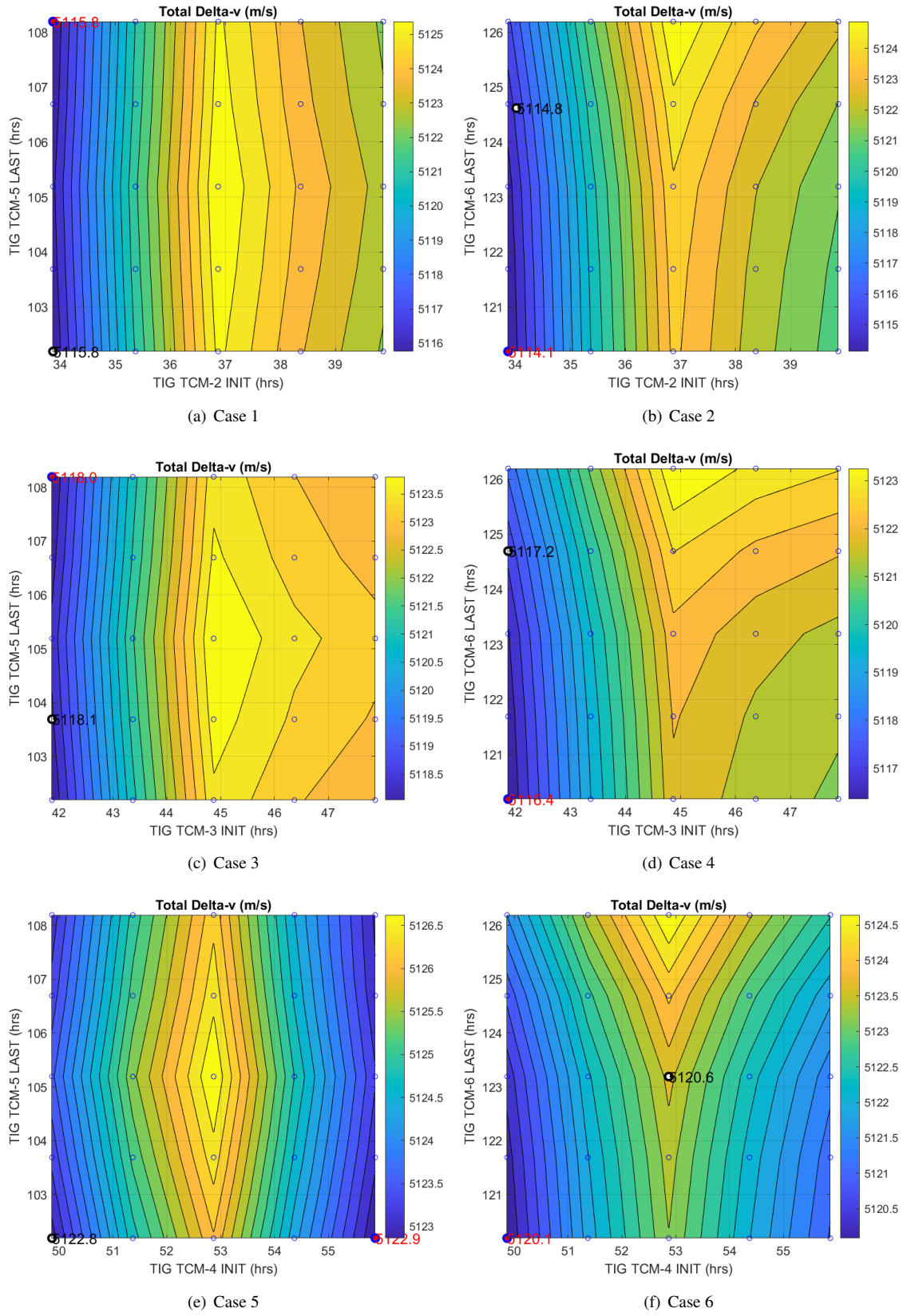


Figure 14: Summary and Mission Maps for Scenario 3 (Delta-v)

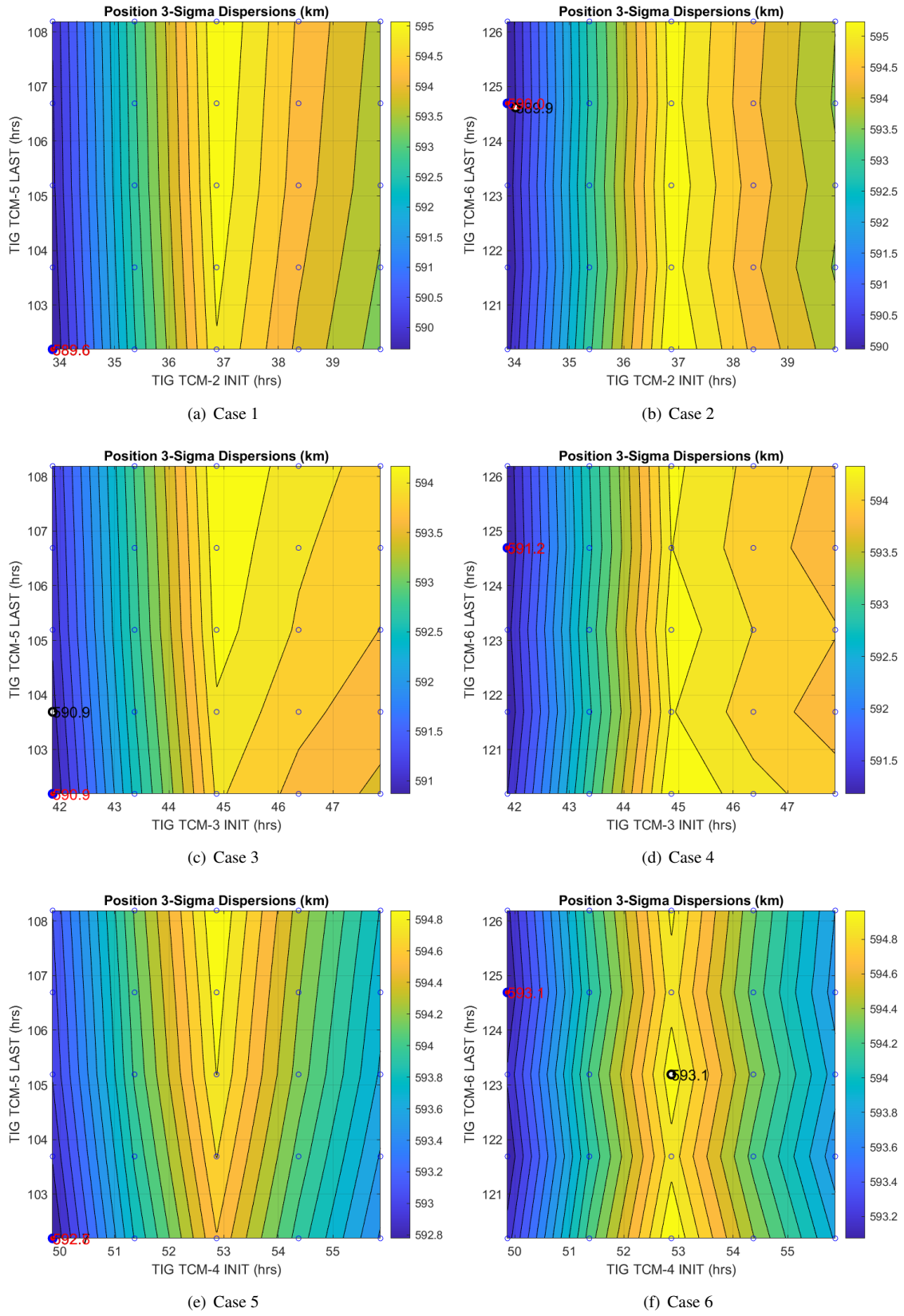


Figure 15: Summary and Mission Maps for Scenario 3 (Position)

Mission Map Analysis & Overview

Using the mission maps shown in Figures 11-15, rapid visual analysis and optimization of the outbound cislunar trajectory can be conducted. For each scenario, mission maps can inform the placement of TCMs by analyzing their effect on $3\sigma \Delta V$ usage and position dispersions at LOI. Furthermore, the “weight” or impact created by moving a TCM around can be inferred by the lines within the mission maps. If a section of mission maps contains mostly *horizontal* lines, it suggests that TCM-5 or TCM-6 dominate TCM-2, 3, or 4 because change is created only when varying TCM-5/6. If a section of mission maps contains mostly *vertical* lines, it suggests that TCM-2, 3, or 4 dominate TCM-5 or TCM-6 because change is created only when varying TCM-2, 3, or 4. Finally, it is important to note that shades of *blue* represent minimum values and shades of *yellow* represent maximum values on the mission maps. Because we aim to minimize $3\sigma \Delta V$ usage and position dispersions at LOI, we are interested in the **blue** portion of each mission map.

Scenario 1 Analysis Scenario 1 (targeting the position vector) encourages an earlier placement for all tested TCMs when focusing on the minimization of $3\sigma \Delta V$ usage at LOI. For position dispersion at LOI, the Scenario 1 mission maps demonstrate that TCMs 2-4 have very little impact compared to TCMs 5 and 6. This can be inferred from the very horizontal lines in Figure ???. Furthermore, we can see that placing TCM-5 2 hours earlier in the trajectory and retaining TCM-6 at its nominal location producing optimal results. As a whole, Scenario 1 heavily encourages the earlier placement of TCM-5 when compared with other targeting constraints.

Scenario 2 Analysis Scenario 2 (targeting both the position vector and the velocity vector) deviates from the suggestions made by Scenario 1, demonstrating how changing targeting constraints impacts fuel usage and position dispersions at LOI. Scenario 2 shows that TCMs 2 and 3 should be placed earlier in the trajectory to minimize $3\sigma \Delta V$ usage and position dispersions. TCM-4 can be moved forwards or backwards to produce better fuel usage or position dispersion results, but the horizontal lines present in its mission maps demonstrate the TCM-4 has little effect on the overall result. TCM-5 should be moved 1 or 2 hours later in the trajectory for best results, with the notable exception being when TCM-4 is moved forwards or backwards within the trajectory. In both of these cases, TCM-5 should instead be moved to an earlier point within the trajectory. For TCM-6, the best results are generated when retaining its time of ignition at the nominal placement. When compared to other targeting constraints, Scenario 2 suggests that TCMs 2-4 be moved to an earlier point in the trajectory for best results.

Scenario 3 Analysis Scenario 3 (targeting spacecraft slight path angle, altitude, and velocity magnitude) makes very strong suggestions compared to Scenarios 1 and 2. The dominate vertical lines in all mission maps for Scenario 3 imply that TCM-5 and TCM-6 hold little or no impact on $3\sigma \Delta V$ usage and position dispersions at LOI given the targeting constraints. Like the previous Scenarios, Scenario 3 also encourages an earlier placement for TCM-2 and TCM-3 for the minimization of both fuel usage and position dispersions. Unlike previous Scenarios, Scenario 3 shows TCM-4 as a dominate burn over TCM-5 and TCM-6, suggesting that TCM-4 be moved to an earlier point for best results. Ultimately, Scenario 3 strongly values the placement of TCMs 2-4 over TCMs 5 and 6, and encourages that they be placed earlier in the trajectory for optimal fuel usage and accuracy.

CONCLUSION

The objective of this paper was to utilize the computational speed inherent in LinCov and GRT to generate mission maps. These mission maps enable the rapid visual analysis of a trajectory, removing the need for a computationally heavy optimization tool like a particle swarm or genetic algorithm. In order to generate mission maps for the outbound cislunar trajectory, nominally-zero ΔV TCMs were altered from their initial placements by fixed increments, creating a field of trajectory data. The trajectory’s G matrix was also altered to determine the trajectory’s sensitivity to a change in targeting constraints. By generating mission maps correlating with different trajectory scenarios and cases, trends for reducing $3\sigma \Delta V$ usage and position dispersions at LOI through TCM alterations began to appear. Both TCM-2 and TCM-3 showed reduction in $3\sigma \Delta V$ usage and position dispersions at LOI when moved back 2 hours regardless of targeting constraints. Furthermore, TCM-5 generates significant $3\sigma \Delta V$ usage reductions when moved to a point 1 hour later in

the trajectory. On the other hand, regardless of targeting constraints, burns like TCM-4 and TCM-6 should be left in their nominal position for best results. For targeting constraints, Scenario 1 (targeting the position vector) favors an later placement of TCM-5 when compared to other targeting scenarios. Scenario 2 (targeting both the position vector and the velocity vector) instead suggests that TCMs 2-4 be executed earlier in the trajectory to minimize $3\sigma \Delta V$ usage and position dispersions at LOI. Scenario 3 (targeting spacecraft slight path angle, altitude, and velocity magnitude) also encourages the earlier placement of TCMs 2-4, and argues that TCMs 2-4 have a higher impact on the trajectory compared to TCMs 5 and 6, which can be inferred from the highly vertical lines in Figures ?? & ??. Ultimately, while the analysis results generated through the employment of mission maps in this paper are specific to this trajectory, the systematic process of evaluating an assortment of trajectory variations using a LinCov analysis, GRT, and mission maps can be applied to develop robust, optimal outbound cislunar trajectories and to find desirable engineering trades for future missions to cislunar space.

REFERENCES

- [1] David Woffinden David Geller and Collin York. Generalized reference targeting for spaceflight. In *AAS/AIAA Astrodynamics Specialist Conference*, North Logan, UT, 31 January - 3 February 2025. AAS 25-099.
- [2] D. Geller and D. Christensen. Linear Covariance Analysis for Powered Lunar Descent and Landing. *The Journal of Spacecraft and Rockets*, 46(6):1231–1248, Nov-Dec 2009.
- [3] D. K. Geller. Linear covariance techniques for orbital rendezvous analysis and autonomous on-board mission planning. *Journal of Guidance, Control, and Dynamics*, 29(6):1404–1414, November-December 2006.
- [4] Timothy Goulet, David Woffinden, Nathan Collins, and Blythe Andrews. Robust trajectory design for rendezvous in a near-rectilinear halo orbit. In *AAS GN&C Conference*, Breckenridge, CO, 2023. AAS 23-066.
- [5] Christopher Jewison. *Guidance and Control for Multi-stage Rendezvous and Docking Operations in the Presence of Uncertainty*. PhD thesis, MIT, 2017.
- [6] K. Mand, D. Woffinden, P. Spanos, and R. Zanetti. Rendezvous and proximity operations at the earth-moon l2 lagrange point: Navigation analysis for preliminary trajectory design. In *2014 AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, 2014. AAS 14-376.
- [7] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 1. Academic Press, New York, 1979.
- [8] Travis J. Moesser and David K. Geller. Guidance and navigation linear covariance analysis for lunar powered descent. In *AAS/AIAA Astrodynamics Specialist Conference*, Mackinac Island, Michigan, 19-23 August 2007. AAS 07-313.
- [9] James W. Williams, William E. Brandenburg, David C. Woffinden, and Zachary R. Putnam. Validation of linear covariance techniques for mars entry, descent, and landing guidance and navigation performance analysis. In *AIAA Scitech 2022 Forum*, 2022.
- [10] David Woffinden, Shane Robinson, James Williams, and Zachary Putnam. Linear covariance analysis techniques to generate navigation and sensor requirements for the safe and precise landing - integrated capabilities evolution (splice) project. In *AIAA Scitech 2019 Forum*, San Diego, CA, 7-11 January 2019. AIAA 2019-0662.