

Stochastic Optimal Control to Minimize State Uncertainty

by

Erica L. Jenson

B.S., Aerospace Engineering, The University of Florida, 2017

B.S., Mechanical Engineering, The University of Florida, 2017

M.S., Aerospace Engineering Sciences,

The University of Colorado Boulder, 2019

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Aerospace Engineering Sciences

2022

Committee Members:

Daniel J. Scheeres, Chair

Jay McMahon

Marcus Holzinger

Xudong Chen

Eric Gustafson

Jenson, Erica L. (Ph.D., Aerospace Engineering Sciences)

Stochastic Optimal Control to Minimize State Uncertainty

Thesis directed by Prof. Daniel J. Scheeres

In many spacecraft control scenarios, the standard design procedure begins with a deterministic propellant minimization that does not consider random errors due to navigation uncertainty, control noise, unknown parameters, or mismodeled dynamics. However, these random errors can render a deterministic trajectory infeasible or result in large state deviations that are resource-intensive to correct. Therefore, a deterministic design is followed by feasibility studies to ensure that potential errors will not compromise the mission performance. Alternatively, this dissertation investigates the more direct approach of stochastic optimal control. Stochastic control techniques include uncertainty within the optimization process such that error statistics can be minimized or constrained directly. Two novel stochastic control problems are investigated in this dissertation: 1) open-loop multi-objective trajectory optimization to minimize state error covariance and control energy using indirect methods and 2) closed-loop guidance to minimize mean squared error using dynamic programming. These techniques may be employed independently or in tandem to improve mission robustness. Both approaches consider control-dependent noise that is proportional to the magnitude of the nominal control (a structure that is common in spacecraft thrusters). The stochastic control methods in this dissertation are demonstrated for various mission scenarios including asteroid orbits in microgravity and multi-body orbits in the Earth-Moon system.

Moreover, the growth of state uncertainty over time is dependent on the dynamical system in question, with the degree of nonlinearity playing an important role. This dissertation also studies the relationship between uncertainty and nonlinearity and proposes a semianalytical measure of nonlinearity that is based on the eigenpairs of higher-order tensors. The nonlinearity measure can be used to identify regions of strong nonlinearity, estimate the size of the “linear” region about a nominal trajectory, and inform navigation and control algorithms. Collectively, this research results in a generalized guidance and control framework that facilitates robust performance.

Dedication

To Mom, Dad, and Carolyn.

Acknowledgements

First, I would like to thank Professor Daniel Scheeres for providing this opportunity, advising my research, and affording me the academic freedom to explore many interesting topics. Your guidance over the past five years has shaped the way that I think about astrodynamics problems, and I am a better researcher because of it. Second, I would like to thank Professor Xudong Chen for advising a significant portion of this dissertation — thank you Professor Chen for your contribution to this research and your valuable perspective from outside the aerospace community.

Next, I would like to thank the other members of my dissertation committee for their thoughtful feedback: Professor Jay McMahon, Professor Marcus Holzinger, and Dr. Eric Gustafson. Also, thank you to Eric for hosting me at JPL and believing in the applicability of my work. Thank you to Professor Natasha Bosanac for serving on my comprehensive exam committee and providing helpful research and career advice. I also would like to thank Professor Mai Bando for her time and research collaboration.

This work was supported by a NASA Space Technology Research Fellowship. Thank you to my NASA advisor Dr. Sun Hur-Diaz for your insight, mentorship, and hospitality at Goddard. And thank you to all others at NASA Goddard and JPL who have welcomed me and given me the opportunity to learn from your experience.

Thank you to my graduate student friends at CU Boulder for your friendship, help, and advice. Finally, and most of all, I would like to thank my parents Michelle and Eric, my sister Carolyn, my partner Nathan, and all my friends and family for their support.

Contents

Chapter

| | | |
|----------|---|----------|
| 1 | Introduction | 1 |
| 1.1 | Dissertation Overview | 3 |
| 1.2 | Technical Contributions | 3 |
| 1.3 | Associated Publications | 5 |
| 1.3.1 | Journal Publications | 5 |
| 1.3.2 | Conference Papers | 5 |
| | | |
| 2 | Robust Trajectory Optimization | 7 |
| 2.1 | Introduction | 9 |
| 2.2 | Background | 12 |
| 2.2.1 | Indirect Optimization | 12 |
| 2.2.2 | Shooting Methods | 14 |
| 2.2.3 | Homotopy Methods | 15 |
| 2.3 | Trajectory Optimization Problem Formulation | 16 |
| 2.3.1 | Stochastic Dynamics | 16 |
| 2.3.2 | Cost Functions | 17 |
| 2.4 | Optimal Control Solution | 20 |
| 2.5 | Optimization Algorithm | 24 |
| 2.6 | Terminator Orbit Transfers | 26 |

| | | |
|----------|--|-----------|
| 2.6.1 | Maneuver Scenarios | 26 |
| 2.6.2 | Algorithm Inputs | 28 |
| 2.6.3 | Scenario 1: Orbit Transfer | 29 |
| 2.6.4 | Scenario 2: 180° Phasing Maneuver | 32 |
| 2.6.5 | Scenario 3: Proactive station-keeping | 35 |
| 2.6.6 | Monte Carlo Validation | 35 |
| 2.7 | Discussion | 37 |
| 2.8 | Conclusions | 39 |
| 3 | Continuous Closed-Loop Robust Control | 41 |
| 3.1 | Introduction | 42 |
| 3.2 | Continuous Closed-Loop Control Problem Formulation | 44 |
| 3.3 | Continuous Closed-Loop Control Result | 45 |
| 3.3.1 | Special Case: Optimal Open-Loop Control | 47 |
| 3.3.2 | General Case: Optimal Feedback Control | 49 |
| 3.4 | Continuous Closed-Loop Control Example | 52 |
| 3.5 | Discussion and Conclusions | 54 |
| 4 | Impulsive Closed-Loop Robust Control | 56 |
| 4.1 | Introduction | 57 |
| 4.2 | Impulsive Closed-Loop Control Problem Formulation | 59 |
| 4.3 | Impulsive Closed-Loop Control Result | 61 |
| 4.3.1 | Guiding a Nonlinear System | 63 |
| 4.3.2 | Proof of Theorem 3 | 64 |
| 4.4 | Asteroid Orbiter Robust Guidance | 68 |
| 4.5 | Deterministic Guidance Comparison: Modified LQR | 72 |
| 4.5.1 | Modified LQR | 74 |
| 4.5.2 | Control Midpoints | 75 |

| | | |
|----------|---|------------|
| 4.5.3 | NRHO Application | 76 |
| 4.5.4 | Asteroid Sun-Terminator Orbit Application | 81 |
| 4.6 | Discussion and Conclusions | 86 |
| 5 | Guidance with Noisy Measurements | 87 |
| 5.1 | Introduction | 88 |
| 5.2 | Bounded Cost Function Problem Formulation | 89 |
| 5.3 | Bounded Cost Function Result | 95 |
| 5.4 | Bounded Cost Function Simulation | 96 |
| 5.5 | Discussion and Conclusions | 100 |
| 6 | Tensor Eigenpair Measure of Nonlinearity | 104 |
| 6.1 | Introduction | 105 |
| 6.2 | Background | 107 |
| 6.2.1 | Measures of Nonlinearity | 107 |
| 6.2.2 | Tensor Operations and Eigenpairs | 111 |
| 6.2.3 | Taylor Series Expansions | 116 |
| 6.3 | TEMoN Problem Setup | 120 |
| 6.3.1 | TEMoN Definition | 120 |
| 6.3.2 | Relating α to State Uncertainty | 122 |
| 6.4 | TEMoN Result | 123 |
| 6.4.1 | TEMoN Derivation | 123 |
| 6.4.2 | TEMoN Algorithm | 125 |
| 6.5 | Application to LDTs | 127 |
| 6.5.1 | LDT TEMoN Validation | 127 |
| 6.5.2 | Comparison to NLI and CMoN | 129 |
| 6.5.3 | Linear Region about Equilibrium Points | 130 |
| 6.5.4 | LDT Eigenvector Computation Time | 131 |

| | | |
|----------|--|------------|
| 6.6 | Application to STTs | 132 |
| 6.6.1 | NRHO Sampling Scheme | 132 |
| 6.6.2 | NRHO Nonlinearity | 133 |
| 6.6.3 | STT TEMoN Validation | 133 |
| 6.6.4 | Linear Region about the 9:2 NRHO | 135 |
| 6.6.5 | NRHO Eigenvector Computation Time | 136 |
| 6.6.6 | NRHO LDTs | 136 |
| 6.7 | Application to Parameter Transformations | 137 |
| 6.8 | Discussion | 139 |
| 6.9 | Conclusions | 141 |
| 7 | Conclusions | 142 |
| 7.1 | Summary | 142 |
| 7.2 | Future Work | 143 |
| | Bibliography | 145 |
| | Appendix | |
| A | Stochastic Dynamics and Itô Calculus | 151 |
| B | Dynamical Systems and Orbits | 153 |
| B.1 | Asteroid Sun-Terminator Orbits | 153 |
| B.2 | Near-Rectilinear Halo Orbit | 155 |
| C | Tensor Calculations | 157 |
| C.1 | STT Dynamics | 157 |
| C.2 | Computing Ω Tensors | 158 |
| C.3 | Computing \tilde{T} Tensors | 159 |

| | | |
|-----|---|-----|
| C.4 | Computing $\tilde{\mathcal{T}}$ Tensors | 160 |
| C.5 | Computing Γ Tensors | 160 |

Tables

Table

| | |
|---|-----|
| 2.1 Terminator Orbit Parameters for Trajectory Optimization | 27 |
| 2.2 Algorithm 1 Inputs | 28 |
| 4.1 Terminator Orbit Parameters for Robust Guidance | 70 |
| 4.2 NRHO Baseline Values | 76 |
| 4.3 NRHO Mission Scenarios and Results for 1000-trial Monte Carlo | 78 |
| 4.4 Terminator Orbit Baseline Values | 81 |
| 4.5 Terminator Orbit Mission Scenarios and Results for 1000-trial Monte Carlo | 83 |
| 5.1 Baseline Parameters with Noisy Measurements | 99 |
| 6.1 Equilibrium Point Eigenpair Statistics | 131 |
| 6.2 NRHO Eigenpair Statistics | 136 |
| 6.3 Different TEMoN types applied to orbits A and B. | 139 |

Figures

Figure

- | | | |
|-----|--|----|
| 2.1 | Two mission scenarios: an orbit transfer and a phasing maneuver. $\mathbf{X}(t_0)$, $\mathbf{X}(t_{coast})$, and $\mathbf{X}(t_f)$ are fixed for each initialization of the optimization. | 27 |
| 2.2 | Total number of solutions found. | 29 |
| 2.3 | Transfers from Orbit 1 to Orbit 2 that satisfy the necessary conditions; J_p , J_u , and τ_f are nondimensional. | 30 |
| 2.4 | Three Pareto-optimal transfers are plotted (left), and their final 2- σ covariance ellipses are compared (right). Magenta boxes use dissimilar axis scaling for visibility. | 31 |
| 2.5 | Control profiles for three Pareto-optimal orbit transfers. The minimum covariance solution was found to have a shorter transfer duration, and is therefore terminated sooner. | 32 |
| 2.6 | All 180° phasing maneuvers that satisfy the necessary conditions are plotted in J_p vs. J_u vs. t_f/T_f space, where T_f denotes the approximate period of Orbit 1. | 33 |
| 2.7 | Three Pareto-optimal 180° phasing maneuvers are plotted (left), and their final 2- σ covariance ellipses are compared (right). Magenta boxes use dissimilar axis scaling for visibility. | 34 |
| 2.8 | All proactive station-keeping maneuvers that satisfy the necessary conditions are plotted in nondimensional J_p vs. J_u space. All solutions have a final time of $t_f = 3T_f$ | 35 |
| 2.9 | Three Pareto-optimal proactive station-keeping maneuvers are plotted (left), and their final 2- σ covariance ellipses are compared (right). Magenta boxes use dissimilar axis scaling for visibility. | 36 |

| | | |
|------|--|----|
| 2.10 | Comparing linear and nonlinear (Monte Carlo, $N = 2000$) $2\text{-}\sigma$ covariance propagation for proactive station-keeping maneuvers. | 37 |
| 3.1 | Illustration of the system model in Eq. (3.1). The controller can access sampled measurements, and the goal is to generate feedback control inputs that steer the system as close as possible to a target state. The graphic depicts communication with a ground station, but an onboard controller is also possible. | 45 |
| 3.2 | Numerical simulations of J^* vs. N for the system in Eq.(3.44) ($t_f = 1$, $\sigma = 0.2$, $\mathbf{x}_0 = [1, 1]^\top$, $\mathbf{z}_f = [0, 0]^\top$, $R = 0$). For each case, J^* asymptotically decays and approaches a positive value as its limit. The stable case appears to provide a lower bound for other cases. | 53 |
| 3.3 | Trajectory for $\bar{\mathbf{x}}^*(t) = [\bar{x}_1^*(t), \bar{x}_2^*(t)]^\top$ and the optimal control law for system (3.44) with $\beta = 10$ ($t_f = 1$, $\sigma = 0.2$, $\mathbf{x}_0 = [1, 1]^\top$, $\mathbf{z}_f = [0, 0]^\top$, $R = 0$). Let $y_{t_k} = \bar{\mathbf{x}}^*(t_k)$ for all $k = 1, \dots, N$. Note that $\bar{\mathbf{x}}^*(t_f)$ gets closer to \mathbf{z}_f as N increases. | 54 |
| 4.1 | Illustration of the system model (4.1) over segment k . The controller receives sampled measurements at times t_k (and t_{k+1}) and sends impulsive controls occurring at times $t_{k,j}$, for $j = 1, \dots, p_k$. The graphic depicts communication with a ground station, but an onboard controller is also possible. | 60 |
| 4.2 | Algorithm 3 is demonstrated for guidance about a nominal asteroid orbit ($N = 3$, $p_k = 6$). Red and blue trajectories are sample paths generated by the nonlinear stochastic dynamics. Impulses are scaled for visibility. | 70 |
| 4.3 | The simulation in Figure 4.2 is repeated for ($N = 3$, $p_k = 25$). | 71 |
| 4.4 | Numerical simulation of $\mathbb{E}[J^*]$ vs. N and p_k for guidance about an asteroid orbiting trajectory. $\mathbb{E}[J^*]$ asymptotically decays to a positive value as (N, p_k) are increased. | 72 |
| 4.5 | Navigation update and control scheduling for NRHO Scenarios 1 through 4. The spacecraft is orbiting counter-clockwise from this perspective. | 77 |
| 4.6 | Sample paths for one orbit in NRHO Scenario 2. ΔV vectors are not true to scale. | 79 |

| | | |
|------|---|-----|
| 4.7 | Final state deviations of robust guidance (black circles) and LQR (red diamonds) for NRHO Scenarios 1-4, computed from 1000-trial Monte Carlo simulations. | 79 |
| 4.8 | Mean squared deviation (MSD) of robust guidance and LQR sample paths as we scale the parameters in Table 4.2, computed from 100-trial Monte Carlo simulations of Scenario 1. . . | 80 |
| 4.9 | Mean ΔV of robust guidance and LQR sample paths as we scale the parameters in Table 4.2, computed from 100-trial Monte Carlo simulations of Scenario 1. | 80 |
| 4.10 | Eight evenly-spaced navigation updates/controls over a period of four orbits (one week scenario). In the one month scenario, this schedule is applied four consecutive times. | 82 |
| 4.11 | Final state deviations of robust guidance (black circles) and LQR (red diamonds) for asteroid Sun-terminator orbits, computed from 1000-trial Monte Carlo simulations. | 84 |
| 4.12 | Sample paths for asteroid orbit one week scenario. | 84 |
| 4.13 | Sample paths of one week scenario with $R = I_3$. The robust and LQR control solutions are nearly identical when the propellant cost is prioritized. | 84 |
| 4.14 | Mean ΔV of each control maneuver in the asteroid one week scenario, computed from 1000 Monte Carlo trials. The LQR concentrates control effort toward the end of the trajectory . . | 85 |
| 5.1 | Divergence of upper and lower cost function bounds as the level of along-axis control noise is increased. | 100 |
| 5.2 | Divergence of upper and lower cost function bounds as the level of measurement noise is increased. | 101 |
| 5.3 | Divergence of upper and lower cost function bounds as the initial state uncertainty is increased. | 102 |
| 5.4 | Divergence of upper and lower cost function bounds as the number of measurements are increased. | 102 |
| 5.5 | Variation of upper and lower cost function bounds as the number of maneuvers per segment is increased. | 103 |
| 6.1 | Comparing τ_k^* (TEMoN terms) to s_k^* (S-TEMoN terms) at Lagrange points L_1 through L_5 ($\ \delta\mathbf{x}\ = 10^{-3}$). | 128 |

| | | |
|------|--|-----|
| 6.2 | Comparing TEMoN terms (τ_k^*) to empirically-sampled S-TEMoN terms (s_k^*); the TEMoN detects stronger nonlinearity and is more accurate than the S-TEMoN in all cases. | 128 |
| 6.3 | Comparing the TEMoN and \aleph for $\ \delta\mathbf{x}\ = 10^{-3}$ at each Lagrange point. The TEMoN is a relatively tight upper bound on \aleph | 129 |
| 6.4 | Comparing MoN that are normalized such that the largest value of each MoN equals one; considering a deviation of $\ \delta\mathbf{x}\ = 10^{-3}$ at each Lagrange point. | 130 |
| 6.5 | Computing the TEMoN for different deviation magnitudes; $\tau_k^* > 1$ indicates that a higher-order term is dominating the linear term and the deviation is outside the linear region. | 131 |
| 6.6 | Sampling scheme along 9:2 NRHO based on the arc lengths of orbit radius and speed. | 132 |
| 6.7 | TEMoN and τ_k^* computed using the STTs along a family of NRHOs ($\ \delta\mathbf{x}_0\ = 10^{-3}$). | 134 |
| 6.8 | Comparing the TEMoN, S-TEMoN, and \aleph for the 9:2 NRHO. The TEMoN is an upper bound on \aleph . Large discrepancies between TEMoN and S-TEMoN indicate that empirical sampling is insufficient. | 135 |
| 6.9 | Computing the TEMoN for different deviation magnitudes about the 9:2 NRHO. The legend indicates the magnitude of $\delta\mathbf{x}(t_0)$ | 136 |
| 6.10 | TEMoN applied to the LDTs for the 9:2 NRHO; when applied to the LDTs, the TEMoN only detects instantaneous nonlinearity in the dynamics. | 137 |
| 6.11 | Orbits A and B and their initial conditions (IC). | 139 |

Chapter 1

Introduction

Due to the extreme cost of delivering spacecraft mass to Earth orbit and beyond, propellant consumption is a driving factor in spacecraft mission design — spacecraft trajectories are often designed to minimize the propellant required to reach a destination. In addition to the baseline design, supplemental propellant is needed to correct errors in flight that cause the spacecraft to deviate from its nominal trajectory. Without frequent correction maneuvers, errors resulting from navigation uncertainty, thruster noise, or mismodeled dynamics can lead to mission failure or large deviations that are expensive to correct (in propellant cost and human effort). Because these errors are random and unknown during the design process, the simplest approach is to optimize the baseline trajectory without considering errors (e.g., optimizing the deterministic, nominal trajectory to minimize propellant cost). However, errors must still be accounted for, and the historical procedure involves robustness analysis through linear covariance propagation methods and nonlinear Monte Carlo simulations. If a baseline trajectory remains feasible with the inclusion of random errors, then the trajectory design was successful; otherwise, the design and the robustness analysis must be iterated until the trajectory is sufficiently robust. This process can be time consuming, and there is no guarantee that the trajectory remains propellant-optimal in the presence of uncertainty and correction maneuvers.

Alternatively, this dissertation uses an approach where uncertain errors are considered in the design stage, i.e., stochastic optimal control. Stochastic control methods account for uncertainty within an optimization process by incorporating randomness in the system model, the objective function, and/or the problem constraints. This dissertation will use stochastic control to minimize error in the spacecraft state directly. This approach can be thought of as maximizing the trajectory robustness. Although the errors

affecting the spacecraft performance are random, such as a random solar radiation pressure perturbation, the statistics of these random processes are assumed to be known. Moreover, the path taken by the spacecraft determines how these random errors affect the spacecraft state uncertainty, and a control policy can be designed to minimize the influence of random errors. The state uncertainty when the spacecraft arrives at its target is of particular interest.

Control-dependent uncertainties, also referred to as maneuver execution errors, are considered throughout this dissertation. Due to the increased complexity of control-dependent noise, it is often ignored during trajectory optimization. However, experimental studies indicate that control-dependent noise can be significant. Notably, spacecraft electric propulsion engines experience noisy discharge current fluctuations that are linearly proportional to the commanded thrust by up to 13% [1, 2]. Moreover, the widely-used Gates model for maneuver execution error incorporates both fixed and proportional errors in thrust magnitude and pointing [3–5]. In short, higher thrust levels produce larger thrust errors, and that effect will be modeled throughout this dissertation. This proportional configuration will be referred to as “control-linear” noise. Control-linear noise can also be viewed as a linearized approximation of control-dependent noise in general, expanding the applicability of the control-linear model. Other instances of control-dependent noise can be found in robotics motion planning and the human sensorimotor system [6], so the model is not limited to space mission design.

Small body orbiters are among the missions that may benefit from uncertainty minimization, especially in the case of control-dependent noise. In the low gravity environment around asteroids, maneuvers require very little propellant (e.g. ΔV s of millimeters per second can have a large effect [7, 8]). Unfortunately, these trajectories are also highly sensitive to maneuver execution error and environmental noise. This dissertation will demonstrate that, through stochastic optimization, significant error reduction is possible with small penalties in propellant when compared to deterministic propellant-optimal solutions. Likewise, any mission scenario that requires a high level of precision is a potential application for uncertainty minimization, e.g. rendezvous and docking maneuvers or operations in highly-sensitive, unstable environments. Both open and closed-loop control methods will be considered in this dissertation and demonstrated for a variety of mission scenarios.

In robust control, the underlying dynamics play an important role in the evolution and growth of uncertainty over time. The degree of nonlinearity of the dynamical system is particularly important. Accordingly, this dissertation also proposes a novel measure of nonlinearity (MoN) that is based on tensor eigenpairs. This MoN can be used to explore the relationship between uncertainty and nonlinearity and has numerous applications in guidance, navigation, and uncertainty propagation.

1.1 Dissertation Overview

The research content in this dissertation is divided into five chapters: Chapter 2 covers robust, nonlinear, multi-objective trajectory optimization to minimize control energy and state error covariance using indirect methods. Chapter 3 introduces closed-loop robust control for sampled linear systems using continuous control inputs and dynamic programming. Chapter 4 builds on the foundation of Chapter 3, extending the closed-loop method to a more general problem scenario with impulsive control inputs and applying the robust control law as a neighboring guidance law. Chapter 5 expands the theory in Chapter 4 to the case of imperfect navigation. Finally, Chapter 6 develops semianalytical measures of nonlinearity based on tensor eigenpairs and discusses their applications in guidance, navigation, and control.

To the greatest extent possible, the notation is standardized throughout this document; however, each chapter contains a nomenclature to define variables that are unique to that chapter. Each chapter includes an introduction, background information, problem formulation, solution, pseudoalgorithms, numerical examples, and conclusions. Topics that appear in multiple chapters, e.g., the formulation of the circular-restricted three-body problem, are described in a separate appendix.

1.2 Technical Contributions

This dissertation makes technical contributions in three major areas: robust nonlinear trajectory optimization with control-dependent noise, closed-loop control with control-dependent noise and limited state information, and semianalytical measures of nonlinearity.

Previous work in robust spacecraft trajectory optimization does not address the multi-objective optimization of control energy and covariance, nor the problem of uncertainty minimization with control-

dependent noise. In Chapter 2, a novel multi-objective optimization of control energy and covariance with control-dependent noise is solved using indirect methods. The analytical nature of the indirect solution provides valuable insight into the problem dynamics, and may serve as a validation method for numerical optimization algorithms in the future. Chapter 2 also introduces the concept of proactive station-keeping as an alternative to trajectory correction maneuvers. Rather than designing maneuvers to correct errors as they appear along a ballistic trajectory, the proactive station-keeping scenario applies open loop control to proactively manipulate the covariance in a way that reduces future state error without feedback control.

This dissertation provides an analytical solution to another novel optimization problem in Chapters 3 and 4: closed-loop control to minimize mean squared state error in the case of sampled measurements and control-dependent noise. The algorithm is derived for a linear system, but will be applied to nonlinear astrodynamics systems as a neighboring guidance law (by linearizing state deviations about nominal trajectories). Both continuous and impulsive control formulations are considered such that the results may be applicable to a variety of spacecraft guidance scenarios. The closed-loop control laws presented in this dissertation are derived analytically using dynamic programming. The result is a linear state feedback controller with feedback gains that can be computed offline a priori. In Chapter 5, an analytical approach (using bounded cost functions) is proposed to investigate a previously intractable problem — the optimal control of a linear system with control-dependent noise and noisy, partial-state measurements.

Finally, a new nonlinearity measure based on tensor eigenpairs, the tensor eigenpair measure of nonlinearity (TEMoN), is introduced in Chapter 6. Earlier measures of nonlinearity require empirical sampling or numerical optimization. Alternatively, the method developed in this dissertation is semianalytical. The TEMoN can be used to study the evolution of nonlinearity along a spacecraft trajectory, identify regions of strong nonlinearity, and quantify the size of the linear region about a nominal trajectory. Chapter 6 will describe how this nonlinearity measure may be used alongside guidance and navigation algorithms.

Collectively, this research results in a generalized guidance and control framework that facilitates robust performance. This dissertation takes an analytic-first approach to these problems. When compared to numerical methods, the analytical techniques herein give broader insight into all aspects of a solution — for example, the mathematical structure of a control profile, not simply the numerical value of the control.

Thus, the results in this dissertation serve as an analytical foundation for future numerical methods.

1.3 Associated Publications

This dissertation is comprised of the following publications and related work.

1.3.1 Journal Publications

- (1) E. L. Jenson, X. Chen, and D. J. Scheeres, “Optimal Control of Sampled Linear Systems with Control-Linear Noise,” *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 650-655, Jul. 2020.
- (2) E. L. Jenson, X. Chen, and D. J. Scheeres, “Optimal Spacecraft Guidance With Asynchronous Measurements and Noisy Impulsive Controls,” *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1813-1818, Nov. 2021.
- (3) E. L. Jenson and D. J. Scheeres, “Multi-Objective Optimization of Covariance and Energy for Asteroid Transfers,” *J. Guid. Control Dyn.*, vol. 44, no. 7, pp. 1253-1265, Jul. 2021.
- (4) E. L. Jenson and D. J. Scheeres, “Semianalytical Measures of Nonlinearity Based on Tensor Eigenpairs,” (in review).

1.3.2 Conference Papers

- (1) E.L. Jenson, D.J. Scheeres, “Robust Trajectory Optimization Using Minimum-Uncertainty Cost Functions,” AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, USA, Aug. 2019, AAS 19-838.
- (2) E.L. Jenson, D.J. Scheeres, “Trajectory Optimization Using Combined Minimum-Uncertainty and Minimum-Energy Cost Functions,” 70th International Astronautical Congress, Washington D.C., USA, Oct. 2019, IAC-19-C1-1.8.
- (3) E.L. Jenson, D.J. Scheeres, “Multi-Objective Optimization of Covariance and Energy for Asteroid Transfers,” AAS/AIAA Astrodynamics Specialist Conference, South Lake Tahoe, CA, USA

(virtual), Aug. 2020, AAS-20-550.

- (4) E.L. Jenson, D.J. Scheeres, “Semianalytical Measures of Nonlinearity Based on Tensor Eigenpairs,” AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, USA, (virtual), Aug. 2021, AAS 21-546.
- (5) E.L. Jenson, D.J. Scheeres, X. Chen, “Robust Spacecraft Guidance with Control-Dependent Noise: Analysis and Application,” AIAA SciTech Forum, San Diego, CA , USA, Jan. 2022.

Chapter 2

Robust Trajectory Optimization

Chapter 2 Nomenclature

| | | |
|--------------------------|---|------------------------------------|
| $A(\mathbf{X}'(t))$ | = | linearized dynamics |
| a_{SRP} | = | SRP acceleration |
| B | = | control matrix |
| d | = | central body distance from the Sun |
| F_1 | = | Formulation 1 cost function |
| F_2 | = | Formulation 2 cost function |
| $\mathbf{f}(\mathbf{X})$ | = | unforced nonlinear dynamics |
| \mathbf{g} | = | terminal manifold constraint |
| \mathcal{H} | = | Hamiltonian |
| $I_{m \times m}$ | = | $m \times m$ identity matrix |
| J_P | = | covariance cost function |
| J_u | = | energy cost function |
| J_2 | = | second order zonal harmonic |
| J_3 | = | third order zonal harmonic |
| m | = | number of control inputs |
| n | = | number of states |
| n_b | = | mean motion at Bennu surface |

| | | |
|-----------------------------|---|--|
| P_ℓ | = | ℓ th Legendre polynomial |
| p_0 | = | solar radiation pressure at 1 AU |
| R | = | energy cost weighting matrix |
| r | = | radius |
| \mathbf{r} | = | position vector |
| T | = | orbit period |
| t | = | time |
| $\mathbf{u}(t)$ | = | control vector |
| V_g | = | gravitational potential |
| \mathbf{v} | = | velocity vector |
| W_t | = | standard Wiener process |
| \mathcal{W} | = | weighting matrix for $\Sigma(t_f)$ |
| \mathcal{W}' | = | weighting matrix for $\Phi_p \Sigma(t_f) \Phi_p^\top$ |
| \mathbf{X}_t | = | stochastic state vector |
| $\mathbf{X}'(t)$ | = | nominal state vector |
| $\widetilde{\mathbf{X}}(t)$ | = | nominal state vector augmented with J_u |
| \mathbf{x}_t | = | stochastic state error vector |
| Z | = | spacecraft mass-to-area ratio |
| ΔV | = | impulsive velocity change |
| η | = | cost function weighting parameter |
| $\boldsymbol{\lambda}$ | = | costate vector |
| $\boldsymbol{\lambda}_X$ | = | position and velocity components of the costate vector |
| μ | = | gravitational parameter |
| ρ | = | spacecraft reflectance |
| $\Sigma(t)$ | = | state error covariance matrix |
| σ | = | control noise factor |
| τ | = | nondimensional time |

| | | |
|------------------|---|--|
| $\Phi(t_j, t_i)$ | = | state transition matrix from t_i to t_j |
| Φ_p | = | state transition matrix for covariance propagation after t_f |
| ψ | = | vector of Lagrange multipliers |
| ω | = | rate of coordinate frame rotation |

2.1 Introduction

This chapter is focused on stochastic trajectory optimization for nonlinear dynamical systems. A multi-objective optimization will be performed to minimize both control energy and state error covariance. The term “trajectory optimization” is used here to describe the design of an open-loop control strategy and the corresponding nominal trajectory, i.e., feedback control is not included in this chapter.

The methods detailed here are not application specific, but will be demonstrated for orbit transfers around a small asteroid. Other authors have utilized stochastic optimization techniques to tackle uncertainty in the asteroid environment. The sequential optimization and reliability assessment (SORA) procedure has been applied to asteroid soft landings: a deterministic propellant optimization and a reliability analysis are iterated systematically until robustness requirements are met [9]. Other authors have designed risk-aware trajectories by augmenting primer vector theory with deterministic and stochastic state inequality constraints [10]. Unscented guidance has also been proposed for asteroid proximity operations, in which the sigma points of an unscented transform are controlled [11]. In contrast to these techniques, the work in this chapter seeks to minimize uncertainty directly.

To date, research in uncertainty minimization for astrodynamics trajectory design is limited. Relevant previous work was done by Zimmer in [12] and Hu in [13]. Using indirect optimization methods, Zimmer performed a weighted optimization of control energy and covariance by augmenting the state vector with covariance terms, which were propagated linearly with additive white Gaussian noise. The minimum energy solution was used as an initial guess, and a homotopy was performed by increasing the weights of covariance terms in the cost function. Zimmer’s method can produce significant covariance improvement. However, including covariance in the state vector increases the sensitivity of the problem and impedes convergence in

shooting methods. For a six-dimensional position and velocity state, the covariance adds 21 unique costates, and it is difficult to generate initial guesses independent of the minimum energy solution. By using the minimum energy solution as an initial guess, the solution domain is restricted to a single homotopy, and there is no guarantee that the true minimum covariance solution belongs to this homotopy. (The indirect approach, homotopy processes, shooting methods, and the difference between control energy and propellant are described in the next section.)

In [13], Hu considered a weighted propellant-and-covariance cost function and used desensitized optimal control techniques to improve the accuracy of asteroid landing trajectories. The desensitized optimal control method incorporates linear-quadratic regulator (LQR) feedback control, and dynamical uncertainties are modeled as stochastic accelerations (additive noise). In [12, 13] the authors penalized covariance, but did not seek covariance and energy Pareto fronts. Nor do these methods accommodate control-dependent noise, which was identified as a significant source of uncertainty in Chapter 1.

The predominant contributions of this chapter are as follows: the method in [12] is extended to include control-linear noise, as described in Chapter 1, and a multi-objective approach is formalized to locate trajectories that are Pareto-optimal in control energy and state error covariance. The first modification will be to exclude covariance from the state vector, reducing sensitivity during shooting methods and eliminating the need for a homotopy. This will improve flexibility, facilitate a broader characterization of the solution domain, and enable a multi-objective optimization. The minimum propellant problem, also known as minimum fuel, is not considered here. Rather, control energy is minimized to leverage structural similarities between the energy and covariance cost functions (discussed further in Section 2.4). Moreover, the continuous nature of the minimum energy control is more amenable to shooting methods [14].

Multi-objective optimization is a trade-off between distinct objectives. A feasible solution is said to dominate another solution if it is superior in at least one objective and is no worse in all other objectives. All non-dominated solutions form a Pareto front from which one objective cannot be improved without decreasing performance in another objective. Thus, the multi-objective optimization seeks the set of Pareto-optimal solutions. This chapter will consider two formulations for multi-objective optimization: 1) a combined covariance and energy minimization via weighted cost functions (weighted-sum method) and 2) a minimum

covariance optimization with an energy constraint (ϵ -constraint method). Both involve scalarizing the problem as a single objective. In the weighted-sum case, multiple cost functions are linearly combined and their relative influence is dictated by weighting parameters [15]. Alternatively, in the ϵ -constraint method, one cost function is optimized while the others are treated as constraints [15, 16]

The weighted-sum method is simple to implement, but there is no guarantee that solutions will lie on the Pareto front in general. Conversely, globally-optimal solutions to the ϵ -constraint problem are guaranteed to lie on a Pareto front, making the ϵ -constraint method more rigorous. However, the ϵ -constraint method requires additional constraints, and their attainable values must be determined. A theorem will be presented in this chapter to prove that, in the case of covariance-and-energy optimization, the weighted-sum and ϵ -constraint methods produce the same optimal trajectories. The simpler method (weighted sum) will be implemented without sacrificing optimality, and the weighting parameters that correspond to Pareto-optimal solutions will be derived.

More sophisticated, non-scalarized multi-objective optimization methods utilize non-dominance sorting and genetic algorithms [17–19] or Multi Agent Collaborative Search [20]. These methods are not considered here. Instead, this chapter will prove analytically that the scalarized approach is sufficient despite its simplicity. Due to the rigor of indirect methods and the ϵ -constraint approach, the results here may be useful for validation of other covariance-and-energy optimizations.

The optimization methods presented in this chapter will be demonstrated for orbit transfers and phasing maneuvers between Sun-terminator orbits in the Hill three-body problem (H3BP) augmented with solar radiation pressure (SRP) and J_2 and J_3 spherical harmonics for the asteroid gravity field [21]. Terminator orbits are SRP-perturbed orbits that lie in an asteroid’s Sun-terminator plane; their angular momentum vector is aligned with the sunlight direction. Due to their stability characteristics, terminator orbits provide a suitable configuration for asteroid orbit operations (for example, the OSIRIS-REx mission to the asteroid Bennu [22]). Three terminator orbit maneuver scenarios will be optimized in this chapter: orbit transfers, phasing maneuvers, and “proactive station-keeping.”

Proactive station-keeping addresses a scenario in which a spacecraft is already on track to intersect a target state in the absence of initial state error or noise; however, the existing trajectory has unfavorable

uncertainty dynamics. Traditionally, errors would be estimated along the trajectory and retroactive corrective maneuvers would be planned to target the nominal state. In the proactive station-keeping scenario, open-loop control will be applied to alter the nominal trajectory in a way that produces favorable uncertainty dynamics. Despite the addition of control-dependent noise, this chapter will demonstrate that proactive station-keeping can greatly improve uncertainty performance and potentially eliminate the need for correction maneuvers in flight.

This chapter is organized as follows. Section 2.2 provides background information on indirect methods, shooting methods, and homotopies. Section 2.3 formulates the trajectory optimization problem. Section 2.4 discusses the optimal control derivation and result. An optimal control pseudoalgorithm is formalized in Section 2.5. Section 2.6 provides three examples of covariance-and-energy minimization: orbit transfers, phasing maneuvers, and proactive station-keeping around the asteroid Bennu.

2.2 Background

2.2.1 Indirect Optimization

This section provides a brief description of the indirect optimization approach used throughout this chapter. A trajectory optimization problem is a calculus of variations problem, i.e., an optimization of functionals, which map functions (the control profile and trajectory dynamics) to scalars (the cost function). With continuous control, the control problem is infinite-dimensional but may be reduced to finite dimensions through either analytical analysis (indirect methods) or approximation (direct methods) [23]. In indirect optimization, the necessary conditions for optimality are applied to derive expressions for the optimal control profile and the costate (adjoint) differential equations. Then, the optimization problem is reduced to a boundary value problem where only the states and costates at the boundary points must be determined. Consider the cost function

$$J = K(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{X}, \mathbf{u}, t) dt, \quad (2.1)$$

where the terms K and L are an unspecified terminal cost and a Lagrangian, respectively. This formulation is often called a Bolza-type cost function. In general, a Hamiltonian is formed as follows

$$\mathcal{H} = L(\mathbf{X}, \mathbf{u}, t) + \boldsymbol{\lambda}^\top \frac{d\mathbf{X}(t)}{dt}, \quad (2.2)$$

in which $\boldsymbol{\lambda}$ denotes the costate vector. The optimal control must satisfy the following necessary condition to produce a stationary point of the Hamiltonian.

$$\left. \frac{\partial \mathcal{H}}{\partial \mathbf{u}} \right|_{\mathbf{u}^*} = \mathbf{0} \quad (2.3)$$

The optimal Hamiltonian, \mathcal{H}^* , is computed by substituting \mathbf{u}^* in \mathcal{H} . The costate dynamics are dictated by the second necessary condition as follows.

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}^*}{\partial \mathbf{X}} \quad (2.4)$$

With the state dynamics, costate dynamics, and optimal control being known, the next step is to determine boundary values for the state and costate which lead to feasible solutions. Feasible solutions are those for which no user-defined state or control constraints are violated. Additionally, optimal trajectories must satisfy the following transversality conditions, which contribute more constraints at the initial and final times:

$$\boldsymbol{\lambda}(t_0)^\top = -\frac{\partial K}{\partial \mathbf{X}_0} - \boldsymbol{\psi}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{X}_0}, \quad (2.5)$$

$$\boldsymbol{\lambda}(t_f)^\top = \frac{\partial K}{\partial \mathbf{X}_f} + \boldsymbol{\psi}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{X}_f}, \quad (2.6)$$

$$\mathcal{H}(t_0) = \frac{\partial K}{\partial t_0} + \boldsymbol{\psi}^\top \frac{\partial \mathbf{g}}{\partial t_0}, \quad (2.7)$$

$$\mathcal{H}(t_f) = -\frac{\partial K}{\partial t_f} - \boldsymbol{\psi}^\top \frac{\partial \mathbf{g}}{\partial t_f}. \quad (2.8)$$

The variable $\boldsymbol{\psi}$ is a vector of Lagrange multipliers. The necessary conditions only guarantee stationary points of the cost function, not global minima. Therefore, it is recommended to compare multiple solutions to the necessary conditions.

2.2.2 Shooting Methods

The necessary conditions reduce the optimal control problem to a boundary value problem. The only remaining task is to determine states and costates at the boundary points that satisfy the user-defined constraints and transversality conditions. If the problem is formulated with only two boundary points (at the beginning and end of the trajectory), then the problem is called a two-point boundary value problem (TPBVP). It is also possible to formulate a problem with multiple intermediate boundary points; in this case, the problem is a multi-point boundary value problem (MPBVP) and additional constraints are required to enforce state and costate continuity at the intermediate boundary points. Only TPBVPs are considered in this chapter, and a single shooting method will be used to solve TPBVPs.

In the single-shooting scheme, a guess of the free parameters at one boundary is generated, the trajectory is integrated to the second boundary, the error in the terminal constraints and transversality conditions is computed at the second boundary, and the free parameters at the first boundary are updated using a Newton-Raphson update scheme. The update is iterated until convergence. Allow \mathbf{p} to be a vector of free parameters and \mathbf{c} to be a vector of constraints. The following differential correction scheme will be used to update \mathbf{p} until constraints are satisfied. The parameter ζ is used to scale the size of the update step as needed to improve convergence.

$$\mathbf{p}_{i+1} = \mathbf{p}_i - \zeta \left(\left. \frac{\partial \mathbf{c}(\mathbf{p})}{\partial \mathbf{p}} \right|_{\mathbf{p}_i} \right)^{-1} \mathbf{c}_i \quad (2.9)$$

For each initial guess, the correction is iterated until the constraint error is smaller than a set tolerance.

The generation of a “good” initial guess is often the most difficult step in an indirect optimization problem. Initial guesses may be generated at random or systematically through techniques such as an adjoint-control transformation [24]. In the latter case, the costates that form the primer vector are guessed based on an intuitive guess for the optimal control. Another option is to solve a similar, but easier, optimization problem and use that solution as the initial guess for a more complex problem. This technique is often called a homotopy, and is described further in the next subsection.

2.2.3 Homotopy Methods

Homotopy is a concept in topology that describes the continuous deformation from one function to another. The term homotopy is often used informally in the optimization literature to describe a technique that helps the user generate initial guesses for optimization problems, particularly in the case of indirect methods. Consider two cost functions, J_a and J_b , and assume that J_a is easier to optimize. One problem may be easier than another for numerous reasons: the cost function is less sensitive to the free parameters, the equations are faster to integrate, the constraints are easier to satisfy, etc. In some cases, it is possible to use a solution to J_a as an initial guess for J_b by defining a combined cost function as follows.

$$J = (1 - \eta)J_a + \eta J_b \quad (2.10)$$

Then, the scaling parameter η may be varied incrementally to transition a J_a solution to a J_b solution. By using existing solutions as initial guesses and varying the weighting parameter η , homotopy solution families can be generated. Homotopies may identify solutions in highly sensitive regions that would be difficult to locate from random initial guesses.

One example is the homotopy between minimum energy and minimum propellant cost functions [25]. The minimum energy cost function typically refers to the integral of the squared L^2 norm of the control:

$$J_e = \int_{t_0}^{t_f} \|\mathbf{u}(t)\|^2 dt, \quad (2.11)$$

$$= \int_{t_0}^{t_f} \mathbf{u}(t)^\top \mathbf{u}(t) dt. \quad (2.12)$$

Alternatively, the minimum propellant cost function refers to the integral of the L^2 norm of the control:

$$J_u = \int_{t_0}^{t_f} \|\mathbf{u}(t)\| dt. \quad (2.13)$$

This problem is equivalent to maximizing the final mass, i.e., $J = -m_f$. The minimum energy optimization produces a continuous control profile that is less sensitive to shooting methods. However, the minimum propellant problem results in a bang-bang control structure that is more difficult to converge on using shooting methods. It is possible to perform a homotopy between J_e and J_u , as seen in [25]. Other homotopy use cases include “turning on” a perturbation, improving integrator accuracy, or altering an ephemeris model.

2.3 Trajectory Optimization Problem Formulation

2.3.1 Stochastic Dynamics

Consider the Itô form of a stochastic differential equation with Brownian motion [26]:

$$d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, \mathbf{u}(t), t)dt + \mathbf{h}(\mathbf{X}_t, \mathbf{u}(t), t)dW_t. \quad (2.14)$$

An overview of stochastic dynamics is given in Appendix A. If the dynamics are linear in the control and control-linear noise is assumed, Eq. (2.14) simplifies to:

$$d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t)dt + B\mathbf{u}(t)(dt + \sigma dW_t), \quad (2.15)$$

in which $\mathbf{X}_t \in \mathbb{R}^n$ is the nondeterministic position and velocity state of the system, $\mathbf{f}(\mathbf{X}_t, t) \in \mathbb{R}^n$ is the natural system dynamics, $\mathbf{u}(t) \in \mathbb{R}^m$ is the nominal control input, and W_t is a standard Wiener process driving the control-dependent noise. To simplify the notation, the explicit time dependence of $\mathbf{f}(\mathbf{X}_t, t)$ will be suppressed moving forward. This formulation corresponds to a control-linear noise model, in which the level of noise is proportional to the magnitude of the control input. The scalar constant σ dictates the level of control noise (i.e. $\sigma = 0.01$ enforces a standard deviation of 1% of the control magnitude). The control is assumed to be open-loop.

The state vector is specified $\mathbf{X}_t = [\mathbf{r}^\top, \mathbf{v}^\top]^\top$ where $\mathbf{r} \in \mathbb{R}^{n/2}$ and $\mathbf{v} \in \mathbb{R}^{n/2}$ are the position and velocity vectors, respectively. The initial state \mathbf{X}_{t_0} will be uncertain with known statistics. As is typical in preliminary astrodynamics trajectory design, it is assumed that the acceleration is controlled directly such that

$$B = \begin{bmatrix} [0]_{m \times m} \\ I_{m \times m} \end{bmatrix}. \quad (2.16)$$

Due to initial state uncertainty and the Wiener process W_t , the true state dynamics $d\mathbf{X}_t$ cannot be known to the controller a priori (i.e. the true state \mathbf{X}_t cannot be controlled and optimized). Alternatively, an optimal control law will be designed for a deterministic nominal trajectory, $\mathbf{X}'(t)$, that is not influenced by initial state uncertainty or noise:

$$\frac{d}{dt}\mathbf{X}'(t) = \mathbf{f}(\mathbf{X}'(t)) + B\mathbf{u}(t). \quad (2.17)$$

The state error, \mathbf{x}_t , is defined as the difference between the true state \mathbf{X}_t and the nominal state $\mathbf{X}'(t)$ as follows.

$$\mathbf{x}_t = \mathbf{X}_t - \mathbf{X}'(t) \quad (2.18)$$

The mean error is denoted $\bar{\mathbf{x}}(t)$, and the error covariance is computed by

$$\Sigma(t) = \mathbb{E}[(\mathbf{x}_t - \bar{\mathbf{x}}(t))(\mathbf{x}_t - \bar{\mathbf{x}}(t))^\top]. \quad (2.19)$$

Unlike the stochastic state, $\mathbf{X}'(t)$, $\bar{\mathbf{x}}(t)$ and $\Sigma(t)$ are deterministic and can be controlled and optimized. Such is the foundation of the covariance minimization performed in this chapter.

This chapter will use a linearized approximation for $\bar{\mathbf{x}}(t)$ and $\Sigma(t)$, as described in Appendix A. Using the Itô rule, the linearized state error covariance at the final time is

$$\Sigma(t_f) = \Phi(t_f, t_0)\Sigma(t_0)\Phi(t_f, t_0)^\top + \sigma^2 \int_{t_0}^{t_f} \Phi(t_f, t)B\mathbf{u}(t)\mathbf{u}(t)^\top B^\top \Phi(t_f, t)^\top dt. \quad (2.20)$$

The STM from t_i to t_j is denoted by $\Phi(t_j, t_i)$, and is computed by numerically integrating the following differential equation with $\Phi(t_i, t_i) = I$.

$$\dot{\Phi}(t_j, t_i) = A(\mathbf{X}(t_j))\Phi(t_j, t_i) \quad (2.21)$$

Additionally, the STM partial $\partial\Phi(t_f, t)/\partial\mathbf{X}$, which will be required in the optimal control solution, is computed by numerically integrating the following equation with the initial condition $\partial\Phi(t_f, t_f)/\partial\mathbf{X} = [0]_{n \times n}$.

$$\frac{d}{dt} \left(\frac{\partial\Phi(t_f, t)}{\partial\mathbf{X}} \right) = \frac{\partial A(\mathbf{X}'(t))}{\partial\mathbf{X}} \Phi(t_f, t) + A(\mathbf{X}'(t)) \frac{\partial\Phi(t_f, t)}{\partial\mathbf{X}} \quad (2.22)$$

2.3.2 Cost Functions

Section 2.4 will detail the multi-objective optimization of two cost functions. The first, J_P , is a measure of the size of the error covariance matrix:

$$\begin{aligned} J_P &= \text{trace}(\mathcal{W}'\Phi_p\Sigma(t_f)\Phi_p^\top), \\ &= \text{trace}(\mathcal{W}\Sigma(t_f)), \end{aligned} \quad (2.23)$$

where $\mathcal{W} = \Phi_p^\top \mathcal{W}' \Phi_p$ and $\mathcal{W}' \in \mathbb{R}^{n \times n}$ allows relative weighting between different elements of $\Phi_p \Sigma(t_f) \Phi_p^\top$. The term Φ_p denotes the STM from time t_f to a future time t_p ; Φ_p enables covariance propagation for an additional arc after the control cutoff at t_f . Propagating the covariance further in time can prevent bad behavior, like a control spike at the final time. Because a spike in control noise does not inflate the covariance instantaneously (the effect must be integrated into the dynamics over time), a control spike at the end of the trajectory will not significantly increase the covariance at the final time. However, a control spike at the final time can result in detrimental *future* uncertainty. Such behavior is corrected by propagating the covariance for an additional period after the control cutoff.

The second cost function, J_u , encompasses the weighted energy cost of the control:

$$J_u = \int_{t_0}^{t_f} \mathbf{u}(t)^\top R \mathbf{u}(t) dt, \quad (2.24)$$

where R is a positive definite matrix.

Two multi-objective formulations will be considered for optimization of J_P and J_u . The first, Formulation 1 (weighted-sum), is considered for its ease of implementation. However, Formulation 2 (ϵ -constraint) is a more rigorous approach to multi-objective optimization and is preferred. Both formulations, including their cost functions (F_1 and F_2) and terminal manifold constraints (\mathbf{g}_1 and \mathbf{g}_2), are defined below. Neither control nor path constraints are included in either formulation. The “*” superscript denotes an optimized quantity.

Formulation 1. Covariance and energy are minimized simultaneously in F_1 , where J_P and J_u are relatively-weighted terms. The weighting parameter η is used to trade between J_P and J_u . Note that arbitrarily scaling the cost function F_1 will not change the result. Therefore, optimizing F_1 is equivalent to optimizing $F_1' = (1 - \eta)J_P + \eta J_u$, and this approach resembles the homotopy method described in Section 2.2.3. Setting $\eta = 0$ is equivalent to minimizing only covariance, and $\eta \approx 1$ is equivalent to minimizing only energy. The constraint \mathbf{g}_1 is only a function of the state and time at the beginning and end

of the trajectory.

$$\begin{aligned}
F_1 &= J_P + \frac{\eta}{(1-\eta)} J_u \quad \text{for } \eta \in [0, 1) \\
(\mathbf{X}^*, \mathbf{u}^*) &= \arg \min F_1 \\
\text{subject to } \mathbf{g}_1 &\left(\mathbf{X}^*(t_0), t_0, \mathbf{X}^*(t_f), t_f \right) = \mathbf{0}
\end{aligned} \tag{2.25}$$

Formulation 2. Covariance is minimized and energy is constrained; $J'_u(t_f)$ is the desired final energy cost. Therefore, F_2 only includes J_P . The constraint vector \mathbf{g}_2 includes \mathbf{g}_1 as well as terminal constraints on J_u . The J_u constraint can be varied to form a J_P vs. J_u Pareto front. The state is augmented with J_u , and $\widetilde{\mathbf{X}}(t) = [\mathbf{r}^\top, \mathbf{v}^\top, J_u]^\top$ is the augmented state vector. Thus, J_u can be constrained by simply adding a final state constraint. Note that the J_u state dynamics are $\dot{J}_u = \mathbf{u}^\top R \mathbf{u}$ from Eq. (2.24).

$$\begin{aligned}
F_2 &= J_P \\
(\widetilde{\mathbf{X}}^*, \mathbf{u}^*) &= \arg \min F_2 \\
\text{subject to } \mathbf{g}_2 &= [\mathbf{g}_1, J_u(t_0), J_u(t_f) - J'_u(t_f)] = \mathbf{0}
\end{aligned} \tag{2.26}$$

The cost functions and terminal constraints are different in each formulation. In general, Formulation 1 is simpler to implement. When compared to Formulation 2, Formulation 1 requires one fewer state and two fewer constraints. To explore the design space with Formulation 1, it is only necessary to vary the η parameter between zero and one in all cases. Conversely, some additional work may be required to determine the appropriate range of $J_u(t_f)$ constraints to investigate when working with Formulation 2. Moreover, feasible solutions are not guaranteed to exist for a particular value of $J_u(t_f)$, which may necessitate the use of an inequality constraint in Formulation 2. Regardless, Formulation 2 (ϵ -constraint) is more appropriate for multi-objective optimization; globally-optimal solutions to Formulation 2 are guaranteed to lie on a two-dimensional Pareto front of the two objectives. In contrast, the approach taken in Formulation 1 (weighted-sum) does not guarantee Pareto-optimal solutions in general.

Despite these distinctions, Theorem 1 in the following section will posit that, for the special case of covariance and energy optimization with control-linear noise, both formulations result in the same optimal control necessary conditions and produce the same optimal trajectories. This equivalence will be used to derive the Formulation 1 weighting parameters that produce Pareto-optimal solutions. The optimal control problem will be solved during the proof of Theorem 1.

2.4 Optimal Control Solution

First, note that cost functions F_1 and F_2 take the following forms:

$$F_1 = \text{trace}(\mathcal{W}\Phi(t_f, t_0)\Sigma(t_0)\Phi(t_f, t_0)^\top) + \int_{t_0}^{t_f} \mathbf{u}(t)^\top \left(\sigma^2 B^\top \Phi(t_f, t)^\top \mathcal{W}\Phi(t_f, t) B + \frac{\eta}{(1-\eta)} R \right) \mathbf{u}(t) dt, \quad (2.27)$$

$$F_2 = \text{trace}(\mathcal{W}\Phi(t_f, t_0)\Sigma(t_0)\Phi(t_f, t_0)^\top) + \int_{t_0}^{t_f} \mathbf{u}(t)^\top \sigma^2 B^\top \Phi(t_f, t)^\top \mathcal{W}\Phi(t_f, t) B \mathbf{u}(t) dt. \quad (2.28)$$

Both Eqs. (2.27) and (2.28) are of the Bolza type. Necessary and transversality conditions for indirect optimization were summarized previously in Section 2.2.1. Given these preliminaries, Theorem 1 is stated as follows.

Theorem 1. *Consider the nominal dynamics in Eq. (2.17) and the state error covariance in Eq. (2.20). The solutions to Formulations 1 and 2 are equivalent when*

$$\lambda_{J_u} = \frac{\eta}{1-\eta}, \quad (2.29)$$

where λ_{J_u} is the costate corresponding to J_u . The optimal control is

$$\mathbf{u}^*(t) = -\frac{1}{2} \left(\sigma^2 B^\top \Phi(t_f, t)^\top \Phi_p^\top \mathcal{W}\Phi_p \Phi(t_f, t) B + \frac{\eta}{1-\eta} R \right)^{-1} \lambda_v, \quad (2.30)$$

in which λ_v is the velocity costate. The position and velocity costates are equivalent in both formulations.

Proof of Theorem 1. First, it is convenient to express the Formulation 1 Hamiltonian, \mathcal{H}_1 , in terms of the time-varying matrix $\mathcal{R}_1(t)$ as follows:

$$\mathcal{H}_1 = \mathbf{u}(t)^\top \mathcal{R}_1(t) \mathbf{u}(t) + \lambda_r^\top \mathbf{v} + \lambda_v^\top (\mathbf{f}_v(\mathbf{X}'(t)) + \mathbf{u}(t)), \quad (2.31)$$

$$\mathcal{R}_1(t) = \sigma^2 B^\top \Phi(t_f, t)^\top \mathcal{W}\Phi(t_f, t) B + \frac{\eta}{(1-\eta)} R. \quad (2.32)$$

With the addition of the state J_u in Formulation 2, the Formulation 2 Hamiltonian is

$$\begin{aligned} \mathcal{H}_2 = & \mathbf{u}(t)^\top \sigma^2 B^\top \Phi(t_f, t)^\top \mathcal{W}\Phi(t_f, t) B \mathbf{u}(t) + \lambda_r^\top \mathbf{v} + \lambda_v^\top (\mathbf{f}_v(\mathbf{X}'(t)) + \mathbf{u}(t)). \\ & + \lambda_{J_u} \mathbf{u}^\top(t) R \mathbf{u}(t). \end{aligned} \quad (2.33)$$

As in Eq. (2.31), it is also convenient to define a time-varying matrix $\mathcal{R}_2(t)$ such that

$$\mathcal{H}_2 = \mathbf{u}(t)^\top \mathcal{R}_2(t) \mathbf{u}(t) + \lambda_r^\top \mathbf{v} + \lambda_v^\top (\mathbf{f}_v(\mathbf{X}'(t)) + \mathbf{u}(t)), \quad (2.34)$$

$$\mathcal{R}_2(t) = \sigma^2 B^\top \Phi(t_f, t)^\top W \Phi(t_f, t) B + \lambda_{J_u} R. \quad (2.35)$$

Throughout the remainder of the proof, terms that do not explicitly depend on the optimal control, the costates, or \mathcal{R}_i (e.g. \mathbf{v} , $\mathbf{f}_v(\mathbf{X})$, $\Phi(t_f, t)$, $\partial\Phi(t_f, t)/\partial\mathbf{X}$, and K) are assumed to be equivalent — this assumption will be valid when the optimal controls are shown to be equivalent. The transversality conditions will be addressed first. From Eqs. (2.5)-(2.8), the Formulation 1 transversality conditions are:

$$\boldsymbol{\lambda}_{\mathbf{X}}(t_0)^\top = -\frac{\partial K}{\partial \mathbf{X}_0} - \psi^\top \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_0}, \quad (2.36)$$

$$\boldsymbol{\lambda}_{\mathbf{X}}(t_f)^\top = \frac{\partial K}{\partial \mathbf{X}_f} + \psi^\top \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_f}, \quad (2.37)$$

$$\mathcal{H}_1(t_0) = \frac{\partial K}{\partial t_0} + \psi^\top \frac{\partial \mathbf{g}_1}{\partial t_0}, \quad (2.38)$$

$$\mathcal{H}_1(t_f) = -\frac{\partial K}{\partial t_f} - \psi^\top \frac{\partial \mathbf{g}_1}{\partial t_f}. \quad (2.39)$$

Alternatively, the Formulation 2 transversality conditions include an additional costate and two additional constraints for J_u :

$$[\boldsymbol{\lambda}_{\mathbf{X}}(t_0)^\top, \lambda_{J_u}(t_0)] = \left[-\frac{\partial K}{\partial \mathbf{X}_0}, -\frac{\partial K}{\partial J_u(t_0)} \right] - [\psi^\top, \psi_{J_u(t_0)}, \psi_{J_u(t_f)}] \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_0} & \frac{\partial \mathbf{g}_1}{\partial J_u(t_0)} \\ \frac{\partial J_u(t_0)}{\partial \mathbf{X}_0} & \frac{\partial J_u(t_0)}{\partial J_u(t_0)} \\ \frac{\partial J_u(t_f)}{\partial \mathbf{X}_0} & \frac{\partial J_u(t_f)}{\partial J_u(t_0)} \end{bmatrix}, \quad (2.40)$$

$$[\boldsymbol{\lambda}_{\mathbf{X}}(t_f)^\top, \lambda_{J_u}(t_f)] = \left[\frac{\partial K}{\partial \mathbf{X}_f}, \frac{\partial K}{\partial J_u(t_f)} \right] + [\psi^\top, \psi_{J_u(t_0)}, \psi_{J_u(t_f)}] \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_f} & \frac{\partial \mathbf{g}_1}{\partial J_u(t_f)} \\ \frac{\partial J_u(t_0)}{\partial \mathbf{X}_f} & \frac{\partial J_u(t_0)}{\partial J_u(t_f)} \\ \frac{\partial J_u(t_f)}{\partial \mathbf{X}_f} & \frac{\partial J_u(t_f)}{\partial J_u(t_f)} \end{bmatrix}, \quad (2.41)$$

$$\mathcal{H}(t_0) = \frac{\partial K}{\partial t_0} + [\psi^\top, \psi_{J_u(t_0)}, \psi_{J_u(t_f)}] \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial t_0} \\ \frac{\partial J_u(t_0)}{\partial t_0} \\ \frac{\partial J_u(t_f)}{\partial t_0} \end{bmatrix}, \quad (2.42)$$

$$\mathcal{H}(t_f) = -\frac{\partial K}{\partial t_f} - [\psi^\top, \psi_{J_u(t_0)}, \psi_{J_u(t_f)}] \begin{bmatrix} \frac{\partial \mathbf{g}_1}{\partial t_f} \\ \frac{\partial J_u(t_0)}{\partial t_f} \\ \frac{\partial J_u(t_f)}{\partial t_f} \end{bmatrix}. \quad (2.43)$$

Recognizing that the initial and final states and times are independent, Eqs. (2.42) and (2.43) are reduced to Eqs.(2.38) and (2.39), respectively. Moreover, Eqs. (2.40) and (2.41) are reduced to

$$[\boldsymbol{\lambda}_{\mathbf{X}}(t_0)^\top, \lambda_{J_u}(t_0)] = \left[-\frac{\partial K}{\partial \mathbf{X}_0} - \psi^\top \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_0}, -\psi_{J_u(t_0)} \right], \quad (2.44)$$

$$[\boldsymbol{\lambda}_{\mathbf{X}}(t_f)^\top, \lambda_{J_u}(t_f)] = \left[\frac{\partial K}{\partial \mathbf{X}_f} + \frac{\partial \mathbf{g}_1}{\partial \mathbf{X}_f}, \psi_{J_u(t_f)} \right]. \quad (2.45)$$

Thus, $\mathcal{H}(t_0)$, $\mathcal{H}(t_f)$, $\boldsymbol{\lambda}_{\mathbf{X}}(t_0)$, and $\boldsymbol{\lambda}_{\mathbf{X}}(t_f)$ are identical in both formulations. The next step of the proof is to compare the necessary conditions. From Eq. (2.3), it follows that the optimal control for either formulations is

$$\mathbf{u}^*(t) = -\frac{\mathcal{R}_i^{-1} \boldsymbol{\lambda}_v}{2}. \quad (2.46)$$

Substituting $\mathbf{u}^*(t)$ into \mathcal{H}_i ,

$$\mathcal{H}_i^* = \lambda_r^\top \mathbf{v} + \lambda_v^\top \mathbf{f}_v(\mathbf{X}'(t)) - \frac{1}{4} \lambda_v^\top \mathcal{R}_i^{-1} \lambda_v. \quad (2.47)$$

The costate dynamics for either formulation are derived from the necessary condition in Eq. (2.4):

$$\dot{\lambda} = -\frac{\partial}{\partial \mathbf{X}} (\lambda_r^\top \mathbf{v} + \lambda_v^\top \mathbf{f}_v(\mathbf{X}'(t))) + \frac{1}{4} \lambda_v^\top \frac{\partial \mathcal{R}_i^{-1}}{\partial \mathbf{X}} \lambda_v. \quad (2.48)$$

Employing a matrix identity for the derivative of an inverse matrix yields

$$\dot{\lambda} = -\frac{\partial}{\partial \mathbf{X}} (\lambda_r^\top \mathbf{v} + \lambda_v^\top \mathbf{f}_v(\mathbf{X}'(t))) - \frac{1}{4} \lambda_v^\top \mathcal{R}_i^{-1} \frac{\partial \mathcal{R}_i}{\partial \mathbf{X}} \mathcal{R}_i^{-1} \lambda_v. \quad (2.49)$$

From Eq. (2.46), it is clear that the optimal control is identical in both Formulation 1 and Formulation 2 if the velocity costates, λ_v , are identical and $\mathcal{R}_1(t) = \mathcal{R}_2(t)$ for all time. The transversality conditions have already shown that the costates are equivalent at the initial and final times. From Eq. (2.49), an additional condition is required for the costate dynamics (and the control, by extension) to be equivalent between Formulations 1 and 2: $\partial \mathcal{R}_1(t)/\partial \mathbf{X} = \partial \mathcal{R}_2(t)/\partial \mathbf{X}$ for all time. From Eqs. (2.32) and (2.35)

$$\begin{aligned} \frac{\partial \mathcal{R}_1}{\partial \mathbf{X}} &= \sigma^2 B^\top \left(\frac{\partial \Phi(t_f, t)}{\partial \mathbf{X}}^\top W \Phi(t_f, t) + \Phi(t_f, t)^\top W \frac{\partial \Phi(t_f, t)}{\partial \mathbf{X}} \right) B, \\ &= \frac{\partial \mathcal{R}_2}{\partial \mathbf{X}}, \end{aligned} \quad (2.50)$$

where $\partial \Phi(t_f, t)/\partial \mathbf{X}$ is computed from Eq. (2.22). Thus, $\partial \mathcal{R}_1(t)/\partial \mathbf{X} = \partial \mathcal{R}_2(t)/\partial \mathbf{X}$ holds and $\mathcal{R}_1(t) = \mathcal{R}_2(t)$ is the only remaining condition needed for the control to be equivalent between both formulations. The final step of the proof is to show that this is true given an appropriate selection of η . Recall that $\mathcal{R}_2(t)$ is a function of the energy costate λ_{J_u} . From Eq. (2.4),

$$\dot{\lambda}_{J_u}(t) = 0, \quad (2.51)$$

and λ_{J_u} is constant. From Eqs. (2.32) and (2.35), $\mathcal{R}_1(t) = \mathcal{R}_2(t)$ when $\lambda_{J_u} = \eta/(1 - \eta)$. It follows that $\mathbf{u}^*(t)$ is equivalent in Formulation 1 and Formulation 2 when $\lambda_{J_u} = \eta/(1 - \eta)$. This completes the proof.

■

In summary, the optimal control law for either Formulation 1 or 2 is given in Theorem 1. The costate dynamics are given in Eq. (2.49). It is important to note that both J_P and J_u are quadratic in $\mathbf{u}(t)$. This structural similarity is a special case that has enabled an equivalence between both formulations.

2.5 Optimization Algorithm

Given the statement of Theorem 1, Formulation 1 and Formulation 2 will produce the same optimal trajectories with the appropriate choice of η or J'_u , respectively. For example, consider a scenario in which Formulation 1 is implemented with a weighting parameter of η' and produces an optimal trajectory with an energy cost of J'_u . If the optimization was instead performed using Formulation 2 and an energy constraint of J'_u , the optimal $\lambda_{J_u}^*$ would be such that $\lambda_{J_u}^* = \eta'/(1 - \eta')$. As it is possible to locate the same trajectories with both formulations, Formulation 1 will be applied due to its reduced complexity.

The optimization algorithm is detailed in Algorithm 1. The algorithm assumes that the initial and final state and time are fully constrained. Optimal trajectories are located with backward single shooting methods, as detailed in Section 2.2.2. From Eq. (2.30), the optimal control is a function of the STM from the current time to the final time. The STM is computed by numerically integrating Eq. (2.21). If forward shooting (integrating from t_0 to t_f) is used, $\Phi(t_f, t)$ is not available at time t , because the trajectory from t to t_f has not yet been realized. To accumulate $\Phi(t_f, t)$, it is most convenient to use backward shooting. A fixed-step Runge-Kutta fourth order integration scheme is used for all numerical integration.

The Algorithm 1 procedure is summarized as follows: at each value of η , a maximum of `maxguess` initial guesses of $\lambda(t_f)$ are attempted. Guesses are randomly sampled from zero-mean Gaussian distributions with covariance Σ_{guess} . If an initial guess produces a constraint error below the maximum accepted value (`emax`), the algorithm proceeds to a differential corrector (Newton method). A maximum of `itmax` iterations are allowed by the differential corrector. During each iteration, the step size is iteratively scaled by a factor of `scale` until the constraint error improves or the step scale is smaller than ζ_{min} (gradients are not re-computed during the scaling process). If the constraint error is reduced below the desired tolerance (`tol`), the associated $\lambda(t_f)$ is considered a solution to the necessary conditions. The algorithm can be terminated if one solution is repeated multiple times.

Solutions to the necessary conditions produce stationary points of the cost function, and may be inflection points, local maxima, or local minima. Sufficient conditions can be used to characterize a stationary point as a local minima. Alternatively, the procedure in this paper is repeated for many initial guesses and

Algorithm 1: Find feasible solutions that satisfy the necessary conditions.

Data for dynamics:

$A(\mathbf{X}'(t))$, $\mathbf{f}(\mathbf{X}'(t))$, set of t_{coast} , set of t_f , \mathcal{W} , time history of initial orbit: $\mathbf{X}'(t)$, target final state: $\mathbf{X}'(t_f)$, set of $\eta \in [0, 1)$, σ .

Data for differential corrector (DC):

Constraint function: $\mathbf{g}(\mathbf{X}'(t_0))$, \mathbf{emax} , \mathbf{itmax} , $\mathbf{maxguess}$, $\mathbf{maxrepeat}$, \mathbf{rmin} , $\mathbf{scale} \in (0, 1)$, \mathbf{tol} , set of Σ_{guess} , ζ_{min} .

Result: A set of feasible solutions that satisfy the optimal control necessary conditions.

```

for all combinations  $t_f, t_{coast}, \eta, \Sigma_{guess}$  do
  initialize:  $\mathbf{guess} = 0$ ,  $\mathbf{repeat} = 0$ 
  while  $\mathbf{guess} < \mathbf{maxguess}$  and  $\mathbf{repeat} < \mathbf{maxrepeat}$  do
    update:  $\mathbf{guess} = \mathbf{guess} + 1$ 
    sample  $\lambda(t_f) \sim \mathcal{N}(\mathbf{0}, \Sigma_{guess})$ ;
    integrate  $[\mathbf{X}'(t), \lambda(t)]$  from  $t_f$  to  $t_0$ ;
    compute  $\mathbf{g}(\mathbf{X}'(t_0))$  and  $r_{min}$ ;
    initialize:  $\mathbf{iter} = 0$ ;
    while  $\mathbf{tol} < \|\mathbf{g}\| < \mathbf{emax}$  and  $\mathbf{iter} < \mathbf{itmax}$  do
      update:  $\mathbf{iter} = \mathbf{iter} + 1$ ;
      compute  $\frac{\partial \mathbf{g}}{\partial \lambda(t_f)}$ ;
      initialize:  $\|\mathbf{g}_{new}\| = \infty$  and  $\zeta = 1$ ;
      while  $\|\mathbf{g}_{new}\| > \|\mathbf{g}\|$  and  $\zeta > \zeta_{min}$  do
         $\lambda_{new}(t_f) = \lambda(t_f) - \zeta \left( \frac{\partial \mathbf{g}}{\partial \lambda(t_f)} \right)^{-1} \mathbf{g}$ ;
        integrate  $[\mathbf{X}'_{new}(t), \lambda_{new}(t)]$  from  $t_f$  to  $t_0$ ;
        compute  $\mathbf{g}_{new}(\mathbf{X}'_{new}(t_0))$  and  $r_{min}$ ;
        update:  $\zeta = \mathbf{scale} \times \zeta$ 
      end
      update:  $\mathbf{g} = \mathbf{g}_{new}$  and  $\lambda(t_f) = \lambda_{new}(t_f)$ ;
    end
    if  $\|\mathbf{g}\| \leq \mathbf{tol}$  and  $r_{min} > \mathbf{rmin}$  and solution is novel then
      | save feasible solution;
    else if solution is repeated then
      |  $\mathbf{repeat} = \mathbf{repeat} + 1$ ;
    end
  end
end

```

the costs of all solutions are compared. Partial derivatives for differential correction are computed with forward finite differences. However, analytic [27, 28] or automatic differentiation [29] is possible if improved speed or accuracy is required. The algorithm is easily parallelized.

2.6 Terminator Orbit Transfers

Algorithm 1 has been applied to optimize asteroid orbit maneuvers in the H3BP augmented with SRP and J_2 and J_3 spherical harmonics for the asteroid gravity field [21]. The H3BP system of equations and terminator orbits are described in Appendix B. Spacecraft and dynamical parameters are chosen to represent the OSIRIS-REx mission to the asteroid Bennu; all values are given in Table 2.1. Values are approximate and are not meant to reflect the most up-to-date values for Bennu or the OSIRIS-REx mission. Coefficients J_2 and J_3 are unnormalized in Table 2.1. The set of energy costates, λ_{J_u} , correspond to $\eta \approx \{9.9 \times 10^{-4}, 9.9 \times 10^{-3}, 9.1 \times 10^{-2}, 0.33, 0.50, 0.83, 0.90, 0.98, .99\}$ respectively. The control-linear noise factor, σ , dictates the level of control noise. Control noise is driven by engine hardware and operating conditions. Studies of electric propulsion devices have found that root-mean-square discharge current fluctuations can reach up to 13% the nominal magnitude [2]; $\sigma = 10^{-2}$ is selected for this application, corresponding to noise at 1% of the nominal.

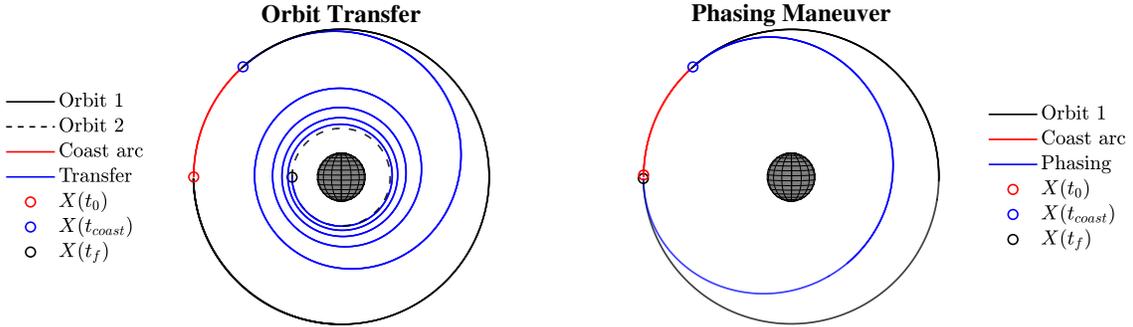
2.6.1 Maneuver Scenarios

Three maneuver scenarios are considered which involve two terminator orbits: Orbit 1 and Orbit 2. Orbits 1 and 2 are non-circular, but are designed to have radii approximately equal to $r_0 \approx 1.5$ km and $r_f \approx 500$ m, respectively. Although these orbits are not periodic, approximate orbit periods will be used as a time reference (e.g. $T_f = 2\pi(r_f^3/\mu)^{1/2}$ will be used as an approximate period for Orbit 2). The three maneuver scenarios are

- (1) orbit transfers from Orbit 1 to Orbit 2,
- (2) 180° phasing maneuvers on Orbit 1, and
- (3) proactive station-keeping for three orbits on Orbit 1.

Table 2.1: Terminator Orbit Parameters for Trajectory Optimization

| Parameter | Symbol | Value | Unit |
|--|--|---|--------------------------|
| Minimum accepted altitude | - | 50 | m |
| Error tolerance | - | 10^{-10} | - |
| Bennu distance from Sun [8] | d | 1.126 | AU |
| Second-order zonal harmonic [30] | J_2 | 0.039156 | - |
| Third-order zonal harmonic [30] | J_3 | -0.014843 | - |
| Initial position variance | $\Sigma_{xx}(t_0), \Sigma_{yy}(t_0), \Sigma_{zz}(t_0)$ | 1 | m^2 |
| Initial velocity variance | $\Sigma_{\dot{x}\dot{x}}(t_0), \Sigma_{\dot{y}\dot{y}}(t_0), \Sigma_{\dot{z}\dot{z}}(t_0)$ | 1 | mm^2/s^2 |
| $\Sigma(t_0)$ off-diagonal terms | - | 0 | - |
| SRP at 1 AU | p_0 | 4.5×10^{-6} | N/m^2 |
| Energy weight matrix | R | $I_{3 \times 3}$ | - |
| Asteroid mean radius [30] | r_b | 246 | m |
| Approx. large orbit radius | r_0 | 1.5 | km |
| Approx. small orbit radius | r_f | 500 | m |
| Weight matrix for $\Phi_p \Sigma(t_f) \Phi_p^\top$ | \mathcal{W}' | $10^{-10} \Sigma(t_0)^{-1}$ | - |
| Spacecraft mass-to-area ratio [8] | Z | 62 | kg/m^2 |
| Energy costates, $\eta/(1-\eta)$ | λ_{J_u} | $\{10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1, 5, 10, 50, 100\}$ | - |
| Gravitational parameter [31] | μ | 4.892 | m^3/s^2 |
| Spacecraft reflectance [8] | ρ | 0.4 | - |
| Control noise factor | σ | 10^{-2} | - |
| Integration time step | $\Delta\tau$ | 5×10^{-2} | - |

Figure 2.1: Two mission scenarios: an orbit transfer and a phasing maneuver. $\mathbf{X}(t_0)$, $\mathbf{X}(t_{coast})$, and $\mathbf{X}(t_f)$ are fixed for each initialization of the optimization.

Proactive station-keeping describes a scenario in which the spacecraft is already on the nominal orbit (in expectation, but with some uncertainty). It is desired for the spacecraft to remain on the nominal orbit after some time (in this case, three approximate orbit periods). Therefore, the minimum energy solution is simple — no control is required. However, the nominal orbit exhibits rapid uncertainty growth. Rather than risking large errors along the nominal orbit, it is possible that applying some control to alter the orbit geometry can

proactively reduce covariance inflation.

As illustrated in Figure 2.1, each scenario assumes an identical initial state and covariance at $t_0 := 0$. The maneuver begins after a coast arc of fixed duration, t_{coast} . The final time and state, t_f and $\mathbf{X}'(t_f)$, are also fixed during each initialization of the optimization. The algorithm is repeated for different combinations of t_{coast} and t_f . For each application, the value of t_{coast} will be varied between zero and t_f . In each case, J_P is configured to minimize the trace of the covariance after propagating for an additional orbit period beyond the control cut-off (i.e. Φ_p is the STM for one approximate period of the final orbit). The additional propagation prevents control spikes at the final time. Trajectories that encroach below a minimum accepted radius of 50 m are discarded.

2.6.2 Algorithm Inputs

Inputs to Algorithm 1 are summarized in Table 2.2. Each solution is evaluated for uniqueness. If the difference between two $\lambda(t_f)$ solutions has a norm greater than `utol`, the solutions are considered unique; otherwise, the solutions are considered repeated.

For each mission scenario, multiple solutions have been found (using Algorithm 1) that satisfy the necessary conditions for multi-objective optimization of covariance and energy. Figure 2.2 shows the total number of solutions for each maneuver scenario and value of λ_{J_u} . Among these solutions, the differential corrector required averages of 9, 10, and 13 $\lambda(t_f)$ iterations (quantity `iter` in Algorithm 1) for orbit

Table 2.2: Algorithm 1 Inputs

| Parameters (all nondimensional) | Symbol | Value |
|--|-----------------------|---------------------------------|
| Maximum error allowed to proceed to differential corrector | <code>emax</code> | 15 |
| Maximum number of initial guesses | <code>maxguess</code> | 40 |
| Maximum number of differential corrector steps | <code>itmax</code> | 30 |
| Minimum radius allowed for a feasible solution | <code>rmin</code> | 1.2 |
| Step scale reduction factor | <code>scale</code> | 0.7 |
| Maximum number of repeated solutions | <code>repeat</code> | 3 |
| Error tolerance | <code>tol</code> | 10^{-10} |
| Uniqueness tolerance | <code>utol</code> | 10^{-8} |
| Covariance matrices for initial guess | Σ_{guess} | $\rho^2 I_{6 \times 6}$ |
| Standard deviations of initial guess | ρ | $\{10^{-6}, 10^{-4}, 10^{-2}\}$ |
| Minimum step scale | ζ_{min} | 10^{-2} |

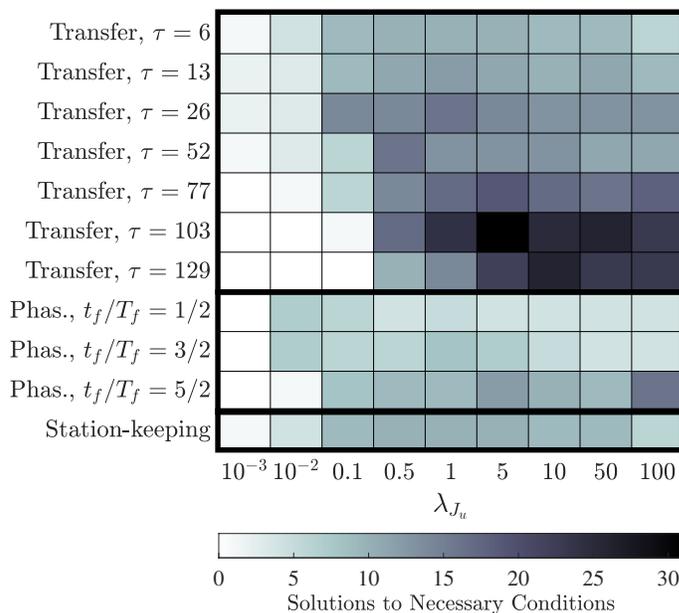


Figure 2.2: Total number of solutions found.

transfers, phasing maneuvers, and proactive station-keeping, respectively. Roughly 8% of initial guesses converged to feasible solutions. In the following subsections, all solutions are plotted in J_p vs. J_u vs. τ_f space (where τ_f is nondimensional time).

2.6.3 Scenario 1: Orbit Transfer

Figure 2.3 shows all trajectories that satisfy necessary conditions for the orbit transfer scenario plotted in J_p vs. J_u vs. τ_f space. A J_p vs. J_u Pareto-front is formed by the set of solutions from which J_p cannot be decreased without an increase in J_u , and vice versa. An example Pareto-optimal trajectory is identified in Figure 2.3, which will be compared alongside the minimum covariance and minimum energy trajectories in Figure 2.4. Figure 2.4 shows the three trajectories in position space and their corresponding covariance ellipses. Covariances are propagated for an additional orbit after the control cutoff (as was done in the optimization). The minimum covariance solution maintains a reasonable energy cost that is equivalent to a ΔV of 7.70 cm/s, whereas the minimum energy solution requires a ΔV of 4.24 cm/s. The control profiles are plotted in Figure 2.5.

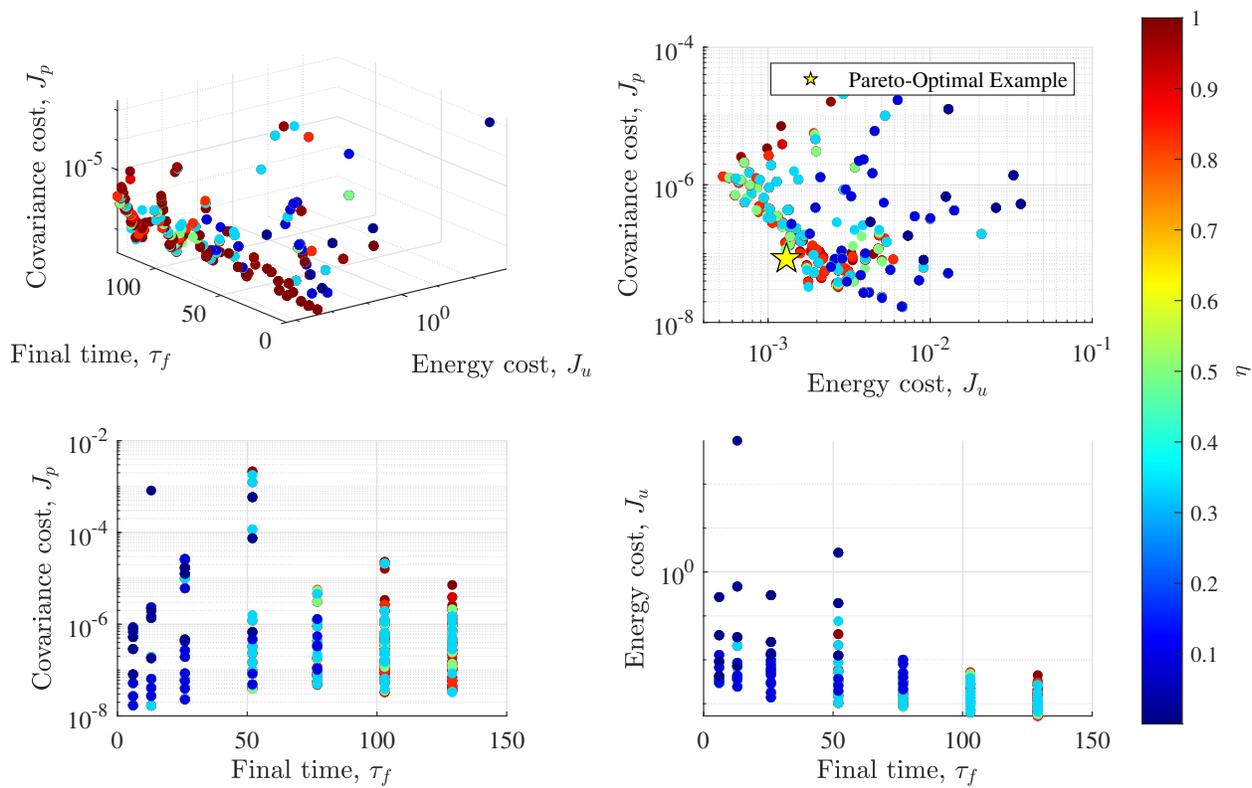


Figure 2.3: Transfers from Orbit 1 to Orbit 2 that satisfy the necessary conditions; J_p , J_u , and τ_f are nondimensional.

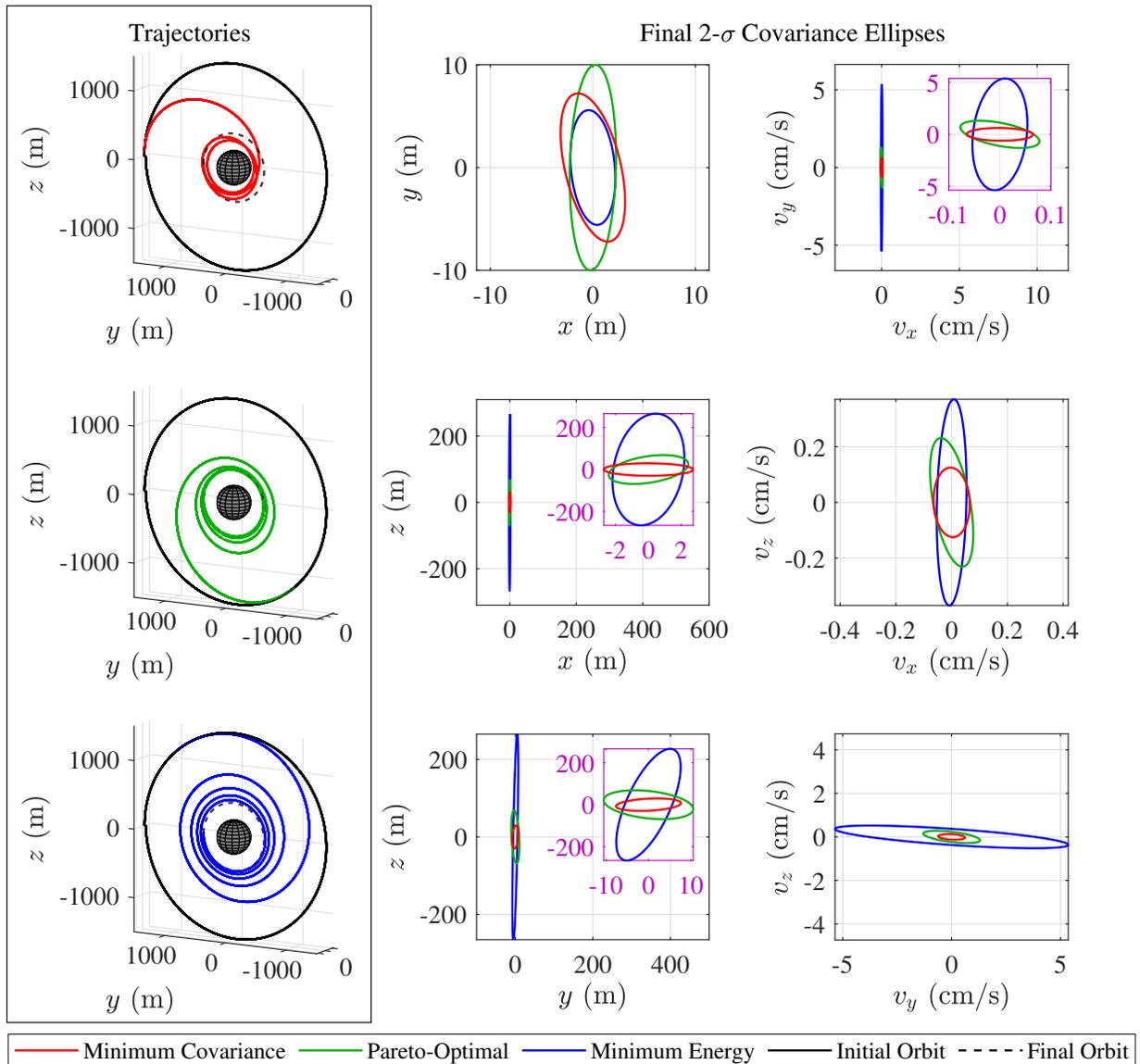


Figure 2.4: Three Pareto-optimal transfers are plotted (left), and their final 2- σ covariance ellipses are compared (right). Magenta boxes use dissimilar axis scaling for visibility.

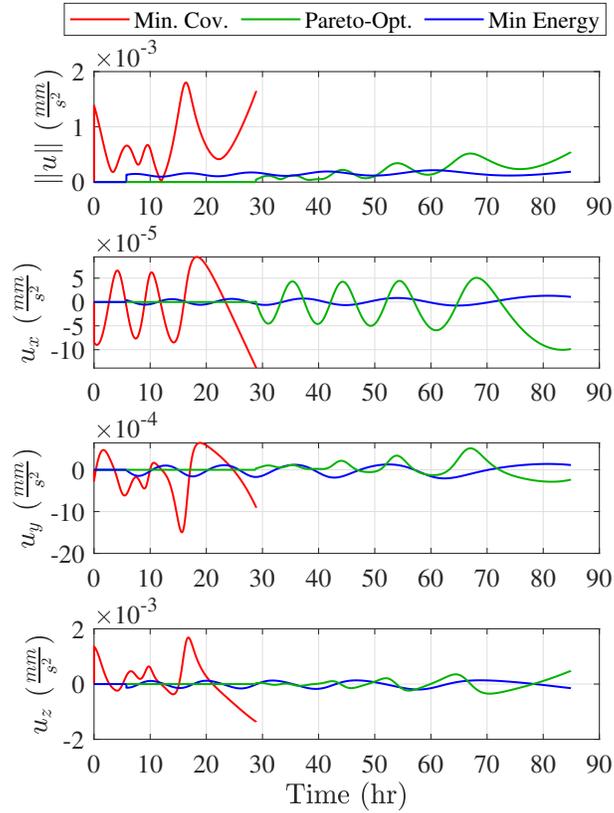


Figure 2.5: Control profiles for three Pareto-optimal orbit transfers. The minimum covariance solution was found to have a shorter transfer duration, and is therefore terminated sooner.

2.6.4 Scenario 2: 180° Phasing Maneuver

The target final state is chosen by integrating the nominal initial state $\mathbf{X}(t_0)$ for the approximate orbit period of T_f (for this scenario, T_f is the period of a circular orbit with radius r_0). Thus, 180° phasing is accomplished by reaching the final state at a final time of either $1/2$, $3/2$, or $5/2$ T_f . Figure 2.6 shows all trajectories that satisfy the necessary conditions plotted in J_P vs. J_u vs. t_f/T_f space. A Pareto-optimal example trajectory is selected in Figure 2.6. The minimum covariance, minimum energy, and Pareto-optimal example trajectories and their final covariance ellipses are plotted in Figure 2.7. The minimum covariance and minimum energy phasing maneuvers require ΔV 's of 2.79 cm/s and 1.20 cm/s, respectively.

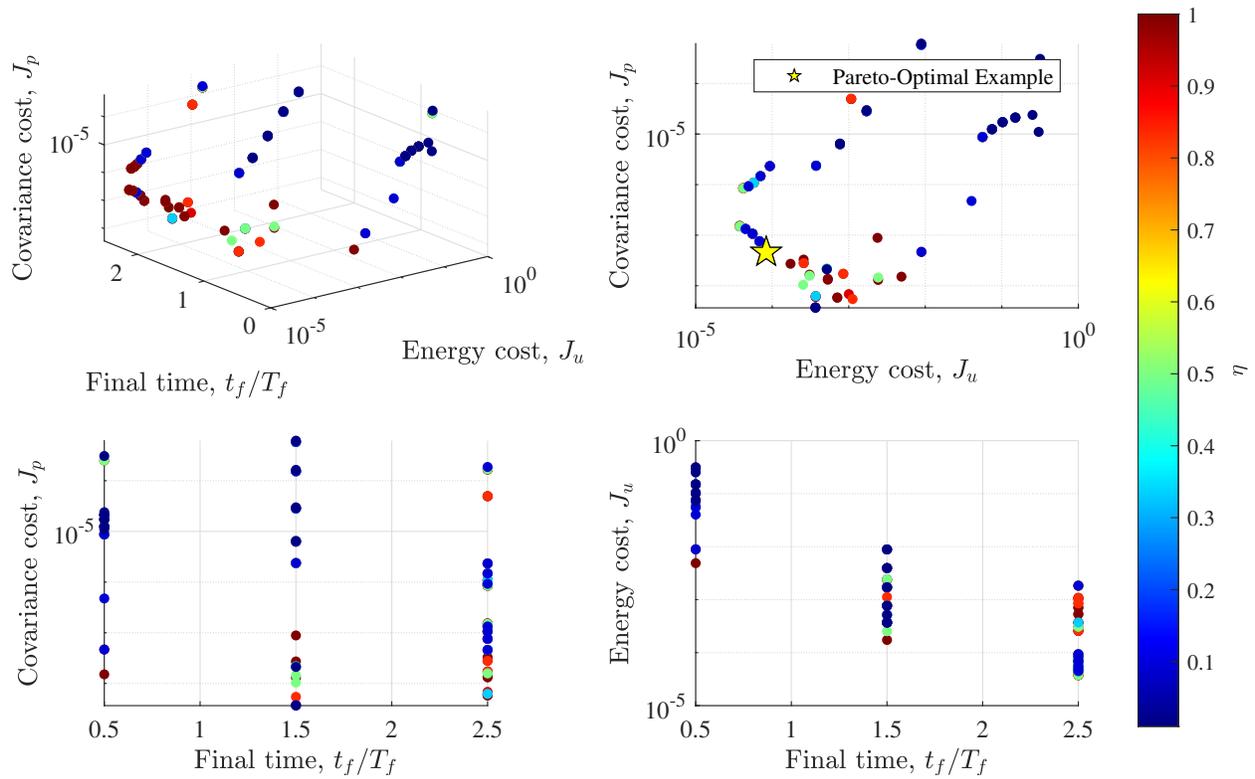


Figure 2.6: All 180° phasing maneuvers that satisfy the necessary conditions are plotted in J_p vs. J_u vs. t_f/T_f space, where T_f denotes the approximate period of Orbit 1.

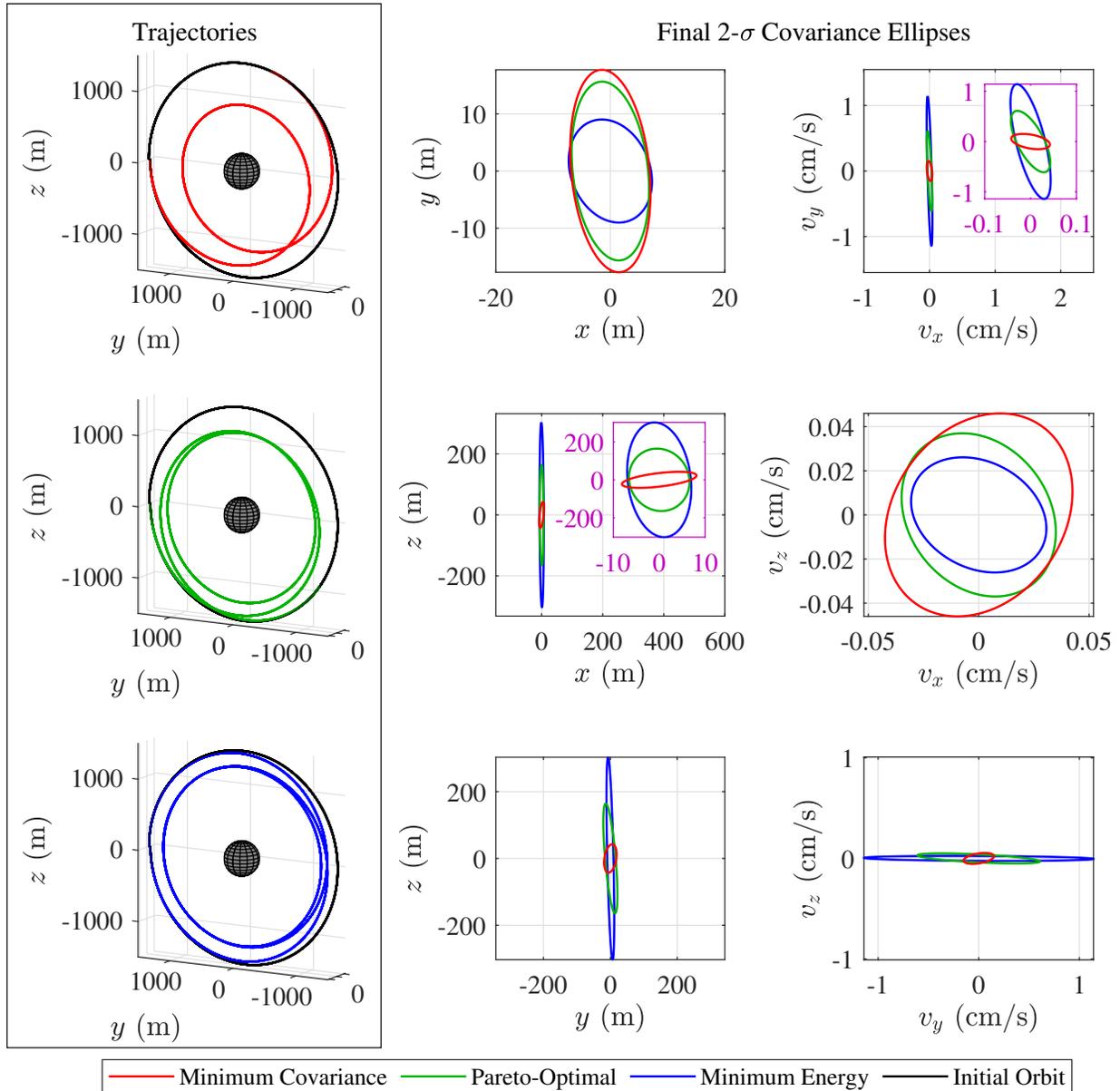


Figure 2.7: Three Pareto-optimal 180° phasing maneuvers are plotted (left), and their final $2\text{-}\sigma$ covariance ellipses are compared (right). Magenta boxes use dissimilar axis scaling for visibility.

2.6.5 Scenario 3: Proactive station-keeping

The proactive station-keeping scenario is equivalent to a phasing maneuver of zero degrees. The target final state is selected by integrating the nominal initial state $\mathbf{X}'(t_0)$ for three approximate orbit periods: $t_f = 3T_f$. The minimum energy solution is to do nothing — no control is required to reach the final state. However, the J_P vs. J_u results show that it is possible to minimize covariance at the expense of J_u . Two Pareto-optimal examples are selected in Figure 2.8. Both example trajectories, as well as the minimum covariance trajectory, are plotted in Figure 2.9 along with their final covariance ellipses. The minimum energy trajectory provides significant covariance improvement at the ΔV cost of 2.53 cm/s.

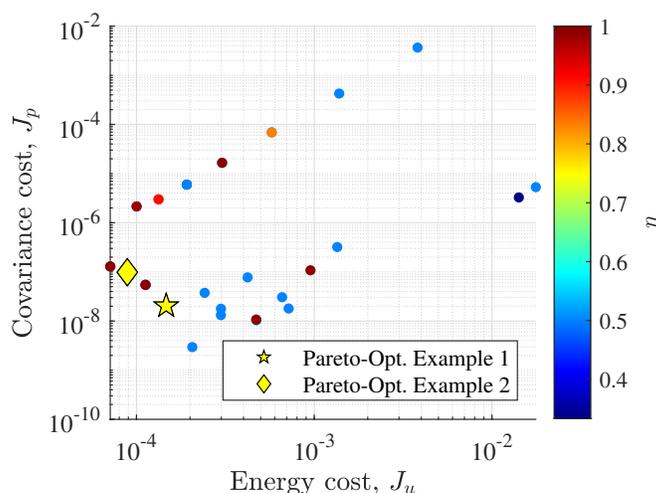


Figure 2.8: All proactive station-keeping maneuvers that satisfy the necessary conditions are plotted in nondimensional J_p vs. J_u space. All solutions have a final time of $t_f = 3T_f$.

2.6.6 Monte Carlo Validation

Finally, Monte Carlo simulations are used to compare the linear covariance dynamics to a true non-linear covariance propagation. Only results for the proactive station-keeping scenario are included here, but other scenarios were found to produce similar results. Specifically, the minimum covariance result in Figure 2.10 below corresponds to the top left trajectory in Figure 2.9. For both minimum covariance and minimum energy solutions, 2000 noisy trajectories were simulated by randomly sampling the initial state and approximating the control noise with a zero-order hold. The control noise standard deviation was adjusted

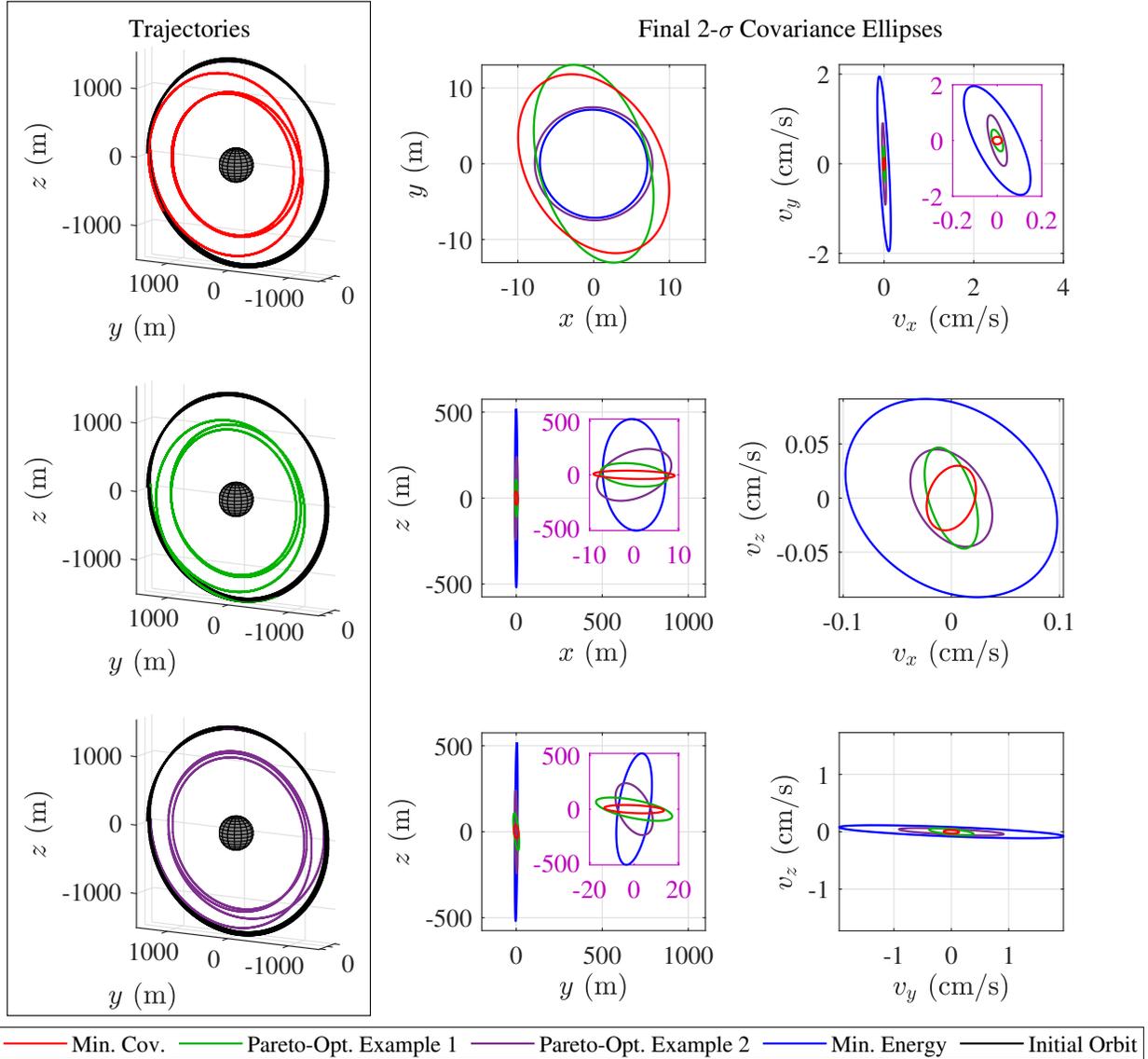


Figure 2.9: Three Pareto-optimal proactive station-keeping maneuvers are plotted (left), and their final 2- σ covariance ellipses are compared (right). Magenta boxes use dissimilar axis scaling for visibility.

appropriately given the relationship between continuous and discrete white noise:

$$\frac{\sigma^2 \delta_{Kronecker}}{\Delta\tau} = \sigma^2 \delta_{Dirac} \quad (2.52)$$

(i.e. in discrete time the control noise is randomly sampled with a standard deviation of $\sigma^2/\Delta\tau$, where $\Delta\tau$ is the integration time step) [32]. The nonlinear covariance is simply computed by calculating the covariance of all noisy final states. The linear propagation and the nonlinear propagation are compared in Figure 2.10. For

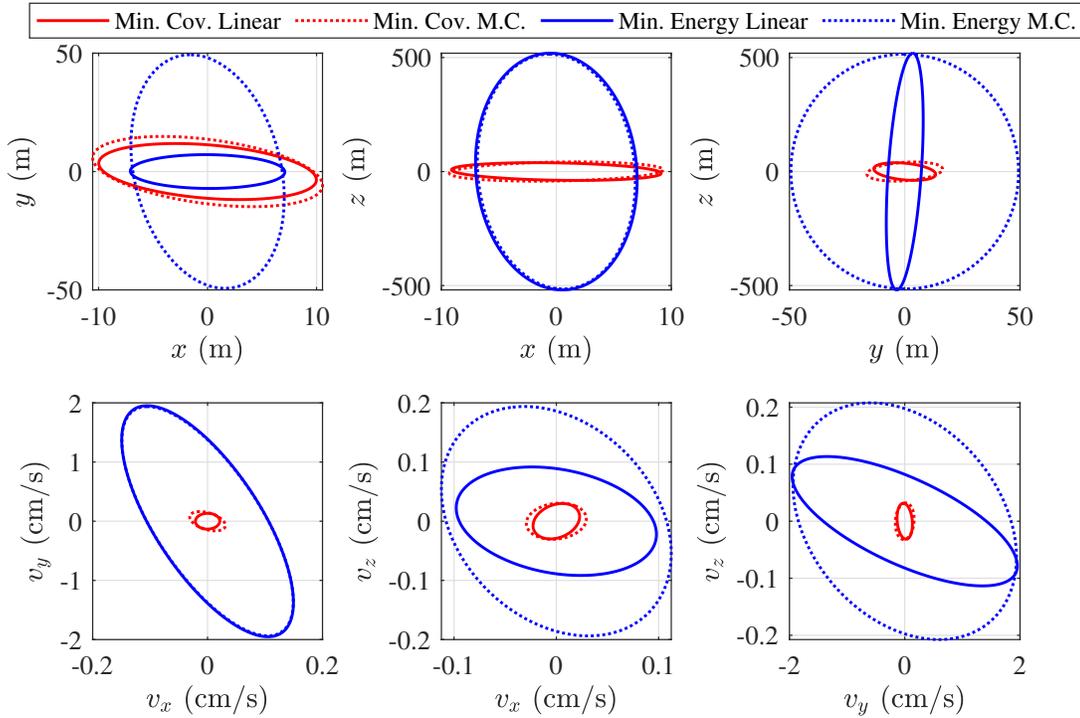


Figure 2.10: Comparing linear and nonlinear (Monte Carlo, $N = 2000$) $2\text{-}\sigma$ covariance propagation for proactive station-keeping maneuvers.

the minimum covariance trajectory, the linear propagation appears to adequately capture the true dynamics. However, the minimum energy trajectory appears to be more nonlinear. The trajectories in Figure 2.10 have the same initial and final times, so their sizes cannot be attributed to a difference in simulation time.

2.7 Discussion

In all three asteroid orbiter scenarios, the results indicate that significant covariance reduction is possible with relatively low energy costs. When compared to minimum energy solutions, minimum covariance solutions are achievable with ΔV increases of only centimeters per second (thus, minimum covariance optimization is well-motivated by asteroid missions). However, the difference between minimum covariance and minimum energy solutions does correspond to energy increases of roughly 50% or more. Therefore, for applications that are more propellant-expensive than the asteroid environment, a different Pareto-optimal trajectory is likely more appropriate than the minimum covariance trajectory. Overall, trajectories with

smaller final uncertainty are structurally similar — covariance reduction appears to correspond to lower average radii and more revolutions over the course of a maneuver. Thus, lower altitude maneuvers produce more favorable uncertainty dynamics than higher altitude maneuvers.

This behavior is even seen in the proactive station-keeping scenario, in which no control is required for the nominal initial state to reach the final target state. However, the unforced uncertainty propagation along the large orbit (Orbit 1) results in a significant amount of covariance growth as seen in Figure 2.9. Alternatively, lowering the orbit radius improves covariance performance. This example demonstrates how maneuvering can reduce uncertainty despite the addition of control noise. Here it is instructive to revisit Eq. (A.9). The term $\sigma^2 B \mathbf{u}(t) \mathbf{u}^\top(t) B^\top$ is guaranteed to be a positive definite contribution to the covariance dynamics. However, $A(t) \Sigma(t)$ and its transpose are not positive definite. Therefore, it is possible for a reduction in $A(t) \Sigma(t)$ (through applied control) to outweigh the influence of $\sigma^2 B \mathbf{u}(t) \mathbf{u}^\top(t) B^\top$. This covariance improvement is the result of complex interactions between $\mathbf{u}(t)$, $A(t)$, and $\dot{\Sigma}$. These complexities make the covariance behavior difficult to predict qualitatively, further establishing the need for a rigorous optimization procedure. Additional work is required to determine when proactive station-keeping maneuvers are propellant-saving, likely through the application of brute force Monte Carlo. Proactive maneuvering could also be used to reduce the need for human-in-the-loop when retroactive maneuvers cannot be planned autonomously.

Monte Carlo results in Figure 2.10 indicate that the linear regime is sufficient for covariance propagation about the minimum covariance solution. However, the minimum energy trajectory is more nonlinear. For the results in Figure 2.10, both the minimum energy and minimum covariance trajectories are constrained to have the same final time. Thus, the nonlinearity of the minimum energy solution is not due to a longer final time. Monte Carlo simulations for other scenarios produced similar results. This suggests that minimizing covariance also corresponds to reduced nonlinearity. Additional work is required to investigate this relationship and its advantages. Because uncertainty characteristics of minimum covariance trajectories are well captured by the linear regime, these trajectories may be more conducive to linear filtering and control algorithms (e.g. Kalman filters or linear-quadratic controllers). This phenomenon is explored further in Chapter 6. Additional future work includes incorporating a higher-order covariance propagation to cap-

ture nonlinearity as trajectories deviate from the minimum covariance solution. A higher-order propagation can be achieved by maintaining higher-order terms in the Taylor series expansion (TSE) of the state error dynamics.

With some modification, the method in this chapter could also accommodate additive dynamical noise due to stochastic accelerations. For example, Eq. (2.15) could be modified as follows:

$$d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t)dt + B\mathbf{u}(t)(dt + \sigma dW_t) + G(t)d\boldsymbol{\nu}_t, \quad (2.53)$$

where $\boldsymbol{\nu}_t$ is a vector of Wiener processes. Then, the final covariance would become:

$$\begin{aligned} \Sigma(t_f) &= \Phi(t_f, t_0)\Sigma(t_0)\Phi(t_f, t_0)^\top + \sigma^2 \int_{t_0}^{t_f} \Phi(t_f, t)B\mathbf{u}(t)\mathbf{u}(t)^\top B^\top \Phi(t_f, t)^\top dt \\ &+ \int_{t_0}^{t_f} \Phi(t_f, t)G(t)G(t)^\top \Phi(t_f, t)^\top dt \end{aligned} \quad (2.54)$$

In [13], uncertain gravitational parameters are modeled by additive noise — this simplification may be sufficient in some cases. Alternatively, uncertain parameters can be included in the state vector as nondynamic “states” with some initial uncertainty. Additive noise and uncertain parameters are left for the future work.

2.8 Conclusions

This chapter developed a method for multi-objective energy and covariance optimization. Linear covariance propagation was used with initial state uncertainty and control-dependent noise. Two optimal control formulations were considered: 1) a combined covariance and energy cost function (weighted-sum method) and 2) a covariance cost function with an energy constraint (ϵ -constraint method). In general, the first method is easier to apply, but the second is more rigorous for multi-objective optimization. However, a theorem was proven to demonstrate that the two formulations are equivalent for the case of covariance and energy optimization with control-dependent noise and a linear covariance propagation. The theorem enabled the simpler method (Formulation 1) to be implemented without loss of optimality. The optimization was demonstrated for asteroid orbit maneuvers in the H3BP with SRP and J_2 and J_3 zonal harmonics. Three maneuver scenarios were optimized: orbit transfers, phasing maneuvers, and proactive station-keeping.

In proactive station-keeping, control is applied to alter a nominal trajectory in a way that produces favorable uncertainty dynamics. In each scenario, covariance-and-energy Pareto fronts indicate that signif-

icant covariance reduction is possible with feasible penalties in energy (corresponding to centimeters per second of ΔV in the asteroid environment). Moreover, Monte Carlo simulations demonstrated that linear covariance propagation was sufficient to capture the uncertainty of the minimum covariance trajectories designed in this chapter.

Chapter 3

Continuous Closed-Loop Robust Control

Chapter 3 Nomenclature

| | | |
|-----------------|---|---|
| A | = | LTI dynamics matrix |
| B | = | LTI control matrix |
| I | = | identity matrix |
| J | = | cost function |
| $K_k(t)$ | = | state feedback gain after k th measurement |
| $L_k(t)$ | = | target state feedback gain after k th measurement |
| m | = | dimension of control input |
| n | = | dimension of system state |
| N | = | number of measurements |
| R | = | positive definite matrix |
| t | = | time |
| t_k | = | time of k th measurement |
| t_f | = | final time |
| $\mathbf{u}(t)$ | = | control |
| V_k | = | value function from t_k to t_f |
| dW_t | = | scalar Wiener process |
| \mathbf{x}_0 | = | known initial state |

| | | |
|----------------|---|--|
| \mathbf{x}_t | = | stochastic system state at time t |
| \mathbf{y}_t | = | perfect, full-state measurement |
| \mathbf{z}_f | = | target state |
| Σ | = | state error covariance |
| σ | = | control-linear noise factor |
| τ_k | = | time between measurements, $t_{k+1} - t_k$ |
| Φ | = | state transition matrix from initial to final time, e^{At_f} |
| Φ_k | = | state transition matrix between measurements, $e^{A\tau_k}$ |

3.1 Introduction

A theory for closed-loop control of sampled linear systems with control-linear noise is developed in this chapter. This chapter sets the foundation for Chapter 4, in which linear control laws are applied for neighboring guidance about nonlinear spacecraft trajectories. The term “sampled” refers to a control system that only receives a finite number of measurements, i.e., the controller does not have access to continuous state information. This setup may also be described as intermittent control, a medium between discrete and continuous control wherein control policies are only updated at specific intervals. For example, if state measurements are sampled and controls are only updated at the measurement times, the controls are effectively open-loop during the period between two measurements. This formulation is also appropriate for spacecraft that cannot continuously prioritize navigation and receive limited navigation updates.

This chapter focuses on a special class of continuous-time, stochastic linear systems with control-linear (or control-multiplicative) noise. The control-linear noise model was motivated in Chapter 1. In short, it has been observed that higher levels of commanded thrust cause larger thrust errors. This chapter will develop a linear state feedback control law to minimize mean squared errors about a target state in the presence of control-linear noise.

Optimal feedback control that is linear in states or measurement outputs is certainly not new. This is a common property shared by solutions to a broad class of optimal control problems, including the clas-

sical LQR and the linear quadratic Gaussian (LQG) control problems. Moreover, many existing works on stochastic systems with control-dependent noise address problems of feedback stabilization, i.e., problems of minimizing certain (quadratic) cost functions in both the state and control. Often, an assumption is made that the controller can access noiseless, full-state measurements at any time so that the optimal feedback control laws are functions of the current states. These works include [33], McLane’s seminal work in the 1970s [34, 35], and many more [6, 36–40].

However, the problem addressed in this chapter is different from the aforementioned work, with the first difference being the inclusion of a target state. Rather than feedback stabilizing the system at the origin, a control law will be derived to steer the system from an initial state to a final state that is as close as possible to a given (but arbitrary) target state. Note that if the target state happens to be the origin, then the problem is reduced to the problem of feedback stabilization. “Closeness” will be measured by the mean squared deviation of the final state from the target state. This problem is similar to the problem of stochastic reachability [41, 42] — the technical core of the reachability analysis is the minimization of mean squared state deviations. However, for the control-dependent case, the authors of [41, 42] assumed that the controller can access noiseless, full-state measurements at any time. Unlike in the previous work on feedback stabilization and stochastic reachability, this chapter assumes a more realistic measurement process where only a finite number of sampled measurements are available over a given time period.

The main contribution of this chapter is to provide a complete solution to the aforementioned optimal control problem using dynamic programming. In dynamic programming, a complex optimization problem is broken into recursive, manageable sub-problems at different instances in time. Despite the fact that the noise is control-linear and explicit solutions are often hard to obtain beyond the class of linear-quadratic Gaussian (LQG) problems, an explicit formula for the optimal feedback control law will be derived in this chapter. In particular, the control law will be linear in the initial state, the target state, and the sampled measurement outputs. The matrix-valued feedback state gains are time-varying, piecewise differentiable and can be computed offline by the algorithm provided in this chapter.

The remainder of the chapter is organized as follows: the problem is formulated in Section 3.2. The main result is presented as a Theorem in Section 3.3. A proof of the Theorem is also provided in

Section 3.3. Finally, numerical examples are given in Section 3.4, wherein it is demonstrated that the mean squared deviation decays to a limiting value as the number of measurement outputs approaches infinity.

3.2 Continuous Closed-Loop Control Problem Formulation

This chapter addresses linear time-invariant stochastic systems with control-linear noise. Let $\mathbf{x}_t \in \mathbb{R}^n$ be the state of the system at time t , $\mathbf{u}(t) \in \mathbb{R}^m$ be the nominal control input, and \mathbf{y}_{t_k} be the sampled measurement outputs at time t_k . The system dynamics are given by

$$\begin{cases} d\mathbf{x}_t &= A\mathbf{x}_t dt + B\mathbf{u}(t)(dt + \sigma dW_t), \\ \mathbf{y}_{t_k} &= \mathbf{x}_{t_k} \quad 0 < t_1 < t_2 < \dots, \end{cases} \quad (3.1)$$

where A and B are $n \times n$ and $n \times m$ matrices, respectively, and W_t is a scalar Wiener process. The constant σ regulates the standard deviation of the control noise (i.e. $\sigma = 0.1$ corresponds to a standard deviation of 10% of the control magnitude). The initial state \mathbf{x}_0 at time $t = 0$ is assumed to be known to the controller. The full-state measurement outputs \mathbf{y}_{t_k} are noiseless and the measurement times t_k are scheduled a priori.

A feedback control law is desired to steer the system in Eq. (3.1) from a known initial state \mathbf{x}_0 to a final state \mathbf{x}_{t_f} such that the squared deviation of \mathbf{x}_{t_f} from a target state \mathbf{z}_f is minimized in expectation. It is assumed that there are N measurement outputs $\mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_N}$, with $0 < t_1 < \dots < t_N < t_f$, available over the period $[0, t_f]$. At any time $t \in [0, t_f]$, the control $\mathbf{u}(t)$ is allowed to depend on all past measurements $\mathbf{y}_{t_k} (= \mathbf{x}_{t_k})$, for all $t_k < t$, as well as the initial state \mathbf{x}_0 . Then, the optimization problem can be stated as follows.

Continuous Closed-Loop Control Problem: Consider the stochastic control system in Eq. (3.1). Given an initial state \mathbf{x}_0 , a target state \mathbf{z}_f at a final time t_f , and N measurement outputs \mathbf{y}_{t_k} , find the optimal control law $\mathbf{u}^*(t)$ over $[0, t_f]$ that minimizes the following cost function:

$$J := \mathbb{E} \left[\|\mathbf{x}_{t_f} - \mathbf{z}_f\|^2 + \int_0^{t_f} \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau \middle| \mathbf{x}_0 \right], \quad (3.2)$$

where R is an $m \times m$ positive definite matrix. The first term penalizes deviation of the final state from the target, and the second term penalizes energy consumption over the period. The positive-definiteness of R

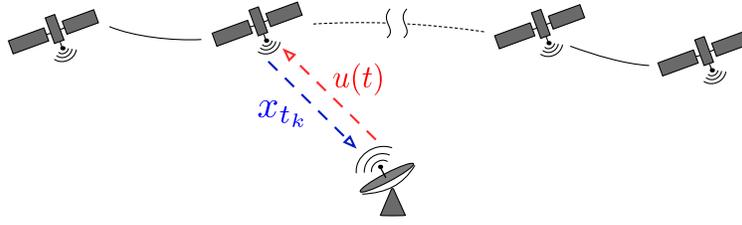


Figure 3.1: Illustration of the system model in Eq. (3.1). The controller can access sampled measurements, and the goal is to generate feedback control inputs that steer the system as close as possible to a target state. The graphic depicts communication with a ground station, but an onboard controller is also possible.

is a regularization condition which guarantees that the optimal control $\mathbf{u}^*(t)$ will be unique. The condition can be relaxed in some cases, as will be discussed later.

Notation: Throughout this chapter, $\tau_k := t_{k+1} - t_k$ is the time between two consecutive measurements. The STM from the measurement at t_k to the measurement at t_{k+1} is denoted $\Phi_k := e^{A\tau_k}$. The STM from the initial time, $t_0 = 0$, to the final time, t_f is denoted by $\Phi := e^{At_f}$. A superscript “*” indicates an optimized quantity.

3.3 Continuous Closed-Loop Control Result

A complete solution to the optimization problem in Eq. (3.2) is provided in this section. The final result is summarized in Theorem 2. Like the LQR or LQG problems, the optimal feedback control law will be linear in the initial state \mathbf{x}_0 , the target state \mathbf{z}_f , and the measurements \mathbf{y}_{t_k} . Moreover, the state feedback gains can be computed offline using Algorithm 2 given below. The offline computation is analogous to the standard LQR/LQG problem in which one computes the solution of the differential Riccati equation to obtain the state feedback gains.

Theorem 2. Consider the stochastic system in Eq. (3.1) and the optimal control problem in Eq. (3.2). The optimal control law $\mathbf{u}^*(t)$ over the time period $[0, t_f]$ is given by

$$\mathbf{u}^*(t) = K_k(t)\mathbf{y}_{t_k} + L_k(t)\mathbf{z}_f, \quad \text{for } t \in [t_k, t_{k+1}), \quad (3.3)$$

for all $k = 0, \dots, N$, where \mathbf{y}_0 is identified with \mathbf{x}_0 . The minimized cost function is a quadratic form given

by:

$$J^* = \mathbf{x}_0^\top W_0 \mathbf{x}_0 - 2\mathbf{x}_0^\top C_0 \mathbf{z}_f + \mathbf{z}_f^\top F_0 \mathbf{z}_f. \quad (3.4)$$

All the matrices $K_k(t)$, $L_k(t)$, W_0 , C_0 , and F_0 in the above expressions are computed in Algorithm 2.

Algorithm 2: Offline computation of state feedback gains for optimal continuous control law.

Data: Matrices A , B , R , scalar σ , measurement times t_1, \dots, t_N , and final time $t_f =: t_{N+1}$.

Result: Matrices $K_k(t)$ and $L_k(t)$ for all $k = 0, \dots, N$.

initialize;

$$W_{N+1} := I; \quad F_{N+1} := I; \quad C_{N+1} := I.$$

for $k := N$ **to** 0 **do**

$$Q_k(t) := R + \sigma^2 B^\top e^{A^\top(\tau_k - t)} W_{k+1} e^{A(\tau_k - t)} B;$$

$$M_k := \int_0^{\tau_k} e^{A(\tau_k - \tau)} B Q_k^{-1}(\tau) B^\top e^{A^\top(\tau_k - \tau)} d\tau;$$

$$W_k := \Phi_k^\top [W_{k+1}^{-1} + M_k]^{-1} \Phi_k;$$

$$F_k := F_{k+1} - C_{k+1}^\top M_k [I + W_{k+1} M_k]^{-1} C_{k+1};$$

$$C_k := \Phi_k^\top [I + W_{k+1} M_k]^{-1} C_{k+1};$$

Obtain $K_k(t)$ and $L_k(t)$, for $t \in [t_k, t_{k+1})$, by

$$K_k(t) := -Q_k^{-1}(t - t_k) B^\top e^{A^\top(t_k - t)} W_k;$$

$$L_k(t) := Q_k^{-1}(t - t_k) B^\top e^{A^\top(t_k - t)} C_k;$$

end

Remark 1. In Algorithm 2, each M_k is positive semi-definite. They will be positive definite if (A, B) is a controllable pair, but that condition is not assumed here. Additionally, each W_k is positive definite. Thus, the inverse $[W_{k+1}^{-1} + M_k]^{-1}$ exists and,

$$[W_{k+1}^{-1} + M_k]^{-1} W_{k+1}^{-1} = [I + W_{k+1} M_k]^{-1}. \quad (3.5)$$

Therefore, the inverse $[I + W_{k+1} M_k]^{-1}$ also exists and the algorithm is well behaved.

Some important notes on the solution are given below. Let $\bar{\mathbf{x}}^*(t)$ be the expected value of \mathbf{x}_t for system (3.1) driven by the optimal feedback control law $\mathbf{u}^*(t)$. With the control law given in Theorem 2, $\bar{\mathbf{x}}^*(t_k)$ for $k = 1, \dots, N + 1$ (with $t_{N+1} = t_f$ the final time) can be obtained as follows:

$$\bar{\mathbf{x}}^*(t_k) = [I + M_{k-1} W_k]^{-1} (\Phi_{k-1} \mathbf{y}_{t_{k-1}} + M_{k-1} C_k \mathbf{z}_f). \quad (3.6)$$

This expression is validated later in the proof of the main theorem. Note, in particular, that $\bar{\mathbf{x}}^*(t_f)$ is not necessarily equal to the target \mathbf{z}_f . This makes the optimal control $\mathbf{u}^*(t)$ essentially different from the one for stochastic linear systems with strictly additive noise.

It is possible to impose the constraint $\bar{\mathbf{x}}(t_f) = \mathbf{z}_f$, but the optimal control law for the constrained problem does not necessarily minimize the squared deviation. In fact, the constrained formulation will result in a significant increase in the cost (3.2), especially when measurements are taken near the final time (i.e., $t_f - t_N$ is small). The reason for this behavior is that significant control effort is required to steer the expected value of the system from \mathbf{x}_{t_N} to the target \mathbf{z}_{t_f} (so as to meet the constraint), introducing a significant amount of control-dependent noise during the end of the trajectory.

Including $\int_0^{t_f} \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau$ in J guarantees that $Q(t)$ is positive definite and thus invertible, even when B is not full column rank. In cases where B is full column rank, a positive semidefinite R (or even a zero matrix) is permissible. Because the expected value of the final state is unconstrained, over-weighting R will lead to a large “miss” of the target state: as $R \rightarrow \infty$, $\bar{\mathbf{x}}_{t_f}^* \rightarrow \Phi \mathbf{x}_0$. Care should be taken when selecting R to avoid over-penalizing control effort.

3.3.1 Special Case: Optimal Open-Loop Control

Before providing a proof of Theorem 2, the simple open-loop control case will be addressed; results from the open-loop case will become useful during the proof of Theorem 2. The open-loop case does not include measurement outputs, i.e., $N = 0$. Following Algorithm 2, let

$$K(t) := -Q^{-1}(t) B^\top e^{A^\top(t_f-t)} (M + I)^{-1} \Phi, \quad (3.7)$$

$$L(t) := Q^{-1}(t) B^\top e^{A^\top(t_f-t)} (M + I)^{-1}, \quad (3.8)$$

where $Q(t)$ and M are given by

$$Q(t) := R + \sigma^2 B^\top e^{A^\top(t_f-t)} e^{A(t_f-t)} B, \quad (3.9)$$

$$M := \int_0^{t_f} e^{A(t_f-\tau)} B Q^{-1}(\tau) B^\top e^{A^\top(t_f-\tau)} d\tau. \quad (3.10)$$

Correspondingly, Theorem 2 is reduced to the following:

Proposition 1. Consider system (3.1) without measurement outputs. The optimal control $\mathbf{u}^*(t)$ for minimizing Eq. (3.2) is

$$\mathbf{u}^*(t) = K(t)\mathbf{x}_0 + L(t)\mathbf{z}_f, \quad (3.11)$$

where $K(t)$ and $L(t)$ are given by Eqs. (3.7) and (3.8). The minimized cost is given by

$$J^* = (\Phi\mathbf{x}_0 - \mathbf{z}_f)^\top (M + I)^{-1} (\Phi\mathbf{x}_0 - \mathbf{z}_f), \quad (3.12)$$

where M is given in Eq. (3.10).

Some preliminaries are required to establish Prop. 1. Recall that $\bar{\mathbf{x}}(t)$ is the expected value of \mathbf{x}_t , and let

$$\Sigma(t) := \mathbb{E}[(\mathbf{x}_t - \bar{\mathbf{x}}(t))(\mathbf{x}_t - \bar{\mathbf{x}}(t))^\top] \quad (3.13)$$

be the covariance matrix at time t . By the Itô rule and the expectation rule (see Appendix A), the dynamics of $\bar{\mathbf{x}}(t)$ and $\Sigma(t)$ obey the following linear differential equations for a given open-loop control input:

$$\dot{\bar{\mathbf{x}}}(t) = A\bar{\mathbf{x}}(t) + B\mathbf{u}(t), \quad (3.14)$$

$$\dot{\Sigma}(t) = A\Sigma(t) + \Sigma(t)A^\top + \sigma^2 B\mathbf{u}(t)\mathbf{u}^\top(t)B^\top. \quad (3.15)$$

The initial condition, \mathbf{x}_0 , of system (3.1) is known such that $\Sigma(0) = 0$. The differential equations can be solved explicitly:

$$\bar{\mathbf{x}}(t_f) = \Phi\mathbf{x}_0 + \int_0^{t_f} e^{A(t_f-\tau)} B\mathbf{u}(\tau) d\tau, \quad (3.16)$$

$$\Sigma(t_f) = \sigma^2 \int_0^{t_f} e^{A(t_f-\tau)} B\mathbf{u}(\tau)\mathbf{u}^\top(\tau)B^\top e^{A^\top(t_f-\tau)} d\tau. \quad (3.17)$$

With the above expressions, the cost function in Eq. (3.2) is rewritten as follows. For the first term,

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}_{t_f} - \mathbf{z}_f\|^2] &= \mathbb{E}[\|(\mathbf{x}_{t_f} - \bar{\mathbf{x}}(t_f)) + (\bar{\mathbf{x}}(t_f) - \mathbf{z}_f)\|^2], \\ &= \text{tr}(\Sigma(t_f)) + \|\bar{\mathbf{x}}(t_f) - \mathbf{z}_f\|^2. \end{aligned} \quad (3.18)$$

This expansion is analogous to a bias-variance decomposition. Combining the solution of $\Sigma(t)$ provided in Eq. (3.17) with the second term of Eq. (3.2) results in

$$J = \int_0^{t_f} \mathbf{u}^\top(\tau)Q(\tau)\mathbf{u}(\tau) d\tau + \|\bar{\mathbf{x}}(t_f) - \mathbf{z}_f\|^2, \quad (3.19)$$

where $Q(t)$ is given in Eq. (3.9). Using the preliminaries above, Prop. 1 is proven below.

Proof of Proposition 1. From Eq. (3.16), $\bar{\mathbf{x}}(t_f)$ is linear in $\mathbf{u}(t)$, so by Eq. (3.19), J is quadratic in $\mathbf{u}(t)$. The optimal control $\mathbf{u}^*(t)$ can be obtained by first-order analysis. Consider adding a small perturbation $\delta\mathbf{u}(t)$ to the optimal control. Then, the variation of the cost $J = J^* + \delta J$, up to the first order, is given by

$$\delta J = 2 \int_0^{t_f} \delta\mathbf{u}^\top(\tau) \left(Q(\tau)\mathbf{u}^*(\tau) + B^\top e^{A^\top(t_f-\tau)}(\bar{\mathbf{x}}^*(t_f) - \mathbf{z}_f) \right) d\tau. \quad (3.20)$$

A necessary condition for $\mathbf{u}^*(t)$ to be optimal is that δJ vanishes for all $\delta\mathbf{u}$:

$$Q(\tau)\mathbf{u}^*(\tau) + B^\top e^{A^\top(t_f-\tau)}(\bar{\mathbf{x}}^*(t_f) - \mathbf{z}_f) = 0. \quad (3.21)$$

It follows that

$$\mathbf{u}^*(t) = -Q^{-1}(\tau)B^\top e^{A^\top(t_f-t)}(\bar{\mathbf{x}}^*(t_f) - \mathbf{z}_f). \quad (3.22)$$

Combining the above equation with Eq. (3.16),

$$\bar{\mathbf{x}}^*(t_f) = (M + I)^{-1}(\Phi\mathbf{x}_0 + M\mathbf{z}_f). \quad (3.23)$$

The optimal control $\mathbf{u}^*(t)$ is then obtained by replacing $\bar{\mathbf{x}}^*(t_f)$ in Eq. (3.22) with the above expression. Moreover, the minimized cost can be obtained by Eq. (3.19):

$$J^* = (\bar{\mathbf{x}}^*(t_f) - \mathbf{z}_f)^\top (M + I) (\bar{\mathbf{x}}^*(t_f) - \mathbf{z}_f), \quad (3.24)$$

$$= (\Phi\mathbf{x}_0 - \mathbf{z}_f)^\top (M + I)^{-1} (\Phi\mathbf{x}_0 - \mathbf{z}_f). \quad (3.25)$$

This completes the proof. ■

3.3.2 General Case: Optimal Feedback Control

Theorem 2 is proven in this subsection. The problem will be solved backwards in time using dynamic programming [23]. To that end, the following value functions are introduced: For any $k = 0, \dots, N$, any state \mathbf{x}_{t_k} , and any feedback control law \mathbf{u} over $[t_k, t_f]$ (feedback in the sense that $\mathbf{u}(t)$ can depend on the measurements $y_{t_{k'}}$, for $t_k \leq t_{k'} < t$, where $\mathbf{x}_{t_k} = \mathbf{y}_{t_k}$), the value function is defined as

$$V_k(\mathbf{x}_{t_k}, \mathbf{z}_f, \mathbf{u}) := \mathbb{E} \left[\|\mathbf{x}_{t_f} - \mathbf{z}_f\|^2 + \int_{t_k}^{t_f} \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau \mid \mathbf{x}_{t_k} \right], \quad (3.26)$$

where \mathbf{x}_{t_f} is the solution of system (3.1) and \mathbf{x}_{t_k} is the initial condition at time t_k . For ease of notation, $\mathbb{E}[\cdot]$ will be written instead of $\mathbb{E}[\cdot|\mathbf{x}_{t_k}]$ to denote the conditional expectation. Thus, the original cost J given in Eq. (3.2) is simply $V_0(\mathbf{x}_0, \mathbf{z}_f, \mathbf{u})$. Similarly, let

$$V_k^*(\mathbf{x}_{t_k}, \mathbf{z}_f) := \min_{\mathbf{u}} V_k(\mathbf{x}_{t_k}, \mathbf{z}_f, \mathbf{u}). \quad (3.27)$$

Dynamic programming will be used to generate a recursive formula for V_k^* backward in time. In particular, each V_k^* is a quadratic form in \mathbf{x}_{t_k} and \mathbf{z}_f .

Proof of Theorem 2. The proof begins with V_N . Because \mathbf{x}_{t_N} is known to the controller (by the full-state, noiseless measurement output) and because there is no measurement output over the time interval (t_N, t_f) , the problem of minimizing V_N is the same as the open-loop control problem solved in Section 3.3.1. From Prop. 1,

$$V_N^*(\mathbf{x}_{t_N}, \mathbf{z}_f) = \mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N} - 2\mathbf{x}_{t_N}^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f. \quad (3.28)$$

The three matrices W_N , C_N , and F_N are given by

$$W_N = \Phi_N^\top (M + I)^{-1} \Phi_N, \quad (3.29)$$

$$C_N = \Phi_N^\top (M + I)^{-1}, \quad (3.30)$$

$$F_N = (M + I)^{-1}. \quad (3.31)$$

The matrix M is defined in Eq. (3.10) (but with the lower limit of the integral replaced with t_N). This result is consistent with Algorithm 2; the algorithm generates the same values for matrices W_N , C_N , and F_N . It also follows from Prop. 1 that the optimal control $\mathbf{u}^*(t)$ over the period $[t_N, t_f]$ is given by

$$\mathbf{u}(t) = K_N(t) \mathbf{x}_{t_N} + L_N(t) \mathbf{z}_f, \quad (3.32)$$

where K_N and L_N are computed in Algorithm 2. The two matrices agree with $K(t)$ and $L(t)$ in Eq. (3.7) if one replaces Φ with $\Phi_N = e^{A(t_f - t_N)}$ and changes the lower limit of the integral in Eq. (3.10) to t_N .

Now that the problem has been solved for the final segment of the trajectory, the next step is to consider V_{N-1} and solve the optimal control $\mathbf{u}^*(t)$ over the segment $[t_{N-1}, t_N]$. Note that the following

relation between V_{N-1} and V_N holds:

$$V_{N-1}(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f, \mathbf{u}) = \mathbb{E} \left[V_N(\mathbf{x}_{t_N}, \mathbf{z}_f, \mathbf{u}|_{[t_N, t_f]}) \right] + \int_{t_{N-1}}^{t_N} \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau, \quad (3.33)$$

where $\mathbf{u}|_{[t_N, t_f]}$ indicates that the control input is restricted to the last segment $[t_N, t_f]$ and the expectation is with respect to \mathbf{x}_{t_N} . Although a noiseless measurement \mathbf{y}_{t_N} will be taken at t_N , the state is still unknown immediately before the measurement. Note that \mathbf{x}_{t_N} also depends on $\mathbf{u}|_{[t_{N-1}, t_N]}$. By the principle of optimality, the following holds:

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \min_{\mathbf{u}|_{[t_{N-1}, t_N]}} \left[\mathbb{E} [V_N^*(\mathbf{x}_{t_N}, \mathbf{z}_f)] + \int_{t_{N-1}}^{t_N} \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau \right]. \quad (3.34)$$

The optimization of V_{N-1}^* is solved below. For ease of notation in the following equations, \mathbf{u} will indicate the control law over this segment. First, note that the quadratic form of V_N^* is given in Eq. (3.28), so Eq. (3.34) can be reduced to

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \min_{\mathbf{u}} \left[\mathbb{E}[\mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N}] - 2\bar{\mathbf{x}}(t_N) C_N \mathbf{z}_f^\top + \int_{t_{N-1}}^{t_N} \mathbf{u}^\top(\tau) R \mathbf{u}(\tau) d\tau \right] + \mathbf{z}_f^\top F_N \mathbf{z}_f. \quad (3.35)$$

Recall that $\bar{\mathbf{x}}(t_N) = \mathbb{E}(\mathbf{x}_{t_N})$ is the mean state. Similar to Section 3.3.1, the term $\mathbb{E}[\mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N}]$ can be decomposed as follows:

$$\mathbb{E}[\mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N}] = \text{tr}(\Sigma(t_N) W_N) + \bar{\mathbf{x}}(t_N)^\top W_N \bar{\mathbf{x}}(t_N). \quad (3.36)$$

Equation (3.36) results from replacing \mathbf{x}_{t_N} with $(\mathbf{x}_{t_N} - \bar{\mathbf{x}}(t_N)) + \bar{\mathbf{x}}(t_N)$. Next, note that both the mean $\bar{\mathbf{x}}(t_N)$ and the covariance $\Sigma(t_N)$ are functions of \mathbf{u} as follows:

$$\bar{\mathbf{x}}(t_N) = \Phi_{N-1} \mathbf{x}_{t_{N-1}} + \int_{t_{N-1}}^{t_N} e^{A(t_N-\tau)} B \mathbf{u}(\tau) d\tau, \quad (3.37)$$

$$\Sigma(t_N) = \sigma^2 \int_{t_{N-1}}^{t_N} e^{A(t_N-\tau)} B \mathbf{u}(\tau) \mathbf{u}^\top(\tau) B^\top e^{A^\top(t_N-\tau)} d\tau. \quad (3.38)$$

The derivations of the above expressions are similar to the derivations of Eqs. (3.16) and (3.17). Combining the above equations,

$$\begin{aligned} & V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) \\ &= \min_{\mathbf{u}} \left[\int_{t_{N-1}}^{t_N} \mathbf{u}^\top(\tau) Q_{N-1}(\tau) \mathbf{u}(\tau) d\tau + \bar{\mathbf{x}}^\top(t_N) W_N \bar{\mathbf{x}}(t_N) - 2\bar{\mathbf{x}}^\top(t_N) C_N \mathbf{z}_f \right] + \mathbf{z}_f^\top F_N \mathbf{z}_f. \end{aligned} \quad (3.39)$$

The matrix $Q_{N-1}(t)$ is positive definite and matches the result in Algorithm 2.

The optimization problem in Eq. (3.39) is again quadratic in \mathbf{u} (noting that $\bar{\mathbf{x}}(t_N)$ depends linearly in \mathbf{u}). Repeating the first-order analysis, the optimal control \mathbf{u}^* takes the following form for all $t \in [t_{N-1}, t_N)$.

$$\mathbf{u}^*(t) = -Q_{N-1}^{-1}(t)B^\top e^{A^\top(t_N-t)}(W_N\bar{\mathbf{x}}(t_N) - C_N\mathbf{z}_f) \quad (3.40)$$

Combining the above expression with Eq. (3.37),

$$\bar{\mathbf{x}}^*(t_N) = [I + M_{N-1}W_N]^{-1}(\Phi_{N-1}\mathbf{x}_{t_{N-1}} + M_{N-1}C_N\mathbf{z}_f). \quad (3.41)$$

Therefore, the optimal control $\mathbf{u}^*(t)$ for all $t \in [t_{N-1}, t_N)$ is given by

$$\mathbf{u}^*(t) = K_{N-1}(t)\mathbf{x}_{t_{N-1}} + L_{N-1}(t)\mathbf{z}_f, \quad (3.42)$$

where the matrices $K_{N-1}(t)$ and $L_{N-1}(t)$ match the results provided by Algorithm 2. Likewise, the above expression for the mean $\bar{\mathbf{x}}^*(t_N)$ matches the expression given previously in Eq. (3.6).

The optimal value function V_{N-1}^* can be evaluated with the optimal control \mathbf{u}^* and the corresponding $\bar{\mathbf{x}}(t_N)$ computed above. It directly follows that

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \mathbf{x}_{t_{N-1}}^\top W_{N-1} \mathbf{x}_{t_{N-1}} - 2\mathbf{x}_{t_{N-1}}^\top C_{N-1} \mathbf{z}_f + \mathbf{z}_f^\top F_{N-1} \mathbf{z}_f. \quad (3.43)$$

where the matrices W_{N-1} , C_{N-1} , and F_{N-1} follow the update rule given in Algorithm 2. The above procedure can be repeated to solve all V_k^* and $\mathbf{u}^*|_{[t_k, t_{k+1})}$ backward in time, and the solutions match the results provided in Algorithm 2 and Theorem 2. ■

3.4 Continuous Closed-Loop Control Example

This section provides numerical examples of the result in Theorem 2. First, the change in the minimized cost J^* with the increase of measurement outputs, N , is investigated. Using the explicit expression for J^* given in Theorem 2, J^* is computed with N ranging from 0 to 50 for four two-dimensional linear systems. In each system, (A, B) takes the following form:

$$A = \begin{bmatrix} 0 & 1 \\ \beta & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (3.44)$$

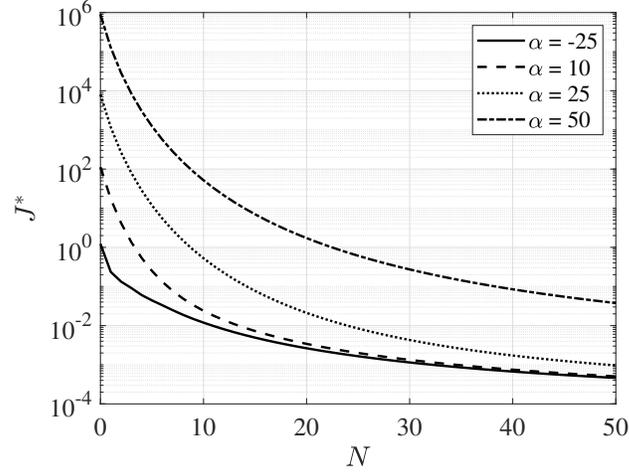


Figure 3.2: Numerical simulations of J^* vs. N for the system in Eq.(3.44) ($t_f = 1$, $\sigma = 0.2$, $\mathbf{x}_0 = [1, 1]^\top$, $\mathbf{z}_f = [0, 0]^\top$, $R = 0$). For each case, J^* asymptotically decays and approaches a positive value as its limit. The stable case appears to provide a lower bound for other cases.

where $\beta \in \{-25, 10, 25, 50\}$ corresponds to one stable system and three hyperbolically unstable systems, respectively. For all simulations, $t_f = 1$, $\sigma = 0.2$, $\mathbf{x}_0 = [1, 1]^\top$, and $\mathbf{z}_f = [0, 0]^\top$. Measurement times t_k are evenly spaced such that $t_k = t_{k-1} + t_f/(N + 1)$. The B matrix is full column rank, so a positive semidefinite R matrix is permissible; $R = 0$ is chosen for all simulations.

Figure 3.2 shows the relationship between J^* and N for $\beta \in \{-25, 10, 25, 50\}$. Numerical simulations show that J^* monotonically decreases as N grows. Moreover, in each simulation $J^*(N)$ asymptotically decays and approaches a positive value as its limit. This behavior is expected; even with continuous measurements, the final mean squared state deviation will be nonzero due to the influence of control-linear noise.

It is also interesting to study the trajectory of $\bar{\mathbf{x}}^*(t)$, i.e., the mean of \mathbf{x}_t . Next, the system in Eq. (3.44) is simulated with $\beta = 10$. For the purpose of demonstration, the sampled measurements are chosen to be the mean state, i.e., $\mathbf{y}_{t_k} = \bar{\mathbf{x}}^*(t_k)$. Figure 3.3 shows the corresponding trajectory of $\bar{\mathbf{x}}^*(t)$ as well as the optimal control law $\mathbf{u}^*(t)$ for three cases: $N = 0, 3$, and 10. The simulation results show that as the number of measurements is increased, $\bar{\mathbf{x}}^*(t_f)$ gets closer to the target state \mathbf{z}_f . Regarding the optimal control law $\mathbf{u}^*(t)$, Figure 3.3 indicates that for positive N , the magnitude of $\mathbf{u}^*(t)$ is close to zero in the last few segments. However, this is not the case if $N = 0$. The same figure also indicates that, within each segment,

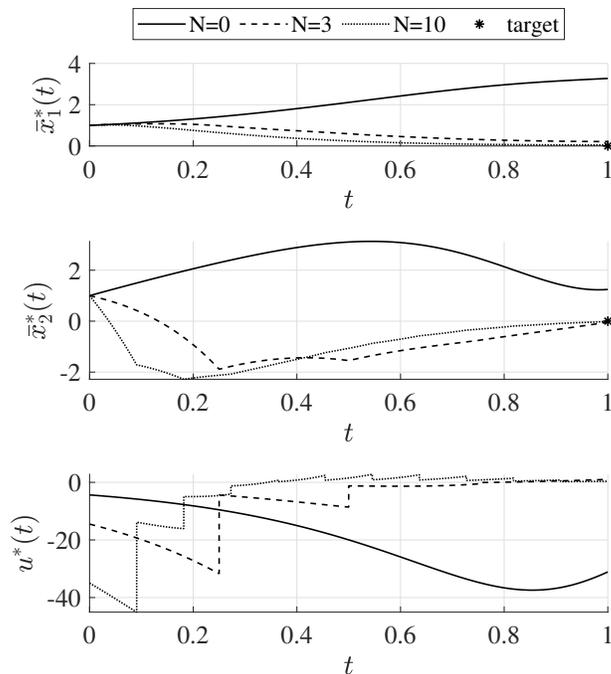


Figure 3.3: Trajectory for $\bar{\mathbf{x}}^*(t) = [\bar{x}_1^*(t), \bar{x}_2^*(t)]^\top$ and the optimal control law for system (3.44) with $\beta = 10$ ($t_f = 1$, $\sigma = 0.2$, $\mathbf{x}_0 = [1, 1]^\top$, $\mathbf{z}_f = [0, 0]^\top$, $R = 0$). Let $y_{t_k} = \bar{\mathbf{x}}^*(t_k)$ for all $k = 1, \dots, N$. Note that $\bar{\mathbf{x}}^*(t_f)$ gets closer to \mathbf{z}_f as N increases.

the optimal control reserves control effort until the end of the segment. The reason may be the following: The uncertainty caused by the control-linear noise introduced toward the end of a segment can be quickly remedied by a new measurement output. This behavior may be problematic if a measurement is lost.

3.5 Discussion and Conclusions

This chapter addressed the feedback control of stochastic linear systems with control-linear noise, sampled measurement outputs, and continuous control. An optimization problem was formulated to steer a stochastic system from an initial condition to a final state that minimizes mean squared deviation from a target state. A complete solution to the problem was derived using dynamic programming. In particular, it was demonstrated that the optimal control law is linear in the initial state, the target state, and the measurement outputs. Moreover, the state feedback gains can be computed explicitly a priori, as given in Algorithm 2. Numerical simulations showed that the mean squared deviation decays dramatically as the number of measurement outputs increases, especially for systems with unstable modes.

Some interesting problems are left for the future work. Specifically, optimal measurement scheduling may produce greater cost reduction. Additionally, the loss of scheduled measurements may pose an issue. The numerical simulations in this chapter indicated that the optimal control inputs tend to increase in magnitude in anticipation of upcoming measurements. These spikes in control effort may cause significant covariance inflation if scheduled measurements are lost, and future work is needed to evaluate robustness to missed measurements.

In the next chapter, these results are extended to linear time-varying systems with additive noise, multiple sources of control-linear noise, and impulsive control inputs that are asynchronous with measurement outputs. This chapter assumed that all sampled measurements were perfect (non-noisy) and full-state. Noisy, partial-state measurements will be addressed in Chapter 5.

Chapter 4

Impulsive Closed-Loop Robust Control

Chapter 4 Nomenclature

| | | |
|----------------------------|---|--|
| $A(t)$ | = | linearized dynamics |
| $B_0(t), \dots, B_\ell(t)$ | = | control matrices |
| $\mathbf{f}(\mathbf{X}_t)$ | = | nonlinear state dynamics |
| $G(t)$ | = | additive noise matrix |
| I | = | identity matrix |
| J | = | cost function |
| $K_{k,j}$ | = | state feedback gain |
| $L_{k,j}$ | = | target state feedback gain |
| ℓ | = | number of control-linear noise sources |
| m | = | dimension of control input |
| n | = | dimension of system state |
| N | = | number of measurements |
| p_k | = | number of controls in segment k |
| R | = | positive definite matrix |
| \mathbf{r} | = | position vector |
| t | = | time |
| t_k | = | time of k th measurement |

| | | |
|---------------------------|---|--|
| $t_{k,j}$ | = | time of j th control after the k th measurement |
| t_f | = | final time |
| $\Delta \mathbf{u}_{k,j}$ | = | j th impulsive control after the k th measurement |
| V_k | = | value function |
| \mathbf{v} | = | velocity vector |
| dW_t | = | Wiener process |
| \mathbf{X}_t | = | stochastic nonlinear system state at time t |
| $\mathbf{X}'(t)$ | = | nominal nonlinear system state at time t |
| \mathbf{x}_0 | = | known initial state of linear (or linearized) system |
| \mathbf{x}_t | = | stochastic linear (or linearized) system state at time t |
| $\hat{\mathbf{x}}_t$ | = | state estimate at time t |
| \mathbf{y}_t | = | perfect, full-state measurement |
| \mathbf{z}_f | = | target state |
| Σ | = | state covariance |
| σ_{add} | = | additive noise standard deviation |
| σ_{SRP} | = | SRP noise standard deviation |
| τ | = | variable of integration |
| Φ | = | state transition matrix from initial to final time, e^{At_f} |
| Φ_k | = | state transition matrix from t_k to t_{k+1} |
| $\Phi_{k,j}$ | = | state transition matrix from $t_{k,j}$ to t_{k+1} |
| Φ_τ | = | state transition matrix from τ to t_{k+1} |

4.1 Introduction

This chapter extends the results in the previous chapter to the spacecraft guidance problem, and addresses the optimal intermittent control of hybrid, stochastic linear systems with time-varying dynamics, sampled measurements, additive noise, and noisy impulsive control inputs. The control law will be derived

for linear systems, but will be applied to nonlinear astrodynamics systems as a neighboring guidance law by linearizing state deviations about nominal trajectories.

The term hybrid refers to a dynamical system with piecewise continuous states and discrete state jumps, i.e., the system dynamics are in continuous time, and the impulsive controls give rise to state jumps. Due to impulsive control fluctuations, the differences between the states before and after jumps will be treated as random variables, which obey Gaussian distributions. This system model is motivated by spacecraft maneuver design in which short-duration thrust accelerations are often approximated as instantaneous velocity changes. Other examples of hybrid systems (with impulsive controls) are given in [43] and include mechanical systems with impacts that induce instantaneous state changes, financial systems where a monetary deposit is an instantaneous state change, and ecological systems where the number of individuals in a species population varies instantaneously with birth, death, capture, and release.

As in the previous chapter, the goal is to develop an optimal feedback control law to minimize the mean squared deviation of the final system state from a target state. This chapter will also consider additive noise in the continuous-time dynamics and a more versatile control-linear noise model. The model in this chapter will allow multiple sources of control-linear noise, and each noise source may act in an arbitrary direction. In the case of spacecraft guidance, this model can accommodate independent noise from multiple thrusters simultaneously, as well as thrust direction uncertainty.

During the derivation of the optimal control law, sampled measurements are again assumed to be perfect and full-state. However, noisy measurements will be introduced in some of the simulations later in this chapter and will be investigated further in Chapter 5. The measurements are not required to occur at the same times as the impulsive controls, i.e., the controls and measurements are asynchronous.

The sampled measurement model with impulsive control is motivated by operational constraints that are common in spacecraft guidance. With limited power and computational resources, spacecraft cannot prioritize measurement and control at all times. For example, maneuvers cannot be performed while the spacecraft is maintaining a fixed orientation for scientific observation, transceiving, or solar panel charging. Likewise, measurements can also be limited. Common measurement types include range and range rate from a ground station, and these measurements are only available when the ground station is in the space-

craft's line-of-sight and the spacecraft is pointing its antenna appropriately. In practice, navigation updates and trajectory correction maneuvers often involve a human in-the-loop and are planned in an ad hoc manner. However, autonomous guidance is desirable when continuous human input is impractical or becomes prohibitively expensive, e.g., for the growing number of low-cost missions in cislunar and deep space.

The major contribution of this chapter is to provide a complete solution to the optimal control problem described above. As in the previous chapter, the optimal controls will be linear state feedback and feedback gains can be computed offline before flight. Therefore, the method is computationally efficient and may be appropriate for autonomous guidance. The example applications in this chapter will demonstrate that the algorithm can be applied as a neighboring guidance algorithm about a nonlinear spacecraft trajectory. To illustrate the effectiveness of this robust guidance law, its performance will be compared to a deterministic guidance law — an LQR that is modified to have the same measurement and control schedule as the robust guidance law.

This chapter is organized as follows. The optimization problem is formulated in Section 4.2. The main result and proof are provided in Section 4.3. In Section 4.4, the optimal guidance algorithm is demonstrated for an asteroid orbiter through numerical simulations. In Section 4.5, the robust guidance law is compared to a deterministic guidance law and the validity of the perfect measurement assumption is investigated.

4.2 Impulsive Closed-Loop Control Problem Formulation

This chapter addresses the closed-loop control of linear time-varying stochastic systems with sampled measurements, control-linear noise, additive noise, and impulsive controls. Section 4.3 will describe how the result may be applied as a neighboring guidance law for a *nonlinear* system. First, consider a continuous-time, linear time-varying stochastic system in the Itô sense, with noisy impulsive control inputs. Itô stochastic differential equations are described in Appendix A. Let $\mathbf{x}_t \in \mathbb{R}^n$ be the system state at time t , \mathbf{y}_{t_k} be the sampled measurement output at t_k , and $\Delta \mathbf{u}_{k,j} \in \mathbb{R}^m$ be a nominal impulsive control input at times $t_{k,j}$. The system state is discontinuous at impulse times $t_{k,j}$. The states before and after an impulse

are denoted $\mathbf{x}_{t_k,j}^-$ and $\mathbf{x}_{t_k,j}^+$, respectively. Given the notation above, the system dynamics are

$$\begin{cases} d\mathbf{x}_t = A(t)\mathbf{x}_t dt + G(t)dW_t, \\ \mathbf{x}_{t_k,j}^+ = \mathbf{x}_{t_k,j}^- + \left[B_0(t_{k,j}) + \sum_{i=1}^{\ell} B_i(t_{k,j})e_i \right] \Delta \mathbf{u}_{k,j}, \\ \mathbf{y}_{t_k} = \mathbf{x}_{t_k}, \quad 0 < t_1 < t_2 < \dots \end{cases} \quad (4.1)$$

where dW_t is a standard Wiener process in \mathbb{R}^q , $e_i \sim \mathcal{N}(0, 1)$ are independent and identically distributed normal random variables, and $A(t)$, $G(t)$, and $B_0(t) \dots B_\ell(t)$ are matrices of appropriate dimensions. Each impulse injects ℓ independent sources of error.

It is assumed that the initial state \mathbf{x}_0 is known to the controller, the final time t_f is fixed, and the measurements and impulsive controls are scheduled a priori. The controller receives N full-state, noiseless measurement outputs $\mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_N}$ at times $t_1 < t_2 < \dots < t_N < t_f$. The full-state, noiseless measurement assumption is key to enabling the analytical optimization performed in this chapter; relaxing this condition presents significant challenges that are discussed in the next section.

The period between two measurements \mathbf{y}_{t_k} and $\mathbf{y}_{t_{k+1}}$ is referred to as segment k . During segment k , p_k impulsive control inputs $\Delta \mathbf{u}_{k,j}$ occur at times $t_{k,j}$ where $t_k \leq t_{k,1} < \dots < t_{k,p_k} < t_{k+1}$. See Figure 4.1 for illustration. At each impulse time $t_{k,j}$, the control is allowed to depend on all past measurements \mathbf{y}_{t_i} , for $t_i \leq t_{k,j}$, as well as the known initial state \mathbf{x}_0 .

Problem: Consider the stochastic control system in Eq. (4.1). Given an initial condition \mathbf{x}_0 , a target state \mathbf{z}_f , a final time t_f , N measurement outputs \mathbf{y}_{t_k} , and the fixed times of $\sum_{k=0}^N p_k$ total impulsive control

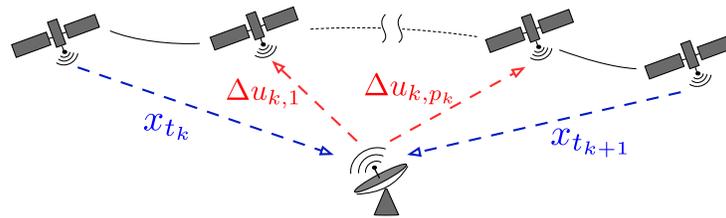


Figure 4.1: Illustration of the system model (4.1) over segment k . The controller receives sampled measurements at times t_k (and t_{k+1}) and sends impulsive controls occurring at times $t_{k,j}$, for $j = 1, \dots, p_k$. The graphic depicts communication with a ground station, but an onboard controller is also possible.

inputs, find the optimal impulsive controls $\Delta \mathbf{u}_{k,j}$ that minimize the following cost function:

$$J := \mathbb{E} \left[\|\mathbf{x}_{t_f} - \mathbf{z}_f\|^2 + \sum_{k=0}^N \sum_{j=1}^{p_k} \Delta \mathbf{u}_{k,j}^\top R \Delta \mathbf{u}_{k,j} \right] \quad (4.2)$$

where R is positive definite. The first term of Eq. (4.2) penalizes deviation of the final state from the target. The second term penalizes impulsive control effort over the period and is a regularization condition that guarantees that the optimal controls $\Delta \mathbf{u}_{k,j}^*$ are unique. However, R need not be positive definite in all cases, and the positive-definite restriction will be relaxed later in this section.

Notation: In the sequel, let Φ_k be the STM associated with the matrix $A(t)$ from t_k to t_{k+1} , $\Phi_{k,j}$ be the STM from $t_{k,j}$ to t_{k+1} , and Φ_τ be the STM from τ to t_{k+1} , where τ is a variable of integration. Some expected values are denoted by overbars as in $\mathbb{E}[\mathbf{x}_t] = \bar{\mathbf{x}}(t)$. A superscript * indicates an optimized quantity.

4.3 Impulsive Closed-Loop Control Result

The result of the optimization problem is stated in Theorem 3 and will be proven below. The feedback controls that minimize Eq. (4.2) are linear in the initial state \mathbf{x}_0 , the target state \mathbf{z}_f , and the measurements \mathbf{y}_{t_k} . The state feedback gains can be computed offline using Algorithm 3.

Theorem 3. *Consider the stochastic system in Eq. (4.1) and the optimal control problem in Eq. (4.2). The optimal impulsive controls $\Delta \mathbf{u}_{k,j}^*$ are given by*

$$\Delta \mathbf{u}_{k,j}^* = K_{k,j} \mathbf{y}_{t_k} + L_{k,j} \mathbf{z}_f, \quad (4.3)$$

for $k = 0, \dots, N$ and $j = 1, \dots, p_k$, where \mathbf{y}_0 is identified with \mathbf{x}_0 . The minimized cost function is a quadratic form:

$$J^* = \mathbf{x}_0^\top W_0 \mathbf{x}_0 - 2\mathbf{x}_0^\top C_0 \mathbf{z}_f + \mathbf{z}_f^\top F_0 \mathbf{z}_f + \varphi_0. \quad (4.4)$$

The scalar φ_0 and the matrices $K_{k,j}$, $L_{k,j}$, W_0 , C_0 , and F_0 in the above expressions are computed in Algorithm 3.

Before the proof of Theorem 3, some notes will be made on the optimal control result. From Remark 1, inverses $[W_{k+1}^{-1} + M_k]^{-1}$ and $[I + W_{k+1} M_k]^{-1}$ are guaranteed to exist, and Algorithm 3 is well

Algorithm 3: Offline computation of state feedback gains and coefficients of the quadratic form J^* .

Data: Matrices $A(t)$, $B_0(t)$, \dots , $B_\ell(t)$, R , measurement times t_1, \dots, t_N , impulse times

$t_{k,1}, \dots, t_{k,p_k}$, for all $k = 0, \dots, N$, and final time $t_f =: t_{N+1}$.

Result: Scalar φ_0 , matrices W_0 , C_0 , F_0 , $K_{k,j}$ and $L_{k,j}$ for all $k = 0, \dots, N$ and $j = 1, \dots, p_k$.
initialize; $W_{N+1} := I$; $F_{N+1} := I$; $C_{N+1} := I$; $\varphi_{N+1} := 0$.

for $k := N$ **to** 0 **do**

for $j := 1$ **to** p_k **do**

$$Q_{k,j} := R + \sum_{i=1}^{\ell} B_i(t_{k,j})^\top \Phi_{k,j}^\top W_{k+1} \Phi_{k,j} B_i(t_{k,j});$$

end

$$M_k := \sum_{j=1}^{p_k} \Phi_{k,j} B_0(t_{k,j}) Q_{k,j}^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top;$$

$$W_k := \Phi_k^\top [W_{k+1}^{-1} + M_k]^{-1} \Phi_k;$$

$$C_k := \Phi_k^\top [I + W_{k+1} M_k]^{-1} C_{k+1};$$

$$F_k := F_{k+1} - C_{k+1}^\top M_k [I + W_{k+1} M_k]^{-1} C_{k+1};$$

$$\Gamma_k := \int_{t_k}^{t_{k+1}} \Phi_\tau G(\tau) G(\tau)^\top \Phi_\tau^\top d\tau;$$

$$\varphi_k := \varphi_{k+1} + \text{tr}(W_{k+1} \Gamma_k);$$

and

$$K_{k,j} := -Q_{k,j}^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top [W_{k+1}^{-1} + M_k]^{-1} \Phi_k;$$

$$L_{k,j} := Q_{k,j}^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top [I + W_{k+1} M_k]^{-1} C_{k+1};$$

end

behaved. Another consideration is the rank of $Q_{k,j}$. Assuming that R is positive definite guarantees the existence of $Q_{k,j}^{-1}$. However, by inspection of Algorithm 3 it is clear that R is not required to be positive definite in all cases; R can be positive semidefinite (or simply zero), if

$$\sum_{i=1}^{\ell} B_i(t_{k,j})^\top \Phi_{k,j}^\top W_{k+1} \Phi_{k,j} B_i(t_{k,j}) \quad (4.5)$$

is invertible.

As in Chapter 3, the mean of the optimal final state $\bar{\mathbf{x}}^*(t_f)$ differs from the target state \mathbf{z}_f . Applying the optimal control in Theorem 3 to system (4.1) will result in

$$\bar{\mathbf{x}}^*(t_f) = [I + M_N W_{N+1}]^{-1} (\Phi_N \mathbf{y}_{t_N} + M_N C_{N+1} \mathbf{z}_f). \quad (4.6)$$

The discrepancy between $\bar{\mathbf{x}}^*(t_f)$ and \mathbf{z}_f is due in part to the control penalty term in Eq. (4.2). However, $\bar{\mathbf{x}}^*(t_f)$ and \mathbf{z}_f will still differ when $R = 0$. Steering the system to the target state in expectation can require large impulsive controls and introduce large amounts of control-linear noise that inflate the mean squared deviation. Thus, it is undesirable to constrain $\bar{\mathbf{x}}^*(t_f)$ to match the target, as was observed in the continuous control case.

The full-state and noiseless measurement assumption enables a complete and analytical solution to the optimization problem. Although the impulsive controls are allowed to depend on all prior measurements, the solution is Markovian in the sense that each optimal impulsive control depends only on the most recent measurement. However, in the case of partial and noisy measurements, the optimal impulsive control at $t_{k,j}$ may depend on *all* past measurements and may no longer be linear in those measurements. One may consider replacing the perfect, full-state measurement \mathbf{y}_{t_k} with an estimate of the state \mathbf{x}_{t_k} obtained by running a Kalman filter, but the control is not guaranteed to be optimal in that case — this approach is investigated in Section 4.5. The case of partial and noisy measurements is discussed further in the next chapter.

4.3.1 Guiding a Nonlinear System

Algorithm 3 is optimized for linear systems. From a practical point of view, the algorithm may be applied to nonlinear systems by linearizing the dynamics about certain nominal trajectories. This process is

often called neighboring guidance. Let $\mathbf{X}_t \in \mathbb{R}^n$ be the state of a stochastic, hybrid nonlinear system and $\mathbf{X}'(t) \in \mathbb{R}^n$ be the state of a nominal trajectory which is deterministic. The dynamics of $\mathbf{X}'(t)$ and \mathbf{X}_t are given by

$$\dot{\mathbf{X}}'(t) = f(\mathbf{X}'(t)), \quad (4.7)$$

$$\begin{cases} d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t)dt + G(t)dW_t, \\ \mathbf{X}_{t_k, j}^+ = \mathbf{X}_{t_k, j}^- + \left[B_0(t_k, j) + \sum_{i=1}^{\ell} B_i(t_k, j)e_i \right] \Delta \mathbf{u}_{k, j}. \end{cases} \quad (4.8)$$

Consider the deviation $\mathbf{x}_t := \mathbf{X}_t - \mathbf{X}'(t)$. The dynamics of \mathbf{x}_t can be approximated by the following stochastic linear system:

$$\begin{cases} d\mathbf{x}_t = d(\mathbf{X}_t - \mathbf{X}'(t)) \approx A(t)\mathbf{x}_t dt + G(t)dW_t, \\ \mathbf{x}_{t_k, j}^+ = \mathbf{x}_{t_k, j}^- + \left[B_0(t_k, j) + \sum_{i=1}^{\ell} B_i(t_k, j)e_i \right] \Delta \mathbf{u}_{k, j}, \end{cases} \quad (4.9)$$

where $A(t) := \partial f(\mathbf{X})/\partial \mathbf{X}|_{\mathbf{X}=\mathbf{X}'(t)}$. Then, Algorithm 3 can simply be applied to the linearized dynamics. However, it is important to keep in mind that the linearization is an *approximation* of the dynamics and the control law is not guaranteed to be optimal for the linearized system, especially when considering large deviations from the nominal trajectory.

4.3.2 Proof of Theorem 3

As in the Chapter 3, dynamic programming is used to solve the Hamilton-Jacobi-Bellman equation and prove Theorem 3. First, some preliminaries are needed. To simplify the notation, let $\Delta \mathbf{u}_k := (\Delta \mathbf{u}_{k,1}, \dots, \Delta \mathbf{u}_{k,p_k})$ denote all impulsive control for segment k . Next, define a value function V_k that encompasses the cost from measurement time t_k to the final time t_f :

$$V_k(\mathbf{x}_{t_k}, \mathbf{z}_f, \Delta \mathbf{u}_k, \dots, \Delta \mathbf{u}_N) := \mathbb{E} \left[\|\mathbf{x}_{t_f} - \mathbf{z}_f\|^2 + \sum_{i=k}^N \sum_{j=1}^{p_i} \Delta \mathbf{u}_{i,j}^\top R \Delta \mathbf{u}_{i,j} \right], \quad (4.10)$$

for any $k = 0 \dots N$, any initial state \mathbf{x}_{t_k} ($= \mathbf{y}_{t_k}$), and any set of impulsive feedback controls $\Delta \mathbf{u}_k, \dots, \Delta \mathbf{u}_N$. Each $\Delta \mathbf{u}_{i,j}$ can depend on the measurements $y_{t_{i'}}$ for $t_{i'} \leq t_i < t_{i,j}$. The goal is to optimize V_k over

$\Delta \mathbf{u}_k, \dots, \Delta \mathbf{u}_N$:

$$V_k^*(\mathbf{x}_{t_k}, z_f) := \min_{\Delta \mathbf{u}_k, \dots, \Delta \mathbf{u}_N} V_k(\mathbf{x}_{t_k}, z_f, \Delta \mathbf{u}_k, \dots, \Delta \mathbf{u}_N). \quad (4.11)$$

Let

$$\Sigma(t) := \mathbb{E}[(\mathbf{x}_t - \bar{\mathbf{x}}(t))(\mathbf{x}_t - \bar{\mathbf{x}}(t))^\top] \quad (4.12)$$

be the covariance of \mathbf{x}_t at time t . When there is no sampled measurement or impulsive control event, the mean $\bar{\mathbf{x}}(t)$ and the covariance $\Sigma(t)$ obey the following linear differential equations:

$$\dot{\bar{\mathbf{x}}}(t) = A(t)\bar{\mathbf{x}}(t), \quad (4.13)$$

$$\dot{\Sigma}(t) = A(t)\Sigma(t) + \Sigma(t)A(t)^\top + G(t)G(t)^\top. \quad (4.14)$$

Given initial conditions $\bar{\mathbf{x}}^+(t_{k,j})$ and $\Sigma^+(t_{k,j})$ immediately following a measurement or an impulsive control input, the solutions to the above equations can be computed as follows:

$$\bar{\mathbf{x}}(t) = \Phi(t, t_{k,j})\bar{\mathbf{x}}^+(t_{k,j}), \quad (4.15)$$

$$\Sigma(t) = \Phi(t, t_{k,j})\Sigma^+(t_{k,j})\Phi(t, t_{k,j})^\top + \int_{t_{k,j}}^t \Phi(t, \tau)G(\tau)G(\tau)^\top \Phi(t, \tau)^\top d\tau. \quad (4.16)$$

Next, consider discontinuities in $\bar{\mathbf{x}}(t)$ and $\Sigma(t)$ due to the impulsive controls. From Eq. (4.1),

$$\bar{\mathbf{x}}^+(t_{k,j}) = \bar{\mathbf{x}}^-(t_{k,j}) + B_0(t_{k,j})\Delta \mathbf{u}_{k,j}, \quad (4.17)$$

$$\Sigma^+(t_{k,j}) = \Sigma^-(t_{k,j}) + \sum_{i=1}^{\ell} B_i(t_{k,j})\Delta \mathbf{u}_{k,j}\Delta \mathbf{u}_{k,j}^\top B_i^\top(t_{k,j}). \quad (4.18)$$

Further, recall that $\bar{\mathbf{x}}^+(t_k) = \mathbf{y}_{t_k}$ and $\Sigma^+(t_k) = 0$ after the measurement at t_k . Using Eqs. (4.15) and (4.16) to propagate $\bar{\mathbf{x}}$ and Σ when there is no impulsive control input, and using Eqs. (4.17) and (4.18) to update $\bar{\mathbf{x}}$ and Σ right after the inputs,

$$\bar{\mathbf{x}}^-(t_{k+1}) = \sum_{j=1}^{p_k} \Phi_{k,j} B_0(t_{k,j})\Delta \mathbf{u}_{k,j} + \Phi_k \mathbf{y}_{t_k}, \quad (4.19)$$

$$\Sigma^-(t_{k+1}) = \Gamma_k + \sum_{j=1}^{p_k} \sum_{i=1}^{\ell} \Phi_{k,j} B_i(t_{k,j})\Delta \mathbf{u}_{k,j}\Delta \mathbf{u}_{k,j}^\top B_i^\top(t_{k,j})\Phi_{k,j}^\top, \quad (4.20)$$

where Γ_k is defined for convenience:

$$\Gamma_k := \int_{t_k}^{t_k+1} \Phi_\tau G(\tau)G(\tau)^\top \Phi_\tau^\top d\tau. \quad (4.21)$$

Proof of Theorem 3. Using dynamic programming backwards in time, a recursive equation will be derived for V_k^* beginning with the last segment $[t_N, t_f]$. Using the bias-variance decomposition from Section 3.3, the first term of Eq. (4.10) can be rewritten as

$$\mathbb{E}[\|\mathbf{x}_{t_f} - \mathbf{z}_f\|^2] = \text{tr}(\Sigma(t_f)) + \|\bar{\mathbf{x}}(t_f) - \mathbf{z}_f\|^2. \quad (4.22)$$

It follows that V_N can be expressed as:

$$V_N(\mathbf{x}_{t_N}, \mathbf{z}_f, \Delta \mathbf{u}_N) = \text{tr}(\Sigma(t_f)) + \|\bar{\mathbf{x}}(t_f) - \mathbf{z}_f\|^2 + \sum_{j=1}^{p_N} \Delta \mathbf{u}_{N,j}^\top R \Delta \mathbf{u}_{N,j}. \quad (4.23)$$

To develop a recursive formula for V_k , the scalar $\varphi_{N+1} := 0$ and the matrices $W_{N+1} := I$, $C_{N+1} := I$, and $F_{N+1} := I$ are introduced. These values are used to simplify the notation in future calculations, and their purpose will become apparent as the optimization progresses backwards in time. These quantities can be introduced in Eq. (4.23) without changing the value function as follows:

$$\begin{aligned} V_N(\mathbf{x}_{t_N}, \mathbf{z}_f, \Delta \mathbf{u}_N) &= \varphi_{N+1} + \text{tr}(\Sigma(t_f)W_{N+1}) + \bar{\mathbf{x}}(t_f)^\top W_{N+1} \bar{\mathbf{x}}(t_f) - 2\bar{\mathbf{x}}(t_f)^\top C_{N+1} \mathbf{z}_f \\ &\quad + \mathbf{z}_f^\top F_{N+1} \mathbf{z}_f + \sum_{j=1}^{p_N} \Delta \mathbf{u}_{N,j}^\top R \Delta \mathbf{u}_{N,j} \end{aligned} \quad (4.24)$$

Using Eq. (4.20), $\Sigma(t_f)$ is expanded as follows

$$\begin{aligned} V_N(\mathbf{x}_{t_N}, \mathbf{z}_f, \Delta \mathbf{u}_N) &= \varphi_{N+1} + \text{tr}(\Gamma_k W_{N+1}) + \sum_{j=1}^{p_N} \sum_{i=1}^{\ell} \Delta \mathbf{u}_{N,j}^\top B_i(t_{N,j})^\top \Phi_{N,j}^\top W_{N+1} \Phi_{N,j} B_i(t_{N,j}) \Delta \mathbf{u}_{N,j} \\ &\quad + \bar{\mathbf{x}}(t_f)^\top W_{N+1} \bar{\mathbf{x}}(t_f) - 2\bar{\mathbf{x}}(t_f)^\top C_{N+1} \mathbf{z}_f + \mathbf{z}_f^\top F_{N+1} \mathbf{z}_f + \sum_{j=1}^{p_N} \Delta \mathbf{u}_{N,j}^\top R \Delta \mathbf{u}_{N,j}. \end{aligned} \quad (4.25)$$

Next, terms that are explicitly quadratic in $\Delta \mathbf{u}_{N,j}$ are grouped together:

$$\begin{aligned} V_N(\mathbf{x}_{t_N}, \mathbf{z}_f, \Delta \mathbf{u}_N) &= \varphi_{N+1} + \text{tr}(\Gamma_k W_{N+1}) + \bar{\mathbf{x}}(t_f)^\top W_{N+1} \bar{\mathbf{x}}(t_f) - 2\bar{\mathbf{x}}(t_f)^\top C_{N+1} \mathbf{z}_f \\ &\quad + \mathbf{z}_f^\top F_{N+1} \mathbf{z}_f + \sum_{j=1}^{p_N} \Delta \mathbf{u}_{N,j}^\top Q_{N,j} \Delta \mathbf{u}_{N,j}, \end{aligned} \quad (4.26)$$

where the matrices $Q_{N,j}$ match the ones given in Algorithm 3. From Eq. (4.19), $\bar{\mathbf{x}}(t_f)$ is linear in each $\Delta \mathbf{u}_{N,j}$. Then, from Eq. (4.26), V_N is quadratic in $\Delta \mathbf{u}_{N,j}$. Therefore, each $\Delta \mathbf{u}_{N,j}$ can be optimized individually by setting $\partial V_N / \partial \Delta \mathbf{u}_{N,j} = 0$. The optimal controls $\Delta \mathbf{u}_{N,j}^*$ are

$$\Delta \mathbf{u}_{N,j}^* = K_{N,j} \mathbf{y}_{t_N} + L_{N,j} \mathbf{z}_f, \quad (4.27)$$

and the corresponding minimized V_N^* is

$$V_N^*(\mathbf{x}_{t_N}, \mathbf{z}_f) = \varphi_N + \mathbf{y}_{t_N}^\top W_N \mathbf{y}_{t_N} - 2\mathbf{y}_{t_N}^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f. \quad (4.28)$$

The quantities $K_{N,j}$, $L_{N,j}$, φ_N , W_N , C_N , and F_N match the ones given in Algorithm 3. The mean of the optimal final state in Eq. (4.6) can be obtained by substituting $\Delta \mathbf{u}_{N,j}^*$ in the update equation for $\bar{\mathbf{x}}(t)$ given in Eq. (4.19).

The next step is to move backwards in time and optimize over segment $N - 1$, i.e., $[t_{N-1}, t_N]$. First, the following relationship between V_N and V_{N-1} is noted:

$$V_{N-1}(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f, \Delta \mathbf{u}_{N-1}, \Delta \mathbf{u}_N) = \mathbb{E}[V_N(\mathbf{x}_{t_N}, \mathbf{z}_f, \Delta \mathbf{u}_N)] + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j}. \quad (4.29)$$

The expectation is taken with respect to \mathbf{x}_{t_N} , which depends on $\Delta \mathbf{u}_{N-1}$. From Bellman's Principle of Optimality,

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \min_{\Delta \mathbf{u}_{N-1}} \left[\mathbb{E}[V_N^*(\mathbf{x}_{t_N}, \mathbf{z}_f)] + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right]. \quad (4.30)$$

Expanding V_N^* in Eq. (4.30),

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \min_{\Delta \mathbf{u}_{N-1}} \left[\varphi_N + \mathbb{E} \left[\mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N} \right] - 2\bar{\mathbf{x}}^-(t_N)^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right]. \quad (4.31)$$

Similar to Eq. (4.22), $\mathbb{E} \left[\mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N} \right]$ is also expanded as follows:

$$\mathbb{E} \left[\mathbf{x}_{t_N}^\top W_N \mathbf{x}_{t_N} \right] = \text{tr}(\Sigma^-(t_N) W_N) + \bar{\mathbf{x}}^-(t_N)^\top W_N \bar{\mathbf{x}}^-(t_N). \quad (4.32)$$

Therefore,

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \min_{\Delta \mathbf{u}_{N-1}} \left[\varphi_N + \text{tr}(\Sigma^-(t_N) W_N) + \bar{\mathbf{x}}^-(t_N)^\top W_N \bar{\mathbf{x}}^-(t_N) - 2\bar{\mathbf{x}}^-(t_N)^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right]. \quad (4.33)$$

Note that Eq. (4.33) has the same quadratic structure as in Eq. (4.24). Moreover, the quantities φ_N , W_N , C_N , and F_N are not functions of the state or control. The exact process that was used for optimizing V_N

can be used to optimize V_{N-1} such that

$$\Delta \mathbf{u}_{N-1,j}^* = K_{N-1,j} \mathbf{y}_{t_N} + L_{N-1,j} \mathbf{z}_f, \quad (4.34)$$

$$V_{N-1}^*(\mathbf{x}_{t_{N-1}}, \mathbf{z}_f) = \varphi_{N-1} + \mathbf{y}_{t_{N-1}}^\top W_{N-1} \mathbf{y}_{t_{N-1}} - 2\mathbf{y}_{t_{N-1}}^\top C_{N-1} \mathbf{z}_f + \mathbf{z}_f^\top F_{N-1} \mathbf{z}_f. \quad (4.35)$$

$K_{N-1,j}$, $L_{N-1,j}$, φ_{N-1} , W_{N-1} , C_{N-1} , and F_{N-1} follow the recursive update equations in Algorithm 3. This procedure can be repeated backwards in time for all V_k , and the solution matches the results in Theorem 3 and Algorithm 3. This completes the proof. \blacksquare

4.4 Asteroid Orbiter Robust Guidance

In this section, the optimal control law will be applied as a neighboring guidance law about a spacecraft trajectory. Consider a spacecraft orbiting a small near-Earth asteroid. Let $\mathbf{X}' = [\mathbf{r}'^\top, \mathbf{v}'^\top]^\top$ be the nominal, six-dimensional position and velocity state of the spacecraft. The spacecraft dynamics are described by the H3BP, which is detailed in Appendix B. The asteroid is assumed to be in a circular orbit about the Sun and only two known forces act on the spacecraft: asteroid point-mass gravity and SRP. The ‘‘cannonball’’ SRP model is assumed [44]. Under these assumptions, the nominal spacecraft acceleration without noise and control is given by

$$\frac{d\mathbf{r}'}{dt} = \mathbf{v}' \quad (4.36)$$

$$\frac{d\mathbf{v}'}{dt} = -2\Omega \hat{\mathbf{e}}_z \times \mathbf{v}' + \Omega^2 (3\hat{\mathbf{e}}_x \hat{\mathbf{e}}_x^\top - \hat{\mathbf{e}}_z \hat{\mathbf{e}}_z^\top) - \frac{\mu_A}{r'^3} \mathbf{r}' + a_{SRP} \hat{\mathbf{e}}_x \quad (4.37)$$

where $r' = \|\mathbf{r}'\|$. All constants are defined in Table 4.1.

In addition to the nominal dynamics given above, the stochastic system will also include additive noise and control-linear noise. Additive noise is included along each axis with a standard deviation of σ_{add} to account for dynamical mismodeling. A fourth additive noise process is included along the $\hat{\mathbf{e}}_x$ axis for SRP acceleration uncertainty with a standard deviation of σ_{SRP} . Accordingly, G is defined by

$$G = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 1} \\ \sigma_{add} I_3 & \begin{bmatrix} \sigma_{SRP} \\ 0_{2 \times 1} \end{bmatrix} \end{bmatrix}. \quad (4.38)$$

The stochastic dynamics now become

$$d\mathbf{r}_t = d\mathbf{v}_t dt, \quad (4.39)$$

$$d\mathbf{v}_t = -2\Omega\hat{\mathbf{e}}_z \times \mathbf{v}_t dt + \Omega^2(3\hat{\mathbf{e}}_x\hat{\mathbf{e}}_x^\top - \hat{\mathbf{e}}_z\hat{\mathbf{e}}_z^\top)dt - \frac{\mu_A}{r_t^3}\mathbf{r}_t dt + a_{SRP}\hat{\mathbf{e}}_x + GdW_t, \quad (4.40)$$

where $r_t = \|\mathbf{r}_t\|$. Let $\mathbf{x}_t := \mathbf{X}_t - \mathbf{X}'(t)$. As described in Section 4.3.1, the dynamics of \mathbf{x}_t are approximated by the stochastic linear system in Eq. (4.9), with $A(t)$ given by:

$$A(t) := \left[\begin{array}{cc} 0_{3 \times 3} & I_3 \\ \frac{\partial(d\mathbf{v}/dt)}{\partial \mathbf{r}} & \frac{\partial(d\mathbf{v}/dt)}{\partial \mathbf{v}} \end{array} \right] \Big|_{\mathbf{X}=\mathbf{X}'(t)}. \quad (4.41)$$

The hybrid system will also include impulsive control inputs. It is assumed that the spacecraft attitude is fixed with respect to the Sun and the spacecraft is equipped with six axially-aligned thrusters in the $\pm\hat{\mathbf{e}}_x$, $\pm\hat{\mathbf{e}}_y$, and $\pm\hat{\mathbf{e}}_z$ directions that produce impulsive velocity changes. In this configuration, a maneuver may require the use of up to three thrusters simultaneously. These will be control inputs $\Delta\mathbf{u}_{k,j}$ added to the stochastic dynamics. The corresponding control matrices are given by:

$$B_0 = \begin{bmatrix} 0_{3 \times 3} \\ I_3 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0_{3 \times 3} \\ \varepsilon_{\parallel} & 0 & 0 \\ \varepsilon_{\perp} & 0 & 0 \\ \varepsilon_{\perp} & 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0_{3 \times 3} \\ 0 & \varepsilon_{\perp} & 0 \\ 0 & \varepsilon_{\parallel} & 0 \\ 0 & \varepsilon_{\perp} & 0 \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0_{3 \times 3} \\ 0 & 0 & \varepsilon_{\perp} \\ 0 & 0 & \varepsilon_{\perp} \\ 0 & 0 & \varepsilon_{\parallel} \end{bmatrix}, \quad (4.42)$$

where ε_{\parallel} and ε_{\perp} dictate parallel and perpendicular control-linear noise and are given in Table 4.1. Thus, B_1 , B_2 , and B_3 serve to inject uncertainty parallel and perpendicular to the axis of each thruster when the thruster is in use.

The nominal trajectory will be a frozen Sun-terminator orbit [21] with dynamics (4.37). The initial state of the nominal orbit is $\mathbf{X}'(t_0) = [19.8, 912, 0, 0, 0, 0.0772]^\top$ (m,m/s), assuming $t_0 = 0$ and $t_f = 24$ hr. A target of $\mathbf{z}_f = 0$ is chosen to reduce mean squared deviation about the nominal orbit, and $R = 0$ is chosen to omit the control penalty term in Eq. (4.2). To facilitate unbiased minimization of both position and velocity deviations when minimizing the mean squared deviation of \mathbf{x}_{t_f} , \mathbf{x}_t is scaled such that the nominal position and velocity states have the same order of magnitude. Accordingly, the problem

Table 4.1: Terminator Orbit Parameters for Robust Guidance

| Symbol | Description | Value | Units |
|---------------------------|---|--------------------------|-----------|
| Ω | Rate of coordinate frame rotation | 2×10^{-7} | s^{-1} |
| μ_A | Asteroid gravitational parameter | 5 | m^3/s^2 |
| a_{SRP} | SRP acceleration magnitude | 1.064×10^{-4} | mm/s^2 |
| σ_{add} | Std. dev. of additive noise | $10^{-3} \mu_A / r_0'^2$ | m/s^2 |
| σ_{SRP} | Std. dev of a_{SRP} | $10^{-2} a_{SRP}$ | mm/s^2 |
| ε_{\parallel} | Along-axis $\Delta \mathbf{u}$ uncertainty factor | 10% | - |
| ε_{\perp} | Off-axis $\Delta \mathbf{u}$ uncertainty factor | 5% | - |

is nondimensionalized by r_0' in the length scale (r_0' is the nominal orbit radius at t_0) and $\sqrt{r_0'^3/\mu_A}$ in the time scale before executing Algorithm 3. This scaling also ensures that W_k is well-conditioned and can be inverted without large numerical error. The original scaling is used in the following results.

First, a single sample path is simulated with $N = 3$, $p_k = 6$ for all k , and the large initial state deviation $\mathbf{x}_{t_0} = [-200, 100, 200, 10, -5, -100]^T$ (m, mm/s). The measurements and control impulses are evenly distributed. The initial state \mathbf{x}_{t_0} is not included in N . Thus, a measurement occurs every $t_f/(N + 1) = 6$ hr. The result is shown in Figure 4.2. Although the linear model is used to compute the optimal controls, both red and blue trajectories in Figure 4.2 are sample paths generated by the nonlinear stochastic

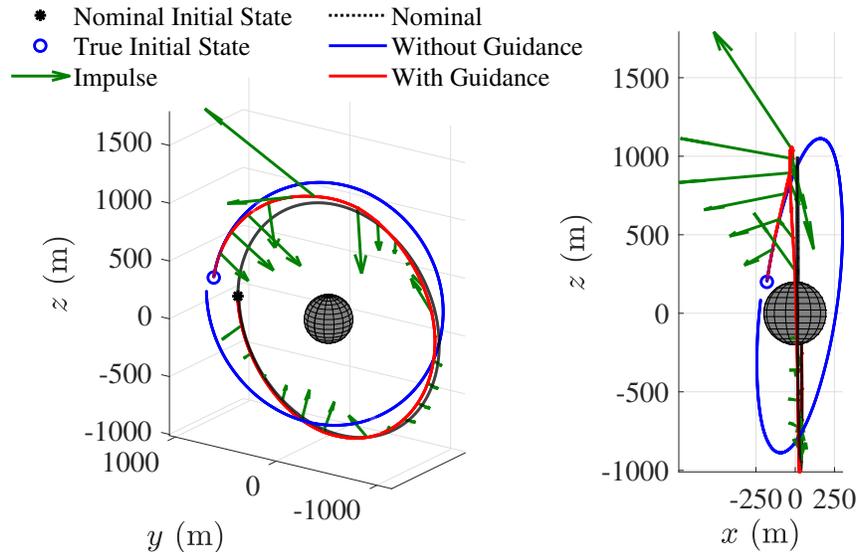


Figure 4.2: Algorithm 3 is demonstrated for guidance about a nominal asteroid orbit ($N = 3$, $p_k = 6$). Red and blue trajectories are sample paths generated by the nonlinear stochastic dynamics. Impulses are scaled for visibility.

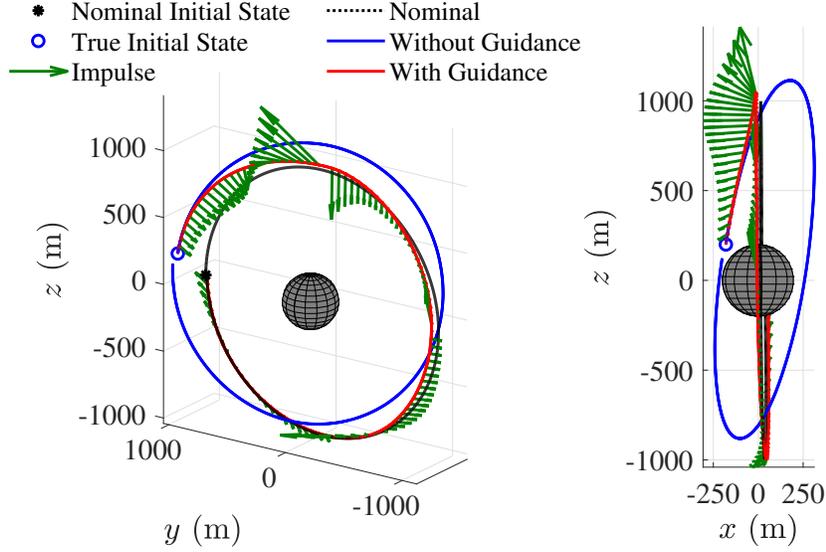


Figure 4.3: The simulation in Figure 4.2 is repeated for $(N = 3, p_k = 25)$.

dynamics. The optimal control requires an impulsive velocity change of 6.23 cm/s in total and achieves a final mean state deviation of $\bar{\mathbf{x}}^*(t_f) = [3.0 \times 10^{-3}, 1.1, -0.66, 4.3 \times 10^{-3}, -0.035, -0.026]$ (m, mm/s) about the nominal trajectory as computed from Eq. (4.6).

The simulation is then repeated for $N = 3$ and $p_k = 25$, and the result is shown in Figure 4.3. The $p_k = 25$ case requires 7.5 cm/s of impulsive control and results in a final mean state deviation of $\bar{\mathbf{x}}^*(t_f) = [0.059, 0.20, -0.19, -4.1 \times 10^{-3}, -5.4 \times 10^{-3}, 7.8 \times 10^{-3}]$ (m, mm/s).

Next, a broader numerical study is conducted by repeating Algorithm 1 for multiple values of $N \in [0, 25]$ and $p_k \in [1, 25]$, assuming p_k are the same across all segments. Because the performance is dependent upon the initial state deviation \mathbf{x}_{t_0} , a Monte Carlo simulation at each combination of (N, p_k) is conducted by randomly selecting 100 values of \mathbf{x}_{t_0} . For the purpose of the simulation, initial variances of 2500 m² in position and 100 mm/s² in velocity are considered for \mathbf{x}_{t_0} (off-diagonal terms in the covariance matrix are zero). Figure 4.4 shows a contour plot of $E[J^*]$ for each combination of (N, p_k) , in which J^* is computed from Theorem 3. The results indicate that as N and/or p_k increase, $\mathbb{E}[J^*]$ decays asymptotically.

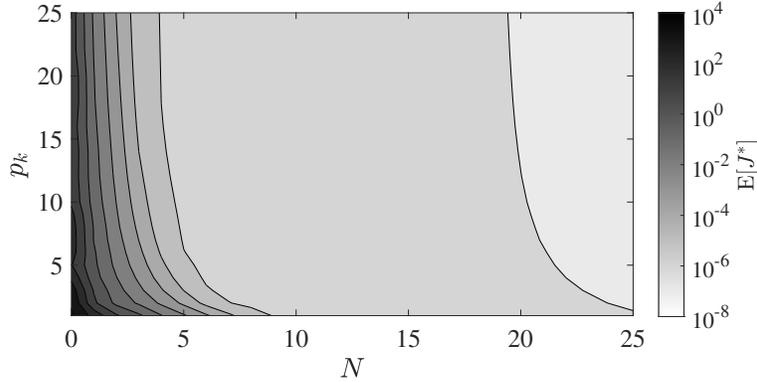


Figure 4.4: Numerical simulation of $\mathbb{E}[J^*]$ vs. N and p_k for guidance about an asteroid orbiting trajectory. $\mathbb{E}[J^*]$ asymptotically decays to a positive value as (N, p_k) are increased.

4.5 Deterministic Guidance Comparison: Modified LQR

In this section, the robust guidance law will be compared to a deterministic feedback control law — an LQR that is configured to have the same measurement and control schedules as the robust guidance law. The deterministic control law will be referred to as a “modified LQR” or simply “LQR” throughout this section. The robust guidance law and the modified LQR will be compared for two mission scenarios: near-rectilinear halo orbits (NRHOs) in the Earth-Moon system and asteroid Sun-terminator orbits. Both orbits are detailed further in Appendix B.

The southern L_2 9:2 synodic resonant NRHO is of particular interest, as it represents the planned orbit for the NASA Lunar Gateway [45]. As a crewed spacecraft, the Gateway will benefit from extensive navigation and maneuver design resources. Proposed station-keeping methods for the Gateway include x -axis crossing and Cauchy-Green Tensor targeting. Both methods involve a human in-the-loop or are computationally expensive for onboard implementation [46]. Conversely, other spacecraft in nearby orbits are likely to be budgeted fewer resources than the Gateway and may also suffer from larger orbit insertion errors and less precise actuators. These lesser spacecraft are candidates for robust, autonomous guidance. Because the robust guidance law is a linear feedback controller with feedback gains that can be computed offline, the guidance law has potential for autonomous, onboard implementation. Asteroid orbiters may also benefit from autonomous guidance due to long transmission delays between asteroids and the Earth.

These applications are chosen for two major reasons: 1) they are relevant to current and planned space missions and 2) they experience very different dynamics. In the NRHO case, the orbital dynamics are strongly nonlinear near perilune and relatively linear near apolune [47]. Guidance on NRHOs may be highly sensitive to navigation and control scheduling, i.e., the control performance may depend on the proximity of these events to perilune. However, the dynamics of the asteroid Sun-terminator orbit are similar to that of a circular orbit, and the degree of nonlinearity is fairly constant throughout. Thus, terminator orbit guidance is likely less sensitive to event scheduling.

By design, the robust guidance law will outperform the LQR in the presence of control-dependent noise, using mean squared deviation as the performance metric. However, the relative improvement of the robust guidance law depends on many factors, including the contribution of control noise when compared to other sources of uncertainty and the accuracy of the linearized dynamics model.

The hybrid system dynamics are given by Eq. (4.8). The most significant variation in this section is the inclusion of navigation uncertainty in the control simulation. The algorithm is unchanged from Algorithm 3, but navigation uncertainty will be included in the numerical simulation to investigate the effect of imperfect navigation on the control performance. It is assumed that navigation updates (i.e., state estimates) are generated via an external, unspecified estimation process. Let $\hat{\mathbf{X}}_{t_k}$ be the estimated system state at time t_k . We assume that the controller receives a total of N navigation updates, which are scheduled a priori and discretize the trajectory into segments. The period between two state updates $\hat{\mathbf{X}}_{t_k}$ and $\hat{\mathbf{X}}_{t_{k+1}}$ is still referred to as segment k . During segment k , a total of p_k impulsive control inputs $\Delta \mathbf{u}_{k,j}$ will be applied as before. Using the methodology in Section 4.3.1, the dynamics of \mathbf{x}_t are approximated by the following stochastic linear system:

$$\begin{cases} d\mathbf{x}_t \approx A(t)\mathbf{x}_t dt + G(t)dW_t, \\ \mathbf{x}_{t_{k,j}}^+ = \mathbf{x}_{t_{k,j}}^- + \left[B_0(t_{k,j}) + \sum_{i=1}^{\ell} B_i(t_{k,j})e_i \right] \Delta \mathbf{u}_{k,j}, \\ \hat{\mathbf{x}}_{t_k} = \hat{\mathbf{X}}_{t_k} - \mathbf{X}'(t_k), \end{cases} \quad (4.43)$$

where the linear dynamics matrix is evaluated along the nominal trajectory: $A(t) := \partial \mathbf{f}(\mathbf{X}, t) / \partial \mathbf{X} |_{\mathbf{X}=\mathbf{X}'(t)}$.

In this section, the robust guidance algorithm will be applied to the case of imperfect state estimates

by simply replacing each perfect, full-state measurements with the imperfect state estimate at time t_k :

$$(\Delta \mathbf{u}_{k,j}^*) \approx K_{k,j} \hat{\mathbf{x}}_{t_k} + L_{k,j} \mathbf{z}_f. \quad (4.44)$$

Although the robust guidance algorithm assumes perfect, full-state measurements, the following results will show that good performance can still be achieved with imperfect state estimates.

In the following examples, stochastic sample paths are generated from the nonlinear dynamics. Dynamics are propagated with a variable-step Runge-Kutta 78 (7th-order integrator with an 8th-order error estimate). The relationship in Eq. (2.52) is used to approximate continuous additive noise in discrete time.

4.5.1 Modified LQR

In order to demonstrate the advantage of stochastic control, the robust guidance law will be compared to a modified LQR that has no knowledge of the system's additive or control-dependent noise. The modified LQR assumes the following dynamics:

$$\begin{cases} \frac{d\tilde{\mathbf{x}}}{dt} = A(t)\tilde{\mathbf{x}}(t), \\ \tilde{\mathbf{x}}_{t_{k,j}}^+ = \tilde{\mathbf{x}}_{t_{k,j}}^- + B_0(t_{k,j})\Delta \mathbf{u}_{k,j}, \end{cases} \quad (4.45)$$

where $\tilde{\mathbf{x}}$ is a fictitious state deviation from the nominal trajectory (assuming deterministic dynamics) that is used in the derivation of the modified LQR solution. The LQR cost function is given below.

$$J_{Lqr} := \|\tilde{\mathbf{x}}(t_f) - \mathbf{z}_f\|^2 + \sum_{k=0}^N \sum_{j=1}^{p_k} \Delta \mathbf{u}_{k,j}^\top R \Delta \mathbf{u}_{k,j} \quad (4.46)$$

To enable a one-to-one comparison of both control laws, Eq. (4.46) is modified from the classical finite-horizon LQR cost to have a target state of \mathbf{z}_f , as well as the same measurement and control schedules as the robust guidance algorithm. Moving forward, this setup will simply be referred to as the ‘‘LQR’’ although it is slightly different than the standard LQR problem. The optimal LQR control inputs can also be derived from dynamic programming:

$$(\Delta \mathbf{u}_{k,j}^*)_{Lqr} = \tilde{K}_{k,j} \tilde{\mathbf{x}}(t_k) + \tilde{L}_{k,j} \mathbf{z}_f \quad (4.47)$$

$$\approx \tilde{K}_{k,j} \hat{\mathbf{x}}_{t_k} + \tilde{L}_{k,j} \mathbf{z}_f \quad (4.48)$$

Algorithm 4: Offline computation of state feedback gains.

Data: Matrices $A(t)$, R , measurement times t_1, \dots, t_N , impulse times $t_{k,1}, \dots, t_{k,p_k}$, for all $k = 0, \dots, N$, and final time $t_f =: t_{N+1}$.

Result: Matrices $\tilde{K}_{k,j}$ and $\tilde{L}_{k,j}$ for all $k = 0, \dots, N$ and $j = 1, \dots, p_k$.

initialize; $\tilde{W}_{N+1} := I$; $\tilde{C}_{N+1} := I$;

for $k := N$ **to** 0 **do**

$$\tilde{Q}_k := \sum_{j=1}^{p_k} \Phi_{k,j} B_0(t_{k,j}) R^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top;$$

$$\tilde{W}_k := \Phi_k^\top [\tilde{W}_{k+1}^{-1} + \tilde{Q}_k]^{-1} \Phi_k;$$

$$\tilde{C}_k := \Phi_k^\top [I + \tilde{W}_{k+1} \tilde{Q}_k]^{-1} \tilde{C}_{k+1};$$

$$\tilde{K}_{k,j} := -R^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top [\tilde{W}_{k+1}^{-1} + \tilde{Q}_k]^{-1} \Phi_k;$$

$$\tilde{L}_{k,j} := R^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top [I + \tilde{W}_{k+1} \tilde{Q}_k]^{-1} \tilde{C}_{k+1};$$

end

As in Eq. (4.44), the fictitious state $\tilde{\mathbf{x}}(t_k)$ is replaced with the best state estimate at that time. The control gain matrices $\tilde{K}_{k,j}$ and $\tilde{L}_{k,j}$ are computed via the recursive equations in Algorithm 4. Note that neglecting additive noise in Eq. (4.45) does not disadvantage the LQR with respect to the robust guidance law. In the robust guidance law, the additive noise only serves to inflate the mean squared deviation (i.e., to increase the cost function), but does not alter the optimal controls (see Algorithm 3).

4.5.2 Control Midpoints

The robust and LQR guidance laws are derived with a fixed final time. When applying guidance over a long horizon, the trajectory may be segmented into arcs of shorter duration. For example, a trajectory may be controlled with a final time of t_f in Algorithm 3. Alternatively, Algorithm 3 could be applied to two separate arcs, such as $[t_0, t_f/2]$ and $[t_f/2, t_f]$. The latter example corresponds to one control midpoint at $t_f/2$. By including control midpoints in this manner, the guidance laws will minimize state deviations at intermediate points along the trajectory, as well as the final time. Note that the nominal orbit does not change in this case, and all control gain matrices can still be computed offline regardless of the number of control midpoints.

Table 4.2: NRHO Baseline Values

| Initial state error 3σ | Navigation error 3σ | Additive noise, σ_{add} | Control-linear noise, $(\varepsilon_{\parallel}, \varepsilon_{\perp})$ | R |
|-------------------------------|----------------------------|--------------------------------|--|----------------|
| 10 km, 10 cm/s | 10 m, 1 mm/s | 10^{-10} m/s^2 | 5%, 1.5% | $10^{-12} I_3$ |

4.5.3 NRHO Application

In this section, robust and LQR guidance laws will be applied to an NRHO modeled in the CR3BP. The nominal orbit is a southern L_2 9:2 synodic resonant NRHO in the Earth-Moon system (representative of the Lunar Gateway orbit). The CR3BP dynamics and the 9:2 NRHO are detailed further in Appendix B. Baseline uncertainty values are given in Table 4.2. It is assumed that

$$G = \begin{bmatrix} 0_{3 \times 3} \\ \sigma_{add} I_3 \end{bmatrix}, \quad (4.49)$$

and B_i matrices are given in Eq. (4.42).

Four NRHO mission scenarios are summarized in Table 4.3. All simulations begin with an initial state at apolune (farthest from the Moon). In each case, it is assumed that impulsive controls are applied 12 hours after navigation updates. Scenario 1 involves short horizon guidance over two orbits (approximately two weeks) with two navigation updates/controls per orbit. The navigation and control schedule for this scenario can be seen in Figure 4.5 (left). In Scenario 1, events are not scheduled near perilune. Scenario 2 involves short horizon guidance with three navigation updates and controls per orbit. In this case, a navigation update occurs at perilune, where dynamics are strongly nonlinear. The navigation and control schedule for the second scenario can also be seen in Figure 4.5 (right). Scenario 3 involves long horizon guidance over eight orbits (approximately 2 months) with two navigation updates/controls per orbit and without control midpoints (refer to Section 4.5.2). Finally, Scenario 4 involves long horizon guidance over eight orbits with two navigation updates/controls per orbit and three control midpoints at $t_f/4$, $t_f/2$, and $3t_f/4$. Thus, the control algorithms are applied to the time segments $[t_0, t_f/4]$, $[t_f/4, t_f/2]$, $[t_f/2, 3t_f/4]$, and $[3t_f/4, t_f]$ independently. Scenarios 3 and 4 are also illustrated by Figure 4.5 (left).

Table 4.3 lists the final mean squared deviation and mean ΔV of each mission scenario, computed via Monte Carlo simulation. Figure 4.7 shows final deviations from the nominal state for each mission scenario.

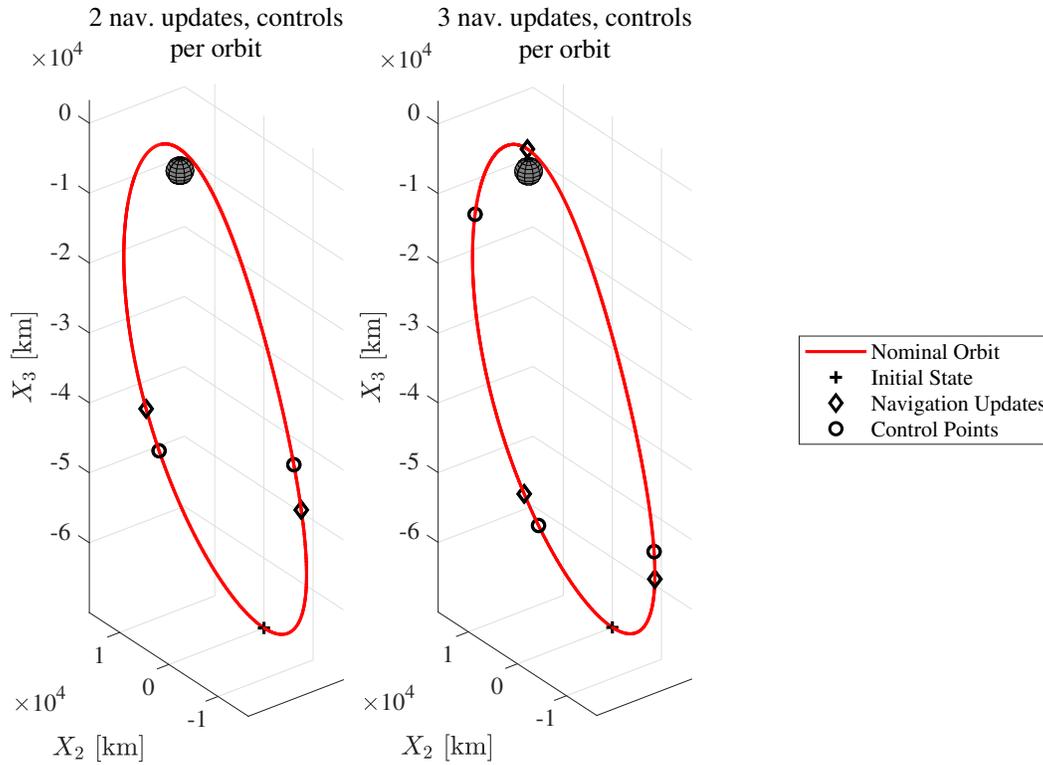


Figure 4.5: Navigation update and control scheduling for NRHO Scenarios 1 through 4. The spacecraft is orbiting counter-clockwise from this perspective.

In terms of mean squared deviation, the robust guidance law outperforms the LQR in each scenario, with varying degrees of improvement. The results are further described below:

Scenario 1: The robust guidance law significantly reduces the mean squared deviation at the cost of roughly 4 cm/s of ΔV , when compared to the LQR.

Scenario 2: In this scenario, events are scheduled in the highly sensitive region near perilune. The robust guidance law reduces mean squared deviation by three orders of magnitude, corresponding to an improvement of nearly one order of magnitude in state error (non-squared deviations). This increased robustness also results in significant ΔV improvement. These results suggest that the robust guidance law performs better in regions with sensitive, strongly nonlinear dynamics. This characteristic of the robust guidance law will be discussed further in the next section. Figure 4.6 shows one sample path for Scenario 2 (ΔV s are plotted for visibility and are not true to scale). It is clear in Figure 4.6 that the robust and LQR

Table 4.3: NRHO Mission Scenarios and Results for 1000-trial Monte Carlo

| Scenario: | 1 | 2 | 3 | 4 |
|------------------------------------|------------------------|----------------------------------|-----------------------|---------------------------------|
| Description: | Short horizon | Short horizon, meas. at perilune | Long horizon | Long horizon, control midpoints |
| # of orbits | 2 | 2 | 8 | 8 |
| Control midpoints (see Sec. 4.5.2) | - | - | - | $t_f/4, t_f/2, 3t_f/4$ |
| # of nav. updates & controls | 4 | 6 | 16 | 16 |
| Mean ΔV , robust (cm/s) | 9.75 | 25.5 | 3.54 | 137 |
| Mean ΔV , LQR (cm/s) | 5.96 | 202 | 3.27 | 138 |
| Mean squared deviation, robust | 2.30×10^{-10} | 8.21×10^{-10} | 2.37×10^{-8} | 5.48×10^{-12} |
| Mean squared deviation, LQR | 1.75×10^{-9} | 6.74×10^{-7} | 4.10×10^{-8} | 1.72×10^{-8} |

guidance laws produce very different solutions.

Scenario 3: This is the long horizon case without control midpoints. The robust guidance law provides a relatively small improvement in mean squared deviation when compared to the LQR. No control midpoints are included in Scenario 3, so state deviations are only minimized at the final time.

Scenario 4: This is the long horizon case with three control midpoints. In this case, the guidance laws will minimize state deviations at the final time, as well as three intermediate times (i.e., every two orbits). The robust guidance law outperforms the LQR significantly — robust guidance reduces mean squared deviation by four orders of magnitude, corresponding to two orders of magnitude of improvement in state error (non-squared deviations). The ΔV is comparable for both methods.

Next, both guidance laws are analyzed as the state error, navigation error, additive noise standard deviation, and control-linear noise factors are scaled in Scenario 1. The Monte Carlo simulation is repeated as the scaling parameters in Table 4.2 are varied. Figure 4.8 shows the mean squared deviation as each parameter is varied individually. The robust guidance law continues to outperform the LQR in terms of mean squared deviation the initial state error, control noise, and additive noise are scaled. The robust guidance law is notably more robust to initial state error. In the case of navigation error, however, the robust guidance law fails to outperform the LQR for large values of navigation error — this is the point at which the perfect navigation assumption (and the approximation in Eq. (4.44)) begins to break down. Figure 4.9 shows the ΔV performance of both guidance laws as the parameters in Table 4.2 are increased. In mission Scenario 1, the LQR consistently outperforms the robust guidance law in terms of ΔV .

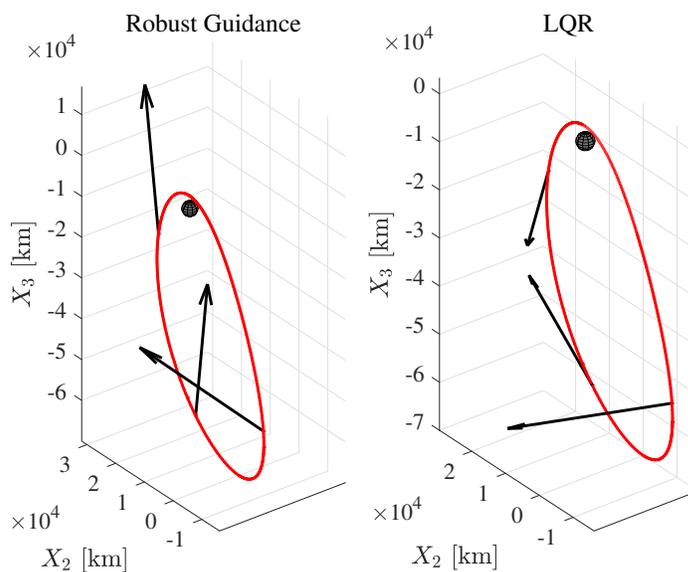


Figure 4.6: Sample paths for one orbit in NRHO Scenario 2. ΔV vectors are not true to scale.

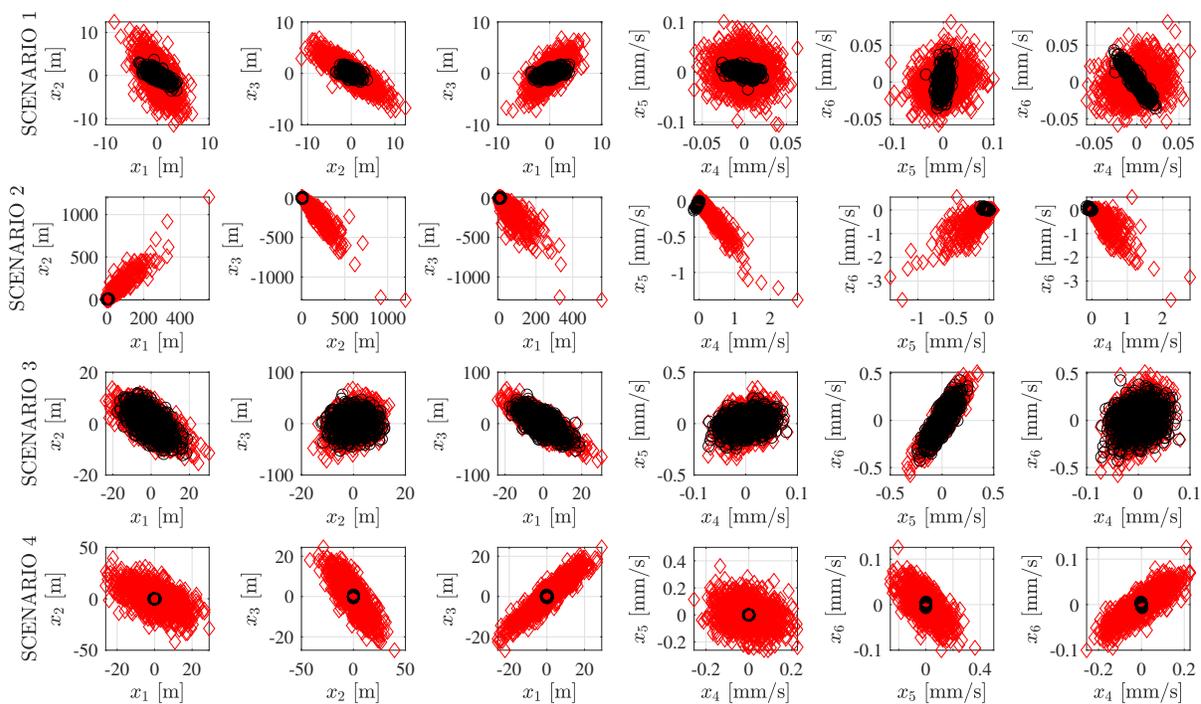


Figure 4.7: Final state deviations of robust guidance (black circles) and LQR (red diamonds) for NRHO Scenarios 1-4, computed from 1000-trial Monte Carlo simulations.

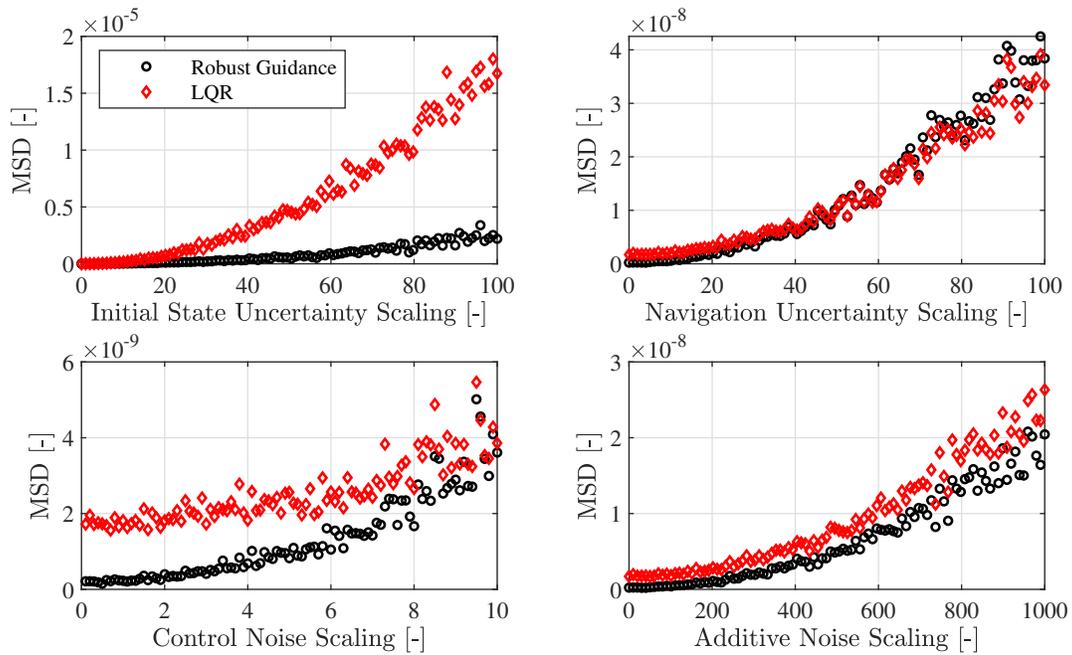


Figure 4.8: Mean squared deviation (MSD) of robust guidance and LQR sample paths as we scale the parameters in Table 4.2, computed from 100-trial Monte Carlo simulations of Scenario 1.

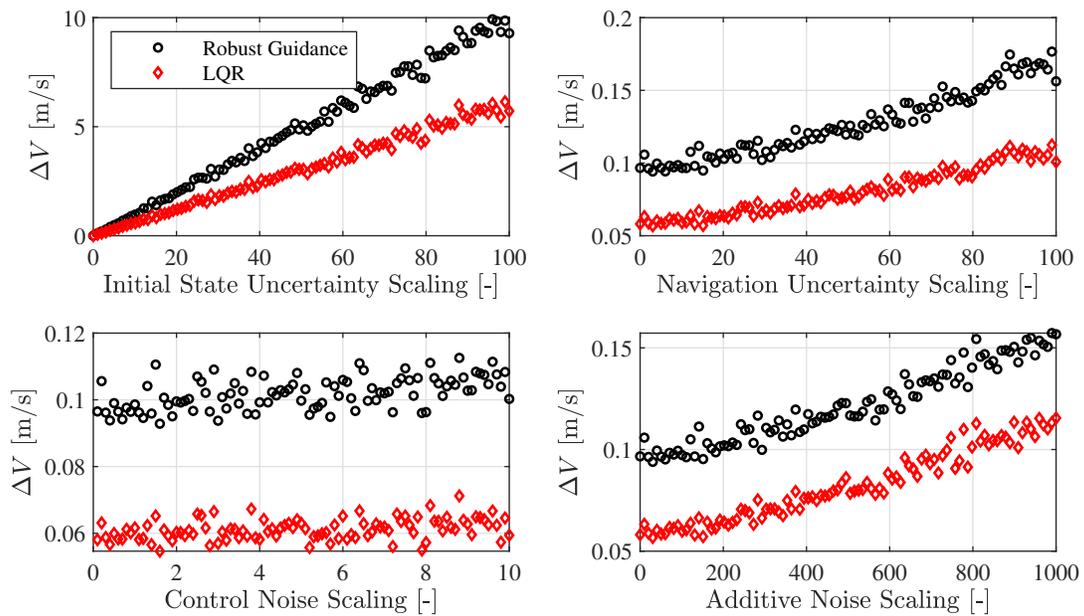


Figure 4.9: Mean ΔV of robust guidance and LQR sample paths as we scale the parameters in Table 4.2, computed from 100-trial Monte Carlo simulations of Scenario 1.

Table 4.4: Terminator Orbit Baseline Values

| Description | Value |
|--|--|
| Gravitational parameter, μ_A | $5 \text{ m}^3/\text{s}^2$ |
| Initial semimajor axis | 1500 m |
| Initial eccentricity | 0.0684 |
| Initial inclination | 0 rad |
| Initial RAAN | $-\pi/2$ rad |
| Initial argument of periapsis | $\pi/2$ rad |
| Initial true anomaly | 0 rad |
| Orbit period, T_P | 1.9 days |
| Coordinate frame rotation rate, Ω | $2 \times 10^{-7} \text{ s}^{-1}$ |
| SRP acceleration magnitude, a_{SRP} | $1.064 \times 10^{-4} \text{ mm/s}^2$ |
| Initial state error 3σ | 1 m, 0.1 mm/s |
| Navigation error 3σ | 1 m, 0.1 mm/s |
| Additive noise, σ_{add} | 0.001% ($\mu_A/r_0'^2$) m/s^2 |
| SRP noise, σ_{SRP} | 0.1% $a_{SRP} \text{ mm/s}^2$ |
| Control-linear noise, $(\varepsilon_{\parallel}, \varepsilon_{\perp})$ | 5%, 1.5% |
| R | $10^{-12} I_3$ |

4.5.4 Asteroid Sun-Terminator Orbit Application

The next example considers a spacecraft orbiting a small near-Earth asteroid modeled in the H3BP with SRP. All constants are defined in Table 4.4, and are chosen to be representative of the OSIRIS-REx mission to the asteroid Bennu [8, 30, 31]. The nominal orbit is a Sun-terminator orbit with a semimajor axis of roughly 1500 m. The initial state is computed using the shifted origin and eccentricity corrections in [21], and the orbital elements of the initial state are given in Table 4.4. It is assumed that G is given in Eq. (4.38) and B_i matrices are given in Eq. (4.42). Before applying Algorithm 3 and Algorithm 4, the dynamics are nondimensionalized by $r'(t_0)$ in the length scale and $\sqrt{r'(t_0)^3/\mu_A}$ in the time scale. Scaling the dynamics enables unbiased minimization of position and velocity state deviations and facilitates numerical stability in Algorithms 3 and 4.

Monte Carlo simulations with 1000 trials are conducted for the robust and LQR asteroid guidance scenarios. The baseline values for all scenarios are given in Table 4.4. As in the previous example, navigation updates have a zero-mean state error. The 3σ initial state error and navigation errors in Table 4.4 are comparable to simulations of autonomous asteroid guidance and navigation in [48]. Control-linear noise

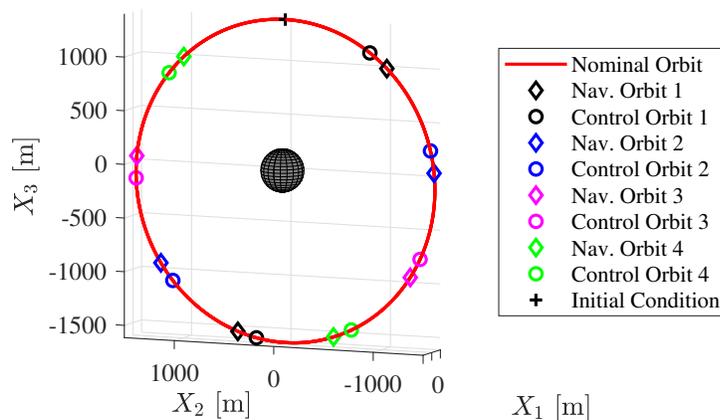


Figure 4.10: Eight evenly-spaced navigation updates/controls over a period of four orbits (one week scenario). In the one month scenario, this schedule is applied four consecutive times.

with 5% along-axis uncertainty and 1.5% cross-axis uncertainty is also included — these values are comparable to those observed on the OSIRIS-REx mission [49]. The control weighting matrix, R , will be set to $10^{-12}I_3$ to prioritize the minimization of state deviations rather than control effort.

The two asteroid mission scenarios are summarized in Table 4.5. In each case, the controls are applied one hour after navigation updates. The first scenario occurs over a period of four orbits (roughly one week) with eight navigation updates/controls and without control midpoints. The controls are evenly spaced over the one week period. The second scenario occurs over a period of 16 orbits (roughly one month) with a control midpoint placed every four orbits (at $4T_P$, $8T_P$, and $12T_P$, where T_P is the orbit period) and eight evenly-spaced navigation updates/controls during each of the following segments: $[0, 4T_P]$, $[4T_P, 8T_P]$, $[8T_P, 12T_P]$, and $[12T_P, 16T_P]$. Thus, the one month scenario is equivalent to repeating the one week scenario four consecutive times. The navigation and control schedules for either scenario are shown in Figure 4.10.

Table 4.5 lists the final mean squared deviation and mean ΔV of each mission scenario, computed via Monte Carlo simulation. Table 4.5 also provides statistics for the number of Monte Carlo samples that

Table 4.5: Terminator Orbit Mission Scenarios and Results for 1000-trial Monte Carlo

| Scenario: | 1 | 2 |
|---------------------------------------|-----------------------|-----------------------|
| Description: | ~ 1 week | ~ 1 month |
| Number of orbits | 4 | 16 |
| Control midpoints (see Sec. 4.5.2) | 0 | $4T_P, 8T_P, 12T_P$ |
| Number of nav. updates & controls | 8 | 32 |
| Mean ΔV , robust (mm/s) | 1.14 | 5.42 |
| Mean ΔV , LQR (mm/s) | 0.763 | 4.00 |
| Mean squared deviation, robust | 7.66×10^{-5} | 5.27×10^{-3} |
| Mean squared deviation, LQR | 7.95×10^{-5} | 0.400 |
| Samples that escape, robust | 0% | 0.4% |
| Samples that escape, LQR | 0% | 34% |

escape the asteroid's orbit, i.e., the percentage of failed cases. Figure 4.11 shows final deviations about the nominal mean state for each mission scenario. The performance in each scenario is further described below:

Scenario 1: In the one week scenario, the robust guidance law provides a marginal improvement in mean squared deviation; this small improvement likely does not justify the increased propellant cost of the robust guidance law. Thus, it can be concluded that control-linear noise is not a dominant source of uncertainty in this scenario. Note that no Monte Carlo sample paths escaped the asteroid system for both methods.

Scenario 2: In the one month scenario, however, 34% of LQR Monte Carlo trials escaped the asteroid's orbit, whereas only 0.4% of robust guidance trials escaped. Therefore, control-dependent noise has a significant affect on performance in Scenario 2, and the robust guidance law is far superior to the LQR in this scenario.

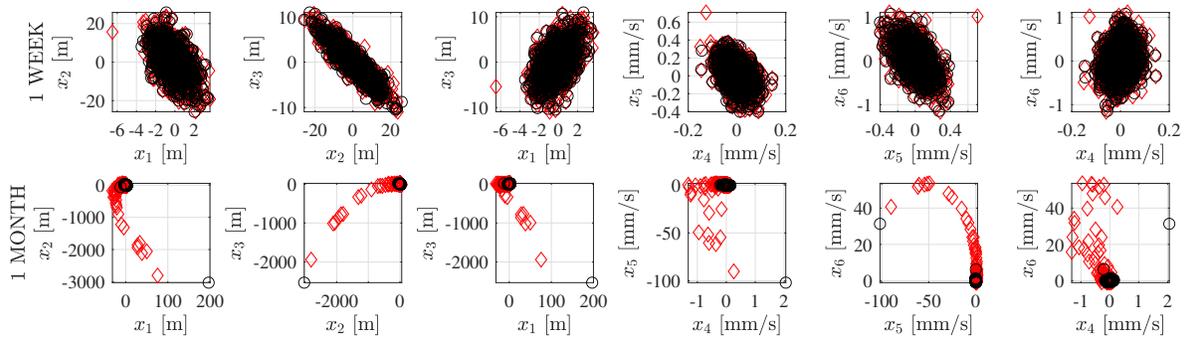


Figure 4.11: Final state deviations of robust guidance (black circles) and LQR (red diamonds) for asteroid Sun-terminator orbits, computed from 1000-trial Monte Carlo simulations.

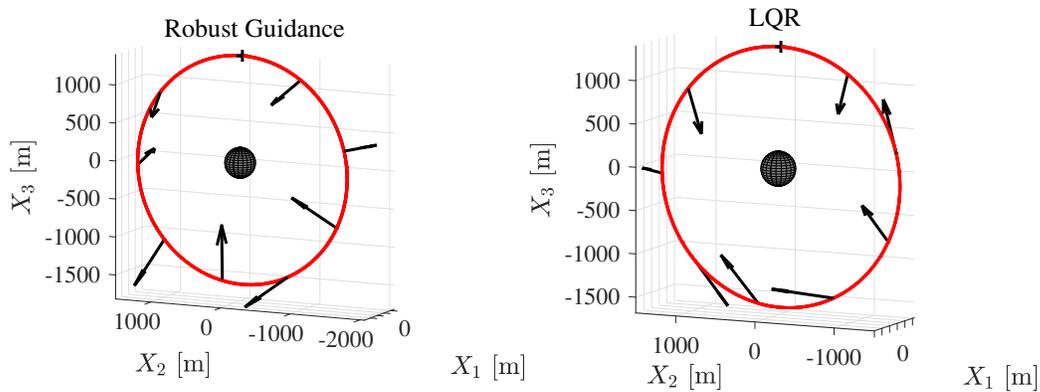


Figure 4.12: Sample paths for asteroid orbit one week scenario.

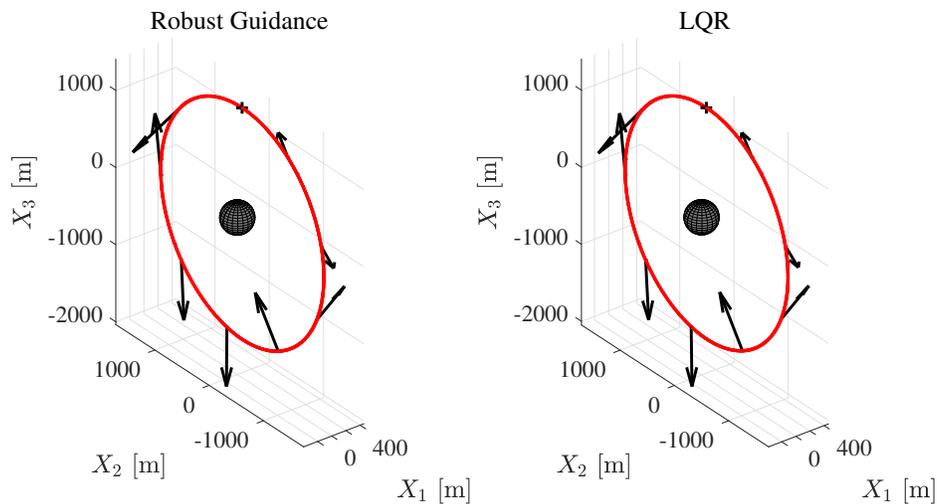


Figure 4.13: Sample paths of one week scenario with $R = I_3$. The robust and LQR control solutions are nearly identical when the propellant cost is prioritized.

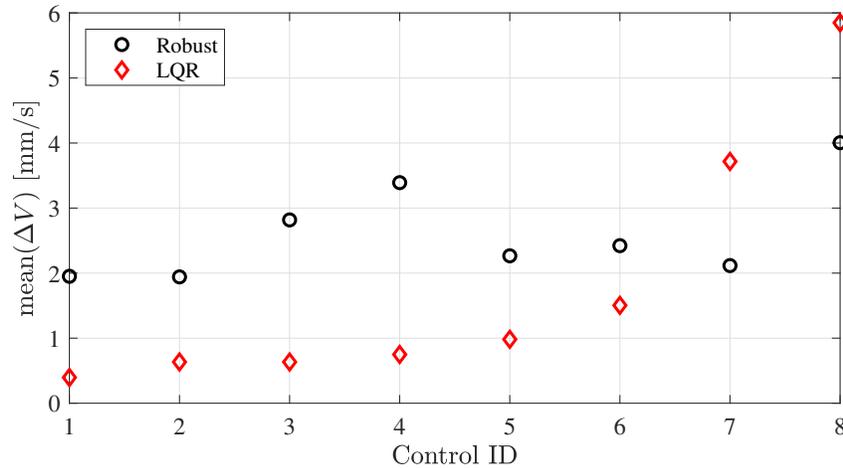


Figure 4.14: Mean ΔV of each control maneuver in the asteroid one week scenario, computed from 1000 Monte Carlo trials. The LQR concentrates control effort toward the end of the trajectory

Figure 4.12 shows one sample path of the one week scenario — it is clear that the robust and LQR algorithms produce very different control laws. Additionally, Figure 4.13 shows a simulation where the control penalty matrix R is increased from $10^{-12}I_3$ to I_3 . In this case, the influence of control-dependent noise in the robust guidance law (Algorithm 3) is dominated by R , and the robust guidance solution is indistinguishable from the LQR.

Finally, it is instructive to visualize the distribution of control effort for each guidance law. The one week guidance scenario includes a total of eight individual controls. In Figure 4.14, the mean magnitude of each control impulse is computed from the Monte Carlo simulation. Figure 4.14 shows that the LQR performs larger controls toward the end of the simulation, whereas the robust guidance law performs larger controls earlier in the trajectory. By front-loading the control in this way, robust guidance tends to steer the system closer to the nominal trajectory earlier in the simulation. Immediately driving the system close to the nominal trajectory is advantageous when applying neighboring guidance, as this is more likely to maintain the spacecraft in the linear region and maintain the accuracy of the linear dynamics approximation. This behavior may explain why the robust guidance law produces far better performance than the LQR in the one month scenario (as well as the NRHO Scenario 3), i.e. the LQR solution drifts beyond the linear region and the linearization fails.

4.6 Discussion and Conclusions

This chapter addressed the optimal control of stochastic linear time-varying systems with additive noise, impulsive control inputs, and sampled measurements. Measurement and control schedules were allowed to be asynchronous, and the model accommodates arbitrarily many sources of control-linear noise in any direction. The system model is motivated by, but not exclusive to, applications in space flight dynamics. An optimal feedback control law was derived to steer the system from a known initial state to a final state that minimizes mean squared deviation about a target state. The optimal control is linear in the initial state, the target state, and the measurement outputs, and the state feedback gains can be computed offline, such that the control has the potential for autonomous, on-board implementation.

By linearizing spacecraft dynamics about nominal trajectories, the optimal control law was demonstrated for neighboring guidance about an asteroid Sun-terminator orbit and an NRHO. In both cases, the robust guidance law was compared to a deterministic control law: an LQR that was modified to have the same asynchronous measurement and control schedule as the robust guidance law. Overall, robust guidance outperforms the LQR in terms of mean squared state deviations and is capable of minimizing final state error by orders of magnitude at reasonable ΔV costs. It was also observed that the robust guidance law tends to concentrate control effort toward the beginning of the trajectory, which drives the spacecraft closer to the nominal trajectory earlier in the period — this behavior has the secondary benefit of maintaining the spacecraft in the linear region and the robust guidance law outperformed the LQR significantly when the dynamics of the nominal trajectory were strongly nonlinear.

The applicability of the perfect measurement assumption was also investigated numerically by introducing noisy navigation updates in Monte Carlo simulations. As expected, the perfect measurement assumption breaks down when large navigation error is present. The next chapter delves further into the problem of noisy measurements.

Future work includes the problem of optimal measurement and control scheduling. Additionally, the timing of measurements and controls can also be made non-deterministic, and evaluating the robustness of the guidance law to delayed or missed measurements/control is another area for future work.

Chapter 5

Guidance with Noisy Measurements

Chapter 5 Nomenclature

| | | |
|----------------------------|---|--|
| $A(t)$ | = | linearized dynamics |
| $B_0(t), \dots, B_\ell(t)$ | = | control matrices |
| $D(t)$ | = | measurement error covariance |
| $G(t)$ | = | additive noise matrix |
| $H(t)$ | = | linear measurement model |
| I | = | identity matrix |
| J | = | cost function |
| J' | = | cost function lower bound |
| J'' | = | cost function upper bound |
| $K_{k,j}$ | = | state feedback gain |
| $L_{k,j}$ | = | target state feedback gain |
| ℓ | = | number of control-linear noise sources |
| m | = | dimension of control input |
| n | = | dimension of system state |
| N | = | number of measurements |
| p_k | = | number of controls in segment k |
| R | = | positive definite matrix |

| | | |
|---------------------------|---|--|
| t | = | time |
| t_k | = | time of k th measurement |
| $t_{k,j}$ | = | time of j th control after the k th measurement |
| t_f | = | final time |
| $\Delta \mathbf{u}_{k,j}$ | = | j th impulsive control input after the k th measurement |
| V_k | = | value function |
| dW_t | = | standard Wiener process |
| \mathbf{x}_0 | = | known initial state of linear (or linearized) system |
| \mathbf{x}_t | = | stochastic linear system state at time t |
| \mathbf{y}_t | = | state measurement |
| \mathbf{z}_f | = | target state |
| ν_t | = | measurement error |
| Σ | = | state covariance |
| τ | = | variable of integration |
| Φ | = | state transition matrix from initial to final time, e^{At_f} |
| Φ_k | = | state transition matrix from t_k to t_{k+1} |
| $\Phi_{k,j}$ | = | state transition matrix from $t_{k,j}$ to t_{k+1} |
| Φ_τ | = | state transition matrix from τ to t_{k+1} |

5.1 Introduction

The full-state, noiseless measurement assumption was the key to generating analytical solutions in Chapters 3 and 4. Although the controls were allowed to depend on all previous measurements, the optimal controls were simply linear in the prior measurement when that measurement was full-state and noiseless. However, full-state measurements are rarely available in reality and noiseless measurements are highly improbable, necessitating the inclusion of a state estimation process. In the case of partial-state or noisy measurements with an estimator, it is likely that the optimal controls depend on all past measurements and

are no longer linear in those measurements. Moreover, the “separation principle” for control and estimation no longer holds in the case of control-dependent noise with an estimator. Rather, the control-dependent noise results in a complex dependency between controller and estimator, and analytical solutions are intractable in most cases.

The simplest solution to the noisy measurement problem is to replace each perfect measurement by the best state estimate at that time, as was done in Section 4.5. However, this approach requires time-consuming analysis to guarantee performance, as the guidance law is not optimized under those conditions. In [6], the authors develop a sub-optimal iterative algorithm to preserve the separation principle in the discrete time problem with control-dependent noise and noisy measurements. The control is optimized assuming a fixed uncertainty. The uncertainty is then updated as a function of the control, and the process is repeated until convergence. Although the iterative method may produce better results than if one ignored control-dependent noise entirely, it is sub-optimal because the coupling between control and estimation is neglected during the optimization step.

Alternatively, a bounded cost function approach is proposed in this chapter. Upper and lower bounds for the true cost function, J , will be selected such that

$$J'^* \leq J^* \leq J''^*. \quad (5.1)$$

The bounds J' and J'' are chosen to be quadratic in the control and can be optimized analytically. The bounding method does not produce an optimal control for the true cost. However, J'^* and J''^* provide a metric for evaluating candidate controls.

5.2 Bounded Cost Function Problem Formulation

Consider the linear dynamical system from the previous chapter, but with the noisy, partial state measurements \mathbf{y}_{t_k} :

$$\begin{cases} d\mathbf{x}_t = A(t)\mathbf{x}_t dt + G(t)dW_t, \\ \mathbf{x}_{t_k,j}^+ = \mathbf{x}_{t_k,j}^- + \left[B_0(t_{k,j}) + \sum_{i=1}^{\ell} B_i(t_{k,j})e_i \right] \Delta \mathbf{u}_{k,j}, \\ \mathbf{y}_{t_k} = H(t_k)\mathbf{x}_{t_k} + \nu_{t_k}, \quad 0 < t_1 < t_2 < \dots \end{cases} \quad (5.2)$$

where ν_t is a random vector, and $\mathbb{E}[\nu_t \nu_t^\top] = D(t)$. Assume that H is of rank n . For a linear system with white Gaussian noise, the Kalman filter will be the optimal state estimator.

The bounded cost function approach in this chapter may also be applied to the linearization of a nonlinear system, but the bounds are not guaranteed to be strict bounds in that case, just as the robust guidance law and Kalman filter are not guaranteed to work for the linearized system (especially for large deviations from the reference trajectory).

For compactness in future equations, let

$$\mathcal{H}(t) = H(t)^\top D(t)^{-1} H(t), \quad (5.3)$$

and assume that $\mathcal{H}(t_k)^{-1}$ exists. As before, $\Sigma(t)$ is the state covariance at time t . Allow $\hat{\mathbf{x}}_{t_k}$ to be a state estimate at time t_k , and assume that state estimates and covariance updates are generated from a standard Kalman filter:

$$\Sigma^\oplus(t_k) = \left[(\Sigma^\ominus(t_k))^{-1} + \mathcal{H}(t_k) \right]^{-1}, \quad (5.4)$$

$$\mathcal{K}_k = \Sigma^\ominus(t_k) H(t_k)^\top \left[H(t_k) \Sigma^\ominus(t_k) H(t_k)^\top + D(t_k) \right]^{-1}, \quad (5.5)$$

$$\hat{\mathbf{x}}_{t_k} = \bar{\mathbf{x}}(t_k) + \mathcal{K}_k (\mathbf{y}_{t_k} - H(t_k) \bar{\mathbf{x}}(t_k)), \quad (5.6)$$

where superscripts \ominus/\oplus indicate quantities before/after a Kalman filter update and \mathcal{K}_k is the Kalman gain.

Again, consider the cost function in Eq. (4.2). Some preliminaries are needed before suitable bounds for the cost can be identified. First, the value function at t_N is computed in the same manner as Section 4.3. With the inclusion of the partial, noisy state measurement and Kalman filter update at time t_N , the value function V_N is now a function of the state estimate and state error covariance as follows:

$$\begin{aligned} V_N(\hat{\mathbf{x}}_{t_N}, \Delta \mathbf{u}_N, \mathbf{z}_f, \Sigma^\oplus(t_N)) &= \varphi_{N+1} + \text{tr}(\Sigma^\ominus(t_f) W_{N+1}) + \bar{\mathbf{x}}(t_f)^\top W_{N+1} \bar{\mathbf{x}}(t_f) - 2\bar{\mathbf{x}}(t_f)^\top C_{N+1} \mathbf{z}_f \\ &\quad + \mathbf{z}_f^\top F_{N+1} \mathbf{z}_f + \sum_{j=1}^{p_N} \Delta \mathbf{u}_{N,j}^\top R \Delta \mathbf{u}_{N,j}. \end{aligned} \quad (5.7)$$

This is in contrast to Eq. (4.24), which was only a function of the controls \mathbf{u}_N , the target state \mathbf{z}_f , and the perfect, full-state measurement at time t_N . Following the same first-order optimization process as Sec-

tion 4.3, the optimal value function is given by

$$V_N^* = \hat{\mathbf{x}}_{t_N}^\top W_N \hat{\mathbf{x}}_{t_N} - 2\hat{\mathbf{x}}_{t_N}^\top \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \varphi_N + \text{tr} \left(\Phi_N^\top W_{N+1} \Phi_N \Sigma^\oplus(t_N) \right) \quad (5.8)$$

and the optimal control is linear in the state estimate and target state:

$$\mathbf{u}_{N,j}^* = K_{N,j} \hat{\mathbf{x}}_N + L_{N,j} \mathbf{z}_f. \quad (5.9)$$

The gain matrices during the final segment from t_N to t_f are still computed from Algorithm 3, i.e, relaxing the full-state measurement assumption does not change the gain matrices in the final segment of the trajectory. However, the problem will become intractable as the computation proceeds backwards in time. Recall that

$$V_{N-1}^* = \min_{\Delta \mathbf{u}_{N-1}} \left[\mathbb{E}(V_N^*) + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right]. \quad (5.10)$$

It follows that

$$\begin{aligned} V_{N-1}^*(\hat{\mathbf{x}}_{t_{N-1}}, \Delta \mathbf{u}_{N-1}, \mathbf{z}_f, \Sigma^\oplus(t_{N-1})) &= \min_{\Delta \mathbf{u}_{N-1}} \left[-2\bar{\mathbf{x}}(t_N)^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \varphi_N \right. \\ &\quad \left. + \text{tr} \left(\Phi_N^\top W_{N+1} \Phi_N \Sigma^\oplus(t_N) \right) + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right. \\ &\quad \left. + \mathbb{E} \left\{ [\bar{\mathbf{x}}(t_N) + \mathcal{K}_N(\mathbf{y}_{t_N} - H(t_N)\bar{\mathbf{x}}(t_N))]^\top W_N [\bar{\mathbf{x}}(t_N) + \mathcal{K}_N(\mathbf{y}_{t_N} - H(t_N)\bar{\mathbf{x}}(t_N))] \right\} \right]. \quad (5.11) \end{aligned}$$

With some manipulation, the cost function can be expressed more simply as

$$\begin{aligned} V_{N-1}^* &= \min_{\Delta \mathbf{u}_{N-1}} \left\{ \bar{\mathbf{x}}(t_N)^\top W_N \bar{\mathbf{x}}(t_N) - 2\bar{\mathbf{x}}(t_N)^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \varphi_N + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right. \\ &\quad \left. + \text{tr} (W_N \Sigma^\ominus(t_N)) + \text{tr} \left[\left(\Phi_N^\top W_{N+1} \Phi_N - W_N \right) \Sigma^\oplus(t_N) \right] \right\}. \quad (5.12) \end{aligned}$$

First, note that the covariance immediately after the Kalman filter state update at t_N , i.e., $\Sigma^\oplus(t_N)$, is non-linear in the controls during segment $(N-1)$:

$$\begin{aligned} &\Sigma^\oplus(t_N) \\ &= \left[\left(\Phi_{N-1} \Sigma(t_{N-1})^\oplus \Phi_{N-1}^\top + \Gamma_{N-1} + \sum_{j=1}^{p_{N-1}} \sum_{i=1}^{\ell} \Phi_{N-1,j} B_i \Delta \mathbf{u}_{N-1,j} \Delta \mathbf{u}_{N-1,j}^\top B_i^\top \Phi_{N-1,j}^\top \right)^{-1} + \mathcal{H}(t_N) \right]^{-1}. \quad (5.13) \end{aligned}$$

Because $\Sigma^\oplus(t_N)$ appears in the value function V_{N-1}^* , it is clear that the optimization problem with noisy, partial-state measurements is not quadratic in the controls and first-order methods are no longer sufficient. However, Eqs. (5.12) and (5.13) do give sufficient insight to develop upper and lower bounds for V_{N-1}^* (and all V_k^* , by extension).

Begin by noting that

$$\left[\left(\Phi_{N-1} \Sigma(t_{N-1})^\oplus \Phi_{N-1}^\top + \Gamma_{N-1} \right)^{-1} + \mathcal{H}(t_N) \right]^{-1} \leq \Sigma^\oplus(t_N). \quad (5.14)$$

Moreover, assume that

$$\Phi_N^\top W_{N+1} \Phi_N - W_N \quad (5.15)$$

is positive semidefinite (which must be verified numerically during implementation). Then, a lower bound on the value function at t_{N-1} can be defined such that

$$V_{N-1}^{*'} \leq V_{N-1}^*. \quad (5.16)$$

Using the substitution in inequality (5.14), the lower bound in question is

$$\begin{aligned} V_{N-1}^{*'} = \min_{\Delta \mathbf{u}_{N-1}} & \left\{ \bar{\mathbf{x}}(t_N)^\top W_N \bar{\mathbf{x}}(t_N) - 2\bar{\mathbf{x}}(t_N)^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \varphi_N + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right. \\ & \left. + \text{tr} (W_N \Sigma^\ominus(t_N)) + \text{tr} \left[\left(\Phi_N^\top W_{N+1} \Phi_N - W_N \right) \left(\left(\Phi_{N-1} \Sigma(t_{N-1})^\oplus \Phi_{N-1}^\top + \Gamma_{N-1} \right)^{-1} + \mathcal{H}(t_N) \right)^{-1} \right] \right\} \end{aligned} \quad (5.17)$$

which is quadratic in the control. Thus, the optimal control and optimal value of the lower bound can be computed analytically using first-order methods as before. Certainly, the problem will continue to become non-quadratic in the controls as the dynamic programming strategy proceeds backwards in time to each new segment. Accordingly, a similar substitution to inequality (5.14) must be made for each time segment. Because the equations are recursive and exhibit a repeating pattern, the details are omitted here.

The problem of generating a close upper bound is more difficult. By inspection of Eq. (5.4),

$$\Sigma^\oplus(t_{N-1}) = [\Sigma^\ominus(t_{N-1})^{-1} + \mathcal{H}(t_{N-1})]^{-1}, \quad (5.18)$$

$$\leq \mathcal{H}(t_{N-1})^{-1}. \quad (5.19)$$

From Eq. (5.13),

$$\Sigma^\oplus(t_N) \leq \left[\left(\Phi_{N-1} \mathcal{H}(t_{N-1})^{-1} \Phi_{N-1}^\top + \Gamma_{N-1} + \sum_{j=1}^{p_{N-1}} \sum_{i=1}^{\ell} \Phi_{N-1,j} B_i \Delta \mathbf{u}_{N-1,j} \Delta \mathbf{u}_{N-1,j}^\top B_i^\top \Phi_{N-1,j}^\top \right)^{-1} + \mathcal{H}(t_N) \right]^{-1}. \quad (5.20)$$

Using Lemma 1 below, it can be concluded that

$$\Sigma^\oplus(t_N) \leq \left[\left(\Phi_{N-1} \mathcal{H}(t_{N-1})^{-1} \Phi_{N-1}^\top + \Gamma_{N-1} \right)^{-1} + \mathcal{H}(t_N) \right]^{-1} + \Psi^{-1} \left(\sum_{j=1}^{p_{N-1}} \sum_{i=1}^{\ell} \Phi_{N-1,j} B_i \Delta \mathbf{u}_{N-1,j} \Delta \mathbf{u}_{N-1,j}^\top B_i^\top \Phi_{N-1,j}^\top \right) \Psi^{-\top}, \quad (5.21)$$

where Ψ is defined for compactness:

$$\Psi = I + \left(\Phi_{N-1} \mathcal{H}(t_{N-1})^{-1} \Phi_{N-1}^\top + \Gamma_{N-1} \right)^{-1} \mathcal{H}(t_N). \quad (5.22)$$

Assuming again that expression (5.15) is positive semidefinite, an upper bound can be defined such that

$$V_{N-1}^* \leq V_{N-1}^{\prime\prime*}, \quad (5.23)$$

$$\begin{aligned} V_{N-1}^{\prime\prime*} = \min_{\Delta \mathbf{u}_{N-1}} & \left\{ \bar{\mathbf{x}}(t_N)^\top W_N \bar{\mathbf{x}}(t_N) - 2\bar{\mathbf{x}}(t_N)^\top C_N \mathbf{z}_f + \mathbf{z}_f^\top F_N \mathbf{z}_f + \varphi_N + \sum_{j=1}^{p_{N-1}} \Delta \mathbf{u}_{N-1,j}^\top R \Delta \mathbf{u}_{N-1,j} \right. \\ & + \text{tr} (W_N \Sigma^\ominus(t_N)) + \text{tr} \left[\left(\Phi_N^\top W_{N+1} \Phi_N - W_N \right) \left(\left(\Phi_{N-1} \mathcal{H}(t_{N-1})^{-1} \Phi_{N-1}^\top + \Gamma_{N-1} \right)^{-1} + \mathcal{H}(t_N) \right)^{-1} \right] \\ & \left. + \text{tr} \left[\left(\Phi_N^\top W_{N+1} \Phi_N - W_N \right) \Psi^{-1} \left(\sum_{j=1}^{p_{N-1}} \sum_{i=1}^{\ell} \Phi_{N-1,j} B_i \Delta \mathbf{u}_{N-1,j} \Delta \mathbf{u}_{N-1,j}^\top B_i^\top \Phi_{N-1,j}^\top \right) \Psi^{-\top} \right] \right\}. \end{aligned} \quad (5.24)$$

The upper bound is quadratic in the control, enabling the use of first-order methods. The upper bound also has a physical interpretation: it corresponds to an approximation of the value function when the control noise is assumed to be small in comparison to the navigation uncertainty (see Remark 2). As with the lower bound, the solution to the upper bound will also become non-quadratic at every step, and a substitution similar to inequality (5.21) must be repeated at every time segment.

Lemma 1. For positive definite matrices X and Y and arbitrary matrix Z ,

$$\left[\left(Y + ZZ^\top \right)^{-1} + X \right]^{-1} \leq (Y^{-1} + X)^{-1} + (I + YX)^{-1} ZZ^\top (I + XY)^{-1}. \quad (5.25)$$

Proof of Lemma 1. Using the Woodbury matrix identity,

$$\left(Y + ZZ^\top \right)^{-1} = Y^{-1} - Y^{-1}Z \left(I + Z^\top Y^{-1}Z \right)^{-1} Z^\top Y^{-1}, \quad (5.26)$$

It follows that

$$\left[\left(Y + ZZ^\top \right)^{-1} + X \right]^{-1} = \left[Y^{-1} + X - Y^{-1}Z \left(I + Z^\top Y^{-1}Z \right)^{-1} Z^\top Y^{-1} \right]^{-1}. \quad (5.27)$$

For compactness, allow the following temporary substitutions:

$$A = Y^{-1} + X, \quad (5.28)$$

$$B = Y^{-1}Z, \quad (5.29)$$

$$C = \left(I + Z^\top Y^{-1}Z \right)^{-1}, \quad (5.30)$$

such that

$$\left[\left(Y + ZZ^\top \right)^{-1} + X \right]^{-1} = \left[A - BCB^\top \right]^{-1}. \quad (5.31)$$

Applying the Woodbury identify for a second time results in

$$\left[A - BCB^\top \right]^{-1} = A^{-1} + A^{-1}B \left(C^{-1} - B^\top A^{-1}B \right)^{-1} B^\top A^{-1}. \quad (5.32)$$

From Eqs. (5.28) through (5.30),

$$C^{-1} - B^\top A^{-1}B = I + Z^\top Y^{-1} \left[Y - (Y^{-1} + X)^{-1} \right] Y^{-1}Z. \quad (5.33)$$

Because X is positive semidefinite,

$$Y^{-1} + X \geq Y^{-1}, \quad (5.34)$$

$$(Y^{-1} + X)^{-1} \leq Y, \quad (5.35)$$

$$Y - (Y^{-1} + X)^{-1} \geq [0]. \quad (5.36)$$

By extension,

$$C^{-1} - B^\top A^{-1} B \geq I, \quad (5.37)$$

$$\left(C^{-1} - B^\top A^{-1} B \right)^{-1} \leq I, \quad (5.38)$$

$$A^{-1} B \left(C^{-1} - B^\top A^{-1} B \right)^{-1} B^\top A^{-1} \leq A^{-1} + A^{-1} B B^\top A^{-1}. \quad (5.39)$$

Finally, making the reverse substitution produces the result in Lemma 1. This completes the proof. \blacksquare

Remark 2. Assume $\|Y\| \gg \|ZZ^\top\|$, e.g., the control noise is small in Eq. (5.13). Then,

$$\left[\left(Y + ZZ^\top \right)^{-1} + X \right]^{-1} \approx \left[Y^{-1} - Y^{-1} ZZ^\top Y^{-1} + X \right]^{-1} \quad (5.40)$$

$$\approx \left[(Y^{-1} + X) - Y^{-1} ZZ^\top Y^{-1} \right]^{-1} \quad (5.41)$$

Next, assuming $\|Y^{-1} + X\| \gg \|Y^{-1} ZZ^\top Y^{-1}\|$,

$$\left[(Y^{-1} + X) - Y^{-1} ZZ^\top Y^{-1} \right]^{-1} \approx (Y^{-1} + X)^{-1} + (Y^{-1} + X)^{-1} Y^{-1} ZZ^\top Y^{-1} (Y^{-1} + X)^{-1} \quad (5.42)$$

$$\approx (Y^{-1} + X)^{-1} + (I + YX)^{-1} ZZ^\top (I + XY)^{-1} \quad (5.43)$$

With the appropriate substitutions, the approximation in Eq. (5.43) is identical to the right-hand side of the inequality in (5.21).

5.3 Bounded Cost Function Result

Theorem 4. Consider the stochastic system in Eq. (5.2) and the optimal control problem in Eq. (4.2). The optimal cost can be bounded such that

$$V_0'^* \leq J \leq V_0''^* \quad (5.44)$$

where

$$V_0'^* = \hat{\mathbf{x}}_{t_0}^\top W_0' \hat{\mathbf{x}}_{t_0} - 2\hat{\mathbf{x}}_{t_0}^\top C_0' \mathbf{z}_f + \mathbf{z}_f^\top F_0' \mathbf{z}_f + \varphi_0', \quad (5.45)$$

$$V_0''^* = \hat{\mathbf{x}}_{t_0}^\top W_0'' \hat{\mathbf{x}}_{t_0} - 2\hat{\mathbf{x}}_{t_0}^\top C_0'' \mathbf{z}_f + \mathbf{z}_f^\top F_0'' \mathbf{z}_f + \varphi_0''. \quad (5.46)$$

The optimal controls associated with the upper and lower bounds are denoted $\mathbf{u}_{k,j}^{l*}$ and $\mathbf{u}_{k,j}^{u*}$, respectively and are given by

$$\mathbf{u}_{k,j}^{l*} = K_{k,j}' \hat{\mathbf{x}}_{t_k} + L_{k,j}' \mathbf{z}_f, \quad (5.47)$$

$$\mathbf{u}_{k,j}^{u*} = K_{k,j}'' \hat{\mathbf{x}}_{t_k} + L_{k,j}'' \mathbf{z}_f, \quad (5.48)$$

for all $k = 0, \dots, N$, and $j = 0, \dots, p_k$, where $\hat{\mathbf{x}}_{t_k}$ is the Kalman filter state estimate at time t_k . The quantities $K_{k,j}'$, $L_{k,j}'$, W_0' , C_0' , F_0' , φ_0' , $K_{k,j}''$, $L_{k,j}''$, W_0'' , C_0'' , F_0'' , and φ_0'' in the above expressions are computed in Algorithms 5 and 6.

The proof of Theorem 4 follows the same procedure as the proof of Theorem 3. To preserve the quadratic structure of lower and upper bounds, substitutions similar to inequalities (5.14) and (5.21) are repeated for each segment of the dynamic programming solution.

5.4 Bounded Cost Function Simulation

The following examples use the same problem setup as in Section 4.4. The problem is nondimensionalized by the initial orbit radius in the length scale and the mean motion in the time scale. Each example in this section considers an initial nominal state of $\mathbf{X}'(t_0) = [19.8, 912, 0, 0, 0, 0.0772]^\top$ (m,m/s) and an initial state error of $\mathbf{x}_{t_0} = [2, -10, -2, 6 \times 10^{-4}, -10^{-3}, 3 \times 10^{-4}]^\top$ (m,mm/s). The goal is to minimize the final mean squared state deviation after one asteroid Sun-terminator orbit. Full-state noisy measurements are assumed such that

$$H = I, \quad D = \sigma_m^2 I \quad (5.49)$$

in Eq. (5.2). State estimates are generated from a Kalman filter as in Eqs. (5.4) through (5.6). Because the noisy measurement problem with control-dependent noise cannot be solved analytically, Theorem 4 will be used to generate upper and lower bounds on the optimal cost, J^* for J given in Eq. (4.2).

The following simulations will show how the bounds V_0^{l*} and V_0^{u*} change as a number of parameters are varied from a set of baseline parameters, which are given in Table 5.1. All measurements and maneuvers

Algorithm 5: Lower Bound

Data: Matrices $A(t), B_0(t), \dots, B_\ell(t), R, G(t), \mathcal{H}(t), \Sigma^\oplus(t_0)$, measurement times t_1, \dots, t_N , impulse times $t_{k,1}, \dots, t_{k,p_k}$, for all $k = 0, \dots, N$, and final time $t_f := t_{N+1}$.

Result: Scalar φ'_0 , matrices $W'_0, C'_0, F'_0, K'_{k,j}$ and $L'_{k,j}$ for all $k = 0, \dots, N$ and $j = 1, \dots, p_k$. initialize;

$$W'_{N+1} := I; F'_{N+1} := I; C'_{N+1} := I; \varphi'_{N+1} := 0.$$

for $k := N$ **to** 0 **do**

if $k < N$ **and** $(\Phi_{k+1}^\top W'_{k+2} \Phi_{k+1} - W'_{k+1})$ **is negative definite** **then**

Algorithm is invalid; terminate Algorithm.

end

for $j := 1$ **to** p_k **do**

$$Q'_{k,j} := R + \sum_{i=1}^{\ell} B_i(t_{k,j})^\top \Phi_{k,j}^\top W'_{k+1} \Phi_{k,j} B_i(t_{k,j});$$

end

$$M'_k := \sum_{j=1}^{p_k} \Phi_{k,j} B_0(t_{k,j}) Q'^{-1}_{k,j} B_0(t_{k,j})^\top \Phi_{k,j}^\top;$$

$$W'_k := \Phi_k^\top [W'_{k+1} + M'_k]^{-1} \Phi_k;$$

$$C'_k := \Phi_k^\top [I + W'_{k+1} M'_k]^{-1} C'_{k+1};$$

$$F'_k := F'_{k+1} - C'^\top_{k+1} M'_k [I + W'_{k+1} M'_k]^{-1} C'_{k+1};$$

$$\Gamma_k := \int_{t_k}^{t_{k+1}} \Phi_\tau G(\tau) G(\tau)^\top \Phi_\tau^\top d\tau;$$

$$\varphi'_k := \varphi'_{k+1} + \text{tr}(W'_{k+1} \Gamma_k);$$

and

$$K'_{k,j} := -Q'^{-1}_{k,j} B_0(t_{k,j})^\top \Phi_{k,j}^\top [W'_{k+1} + M'_k]^{-1} \Phi_k;$$

$$L'_{k,j} := Q'^{-1}_{k,j} B_0(t_{k,j})^\top \Phi_{k,j}^\top [I + W'_{k+1} M'_k]^{-1} C'_{k+1};$$

end

$$\varphi'_0 = \varphi'_0 + \text{tr}(\Phi_0^\top W'_1 \Phi_0 \Sigma^\oplus(t_0));$$

$$\beta_0 := \Sigma^\oplus(t_0);$$

for $k := 1$ **to** N **do**

$$\beta_k := \left[\left(\Phi_{k-1} \beta_{k-1} \Phi_{k-1}^\top + \Gamma_{k-1} \right)^{-1} + \mathcal{H}(t_k) \right]^{-1};$$

$$\varphi'_0 = \varphi'_0 + \text{tr}[(\Phi_k^\top W'_{k+1} \Phi_k - W'_k) \beta_k];$$

end

Algorithm 6: Upper Bound

Data: Matrices $A(t), B_0(t), \dots, B_\ell(t), R, G(t), \mathcal{H}(t), \Sigma^\oplus(t_0)$, measurement times t_1, \dots, t_N , impulse times $t_{k,1}, \dots, t_{k,p_k}$, for all $k = 0, \dots, N$, and final time $t_f =: t_{N+1}$.

Result: Scalar φ_0'' , matrices $W_0'', C_0'', F_0'', K_{k,j}''$ and $L_{k,j}''$ for all $k = 0, \dots, N$ and $j = 1, \dots, p_k$. initialize;

$$W_{N+1}'' := I; F_{N+1}'' := I; C_{N+1}'' := I; \varphi_{N+1}'' := 0.$$

for $k := N$ **to** 0 **do**

$$\Gamma_k := \int_{t_k}^{t_{k+1}} \Phi_\tau G(\tau) G(\tau)^\top \Phi_\tau^\top d\tau;$$

if $k = N$ **then**

$$\begin{aligned} \Omega_k &:= W_{k+1}''; \\ \varphi_k'' &:= \varphi_{k+1}'' + \text{tr}(W_{k+1}'' \Gamma_k) \end{aligned}$$

else

If $(\Phi_{k+1}^\top W_{k+2}'' \Phi_{k+1} - W_{k+1}'')$ **is negative definite, Algorithm is invalid; terminate Algorithm.**

$$\begin{aligned} \Lambda_k &:= \left[I + \mathcal{H}(t_{k+1}) \left(\Phi_k \mathcal{H}(t_k)^{-1} \Phi_k^\top + \Gamma_k \right) \right]^{-1}; \\ \Omega_k &:= W_{k+1}'' + \Lambda_k \left(\Phi_{k+1}^\top W_{k+2}'' \Phi_{k+1} - W_{k+1}'' \right) \Lambda_k^\top; \\ \varphi_k'' &:= \varphi_{k+1}'' + \text{tr}(W_{k+1}'' \Gamma_k) + \text{tr} \left[\left(\Phi_{k+1}^\top W_{k+2}'' \Phi_{k+1} - W_{k+1}'' \right) \left(\Phi_k \mathcal{H}(t_k)^{-1} \Phi_k^\top + \Gamma_k \right) \Lambda_k \right]; \end{aligned}$$

end

for $j := 1$ **to** p_k **do**

$$Q_{k,j}'' := R + \sum_{i=1}^{\ell} B_i(t_{k,j})^\top \Phi_{k,j}^\top \Omega_k \Phi_{k,j} B_i(t_{k,j});$$

end

$$M_k'' := \sum_{j=1}^{p_k} \Phi_{k,j} B_0(t_{k,j}) Q_{k,j}''^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top;$$

$$W_k'' := \Phi_k^\top [W_{k+1}''^{-1} + M_k'']^{-1} \Phi_k;$$

$$C_k'' := \Phi_k^\top [I + W_{k+1}'' M_k'']^{-1} C_{k+1}'';$$

$$F_k'' := F_{k+1}'' - C_{k+1}''^\top M_k'' [I + W_{k+1}'' M_k'']^{-1} C_{k+1}'';$$

and

$$K_{k,j}'' := -Q_{k,j}''^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top [W_{k+1}''^{-1} + M_k'']^{-1} \Phi_k;$$

$$L_{k,j}'' := Q_{k,j}''^{-1} B_0(t_{k,j})^\top \Phi_{k,j}^\top [I + W_{k+1}'' M_k'']^{-1} C_{k+1}'';$$

end

$$\varphi_0'' = \varphi_0'' + \text{tr} \left(\Phi_0^\top W_1'' \Phi_0 \Sigma^\oplus(t_0) \right)$$

are evenly spaced in time. Figures 5.1 through 5.5 show how the bounds vary with the increase of along-axis control noise, measurement noise, initial state uncertainty, number of measurements, and number of controls, respectively. The problem dynamics and baseline parameters dictate how the bounds evolve, and the values in these plots are specific to the example given here. However, the trends provide insight for other scenarios.

Figure 5.1 addresses the control noise case. Note that the lower bound was derived by assuming that the Kalman filter has no knowledge of the control noise and is overly confident in the state estimate. Therefore, the lower bound will be tight when the control noise is small and becomes less accurate as the control noise is increased. Additionally, from Remark 2, the upper bound resembles an approximation of the cost function when the control noise is small. Thus, the upper bound is also most accurate for small control noise. As expected, Figure 5.1 shows that the bounds diverge as the control noise is increased.

Despite this divergence, the bound are relatively close for realistic levels of control noise, e.g., for a control noise factor of 0.05 (5% magnitude uncertainty), the bounds differ by roughly 6%. Although the true value of the optimal cost is unknown, the optimal cost has been approximated with an accuracy of 6%. The bounds also provide a metric for evaluating candidate controls — if a control produces a cost that lies within the bounds, it is close to optimal. Moreover, the controls associated with the bounds provide an initial guess for the optimal control.

Figures 5.2, 5.3, and 5.4 also show that the bounds diverge as the measurement noise, initial state uncertainty, and number of measurements are increased. The measurement noise standard deviation in

Table 5.1: Baseline Parameters with Noisy Measurements

| Symbol | Description | Value | Units |
|---------------------------|---|--|--------------------------------------|
| N | Number of measurements | 3 | - |
| p_k | Number of controls per segment | 10 | - |
| σ_{add} | Std. dev. of additive noise | $10^{-3}\mu_A/r_0^2$ | m/s ² |
| σ_{SRP} | Std. dev of a_{SRP} | $10^{-2}a_{SRP}$ | mm/s ² |
| ε_{\parallel} | Along-axis $\Delta\mathbf{u}$ uncertainty factor | 5% | - |
| ε_{\perp} | Off-axis $\Delta\mathbf{u}$ uncertainty factor | 0.5% | - |
| σ_m | Std. dev. of nondimensionalized measurement noise | 10^{-3} | - |
| $\Sigma^{\oplus}(t_0)$ | Initial state error covariance | $s \times \text{diag}(100, 100, 100, 1, 1, 1)$ | m ² , (mm/s) ² |
| s | Initial covariance scaling parameter | 1 | - |

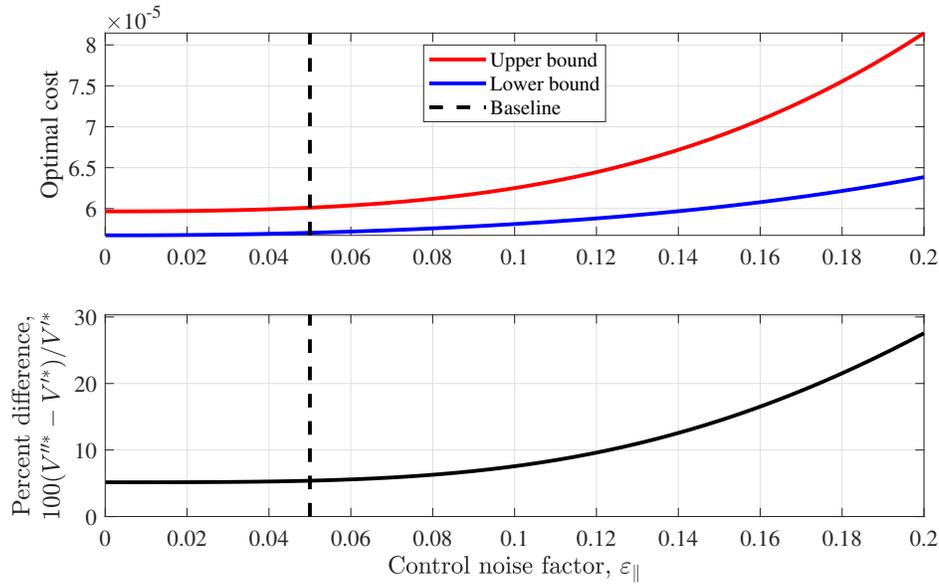


Figure 5.1: Divergence of upper and lower cost function bounds as the level of along-axis control noise is increased.

Figure 5.2 is nondimensionalized. In this case, the baseline control noise corresponds to a standard deviation of 1 m in position and 0.07 mm/s in velocity. The bounds quickly diverge as the measurement noise is increased. Thus, the measurement noise is likely the deciding factor that determines whether the bounded cost function method is useful.

Finally, Figure 5.5 shows how the bounds vary as the number of controls per segment is increased. Both upper and lower bounds converge to a positive limiting value as the number of controls increases (as was also seen in Section 4.4 for the perfect measurement case). The percent difference between the bounds also appears to converge to a positive value.

5.5 Discussion and Conclusions

The problem of optimal control with noisy measurements and control-dependent noise is difficult to solve analytically, and a bounded cost method has been introduced in this chapter to analytically bound the optimal cost. The upper and lower bounds are chosen to be quadratic in the controls so that they can be optimized using first-order methods. The bounds provide a metric for evaluating candidate controls — if a set of controls produce a solution that lies within the bounds, then the control is close to optimal. The

solution to each bound also provides a set of controls that can serve as an initial guess for the true optimal. A conservative approach would be to apply the control associated with the upper bound, because the true cost is guaranteed to be less than or equal to the upper bound for a particular set of controls. The level of measurement noise is likely the driving factor that determines whether the bounds are tight enough to provide useful information.

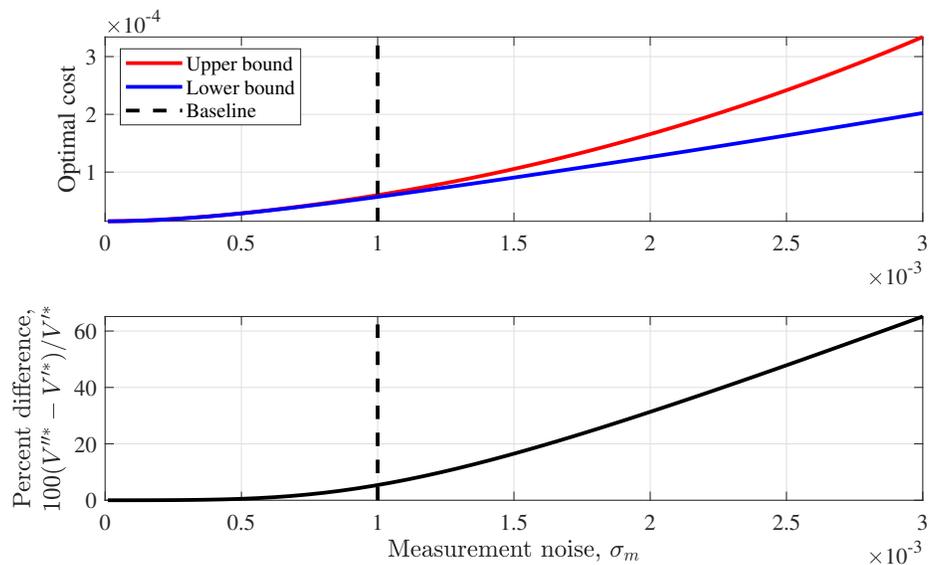


Figure 5.2: Divergence of upper and lower cost function bounds as the level of measurement noise is increased.

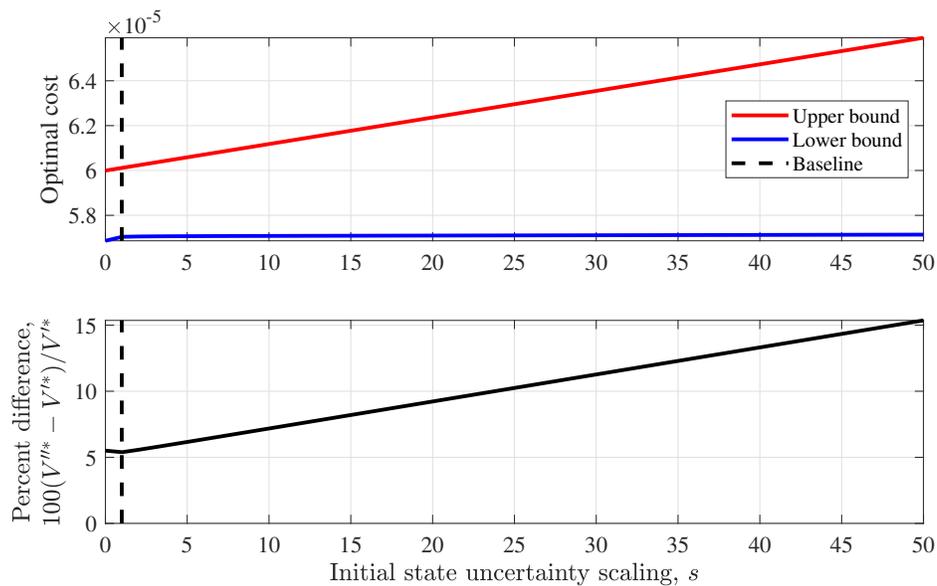


Figure 5.3: Divergence of upper and lower cost function bounds as the initial state uncertainty is increased.

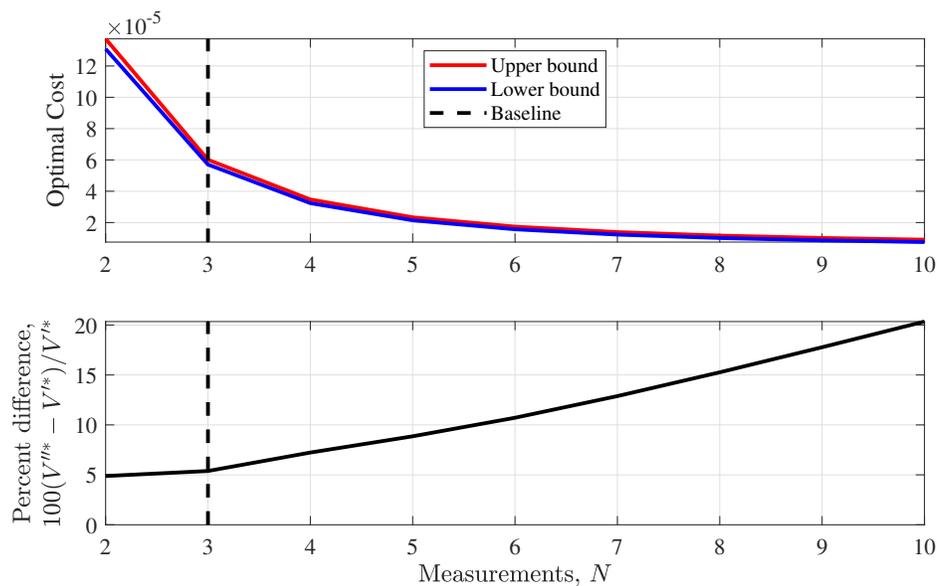


Figure 5.4: Divergence of upper and lower cost function bounds as the number of measurements are increased.

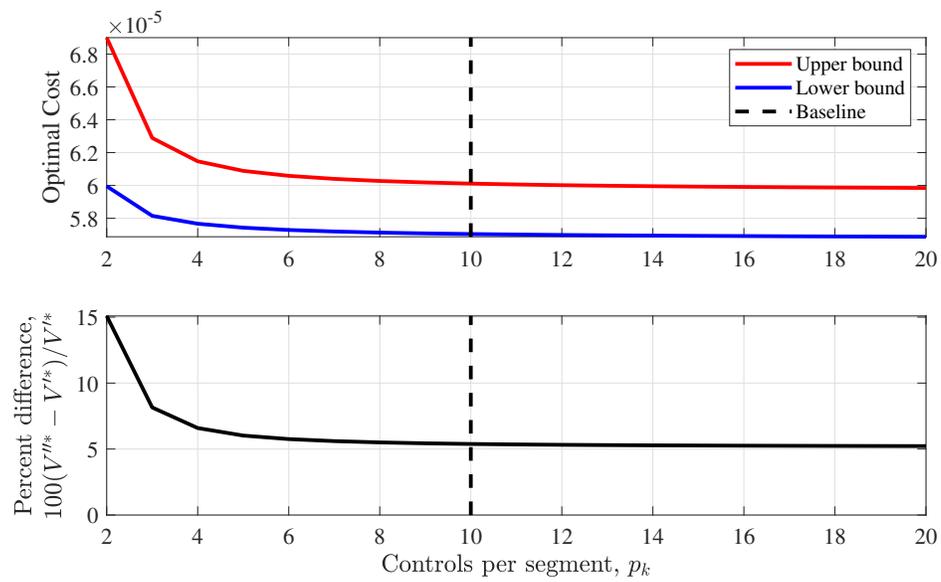


Figure 5.5: Variation of upper and lower cost function bounds as the number of maneuvers per segment is increased.

Chapter 6

Tensor Eigenpair Measure of Nonlinearity

Chapter 6 Nomenclature

| | | |
|----------------------------------|---|--|
| A | = | linear dynamics matrix |
| $\mathcal{A}^{(m)}$ | = | m th-order local dynamics tensor |
| \mathbf{f} | = | system dynamics |
| L_k | = | Lagrangian |
| \mathbf{l} | = | linear model |
| m | = | tensor order |
| n | = | dimension of state/model |
| $n_{\mathbf{q}}, n_{\mathbf{w}}$ | = | dimensions of \mathbf{q}, \mathbf{w} |
| \mathbf{q}, \mathbf{w} | = | two parameterizations of a system state |
| p | = | TSE order |
| $\mathbf{S}_{\mathcal{T},m}$ | = | S-TEMoN |
| $\mathbf{T}_{\mathcal{T},m}$ | = | TEMoN |
| t | = | time |
| $\mathcal{T}^{(m)}$ | = | generic m th-order tensor |
| \mathbf{v} | = | real-valued deviation vector |
| $\hat{\mathbf{v}}_k^*$ | = | direction of maximum k th-order nonlinearity |
| \mathbf{x} | = | dynamical system state |

| | | |
|--------------------------|---|---|
| z | = | alternative parameterization of v |
| $\aleph_{\mathcal{T},m}$ | = | m th-order nonlinearity measure |
| α | = | magnitude of v |
| γ | = | Lagrange multiplier |
| η | = | nonlinear model |
| λ | = | tensor Z-eigenvalue |
| τ_k | = | k th-order term in TEMoN |
| Φ | = | state transition matrix |
| $\bar{\Phi}^{(m)}$ | = | m th-order state transition tensor |
| ϕ | = | solution flow |
| $\Omega^{(m)}$ | = | m th-order transformation-propagation-transformation tensor |

6.1 Introduction

A correlation between uncertainty and nonlinearity was observed in Chapter 2. Nonlinearity measures may be used to further investigate that relationship. This chapter proposes a novel method to measure the nonlinearity of static and dynamic systems. Nonlinearity is the crux of analytical dynamics, and local linearization (based on the TSE) is a common practice used to facilitate analytical solutions. Some nonlinear models are “less nonlinear” than others and are more amenable to these linear approximations. In turn, linear control, navigation, and uncertainty propagation methods (e.g. an LQR or a Kalman filter) may be applicable to weakly nonlinear models. For a strongly nonlinear system, however, the linear approximation may only be valid in a very limited region, rendering linear methods insufficient. As a result, numerous measures of nonlinearity (MoN) have been proposed to quantify relative degrees of nonlinearity, and their applications vary greatly.

For example, MoN can be used to compare the nonlinearity of different state parameterizations, evaluate the relative nonlinearity of different trajectories, and study the evolution of nonlinearity along a single trajectory. MoN have been used to explore the influence of coordinate frame selection on the accuracy of

uncertainty propagation [50] and to govern the splitting of Gaussian mixture models [51–53]. In control theory, MoN have been utilized to trigger sensitivity updates in model predictive control [54] and identify regions of strong nonlinearity that may necessitate frequent gain scheduling [55]. In estimation, MoN have been applied to predict performance degradation due to nonlinearities in different measurement models [56]. They have also been used extensively to justify the selection of linear versus nonlinear filters [57–61].

In the field of astrodynamics, nonlinearity is often accepted as an unfortunate and unavoidable aspect of space mission design. However, an understanding of nonlinearity and its effects during mission planning can enhance spacecraft guidance, navigation, and control performance. A detailed analysis of nonlinearity can enable mission designers to avoid strongly nonlinear regions, configure maneuver and navigation schedules to better leverage linear techniques, and determine when nonlinear methods are truly necessary. In this way, MoN can be a useful tool in the mission design process.

This chapter proposes a MoN that is based on tensor eigenpairs — the tensor eigenpair measure of nonlinearity (TEMoN). Unlike many heuristic MoN in the literature, the TEMoN is semianalytical and its computation does not rely on empirical sampling or numerical optimization. This method will utilize tensor eigenpairs to locate directions of strong nonlinearity. The relevant tensors will be derived from the higher-order terms in a TSE, such as the local dynamics tensors (LDTs) and state transition tensors (STTs). LDTs and STTs constitute the higher-order terms in a TSE of a system’s dynamics and solution flow, respectively [62]. LDTs and STTs are natural mechanisms for studying nonlinearity; for a linear model, these higher-order terms are zero. Therefore, the model nonlinearity is dictated by higher-order tensors. The TEMoN will also be applied to parameter transformations.

This chapter begins with an extensive background section. Existing MoN are reviewed in Section 6.2.1. Tensor operations and eigenpairs are described in Section 6.2.2. Relevant TSE, including LDTs and STTs, are introduced in Section 6.2.3. The TEMoN is detailed in Sections 6.3 and 6.4. Finally, the TEMoN is applied to LDTs, STTs, and nonlinear parameter transformations in Section 6.5 through 6.7.

6.2 Background

6.2.1 Measures of Nonlinearity

To streamline this discussion, MoN will be grouped into two categories: deviation-based and curvature-based. Deviation MoN are based on measuring deviations between nonlinear functions and linear approximations and they include the earliest examples of MoN in the literature. Alternatively, curvature MoN are based on differential geometry. Nonlinearity measures also exist for time series data as a part of system identification [63], but time series methods are not considered here. Within these two categories, MoN may be local or global. Local MoN measure nonlinearity about a single point (often the linearization point in a TSE), whereas global MoN measure nonlinearity over the entire solution locus. MoN can be further distinguished by the models that they address: static and/or dynamic, open and/or closed-loop, deterministic and/or stochastic, and derivative and/or derivative-free.

6.2.1.1 Deviation-Based MoN

The first local MoN was proposed by Beale in 1960 to determine confidence regions for least-squares estimates [64, 65]. Beale's MoN measures the deviations between a nonlinear function and nearby approximations of the nonlinear function that are based on the TSE. In practice, the MoN may be computed via empirical sampling. Beale also proposes an intrinsic MoN that corresponds to the state parameterization which minimizes the MoN, as well as a theoretical MoN that corresponds to an infinite number of sampling points.

Desoer proposed a global MoN to study the linearizing effect of feedback control. The Desoer MoN measures the deviation between a nonlinear function ($\boldsymbol{\eta}$) and its best linear approximation (\boldsymbol{l}), over a subset of input values (\boldsymbol{u}), as follows [66]:

$$\inf_{\boldsymbol{l}} \sup_{\boldsymbol{u}} |\boldsymbol{\eta}(\boldsymbol{u}) - \boldsymbol{l}(\boldsymbol{u})|. \quad (6.1)$$

The minimax optimization in Eq. (6.1) varies in difficulty based on the nonlinear system in question, and the result is influenced by the allowable subset of inputs, $\boldsymbol{u} \in \mathcal{U}$.

Helbig et al proposed a modified version of Desoer's MoN that also accounts for the initial conditions

of a dynamical system:

$$\inf_l \sup_{\mathbf{u}, \mathbf{x}_N(t_0)} \inf_{\mathbf{x}_L(t_0)} \frac{|\boldsymbol{\eta}(\mathbf{u}, \mathbf{x}_N(t_0), t) - \mathbf{l}(\mathbf{u}, \mathbf{x}_L(t_0), t)|}{|\boldsymbol{\eta}(\mathbf{u}, \mathbf{x}_N(t_0), t)|}, \quad (6.2)$$

where $t \geq t_0$ is some future time and $\mathbf{x}_N(t_0)$ and $\mathbf{x}_L(t_0)$ are the initial conditions of nonlinear and linear dynamical systems, respectively [67]. Like the Desoer MoN, the Helbig MoN is disadvantaged by its difficult computation; numerical optimization methods are typically required.

Liu and Li proposed a global MoN for nonlinear estimation that accommodates noise:

$$\text{rmse} := \sqrt{\min_{\mathbf{l}, \psi(\boldsymbol{\omega}, \boldsymbol{\nu})} \mathbb{E} [\|\mathbf{l}(\mathbf{x}, \boldsymbol{\omega}) - \boldsymbol{\eta}(\mathbf{x}, \boldsymbol{\nu})\|^2]}, \quad (6.3)$$

where $\boldsymbol{\eta}$ is a nonlinear function (measurement) of the state \mathbf{x} , $\boldsymbol{\nu}$ is the measurement noise with a known cumulative distribution function (CDF), \mathbf{l} is an affine function of \mathbf{x} , $\boldsymbol{\omega}$ is zero-mean additive noise, and $\psi(\boldsymbol{\omega}, \boldsymbol{\nu})$ is the joint CDF of $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$ [68]. In other words, this measure is the minimum root mean square error (rmse) when approximating the nonlinear function $\boldsymbol{\eta}$ with a linear function L . In general, numerical minimization of the rmse in Eq. (6.3) is difficult; however, the problem can be converted to a simpler parameter optimization with some approximation.

Junkins and Singla proposed nonlinearity indices (NLI) that measure the distance between two linear systems: (1) a nonlinear system linearized about its mean state and (2) the same nonlinear system linearized about worst-case state deviations [50]. The NLI may be applied to static (algebraic) or dynamic systems. In the static case, the NLI is computed as the supremum difference in the linearization matrix, $A(\mathbf{x})$, due to worst-case deviations in \mathbf{x} :

$$\text{NLI}_s := \sup_{i=1, \dots, N} \frac{\|A(\mathbf{x}_i) - A(\bar{\mathbf{x}})\|}{\|A(\bar{\mathbf{x}})\|}, \quad A(\mathbf{x}_i) = \left. \frac{\partial \boldsymbol{\eta}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}_i}, \quad (6.4)$$

where $\boldsymbol{\eta}$ is a nonlinear static system, $\bar{\mathbf{x}}$ is the mean state, \mathbf{x}_i are worst-case deviated states, N is the total number of sampled deviated states, and A is the Jacobian matrix of $\boldsymbol{\eta}$ with respect to \mathbf{x} . Note that A may also be the instantaneous linearized dynamics matrix of a dynamical system. For example, given the dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t), \quad (6.5)$$

the linearized dynamics matrix at time t may be computed

$$A(\mathbf{x}_i(t), t) = \left. \frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \right|_{\mathbf{x}_i(t)}, \quad (6.6)$$

and Eq. (6.4) is applied directly. For dynamical systems over a period of time, the dynamic NLI measures the supremum difference in the state transition matrix (STM) due to worst-case deviations in the initial state:

$$\text{NLI}_d := \sup_{i=1, \dots, N} \frac{\|\Phi_i(t, t_0) - \Phi^*(t, t_0)\|}{\|\Phi^*(t, t_0)\|}, \quad (6.7)$$

where Φ^* is the STM computed from the mean initial state and Φ_i are computed from deviated initial states. To compute Eqs. (6.4) and (6.7), deviations from the mean are uniformly sampled from the n -dimensional hypersurface of worst-case state deviations. To facilitate ease of uniform sampling, the dynamics are scaled such that the worst-case surface forms a unit hypersphere. State deviations may also be sampled randomly from a probability distribution, but uniform sampling is recommended in [50] to reduce the necessary number of sampling points. The worst-case hypersurface and the number of sampling points will influence the result, and these selections must be justified. Repeating the process for different worst-case surfaces and/or sampling schemes can be time-consuming, particularly in the case of Eq. (6.7) where STMs must be numerically integrated.

Park and Scheeres defined a local nonlinearity index (LNLI) to incorporate higher-order terms of the TSE [62]. Like the NLI, the LNLI is also based on empirical sampling. Other authors have proposed the gap metric as a measure of closed-loop nonlinearity [69–71]. The gap metric is a topological measure that quantifies the distance between two linear systems (or two linearizations of a nonlinear system). The gap metric has many advantageous properties; however, the gap between two linear systems is a debatable measure of nonlinearity. Hahn and Edgar propose a local, Gramian-based MoN in which a system's controllability and observability Gramians at a nominal point are compared to Gramians computed from empirical sampling [72]. The Gramian method suggests that nonlinearity is only relevant if it affects a system's controllability and observability. Lyapunov exponents may also be used as a MoN [50].

6.2.1.2 Curvature MoN

Bates and Watts proposed relative curvature measures of nonlinearity (CMoN), based on differential geometry [73]. CMoN measure the curvature of a nonlinear function with respect to its tangent plane approximation (linearization). Consider a point in parameter space, θ_0 , which maps to a point on the solution locus, $\eta(\theta_0)$. An arbitrary straight line through θ_0 in the direction v can be denoted

$$\theta(b) = \theta_0 + bv, \quad (6.8)$$

where b is the independent variable. This line in parameter space maps to a curve on the solution locus, denoted $\eta_v(b)$. The tangent to this curve at $b = 0$ is the first derivative of η_v , i.e. $\dot{\eta}_v = d\eta_v/db|_{b=0}$. The second derivative, $\ddot{\eta}_v$, can also be computed in a similar manner. Note that the first and second derivatives can be related to the Jacobian and Hessian terms in a TSE of η . A general CMoN can be defined as

$$\text{CMoN} := \max_v \frac{\|\ddot{\eta}_v\|}{\|\dot{\eta}_v\|^2}. \quad (6.9)$$

The vector $\ddot{\eta}_v$ can be decomposed into the components $\ddot{\eta}_v^N$ and $\ddot{\eta}_v^T$, which are normal and parallel to the tangent hyperplane, respectively. These components form the basis of intrinsic (CMoN_N) and parameter-dependent (CMoN_T) curvature measures.

$$\text{CMoN}_N := \max_v \frac{\|\ddot{\eta}_v^N\|}{\|\dot{\eta}_v\|^2} \quad (6.10)$$

$$\text{CMoN}_T := \max_v \frac{\|\ddot{\eta}_v^T\|}{\|\dot{\eta}_v\|^2} \quad (6.11)$$

The vector $\ddot{\eta}_v$ is decomposed via projection onto the tangent hyperplane. Note that if θ and η are of the same dimension, the projection matrix is an identity matrix, i.e., the vector already lies in the tangent hyperplane and there is no intrinsic curvature. The vector v that maximizes the CMoN can be determined through numerical optimization or sampling. The CMoN does not consider derivatives of higher order, which also contribute to nonlinearity.

6.2.1.3 Tensor Eigenpair Measure of Nonlinearity

The aforementioned MoN all require some level of empirical sampling or numerical optimization for their computation and many of the methods are heuristic. The semianalytical CMoN is equivalent to normalizing the Hessian term in a TSE (the local curvature) by the squared norm of the linear term (the tangent vector). The TEMoN proposed in this chapter will extend this concept to an arbitrary number of higher-order TSE terms. Consider a general nonlinear function $\eta(\mathbf{V})$. Let $\mathbf{v} = \mathbf{V} - \mathbf{V}^*$, where \mathbf{V}^* is the nominal value of the parameters \mathbf{V} . The TSE of η with respect to \mathbf{V}^* can be written as

$$\delta\eta = \sum_{k=1}^{\infty} \frac{1}{k!} \frac{\partial^k \eta}{\partial \mathbf{V}^k} \Big|_{\mathbf{V}^*} \mathbf{v}^k \quad (6.12)$$

using the notation described further in Section 6.2.2 for tensor-vector multiplication. Thus, $\delta\eta$ is the summation of one linear term and a series of multilinear (higher-order) terms. The TEMoN will quantify the influence of multilinear terms on the squared norm of $\delta\eta$:

$$\delta\eta^\top \delta\eta = \mathbf{v}^\top \frac{\partial \eta}{\partial \mathbf{V}} \Big|_{\mathbf{V}^*} \frac{\partial \eta}{\partial \mathbf{V}} \Big|_{\mathbf{V}^*} \mathbf{v} + \mathcal{N}(\mathbf{v}^3, \mathbf{v}^4, \dots, \mathbf{v}^\infty). \quad (6.13)$$

The term \mathcal{N} comprises the model nonlinearity, i.e., it is dependent on the multilinear terms in $\delta\eta$. If η is linear, then $\mathcal{N} = 0$. Therefore, the tensors in \mathcal{N} can be used to quantify nonlinearity with respect to the input \mathbf{v} . The TEMoN will use tensor eigenpairs to evaluate nonlinearity and identify directions of strong nonlinearity.

6.2.2 Tensor Operations and Eigenpairs

Tensors are the generalization of scalars, vectors, and matrices to m th-order multiway arrays. The tensor order (m) is the number of indices needed to define the tensor, i.e., $\{i_1, i_2, \dots, i_m\}$. The dimension n_{i_k} defines the range of index i_k . The following convention will be used throughout this chapter: n_1 -dimensional vectors are first-order tensors, $n_1 \times n_2$ matrices are second-order tensors, $n_1 \times n_2 \times n_3$ arrays are third-order tensors, and so on.

6.2.2.1 Multilinear Forms

The superscript (\cdot) will be used to specify the tensor order. First, consider an m th-order tensor $\mathcal{T}^{(m)}$ over the real field with dimensions $n = n_1 = n_2 = \dots = n_m$. The tensor $\mathcal{T}^{(m)}$ consists of n^m real entries indexed by

$$\mathcal{T}_{i_1 i_2 \dots i_m}, \quad \text{for } 1 \leq i_1, i_2, \dots, i_m \leq n. \quad (6.14)$$

A multilinear form defined by the tensor $\mathcal{T}^{(m)}$ can be written as

$$\mathcal{T}^{(m)} \mathbf{v}^m = \mathcal{T}_{i_1 i_2 \dots i_m} v_{i_1} v_{i_2} \dots v_{i_m} \quad (6.15)$$

for the vector $\mathbf{v} \in \mathbb{R}^n$. The notations on the left- and right-hand sides of Eq. (6.15) will be used interchangeably. This chapter uses a summation convention (similar to Einstein summation) in which all indices i that appear exclusively on the right-hand side of an equation indicate a summation over $i \in \{1, \dots, n_i\}$. Thus, Eq. (6.15) is equivalent to

$$\mathcal{T}^{(m)} \mathbf{v}^m = \sum_{i_1, \dots, i_m=1}^n \mathcal{T}_{i_1 i_2 \dots i_m} v_{i_1} v_{i_2} \dots v_{i_m}. \quad (6.16)$$

When indices appear on both sides of an equation, the summation is only performed over indices that are unique to the right-hand side, e.g., for the generic second-order tensors \mathcal{B} , \mathcal{C} , and \mathcal{D} , the expression

$$\mathcal{B}_{i_1 i_2} = \mathcal{C}_{i_1 k} \mathcal{D}_{k i_2} \quad (6.17)$$

is equivalent to

$$\mathcal{B}_{i_1 i_2} = \sum_{k=1}^n \mathcal{C}_{i_1 k} \mathcal{D}_{k i_2}. \quad (6.18)$$

Equation (6.15) represents a scalar-valued m th-order homogeneous polynomial of the variables v_1 through v_n . The elements of $\mathcal{T}^{(m)}$ dictate the coefficients of the homogeneous polynomial. Equation (6.15) can also be thought of as multiplying the vector \mathbf{v} by all “modes” of the tensor to produce a scalar output. This is analogous to pre- and post-multiplying an $n \times n$ matrix $\mathcal{T}^{(2)}$ by the vector \mathbf{v} to produce a scalar. The result is a second-order homogeneous polynomial of the variables v_1 through v_n :

$$\mathbf{v}^\top \mathcal{T}^{(2)} \mathbf{v} = \mathcal{T}_{i_1 i_2} v_{i_1} v_{i_2}. \quad (6.19)$$

A multilinear form may also exclude one or more tensor modes. If the first mode is excluded, then $\mathcal{T}^{(m)}\mathbf{v}^{m-1}$ is a column n -vector whose j th entry is defined by

$$(\mathcal{T}^{(m)}\mathbf{v}^{m-1})_j = \mathcal{T}_{ji_2\dots i_m}v_{i_2}\cdots v_{i_m}. \quad (6.20)$$

Each element of the vector $\mathcal{T}^{(m)}\mathbf{v}^{m-1}$ is an $(m-1)$ th-order homogeneous polynomial of the variables v_1 through v_n . In the matrix case, Eq. (6.20) is equivalent to multiplying a matrix on the right-hand side by a vector, producing a column vector output:

$$\mathcal{T}^{(2)}\mathbf{v} = \begin{bmatrix} (\mathcal{T}^{(2)}\mathbf{v})_1 \\ \vdots \\ (\mathcal{T}^{(2)}\mathbf{v})_n \end{bmatrix} = \begin{bmatrix} \mathcal{T}_{1i_2}v_{i_2} \\ \vdots \\ \mathcal{T}_{ni_2}v_{i_2} \end{bmatrix}. \quad (6.21)$$

Regarding tensors that have indices of different dimension, only one case is relevant in this chapter: $\mathcal{U}^{(m)}$ for $n = n_2 = n_3 = \cdots = n_m$ and $n_1 \neq n$. Thus, $\mathcal{U}^{(m)}$ is indexed by

$$\mathcal{U}_{i_1 i_2 \dots i_m}, \quad \text{for} \quad \begin{cases} 1 \leq i_1 \leq n_1 \\ 1 \leq i_2, i_3, \dots, i_m \leq n \end{cases}. \quad (6.22)$$

It follows that

$$(\mathcal{U}^{(m)}\mathbf{v}^{m-1})_j = \mathcal{U}_{ji_2\dots i_m}v_{i_2}\cdots v_{i_m}. \quad (6.23)$$

6.2.2.2 Tensor Supersymmetry

Tensors are considered supersymmetric if they are invariant under permutations of their indices. Consider a generic non-symmetric tensor $\mathcal{T}^{(m)}$ of dimensions $n_1 = n_2 = \cdots = n_m$. A supersymmetric tensor $\mathcal{S}^{(m)}$ may be generated such that

$$\mathcal{S}^{(m)}\mathbf{v}^m = \mathcal{T}^{(m)}\mathbf{v}^m. \quad (6.24)$$

For example, consider a multilinear form based on the second-order, two-dimensional tensor $\mathcal{T}^{(2)}$:

$$\mathcal{T}^{(2)}\mathbf{v}^2 = \mathcal{T}_{11}v_1^2 + (\mathcal{T}_{12} + \mathcal{T}_{21})v_1v_2 + \mathcal{T}_{22}v_2^2. \quad (6.25)$$

The tensor $\mathcal{S}^{(2)}$ that satisfies $\mathcal{S}^{(2)}\mathbf{v}^2 = \mathcal{T}^{(2)}\mathbf{v}^2$ is given by

$$\mathcal{S}_{11} = \mathcal{T}_{11}, \quad (6.26)$$

$$\mathcal{S}_{12} = (\mathcal{T}_{12} + \mathcal{T}_{21})/2, \quad (6.27)$$

$$\mathcal{S}_{21} = (\mathcal{T}_{12} + \mathcal{T}_{21})/2, \quad (6.28)$$

$$\mathcal{S}_{22} = \mathcal{T}_{22}. \quad (6.29)$$

$$(6.30)$$

This concept of “symmetrization” extends to tensors of higher order and dimension. This chapter uses the “symmetrize” function in the Tensor Toolbox to generate supersymmetric tensors [74]. The gradient of a supersymmetric multilinear form is easily computed as follows [75]:

$$\nabla_{\mathbf{v}} \left(\mathcal{S}^{(k)} \mathbf{v}^k \right) = k \mathcal{S}^{(k)} \mathbf{v}^{k-1}. \quad (6.31)$$

6.2.2.3 Tensor Eigenpairs

Tensor eigenpairs were originally introduced by Lim and Qi [76, 77]. Just as matrix eigenpairs are only defined for square matrices, tensor eigenpairs are only defined for tensors in which all indices are of equal dimension, i.e., the tensor in Eq. (6.14). The tensor eigenpair problem is inherently more complex than the matrix problem, leading to numerous tensor eigenpair definitions. In the matrix scenario, right-hand eigenvectors are standard and matrix eigenpairs are often defined by the equation

$$M\mathbf{v}_r = \lambda_r \mathbf{v}_r, \quad (6.32)$$

where M is a matrix, \mathbf{v}_r is a right-hand eigenvector of M , and λ_r is the corresponding eigenvalue. Equation (6.32) describes a right-hand eigenpair, as \mathbf{v}_r appears on the right-hand side of M . Although less common in applied mathematics, left-hand eigenpairs are also valid:

$$\mathbf{v}_l M = \lambda_l \mathbf{v}_l. \quad (6.33)$$

By extension, an m th-order tensor has m eigenpair modes. Equation (6.32) describes the mode-1 eigenpairs of a second-order tensor, whereas Eq. (6.33) describes mode-2 eigenpairs.

In Eqs. (6.32) and (6.33), the eigenvector appears once in each term and can be scaled arbitrarily. In contrast, some tensor eigenpair definitions do not allow arbitrary scaling of the eigenvectors. For example, the following equation is a common mode-1 tensor eigenpair definition in which (λ, \mathbf{v}) is an eigenpair of the m th-order tensor $\mathcal{T}^{(m)}$ [77]:

$$\mathcal{T}^{(m)}\mathbf{v}^{m-1} = \lambda\mathbf{v}, \quad \mathbf{v}^T\mathbf{v} = 1. \quad (6.34)$$

If $\lambda \in \mathbb{C}$ and $\mathbf{v} \in \mathbb{C}^n$, the pair is referred to as an E-eigenpair [77]. Otherwise, if $\lambda \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^n$, the pair is a Z-eigenpair [78]. Equation (6.34) may appear similar to Eq. (6.32). However, for $m > 2$, the eigenvector cannot be scaled arbitrarily and must be normalized. In addition to E- and Z-eigenpairs, numerous eigenpair definitions exist, including many in which the eigenvector may be scaled arbitrarily. Generalized eigenpairs have also been proposed to consolidate existing eigenpair definitions in a unified framework [78, 79]. In this chapter, the Z-eigenpair is used to compute the TEMoN. Therefore, other eigenpair definitions are omitted here.

Computing tensor eigenpairs involves solving a large system of polynomials. The results in this chapter will use the dynamical systems (DS) method in [80, 81], which can locate Z-eigenpairs of supersymmetric and non-symmetric tensors. In [80], the DS method is shown to be faster than other, less flexible eigenpair methods including the shifted symmetric higher-order power method [74, 82] and semidefinite programming [83]. The DS method is based on an observation relating tensor and matrix eigenvectors: a tensor Z-eigenvector must also be an eigenvector of the “collapsed” tensor. Collapsing a tensor refers to a tensor-vector multiplication that excludes the first and second tensor modes to produce a matrix output. A mapping function is then formulated to locate an eigenvector of the collapsed tensor. The mapping function is incorporated in a dynamical system that must converge on a tensor eigenvector, and eigenvectors are computed through numerical integration.

In this chapter, the DS method is implemented with a fourth-order Runge-Kutta (RK4) using the default tolerance in [81]. Mapping functions are selected to target eigenvectors which correspond to the largest algebraic eigenvalue, the largest magnitude eigenvalue, the second-largest algebraic eigenvalue, the second-largest magnitude eigenvalue, the smallest algebraic eigenvalue, and the second-smallest algebraic

eigenvalue. Other mapping functions are possible, but are not utilized here. The algorithm is repeated for multiple random initial guesses to converge on different eigenvectors. The number of initial guesses and the maximum number of RK4 iterations is varied based on application and tensor order.

Some properties of Z-eigenpairs can be used to accelerate the eigenpair computation. For an even-order tensor, if (λ, \mathbf{v}) is an eigenpair, then $(\lambda, -\mathbf{v})$ is also an eigenpair. Alternatively, for an odd-order tensor, if (λ, \mathbf{v}) is an eigenpair, then $(-\lambda, -\mathbf{v})$ is also an eigenpair. Moreover, the number of real eigenpairs is bounded by $((m-1)^n - 1)/(m-2)$ [84] for $m \geq 3$ ($m = 2$ corresponds to the matrix eigenpair problem). Thus, a six-dimensional, third-order tensor may have up to 63 eigenpairs, a six-dimensional fourth-order tensor may have up to 364 eigenpairs, and so on.

6.2.3 Taylor Series Expansions

This section describes some TSE of interest, including expansions of state dynamics, solution flows, and parameter transformations. Consider the general nonlinear function $\boldsymbol{\eta}(\mathbf{V})$ and allow $\mathbf{v} = \mathbf{V} - \mathbf{V}^*$, where \mathbf{V}^* are the nominal parameters. The TSE of $\boldsymbol{\eta}$ with respect to \mathbf{V}^* was given previously in Eq. (6.12). For simplicity of notation, each tensor can be replaced by $\mathcal{T}^{(m)}$ such that

$$\delta\boldsymbol{\eta} = \mathcal{T}^{(2)}\mathbf{v} + \frac{1}{2!}\mathcal{T}^{(3)}\mathbf{v}^2 + \frac{1}{3!}\mathcal{T}^{(4)}\mathbf{v}^3 + \frac{1}{4!}\mathcal{T}^{(5)}\mathbf{v}^4 + \dots \quad (6.35)$$

represents the TSE of $\boldsymbol{\eta}$ with respect to the nominal parameters \mathbf{V}^* . It is assumed that each tensor is evaluated at the nominal. The multilinear terms, $\mathcal{T}^{(m)}\mathbf{v}^{m-1}$ for $m > 2$, comprise the nonlinear portion of $\delta\boldsymbol{\eta}$.

It is important to distinguish between the truncation order of a TSE and the order of the tensor associated with each TSE term. For example, the state transition matrix (STM) is a second-order ($n \times n$) tensor by convention. However, the STM is associated with the first-order derivative in the TSE. To avoid confusion, the tensor order is denoted by m and the derivative order is denoted by p throughout this section; $p = m - 1$. All terms $p \geq 2$ and $m \geq 3$ are “higher-order” terms that contribute to the model nonlinearity.

The squared L^2 norm of $\delta\boldsymbol{\eta}$, i.e., $\|\delta\boldsymbol{\eta}\|^2 = \delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta}$, can also be written as a summation of multilinear

forms:

$$\delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta} = \tilde{\mathcal{T}}^{(2)}\mathbf{v}^2 + \tilde{\mathcal{T}}^{(3)}\mathbf{v}^3 + \tilde{\mathcal{T}}^{(4)}\mathbf{v}^4 + \tilde{\mathcal{T}}^{(5)}\mathbf{v}^5 + \dots \quad (6.36)$$

The tensors $\tilde{\mathcal{T}}^{(m)}$ are functions of all $\mathcal{T}^{(m')}$ for $m' \leq m$. Their calculation is detailed in Appendix C for orders $m = \{2, 3, 4, 5\}$. In Eq. (6.36), $\tilde{\mathcal{T}}^{(2)}\mathbf{v}^2$ will be referred to as the *linear* contribution because it is the only term that is non-zero when $\boldsymbol{\eta}$ is linear. The terms $\tilde{\mathcal{T}}^{(m)}\mathbf{v}^m$ for $m > 2$ constitute the *nonlinear* contribution to $\delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta}$.

6.2.3.1 LDTs and STTs

Local dynamics tensors (LDTs) and state transition tensors (STTs) constitute the higher-order terms in a TSE of a system's dynamics and solution flow, respectively [62]. Consider the following state dynamics

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t), \quad (6.37)$$

which correspond to the solution flow

$$\mathbf{x}(t) = \boldsymbol{\phi}(t; \mathbf{x}(t_0), t_0), \quad (6.38)$$

where \mathbf{x} is an n -dimensional state vector and $\mathbf{x}(t_0)$ is the initial state at time t_0 . A nominal trajectory will be denoted \mathbf{x}^* , and deviations from the nominal trajectory will be labeled $\delta\mathbf{x}$ such that:

$$\delta\mathbf{x} = \mathbf{x} - \mathbf{x}^* \quad (6.39)$$

and

$$\delta\dot{\mathbf{x}} = \dot{\mathbf{x}} - \dot{\mathbf{x}}^*. \quad (6.40)$$

A p th-order TSE of the dynamics evaluated at the nominal state results in:

$$\delta\dot{\mathbf{x}} = \sum_{k=1}^p \frac{1}{k!} \left. \frac{\partial^k \dot{\mathbf{x}}}{\partial \mathbf{x}^k} \right|_{\mathbf{x}^*} \delta\mathbf{x}^k, \quad (6.41)$$

$$= \sum_{k=1}^p \frac{1}{k!} \mathcal{A}^{(k+1)} \delta\mathbf{x}^k, \quad (6.42)$$

where $\mathcal{A}^{(k+1)}$ is a $(k+1)$ th-order tensor. Specifically, $\mathcal{A}^{(k+1)}$ is an LDT. Each term in Eq. (6.42) corresponds to a mode-1 multilinear form as in Eq. (6.20).

Similarly, a p th-order TSE of the solution flow evaluated at $\mathbf{x}^*(t_0)$ results in

$$\delta \mathbf{x}(t) = \sum_{k=1}^p \frac{1}{k!} \frac{\partial^k \mathbf{x}(t)}{\partial \mathbf{x}(t_0)^k} \Big|_{\mathbf{x}^*(t_0)} \delta \mathbf{x}(t_0)^k, \quad (6.43)$$

$$= \sum_{k=1}^p \frac{1}{k!} \Phi^{(k+1)} \delta \mathbf{x}(t_0)^k, \quad (6.44)$$

where $\Phi^{(k+1)}$ is the $(k+1)$ th-order STT. The state and time dependencies of \mathcal{A} and Φ are omitted in the notation for simplicity.

The multilinear forms in Eqs. (6.42) and (6.44) can be expressed in the summation convention of Eq. (6.20) as follows:

$$\delta \dot{x}_j = \sum_{k=1}^p \frac{1}{k!} \mathcal{A}_{j i_1 \dots i_k} \delta x_{i_1} \dots \delta x_{i_k}, \quad (6.45)$$

$$\mathcal{A}_{j i_1 \dots i_k} = \frac{\partial^k f_j}{\partial x_{i_1} \dots \partial x_{i_k}} \Big|_{\mathbf{x}=\mathbf{x}^*}, \quad (6.46)$$

$$\delta x_j(t) = \sum_{k=1}^p \frac{1}{k!} \Phi_{j i_1 \dots i_k} \delta x_{i_1}(t_0) \dots \delta x_{i_k}(t_0), \quad (6.47)$$

$$\Phi_{j i_1 \dots i_k} = \frac{\partial^k x_j(t)}{\partial x_{i_1}(t_0) \dots \partial x_{i_k}(t_0)} \Big|_{\mathbf{x}(t_0)=\mathbf{x}^*(t_0)}, \quad (6.48)$$

where x_j is the j th element of the state \mathbf{x} . The LDTs can be computed via analytic or automatic differentiation. The STTs are typically computed by numerical integration. The STT dynamics are derived by equating Eq. (6.42) with the time derivative of Eq. (6.44), as in [62]. The STT dynamics are given in Appendix C.

6.2.3.2 Parameter Transformations

TSE of algebraic functions may also be of interest, such nonlinear parameter transformations or nonlinear measurement models used during state estimation. From a mission design perspective, the nonlinearity of a parameter transformation is important when different parameters are used for different mission

operations. For example, the dynamics of the Milankovitch orbital elements are known to be relatively linear, making the Milankovitch elements a clever parameterization for uncertainty propagation. Conversely, state estimation and control algorithms are often parameterized in the Cartesian frame. A natural approach may be to transform a Cartesian state to Milankovitch elements, perform uncertainty propagation using the Milankovitch elements, and convert back to the Cartesian state at the end of the propagation. A similar transformation-propagation-transformation (TPT) strategy is summarized as follows:

- (1) Consider the parameter sets $\mathbf{q} \in \mathbb{R}^{n_q}$ and $\mathbf{w} \in \mathbb{R}^{n_w}$, which share the nonlinear relationship $\mathbf{q} = \boldsymbol{\eta}(\mathbf{w})$ and the inverse relationship $\mathbf{w} = \boldsymbol{\eta}^{-1}(\mathbf{q})$.
- (2) At time t_0 , transform $\mathbf{w}(t_0)$ to $\mathbf{q}(t_0)$.
- (3) Propagate \mathbf{q} from time t_0 to time t .
- (4) At time t , convert $\mathbf{q}(t)$ to $\mathbf{w}(t)$.

Although the dynamics of \mathbf{q} may be relatively linear, the parameter transformations at steps 2 and 4 are nonlinear and the nonlinearity of $\dot{\mathbf{w}}$ has not entirely been avoided. Consider a nominal trajectory described by either \mathbf{w}^* or \mathbf{q}^* . Deviations from the nominal are denoted by either $\delta\mathbf{w}$ or $\delta\mathbf{q}$. Beginning with the parameter transformation in step 2, a TSE of $\mathbf{q}(t_0)$ about the reference point $\mathbf{w}^*(t_0)$ results in

$$\delta\mathbf{q}(t_0) = \sum_{k=1}^{\infty} \frac{1}{k!} \left. \frac{\partial^k \mathbf{q}}{\partial \mathbf{w}^k} \right|_{\mathbf{w}^*(t_0)} \delta\mathbf{w}(t_0)^k, \quad (6.49)$$

$$= \sum_{k=1}^{\infty} \frac{1}{k!} \mathbf{Q}^{(k+1)} \delta\mathbf{w}(t_0)^k. \quad (6.50)$$

The propagation of $\delta\mathbf{q}$ from t_0 to t (step 3) is described by the STTs:

$$\delta\mathbf{q}(t) = \sum_{k=1}^{\infty} \frac{1}{k!} \boldsymbol{\Phi}^{(k+1)} \delta\mathbf{q}(t_0)^k. \quad (6.51)$$

Finally, the reverse parameter transformation in step 4 calls for a TSE of $\delta\mathbf{w}(t)$ about the reference point $\mathbf{q}^*(t)$:

$$\delta\mathbf{w}(t) = \sum_{k=1}^{\infty} \frac{1}{k!} \left. \frac{\partial^k \mathbf{w}}{\partial \mathbf{q}^k} \right|_{\mathbf{q}^*(t)} \delta\mathbf{q}(t)^k, \quad (6.52)$$

$$= \sum_{k=1}^{\infty} \frac{1}{k!} \mathbf{W}^{(k+1)} \delta\mathbf{q}(t)^k. \quad (6.53)$$

If $n_q = n_s$, then every index of $\mathcal{Q}^{(m)}$ and $\mathcal{S}^{(m)}$ are of equal dimension, as in Eq. (6.14). Otherwise, if $n_q \neq n_s$, then $\mathcal{Q}^{(m)}$ is of dimensions $n_q \times n_s \times \cdots \times n_s$ and $\mathcal{S}^{(m)}$ is of dimensions $n_s \times n_q \times \cdots \times n_q$, as in Eq. (6.22).

Combining Eqs. (6.50), (6.51), and (6.53) results in

$$\delta\mathbf{w}(t) = \sum_{k=1}^{\infty} \frac{1}{k!} \mathcal{W}^{(k+1)} \left[\sum_{j=1}^{\infty} \frac{1}{j!} \Phi^{(j+1)} \left(\sum_{i=1}^{\infty} \frac{1}{i!} \mathcal{Q}^{(i+1)} \delta\mathbf{w}(t_0)^i \right)^j \right]^k. \quad (6.54)$$

The above relationship between $\delta\mathbf{w}(t_0)$ and $\delta\mathbf{w}(t)$ can be expressed more concisely as follows

$$\delta\mathbf{w}(t) = \sum_{k=1}^{\infty} \frac{1}{k!} \Omega^{(k+1)} \delta\mathbf{w}(t_0)^k. \quad (6.55)$$

Effectively, the Ω tensors encompass the entire TPT process and are not equal to the STTs of \mathbf{w} in general. Moreover, the nonlinearity of the TPT process is not necessarily equal to the nonlinearity of directing propagating the dynamics in terms of the parameters \mathbf{w} . To analyze the nonlinearity of the entire TPT process, the TEMoN may be applied directly to the set of Ω tensors. Only the $\Omega^{(2)}$ and $\Omega^{(3)}$ tensors will be considered in this chapter, and their computation is detailed in Appendix C.

6.3 TEMoN Problem Setup

The TEMoN is proposed in this section. The TEMoN will utilize the higher-order terms of a TSE to quantify nonlinearity. The influence of each term in the TSE is dependent on the input deviation considered, i.e., $\delta\mathbf{x}$ in Eq. (6.42), $\delta\mathbf{x}(t_0)$ in Eq. (6.44), or $\delta\mathbf{w}(t_0)$ in Eq. (6.55). The TEMoN has two primary goals: determine input deviations that produce the strong nonlinearity and quantify the associated degree of nonlinearity.

6.3.1 TEMoN Definition

Recall from Eq. (6.36) that $\delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta}$ includes linear and nonlinear terms. The term $\tilde{\mathcal{T}}^{(2)} \mathbf{v}^2$ is referred to as the linear term because it is the only non-zero term when $\boldsymbol{\eta}$ is linear. Higher-order terms comprise the nonlinear portion of $\delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta}$. Consider the following nonlinearity measure in which the influence of these

nonlinear terms is maximized over a subset of input vectors \mathbf{v} :

$$\aleph_{\mathcal{T},m} := \max_{\mathbf{v} \in \mathcal{V}} \left| \frac{\tilde{\mathcal{T}}^{(3)}\mathbf{v}^3 + \tilde{\mathcal{T}}^{(4)}\mathbf{v}^4 + \dots + \tilde{\mathcal{T}}^{(m)}\mathbf{v}^m}{\tilde{\mathcal{T}}^{(2)}\mathbf{v}^2} \right| \quad \text{for } m > 2. \quad (6.56)$$

The “ \mathcal{T}, m ” subscript specifies the type of TSE expansion and the truncation order of the nonlinear terms.

The subset \mathcal{V} includes all real-valued vectors of magnitude α :

$$\mathcal{V} = \left\{ \mathbf{v} \in \mathbb{R}^n \mid \mathbf{v}^\top \mathbf{v} = \alpha^2 \right\}. \quad (6.57)$$

Normalizing higher-order terms by the linear term is important when quantifying nonlinearity. A higher-order term will be insignificant if it is dominated by the linear term and vice versa. If the order m is sufficiently large, then $\aleph_{\mathcal{T},m}$ approximates a ratio of the total model nonlinearity to the model linearity. However, $\aleph_{\mathcal{T},m}$ presents a nontrivial nonlinear programming problem.

Alternatively, consider the TEMoN denoted by $\mathsf{T}_{\mathcal{T},m}$:

$$\mathsf{T}_{\mathcal{T},m} := \max_{\mathbf{v} \in \mathcal{V}} \tau_3 + \max_{\mathbf{v} \in \mathcal{V}} \tau_4 + \dots + \max_{\mathbf{v} \in \mathcal{V}} \tau_m \quad \text{for } m > 2, \quad (6.58)$$

$$= \tau_3^* + \tau_4^* + \dots + \tau_m^*, \quad (6.59)$$

where

$$\tau_k := \left| \frac{\tilde{\mathcal{T}}^{(k)}\mathbf{v}^k}{\tilde{\mathcal{T}}^{(2)}\mathbf{v}^2} \right|. \quad (6.60)$$

Unlike \aleph , the TEMoN considers each higher-order term individually and will produce a direction of strong nonlinearity for each τ_k . By inspection of Eqs. (6.56) and (6.58), it is clear that

$$\mathsf{T}_{\mathcal{T},m} \geq \aleph_{\mathcal{T},m} \quad (6.61)$$

and the TEMoN is an upper bound on \aleph . The TEMoN can be computed semianalytically as seen in the next section. In Sections 6.5 and 6.6, the TEMoN is compared to an approximation of \aleph by randomly sampling N different input deviations:

$$\aleph_{\mathcal{T},m} \approx \max_{i=1,\dots,N} \left| \frac{\tilde{\mathcal{T}}^{(3)}\mathbf{v}_i^3 + \tilde{\mathcal{T}}^{(4)}\mathbf{v}_i^4 + \dots + \tilde{\mathcal{T}}^{(m)}\mathbf{v}_i^m}{\tilde{\mathcal{T}}^{(2)}\mathbf{v}_i^2} \right| \quad \text{for } m > 2. \quad (6.62)$$

The TEMoN itself can also be approximated via random sampling in a similar manner. A ‘‘sampled TEMoN’’ (S-TEMoN) will be used to validate the TEMoN. The S-TEMoN is denoted $S_{\mathcal{T},m}$:

$$S_{\mathcal{T},m} := \max_{i=1,\dots,N} s_3 + \max_{i=1,\dots,N} s_4 + \dots + \max_{i=1,\dots,N} s_m \quad \text{for } m > 2, \quad (6.63)$$

$$= s_3^* + s_4^* + \dots + s_m^*, \quad (6.64)$$

$$s_k = \left| \frac{\tilde{\mathcal{T}}^{(m)} \mathbf{v}_i^m}{\tilde{\mathcal{T}}^{(2)} \mathbf{v}_i^2} \right|. \quad (6.65)$$

6.3.2 Relating α to State Uncertainty

For a deterministic input deviation, the deviation magnitude is $\alpha^2 = \mathbf{v}^\top \mathbf{v}$ from Eq. (6.57). When the input is nondeterministic, α can be related to parameter uncertainty by making the following assumption

$$\alpha^2 = \mathbb{E} \left[\mathbf{v}^\top \mathbf{v} \right]. \quad (6.66)$$

Furthermore, assuming that the input deviation is zero-mean and Gaussian, i.e., $\mathbf{v} = \mathcal{N}(\mathbf{0}, \Sigma)$,

$$\alpha^2 = \text{tr} \left(\mathbb{E} \left[\mathbf{v} \mathbf{v}^\top \right] \right), \quad (6.67)$$

$$= \text{tr}(\Sigma), \quad (6.68)$$

where $\text{tr}(\cdot)$ represents the matrix trace. Then α is computed as

$$\alpha = \sqrt{\text{tr}(\Sigma)}. \quad (6.69)$$

When applying the TEMoN to the STTs such that $\mathbf{v} = \delta \mathbf{x}(t_0)$, this process is equivalent to assuming that the initial state error is zero-mean and Gaussian. The Gaussian assumption is made at the initial epoch only, and it is not necessary to assume that the uncertainty distribution remains Gaussian over time.

6.4 TEMoN Result

6.4.1 TEMoN Derivation

Each τ_k term in the TEMoN can be optimized individually. The first step in the optimization of τ_k is to define a constrained Lagrangian as follows

$$L_k = \frac{\tilde{\mathcal{T}}^{(k)} \mathbf{v}^k}{\tilde{\mathcal{T}}^{(2)} \mathbf{v}^2} - \gamma \left(\mathbf{v}^\top \mathbf{v} - \alpha^2 \right). \quad (6.70)$$

The scalar γ is a Lagrange multiplier, and a constraint is placed on the magnitude of the deviation vector, \mathbf{v} . The absolute value operator is not included in the Lagrangian; instead, the maximum and minimum will both be considered. The vector \mathbf{v}^* that maximizes or minimizes the Lagrangian will be a Karush-Kuhn-Tucker point of the Lagrangian such that

$$\nabla L_k(\mathbf{v}^*) = \mathbf{0}. \quad (6.71)$$

Next, Eq. (6.70) will be simplified by making the substitution

$$\mathbf{v} = \left(\mathcal{T}^{(2)} \right)^{-1} \mathbf{z} \quad (6.72)$$

and optimizing the vector \mathbf{z} rather than \mathbf{v} . This substitution may also be thought of as performing a linear transformation of parameters to facilitate ease of computation and promote numerical stability. Note that a linear parameter transformation does not alter the inherent nonlinearity of the model [73]. Applying the parameter transformation to $\tilde{\mathcal{T}}^{(2)} \mathbf{v}^2$ (noting that $\tilde{\mathcal{T}}^{(2)} = \mathcal{T}^{(2)\top} \mathcal{T}^{(2)}$ from Appendix C):

$$\tilde{\mathcal{T}}^{(2)} \mathbf{v}^2 = \mathbf{v}^\top \mathcal{T}^{(2)\top} \mathcal{T}^{(2)} \mathbf{v}, \quad (6.73)$$

$$= \mathbf{z}^\top \left(\mathcal{T}^{(2)} \right)^{-\top} \mathcal{T}^{(2)\top} \mathcal{T}^{(2)} \left(\mathcal{T}^{(2)} \right)^{-1} \mathbf{z}, \quad (6.74)$$

$$= \mathbf{z}^\top \mathbf{z}. \quad (6.75)$$

Incorporating the parameter transformation in the Lagrangian results in

$$L_k = \frac{\tilde{\mathcal{T}}^{(k)} \left(\left(\mathcal{T}^{(2)} \right)^{-1} \mathbf{z} \right)^k}{\mathbf{z}^\top \mathbf{z}} - \gamma \left(\mathbf{z}^\top Y \mathbf{z} - \alpha^2 \right), \quad (6.76)$$

where

$$Y = \left(\mathcal{T}^{(2)} \right)^{-\top} \left(\mathcal{T}^{(2)} \right)^{-1}. \quad (6.77)$$

The numerator in Eq. (6.76) can be reformulated as a multilinear map of \mathbf{z} :

$$L_k = \frac{\check{\mathcal{T}}^{(k)} \mathbf{z}^k}{\mathbf{z}^\top \mathbf{z}} - \gamma \left(\mathbf{z}^\top \mathbf{Y} \mathbf{z} - \alpha^2 \right). \quad (6.78)$$

The $\check{\mathcal{T}}^{(k)}$ computation is detailed in Appendix C. Furthermore, replacing $\check{\mathcal{T}}^{(k)} \mathbf{z}^k$ with the equivalent supersymmetric expression will simplify future computations; $\check{\mathcal{S}}^{(k)}$ is the supersymmetric tensor that satisfies

$$\check{\mathcal{S}}^{(k)} \mathbf{z}^k = \check{\mathcal{T}}^{(k)} \mathbf{z}^k. \quad (6.79)$$

In the next step, $\check{\mathcal{T}}^{(k)}$ is replaced by $\check{\mathcal{S}}^{(k)}$, which does not change the value of the Lagrangian:

$$L_k = \frac{\check{\mathcal{S}}^{(k)} \mathbf{z}^k}{\mathbf{z}^\top \mathbf{z}} - \gamma \left(\mathbf{z}^\top \mathbf{Y} \mathbf{z} - \alpha^2 \right). \quad (6.80)$$

Computing the gradient of L_k with respect to the Lagrange multiplier γ results in

$$\mathbf{z}^\top \mathbf{Y} \mathbf{z} = \alpha^2. \quad (6.81)$$

Using Eq. (6.31) to compute the gradient of L_k with respect to \mathbf{z} ,

$$\nabla_{\mathbf{z}} L_k = \frac{(\mathbf{z}^\top \mathbf{z}) k \check{\mathcal{S}}^{(k)} \mathbf{z}^{k-1} - 2 \check{\mathcal{S}}^{(k)} \mathbf{z}^k \mathbf{z}}{(\mathbf{z}^\top \mathbf{z})^2} - 2\gamma \mathbf{Y} \mathbf{z}, \quad (6.82)$$

$$= 0. \quad (6.83)$$

It follows that

$$\mathbf{Y}^{-1} \left[(\mathbf{z}^\top \mathbf{z}) k \check{\mathcal{S}}^{(k)} \mathbf{z}^{k-1} - 2 \check{\mathcal{S}}^{(k)} \mathbf{z}^k \mathbf{z} \right] = 2\gamma (\mathbf{z}^\top \mathbf{z})^2 \mathbf{z}. \quad (6.84)$$

Equation (6.84) is a column vector equation, and each term on the left-hand side is a $(k + 1)$ th-order homogeneous polynomial. Therefore, a mode-1 multilinear form can be constructed from a $(k + 2)$ th-order tensor that produces the column vector output on the left-hand side of Eq. (6.84). The corresponding $(k + 2)$ th-order tensor will be denoted by $\Gamma^{(k+2)}$ such that

$$\Gamma^{(k+2)} \mathbf{z}^{k+1} = 2\gamma (\mathbf{z}^\top \mathbf{z})^2 \mathbf{z}. \quad (6.85)$$

The computation of $\Gamma^{(k+2)}$ is detailed in Appendix C. Finally, the magnitude of \mathbf{z} is factored out so that the following equation describes a Z-eigenpair relationship:

$$\Gamma^{(k+2)} \hat{\mathbf{z}}^{k+1} = \lambda \hat{\mathbf{z}}. \quad (6.86)$$

The eigenvalue, λ , is a function of $\|z\|$:

$$\lambda = 2\gamma\|z\|^{4-k}. \quad (6.87)$$

However, λ is a property of the tensor $\Gamma^{(k+2)}$ and remains scale invariant; the Lagrange multiplier, γ , will vary to accommodate changes in $\|z\|$. Thus, the eigenvectors of $\Gamma^{(k+2)}$ produce Karush-Kuhn-Tucker points of the Lagrangian. To determine the optimal eigenvector, each eigenvector of $\Gamma^{(k+2)}$ must be substituted back into τ_k for comparison. From Eq. (6.81),

$$z = \frac{\alpha \hat{z}}{\left\| (\mathcal{T}^{(2)})^{-1} \hat{z} \right\|}. \quad (6.88)$$

The eigenpair that maximizes τ_k will be denoted by $(\lambda_k^*, \hat{z}_k^*)$.

6.4.2 TEMoN Algorithm

A pseudoalgorithm to compute the TEMoN is presented in Algorithm 7. Each τ_k in the TEMoN is maximized individually. From the Karush-Kuhn-Tucker conditions, the vector that maximizes τ_k corresponds to a Z-eigenvector of the tensor $\Gamma^{(k+2)}$, which is computed from the TSE and is a non-symmetric tensor in general. The DS method in [80] may be used to compute Z-eigenpairs of $\Gamma^{(k+2)}$. The eigenvector that maximizes τ_k is denoted by \hat{z}_k^* such that

$$\tau_k^* = \frac{\check{\mathcal{S}}^{(k)} \hat{z}_k^{*k}}{\left\| (\mathcal{T}^{(2)})^{-1} \hat{z}_k^* \right\|^{k-2}} \alpha^{k-2}. \quad (6.89)$$

Individual τ_k^* are summed to compute the TEMoN:

$$\mathbf{T}_{\mathcal{T},m} = \sum_{k=3}^m \tau_k^*. \quad (6.90)$$

The input deviation vector that produces strong nonlinearity, \mathbf{v}_k^* , is computed from \hat{z}_k^* as follows:

$$\mathbf{v}_k^* = \frac{\alpha (\mathcal{T}^{(2)})^{-1} \hat{z}_k^*}{\left\| (\mathcal{T}^{(2)})^{-1} \hat{z}_k^* \right\|}. \quad (6.91)$$

The magnitude of the deviation vector (α) can be scaled independent of the eigenpair computation. This is an important feature that allows the user to test different deviation magnitudes without recomputing the eigenpairs.

Algorithm 7: Computing the TEMoN and identifying directions of strong nonlinearity.

Data: Tensors $\mathcal{T}^{(2)}, \mathcal{T}^{(3)}, \dots, \mathcal{T}^{(m)}$ for $m > 2$, scalar $\alpha = \|\mathbf{v}\|$.

Result: TEMoN $_m$, directions of strong nonlinearity \mathbf{v}_3^* through \mathbf{v}_m^* .

for $k = 3$ **to** m **do**

 Compute $\tilde{\mathcal{T}}^{(k)}$ from $\mathcal{T}^{(2)}, \mathcal{T}^{(3)}, \dots, \mathcal{T}^{(k)}$ (see Appendix C).

 Compute $\check{\mathcal{T}}^{(k)}$ from $\mathcal{T}^{(2)}$ and $\tilde{\mathcal{T}}^{(k)}$ (see Appendix C).

 Symmetrize $\check{\mathcal{T}}^{(k)}$ to get $\check{\mathcal{S}}^{(k)}$ (see Section 6.2.2.2).

 Compute $\Gamma^{(k+2)}$ from $\check{\mathcal{S}}^{(k)}$ (see Appendix C).

 Compute all Z-eigenpairs of $\Gamma^{(k+2)}$ using the DS method (see Section 6.2.2.3).

 Identify $\hat{\mathbf{z}}_k^*$ as the eigenvector that maximizes $\frac{\check{\mathcal{S}}^{(k)} \hat{\mathbf{z}}_k^k}{\|(\mathcal{T}^{(2)})^{-1} \hat{\mathbf{z}}_k^*\|^{k-2}}$

 Compute direction of strong nonlinearity $\hat{\mathbf{v}}_k^* = \frac{(\mathcal{T}^{(2)})^{-1} \hat{\mathbf{z}}_k^*}{\|(\mathcal{T}^{(2)})^{-1} \hat{\mathbf{z}}_k^*\|}$

 Compute $\tau_k^* = \frac{\check{\mathcal{S}}^{(k)} \hat{\mathbf{z}}_k^{*k}}{\|(\mathcal{T}^{(2)})^{-1} \hat{\mathbf{z}}_k^*\|^{k-2}} \alpha^{k-2}$

end

Compute $\mathbf{T}_{\mathcal{T},m} = \sum_{k=3}^m \tau_k^*$

Optional: Scale α and recompute τ_k^* and $\mathbf{T}_{\mathcal{T},m}$ without repeating the eigenpair computation.

6.5 Application to LDTs

In this section, the TEMoN is applied to LDTs which are evaluated at equilibrium points in the CR3BP. The CR3BP dynamics are detailed in Appendix B. The CR3BP has five equilibrium points, also known as Lagrange points, which are denoted by L_1 through L_5 . In this example, the fifth-order TEMoN is applied to the LDTs at each Lagrange point as follows

$$T_{\mathcal{A},5} = \max_{\delta \mathbf{x} \in \mathcal{X}} \left| \frac{\tilde{\mathcal{A}}^{(3)} \delta \mathbf{x}^3}{\tilde{\mathcal{A}}^{(2)} \delta \mathbf{x}^2} \right| + \max_{\delta \mathbf{x} \in \mathcal{X}} \left| \frac{\tilde{\mathcal{A}}^{(4)} \delta \mathbf{x}^4}{\tilde{\mathcal{A}}^{(2)} \delta \mathbf{x}^2} \right| + \max_{\delta \mathbf{x} \in \mathcal{X}} \left| \frac{\tilde{\mathcal{A}}^{(5)} \delta \mathbf{x}^5}{\tilde{\mathcal{A}}^{(2)} \delta \mathbf{x}^2} \right|, \quad (6.92)$$

where $\mathcal{A}^{(k)}$ are the LDTs. From Eq. (6.61), the TEMoN is an upper bound on:

$$\aleph_{\mathcal{A},5} \approx \max_{i=1,\dots,N} \left| \frac{\tilde{\mathcal{A}}^{(3)} \delta \mathbf{x}_i^3 + \tilde{\mathcal{A}}^{(4)} \delta \mathbf{x}_i^4 + \tilde{\mathcal{A}}^{(5)} \delta \mathbf{x}_i^5}{\tilde{\mathcal{A}}^{(2)} \delta \mathbf{x}_i^2} \right|. \quad (6.93)$$

6.5.1 LDT TEMoN Validation

To confirm that Algorithm 7 is a valid computation of the TEMoN, each term in the TEMoN (τ_k^*) will be compared to each term of the empirically-sampled S-TEMoN (s_k^*) for equilibrium points in the CR3BP. A deviation magnitude of $\alpha = 10^{-3}$ will be used in this example. Figure 6.1 includes a comparison of:

- (1) τ_3^* through τ_5^* computed from Algorithm 7,
- (2) s_3^* through s_5^* computed with $N = 10^5$ random samples of $\delta \mathbf{x}$, and
- (3) s_3^* through s_5^* computed with $N = 10^5$ random samples of $\hat{\mathbf{z}}$, which is then converted to $\delta \mathbf{x}$ via

$$\delta \mathbf{x} = \frac{\alpha (\mathcal{A}^{(2)})^{-1} \hat{\mathbf{z}}}{\|(\mathcal{A}^{(2)})^{-1} \hat{\mathbf{z}}\|}. \quad (6.94)$$

Figure 6.1 indicates that the eigenpair method in Algorithm 7 is a valid optimization of the TEMoN; τ_k^* and the s_k^* are in close agreement when $\hat{\mathbf{z}}$ is sampled instead (option 3, Figure 6.1 right). However, when $\delta \hat{\mathbf{x}}$ is sampled directly (option 2, Figure 6.1 center), the S-TEMoN underestimates strong nonlinearity when compared to the TEMoN. Thus, the parameter transformation from $\delta \mathbf{x}$ to \mathbf{z} reduces the sensitivity of S-TEMoN to the input deviation and is recommended to improve the accuracy of the S-TEMoN and \aleph .

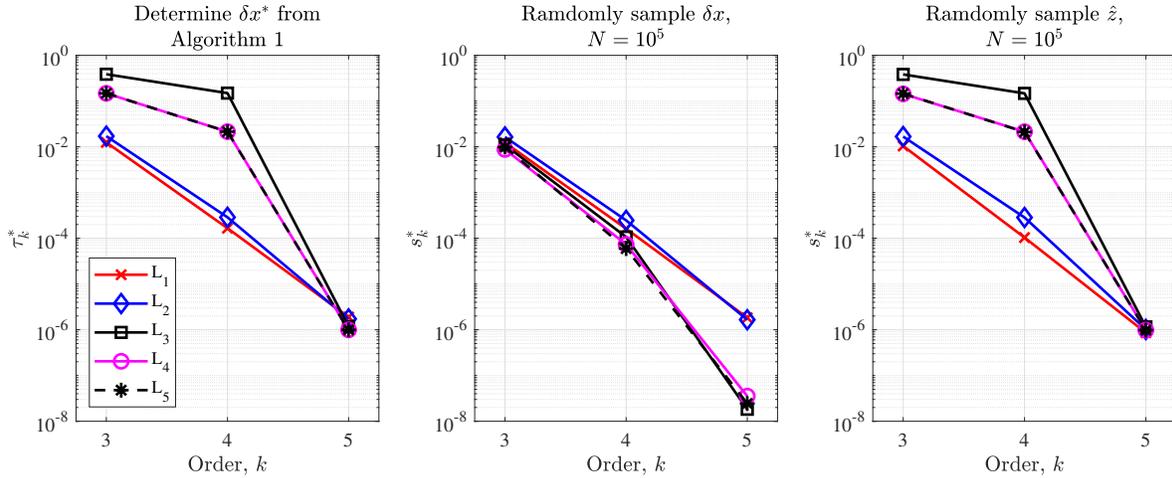


Figure 6.1: Comparing τ_k^* (TEMoN terms) to s_k^* (S-TEMoN terms) at Lagrange points L_1 through L_5 ($\|\delta x\| = 10^{-3}$).

Figure 6.2 shows a direct comparison of τ_k^* and s_k^* when s_k^* is computed from randomly sampling \hat{z} . For each equilibrium point and tensor order, τ_k^* is greater than s_k^* by up to 55%. Therefore, the TEMoN identifies stronger nonlinearity and is superior to the empirically-sampled S-TEMoN.

Next, $\aleph_{\mathcal{A},5}$ is approximated with $N = 10^5$ random samples of \hat{z} to demonstrate that the TEMoN is indeed an upper bound on \aleph . This comparison is shown in Figure 6.3 for $\alpha = 10^{-3}$. The TEMoN is roughly 2% to 20% greater than \aleph , and is a relatively tight upper bound.

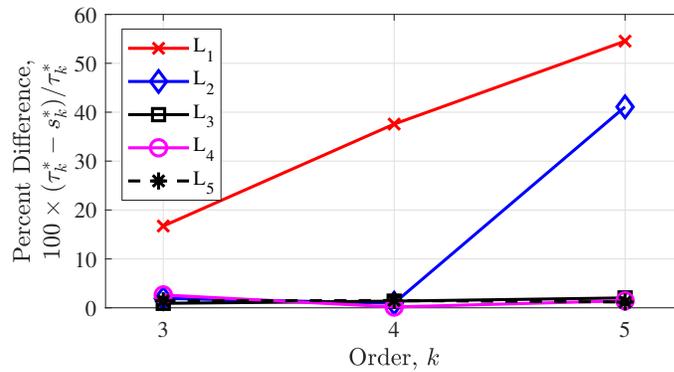


Figure 6.2: Comparing TEMoN terms (τ_k^*) to empirically-sampled S-TEMoN terms (s_k^*); the TEMoN detects stronger nonlinearity and is more accurate than the S-TEMoN in all cases.

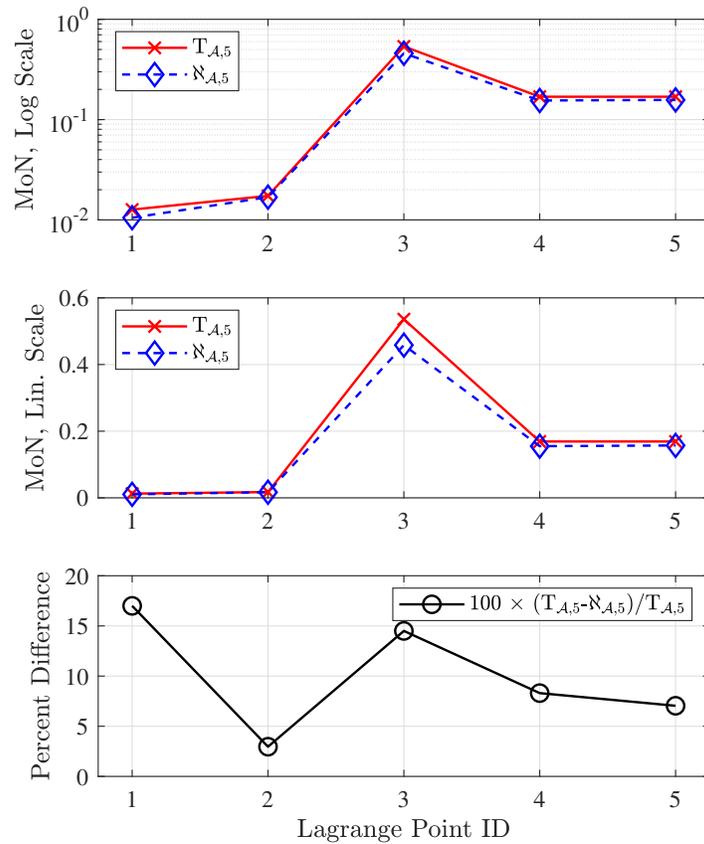


Figure 6.3: Comparing the TEMoN and \aleph for $\|\delta x\| = 10^{-3}$ at each Lagrange point. The TEMoN is a relatively tight upper bound on \aleph .

6.5.2 Comparison to NLI and CMoN

The following MoN are compared in Figure 6.4 for a state deviation of $\alpha = 10^{-3}$:

- (1) the 5th-order TEMoN, $T_{A,5}$,
- (2) the static NLI from Eq. (6.4) computed from 10^5 random samples as in [50], and
- (3) the CMoN from Eq. (6.9) computed from 10^5 random samples. as in [73].

In Figure 6.4, each MoN is normalized by the maximum value of that MoN across all equilibrium points; thus, the maximum value of each normalized MoN is equal to one. The TEMoN and CMoN both indicate that L_3 is the most nonlinear equilibrium point, but the NLI indicates that L_2 is most nonlinear.

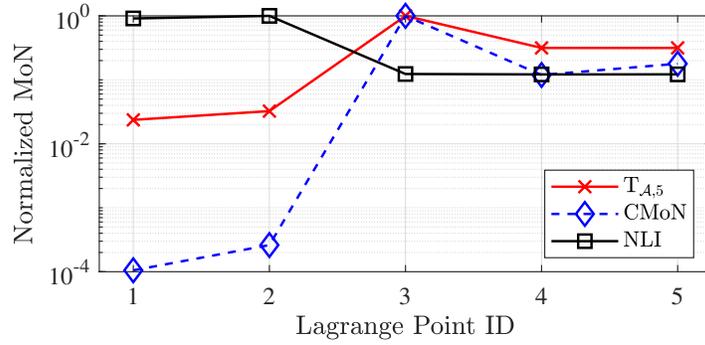


Figure 6.4: Comparing MoN that are normalized such that the largest value of each MoN equals one; considering a deviation of $\|\delta\mathbf{x}\| = 10^{-3}$ at each Lagrange point.

This example illustrates the significance of higher-order terms. The NLI is based solely on deviations in the linear term of the TSE, whereas the CMoN and TEMoN explicitly include higher-order terms and are more accurate. Discrepancies between the CMoN and the TEMoN can be attributed to two factors. First, the TEMoN includes higher-order terms that are not included in the CMoN. Second, the denominator of the CMoN is squared, whereas the denominator in the TEMoN is not.

6.5.3 Linear Region about Equilibrium Points

Finally, the deviation magnitude, $\alpha = \|\delta\mathbf{x}\|$, will be varied to demonstrate how the TEMoN may be used to predict the size of a linear region. The term “linear region” is used to describe the region about each equilibrium point wherein a linear approximation of the dynamics is sufficient. Figure 6.5 (left) shows τ_3^* through τ_5^* with a relatively large deviation magnitude of $\alpha = 10^{-1}$. Values of τ_k^* that are greater than one indicate that a higher-order term is dominating the linear term. In Figure 6.5 (left), it is clear that the deviation $\alpha = 10^{-1}$ is beyond the linear region for all equilibrium points. Figure 6.5 (right) shows τ_k^* with a much smaller deviation magnitude of $\alpha = 10^{-5}$. In this case, all τ_k^* are much less than one, indicating that the linear term is dominant and the deviation is within the linear region.

By extension, this method can also be used to justify the truncation order of a TSE when higher-order terms are included. All results indicate that L_3 is the most nonlinear equilibrium point, whereas L_1 is the most linear. However, as the order of the TEMoN is increased, τ_k for L_3 , L_4 and L_5 decay faster than τ_k

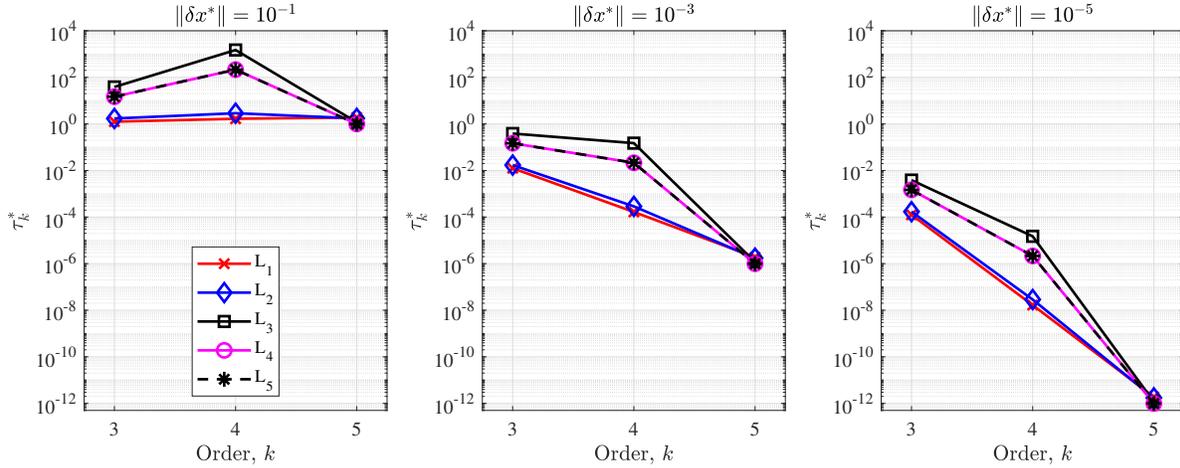


Figure 6.5: Computing the TEMoN for different deviation magnitudes; $\tau_k^* > 1$ indicates that a higher-order term is dominating the linear term and the deviation is outside the linear region.

for L_1 and L_2 . In the $\alpha = 10^{-3}$ example, the fifth-order terms for L_1 through L_5 have a similar order of magnitude and it may be sufficient to truncate all expansions at fourth order ($m = 4, p = 3$) despite large differences in nonlinearity. This example illustrates the utility of analyzing τ_k^* individually.

6.5.4 LDT Eigenvector Computation Time

Table 6.1 includes statistics for the eigenpairs computed in this section, including the percent of initial guesses that converged on an eigenvector, median number of RK4 iterations for convergence, and the median computation time in seconds. For each tensor, the DS method was applied in Julia with a maximum of 100 iterations, 6 maps, and 10 random initial guesses (60 total initializations). The code used in this example has not been optimized for speed, and these comparisons are only meant to demonstrate relative differences in computation time as the tensor order is increased.

Table 6.1: Equilibrium Point Eigenpair Statistics

| Order of Γ Tensor: | 5th-order | 6th-order | 7th-order |
|---|-----------|-----------|-----------|
| Percent converged | 25 | 32 | 21 |
| Median # RK4 iterations | 40 | 42 | 86 |
| Median computation time per eigenvector (seconds) | 0.0342 | 0.220 | 2.68 |

6.6 Application to STTs

In this section, the TEMoN is computed using the STTs of southern L_2 NRHOs in the Earth-Moon system. The NRHO dynamics are described by the CR3BP. NRHOs are known to experience rapid uncertainty growth and filter divergence at perilune, which other authors have attributed to strong nonlinearities [45,46].

6.6.1 NRHO Sampling Scheme

The TEMoN is computed at multiple sample points along each orbit, utilizing the STTs computed from the initial epoch (apolune) to each sample point. Sample points should be placed for good coverage along each orbit. In this example, sample points are evenly-spaced with respect to the parameter s :

$$s(t) = \frac{s_{radius}(t)}{s_{radius}(t_f)} + \frac{s_{speed}(t)}{s_{speed}(t_f)}, \quad (6.95)$$

where s_{radius} and s_{speed} are the arc lengths of the orbit radius and speed over time. This metric is ad hoc, and other metrics may also be appropriate (e.g., FTLE, tensor Frobenius norms, or acceleration magnitude, to name a few). The sampling scheme is illustrated in Figure 6.6 for the 9:2 lunar synodic resonant NRHO.

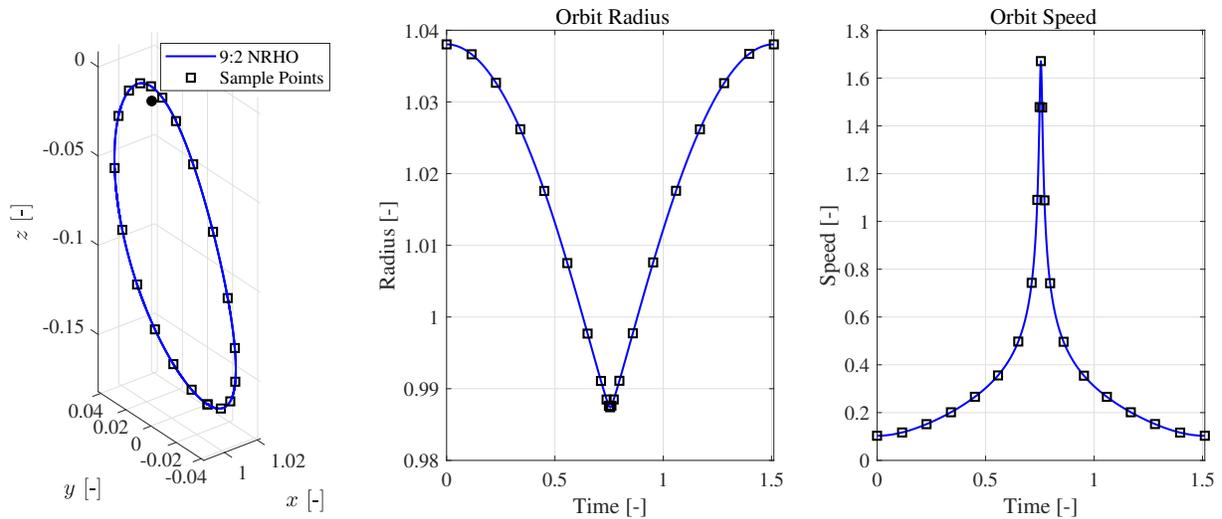


Figure 6.6: Sampling scheme along 9:2 NRHO based on the arc lengths of orbit radius and speed.

6.6.2 NRHO Nonlinearity

At each sample point, the TEMoN is formulated as follows:

$$T_{\Phi,5} = \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \left| \frac{\tilde{\Phi}^{(3)} \delta \mathbf{x}_0^3}{\tilde{\Phi}^{(2)} \delta \mathbf{x}_0^2} \right| + \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \left| \frac{\tilde{\Phi}^{(4)} \delta \mathbf{x}_0^4}{\tilde{\Phi}^{(2)} \delta \mathbf{x}_0^2} \right| + \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \left| \frac{\tilde{\Phi}^{(5)} \delta \mathbf{x}_0^5}{\tilde{\Phi}^{(2)} \delta \mathbf{x}_0^2} \right|. \quad (6.96)$$

$T_{\Phi,5}$ is an upper bound on

$$\aleph_{\Phi,5} \approx \max_{i=1,\dots,N} \left| \frac{\tilde{\Phi}^{(3)} \delta \mathbf{x}_{0,i}^3 + \tilde{\Phi}^{(4)} \delta \mathbf{x}_{0,i}^4 + \tilde{\Phi}^{(5)} \delta \mathbf{x}_{0,i}^5}{\tilde{\Phi}^{(2)} \delta \mathbf{x}_{0,i}^2} \right|. \quad (6.97)$$

Figure 6.7 shows $T_{\Phi,5}$, τ_3^* , τ_4^* , and τ_5^* for a family of NRHOs. The color gradient is interpolated linearly between sample points. In this example, the initial condition is at apolune, and the initial velocity is in the negative \hat{e}_y direction. The TEMoN detects a spike in nonlinearity at perilune, as expected. The color gradient is discontinuous at apolune (the initial/final state) because nonlinearity tends to increase over time with the growth of the STTs. This trend can be attributed to the compounding effect of nonlinearity in the dynamics of higher-order STTs, which causes the higher-order STTs to grow at a faster rate than the STM. The user can reinitialize the STTs and the TEMoN if necessary to maintain the sensitivity of the TEMoN to small changes in nonlinearity. The evolution of nonlinearity over the course of one orbit period can inform navigation and maneuver scheduling — if these events are sensitive to nonlinearity, they should be avoided at perilune whenever possible. Conversely, this sensitivity during periods of strong nonlinearity may be harnessed to enable large, inexpensive maneuvers.

6.6.3 STT TEMoN Validation

In Figure 6.8, the TEMoN ($T_{\Phi,5}$) is compared to the S-TEMoN ($S_{\Phi,5}$) and $\aleph_{\Phi,5}$ for one orbit of the 9:2 NRHO with an initial condition at apolune. The S-TEMoN and \aleph are computed with $N = 10^5$ random samples of \hat{z} which is then converted to $\delta \mathbf{x}_0$ via

$$\delta \mathbf{x}_0 = \frac{\alpha (\Phi^{(2)})^{-1} \hat{z}}{\left\| (\Phi^{(2)})^{-1} \hat{z} \right\|} \quad (6.98)$$

From Figure 6.8, the TEMoN is an upper bound on $\aleph_{\Phi,5}$, as expected. Large differences between $T_{\Phi,5}$ and $S_{\Phi,5}$ suggest that the random sampling method used to compute the S-TEMoN and \aleph is unreliable, and the TEMoN is a more robust measure of nonlinearity than these empirical measures.

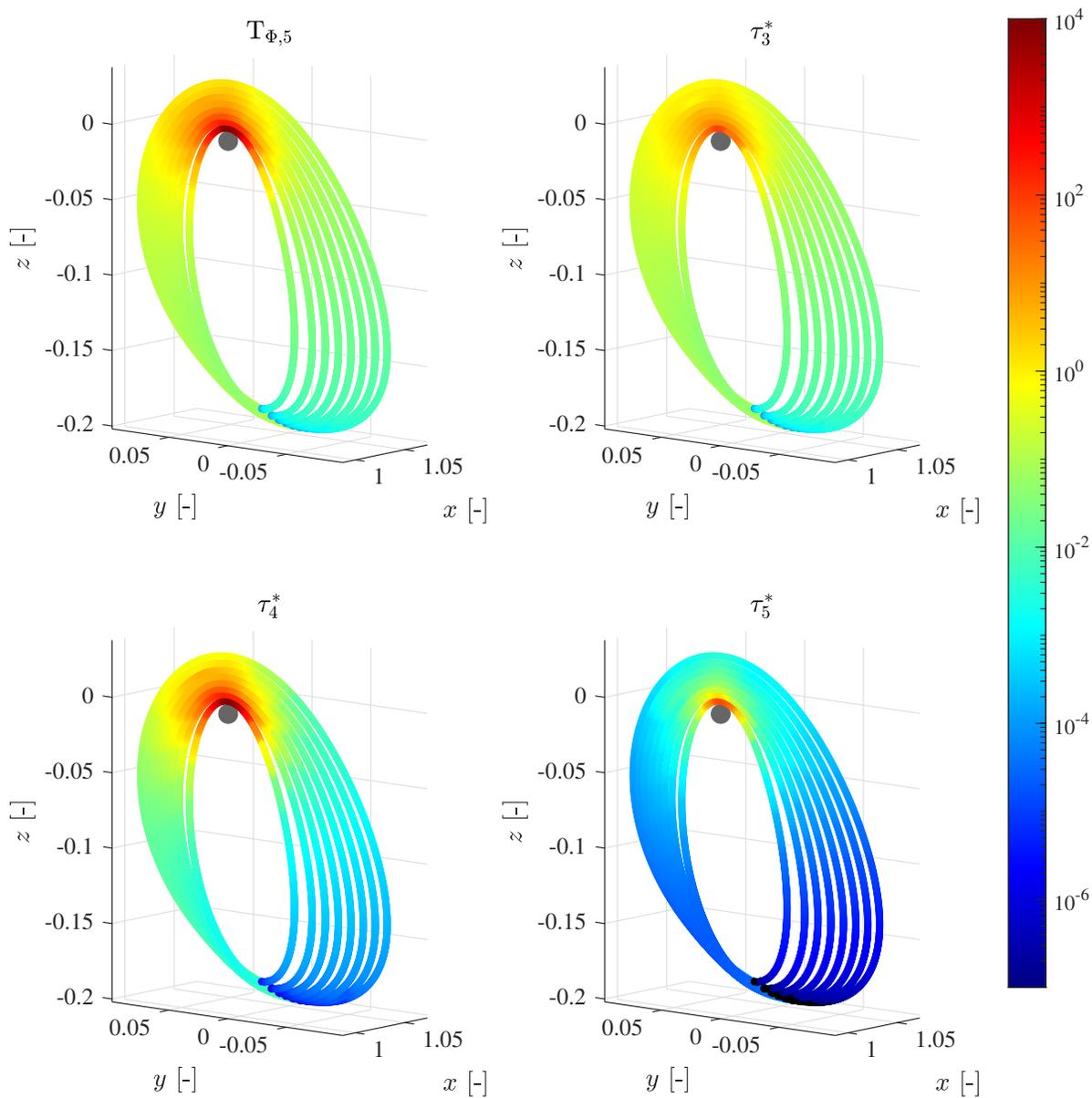


Figure 6.7: TEMoN and τ_k^* computed using the STTs along a family of NRHOs ($\|\delta\mathbf{x}_0\| = 10^{-3}$).

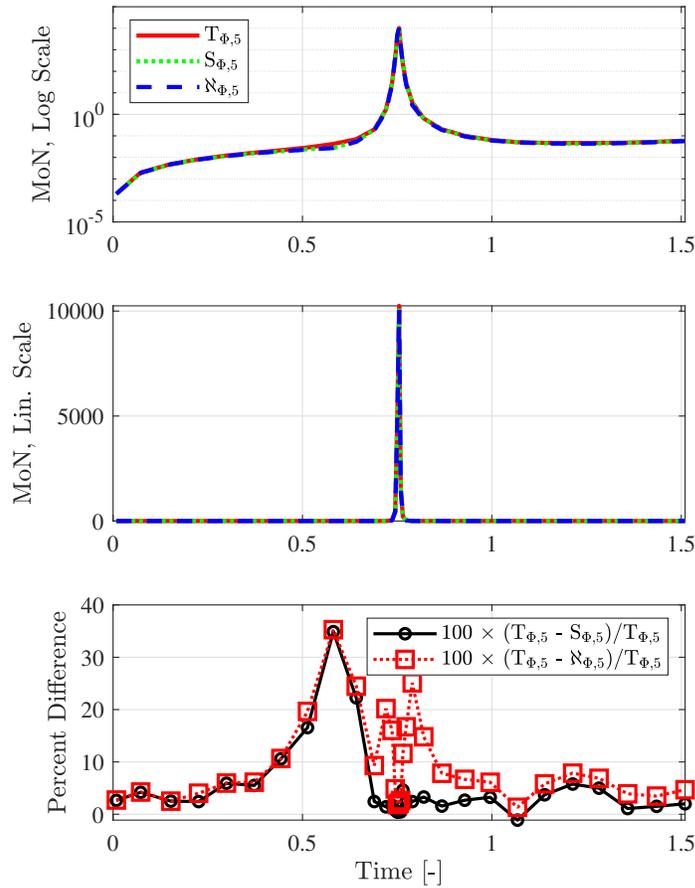


Figure 6.8: Comparing the TEMoN, S-TEMoN, and N for the 9:2 NRHO. The TEMoN is an upper bound on N . Large discrepancies between TEMoN and S-TEMoN indicate that empirical sampling is insufficient.

6.6.4 Linear Region about the 9:2 NRHO

The TEMoN may also be used to predict the size of the linear region about a nominal trajectory. If STTs are used during uncertainty propagation, guidance, or navigation, then the TEMoN can indicate the appropriate STT truncation order. Figure 6.9 shows the TEMoN computed along the 9:2 NRHO with different deviation magnitudes. A TEMoN greater than one indicates that higher-order terms are dominating the linear term. In the strongly nonlinear region at perilune, deviation magnitudes greater than 10^{-6} appear to be outside of the linear region. Conversely, the linear region is much larger in parts of the orbit that are less nonlinear.

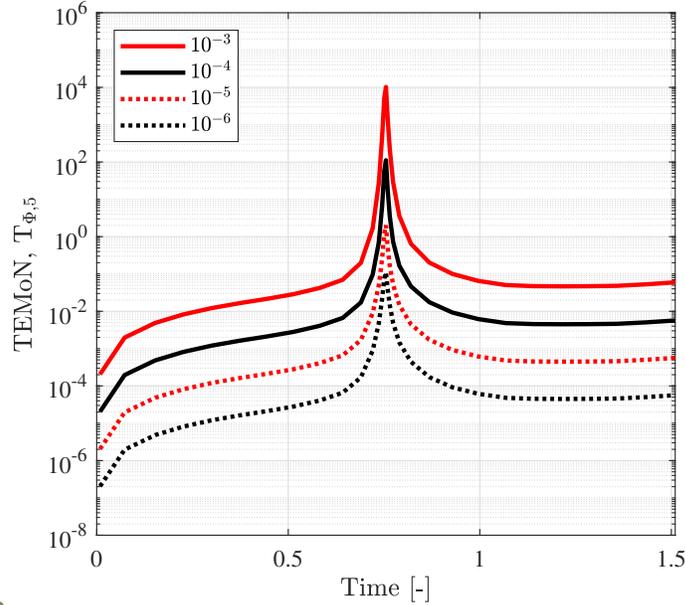


Figure 6.9: Computing the TEMoN for different deviation magnitudes about the 9:2 NRHO. The legend indicates the magnitude of $\delta\mathbf{x}(t_0)$.

6.6.5 NRHO Eigenvector Computation Time

Table 6.2 includes statistics for the eigenpairs computed in this section. The DS method was applied in Julia with 6 maps for each tensor, up to 15 initial guesses per map, and a maximum of 100 iterations per initial guess. As in the previous section, the algorithm has not been optimized for speed, and the comparisons in Table 6.2 are only meant to demonstrate relative differences in computation time.

Table 6.2: NRHO Eigenpair Statistics

| Order of Γ Tensor: | 5th-order | 6th-order | 7th-order |
|---|-----------|-----------|-----------|
| Percent converged | 48 | 49 | 50 |
| Median # RK4 iterations | 46 | 46 | 46 |
| Median computation time per eigenvector (seconds) | 0.0945 | 0.6125 | 3.70 |

6.6.6 NRHO LDTs

It was discussed in Section 6.6.2 that the STTs tend to grow at a faster rate than the STM, naturally causing nonlinearity to grow over time. Therefore, the user may want to reinitialize the STTs and the

TEMoN over long trajectories to preserve the sensitivity of the TEMoN to small changes in nonlinearity. Alternatively, the TEMoN can also be applied to the LDTs throughout a trajectory. When applied to the LDTs, the TEMoN serves as an instantaneous measure of nonlinearity in the dynamics with respect to the current state only, i.e., the time dependence of the STTs is lost. Figure 6.10 shows the 3rd-order TEMoN applied to the LDTs computed along the 9:2 NRHO. Unlike the STT TEMoN, the LDT TEMoN is only dependent on the current state and time.

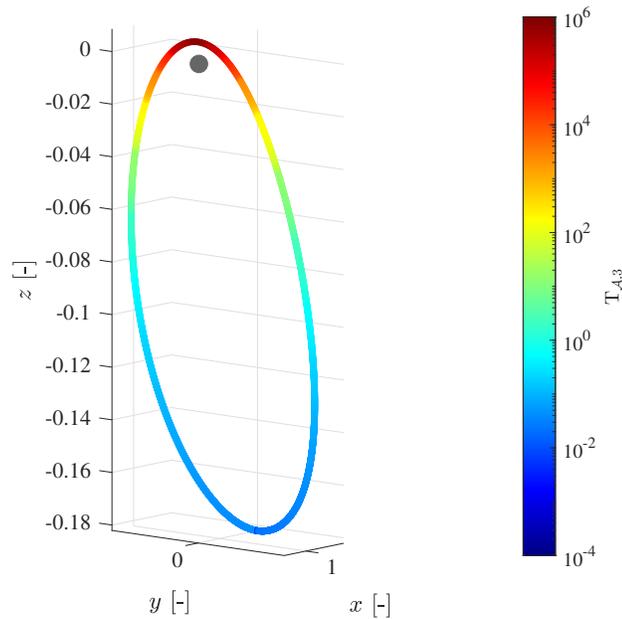


Figure 6.10: TEMoN applied to the LDTs for the 9:2 NRHO; when applied to the LDTs, the TEMoN only detects instantaneous nonlinearity in the dynamics.

6.7 Application to Parameter Transformations

In this section, the TEMoN is applied to the TPT process described in Section 6.2.3.2 for an example in the two-body problem: a spacecraft orbiting a small asteroid. An initial Cartesian position and velocity state will be converted to orbital elements for dynamics propagation and then converted back to a Cartesian state at the end of a single orbit. In Section 6.2.3.2 the TPT process was captured by the tensors $\Omega^{(k)}$. The TEMoN will be applied to $\Omega^{(3)}$ in this example and compared to (1) the TEMoN applied directly to the STTs associated with a dynamics propagation in Cartesian coordinates and (2) the TEMoN applied directly

to the STTs associated with a dynamics propagation in orbital elements. The Cartesian and orbital element STTs will be denoted by $\Phi^{(k)}$ and $\Psi^{(k)}$, respectively. The TEMoN for the TPT process, the Cartesian propagation, and the orbital elements propagation are given by the following equations.

$$\mathbf{T}_{\Omega,3} = \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \left| \frac{\tilde{\Omega}^{(3)} \delta \mathbf{x}_0^3}{\tilde{\Omega}^{(2)} \delta \mathbf{x}_0^2} \right|, \quad (6.99)$$

$$\mathbf{T}_{\Phi,3} = \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \left| \frac{\tilde{\Phi}^{(3)} \delta \mathbf{x}_0^3}{\tilde{\Phi}^{(2)} \delta \mathbf{x}_0^2} \right|, \quad (6.100)$$

$$\mathbf{T}_{\Psi,3} = \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \left| \frac{\tilde{\Psi}^{(3)} \delta \mathbf{x}_0^3}{\tilde{\Psi}^{(2)} \delta \mathbf{x}_0^2} \right|. \quad (6.101)$$

The set of orbital elements is denoted by ζ and includes the semimajor axis, eccentricity, inclination, argument of periapsis, right ascension of the ascending node, and true anomaly: $\zeta^\top = [a, e, i, \omega, \Omega, \theta]^\top$. Two orbit geometries are considered in this example with a nondimensional gravitational parameter of $\mu = 1$. Orbits A and B are shown in Figure 6.11 and their initial orbital elements are as follows: $\zeta_A^\top = [1, 10^{-2}, 10^{-2}, 10^{-2}, 10^{-2}, 10^{-2}]^\top$ and $\zeta_B^\top = [1, 0.5, \pi/2, \pi/4, \pi/3, \pi/6]^\top$.

TEMoN results are given in Table 6.3, as well as the optimal eigenpairs for the examples that utilize Cartesian coordinates. As expected, $\mathbf{T}_{\Psi,3}$ is much smaller than $\mathbf{T}_{\Phi,3}$ because the dynamics of the orbital elements are known to be significantly more linear than Cartesian coordinate dynamics. The results also confirm that nonlinearity is not avoided by the TPT process. Rather, $\mathbf{T}_{\Omega,3}$ and $\mathbf{T}_{\Phi,3}$ are very close in magnitude and $\Omega^{(3)}$ and $\Phi^{(3)}$ have very similar eigenvectors despite being different tensors. Thus, the TPT strategy and the Cartesian propagation have similar directions of strong nonlinearity.

Table 6.3: Different TEMoN types applied to orbits A and B.

| TEMoN Type | Orbit A | | | Orbit B | | |
|--|------------------------|--------|--|------------------------|-----------------------|--|
| | TEMoN | | $\lambda^*, \delta \hat{\mathbf{x}}^*$ | TEMoN | | $\lambda^*, \delta \hat{\mathbf{x}}^*$ |
| Cartesian STTs $T_{\Phi,3}$ | 0.05369 | 39.76, | $\begin{bmatrix} -0.03879 \\ 0.7024 \\ 7.028 \times 10^{-3} \\ -0.7096 \\ -0.03905 \\ -3.195 \times 10^{-4} \end{bmatrix}$ | 2.222 | $-8.192 \times 10^5,$ | $\begin{bmatrix} -0.2911 \\ -0.2911 \\ 0.06884 \\ 9.549 \times 10^{-4} \\ 9.548 \times 10^{-4} \\ -0.9087 \end{bmatrix}$ |
| With parameter transformation, $T_{\Omega,3}$ | 0.05406 | 34.05, | $\begin{bmatrix} -0.03857 \\ 0.7025 \\ 7.028 \times 10^{-3} \\ -0.7096 \\ -0.03927 \\ -3.216 \times 10^{-4} \end{bmatrix}$ | 2.201 | $-1.151 \times 10^6,$ | $\begin{bmatrix} 0.2911 \\ 0.2911 \\ -0.06887 \\ -9.657 \times 10^{-4} \\ -9.657 \times 10^{-4} \\ 0.9087 \end{bmatrix}$ |
| Orbital element STTs, $T_{\Psi,3}$ | 4.645×10^{-4} | | [-] | 1.016×10^{-3} | | [-] |

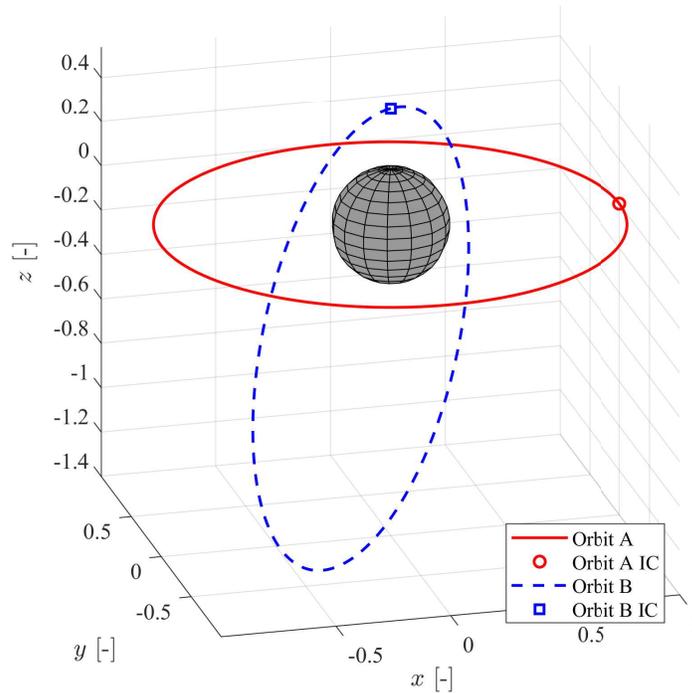


Figure 6.11: Orbits A and B and their initial conditions (IC).

6.8 Discussion

The examples in this chapter demonstrate that, not only is the TEMoN a semianalytical measure of nonlinearity, the TEMoN computation can significantly outperform Monte Carlo sampling in terms of

accuracy. Furthermore, the TEMoN may be computed in reasonable computation times, with the DS method converging on eigenvectors of 5th-order tensors in a matter of milliseconds. Tensor eigenpair algorithms are an ongoing area of research with many developments in recent years. Future work involves investigating ways in which the DS method (or other methods) may be better leveraged to compute the TEMoN. This includes determining an optimal numerical integration scheme and a suitable set of mapping functions to limit the number of initial guesses and the computation time needed to compute the TEMoN.

Tensor eigenpairs may also be used to study other characteristics of dynamical systems. Specifically, the $\tilde{\mathcal{T}}^{(2)}$ tensor in Eq. (6.36) can be related to the Cauchy-Green tensor (CGT) and the finite-time Lyapunov exponent (FTLE). The CGT, also referred to as the Cauchy strain tensor or the Cauchy-Green deformation tensor, originated in the field of continuum mechanics [85, 86]. During the deformation and displacement of a continuum body, a deformation gradient tensor may be defined to describe local changes in the neighborhood of a particular point. This deformation gradient tensor is functionally equivalent to the STM of a dynamical system and will therefore be denoted by $\Phi(t, t_0)$. The CGT is then defined as

$$\tilde{\Phi}^{(2)} = \Phi(t, t_0)^\top \Phi(t, t_0). \quad (6.102)$$

It is well-known that the state deviation $\delta \mathbf{x}(t_0)$ that maximizes $\|\delta \mathbf{x}(t)\|$ is an eigenvector of $\tilde{\Phi}^{(2)}$ and the maximum stretching of $\delta \mathbf{x}(t_0)$ is determined from the maximum eigenvalue of $\tilde{\Phi}^{(2)}$ [87]:

$$\delta \mathbf{x}_0^* = \arg \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \|\tilde{\Phi}^{(2)} \delta \mathbf{x}_0\| \quad (6.103)$$

$$= \arg \max_{\delta \mathbf{x}_0 \in \mathcal{X}} \tilde{\Phi}^{(2)} \delta \mathbf{x}_0^2 \quad (6.104)$$

$$\tilde{\Phi}^{(2)} \delta \mathbf{x}_0^* = \lambda_{max} \delta \mathbf{x}_0^* \quad (6.105)$$

Furthermore, the maximum eigenvalue (λ_{max}) can be related to the finite-time Lyapunov exponent (FTLE) [86] to quantify stretching over some time interval:

$$\text{FTLE} = \frac{1}{|t - t_0|} \log \sqrt{\lambda_{max}}. \quad (6.106)$$

Because the STM is a linear approximation, the FTLE is also an approximation. Consider that the higher-order tensors $\tilde{\Phi}^{(m)}$ for $m > 2$ (see Eqs. (6.35) and (6.36)) are higher-order analogs of the CGT and the

eigenpairs of these higher-order tensors may be used to more accurately quantify stretching in the phase space. A similar procedure to Section 6.4 may be used to compute the maximum *nonlinear* stretching over some time interval (in this case, normalization by $\tilde{\Phi}^{(2)}v^2$ should be omitted).

6.9 Conclusions

Previously developed measures of nonlinearity are heuristic or their computation relies on empirical sampling or numerical optimization. Such methods are often time consuming and difficult to validate. This chapter proposes a novel, semianalytical measure of nonlinearity based on tensor eigenpairs — the TEMoN. The tensor in question is derived from the TSE. Tensor eigenpairs are used to quantify model nonlinearity, as well as to determine directions of strong nonlinearity. Moreover, the TEMoN can be used to predict the size of the linear region about a reference trajectory or to justify the truncation order of a TSE. The measure may be applied to static or dynamic models, and has been demonstrated for the LDTs at equilibrium points in the CR3BP and the STTs along NRHOs. Moreover, the TEMoN has been used to analyze the nonlinearity of a transformation-propagation-transformation strategy in which Cartesian coordinates are transformed into orbital elements for the purpose of dynamics propagation, and then transformed back into Cartesian coordinates at the final time. The results confirm that the degree of nonlinearity and the directions of strong nonlinearity are comparable for both processes. In summary, the TEMoN may serve as a mission design tool with numerous applications in control, estimation, and uncertainty propagation.

Chapter 7

Conclusions

7.1 Summary

This dissertation focused primarily on the stochastic optimization of spacecraft trajectories and correction maneuvers with the goal of minimizing spacecraft state errors. Control-linear noise was considered throughout, which is a specific case of control-dependent noise where control errors are linearly proportional to the level of nominal control. This configuration is often ignored during maneuver optimization due to its increased complexity. However, experimental studies of spacecraft electric propulsion engines indicate that control-linear noise can be significant, and control-linear noise is also used in the standard models for chemical propulsion. Previous work in uncertainty minimization for spacecraft trajectories is also limited.

This dissertation began with a nonlinear, open-loop, continuous-thrust spacecraft trajectory optimization to minimize terminal state error covariance and control energy in the presence of control-linear noise. A linear covariance propagation was assumed, and a multi-objective optimization based on the ϵ -constraint method was performed using indirect methods. Although the problem setup was general, results were presented for orbit transfers, phasing maneuvers, and proactive station-keeping maneuvers along asteroid Sun-terminator orbits. The results indicate that significant covariance reduction is possible with slight increases in propellant cost. The proactive station-keeping scenario was presented as an alternative to reactive station keeping: rather than utilizing trajectory correction maneuvers to correct errors as they appear along a ballistic trajectory, the proactive station-keeping scenario applies open-loop control in a way that minimizes state uncertainty without the need for feedback control.

Robust closed-loop control was also investigated in this dissertation, beginning with the analytical

optimization of a control law for linear systems with control-linear noise, sampled perfect measurements, and continuous control using dynamic programming. The sampled measurement formulation corresponds to a special case of intermittent control, wherein the controller receives state measurements at limited times and the control can be considered open-loop between the measurement times. This setup is consistent with spacecraft control, where navigation cannot be prioritized at all times and state knowledge is limited. The method was then extended to a more general, hybrid dynamics model with impulsive control, additive noise, and multiple sources of control-linear noise. The robust control law was applied as a neighboring guidance law in various mission scenarios by linearizing spacecraft dynamics about nominal trajectories. The guidance law was then compared extensively to a deterministic control law to demonstrate the benefits of robust control. The derivation of the guidance law relied on a perfect measurement assumption. However, a bounded cost function approach was developed to analytically characterize control performance when the perfect measurement assumption is relaxed.

Finally, a novel nonlinearity measure was proposed: the tensor eigenpair measure of nonlinearity (TEMoN). The TEMoN is based on the eigenpairs of higher-order tensors in a Taylor series expansion, and may be applied to static or dynamic models. Some examples include application to the local dynamics tensors, state transition tensors, and parameter transformations. Unlike other MoN, the TEMoN is semianalytical and does not require empirical sampling, approximation, or numerical optimization. Nonlinearity measures like the TEMoN can be used to identify regions of strong nonlinearity and quantify the size of a linear region, with many applications in guidance, navigation, and control.

7.2 Future Work

This dissertation opens numerous avenues for future work. Many extensions can be made to the multi-objective nonlinear trajectory optimization performed in Chapter 2. First, additive noise may be included in the covariance propagation without changing the methodology presented here. Second, uncertain static parameters may be considered by augmenting the state vector with these parameters so that their initial uncertainty may influence the covariance propagation. Other potential modifications include constraining the magnitude of the optimal control or using the method here as an initial guess for a propellant-and-covariance

minimization (as opposed to energy-and-covariance). The accuracy of the covariance propagation may also be improved by including higher-order terms. Finally, implementing this optimization in a more advanced multi-objective optimization framework based on evolutionary algorithms is a natural next step, and the results here may serve as validation for more sophisticated numerical optimization algorithms.

Many interesting problems arise in the area of closed-loop control, such as the problem of optimal measurement and control scheduling. In this dissertation, measurement and control times were assumed to be fixed; however, additional uncertainty reduction is certainly possible by optimizing the timing of these events. Likewise, the problem could be modified to include uncertainty in the measurement and control times. Additionally, analysis may be performed to evaluate the robustness in the event that measurements or controls are delayed or missed entirely. Finally, the problem of imperfect navigation creates many opportunities for future work. For example, iterative backward-forward sweep methods could be generated to account for noisy measurements and Kalman filter updates, and the optimality of such methods may be evaluated using the bounded cost function approached proposed in this dissertation.

In the area of nonlinearity measures, the TEMoN algorithm can likely be improved by optimizing the eigenpair computation and investigating alternative eigenpair algorithms. The method may also be adapted to quantify nonlinear stretching in a similar manner to the Cauchy-Greene tensor, i.e., by generating “higher-order Cauchy-Greene tensors.” The eigenpairs of these higher-order tensors may be used to generate a higher-order finite-time Lyapunov exponent.

Bibliography

- [1] D. Nicolini, P. Frigot, F. Musso, S. Cesare, G. Castorina, L. Ceruti, F. Bartola, P. Zanella, F. Ceccanti, L. Priami, and L. Paita, "Direct thrust and thrust noise measurements on the LISA pathfinder field emission thruster," in Proc. 31st International Electric Propulsion Conference, (Ann Arbor, Michigan, USA), Sept. 2009.
- [2] M. S. McDonald, M. J. Sekerak, A. D. Gallimore, and R. R. Hofer, "Plasma oscillation effects on nested hall thruster operation and stability," in Proc. 2013 IEEE Aerospace Conference, (Big Sky, MT, USA), Mar. 2013.
- [3] C. R. Gates, "Technical Report No. 32-504: A Simplified Model of Midcourse Maneuver Execution Errors," tech. rep., Jet Propulsion Laboratory, Oct. 1963.
- [4] S. V. Wagner and T. D. Goodson, "Execution-error modeling and analysis of the cassini-huygens spacecraft through 2007," in Proc. AAS/AIAA Space Flight Mechanics Meeting, (Galveston, TX, U.S.A), Jan. 2008.
- [5] T. D. Goodson, "Execution-error modeling and analysis of the GRAIL spacecraft pair," in Proc. 23rd AAS/AIAA Spaceflight Mechanics Meeting, (Kauai, HI, U.S.A), Feb. 2013.
- [6] E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system," Neural Comput., vol. 17, no. 5, pp. 1084–1108, 2005.
- [7] B. Williams, P. Antreasian, E. Carranza, C. Jackman, J. Leonard, D. Nelson, B. Page, D. Stanbridge, D. Wibben, K. Williams, M. Moreau, K. Berry, K. Getzandanner, A. Liounis, A. Mashiku, D. Highsmith, B. Sutter, and D. Lauretta, "Osiris-rex flight dynamics and navigation design," Space Science Reviews, vol. 214, no. 69, pp. 1–43, 2018.
- [8] S. G. Hesar, D. J. Scheeres, and J. W. McMahon, "Sensitivity analysis of the osiris-rex terminator orbits to maneuver errors," Journal of Guidance, Control, and Dynamics, vol. 40, no. 1, pp. 81–05, 2017.
- [9] Y. Ren and J. Shan, "Reliability-based soft landing trajectory optimization near asteroid with uncertain gravitational field," Journal of Guidance, Control, and Dynamics, vol. 38, no. 9, pp. 1810–1820, 2015.
- [10] K. Oguri and J. W. McMahon, "Robust spacecraft guidance around small bodies under uncertainty: Stochastic optimal control approach," Journal of Guidance, Control, and Dynamics, 2021.
- [11] I. M. Ross, R. J. Proulx, and M. Karpenko, "Unscented optimal control for orbital and proximity operations in an uncertain environment: A new zermelo problem," in Proc. AIAA/AAS Astrodynamics Specialist Conference, (San Diego, CA, U.S.A), Aug. 2014.

- [12] S. Zimmer, Reducing Spacecraft State Uncertainty Through Indirect Trajectory Optimization. PhD thesis, Univ. of Texas, Austin, TX, U.S.A, Dec. 2005.
- [13] H. Hu, S. Zhu, and P. Cui, “Desensitized optimal trajectory for landing on small bodies with reduced landing error,” Aerospace Science and Technology, vol. 48, pp. 178–185, 2015.
- [14] J. Gergaud and T. Haberkorn, “Homotopy method for minimum consumption orbit transfer problem,” ESAIM: Control, Optimisation and Calculus of Variations, vol. 12, no. 2, pp. 294–310, 2006.
- [15] J. H. Cho, Y. Wang, R. Chen, K. S. Chan, and A. Swami, “A survey on modeling and optimizing multi-objective systems,” IEEE Communications Surveys and Tutorials, vol. 19, no. 3, pp. 1867–1900, 2017.
- [16] G. Mavrotas, “Effective implementation of the epsilon-constraint method in multi objective mathematical programming problems,” Applied Mathematics and Computation, vol. 213, no. 2, pp. 455–465, 2009.
- [17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.
- [18] V. Coverstone-Carroll, J. Hartmann, and W. Mason, “Optimal multi-objective low-thrust spacecraft trajectories,” Computer Methods in Applied Mechanics and Engineering, vol. 186, no. 2-4, pp. 387–407, 2000.
- [19] J. A. Englander, M. A. Vavrina, and A. R. Ghosh, “Multi-objective hybrid optimal control for multiple-flyby low-thrust mission design,” in Proc. of 25th AAS/AIAA Space FlightMechanics Meeting, (Williamsburg, VA, U.S.A), Jan. 2015.
- [20] M. Vasile and L. Ricciardi, “Multi agent collaborative search,” in Studies in Computational Intelligence (O. Schütze, L. Trujillo, P. Legrand, and Y. Maldonado, eds.), pp. 223–252, Springer, Cham., 2016.
- [21] S. Takahashi and D. J. Scheeres, “Higher order corrections for frozen terminator orbit design,” Journal of Guidance, Control, and Dynamics, vol. 43, no. 9, pp. 1642–1655, 2020.
- [22] D. R. Wibben, A. Levine, S. Rieger, J. V. McAdams, P. G. Antreasian, J. M. Leonard, M. C. Moreau, , and D. S. Lauretta, “Osiris-rex frozen orbit design and flight experience,” in Proc. of 2019 AAS/AIAA Astrodynamics Specialist Conference, (Portland, Maine, USA), Aug. 2019.
- [23] A. E. Bryson and Y. Ho, Applied Optimal Control. New York, NY, U.S.A: Taylor and Francis Group, 1975.
- [24] C. L. Ranieri and C. A. Ocampo, “Optimization of roundtrip, time-constrained, finite burn trajectories via an indirect method,” Journal of Guidance, Control, and Dynamics, vol. 28, no. 2, pp. 306–314, 2005.
- [25] J.-B. Caillau, B. Daoud, and J. Gergaud, “Minimum fuel control of the planar circular restricted three-body problem,” Celestial Mechanics and Dynamical Astronomy, vol. 114, pp. 137–150, 2012.
- [26] P. Baldi, Stochastic Calculus: An Introduction Through Theory and Exercises, ch. 8, pp. 219–220. Springer, Cham, 2017.

- [27] D. H. Ellison, B. A. Conway, J. A. Englander, and M. T. Ozimek, "Analytic gradient computation for bounded-impulse trajectory models using two-sided shooting," Journal of Guidance, Control, and Dynamics, vol. 41, no. 7, pp. 1449–1462, 2018.
- [28] Y. Meng, H. Zhang, and Y. Gao, "Low-thrust minimum-fuel trajectory optimization using multiple shooting augmented by analytical derivatives," Journal of Guidance, Control, and Dynamics, vol. 42, no. 3, pp. 662–677, 2019.
- [29] G. Lantoine, R. P. Russell, and T. Dargent, "Using multicomplex variables for automatic computation of high-order derivatives," ACM Transactions on Mathematical Software, vol. 38, no. 3, 2012.
- [30] J. W. McMahon, D. J. Scheeres, S. G. Hesar, D. Farnocchia, S. Chelsey, and D. Lauretta, "The osiris-rex radio science experiment at bennu," Space Science Reviews, vol. 214, no. 43, pp. 1–41, 2018.
- [31] D. J. Scheeres, J. W. McMahon, A. S. French, D. N. Brack, S. R. Chesley, D. Farnocchia, Y. Takahashi, J. M. Leonard, J. Geeraert, B. Page, P. Antreasian, K. Getzandanner, D. Rowlands, E. M. Mazarico, J. Small, D. E. Highsmith, M. Moreau, J. P. Emery, B. Rozitis, M. Hirabayashi, P. Sánchez, S. V. wal, P. Tricarico, R.-L. Ballouz, C. L. Johnson, M. M. A. Asad, H. C. M. Susorney, O. S. Barnouin, M. G. Daly, J. A. Seabrook, R. W. Gaskell, E. E. Palmer, J. R. Weirich, K. J. Walsh, E. R. Jawin, E. B. Bierhaus, P. Michel, W. F. Bottke, M. C. Nolan, H. C. C. Jr, D. S. Lauretta, and The OSIRIS-REx Team, "The dynamic geophysical environment of (101955) bennu based on osiris-rex measurements," Nature Astronomy, vol. 3, p. 352–361, 2019.
- [32] M. Pachter and M. M. Briggs, "Simulation of white noise in dynamical systems," Simulation, vol. 56, no. 1, pp. 19–25, 1991.
- [33] D. L. Kleinman, "Optimal stationary control of linear systems with control-dependent noise," IEEE Trans. Autom. Control, vol. AC-14, pp. 673–677, Dec. 1969.
- [34] P. J. McLane, "Linear optimal control of a linear system with state and control dependent noise," J. Dyn. Sys., Meas., Control, vol. 91, pp. 34–40, 1972.
- [35] P. J. McLane, "Optimal stochastic control of linear systems with state- and control-dependent disturbances," IEEE Trans. Autom. Control, vol. AC-16, pp. 793–798, Dec. 1971.
- [36] J. B. Moore, X. Y. Zhou, and A. E. B. Lim, "Discrete time LQG controls with control dependent noise," Syst. Control Lett., vol. 36, no. 3, pp. 199–206, 1999.
- [37] E. D. Gustafson, Stochastic Optimal Control of Spacecraft. PhD thesis, Univ. of Michigan, Ann Arbor, MI, U.S.A, 2010.
- [38] K. Fujimoto, S. Ogawa, Y. Ota, and M. Nakayama, "Optimal control of linear systems with stochastic parameters for variance suppression: The finite time horizon case," IFAC Proc. Vol., vol. 44, no. 1, pp. 12605–12610, 2011.
- [39] A. Bratus, I. Yegorov, and D. Yurchenko, "Optimal bounded noisy feedback control for damping random vibrations," J. Vib. Control, vol. 24, no. 10, pp. 1874–1888, 2018.
- [40] Q. Zhu and H. Wang, "Output feedback stabilization of stochastic feedforward systems with unknown control coefficients and unknown output function," Automatica, vol. 87, pp. 166–175, 2018.

- [41] I. A. Digailova and A. B. Kurzhanski, "Reachability analysis under control-dependent stochastic noise," IFAC Proc. Vol., vol. 38, no. 1, pp. 1209–1214, 2005.
- [42] I. A. Digailova, A. B. Kurzhanski, and P. Varaiya, "Stochastic reachability and measurement feedback under control-dependent noise," IFAC Proc. Vol., vol. 41, no. 2, pp. 14336–14341, 2008.
- [43] T. Yang, Impulsive Control Theory. Springer, 2001.
- [44] B. D. Tapley, B. E. Schutz, and G. H. Born, Statistical Orbit Determination, ch. 2, p. 68. Burlington, MA, USA: Elsevier Academic Press, 2004.
- [45] N. L. Parrish, M. J. Bolliger, E. Kayser, M. R. Thompson, J. S. Parker, B. W. Cheetham, D. C. Davis, and D. J. Sweeney, "Near rectilinear halo orbit determination with simulated dsn observations," in Proc. AIAA SciTech Forum, (Orlando, FL, USA), pp. 1–19, Jan. 2020.
- [46] D. Davis, S. Bhatt, K. Howell, J.-W. Jang, R. Whitley, F. Clark, D. Guzzetti, E. Zimovan, and G. Barton, "Orbital maintenance and navigation of human spacecraft at cislunar near rectilinear halo orbits," in Proc. 27th AAS/AIAA Spaceflight Mechanics Meeting, (San Antonio, TX, USA), pp. 1–20, Feb. 2017.
- [47] E. L. Jenson and D. J. Scheeres, "Semianalytical measures of nonlinearity based on tensor eigenpairs," in Proc. AAS/AIAA Astrodynamics Specialist Conference, (virtual), Aug. 2021.
- [48] S. Takahashi and D. J. Scheeres, "Autonomous exploration of a small near-earth asteroid," Journal of Guidance, Control, and Dynamics, vol. 44, no. 4, pp. 701–718, 2021.
- [49] K. Berry, K. Getzandanner, M. Moreau, P. Antreasian, A. Polit, M. Nolan, H. Enos, and D. Lauretta, "Revisiting OSIRIS-REx touch-and-go (TAG) performance given the realities of asteroid bennu," in AAS Guidance, Navigation, and Control Conference, (San Diego, CA, USA), 2020.
- [50] J. L. Junkins and P. Singla, "How nonlinear is it? a tutorial on nonlinearity of orbit and attitude dynamics," The Journal of the Astronautical Sciences, vol. 52, no. 1 and 2, pp. 7–60, 2004.
- [51] K. Vishwajeet and P. Singla, "Adaptive split/merge-based gaussian mixture model approach for uncertainty propagation," Journal of Guidance, Control, and Dynamics, vol. 41, no. 3, pp. 603–617, 2018.
- [52] J. Havlík and O. Straka, "Measures of nonlinearity and non-gaussianity in orbital uncertainty propagation," in Proc. of 22nd International Conference on Information Fusion, (Ottawa, Canada), pp. 1 – 8, 2019.
- [53] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," Journal of Guidance, Control, and Dynamics, vol. 36, no. 4, pp. 1047–1057, 2013.
- [54] Y. Chen, D. Cuccato, M. Bruschetta, and A. Beghi, "An inexact sensitivity updating scheme for fast nonlinear model predictive control based on a curvature-like measure of nonlinearity," in Proc. of IEEE 56th Conference on Decision and Control, (Melbourne, VIC, Australia), pp. 4382–4387, 2017.
- [55] A. Omran and B. Newman, "Nonlinearity index theory for aircraft dynamic assessment," Journal of Guidance, Control, and Dynamics, vol. 36, no. 1, pp. 293–303, 2013.
- [56] P. Cui, S. Wang, A. Gao, and Z. Yu, "X-ray pulsars/doppler integrated navigation for mars final approach," Advances in Space Research, vol. 57, pp. 1889–1900, 2016.

- [57] S. K. Biswas, L. Qiao, and A. G. Dempster, "A quantified approach of predicting suitability of using the unscented kalman filter in a non-linear application," *Automatica*, vol. 122, pp. 1–12, 2020.
- [58] T. Sun and M. Xin, "Hypersonic entry vehicle state estimation using nonlinearity-based adaptive curvature kalman filters," *Acta Astronautica*, vol. 134, pp. 221–230, 2017.
- [59] P. Wang, E. Blasch, X. R. Li, E. Jones, R. Hanak, W. Yin, A. Beach, , and P. Brewer, "Degree of nonlinearity (don) measure for target tracking in videos," in *Proc. of 19th International Conference on Information Fusion*, (Heidelberg, Germany), pp. 1 – 8, 2016.
- [60] M. Mallick and B. Ristic, "Comparison of measures of nonlinearity for bearing-only and gmti filtering," in *Proc. of 20th International Conference on Information Fusion*, (Xi'an, China), pp. 1 – 8, 2017.
- [61] R. Niu, P. K. Varshney, M. Alford, A. Bubalo, E. Jones, and M. Scalzo, "Curvature nonlinearity measure and filter divergence detector for nonlinear tracking problems," in *Proc. of 11th International Conference on Information Fusion*, (Cologne, Germany), pp. 1–8, 2008.
- [62] R. S. Park and D. J. Scheeres, "Nonlinear mapping of gaussian statistics: Theory and applications to spacecraft trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1367–1375, 2006.
- [63] S. M. Hosseini, T. A. Johansen, and A. Fatehi, "Comparison of nonlinearity measures based on time series analysis for nonlinearity detection," *Modeling, Identification and Control*, vol. 32, no. 4, pp. 123–130, 2011.
- [64] E. M. L. Beale, "Confidence regions in non-linear estimation," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 22, no. 1, pp. 41–76, 1960.
- [65] I. Guttman and D. A. Meeter, "On beale's measures of non-linearity," *Technometrics*, vol. 7, no. 4, pp. 623–637, 1965.
- [66] C. A. Desoer and Y.-T. Wang, "Foundations of feedback theory for nonlinear dynamical systems," *IEEE Transactions on Circuits and Systems*, vol. CAS-27, no. 2, pp. 104–123, 1980.
- [67] A. Helbig, W. Marquardt, and F. Allgöwer, "Nonlinearity measures: definition, computation and applications," *Journal of Process Control*, vol. 10, no. 2-3, pp. 113–123, 2000.
- [68] Y. Liu and X. R. Li, "Measure of nonlinearity for estimation," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2377 – 2388, 2015.
- [69] W. Tan, H. J. Marquez, T. Chen, and J. Liu, "Analysis and control of a nonlinear boiler-turbine unit," *Journal of Process Control*, vol. 15, no. 8, pp. 883–891, 2005.
- [70] G. T. Tan, M. Huzmezan, and K. E. Kwok, "On measuring closed-loop nonlinearity - a vinnicombe metric approach," in *42nd IEEE Conference on Decision and Control*, (Maui, Hawaii, USA), pp. 6163–6168, 2003.
- [71] J. Du, C. Song, , and P. Li, "A gap metric based nonlinearity measure for chemical processes," in *American Control Conference*, (St. Louis, MO, USA), pp. 4440–4445, 2009.
- [72] J. Hahn and T. F. Edgar, "A gramian based approach to nonlinearity quantification and model classification," *Industrial & Engineering Chemistry Research*, vol. 40, no. 24, pp. 5724–5731, 2001.

- [73] D. M. Bates and D. G. Watts, “Relative curvature measures of nonlinearity,” Journal of the Royal Statistical Society Series B (Methodological), vol. 42, no. 1, pp. 1–25, 1980.
- [74] B. W. Bader, T. G. Kolda, et al., “Tensor toolbox for matlab, version 3.2.1,” 2021.
- [75] T. G. Kolda and J. R. Mayo, “An adaptive shifted power method for computing generalized tensor eigenpairs,” SIAM J. Matrix Analysis and Applications, vol. 35, pp. 1563–1582, 2014.
- [76] L.-H. Lim, “Singular values and eigenvalues of tensors: a variational approach,” in Proc. of 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, (Puerto Vallarta, Mexico), p. 129–132, 2005.
- [77] L. Qi, “Eigenvalues of a real supersymmetric tensor,” Journal of Symbolic Computation, vol. 40, no. 6, pp. 1302–1324, 2005.
- [78] L. Chen, L. Han, and L. Zhou, “Computing tensor eigenvalues via homotopy methods,” SIAM Journal on Matrix Analysis and Applications, vol. 37, no. 1, p. 290–319, 2016.
- [79] K.C.Chang, K. Pearson, and T. Zhang, “On eigenvalue problems of real symmetric tensors,” Journal of Mathematical Analysis and Applications, vol. 350, no. 1, pp. 416–422, 2009.
- [80] A. R. Benson and D. F. Gleich, “Computing tensor z-eigenvectors with dynamical systems,” SIAM J. Matrix Anal. Appl., vol. 40, no. 4, pp. 1311–1324, 2019.
- [81] A. R. Benson, “Tze-dynsys github repository,” 2020.
- [82] T. G. Kolda and J. R. Mayo, “Shifted power method for computed tensor eigenpairs,” SIAM J. Matrix Analysis and Applications, vol. 32, p. 1095–1124, 2011.
- [83] C.-F. Cui, Y.-H. Dai, and J. Nie, “All real eigenvalues of symmetric tensors,” SIAM Journal on Matrix Analysis and Applications, vol. 35, no. 4, p. 1582–1601, 2014.
- [84] D. Cartwright and B. Sturmfels, “The number of eigenvalues of a tensor,” Linear Algebra and its Applications, vol. 435, pp. 942–952, 2013.
- [85] D. R. Smith, An Introduction to Continuum Mechanics — after Truesdell and Noll, ch. 3, p. 111. Springer Science and Business Media Dordrech, 1993.
- [86] C. R. Short, D. Blazevski, K. C. Howell, and G. Haller, “Stretching in phase space and applications in general nonautonomous multi-body problems,” Celestial Mechanics and Dynamical Astronomy, vol. 122, p. 213–238, 2015.
- [87] D. J. Scheeres, “Navigation of spacecraft in unstable orbital environments,” in Proc. of Libration Point Orbits and Applications, (Aiguablava, Spain), pp. 399–438, June 2003.
- [88] K. Howell and J. Breakwell, “Almost rectilinear halo orbits,” in Proc. 20th Aerospace Sciences Meeting, 1982.

Appendix A

Stochastic Dynamics and Itô Calculus

The Itô form of a stochastic differential equation (SDE) is given by

$$d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, \mathbf{u}(t), t)dt + \mathbf{h}(\mathbf{X}_t, \mathbf{u}(t), t)d\mathbf{W}_t, \quad (\text{A.1})$$

in which $\mathbf{X}_t \in \mathbb{R}^n$ is the state of the stochastic system, $\mathbf{u}(t) \in \mathbb{R}^m$ is the nominal control input, and $W_t \in \mathbb{R}^q$ is a standard Wiener process [26]. The Wiener process, also known as a Brownian motion process, is continuous in time with an initial condition of zero and independent, Gaussian increments, i.e., $(W_{t+s} - W_s) \sim \mathcal{N}(0, tI)$. The Wiener process is often characterized informally as the integral of white, Gaussian noise such that $d\mathbf{W}_t/dt$ represents a white noise process.

Consider the following Itô SDE, which appears in Chapters 2 and 3.

$$d\mathbf{X}_t = \mathbf{f}(\mathbf{X}_t, t)dt + B\mathbf{u}(t)(dt + \sigma dW_t) \quad (\text{A.2})$$

In the above expression, the noise process is scaled linearly by the control. Thus, the model represents control-linear noise. Next, consider the deviation of the stochastic state from a deterministic state, $\mathbf{X}'(t)$:

$$\mathbf{x}_t = \mathbf{X}_t - \mathbf{X}'(t) \quad (\text{A.3})$$

First, linearized dynamics of Eq. (A.2) are derived from a first-order Taylor series expansion of the true state dynamics, $d\mathbf{X}_t$, about the nominal state, $\mathbf{X}'(t)$.

$$d\mathbf{x}_t \approx A(\mathbf{X}'(t))\mathbf{x}_t dt + B\mathbf{u}(t)dW_t, \quad A(\mathbf{X}'(t)) = \left. \frac{\partial \mathbf{f}(\mathbf{X}(t))}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}'(t)} \quad (\text{A.4})$$

Let

$$\Sigma(t) = \mathbb{E}[(\mathbf{x}_t - \bar{\mathbf{x}}(t))(\mathbf{x}_t - \bar{\mathbf{x}}(t))^\top]. \quad (\text{A.5})$$

be the state error covariance, where $\bar{\mathbf{x}}$ is the mean state deviation. From the Expectation rule,

$$\dot{\bar{\mathbf{x}}}(t) = A(t)\bar{\mathbf{x}}(t). \quad (\text{A.6})$$

Under the assumption that $\bar{\mathbf{x}}(0) = \mathbf{0}_{n \times 1}$, integration of Eq. (A.6) yields $\bar{\mathbf{x}}(t) = \mathbf{0}_{n \times 1}$ for all time. Therefore, the error covariance is simplified to

$$\Sigma(t) = \mathbb{E}[\mathbf{x}_t \mathbf{x}_t^\top]. \quad (\text{A.7})$$

The $\Sigma(t)$ dynamics will be derived using Itô's formula for stochastic calculus with Brownian motion [26]. For a single element of the covariance matrix, $\Sigma_{ij} = \mathbb{E}[x_i x_j]$, the dynamics can be computed from the standard chain rule with the addition of an Itô correction term as follows.

$$d(x_i x_j) = x_j dx_i + x_i dx_j + \frac{1}{2} (\sigma B \mathbf{u}(t))^\top \frac{\partial^2 (x_i x_j)}{\partial \mathbf{e}^2} (\sigma B \mathbf{u}(t)) dt \quad (\text{A.8})$$

Evaluating the expectation for each $d\mathbb{E}[x_i x_j]$ and arranging in matrix form yields the following linear matrix differential equation for the state error covariance.

$$\dot{\Sigma}(t) = A(t)\Sigma(t) + \Sigma(t)A(t)^\top + \sigma^2 B \mathbf{u}(t) \mathbf{u}(t)^\top B^\top \quad (\text{A.9})$$

Integrating Eq. (A.9), the linearized state error covariance at the final time is

$$\Sigma(t_f) = \Phi(t_f, t_0) \Sigma(t_0) \Phi(t_f, t_0)^\top + \sigma^2 \int_{t_0}^{t_f} \Phi(t_f, \tau) B \mathbf{u}(\tau) \mathbf{u}(\tau)^\top B^\top \Phi(t_f, \tau)^\top d\tau \quad (\text{A.10})$$

The STM from t_i to t_j is denoted by $\Phi(t_j, t_i)$. This method may also be used to compute the covariance with additive noise.

Appendix B

Dynamical Systems and Orbits

B.1 Asteroid Sun-Terminator Orbits

Sun-terminator orbits are quasi-stable, nearly circular asteroid orbits with an orbit plane that is perpendicular to the Sun-asteroid line. SRP is a dominant force in the asteroid environment, which causes the Sun-terminator orbit plane to be offset from the asteroid center of mass. In this dissertation, asteroid Sun-terminator orbits are modeled the Hill three-body problem (H3BP). Consider an asteroid orbiting the Sun and let $\mathbf{X} = [\mathbf{r}^\top, \mathbf{v}^\top]^\top$ be the nominal, six-dimensional position and velocity state of a spacecraft operating in the asteroid's proximity. In the H3BP, it is assumed that the Sun has a much greater mass than the asteroid and the mass of the spacecraft is negligible. The H3BP coordinate frame, $\{\hat{e}_x, \hat{e}_y, \hat{e}_z\}$, is centered on the asteroid and rotates with the asteroid's orbit about the Sun: \hat{e}_x points in the sunlight direction, \hat{e}_z points in the direction of the asteroid's orbital angular momentum vector, and \hat{e}_y completes the right-handed coordinate frame. It will be assumed that the asteroid is in a circular orbit about the Sun and the rate of the frame rotation is constant. H3BP dynamics augmented with SRP are given by

$$\frac{d^2\mathbf{r}}{dt^2} = -2\omega\tilde{e}_z \cdot \frac{d\mathbf{r}}{dt} + \omega^2(3\hat{e}_x\hat{e}_x^\top - \hat{e}_z\hat{e}_z^\top)\mathbf{r} + \frac{\partial V_g}{\partial \mathbf{r}} + a_{SRP}\hat{e}_x + \mathbf{u}, \quad (\text{B.1})$$

in which V_g is the asteroid gravitational potential, a_{SRP} is the SRP acceleration, and ω represents the constant rate of coordinate frame rotation dictated by the asteroid's angular rate about the Sun [21]. The tilde operator in \tilde{e}_z represents a skew-symmetric matrix equivalent to the vector cross product operation. The SRP acceleration, a_{SRP} , is approximated by the cannonball model below, where p_0 is the SRP at 1 AU, ρ is the spacecraft reflectance, Z is the spacecraft mass-to-area ratio, and d is the asteroid distance from the

Sun [21, 44].

$$a_{SRP} = \frac{p_0(1 + \rho)(\text{AU})^2}{Zd^2} \quad (\text{B.2})$$

AU and d must have the same units in Eq. (B.2). Throughout the dissertation, the initial conditions of Sun-terminator orbits are offset from the $\hat{\mathbf{y}} - \hat{\mathbf{z}}$ plane by $r^3 a_{SRP} / \mu$ in the sunlight direction to balance SRP acceleration on the spacecraft [21]. Additionally, the asteroid gravitational potential is modeled to include J_2 and J_3 zonal harmonics:

$$\begin{aligned} V_g &= \frac{\mu}{r} - \frac{\mu}{r} \sum_{\ell=2}^3 J_\ell \left(\frac{r_b}{r} \right)^\ell P_\ell[\sin \phi], \\ &= \frac{\mu}{r} - \frac{\mu J_2 r_b^2}{2r^3} (3(\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}})^2 - 1) + \frac{\mu J_3 r_b^3}{2r^4} (\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}}) (5(\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}})^2 - 3), \end{aligned} \quad (\text{B.3})$$

where P_ℓ represent Legendre polynomials. In any examples that only consider asteroid point mass gravity, J_2 and J_3 are set to zero. Nondimensionalizing by the constant r_b in the length scale (the mean radius of the asteroid) and $1/n_b$ in the time scale (with n_b being the mean motion at the asteroid surface), the nondimensional spacecraft acceleration is

$$\begin{aligned} \frac{d^2 \mathbf{r}_{nd}}{d\tau^2} &= -2\Omega \tilde{\mathbf{e}}_z \cdot \frac{d\mathbf{r}_{nd}}{d\tau} + \Omega^2 (3\hat{\mathbf{e}}_x \hat{\mathbf{e}}_x^\top - \hat{\mathbf{e}}_z \hat{\mathbf{e}}_z^\top) \mathbf{r}_{nd} + \alpha_{SRP} \hat{\mathbf{e}}_x + \mathcal{U} \\ &+ \left[-\frac{1}{r_{nd}^2} + \frac{\gamma}{r_{nd}^4} (15(\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}})^2 - 3) - \frac{\zeta}{r_{nd}^5} (35(\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}})^3 - 15(\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}})) \right] \hat{\mathbf{r}} \\ &+ \left[-\frac{6\gamma}{r_{nd}^4} (\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}}) + \frac{\zeta}{r_{nd}^5} (15(\hat{\mathbf{e}}_z^\top \hat{\mathbf{r}})^2 - 3) \right] \hat{\mathbf{e}}_z \end{aligned} \quad (\text{B.4})$$

with the following nondimensional parameters:

$$\Omega = \frac{\omega}{n_b}, \quad \alpha_{SRP} = \frac{a_{SRP}}{r_b n_b^2}, \quad \mathcal{U} = \frac{\mathbf{u}}{r_b n_b^2}, \quad \gamma = \frac{\mu J_2 r_b^2}{2r_b^5 n_b^2}, \quad \zeta = \frac{\mu J_3 r_b^3}{2r_b^6 n_b^2}. \quad (\text{B.5})$$

Finally, the linearized dynamics matrix, $A(\mathbf{X}(t))$, is

$$A(\mathbf{X}(t)) = \begin{bmatrix} [0]_{3 \times 3} & I_{3 \times 3} \\ \Omega^2 (3\hat{\mathbf{e}}_x \hat{\mathbf{e}}_x^\top - \hat{\mathbf{e}}_z \hat{\mathbf{e}}_z^\top) + \frac{\partial^2 V_g}{\partial \mathbf{r}_{nd}^2} & -2\Omega \tilde{\boldsymbol{\epsilon}} \end{bmatrix}, \quad (\text{B.6})$$

where the matrix $\partial^2 V_g / \partial \mathbf{r}_{nd}^2$ is given as follows.

$$\frac{\partial^2 V_g}{\partial \mathbf{r}_{nd}^2} = c_1 \hat{\mathbf{r}} \hat{\mathbf{r}}^\top + c_2 \hat{\mathbf{e}}_z \hat{\mathbf{e}}_z^\top + c_3 \hat{\mathbf{e}}_z \hat{\mathbf{r}}^\top + c_4 \hat{\mathbf{r}} \hat{\mathbf{e}}_z^\top + c_5 I_{3 \times 3} \quad (\text{B.7})$$

$$c_1 = 31/r_{nd}^3 + \gamma/r_{nd}^5(15 - 105(\hat{e}_z^\top \hat{r})^2) + \lambda/r_{nd}^6(315(\hat{e}_z^\top \hat{r})^3 - 105(\hat{e}_z^\top \hat{r})) \quad (\text{B.8})$$

$$c_2 = -6\gamma/r_{nd}^5 + 30\lambda(\hat{e}_z^\top \hat{r})/r_{nd}^6 \quad (\text{B.9})$$

$$c_3 = 30\gamma/r_{nd}^5(\hat{e}_z^\top \hat{r}) - \lambda/r_{nd}^6(105(\hat{e}_z^\top \hat{r})^2 - 15) \quad (\text{B.10})$$

$$c_4 = 30\gamma(\hat{e}_z^\top \hat{r})/r_{nd}^5 - L/r_{nd}^6(105(\hat{e}_z^\top \hat{r})^2 - 15) \quad (\text{B.11})$$

$$c_5 = -1/r_{nd}^3 + \gamma/r_{nd}^5(15(\hat{e}_z^\top \hat{r})^2 - 3) - \lambda/r_{nd}^6(35(\hat{e}_z^\top \hat{r})^3 - 15(\hat{e}_z^\top \hat{r})) \quad (\text{B.12})$$

B.2 Near-Rectilinear Halo Orbit

In this dissertation, near-rectilinear halo orbits (NRHOs) are modeled in the circular restricted three-body problem (CR3BP). By convention, the CR3BP is described in a rotating coordinate frame $\{\hat{e}_x, \hat{e}_y, \hat{e}_z\}$ with an origin at the barycenter of the two primary bodies. In the Earth-Moon system, the \hat{e}_x -axis points toward the Moon, the \hat{e}_z -axis is aligned with the angular momentum vector of the primaries, and the \hat{e}_y -axis completes the right-handed coordinate frame. The CR3BP equations of motion dictate the six-dimensional position and velocity state of a spacecraft in this coordinate frame, $\mathbf{X}^\top = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]$. Quantities in the CR3BP are nondimensionalized such that the distance between the Earth and the Moon and the mean motion of the primaries are both unity. The nondimensional, deterministic CR3BP equations are

$$\frac{d^2x}{dt^2} = 2\frac{dy}{dt} + \frac{\partial U}{\partial x} \quad (\text{B.13})$$

$$\frac{d^2y}{dt^2} = -2\frac{dx}{dt} + \frac{\partial U}{\partial y} \quad (\text{B.14})$$

$$\frac{d^2z}{dt^2} = \frac{\partial U}{\partial z} \quad (\text{B.15})$$

with the pseudo-potential U given by

$$U = \frac{1}{2}(x^2 + y^2) + \frac{1 - \mu}{\sqrt{(x + \mu)^2 + y^2 + z^2}} + \frac{\mu}{\sqrt{(x - 1 + \mu)^2 + y^2 + z^2}}. \quad (\text{B.16})$$

The CR3BP has five equilibrium points, also known as Lagrange or libration points, which are denoted by L_1 through L_5 .

Halo orbits bifurcate from in-plane Lyapunov orbits at the collinear equilibrium points (L_1 through L_3) [88]. In the Earth-Moon system, halo orbits from L_1 and L_2 will evolve out of plane until they resemble polar lunar orbits. The 9:2 synodic resonant southern L_2 NRHO is considered throughout this dissertation — this orbit is representative of the Lunar Gateway orbit [46]. The 9:2 metric indicates that the spacecraft experiences nine orbit periods along the NRHO for every two orbits of the Moon about the Earth.

Appendix C

Tensor Calculations

C.1 STT Dynamics

The STT dynamics are given by the following differential equations. The STM is initialized as an identity matrix, and the higher-order STTs begin with the initial condition of $\Phi_{i_1 \dots i_m} = 0$.

$$\frac{d}{dt} \Phi_{j i_1} = \mathcal{A}_{j k_1} \Phi_{k_1 i_1} \quad (\text{C.1})$$

$$\frac{d}{dt} \Phi_{j i_1 i_2} = \mathcal{A}_{j k_1} \Phi_{k_1 i_1 i_2} + \mathcal{A}_{j k_1 k_2} \Phi_{k_1 i_1} \Phi_{k_2 i_2} \quad (\text{C.2})$$

$$\begin{aligned} \frac{d}{dt} \Phi_{j i_1 i_2 i_3} &= \mathcal{A}_{j k_1} \Phi_{k_1 i_1 i_2 i_3} + \mathcal{A}_{j k_1 k_2} (\Phi_{k_1 i_1} \Phi_{k_2 i_2 i_3} + \Phi_{k_1 i_1 i_2} \Phi_{k_2 i_3} + \Phi_{k_1 i_1 i_3} \Phi_{k_2 i_2}) \\ &+ \mathcal{A}_{j k_1 k_2 k_3} \Phi_{k_1 i_1} \Phi_{k_2 i_2} \Phi_{k_3 i_3} \end{aligned} \quad (\text{C.3})$$

$$\begin{aligned} \frac{d}{dt} \Phi_{j i_1 i_2 i_3 i_4} &= \mathcal{A}_{j k_1} \Phi_{k_1 i_1 i_2 i_3 i_4} + \mathcal{A}_{j k_1 k_2} (\Phi_{k_1 i_1 i_2 i_3} \Phi_{k_2 i_4} + \Phi_{k_1 i_1 i_2 i_4} \Phi_{k_2 i_3} + \Phi_{k_1 i_1 i_3 i_4} \Phi_{k_2 i_2} \\ &+ \Phi_{k_1 i_1 i_2} \Phi_{k_2 i_3 i_4} + \Phi_{k_1 i_1 i_3} \Phi_{k_2 i_2 i_4} + \Phi_{k_1 i_1 i_4} \Phi_{k_2 i_2 i_3} + \Phi_{k_1 i_1} \Phi_{k_2 i_2 i_3 i_4}) \\ &+ \mathcal{A}_{j k_1 k_2 k_3} (\Phi_{k_1 i_1 i_2} \Phi_{k_2 i_3} \Phi_{k_3 i_4} + \Phi_{k_1 i_1 i_3} \Phi_{k_2 i_2} \Phi_{k_3 i_4} + \Phi_{k_1 i_1 i_4} \Phi_{k_2 i_2} \Phi_{k_3 i_3} \\ &+ \Phi_{k_1 i_1} \Phi_{k_2 i_2 i_3} \Phi_{k_3 i_4} + \Phi_{k_1 i_1} \Phi_{k_2 i_2 i_4} \Phi_{k_3 i_3} + \Phi_{k_1 i_1} \Phi_{k_2 i_2} \Phi_{k_3 i_3 i_4}) \\ &+ \mathcal{A}_{j k_1 k_2 k_3 k_4} \Phi_{k_1 i_1} \Phi_{k_2 i_2} \Phi_{k_3 i_3} \Phi_{k_4 i_4} \end{aligned} \quad (\text{C.4})$$

C.2 Computing Ω Tensors

To simplify the notation, let $\mathbf{w}^0 = \delta\mathbf{w}(t_0)$, $\mathbf{q}^0 = \delta\mathbf{q}(t_0)$, $\mathbf{w}^t = \delta\mathbf{w}(t)$, and $\mathbf{q}^t = \delta\mathbf{q}(t)$. From Eqs. (6.50) through (6.53) truncated at the third order tensor:

$$q_k^0 \approx \mathcal{Q}_{k\ell_1} w_{\ell_1}^0 + \frac{1}{2} \mathcal{Q}_{k\ell_1\ell_2} w_{\ell_1}^0 w_{\ell_2}^0, \quad (\text{C.5})$$

$$q_i^t \approx \Phi_{ik_1} q_{k_1}^0 + \frac{1}{2} \Phi_{ik_1k_2} q_{k_1}^0 q_{k_2}^0, \quad (\text{C.6})$$

$$w_j^t \approx \mathcal{W}_{ji_1} q_{i_1}^t + \frac{1}{2} \mathcal{W}_{ji_1i_2} q_{i_1}^t q_{i_2}^t. \quad (\text{C.7})$$

Note that j and ℓ indices are of dimension n_w , whereas k and i indices are of dimension n_q . Substituting Eq. (C.6) in Eq. (C.7) and retaining up to third-order tensors results in

$$w_j^t \approx \mathcal{W}_{ji_1} \Phi_{i_1k_1} q_{k_1}^0 + \frac{1}{2} (\mathcal{W}_{ji_1} \Phi_{i_1k_1k_2} + \mathcal{W}_{ji_1i_2} \Phi_{i_1k_1} \Phi_{i_2k_2}) q_{k_1}^0 q_{k_2}^0. \quad (\text{C.8})$$

Furthermore, substituting Eq. (C.5) in Eq. (C.8) results in

$$\begin{aligned} w_j^t \approx & \mathcal{W}_{ji_1} \Phi_{i_1k_1} \mathcal{Q}_{k_1\ell_1} w_{\ell_1}^0 \\ & + \frac{1}{2} [\mathcal{W}_{ji_1} \Phi_{i_1k_1} \mathcal{Q}_{k_1\ell_1\ell_2} + (\mathcal{W}_{ji_1} \Phi_{i_1k_1k_2} + \mathcal{W}_{ji_1i_2} \Phi_{i_1k_1} \Phi_{i_2k_2}) \mathcal{Q}_{k_1\ell_1} \mathcal{Q}_{k_2\ell_2}] w_{\ell_1}^0 w_{\ell_2}^0. \end{aligned} \quad (\text{C.9})$$

From Eq. (6.55),

$$w_j^t \approx \Omega_{j\ell_1} w_{\ell_1}^0 + \frac{1}{2} \Omega_{j\ell_1\ell_2} w_{\ell_1}^0 w_{\ell_2}^0. \quad (\text{C.10})$$

Matching terms in Eqs. (C.9) and (C.10) by tensor order,

$$\Omega_{j\ell_1} = \mathcal{W}_{ji_1} \Phi_{i_1k_1} \mathcal{Q}_{k_1\ell_1}, \quad (\text{C.11})$$

$$\Omega_{j\ell_1\ell_2} = \mathcal{W}_{ji_1} \Phi_{i_1k_1} \mathcal{Q}_{k_1\ell_1\ell_2} + (\mathcal{W}_{ji_1} \Phi_{i_1k_1k_2} + \mathcal{W}_{ji_1i_2} \Phi_{i_1k_1} \Phi_{i_2k_2}) \mathcal{Q}_{k_1\ell_1} \mathcal{Q}_{k_2\ell_2}. \quad (\text{C.12})$$

C.3 Computing $\tilde{\mathcal{T}}$ Tensors

Begin by expanding $\delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta}$ in terms of the TSE in Eq. (6.35) and retaining terms up to fifth order (i.e., the vector appears up to five times in a single term). Note that higher-order terms can be included to compute $\tilde{\mathcal{T}}$ tensors beyond fifth order.

$$\begin{aligned} \delta\boldsymbol{\eta}^\top \delta\boldsymbol{\eta} &= \left(\mathcal{T}^{(2)}\boldsymbol{v}\right)^\top \mathcal{T}^{(2)}\boldsymbol{v} \\ &+ \left(\mathcal{T}^{(2)}\boldsymbol{v}\right)^\top \mathcal{T}^{(3)}\boldsymbol{v}^2 \\ &+ \frac{1}{3} \left(\mathcal{T}^{(2)}\boldsymbol{v}\right)^\top \mathcal{T}^{(4)}\boldsymbol{v}^3 + \frac{1}{4} \left(\mathcal{T}^{(3)}\boldsymbol{v}^2\right)^\top \mathcal{T}^{(3)}\boldsymbol{v}^2 \\ &+ \frac{1}{12} \left(\mathcal{T}^{(2)}\boldsymbol{v}\right)^\top \mathcal{T}^{(5)}\boldsymbol{v}^4 + \frac{1}{6} \left(\mathcal{T}^{(3)}\boldsymbol{v}^2\right)^\top \mathcal{T}^{(4)}\boldsymbol{v}^3 \end{aligned} \quad (\text{C.13})$$

The goal is to determine tensors $\tilde{\mathcal{T}}^{(2)}$ through $\tilde{\mathcal{T}}^{(5)}$ such that Eq. (6.36) and Eq. (C.13) are equivalent. The second-order term in Eq. (C.13) (wherein the vector \boldsymbol{v} appears twice) corresponds to the matrix

$$\tilde{\mathcal{T}}^{(2)} = \mathcal{T}^{(2)\top} \mathcal{T}^{(2)}. \quad (\text{C.14})$$

For the third-order term, the goal is to determine a tensor $\tilde{\mathcal{T}}^{(3)}$ for which

$$\tilde{\mathcal{T}}^{(3)}\boldsymbol{v}^3 = \left(\mathcal{T}^{(2)}\boldsymbol{v}\right)^\top \mathcal{T}^{(3)}\boldsymbol{v}^2. \quad (\text{C.15})$$

The right-hand side of Eq. (C.15) can be expanded as follows:

$$\left(\mathcal{T}^{(2)}\boldsymbol{v}\right)^\top \mathcal{T}^{(3)}\boldsymbol{v}^2 = \left[\mathcal{T}_{1i_1}v_{i_1} \quad \dots \quad \mathcal{T}_{ni_1}v_{i_1} \right]^\top \begin{bmatrix} \mathcal{T}_{1i_2i_3}v_{i_2}v_{i_3} \\ \vdots \\ \mathcal{T}_{ni_2i_3}v_{i_2}v_{i_3} \end{bmatrix} \quad (\text{C.16})$$

$$= \mathcal{T}_{k_1i_1} \mathcal{T}_{k_1i_2i_3} v_{i_1} v_{i_2} v_{i_3}. \quad (\text{C.17})$$

From Eqs. (C.15) and (C.17),

$$\tilde{\mathcal{T}}_{i_1i_2i_3}v_{i_1}v_{i_2}v_{i_3} = \mathcal{T}_{k_1i_1} \mathcal{T}_{k_1i_2i_3}v_{i_1}v_{i_2}v_{i_3}. \quad (\text{C.18})$$

Thus, each individual entry, $\tilde{\mathcal{T}}_{i_1i_2i_3}$, of the tensor $\tilde{\mathcal{T}}^{(3)}$ may be computed as follows.

$$\tilde{\mathcal{T}}_{i_1i_2i_3} = \mathcal{T}_{k_1i_1} \mathcal{T}_{k_1i_2i_3} \quad (\text{C.19})$$

A similar procedure can be used to compute all higher-order $\tilde{\mathcal{T}}$ tensors. Equations for the elements of $\tilde{\mathcal{T}}^{(4)}$ and $\tilde{\mathcal{T}}^{(5)}$ are given below.

$$\tilde{\mathcal{T}}_{i_1 \dots i_4} = \frac{1}{3} \mathcal{T}_{k_1 i_1} \mathcal{T}_{k_1 i_2 i_3 i_4} + \frac{1}{4} \mathcal{T}_{k_1 i_1 i_2} \mathcal{T}_{k_1 i_3 i_4} \quad (\text{C.20})$$

$$\tilde{\mathcal{T}}_{i_1 \dots i_5} = \frac{1}{12} \mathcal{T}_{k_1 i_1} \mathcal{T}_{k_1 i_2 \dots i_5} + \frac{1}{6} \mathcal{T}_{k_1 i_1 i_2} \mathcal{T}_{k_1 i_3 i_4 i_5} \quad (\text{C.21})$$

C.4 Computing $\check{\mathcal{T}}$ Tensors

To simplify the notation, let

$$R = \left(\mathcal{T}^{(2)} \right)^{-1}. \quad (\text{C.22})$$

The goal of this section is to determine a tensor $\check{\mathcal{T}}^{(m)}$ such that

$$\check{\mathcal{T}}^{(m)} \mathbf{z}^m = \tilde{\mathcal{T}}^{(m)} (R\mathbf{z})^m. \quad (\text{C.23})$$

Expanding both both sides of Eq. (C.23),

$$\check{\mathcal{T}}_{i_1 \dots i_m} z_{i_1} \dots z_{i_m} = \tilde{\mathcal{T}}_{k_1 \dots k_m} (R\mathbf{z})_{k_1} \dots (R\mathbf{z})_{k_m} \quad (\text{C.24})$$

$$= \tilde{\mathcal{T}}_{k_1 \dots k_m} (R_{k_1 i_1} z_{i_1}) \dots (R_{k_m i_m} z_{i_m}) \quad (\text{C.25})$$

$$= \tilde{\mathcal{T}}_{k_1 \dots k_m} R_{k_1 i_1} \dots R_{k_m i_m} z_{i_1} \dots z_{i_m}. \quad (\text{C.26})$$

From Eq. (C.26), each individual entry, $\check{\mathcal{T}}_{i_1 \dots i_m}$, of $\check{\mathcal{T}}^{(m)}$ may be computed by the following equation.

$$\check{\mathcal{T}}_{i_1 \dots i_m} = \tilde{\mathcal{T}}_{k_1 \dots k_m} R_{k_1 i_1} \dots R_{k_m i_m} \quad (\text{C.27})$$

C.5 Computing Γ Tensors

Begin by defining two intermediate tensors, $\mathcal{B}^{(k+2)}$ and $\mathcal{C}^{(k+2)}$ such that

$$\mathcal{B}^{(k+2)} \mathbf{z}^{k+1} = \left(\mathbf{z}^\top \mathbf{z} \right) k \check{\mathcal{S}}^{(k)} \mathbf{z}^{k-1}, \quad (\text{C.28})$$

$$\mathcal{C}^{(k+2)} \mathbf{z}^{k+1} = -2 \check{\mathcal{S}}^{(k)} \mathbf{z}^k \mathbf{z}. \quad (\text{C.29})$$

From Eqns. (6.84) and (6.85), $\Gamma^{(k+2)}$ may be computed from $\mathcal{B}^{(k+2)}$ and $\mathcal{C}^{(k+2)}$:

$$\Gamma^{(k+2)} \mathbf{z}^{k+1} = Y^{-1} \left(\mathcal{B}^{(k+2)} + \mathcal{C}^{(k+2)} \right) \mathbf{z}^{k+1}. \quad (\text{C.30})$$

$\mathcal{B}^{(k+2)}$, $\mathcal{C}^{(k+2)}$, and $\Gamma^{(k+2)}$ are computed as follows.

$$\mathcal{B}_{i_1 \dots i_{k+2}} = \begin{cases} \check{\mathcal{S}}_{i_1 \dots i_k}, & \text{if } i_{k+1} = i_{k+2} \\ 0, & \text{otherwise} \end{cases} \quad (\text{C.31})$$

$$\mathcal{C}_{i_1 \dots i_{k+2}} = \begin{cases} \check{\mathcal{S}}_{i_3 \dots i_{k+2}}, & \text{if } i_1 = i_2 \\ 0, & \text{otherwise} \end{cases} \quad (\text{C.32})$$

$$\Gamma_{i_1 \dots i_{k+2}} = Q_{i_1 j}^{-1} \left(\mathcal{B}_{j i_2 \dots i_{k+2}} + \mathcal{C}_{j i_2 \dots i_{k+2}} \right) \quad (\text{C.33})$$