

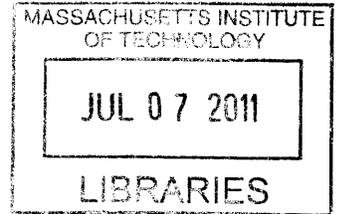
Optimal Trajectory-Shaping with Sensitivity and Covariance Techniques

by

Todd V. Small

B.S. Astronautical Engineering

United States Air Force Academy, 2008



Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

ARCHIVES

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2010

©2010 Todd V. Small. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author:.....

Department of Aeronautics and Astronautics

May 21, 2010

Certified by:.....

Steven R. Hall, Ph.D.

Professor of Aeronautics and Astronautics

Thesis Supervisor

Certified by:.....

Ronald J. Proulx, Ph.D.

Principle Member of the Technical Staff

The Charles Stark Draper Laboratory, Inc.

Thesis Supervisor

Accepted by:.....

Eytan H. Modiano

Associate Professor of Aeronautics and Astronautics

Chair, Committee on Graduate Students

Optimal Trajectory-Shaping with Sensitivity and Covariance Techniques

by
Todd V. Small

Submitted to the Department of Aeronautics and Astronautics
on May 21, 2010, in Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

Abstract

Traditional trajectory design approaches apply optimal control techniques to maximize desired performance, subject to specified constraints. Normal metrics and constraints are composed of the deterministic states and controls in the plant dynamics, so classical design methods do not directly address trajectory robustness in the presence of system uncertainties. This work explores the introduction of uncertainty directly into the trajectory design process. The state transition (sensitivity) and covariance matrices both measure the impact of plant uncertainty, and each of these mathematical constructs can be adjoined to the trajectory optimization problem to generate solutions that are less sensitive to prevalent uncertainties. A simple Zermelo boat problem is used to compare the methodologies for any combination of state initialization errors, state process noise, parametric biases, and parametric process noise, under any predefined feedback control law. The covariance technique is shown to possess several advantages over the sensitivity technique. Subsequently, the covariance method is used to simultaneously design reference trajectories and feedback control laws with closed-loop performance constraints for the Zermelo problem.

The covariance trajectory-shaping technique is then applied to a generic hypersonic recoverable reentry vehicle. The trajectories include uncertainties in atmospheric density, axial and normal force coefficients, commanded attitude, and initial position and velocity. Reachability footprints with uncertainty bounds are generated by the trajectory-shaping methodology, and shown to extend the vehicle's range of confidence. Relative to a fixed recovery site within the footprint boundary, the covariance technique improves the circular error probable (CEP) radius by almost 50%. Lastly, by segmenting the problem, trajectory designs successfully reach the recovery site using a balance of dispersion penalties and maximum intermediate maneuvers. Improvements in final CEP are shown to require sacrifices in planned maneuvering.

Thesis Supervisor: Steven R. Hall, Ph.D.

Title: Professor of Aeronautics and Astronautics and MacVicar Faculty Fellow

Thesis Supervisor: Ronald J. Proulx, Ph.D.

Title: Principle Member of the Technical Staff, C.S.D.L.

Acknowledgments

First, I would like to thank those who made my work at Draper Laboratory and MIT possible. To my advisor Ron Proulx, thank you for your tremendous time, energy, insight, and support throughout my two years at Draper. Thank you for challenging me with high expectations and for generously providing hours of mentorship. It has been an absolute pleasure and privilege to work with you. I would also like to extend a tremendous thank you to David Benson for devising the original concept for this research, and for providing continual insight and guidance as it progressed. Your mentorship was absolutely critical to the completion of this work. Many thanks to Jim Shearer and Roy Setterlund for securing the project foundations, and for trusting my work when efficiency and accuracy were critical. Thank you as well to Dick Phillips for sharing your knowledge and computations regarding circular error probables, which provided an immensely useful metric for depicting reentry trajectory robustness. Thanks also to Professor Steve Hall for your thoughtful comments on my thesis work, and to each of my MIT professors for stretching me academically. I am also indebted to the many individuals who helped me reach MIT and Draper in the first place, including each and every one of the teachers and professors who invested in my education.

I would also like to thank all the great friends who made my time at Draper and MIT enjoyable. To Chris, Will, Don, Jared, and Corey, thanks for being such great housemates and friends. I will certainly miss living at 43 Raymond Street. Will, you deserve extra thanks for your help as we suffered through four difficult classes together. Thanks as well to the rest of the Air Force crew, the CBC gang, and the Worcester crowd for all your friendship and support. I will miss my time in Boston specifically because of you. Finally, I would like to thank my family, whose love and support inspires me every step of the way.

Acknowledgment

May 21, 2010

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Project 23907-001, Internal Research and Development.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Todd V. Small

21 May 2010

Date

Assignment

Draper Laboratory Report Number T-1666

In consideration of the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Todd V. Small

21 May 2010

Date

Contents

1	Introduction	11
1.1	Thesis Overview	15
2	Sensitivity and Covariance Trajectory-Shaping Methods	17
2.1	Nominal Reference Trajectory Design	17
2.2	Sensitivity Matrix	19
2.2.1	Dynamics	20
2.2.2	Mathematical Properties	21
2.2.3	Sensitivity Problem Augmentation	23
2.3	Covariance Matrix	24
2.3.1	Dynamics	24
2.3.2	Mathematical Properties	28
2.3.2.1	Proof of Covariance Transformation Equivalence	29
2.3.3	Covariance Problem Augmentation	32
2.4	Parametric Uncertainties	33
2.5	Mathematical Link	34
3	Zermelo Boat Problem	37
3.1	Zermelo Problem Setup	38
3.2	Fixed-Target Precision	40
3.2.1	Parametric Uncertainties	41
3.2.1.1	Constant Bias	41
3.2.1.2	Process Noise	47
3.2.1.3	Bias and Noise Combination	52
3.2.1.4	Parametric Uncertainty Comparison	56
3.2.2	State and Parametric Uncertainty Combinations	59
3.2.3	Sensitivity vs. Covariance Summary	61
3.3	Maximum Capability Confidence	63
3.4	Simultaneous Feedback Optimization	66
4	Hypersonic Reentry Vehicle Problem	73
4.1	Nominal Trajectory Design	74
4.1.1	Coordinate Frame Summary	74
4.1.1.1	Earth-Centered-Inertial (ECI)	74
4.1.1.2	Earth-Centered-Earth-Fixed (ECEF)	75

4.1.1.3	Up-Downrange-Crossrange (UDC) and Up-East-North (UEN)	75
4.1.1.4	Velocity (V)	76
4.1.1.5	Non-Rolling Body (B)	76
4.1.2	3 DOF Dynamics	77
4.1.3	Vehicle Model	81
4.1.4	Nominal Optimization Problem	82
4.1.4.1	Initial and Terminal Constraints	83
4.1.4.2	Interior Constraints	85
4.1.4.3	Scaling	87
4.2	Nominal Maximum Capabilities	89
4.3	Model Uncertainties	96
4.3.1	Covariance Propagation vs. Discretization	101
4.4	Maximum Range Confidence	103
4.4.1	Circular Error Probable (CEP)	103
4.4.2	Shaping to Reduce CEP	105
4.5	Fixed-Recovery Site Precision	106
4.5.1	Maximum Precision	109
4.5.2	Maximum Crossrange Turn	110
5	Conclusions	117
5.1	Future Work	118
A	Reentry Jacobian Calculations	121

Chapter 1

Introduction

Guidance and mission planning applications begin with the design of a reference trajectory to provide a set of desired states for a system to follow. The reference trajectory can be designed in a multitude of ways, but a common method is to formulate and solve an optimal control problem. The optimization process guarantees that the trajectory optimizes a desired performance index while satisfying any specified constraints. In particular, dynamic constraints ensure the state trajectory evolves in a feasible manner. Guidance and mission planning applications typically generate reference trajectories for two basic scenarios. The first is a fixed-target problem, in which all of the trajectory's initial and final conditions are fixed, and performance metrics are formulated to find different ways to guide the system between fixed terminal conditions. The second is a maximum capability problem, in which one or more of the terminal conditions are left open, and performance metrics are intended to maximize a system capability dependent on the open states. Although not necessarily all-inclusive, this dichotomy provides an important distinction for two significant classes of trajectory design goals.

The trajectory design process can be applied to virtually any system, ranging from chemical mixing processes to electrical circuit problems to atmospheric reentry profiles. Regardless of the application, a set of states are selected to characterize the system, differential equations are chosen to model the dynamic behavior of the selected states, and performance goals and additional constraints are constructed as functions of those states. The dynamics, performance metric, and additional constraints are combined in an optimal control problem whose solution is optimal for the specified metric and feasible given each constraint. However, the design process does not directly account for any system uncertainties, and differences between the modeled system and actual system cause dispersions about the reference trajectory during implementation.

Common system uncertainties include initialization errors, modeling errors, and measurement errors. Initialization errors represent inaccuracies in the trajectory's initial conditions caused by processes that occur before the scope of the posed problem. Modeling errors represent discrepancies between the plant dynamics and the true system which arise from uncaptured state dynamics, model simplifications, or parametric uncertainties within the plant dynamics. If measurements are taken along a trajec-

tory to improve knowledge of the actual states, sensor noise leads to measurement errors that corrupt estimation updates and contribute to imperfect state knowledge. Each of these uncertainties and their associated dispersions can be characterized with statistical models.

Dispersions accumulate differently along different trajectories through the state space, and in problems with sufficient complexity, many *feasible* trajectories exist which satisfy every constraint. However, the optimal control problem only provides the single *optimal* solution for the specified performance index. Discovering alternative trajectories requires alternative performance metrics, but in the standard design problem, the metrics may only be functions of the states and control variables included in the plant dynamics. To design trajectories with smaller expected errors, the designer can attempt to infer how particular uncertainties translate to functions of the states and controls, and formulate metrics appropriately, but even in relatively simple systems, this often proves ineffectual. Instead, the responsibility of error reduction is relegated to the feedback controller design. Once a reference trajectory has been designed and expected uncertainties have been characterized, feedback controllers are constructed in an independent design process to correct deviations from the reference trajectory. Ultimately, the success with which the actual system tracks the reference trajectory during implementation depends on the errors in the assumed plant dynamics, how those errors accumulate over the particular reference trajectory, and the efficiency with which the feedback controller can mitigate dispersions. Robust feedback schemes can often provide sufficient closed-loop tracking along a predefined reference trajectory, but nonetheless, the controller's performance is limited by the error characteristics it inherits from the independent reference design process.

The goal of this thesis is to augment the standard trajectory design process to directly generate solutions that are less sensitive to expected uncertainties, and therefore induce smaller dispersions from the reference. One such method has been championed by Kumar and Seywald [15] and Seywald [13, 14], while another has been supported by Zimmer [18]. Kumar and Seywald define the sensitivity matrix, which mathematically represents the sensitivity of the reference to perturbations in the states. When this matrix, along with its required dynamics, is adjoined to the optimal control problem, the trajectory designer may include functions of the sensitivity matrix in the performance metric and additional constraints. In a similar process, Zimmer augments the optimal control problem with the covariance matrix and Riccati equation dynamics, which allows the designer to include functions of the covariance matrix in the performance metric and constraints. The sensitivity and covariance matrices can both be used to express either the open-loop or closed-loop effects of system uncertainties along a reference trajectory, but each provides a different mathematical perspective and addresses the problem slightly differently.

The authors test their respective techniques on several applications, ranging from simple two-dimensional Zermelo boat problems to orbital insertion scenarios, but neither provides a detailed comparison of the two trajectory-shaping techniques on a common problem. Zimmer cites References 13-15 in his dissertation [18] and claims that the sensitivity method lacks the ability to account for process noise and measurement noise uncertainties, but does not provide direct evidence to support this claim.

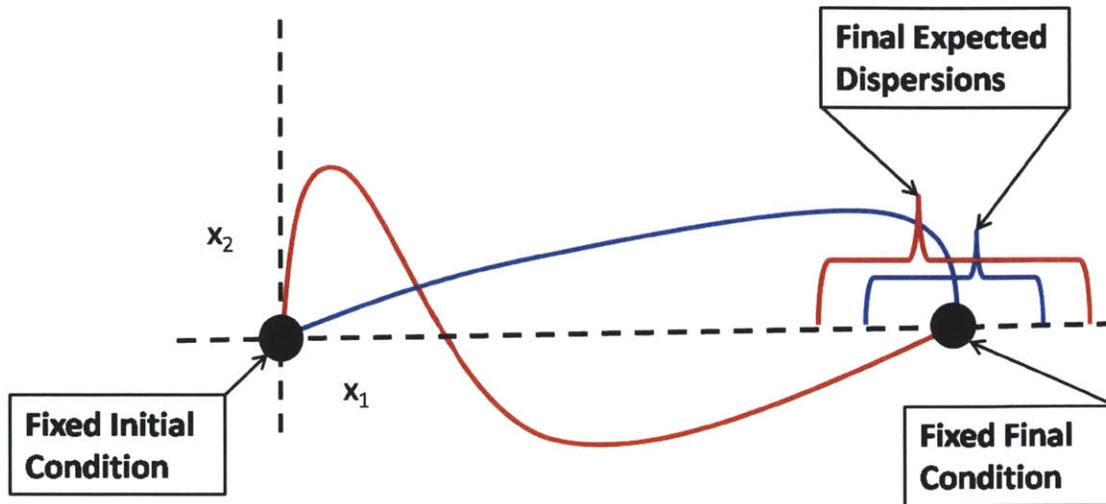


Figure 1-1: Fixed-target trajectory design illustration: Multiple feasible trajectories connect a single pair of initial and final conditions, although optimization techniques only provide the single optimal trajectory for the performance metric used. For a given set of uncertainties, the expected dispersions accumulate differently over different trajectories through the state space, and trajectory-shaping techniques directly solve for trajectories that reach the target with smaller expected dispersions.

To justify a preferred technique, this thesis attempts to convey a more complete comparison of the sensitivity and covariance trajectory-shaping methods, demonstrating their strengths, weaknesses, capabilities, and limitations in a simple problem using both fixed-target and maximum capability design goals. For a fixed target with multiple feasible trajectories, the trajectory-shaping technique allows the designer to directly solve for trajectories that reach the target with smaller expected dispersions, as illustrated in Figure 1-1. For maximum capability problems, the trajectory-shaping technique allows the designer to balance penalties on nominal capabilities and accumulated dispersions. When the correct balance is struck, it is possible to find trajectories that actually extend the expected capability, as illustrated in Figure 1-2.

Kumar and Seywald [15] also mention the ability of the sensitivity method to simultaneously design reference trajectories and feedback control laws, but they do not actually implement this concept. The covariance method possesses the same capability, but Zimmer [18] also chooses not to explore it. Using the same simple problem, this thesis demonstrates the simultaneous design of reference trajectories and feedback controllers, as well as the ability to penalize and constrain the system's closed-loop performance characteristics associated to the matching reference trajectory.

Once the comparisons are complete, a preference for the covariance method is defended, and this technique is applied to a significantly more complex problem involving the design of hypersonic reentry vehicles. In previous work, Undurti [17] conducted research on the capabilities of axisymmetric skid-to-turn (STT) reentry vehicles with lift-to-drag ratios (L/D_s) of approximately one and two, and Abraham-

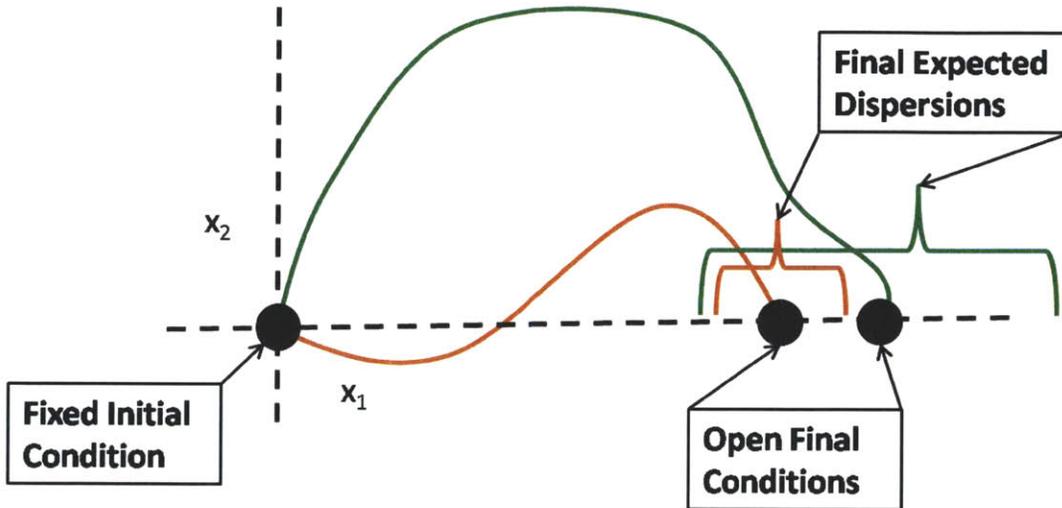


Figure 1-2: Maximum capability trajectory design illustration: When at least one of the initial or final conditions are left open, trajectories can be found which maximize some function of the open states. For a given set of uncertainties, the expected dispersions accumulate differently over different trajectories through the state space, and trajectory-shaping techniques directly solve for trajectories that may actually extend the confident range of the system.

son [1] looked at axisymmetric STT vehicles with L/Ds of approximately two. Recent research in reentry capabilities, however, has gravitated towards high-performance vehicles that use asymmetric characteristics to achieve even greater L/Ds and maneuverability [6, 11]. The performance data gathered from testing such vehicles will be useful for future high-performance reentry missions of all types, from manned exploration capsules to unmanned payload carriers.

While Abrahamson and Undurti used the standard optimization problem to characterize reentry vehicle maximum capabilities, they did not incorporate uncertainty considerations into the design process. Following recent research trends and extending the work completed by Abrahamson and Undurti, this problem poses a fictitious, small-scale, asymmetric recoverable reentry vehicle with a slightly higher L/D of approximately 2.5. To remain generic, the vehicle is not tied to any existing body, and its shape is not even specified in the problem formulation. The basic mission profile is invented to allow the vehicle to collect valuable aerodynamic performance data before deploying a guided parachute that returns it safely to the test crew.

The initial conditions of any endoatmospheric reentry scenario are dictated by an insertion method, but the remaining reentry constraints and the performance metrics can be varied to accomplish different trajectory design goals. For a reentry vehicle intended to gather performance data and test the system's limits, it is highly desirable to characterize the maximum downrange and crossrange capabilities by leaving the final recovery location open. In practice, however, final recovery sites may be established before the flight, and the vehicle may be expected to exercise its performance capabilities before reaching the recovery zone. In this case, it may

be desirable to maximize the precision with which the vehicle can reach the site, or maximize a certain type of maneuver along the trajectory before the vehicle is recovered.

Whatever the mission, uncertainties inevitably exist in the plant dynamics used to generate reference trajectories, and increasing the complexity of the system simply increases the potential sources of uncertainty. For maximum capability exercises, these uncertainties shrink confidence in the vehicle's maximum range. In fixed target cases, they degrade the precision with which the vehicle reaches the recovery site, and limit the size of achievable maneuvers that retain reasonable recovery precision. When typical reentry uncertainties are identified and modeled, the covariance trajectory-shaping technique can successfully address both fixed-recovery site and maximum capability design goals. As in the simple test problem, trajectories can be designed that are expected to reach recovery zones with increased precision, maximize achievable maneuvers before recovery, and even extend the vehicle's expected downrange and crossrange capabilities. Although this study poses a particular vehicle model to demonstrate these concepts, the problem can easily be extended to account for many different reentry scenarios – whether manned or unmanned – by simply exchanging the particular vehicle model and updating trajectory constraints appropriately.

1.1 Thesis Overview

The rest of the thesis is organized as follows. Chapter 2 provides background information on the sensitivity and covariance trajectory-shaping techniques. It summarizes the important properties of the sensitivity and covariance matrices, explains their mathematical significance, and describes how to augment the standard reference trajectory design process to include sensitivity and covariance considerations. Both techniques are then applied to the same simple problem in Chapter 3 to compare and contrast the mechanics of each implementation, and demonstrate their ability to handle different types of uncertainty models. Fixed-target problems are solved to obtain trajectories with reduced dispersions at fixed terminal conditions, and maximum capability problems are solved to more accurately depict the system's maximum capabilities with open terminal conditions. The simple problem is then used to demonstrate simultaneous design of optimal reference trajectories and linear feedback control laws, including the application of constraints on the system's closed-loop performance characteristics. Chapter 4 addresses the hypersonic reentry problem and applies the covariance trajectory-shaping technique to demonstrate fixed-terminal condition and maximum capability improvements in a significantly more complex, but realistic system. Chapter 5 summarizes final conclusions and proposes areas of future work.

Chapter 2

Sensitivity and Covariance Trajectory-Shaping Methods

As described in Chapter 1, the traditional reference trajectory design process employs optimal control techniques to generate reference trajectories without addressing robustness to system uncertainties. The state transition (sensitivity) and covariance matrices both measure the impact of uncertainty. Each of these mathematical constructs can be adjoined to the trajectory optimization problem to allow penalties and constraints on the trajectory robustness to identified uncertainties.

This chapter begins with a definition of the standard reference trajectory design problem without uncertainty considerations, and then introduces the sensitivity and covariance trajectory-shaping techniques. The sensitivity and covariance matrices are defined along with their dynamics and key mathematical properties, followed by an explanation of how to augment the standard trajectory design process to include sensitivity and covariance considerations under various uncertainty models and feedback schemes. Then, since the sensitivity and covariance matrices only account mathematically for state perturbations, the problem is updated once more to address uncertainties in other system parameters. Finally, a mathematical connection between the sensitivity and covariance matrices sheds further insight into the relationship of these two trajectory-shaping techniques before moving on to an example problem in Chapter 3.

2.1 Nominal Reference Trajectory Design

For any nominal system with state vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.1)$$

and control vector

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \quad (2.2)$$

the nominal reference trajectory design process solves for the states and controls that optimize any desired cost function of form

$$J = U(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f) + \int_{t_0}^{t_f} V(\mathbf{x}(t), \mathbf{u}(t), t) dt, \quad (2.3)$$

subject to the dynamic constraints given by an assumed plant dynamics

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (2.4)$$

Within the cost metric, the endpoint cost U can be any function of the terminal states and time, while the integrated cost V can be any function of the states, controls, and time along the entire trajectory. Event constraints, path constraints, state constraints, and control constraints can also be applied to the optimization. Event constraints can be written as

$$\mathbf{e}^L \leq \mathbf{e}(\mathbf{x}(t_0), \mathbf{x}(t_f), t_0, t_f) \leq \mathbf{e}^U, \quad (2.5)$$

which require any function \mathbf{e} of the terminal states and time to remain between the lower and upper bounds given by \mathbf{e}^L and \mathbf{e}^U , respectively. Path constraints can be written as

$$\mathbf{d}^L \leq \mathbf{d}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{d}^U, \quad (2.6)$$

which require any function \mathbf{d} of the states, controls, and time to remain between the lower and upper bounds given by \mathbf{d}^L and \mathbf{d}^U , respectively, everywhere along the trajectory. State bounds can be written as

$$\mathbf{x}^L \leq \mathbf{x}(t) \leq \mathbf{x}^U, \quad (2.7)$$

which require the states everywhere along the trajectory to remain within the lower and upper bounds given by \mathbf{x}^L and \mathbf{x}^U , respectively. Lastly, control bounds can be written as

$$\mathbf{u}^L \leq \mathbf{u}(t) \leq \mathbf{u}^U, \quad (2.8)$$

which require the controls everywhere along the trajectory to remain within the lower and upper bounds given by \mathbf{u}^L and \mathbf{u}^U , respectively. It is important to note that the cost metric and each of the constraints listed above can only be functions of time, the nominal states \mathbf{x} , and the nominal controls \mathbf{u} included in the plant dynamics. As a result, the standard trajectory optimization cannot directly penalize or constrain system responses to uncertainty. However, if the cost and constraints can be augmented to include functions of the sensitivity or covariance matrices, then the system's expected responses to uncertainty under various feedback control schemes

can be directly optimized to improve reference trajectory robustness.

Myriad solution techniques exist for solving optimal control problems. Betts provides a comprehensive overview of the most widely used techniques, including discussions of their specific strengths and weaknesses [2]. As a simplistic dichotomy, most techniques can be classified as either direct or indirect methods. Indirect methods attack the optimization as a traditional two-point boundary value problem. However, to ensure optimality with respect to the cost metric and feasibility with respect to each of the problem constraints, indirect methods require the analytic derivation of the necessary and boundary conditions, which requires expressions for the costates associated with each differential constraint and the Lagrange multipliers associated with each event and path constraint. For a complex system, these derivations can become very difficult and tedious, particularly because the switching behavior between active and inactive path constraints must be defined before executing the optimization. The two-point boundary value problem must also be initialized with guesses for the costates and multipliers, and appropriate guesses can be difficult to intuit. In addition, indirect methods generally demonstrate smaller convergence radii when compared to their direct method counterparts.

Direct methods, on the other hand, convert the trajectory optimization problem into a nonlinear programming problem. These methods employ an iterative search algorithm which requires an initial guess for only the states and controls, and the underlying nonlinear programming software calculates gradients with respect to the cost and constraints to adjust the trajectory until the metric is minimized and all constraints are satisfied within set tolerances. Direct methods generally display larger convergence radii, handle inequality path constraints, and do not require expressions and guesses for the costates or Lagrange multipliers, so they are more appropriate for application to the complex hypersonic test vehicle problem investigated in Chapter 4. Within the direct method branch itself, many different optimization techniques exist. However, for ease of problem formulation and convergence considerations, the DIDO direct pseudospectral collocation technique [12] is used to generate the results presented in this thesis.

2.2 Sensitivity Matrix

Kumar and Seywald [15] and Seywald [13, 14] introduce and utilize the sensitivity matrix for the purposes of robust reference trajectory design. Specifically, the mathematical properties of the sensitivity matrix allow it to represent the sensitivity of any function of the nominal states at any particular time to perturbations in any nominal states at any other times along the reference trajectory. As long as system uncertainties can be characterized as *state* perturbations, this capability can be used to explicitly characterize how the reference trajectory can be expected to respond to particular uncertainties. If the optimal control problem formulation is augmented to include the sensitivity matrix elements as additional states and the sensitivity matrix dynamics as additional dynamic constraints, then the cost metric, event constraints, and path constraints used in the optimization process can specifically penalize or con-

strain any desired functions of the sensitivity matrix. The dynamics of the sensitivity matrix also allow for uncertainty analysis under any linear feedback control law, and the trajectory designer can link the reference trajectory and controller design phases as closely as desired for a particular guidance or mission planning application.

2.2.1 Dynamics

When the plant dynamics given by Eq. (2.4) are linearized along a particular state trajectory $\bar{\mathbf{x}}(t)$, the state transition matrix $\Phi(t, t_0)$ describes how changes in the initial condition $\bar{\mathbf{x}}(t_0) = \mathbf{x}_0$ cause linear dispersions from the nominal solution. In other words, the state transition matrix represents the *sensitivity* of the nominal state trajectory $\bar{\mathbf{x}}(t)$ to perturbations in the initial condition \mathbf{x}_0 . This is exactly the definition of the sensitivity matrix $S(t|t_0, \mathbf{x}_0)$ introduced by Seywald and Kumar [15]. In this context, the sensitivity matrix and the state transition matrix are synonymous, so that

$$S(t | t_0, \mathbf{x}_0) = \Phi(t, t_0) . \quad (2.9)$$

The dynamics which govern the sensitivity matrix are thus equivalent to the well-known dynamics of the state transition matrix, which can be written as the differential equation

$$\dot{S}(t | t_0, \mathbf{x}_0) = GS(t | t_0, \mathbf{x}_0) , \quad (2.10)$$

with the initial condition

$$S(t_0 | t_0, \mathbf{x}_0) = I , \quad (2.11)$$

where G is the Jacobian

$$G = \left. \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}(t)} \quad (2.12)$$

evaluated along the nominal state solution. The initial condition given by Eq. (2.11) is always true, because the nominal trajectory evaluated at time t_0 is always equivalent to the initial condition \mathbf{x}_0 , so that $\bar{\mathbf{x}}(t_0) = \mathbf{x}_0$ by definition. If the initial condition is altered, the nominal trajectory shifts to satisfy this relationship.

The Jacobian G represents the linearization of the system dynamics about the nominal trajectory. However, if an allowance is made for the existence of a linear feedback control law, the controls are expressed as functions of the states as well as time, and the Jacobian calculation is expanded to

$$G_c = \left. \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u}(\mathbf{x}, t), t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}(t), \mathbf{u}=\bar{\mathbf{u}}(\bar{\mathbf{x}}, t)} = \left[\frac{\partial \mathbf{g}}{\partial \mathbf{x}} + \frac{\partial \mathbf{g}}{\partial \mathbf{u}} K \right] \Big|_{\mathbf{x}=\bar{\mathbf{x}}(t), \mathbf{u}=\bar{\mathbf{u}}(\bar{\mathbf{x}}, t)} , \quad (2.13)$$

where $\bar{\mathbf{u}}(\bar{\mathbf{x}}, t)$ denotes the control set along the nominal trajectory, and the matrix

$$K = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \quad (2.14)$$

represents the gain matrix for a linear feedback control scheme. The sensitivity dy-

namics with the updated Jacobian calculation are rewritten as

$$\dot{S}(t | t_0, \mathbf{x}_0) = G_c S(t | t_0, \mathbf{x}_0) . \quad (2.15)$$

Note that the sensitivity dynamics require the definition of a linear control law K , and that the problem changes depending on how K is defined. From the perspective of the plant dynamics, the nominal controls selected by the optimization are simply commands given as functions of time, no matter how K is selected. The nominal trajectory is a purely open-loop entity that does not change with the presence of feedback. However, feedback *does* affect the closed-loop response of the system to perturbations about the nominal trajectory. The feedback commands depend on the functionality between the states and controls given by the K matrix, and the sensitivity matrix dynamics capture the closed-loop behavior. Three major options exist for utilizing K in the desensitized trajectory design process::

1. Set $K = 0$ to omit feedback control and design a reference trajectory using the sensitivity matrix to characterize open-loop responses to perturbations about the nominal trajectory.
2. Predefine K to specify a feedback control law and design a reference trajectory using the sensitivity matrix to characterize closed-loop responses to perturbations about the nominal trajectory.
3. Include K in the optimization to simultaneously design the reference trajectory and the feedback controller using the sensitivity matrix to characterize closed-loop responses to perturbations about the nominal trajectory.

Each option is more complex to implement than its predecessor, so the designer must decide how much synergy is worth the effort.

2.2.2 Mathematical Properties

The sensitivity matrix shares all of the well-known mathematical properties of the state transition matrix, some important ones of which are adapted from Kumar and Seywald [15] and summarized in this section. These properties help describe the physical meaning of the sensitivity matrix, and together they can be used to express the sensitivity of any function of the states at any particular time to state perturbations anywhere else along the nominal trajectory.

1. $S(t | t_0, \mathbf{x}_0)$ represents the sensitivity of a nominal state trajectory to perturbations in its initial conditions, a relationship that can also be expressed mathematically as the partial derivative

$$S(t | t_0, \mathbf{x}_0) = \frac{\partial \bar{\mathbf{x}}(t)}{\partial \mathbf{x}_0} . \quad (2.16)$$

2. $S(t | t_0, \mathbf{x}_0)$ is nonsingular for all t so that its inverse exists, which represents the sensitivity of the initial conditions to state perturbations everywhere else along the nominal trajectory.
3. For all t and $t_1 \in [t_0, t_f]$, with and $\bar{\mathbf{x}}_1 = \bar{\mathbf{x}}(t_1)$,

$$S(t | t_0, \mathbf{x}_0) = S(t | t_1, \bar{\mathbf{x}}_1) S(t_1 | t_0, \mathbf{x}_0) . \quad (2.17)$$

Here, $\bar{\mathbf{x}}_1$ is defined as the nominal trajectory state evaluated at time t_1 . This property can also be expressed and understood as the product of the partial derivatives

$$\frac{\partial \bar{\mathbf{x}}(t)}{\partial \mathbf{x}_0} = \frac{\partial \bar{\mathbf{x}}(t)}{\partial \bar{\mathbf{x}}_1} \frac{\partial \bar{\mathbf{x}}_1}{\partial \mathbf{x}_0} . \quad (2.18)$$

In words, this property demonstrates that the sensitivity of the nominal trajectory to initial condition perturbations is equivalent to the product of two sensitivities: 1) The sensitivity of the nominal trajectory to perturbations at an intermediate time t_1 and 2) The sensitivity of the nominal trajectory at time t_1 to initial condition perturbations.

4. For all t and $t_1 \in [t_0, t_f]$, with $\bar{\mathbf{x}}_1 = \bar{\mathbf{x}}(t_1)$

$$S(t | t_1, \bar{\mathbf{x}}_1)^{-1} = \frac{\partial \bar{\mathbf{x}}_1}{\partial \bar{\mathbf{x}}(t)} . \quad (2.19)$$

This property shows mathematically that $S(t | t_1, \bar{\mathbf{x}}_1)^{-1}$ represents the sensitivity of the nominal trajectory evaluated at time t_1 to perturbations along the entire rest of the trajectory.

5. For any arbitrary, smooth, scalar function $h(\mathbf{x}, t)$ with $t_1 \in [t_0, t_f]$, the sensitivity of $h(\bar{\mathbf{x}}_1, t_1)$ with respect to perturbations at any time t along the nominal trajectory can be specified as

$$\Lambda(t, t_1) = \frac{\partial h(\bar{\mathbf{x}}_1, t_1)}{\partial \bar{\mathbf{x}}(t)} = \frac{\partial h(\mathbf{x}, t)}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}_1, t=t_1} S(t | t_1, \bar{\mathbf{x}}_1)^{-1} . \quad (2.20)$$

This final property provides a mathematical method for expressing the sensitivity of any smooth function of the nominal states at any particular time to perturbations anywhere else along the nominal trajectory

Using these properties, the designer can fully describe the sensitivity characteristics of the nominal trajectory. However, the nominal optimal control problem can only include functions of the *nominal* states and controls within its cost metric and constraints, so the problem must be augmented to allow the designer to penalize and constrain functions of the sensitivity matrix. These properties allow the designer to formulate sensitivity functions to appropriately address reference trajectory design goals with uncertainty considerations.

2.2.3 Sensitivity Problem Augmentation

Including functions of the sensitivity matrix within cost metrics and constraints can be accomplished by expanding the problem state vector to include each element of the sensitivity matrix $S(t)$ – where the simplified notation $S(t)$ is used to denote $S(t | t_0, \mathbf{x}_0)$ – and augmenting the differential constraints to include the sensitivity matrix dynamics. For a system with n nominal states, the sensitivity matrix has $n \times n$ dimension. The expanded state vector becomes

$$\mathbf{x}_S = \begin{bmatrix} \mathbf{x} \\ \text{vec}(S) \end{bmatrix} \quad (2.21)$$

and the augmented set of differential constraints becomes

$$\dot{\mathbf{x}}_S = \begin{bmatrix} \dot{\mathbf{x}} \\ \text{vec}(\dot{S}) \end{bmatrix}, \quad (2.22)$$

where vec is simply a function that reshapes any $n \times n$ matrix into a column vector with n^2 rows, with each row representing an individual element of the matrix. The augmented optimization problem contains n^2 additional states with an equal number of new differential constraints and costates. Numerically solving the augmented problem requires evaluating the sensitivity dynamics, which in turn requires linearizing the system about the nominal trajectory. The linearization can be computed analytically by deriving the Jacobian or it can be done numerically with automatic differentiation software or finite differencing. Either way, including the sensitivity dynamics significantly increases the problem complexity, even without optimizing feedback gains simultaneously. Regardless of the control law, the expanded problem formulation allows the cost metric, event constraints, path constraints, and state constraints to include functions of the sensitivity matrix as well as the nominal states, which can be written as

$$J_S = U(\mathbf{x}_S(t_0), \mathbf{x}_S(t_f), t_0, t_f) + \int_{t_0}^{t_f} V(\mathbf{x}_S(t), \mathbf{u}(t), t) dt, \quad (2.23)$$

$$\mathbf{e}_S^L \leq \mathbf{e}_S(\mathbf{x}_S(t_0), \mathbf{x}_S(t_f), t_0, t_f) \leq \mathbf{e}_S^U, \quad (2.24)$$

$$\mathbf{d}_S^L \leq \mathbf{d}_S(\mathbf{x}_S(t), \mathbf{u}(t), t) \leq \mathbf{d}_S^U, \quad (2.25)$$

and

$$\mathbf{x}_S^L \leq \mathbf{x}_S(t) \leq \mathbf{x}_S^U, \quad (2.26)$$

respectively. The designer's primary task then becomes determining the most important or influential uncertainties in the system, choosing functions of the nominal states which require reduced sensitivities, and using the properties of $S(t)$ to mathematically formulate cost functions and constraints to match the design goals appropriately. It is important to note that the sensitivity matrix does not provide a *statistical* measure of

expected errors due to uncertainties. Rather, the sensitivity matrix describes exactly how the trajectory *would* respond to differential state perturbations at any given time. Statistical models of the uncertainties do not even appear in the sensitivity dynamics or the augmented cost function.

Although the sensitivity problem formulation does not require explicit statistical definitions, the designer must still have detailed knowledge of the nature of the system uncertainties in order to appropriately select and weigh sensitivity penalties and constraints. For instance, some states may have greater uncertainty than others, and sensitivities to perturbations in these states should be more heavily penalized or constrained. The mechanics of appropriate cost selection are illustrated in the simple example problem explored in Chapter 3.

2.3 Covariance Matrix

Unlike the sensitivity matrix, the covariance matrix statistically expresses expected state dispersions from a nominal trajectory induced by uncertainties in the plant dynamics. Utilizing its mathematical properties, the covariance matrix can be transformed to express expected dispersions in terms of almost any arbitrary smooth functions of the nominal states without the need to reformulate the plant dynamics. When the optimal control problem is augmented to include the covariance matrix elements as additional states and its dynamics as additional dynamic constraints, the cost metric, event constraints, and path constraints can specifically penalize or constrain any desired functions of the covariance matrix. Like the sensitivity dynamics, the covariance dynamics can adapt to various linear feedback control laws, and allow the trajectory designer to link the reference trajectory and controller design phases as closely as required for a particular guidance or mission planning application.

2.3.1 Dynamics

The nominal states are modeled deterministically by the dynamics given in Eq. (2.4). However, uncertainties in that model can be expressed as random variables. The random variables can then be applied to the deterministic system through any appropriate functionality to create a stochastic system in which the states are functions of random variables. The covariance matrix captures the expected statistical behavior of the stochastic system, and when appropriate assumptions are made regarding the functionality and distribution of the stochastic state uncertainties, the Riccati equation governs its evolution along nominal trajectories. When the underlying statistical assumptions are adequate for the problem, the covariance matrix and the Riccati equation can be adjoined to the optimal control problem to directly penalize and constrain functions of the system's covariance matrix.

For a stochastic system in which the states are functions of random variables, the state covariance is most generally expressed as the expectation

$$P = E \left[(\mathbf{x} - E[\mathbf{x}]) (\mathbf{x} - E[\mathbf{x}])^T \right], \quad (2.27)$$

where $E[\mathbf{x}]$ denotes the expected value, or mean value, of the state vector subject to random variable uncertainty. Replacing $E[x_i]$ with μ_i to represent the mean values of each state, the covariance matrix for a system of n states is

$$P = \begin{bmatrix} E[(x_1 - \mu_1)(x_1 - \mu_1)] & E[(x_1 - \mu_1)(x_2 - \mu_2)] & \cdots & E[(x_1 - \mu_1)(x_n - \mu_n)] \\ E[(x_2 - \mu_2)(x_1 - \mu_1)] & E[(x_2 - \mu_2)(x_2 - \mu_2)] & \cdots & E[(x_2 - \mu_2)(x_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(x_n - \mu_n)(x_1 - \mu_1)] & E[(x_n - \mu_n)(x_2 - \mu_2)] & \cdots & E[(x_n - \mu_n)(x_n - \mu_n)] \end{bmatrix}. \quad (2.28)$$

The diagonal elements provide the variances for each state, which measure how far the states are expected to deviate from their mean values. The off-diagonal elements express the correlations between each state combination, which measure the linear interdependence in the expected behavior of each state. The general form of the covariance matrix can account for random variable uncertainties with any distributions and functional relationships to the deterministic states. Allowing the uncertainties to act nonlinearly within the nominal state dynamics, the stochastic system can be written as

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}, t), \quad (2.29)$$

where \mathbf{w} is a random variable accounting for nonlinear process noise. In practice, however, the covariance-trajectory shaping technique updates the status of any uncertain parameters within the plant dynamics to states with trivial dynamics, as explained in Section 2.4. This process shifts the linearization of the noise to the Jacobian and allows a simple additive functionality to sufficiently describe the uncertainty models. The stochastic system can thus be reduced to

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) + \mathbf{w}, \quad (2.30)$$

so that \mathbf{w} acts additively on the deterministic dynamics.

A common choice for characterizing uncertainty distributions is the normal (Gaussian) probability distribution. Many random processes in the physical world naturally resemble normal distributions, partly because they are often combinations of many smaller unobserved random events. The central limit theorem states that as the number of independent, similarly distributed random variables increases, the combined distribution approaches a normal distribution [3]. If the distribution also has zero mean, then the expected values of the stochastic dynamics are equivalent to the nominal deterministic dynamics. If the distribution is also assumed to be completely uncorrelated with time, then the combination of each of these assumptions allows the random variable uncertainty to be classified as additive white Gaussian noise (AWGN). Satisfying these assumptions, \mathbf{w} then possesses the statistical properties

$$E[\mathbf{w}(t)] = 0 \quad (2.31)$$

and

$$E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = Q(t)\delta(t - \tau), \quad (2.32)$$

in which $Q(t)$ is the spectral density of the white noise and $\delta(t - \tau)$ is the dirac delta function representing an impulse at time τ . Since the expected state values are

guaranteed to match the deterministic state trajectory under AWGN assumptions,

$$E[\mathbf{x}(t)] = \bar{\mathbf{x}}(t) . \quad (2.33)$$

The difference between the stochastic state realizations and the deterministic states represent dispersions from the deterministic state trajectory, written as

$$\delta\mathbf{x}(t) = \mathbf{x}(t) - \bar{\mathbf{x}}(t) . \quad (2.34)$$

The covariance for the stochastic system becomes

$$P = E\left[(\mathbf{x} - \bar{\mathbf{x}}(t))(\mathbf{x} - \bar{\mathbf{x}}(t))^T\right] = E[\delta\mathbf{x}\delta\mathbf{x}^T] , \quad (2.35)$$

which shows that the covariance of the stochastic system and the covariance of the state dispersions are equivalent under AWGN assumptions. In a continuous system without measurements, the linear evolution of the covariance matrix along a nominal trajectory is defined by the well-known Lyapunov equation

$$\dot{P} = GP + PG^T + Q \quad (2.36)$$

with initial condition

$$P(t_0) = P_0 . \quad (2.37)$$

Here, Q is the process noise spectral density given by Eq. (2.32), G is the linearization of the system model about the nominal trajectory as defined in Eq. (2.12), and P_0 contains the initial expected covariance of the states. Without measurements, the covariance is driven purely by the process noise model, the nominal system dynamics, and the particular trajectory about which the dynamics are linearized.

As with the sensitivity matrix, an allowance can be made for the existence of linear feedback control laws by using the expanded Jacobian calculation given by Eq. (2.13), so that the Lyapunov equation becomes

$$\dot{P} = G_c P + P G_c^T + Q . \quad (2.38)$$

The expanded Jacobian calculation requires the definition of the linear feedback gain matrix K to establish the functionality between the states and controls. As with the sensitivity matrix, K can be 1) set to zero to evaluate open-loop covariances about the reference, 2) predefined to evaluate closed-loop covariances about the reference for a specific feedback control law, or 3) optimized along with the trajectory.

In real-world trajectory implementation, estimates of the actual states are often improved by optimally combining propagated states with measurements taken along the actual trajectory. Errors or uncertainties in the system dynamics lead to differences between the propagated and actual trajectories, and measurements are used to reduce this error. In most cases, rather than detecting the state quantities directly, sensors measure some function of the states. If continuous measurements are taken

along a trajectory, they can be modeled continuously as

$$\mathbf{y} = \mathbf{m}(\mathbf{x}, t) . \quad (2.39)$$

However, sensor models also contain uncertainties that corrupt the accuracy of their outputs, and these uncertainties can also be modeled as random variables. If measurement uncertainty is assumed to be AWGN, the stochastic continuous measurement model becomes

$$\mathbf{y} = \mathbf{m}(\mathbf{x}, t) + \mathbf{v} , \quad (2.40)$$

where \mathbf{v} is a random variable accounting for additive measurement noise. The measurement noise then possesses the statistical properties

$$E[\mathbf{v}(t)] = 0 \quad (2.41)$$

and

$$E[\mathbf{v}(t)\mathbf{v}^T(\tau)] = R(t)\delta(t - \tau) , \quad (2.42)$$

where R is the measurement noise spectral density. If the state propagation and measurements are combined optimally with a continuous extended Kalman filter, the linear evolution of the covariance matrix along a nominal trajectory is defined by the Riccati equation

$$\dot{P} = G_c P + P G_c^T + Q - P M^T R^{-1} M P , \quad (2.43)$$

where M is the linearization of the deterministic portion of the continuous measurement model along the nominal trajectory, calculated by

$$M = \left. \frac{\partial \mathbf{m}(\mathbf{x}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}(t)} . \quad (2.44)$$

With continuous measurements, the covariance along a trajectory is driven by process noise (Q), measurement noise (R), measurement models (M), the deterministic dynamics (G), and the particular trajectory about which they are linearized. If discrete measurements are taken along the trajectory, the discrete Kalman filter can be used to augment Eq. (2.43) to account for covariance updates, but this process will not be utilized in this thesis.

In summary, when uncertainties in the plant dynamics and measurement models can be characterized as AWGN, the covariance of the state dispersions about the nominal trajectory are governed by the Lyapunov equation in the absence of measurements, and by the Riccati equation in the presence of measurements. The covariance matrix and its dynamics can then be augmented to the optimal control problem to penalize and constrain expected dispersions from the reference trajectory. However, the covariance matrix only contains statistical representations of *state* dispersions. The following section describes the additional mathematical properties which allow the final augmented optimal control problem to penalize and constrain the covariance of almost any function of the nominal states.

2.3.2 Mathematical Properties

The covariance matrix exhibits several mathematical properties that facilitate the augmentation of the optimal trajectory design process, including one which allows the designer to apply linear transformations to represent expected dispersions of almost any other function of the states.

1. The covariance matrix is symmetric, so that

$$P = P^T . \quad (2.45)$$

Thus, for a system with n nominal states, the covariance matrix only has $\frac{n(n+1)}{2}$ unique elements. Augmenting the optimal control problem with the covariance matrix requires only $\frac{n(n+1)}{2}$ additional states, differential constraints, and costates, while the sensitivity matrix augmentation requires an additional n^2 .

2. The covariance matrix is positive semi-definite, so that

$$P \geq 0 , \quad (2.46)$$

which implies several others, e.g., all eigenvalues of a covariance matrix are greater than or equal to zero.

3. The diagonal terms of the matrix are the variances for each state, and the off-diagonal terms are products of the standard deviations and correlation coefficients for each state combination, so that

$$P = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \cdots & \rho_{1n}\sigma_1\sigma_n \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \cdots & \rho_{2n}\sigma_2\sigma_n \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n}\sigma_1\sigma_n & \rho_{2n}\sigma_2\sigma_n & \cdots & \sigma_n^2 \end{bmatrix} . \quad (2.47)$$

For all state pairs, $-1 \leq \rho_{ij} \leq 1$, where $\rho_{ij} = 0$ denotes uncorrelated states, and $|\rho_{ij}| = 1$ denotes perfectly linearly dependent states. Knowledge of this property allows the trajectory designer to directly calculate variances and correlation coefficients from a covariance matrix, and use them within cost metrics and problem constraints.

4. The covariance of any arbitrary, smooth function $\mathbf{h}(\mathbf{x})$ can be calculated at any time with the linear transformation

$$P_{\mathbf{h}} = APA^T \quad (2.48)$$

applied to the covariance of the original states. A is calculated as

$$A = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\bar{\mathbf{x}}(t)} . \quad (2.49)$$

A must be continuous with a continuous inverse and must not be an explicit function of time [18]. Using this transformation, the designer can apply penalties and constraints on the covariance of almost any function of the states without the need to reformulate the dynamics and Riccati equation in terms of the new variables.

2.3.2.1 Proof of Covariance Transformation Equivalence

The validity of the linear covariance transformation can be proven by demonstrating its equivalence to the direct integration of the Riccati equation in terms of the new variables. An explanation of Zimmer's proof [18] is provided here.

Since the new variables $\mathbf{h}(\mathbf{x})$ are functions of the states, their dynamics can be related to the state dynamics through the linear transformation

$$\dot{\mathbf{h}} = \mathbf{f}(\mathbf{h}) = A\dot{\mathbf{x}}, \quad (2.50)$$

with A defined in Eq. (2.49). It is important to note that despite the change of variables, the underlying system remains unchanged. The states provide one description of the system behavior, and the new variables simply capture an alternative perspective. Since the transformation of the dynamics is linear, the AWGN uncertainty model is preserved and the stochastic dynamics of the new variables can be expressed

$$\dot{\mathbf{h}} = \mathbf{f}(\mathbf{h}) + \mathbf{w}_{\mathbf{h}}, \quad (2.51)$$

where $\mathbf{w}_{\mathbf{h}}$ is the random variable accounting for process noise in the deterministic \mathbf{h} dynamics, with the statistical properties

$$E[\mathbf{w}_{\mathbf{h}}(t)] = 0 \quad (2.52)$$

and

$$E[\mathbf{w}_{\mathbf{h}}(t) \mathbf{w}_{\mathbf{h}}^T(\tau)] = Q_{\mathbf{h}}(t) \delta(t - \tau). \quad (2.53)$$

It is important to note that the process noise acting on the transformed dynamics differs from the noise acting on the state dynamics. The Riccati equation for the covariance of \mathbf{h} can now be written as

$$\dot{P}_{\mathbf{h}} = FP_{\mathbf{h}} + P_{\mathbf{h}}F^T + Q_{\mathbf{h}} - P_{\mathbf{h}}M_{\mathbf{h}}^TR^{-1}M_{\mathbf{h}}P_{\mathbf{h}}. \quad (2.54)$$

The Riccati equation governing the covariance of \mathbf{h} differs from the Riccati equation governing the state covariance given in three important ways. First, the Jacobian G of the state dynamics has been replaced by the Jacobian F of the \mathbf{h} dynamics. Second, the Q matrix characterizing process noise in the state dynamics is replaced by the $Q_{\mathbf{h}}$ matrix characterizing process noise in the \mathbf{h} dynamics. Third, the M matrix representing the linear transformation of the measurement quantities to equivalent state values is replaced by the $M_{\mathbf{h}}$ matrix, which transforms the same measurements into equivalent \mathbf{h} values. The necessary relationships between the G and F matrices, Q and $Q_{\mathbf{h}}$ matrices, and M and $M_{\mathbf{h}}$ matrices for establishing equivalence can

be determined by finding the time derivative of the covariance transformation and equating it to the new Riccati equation.

The time derivative of the covariance transformation is calculated by applying the product rule to Eq. (2.48), which gives

$$\dot{P}_{\mathbf{h}} = \dot{A}PA^T + A\dot{P}A^T + AP\dot{A}^T. \quad (2.55)$$

The inverse of the covariance transformation provides

$$P = A^{-1}P_{\mathbf{h}}A^{-T}. \quad (2.56)$$

Substituting Eq. (2.43) for \dot{P} and Eq. (2.56) for P and collecting terms appropriately yields

$$\dot{P}_{\mathbf{h}} = (\dot{A}A^{-1} + AGA^{-1})P_{\mathbf{h}} + P_{\mathbf{h}}(\dot{A}A^{-1} + AGA^{-1})^T + AQA^T - P_{\mathbf{h}}A^{-T}M^TR^{-1}MA^{-1}P_{\mathbf{h}}. \quad (2.57)$$

Comparing this expression to Eq. (2.54) shows that the covariance of \mathbf{h} provided by the transformation is equivalent to the covariance of \mathbf{h} provided by the integration of the Riccati equation if

$$F = AGA^{-1} + \dot{A}A^{-1}, \quad (2.58)$$

$$Q_{\mathbf{h}} = AQA^T, \quad (2.59)$$

and

$$M_{\mathbf{h}} = MA^{-1}. \quad (2.60)$$

Proofs for each of these expressions are now sketched:

1) Proof of Eq. (2.58) begins by defining the F Jacobian as the matrix of partial derivatives

$$F = \frac{\partial \dot{\mathbf{h}}}{\partial \mathbf{h}}. \quad (2.61)$$

The linear transformation $\dot{\mathbf{h}} = A\dot{\mathbf{x}}$ allows

$$F = \frac{\partial (A\dot{\mathbf{x}})}{\partial \mathbf{h}}, \quad (2.62)$$

and applying the chain rule leads to

$$\frac{\partial (A\dot{\mathbf{x}})}{\partial \mathbf{h}} = \frac{\partial (A\dot{\mathbf{x}})}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{h}} = \frac{\partial (A\dot{\mathbf{x}})}{\partial \mathbf{x}} A^{-1}. \quad (2.63)$$

The product rule then guarantees that

$$\frac{\partial (A\dot{\mathbf{x}})}{\partial \mathbf{x}} A^{-1} = \frac{\partial A}{\partial \mathbf{x}} \dot{\mathbf{x}} A^{-1} + A \frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}} A^{-1}. \quad (2.64)$$

The partial derivative $\frac{\partial \dot{\mathbf{x}}}{\partial \mathbf{x}}$ is already known to be the Jacobian G of the state dynamics.

If A is not an explicit function of time, so that

$$\dot{A} = \frac{\partial A}{\partial \mathbf{x}} \dot{\mathbf{x}}, \quad (2.65)$$

it is finally shown that

$$F = \dot{A}A^{-1} + AGA^{-1}. \quad (2.66)$$

to satisfy the first condition for equivalence.

2) Proof of Eq. (2.59) begins by applying the linear transformation of Eq. (2.50) directly to the stochastic state dynamics to show that

$$\dot{\mathbf{h}} = A\dot{\mathbf{x}} = A(\mathbf{g}(\mathbf{x}(t), t) + \mathbf{w}) = A\mathbf{g}(\mathbf{x}(t), t) + A\mathbf{w}. \quad (2.67)$$

The stochastic dynamics for the new variables have already been established in Eq. (2.51), and for these two expressions to be equal it must be true that

$$\mathbf{w}_{\mathbf{h}} = A\mathbf{w}. \quad (2.68)$$

Substituting this into Eq. (2.53) shows that the spectral density of the process noise in the \mathbf{h} dynamics can be written

$$Q_{\mathbf{h}} = E[A\mathbf{w}\mathbf{w}^T A^T] = AQA^T \quad (2.69)$$

to satisfy the second condition for equivalence.

3) The proof of Eq. (2.60) begins by recognizing that a change of variables does not change the quantities that are actually measured by any system observers, so the function

$$\mathbf{y} = \mathbf{m}(\mathbf{x}, t)$$

describing the measurement model in terms of the states is still valid. However, instead of requiring the linear transformation

$$M = \frac{\partial \mathbf{m}(\mathbf{x}, t)}{\partial \mathbf{x}},$$

which converts the measured quantities into equivalent state values, the new Riccati equation requires the linear transformation

$$M_{\mathbf{h}} = \frac{\partial \mathbf{m}(\mathbf{x})}{\partial \mathbf{h}}, \quad (2.70)$$

which converts the measured quantities into equivalent values of \mathbf{h} . Applying the chain rule shows that

$$M_{\mathbf{h}} = \frac{\partial \mathbf{m}(\mathbf{x})}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{h}} = MA^{-1} \quad (2.71)$$

to satisfy the third and final condition for equivalence.

2.3.3 Covariance Problem Augmentation

Like before, the nominal trajectory optimization problem can only include functions of the *nominal* states and controls within the cost metric and problem constraints. Penalizing and constraining functions of the covariance matrix can be accomplished by expanding the problem state vector to include each unique element of the covariance matrix P and augmenting the differential constraints to include the Riccati equation. Since the covariance matrix is symmetric, it is fully defined by the elements in its upper triangle. Thus, for a system with n nominal states, the expanded state vector becomes

$$\mathbf{x}_P = \begin{bmatrix} \mathbf{x} \\ \text{symvec}(P) \end{bmatrix}, \quad (2.72)$$

and the augmented set of differential constraints becomes

$$\dot{\mathbf{x}}_P = \begin{bmatrix} \dot{\mathbf{x}} \\ \text{symvec}(\dot{P}) \end{bmatrix}, \quad (2.73)$$

where symvec is simply a function that reshapes an $n \times n$ matrix into a column vector with $\frac{n(n+1)}{2}$ rows, with each row representing an element of the upper triangle. The augmented optimization problem contains $\frac{n(n+1)}{2}$ additional states with an equal number of new differential constraints and costates. Numerically solving the augmented problem requires evaluating the covariance dynamics, which in turn requires linearizing the system about the nominal trajectory. As in the sensitivity dynamics, the linearization can be done analytically by deriving and evaluating the partial derivatives that comprise the Jacobian or numerically with automatic differentiation software or finite differencing. Either way, including the covariance dynamics significantly increases the problem complexity, even without optimizing feedback gains simultaneously. Regardless of the control law, the expanded problem formulation allows the cost metric, event constraints, path constraints, and state constraints to include functions of the covariance matrix as well as the nominal states, so they can be written as

$$J_P = U(\mathbf{x}_P(t_0), \mathbf{x}_P(t_f), t_0, t_f) + \int_{t_0}^{t_f} V(\mathbf{x}_P(t), \mathbf{u}(t), t) dt, \quad (2.74)$$

$$\mathbf{e}_P^L \leq \mathbf{e}_P(\mathbf{x}_P(t_0), \mathbf{x}_P(t_f), t_0, t_f) \leq \mathbf{e}_P^U, \quad (2.75)$$

$$\mathbf{d}_P^L \leq \mathbf{d}_P(\mathbf{x}_P(t), \mathbf{u}(t), t) \leq \mathbf{d}_P^U, \quad (2.76)$$

and

$$\mathbf{x}_P^L \leq \mathbf{x}_P(t) \leq \mathbf{x}_P^U, \quad (2.77)$$

respectively. The designer's primary task then becomes determining the most important or influential uncertainties in the system, selecting appropriate P_0 , Q , and R matrices to model those uncertainties, choosing functions of the nominal states in which to reduce the covariance, and using the mathematical properties of the covari-

ance matrix to formulate cost metrics to match the design goals.

2.4 Parametric Uncertainties

As formulated above, the sensitivity and covariance matrices account for the effects of state uncertainties, but uncertainties are often more accurately attributed to other parameters within the system dynamics. For example, in a rocket ascent problem with atmospheric drag, it may be desirable to account for the uncertainty associated specifically to atmospheric density. However, the sensitivity matrix can only provide information on system responses to perturbations in each of the *states* along the nominal trajectory – not perturbations in other parameters. If atmospheric density is only a parameter within the dynamics, then the sensitivity matrix cannot directly address any uncertainty in atmospheric density. Likewise, the covariance matrix provides expected *state* dispersions. Even if the nonlinear noise model specified in Eq. (2.29) is used to account for nonlinear parametric process noise, other types of uncertainties such as biases and initialization errors can only be assigned as *state* dispersions – not dispersions in other parameters. In the augmented problem formulations, the designer could attempt to couch parametric uncertainties in terms of state uncertainties, but the transformation of a parametric uncertainty to a state uncertainty can be difficult to express accurately across the entire state space of a nonlinear system, making only crude approximations of the desired uncertainty models possible.

One method to circumvent this problem is to update the status of uncertain constant parameters to that of states with trivial dynamics [15]. For a constant parameter p within the system dynamics, p can be assigned its own dynamics

$$\dot{p} = 0 \tag{2.78}$$

with initial condition

$$p(t_0) = p_0, \tag{2.79}$$

where p_0 is the nominal value of the parameter used in the original state dynamics. The parameter p is then augmented to the state vector

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x} \\ p \end{bmatrix} \tag{2.80}$$

and the state dynamics

$$\dot{\mathbf{x}}_a = \mathbf{g}_a(\mathbf{x}(t), \mathbf{u}(\mathbf{x}, t), t) = \begin{bmatrix} \mathbf{g}(\mathbf{x}(t), \mathbf{u}(\mathbf{x}, t), t) \\ \dot{p} \end{bmatrix}. \tag{2.81}$$

For both the sensitivity and covariance dynamics, the Jacobian must be recalculated as

$$G_a = \left. \frac{\partial \mathbf{g}_a(\mathbf{x}_a, \mathbf{u}, t)}{\partial \mathbf{x}_a} \right|_{\mathbf{x}_a = \bar{\mathbf{x}}_a(t), \mathbf{u} = \bar{\mathbf{u}}(\bar{\mathbf{x}}_a, t)} = \left[\frac{\partial \mathbf{g}_a}{\partial \mathbf{x}_a} + \frac{\partial \mathbf{g}_a}{\partial \mathbf{u}} K \right] \Big|_{\mathbf{x}_a = \bar{\mathbf{x}}_a(t), \mathbf{u} = \bar{\mathbf{u}}(\bar{\mathbf{x}}_a, t)} \tag{2.82}$$

to include the additional state. Now the sensitivity matrix also includes sensitivities of each original state to perturbations in the parameter p , which can be included in the cost metric to specifically address uncertainties in that parameter. Likewise, the covariance matrix now accounts for expected dispersions in p and any correlations to dispersions in the original states. Due to its trivial dynamics, including p in the state vector does not alter the nominal system dynamics, so it does not need to be directly solved in the optimization process.

2.5 Mathematical Link

Although the physical and mathematical meanings of the sensitivity and covariance matrices differ, the augmented problem formulations use them to accomplish the same task, which is to generate trajectories that are more robust to system uncertainties. Both perspectives on addressing the problem – reducing sensitivities and reducing expected dispersions – are closely connected because one implies the other. Both intuitively and mathematically, a reference trajectory that is less sensitive to a particular uncertainty deviates less from its design when those uncertainties are realized, and smaller deviations from the nominal trajectory by definition means smaller covariances. The mathematical connection is ultimately rooted in the link between the sensitivity matrix and covariance matrix dynamics. Recalling that the state transition matrix and sensitivity matrix are equivalent, a general solution for the covariance matrix of a continuous system subject to AWGN without measurements can be written as an explicit function of the state transition matrix as

$$P(t) = \Phi(t, t_0) P(t_0) \Phi(t, t_0)^T + \int_{t_0}^t \Phi(t, \tau) Q(\tau) \Phi(t, \tau)^T d\tau. \quad (2.83)$$

Thus, in the absence of measurement updates, propagating the state transition matrix and using it to evaluate Eq. (2.83) is equivalent to directly integrating the Lyapunov equation. In fact, the Lyapunov equation can be obtained directly from Eq. (2.83).

First the covariance given by Eq. (2.83) is rewritten as a string of discrete time steps

$$P(t_{k+1}) = \Phi(t_{k+1}, t_k) P(t_k) \Phi(t_{k+1}, t_k)^T + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) Q(\tau) \Phi(t_{k+1}, \tau)^T d\tau. \quad (2.84)$$

Next, a Taylor series expansion is applied to the state transition matrix. The Taylor series expansion of any function $f(x)$ about an operating point a is defined as

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n. \quad (2.85)$$

With an infinite number of terms, the full Taylor series provides an exact solution for

the function $f(x)$ for all x within the radius of convergence of the series. However, in regions near the operating point, fewer terms are usually sufficient to approximate the solution, and if the region surrounding the operating point shrinks infinitesimally, the linear approximation provided by the first-order expansion becomes an exact solution. The first-order Taylor series expansion of the state transition matrix about any operating point t_k is written as

$$\Phi(t, t_k) = \Phi(t_k, t_k) + \dot{\Phi}(t_k, t_k)(t - t_k) . \quad (2.86)$$

Using the properties and dynamics of the state transition matrix, Eq. (2.86) can be rewritten as

$$\Phi(t, t_k) = I + G(t_k)(t - t_k) . \quad (2.87)$$

Applying a first-order expansion of the state transition matrix at each discrete time interval $\Delta t = t_{k+1} - t_k$ gives

$$\Phi(t_{k+1}, t_k) = I + G(t_k) \Delta t \quad (2.88)$$

for any k , which can then be substituted back into Eq. (2.84) and expanded to give

$$P(t_{k+1}) = P(t_k) + G(t_k)P(t_k)\Delta t + P(t_k)G(t_k)^T\Delta t + G(t_k)P(t_k)G(t_k)^T\Delta t^2 + Q_k . \quad (2.89)$$

With the intention of shrinking Δt infinitesimally, the second-order term can be ignored and Eq. (2.89) can be rewritten as

$$\frac{P(t_{k+1}) - P(t_k)}{\Delta t} = G(t_k)P(t_k) + P(t_k)G(t_k) + \frac{Q_k}{\Delta t} . \quad (2.90)$$

When Δt approaches zero, the left-hand side of the equation becomes the time derivative of the covariance, giving the Lyapunov equation

$$\lim_{\Delta t \rightarrow 0} \frac{P(t_{k+1}) - P(t_k)}{\Delta t} = \dot{P}(t) = G(t)P(t) + P(t)G(t) + Q(t) . \quad (2.91)$$

The leap to the Riccati equation depends on a similar derivation using measurement updates and extended Kalman filtering techniques, but the nature of the connection between the state transition matrices and the covariance dynamics is established without measurements.

As shown by Eq. (2.83), the covariance matrix propagates according to the state transition matrix, and according to Eq. (2.15), the state transition matrix (sensitivity matrix) propagates according to the linearized system dynamics and the particular trajectory about which they are linearized. Thus, shaping a reference trajectory shapes the propagation of its associated sensitivity matrix, which in turn shapes the propagation of the linear covariance.

Chapter 3

Zermelo Boat Problem

Although it is clear from Chapter 2 that sensitivity and covariance are intimately related, the trajectory-shaping techniques contain significant differences in implementation. Although their mathematical connection makes it likely, it is unclear whether or not penalties and constraints on sensitivity and covariance can actually produce equivalent results for common design goals and uncertainty models. Zimmer claims that the sensitivity matrix cannot be used to address process noise [18], but he does not provide direct evidence to support this claim. Using a simple two-dimensional problem, this chapter directly tests the ability of both methods to achieve equivalent results in both open-loop and closed-loop environments for any system with state and parametric process noise, parametric biases, and state initialization errors. The results of this test are used to identify the strengths, weaknesses, capabilities, and limitations of each method and justify a preferred technique.

The methods are compared using a common design goal intended to improve confidence in the final states at a fixed target. The test problem is augmented using both the sensitivity and covariance matrices, and cost metrics are developed in both realms to achieve the common design goal for each posed uncertainty model. The first three uncertainty scenarios utilize parametric uncertainties only, and the trajectory-shaping methods are compared with both open-loop and fixed-gain feedback control. The remaining uncertainty scenario includes a comprehensive combination of state and parametric uncertainties, but is addressed using fixed-gain feedback only, because the first three cases sufficiently establish the method's versatility under various feedback scenarios.

Following the fixed-target cases, a comparison of both methods is used to justify a preferred technique, which is then applied to a second design goal crafted to more accurately reflect the maximum capabilities of the system with open terminal constraints. Cost functions and constraints are developed to maximize a function of the open terminal states while balancing improvement in their confidence.

Finally, the fixed-target scenario is revisited to demonstrate simultaneous optimization of linear feedback control schemes and reference trajectories. Constraints are applied to closed-loop response characteristics to ensure realistic feedback commands are generated. The problem is solved assuming constant gains, but the extension to the variable gain problem becomes apparent.

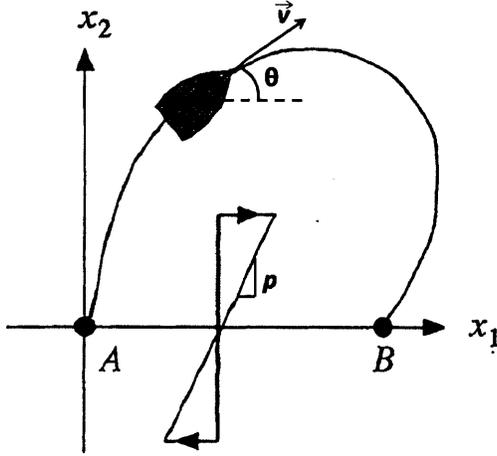


Figure 3-1: Zermelo Problem Physical Illustration [15]

3.1 Zermelo Problem Setup

The same problem used by Kumar and Seywald [15] to introduce desensitized trajectory optimization is adopted here to allow verification of the current sensitivity formulation against their original results before extending the test cases. The system dynamics are meant to simulate a typical Zermelo boat problem, in which a boat begins at a designated point along the shore, travels along a flowing river, and returns to shore at another location. The boat's two-dimensional down-stream and cross-stream position is defined by the $x_1 - x_2$ plane, with the shore given by the x_1 axis. The boat moves at a constant relative speed with unity magnitude at an angle θ measured counterclockwise from the down-stream ($+x_1$) direction. The angle θ is the boat's single control variable in this problem, and the total travel time is fixed to unity for all cases investigated in this chapter. Although the down-stream (x_1) axis simulates a shoreline, the cross-stream position x_2 is not constrained to be positive, which allows the boat to actually travel through the virtual shoreline. Kumar and Seywald allow this behavior in their problem setup, and it is adopted here for the sake of consistency. The nominal strength of the water current increases linearly away from shore, where the slope of this increase is the constant parameter

$$p = 10. \tag{3.1}$$

Figure 3-1 provides a visual representation of the problem setup. The Zermelo dynamics can be written as

$$\begin{aligned} \dot{x}_1 &= \cos(\theta) + px_2, \\ \dot{x}_2 &= \sin(\theta), \end{aligned} \tag{3.2}$$

subject to the conditions

$$x_1(0) = 0, \quad x_2(0) = 0, \quad \text{and} \quad t_f = 1. \quad (3.3)$$

It is assumed, however, that the current strength p is not known precisely, and may even change during the boat's trip. Since p is a constant parameter but not a state within the original system dynamics, its status can be updated to that of a state with the trivial dynamics

$$\dot{p} = 0 \quad (3.4)$$

and the initial condition

$$p(0) = 10.$$

The full state vector for the problem then becomes

$$\mathbf{x}_a = \begin{bmatrix} x_1 \\ x_2 \\ p \end{bmatrix}, \quad (3.5)$$

which allows for the sensitivity and covariance matrices to account for parametric uncertainties in p as well as the original states x_1 and x_2 . Feedback control can also be applied to dispersions about the boat's nominal position trajectory by using the gain matrix

$$K = [K_1 \quad K_2]. \quad (3.6)$$

As noted in Section 2.2.3 of this thesis, the gains within the sensitivity and covariance dynamics can either be 1) set to zero to characterize open-loop responses, 2) predefined to characterize closed-loop responses with a specific controller, or 3) optimized as part of the trajectory design process. The Jacobian of the augmented system with feedback control is derived by taking the partial derivatives of the dynamics with respect to each state, yielding

$$G_c = \left[\begin{array}{ccc} -K_1 \sin(u) & -K_2 \sin(u) + p & x_2 \\ K_1 \cos(u) & K_2 \cos(u) & 0 \\ 0 & 0 & 0 \end{array} \right] \Bigg|_{\mathbf{x}=\bar{\mathbf{x}}(t), \mathbf{u}=\bar{\mathbf{u}}(\bar{\mathbf{x}}, t)}. \quad (3.7)$$

The sensitivity matrix dynamics are then written as

$$\dot{S} = G_c S$$

with the initial condition

$$S(t_0) = I,$$

and the covariance matrix dynamics without measurements are written as

$$\dot{P} = G_c P + P G_c^T + Q$$

with the initial condition

$$P(t_0) = P_0.$$

Both the sensitivity and covariance matrices are 3×3 in dimension.

3.2 Fixed-Target Precision

To compare the abilities of the sensitivity and covariance trajectory-shaping methods to improve confidence in the final states at a fixed target, a common design goal must be specified so that any solution can be compared against a single benchmark. For this case, given a specific target location and uncertainty model, the goal is to find the trajectory that minimizes the unweighted sum of the variances of the boat's final expected position dispersions.

Kumar and Seywald show that for fixed unity time, the boat can reach a maximum down-stream point along the shore of approximately 2.8 units from the launch point when uncertainty considerations are ignored [15]. Choosing the target location to be 2.25 units downshore of the launch point – well within the maximum capability of the boat – the terminal constraints are written as

$$x_1(t_f) = 2.25 \quad \text{and} \quad x_2(t_f) = 0. \quad (3.8)$$

To test the ability of both methods to address various types of uncertainty, the following four models are considered:

1. Constant parametric bias
2. Parametric process noise
3. Parametric bias and process noise combination
4. State insertion error, state process noise, parametric bias, and parametric process noise combination

To provide a basis from which to judge the improvement achieved by the sensitivity and covariance trajectory-shaping methods, an attempt is first made to achieve the design goal using cost metrics that are only functions of the nominal states and controls. Although such metrics do not directly take into account system uncertainties, the responses of the solutions to various uncertainty models can still be evaluated *ex post facto* because nominal trajectories are always dynamically independent from the sensitivity and covariance dynamics. This independence exists because the dynamics of the original states *are not* functions of the sensitivity or covariance, even though the sensitivity and covariance *are* functions of the original states.

For the nominal fixed-target trajectory design problem, the Zermelo dynamic and initial constraints are completely defined by Eqs. (3.2) and (3.3). Without sensitivity or covariance penalties, the designer can only attempt to infer which nominal metrics may lead to reduced position dispersions for each particular uncertainty model. In this case, due to the linear relationship of p with cross-stream distance, the current has a greater impact on the boat's down-stream rate as the boat travels away from the shore, so the effects of p on the boat's trajectory become magnified as x_2 increases.

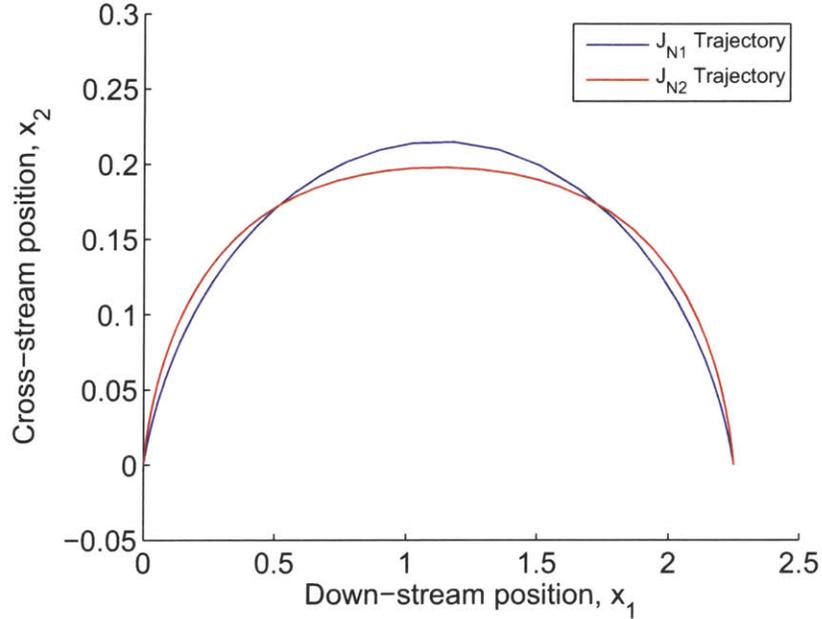


Figure 3-2: Optimal trajectories for nominal cost metrics. The J_{N1} trajectory optimizes the integral of the cross-stream distance, and the J_{N2} trajectory optimizes the integral of the square of the cross-stream distance.

Thus, it seems reasonable to infer that for any of the uncertainty models listed above, trajectories which stay closer to the shore accumulate smaller errors. Two possible metrics, then, would be the integrated cross-stream distance and the integrated square of the cross-stream distance along the trajectory. Mathematically, these metrics can be written as

$$J_{N1} = \int_0^{t_f} x_2(t) dt \quad (3.9)$$

and

$$J_{N2} = \int_0^{t_f} x_2^2(t) dt. \quad (3.10)$$

As shown in Fig. 3-2 greater penalties on cross-stream distance result in optimal solutions that stay closer to shore.

3.2.1 Parametric Uncertainties

3.2.1.1 Constant Bias

The first uncertainty scenario to be investigated assumes that p maintains a constant value, but is not precisely known. The trivial dynamics of p account for its constancy, but uncertainty in this value translates to uncertainty in the initial condition p_0 . In

other words, p has a constant bias. Otherwise the state dynamics are assumed to be ideal.

The sensitivity method is applied to this scenario by first finding the most appropriate sensitivity metric. For any problem, the most appropriate sensitivity metric links the sensitivity of the design goal to perturbations in the uncertain states and parameters. In this case, the design goal seeks to minimize a function of the final position dispersions, and the only uncertainty exists with the initial condition p_0 , so the best sensitivity metric minimizes a function of the sensitivities of the final states $x_1(t_f)$ and $x_2(t_f)$ to perturbations in the initial value p_0 . The sensitivity matrix for the Zermelo dynamics can be expanded as

$$S(t) = \frac{\partial \bar{\mathbf{x}}(t)}{\partial \mathbf{x}_0} = \begin{bmatrix} \frac{\partial x_1(t)}{\partial x_1(t_0)} & \frac{\partial x_1(t)}{\partial x_2(t_0)} & \frac{\partial x_1(t)}{\partial p(t_0)} \\ \frac{\partial x_2(t)}{\partial x_1(t_0)} & \frac{\partial x_2(t)}{\partial x_2(t_0)} & \frac{\partial x_2(t)}{\partial p(t_0)} \\ \frac{\partial x_3(t)}{\partial x_1(t_0)} & \frac{\partial x_3(t)}{\partial x_2(t_0)} & \frac{\partial x_3(t)}{\partial p(t_0)} \end{bmatrix}, \quad (3.11)$$

and the desired sensitivities for this case are found by simply evaluating the sensitivity matrix at the final time and picking off the (1, 3) and (2, 3) elements.

More specifically, since the design goal seeks to minimize the *unweighted sum* of the final position *variances*, and since the final position variances are squares of the final position standard deviations, the sensitivity metric should penalize the *unweighted sum of the squares* of these sensitivity terms. This metric can be expressed mathematically as the endpoint cost

$$J_{S1} = \left(\frac{\partial x_1(t_f)}{\partial p(t_0)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial p(t_0)} \right)^2. \quad (3.12)$$

Although this sensitivity metric was crafted to address a particular uncertainty model, it is important to recall that statistical representations of the uncertainties do not appear anywhere in its evaluation. Thus, for a given set of nominal state dynamics and a given feedback control scheme, this metric produces a unique trajectory that does not change unless the metric is reformulated to address a different uncertainty model.

The covariance method is applied to this scenario by first translating the uncertainty model into appropriate initial covariance and process noise matrices. Since the only initial condition uncertainty exists with p_0 , and since no uncertainties act along the rest of the trajectory, this model can be expressed in the covariance realm as

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_p \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.13)$$

where c_p is a positive constant selected to represent the variance of the assumed uncertainty in the initial condition p_0 .

Next, the most appropriate covariance metric is developed for the current design goal and uncertainty model. The covariance matrix contains the variances of each

state along its diagonal, written as

$$P = \begin{bmatrix} \sigma_{x_1}^2 & \rho_{x_1x_2}\sigma_{x_1}\sigma_{x_2} & \rho_{x_1p}\sigma_{x_1}\sigma_p \\ \rho_{x_1x_2}\sigma_{x_1}\sigma_{x_2} & \sigma_{x_2}^2 & \rho_{x_2p}\sigma_{x_2}\sigma_p \\ \rho_{x_1p}\sigma_{x_1}\sigma_p & \rho_{x_2p}\sigma_{x_2}\sigma_p & \sigma_p^2 \end{bmatrix},$$

and since the design goal seeks to minimize the final position variances, the most appropriate cost metric simply sums the first two diagonal elements of the final covariance matrix. This metric can be expressed as the endpoint cost

$$J_{P1} = \sigma_{x_1}^2(t_f) + \sigma_{x_2}^2(t_f). \quad (3.14)$$

It is important to note that unlike the sensitivity metric, this metric *does* explicitly feature a statistical representation of the uncertainty values in its evaluation. Any changes in the uncertainty model will be reflected as changes in P_0 and Q , which alters the covariance solution to the Riccati equation. Unlike the sensitivity metric, the covariance metric produces different results for different uncertainty models.

Applying J_{S1} as the cost metric with zero feedback gains, the resulting optimal trajectory matches the solution associated with the J_{N1} integrated cross-stream metric. Similarly, applying J_{P1} as the cost metric with zero feedback gains and the P_0 and Q matrices specified in Eq. (3.13) also results in the same optimal trajectory, no matter what magnitude of c_p is selected to represent the bias. The magnitude of c_p does not affect the optimal solution in this scenario because there are no other uncertainties acting along the trajectory, and a shift in the value of c_p simply scales the relative magnitude of the covariance profile without altering its shape. Choosing $c_p = 1$, the open-loop covariance dynamics are propagated along the optimal open-loop and the optimal J_{N2} trajectory to verify that the optimal open-loop trajectory does in fact provide a smaller final position variance.

Although it is now known which trajectory optimizes the open-loop design goal, it is not immediately evident whether the same trajectory also optimizes the design goal in the presence of feedback control. When the closed-loop covariance is propagated along both the optimal open-loop and the J_{N2} trajectory with the unity gain matrix

$$K = \begin{bmatrix} -1 & -1 \end{bmatrix}, \quad (3.15)$$

the optimal open-loop trajectory does in fact accumulate smaller final position variances than J_{N2} . However, it is still possible that another trajectory exists with even smaller unity-gain-matrix variances. To test this possibility, the trajectories are reshaped using the J_{S1} sensitivity metric and the J_{P1} covariance metric with the same P_0 and Q matrices, but this time with the unity gain matrix included in the sensitivity and covariance problem dynamics to account for the controller's presence. Both methods again converge to an identical solution, but one that is different from the optimal open-loop trajectory. Figure 3-3 plots the optimal open-loop, unity-gain-matrix, and J_{N2} trajectories.

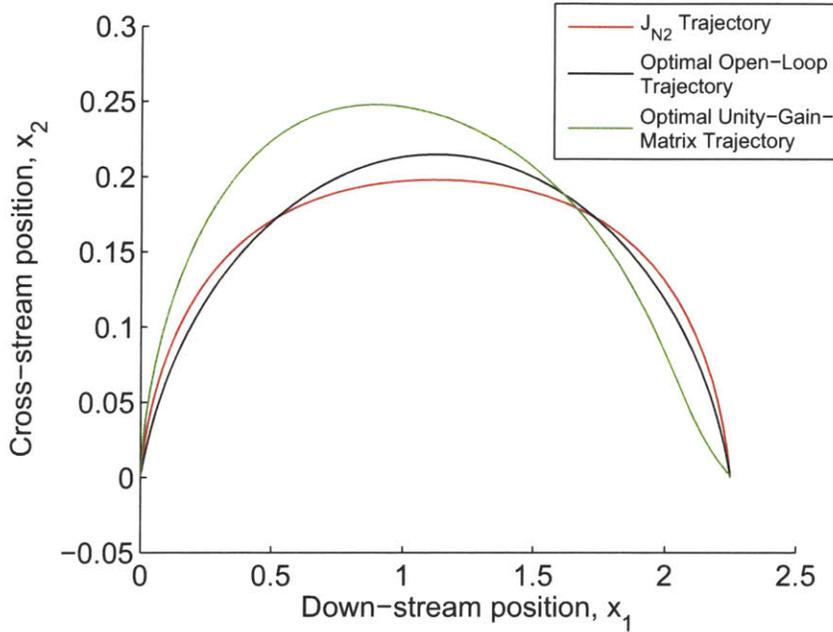


Figure 3-3: Optimal trajectories with a constant parametric bias. The optimal open-loop trajectory optimizes the final open-loop position variance sum, and the optimal unity-gain-matrix trajectory optimizes the final closed-loop position variance sum with unity-gain-matrix feedback. The J_{N2} trajectory optimizes the integral of the square of the cross-stream distance, as defined before.

Figure 3-4 shows the position variance sums along each trajectory when the covariance is propagated with the parametric bias using both zero feedback and unity-gain-matrix feedback. Table 3.1 lists the improvements in open-loop and closed-loop variance achieved by the trajectory-shaping method. The data show several interesting results.

First, as expected, feedback control alone dramatically improves the position variances along each of the individual trajectories. Along each trajectory, simply closing the loop with the unity gain matrix improves the final position variance sum by over 92%.

Second, the improvements from trajectory-shaping appear more dramatic in a closed-loop environment. Trajectory-shaping improves final unity-gain-matrix position variance by 29%, while it only improves final open-loop position variance by 1.66%.

Third, the trajectory that provides the optimal open-loop final position variance does not also provide the optimal variance with unity-gain-matrix control, and vice versa. As expected, in the presence of unity-gain-matrix feedback, the optimal unity-gain-matrix trajectory accumulates smaller errors than both the optimal open-loop and J_{N2} solutions. In fact, the optimal unity-gain-matrix trajectory reaches the target with an improvement in final position variance of more than 29%, while the final variance sums of the optimal open-loop and nominal J_{N2} solutions differ by less than

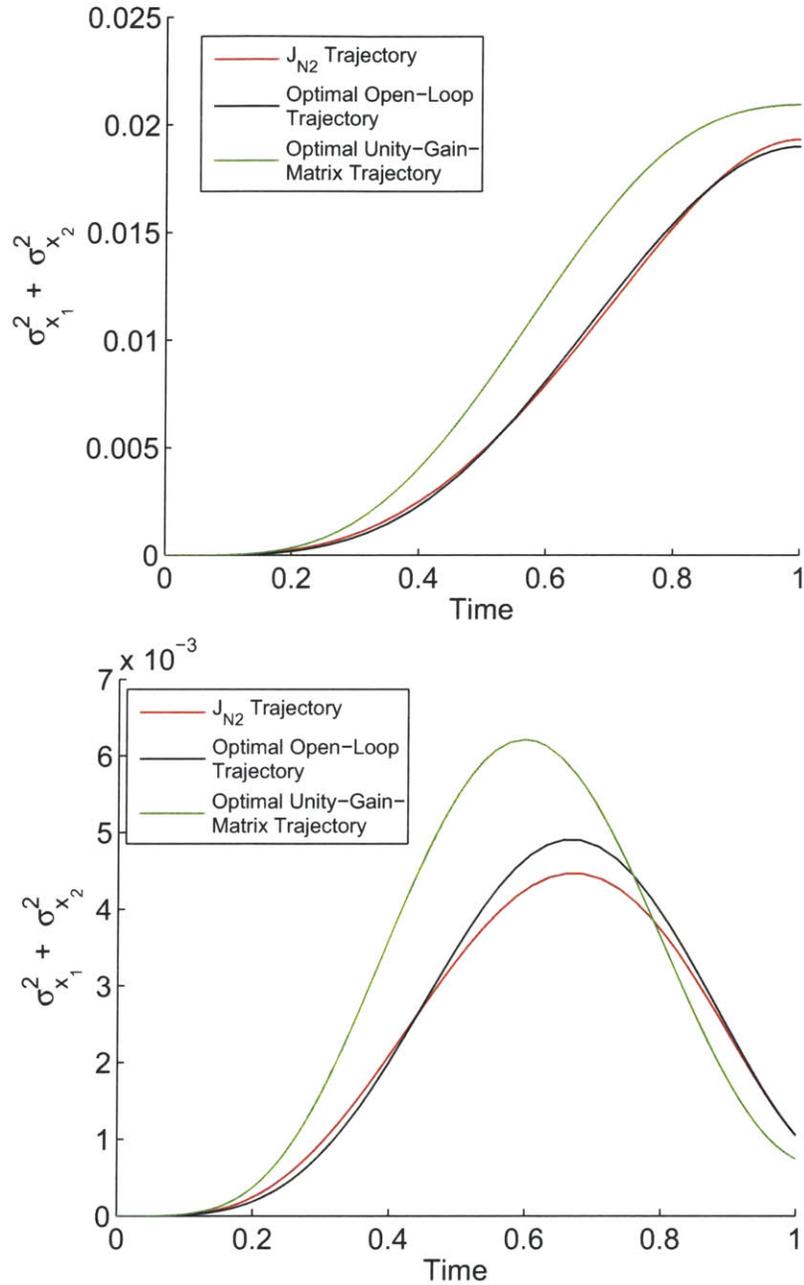


Figure 3-4: Position variance sums propagated along each trajectory given in Fig. 3-3 with a constant parametric bias in which $c_p = 1$. The top figure shows the open-loop propagations, while the bottom figure shows the closed-loop propagations with unity-gain-matrix feedback. As expected, the final open-loop variance is smallest along the optimal open-loop trajectory (given in black), and the final closed-loop variance is smallest along the optimal unity-gain-matrix trajectory (given in green).

Table 3.1: Final position variance sums with a constant parametric bias in which $c_p = 1$. The first column lists the trajectories given in Fig. 3-3, the second column provides their final propagated open-loop position variance sums, and the fourth column displays their final propagated closed-loop position variance sums with unity-gain-matrix feedback. The third and fifth columns show the open-loop and unity-gain-matrix improvement percentages, respectively, using the J_{N2} trajectory as the basis of comparison. Degradations in performance are depicted as negative values, and the optimal results for each column are emphasized in bold.

Trajectory	OL Sum	OL Improvement	CL Sum	CL Improvement
J_{N2}	0.01931	N/A	0.001051	N/A
Optimal Open-Loop	0.01899	1.66%	0.001053	-0.190%
Optimal Unity-Gain-Matrix	0.02093	-8.39%	0.0007461	29.0%

0.2%. Also as expected, when feedback is omitted, the optimal open-loop trajectory accumulates smaller dispersions than both the unity-gain-matrix and nominal J_{N2} trajectories.

However, the optimal open-loop solution appears better suited to handle the addition of feedback than the optimal unity-gain-matrix solution is able to cope with a loss in feedback. Perhaps not as expected, the optimal unity-gain-matrix trajectory actually reaches the target with significantly *worse* precision than even the J_{N2} solution in an open-loop environment. Using the J_{N2} trajectory as the basis of comparison, the optimal open-loop trajectory improves the final open-loop position variance sum by approximately 1.7%, but the optimal unity-gain-matrix trajectory *degrades* the final position variance sum by 8.4%. This scenario provides a preliminary indication that the trajectory-shaping process can be highly dependent on the assumed feedback control scheme — solutions can be optimal for particular feedback laws, but they take full advantage of the controller dynamics to craft trajectories that may not perform as well when the feedback dynamics change significantly. These results appear to demonstrate that the optimal open-loop solution possesses a qualitative edge in robustness over the optimal unity-gain-matrix solution when applied in feedback control environments other than those for which they were specifically designed.

Fourth, these results demonstrate that the sensitivity and covariance methods are both able to achieve equivalent results with appropriate cost metric selection. Specifically, this case showed that when the only uncertainty is a constant parameter bias, minimizing the sum of the final position sensitivities to perturbations in the initial parameter value is equivalent to minimizing the sum of the final position variances.

For the simple open-loop scenario, it proved possible to infer a simple cost metric that achieved the design goal without directly using uncertainty considerations. Since the x_2 dynamics are independent of both x_1 and p , open-loop system errors will only manifest themselves through the px_2 term within the x_1 dynamics, and the uncertainty

in this term will grow as either the uncertainty in p grows or the magnitude of x_2 grows. Minimizing the final open-loop position variance amounts to minimizing the cumulative effect of the uncertainty in px_2 over the entire trajectory, and since p has a *constant* uncertainty, this cumulative effect can be captured purely as the integral of x_2 . Deviations from the optimal integrated cross-stream solution only serve to increase the integrated effect of the uncertainty associated with the px_2 term, which results in greater cumulative position errors.

Nevertheless, the sensitivity and covariance trajectory-shaping methods actually prove the optimality of the J_{N1} trajectory with respect to the open-loop design goal. Without implementing the sensitivity and covariance metrics, this optimality could only have been inferred.

3.2.1.2 Process Noise

The next uncertainty scenario assumes that the value of p is not precisely known and in fact changes over the course of the trajectory. Otherwise the state dynamics are assumed to be ideal. If the mean of these changes is assumed to be uncorrelated in time with zero mean, then the dynamics of p can be modeled stochastically as AWGN. To apply the sensitivity method to this case, the sensitivity metric must be recast to ensure that it makes sense for the new uncertainty type. The sensitivity metric defined for the constant bias scenario does not quite make sense, because it only minimizes the sensitivities of the final position states to perturbations in the *initial* value p_0 . The design goal remains the same, but now uncertainty exists in p over the entire trajectory rather than just its initial value. Instead, the most appropriate metric here should minimize the sensitivities of the final position states to perturbations in *all* values of p along the trajectory. Utilizing the third and fourth properties of the sensitivity matrix defined in Section 2.2.2 of this thesis, this particular sensitivity can be formulated as

$$\frac{\partial \bar{\mathbf{x}}(t_f)}{\partial \mathbf{x}(t)} = \frac{\partial \bar{\mathbf{x}}(t_f)}{\partial \mathbf{x}_0} \frac{\partial \mathbf{x}_0}{\partial \mathbf{x}(t)} = S(t_f) S(t)^{-1}, \quad (3.16)$$

which can be expanded as the matrix of partial derivatives

$$\frac{\partial \bar{\mathbf{x}}(t_f)}{\partial \mathbf{x}(t)} = \begin{bmatrix} \frac{\partial x_1(t_f)}{\partial x_1(t)} & \frac{\partial x_1(t_f)}{\partial x_2(t)} & \frac{\partial x_1(t_f)}{\partial p(t)} \\ \frac{\partial x_2(t_f)}{\partial x_1(t)} & \frac{\partial x_2(t_f)}{\partial x_2(t)} & \frac{\partial x_2(t_f)}{\partial p(t)} \\ \frac{\partial x_3(t_f)}{\partial x_1(t)} & \frac{\partial x_3(t_f)}{\partial x_2(t)} & \frac{\partial x_3(t_f)}{\partial p(t)} \end{bmatrix}, \quad (3.17)$$

in which the elements of interest are the (1,3) and (2,3) elements. Since the design goal is to minimize the unweighted sum of the final position variances, and since the (1,3) and (2,3) elements of the appropriate sensitivity matrix are functions of time along the *entire* trajectory, the metric should be the *integral* of the unweighted sum of the squares of the sensitivities. Also, since the uncertainty in $p(t)$ is assumed to be uncorrelated in time with constant power spectral density, the integral does not need to explicitly weigh perturbations in $p(t)$ more heavily at any particular times.

This metric can be represented as the integral cost

$$J_{S2} = \int_{t_0}^{t_f} \left\{ \left(\frac{\partial x_1(t_f)}{\partial p(t)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial p(t)} \right)^2 \right\} dt. \quad (3.18)$$

Like J_{S1} , this metric will provide a unique trajectory for fixed dynamics and feedback control schemes, unless its structure is altered to reflect changes in the underlying uncertainty model.

To apply the covariance method to this scenario, the uncertainty model must be translated into appropriate initial covariance and process noise matrices. Since the uncertainty in p follows AWGN assumptions with no bias present, this uncertainty model can be expressed in the covariance realm as

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & w_p \end{bmatrix}, \quad (3.19)$$

where w_p is a constant selected to represent the variance of the noise acting on the dynamics of p . Since the design goal has not changed, J_{P1} is still the most appropriate covariance metric for this scenario.

Implementing the J_{S2} sensitivity metric without feedback results in the same optimal trajectory as implementing the J_{P1} covariance metric without feedback, no matter what magnitude of w_p is selected to represent the noise strength in Q . The magnitude of w_p does not affect the optimal solution in this scenario, because there are no other uncertainties acting along the trajectory, and a shift in the value of w_p simply scales the relative magnitude of the covariance profile without altering its shape. Both the sensitivity and covariance methods also lead to identical solutions when the trajectories are reshaped using the unity gain matrix, although this trajectory differs from the optimal open-loop solution. Figure 3-5 shows the optimal open-loop and optimal unity-gain-matrix trajectories for this parametric process noise scenario, along with the optimal J_{N1} and J_{N2} trajectories for comparison.

Arbitrarily choosing $w_p = 1$, both the open-loop and unity-gain-matrix covariance dynamics are propagated along each of the trajectories to verify that the smallest final position variance sums accumulate along the trajectories shaped specifically for each feedback control scheme. Figure 3-6 shows the position variance sums along each trajectory when the covariance is propagated with the parametric process noise using both zero feedback and unity-gain-matrix feedback. Table 3.2 compares the open-loop and closed-loop final variance improvements using the J_{N1} trajectory as the basis of comparison.

For this uncertainty model, the J_{N1} and J_{N2} trajectories did not match either the optimal open-loop or optimal unity-gain-matrix trajectories, demonstrating the value of using the trajectory-shaping techniques to directly improve targeting precision. In both feedback environments, the optimal trajectories traveled farther from shore early in the trajectory in order to stay closer to shore late in the trajectory. This behavior

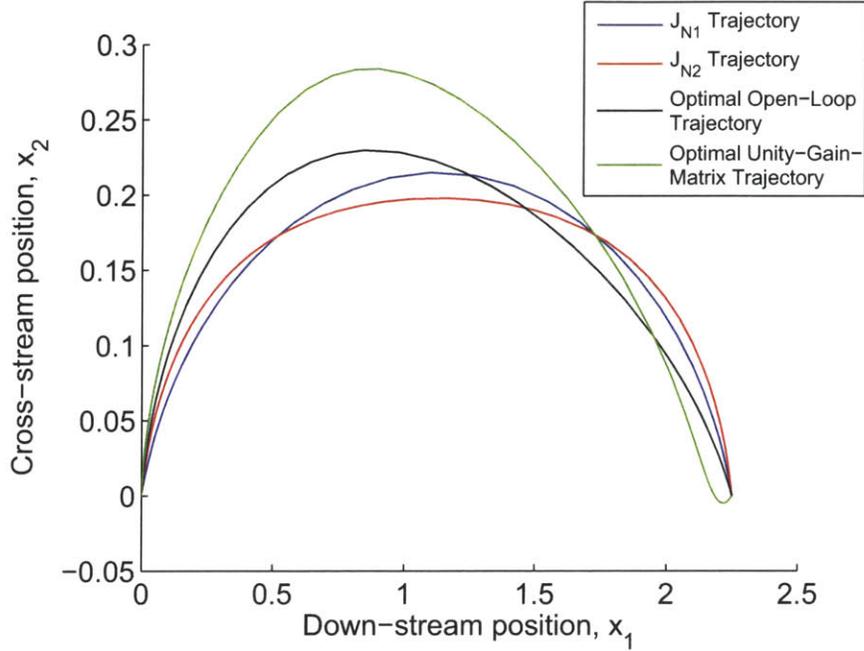


Figure 3-5: Optimal trajectories with parametric process noise. The optimal open-loop trajectory optimizes the final open-loop position variance sum, and the optimal unity-gain-matrix trajectory optimizes the final closed-loop position variance sum with unity-gain-matrix feedback. The J_{N1} and J_{N2} trajectories are unchanged from before.

Table 3.2: Final position variance sums with parametric process noise in which $w_p = 1$. The first column lists the trajectories given in Fig. 3-5, the second column provides their final propagated open-loop position variance sums, and the fourth column displays their final propagated closed-loop position variance sums with unity-gain-matrix feedback. The third and fifth columns show the open-loop and unity-gain-matrix improvement percentages, respectively, using the J_{N1} trajectory as the basis of comparison. Degradations in performance are depicted as negative values, and the optimal results for each column are emphasized in bold.

Trajectory	OL Sum	OL Improvement	CL Sum	CL Improvement
J_{N1}	0.007106	N/A	0.0008787	N/A
J_{N2}	0.007118	-0.169%	0.0009308	-5.93%
Optimal Open-Loop	0.006686	5.91%	0.0006250	28.9%
Optimal Unity-Gain-Matrix	0.007292	-2.62%	0.0005030	42.8%

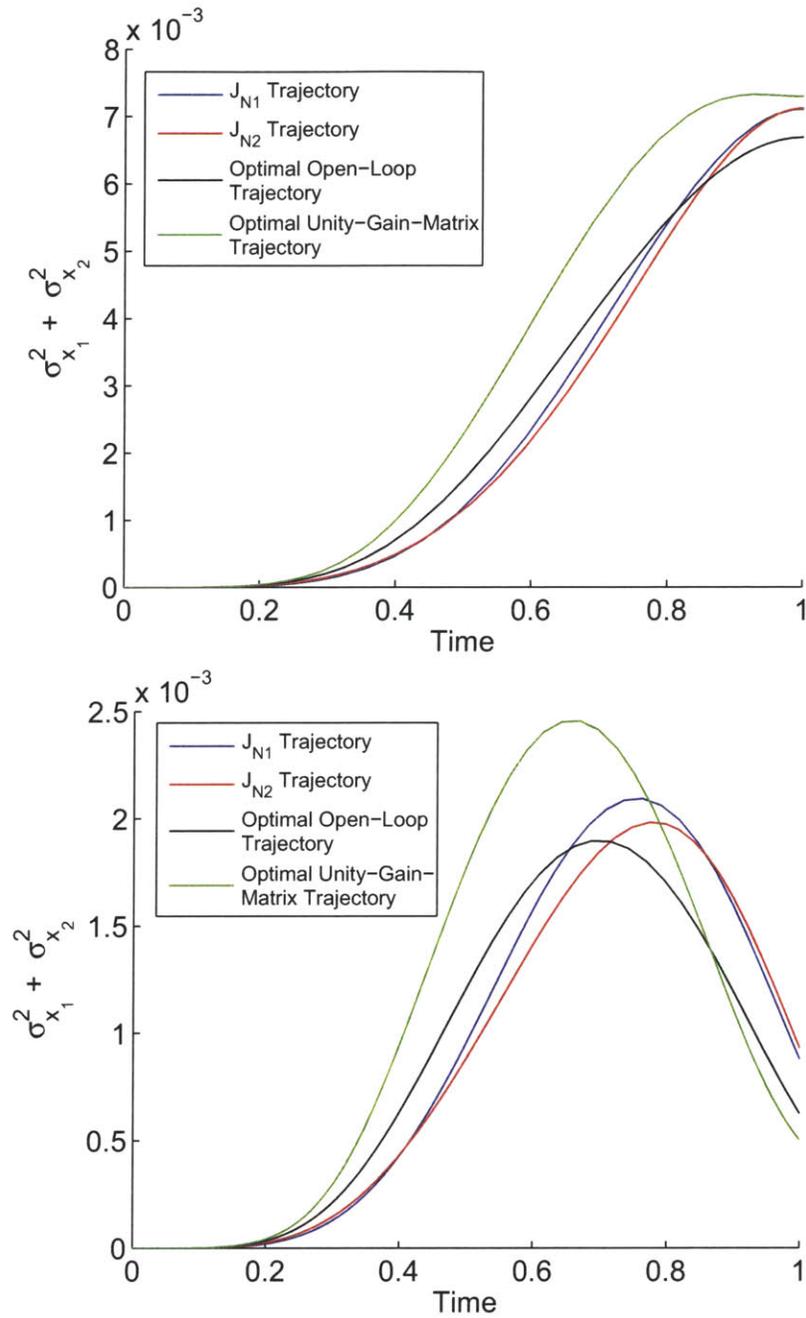


Figure 3-6: Position variance sums propagated along each trajectory given in Fig. 3-5 with parametric process noise in which $w_p = 1$. The top figure shows the open-loop propagations, while the bottom figure shows the closed-loop propagations with unity-gain-matrix feedback. As expected, the final open-loop variance is smallest along the optimal open-loop trajectory (given in black), and the final closed-loop variance is smallest along the optimal unity-gain-matrix trajectory (given in green).

can be explained in both the context of covariance and sensitivity. In the covariance context, the constant strength parameter process noise causes the variance of p and its correlated covariances to grow larger along the trajectory as time progresses. Since uncertainty in the system enters through the px_2 term, less total position error is accumulated by minimizing x_2 late in the trajectory when the uncertainty in p is highest. The velocity and travel times of the boat are both fixed, so the trajectory sweeps farther from shore early in the trajectory. In the sensitivity context, the J_{S2} metric penalizes the integral of the sensitivities of the final position to perturbations in p at each point in time along the trajectory. However, perturbations early in a trajectory naturally have more impact on the final states than perturbations late in a trajectory, a property which the sensitivity matrix captures mathematically within its dynamics. The sensitivity metric integrates the asymmetric integrand, resulting in an asymmetric trajectory.

Several aspects of the process noise trajectory-shaping results are consistent with the bias results. First, applying a simple feedback controller alone reduces the accumulated dispersions along each trajectory by over 86%. Second, the trajectory-shaping improvements are significantly more pronounced in the closed-loop environment, although the improvement percentages are higher in the parametric process noise case. With parametric process noise only, trajectory-shaping in the open-loop environment improves the final position variance by 5.91% (as compared to 1.66% for a pure parametric bias), while the technique improves the unity-gain-matrix variance by 42.8% (as compared to 29.0% with the bias). Third, the trajectory which optimizes the open-loop position variance still does not optimize the closed-loop position variance with the unity gain matrix, nor vice versa. Once again, the optimal open-loop solution appears better suited to handle the addition of feedback than the optimal unity-gain-matrix solution appears able to cope with a loss of feedback.

In this scenario, however, the open-loop robustness advantage appears even more exaggerated. With process noise and feedback, the optimal unity-gain-matrix trajectory improves targeting precision in the unity-gain-matrix environment by over 42% from the nominal J_{N1} trajectory, but the optimal open-loop trajectory — which was not even designed with feedback in mind — still provides a noteworthy improvement of almost 29% when propagated with the unity gain matrix. In the parametric bias scenario, the optimal open-loop trajectory did not improve the final closed-loop position variance. On the other hand, when the open-loop covariance is propagated along the optimal unity-gain-matrix trajectory, its final dispersion statistics are significantly *worse* than even the nominal J_{N1} and J_{N2} trajectories, similar to the parametric bias scenario.

Lastly, these results demonstrate the ability of both trajectory-shaping techniques to handle the presence of process noise in the system model and achieve equivalent results when the cost metrics are formulated appropriately. This is contrary to Zimmer’s assertion that the sensitivity method can only provide equivalent results to the covariance formulation when both measurement noise *and* process noise are omitted.

3.2.1.3 Bias and Noise Combination

The next uncertainty scenario combines the previous two and assumes that the value of p contains a bias as well as uncertainty along the rest of the trajectory. Otherwise, the state dynamics are still assumed to be ideal. With this combined uncertainty model, the sensitivity metric must again be restructured to match the design goal. As before, since p is nominally modeled as a constant, the bias portion of the uncertainty model translates to uncertainty in the initial condition p_0 , and if the perturbations in p throughout the rest of the trajectory are assumed to be uncorrelated in time with zero mean, the process noise portion of the uncertainty model can be represented stochastically as AWGN acting on the dynamics of p . The most appropriate sensitivity metric for minimizing the final position variance sum with the combined uncertainty model must therefore address the sensitivities of the final position states to perturbations in both p_0 as well as $p(t)$. On one hand, the cost metric J_{S2} already takes into account perturbations in p_0 as part of its integral evaluation, but since the bias can cause uncertainties in p_0 independent of uncertainties due to process noise, a more suitable cost metric has the ability to levy additional penalties on sensitivities to perturbations in p_0 . Such a metric can be formulated as the weighted combination

$$J_{S3} = \alpha J_{S1} + \beta J_{S2}, \quad (3.20)$$

where the values of α and β can be adjusted to account for different bias and process noise uncertainty magnitudes, respectively. Setting $\alpha = 0$ recovers the J_{S2} metric formatted to exclusively handle parameter process noise, and setting $\beta = 0$ recovers the J_{S1} metric formatted to exclusively handle a constant parameter bias. Any other combination of positive α and β creates a J_{S3} metric formatted to handle a parameter bias and process noise combination, in which α should reflect the variance of the bias uncertainty and β should reflect the variance of the process noise. While the adjustable weights in the J_{S3} metric allow it to account for all possible bias and process noise combinations, it still only produces a single trajectory for a given set of weights, and these weights must be selected based on knowledge of the bias and process noise magnitudes before solving the trajectory. In addition, due to the specific sensitivity terms included, the J_{S3} metric is only designed to handle bias and process noise uncertainties in p . For uncertainty models that include state uncertainties, the metric must be reformatted once again to include sensitivities to state perturbations.

Applying the covariance technique requires selecting another set of appropriate P_0 and Q matrices. As before, the bias in p can be represented within the P_0 matrix, and the process noise can be expressed in the Q matrix. These conditions can be written as

$$P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_p \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & w_p \end{bmatrix}, \quad (3.21)$$

where c_p is a constant that determines the variance of uncertainty in p_0 provided by the bias, and w_p is a constant representing the variance of the parametric process noise. Since the design goal has not changed, the J_{P1} metric is still the most appropriate

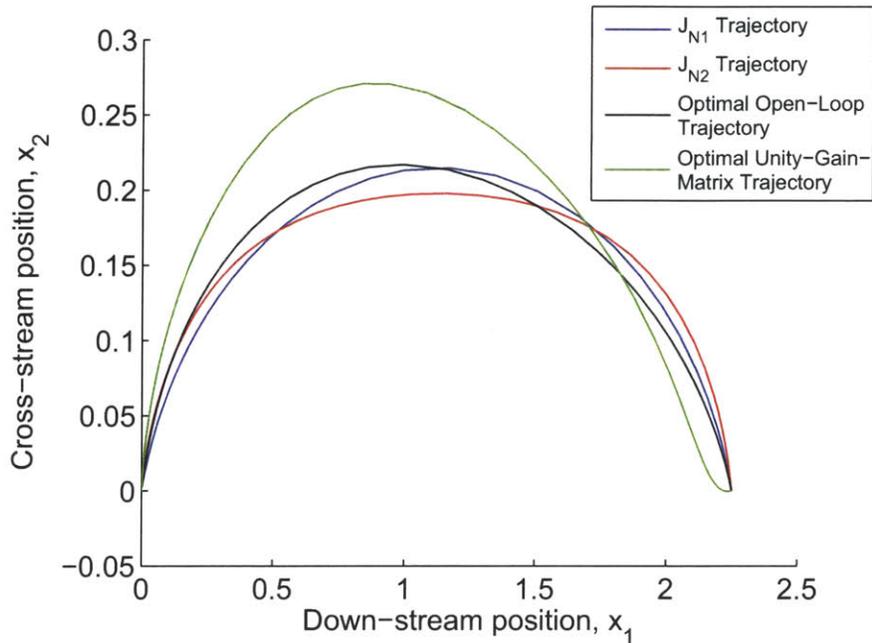


Figure 3-7: Optimal trajectories for combined parametric bias and process noise uncertainty in which $c_p = 1$ and $w_p = 2$. The optimal open-loop trajectory optimizes the final open-loop position variance sum, and the optimal unity-gain-matrix trajectory optimizes the final closed-loop position variance sum with unity-gain-matrix feedback. The J_{N1} and J_{N2} trajectories are unchanged from before.

covariance metric for this scenario.

Without feedback, the J_{S3} sensitivity metric and the J_{P1} covariance metric provide identical optimal trajectories when

$$\frac{\alpha}{\beta} = \frac{c_p}{w_p}. \quad (3.22)$$

When the trajectories are reshaped using the same uncertainty magnitudes with unity-gain-matrix feedback, both methods again produce an identical trajectory, but one that differs from the optimal open-loop trajectory. It is important to note that in order for the sensitivity and covariance methods to produce equivalent results for this combined model, α need not exactly match c_p in value, and neither must β exactly match w_p in value, so long as their ratios remain equivalent. Thus, although the sensitivity method does not include a statistical representation of the bias or process noise in the evaluation of its dynamics or cost metric, specific knowledge of the relative variances of these uncertainties must still be known in order to generate the most appropriate sensitivity metric and achieve the best results. Figure 3-7 shows the optimal open-loop and unity-gain-matrix trajectories for a specific uncertainty combination in which $c_p = 1$ and $w_p = 2$.

The open-loop and unity-gain-matrix covariances are propagated along each trajectory with the combined uncertainty to confirm that the optimal solutions for each

Table 3.3: Final position variance sums with combined parametric bias and process noise in which $c_p = 1$ and $w_p = 2$. The first column lists the trajectories given in Fig. 3-7, the second column provides their final propagated open-loop position variance sums, and the fourth column displays their final propagated closed-loop position variance sums with unity-gain-matrix feedback. The third and fifth columns show the open-loop and unity-gain-matrix improvement percentages, respectively, using the J_{N1} trajectory as the basis of comparison. Degradations in performance are depicted as negative values, and the optimal results for each column are emphasized in bold.

Trajectory	OL Sum	OL Improvement	CL Sum	CL Improvement
J_{N1}	0.03320	N/A	0.002811	N/A
J_{N2}	0.03355	-1.05%	0.002913	-3.63%
Optimal Open-Loop	0.03278	1.27%	0.002334	17.0%
Optimal Unity-Gain-Matrix	0.03617	-8.95%	0.001779	36.7%

controller actually produce the smallest final position variances within their respective feedback environments. Figure 3-8 shows the position variance sums along each trajectory when the covariance is propagated with the combined parametric uncertainty using both zero feedback and unity-gain-matrix feedback. Table 3.3 compares the open-loop and closed-loop final variance improvements generated by the trajectory-shaping techniques, using the optimal J_{N1} trajectory for comparison. The conclusions drawn from the results in this scenario are very similar to those from the previous uncertainty model investigations. First, applying unity-gain-matrix feedback alone improves the final position variances for each trajectory by over 91%. Second, the trajectory-shaping improvements are much more considerable in the closed-loop environment. In the open-loop environment, the final position variance sum improves by only 1.27% with trajectory-shaping – even less than the 1.66% improvement with a pure bias. With unity-gain-matrix feedback, the final position variance sum improves by 36.7% – between the 29% improvement for a pure bias and the 42% improvement for pure process noise. Third, the percentages indicate once again that the optimal open-loop solution appears better suited to handle the addition of feedback than the optimal unity-gain-matrix solution can cope with a loss of feedback. When propagated in an open-loop setting, the optimal unity-gain-matrix trajectory generated almost 9% *worse* final position variances than the J_{N1} trajectory. When the covariance is propagated along the optimal open-loop trajectory with the unity-gain-matrix feedback, however, it still enhances the final expected targeting precision 17% more than the J_{N1} trajectory. Ultimately, these results show the ability of both methods to equivalently handle combinations of parameter bias and process noise when their relative variances can be estimated beforehand.

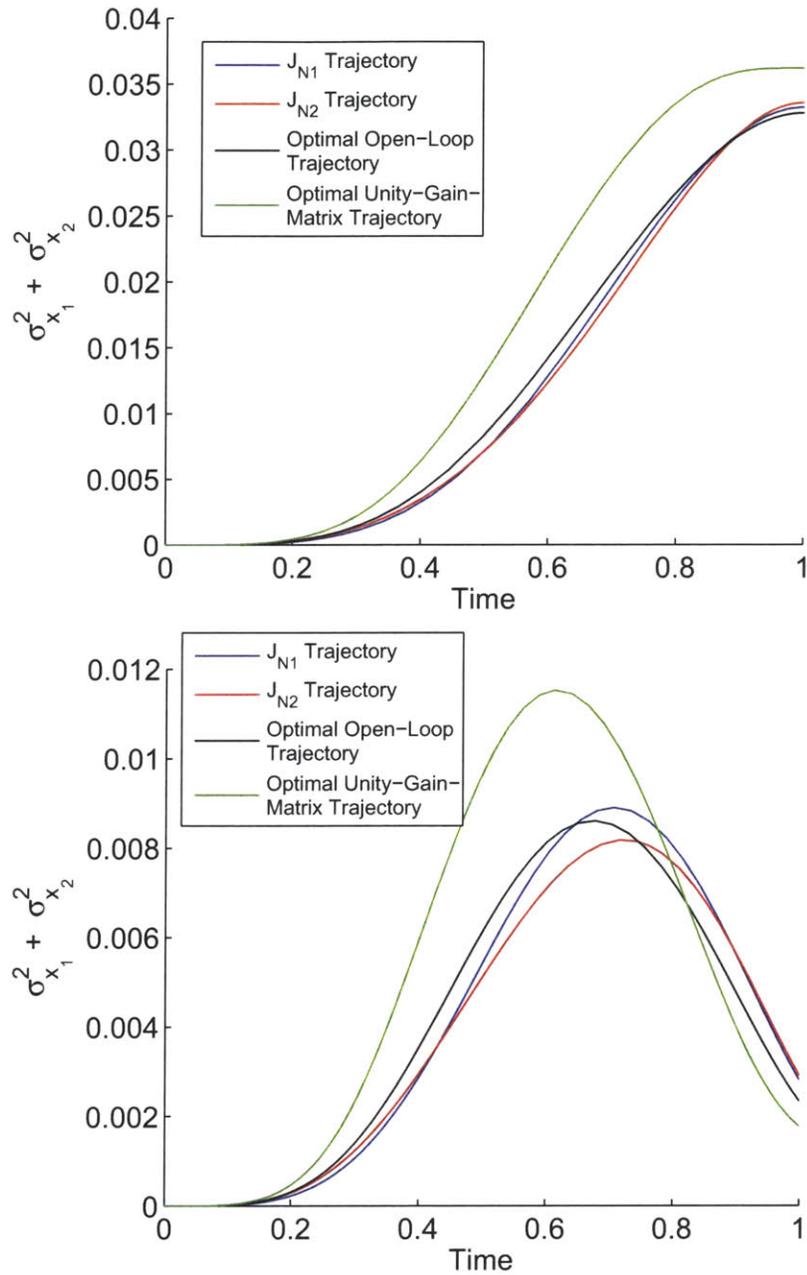


Figure 3-8: Position variance sums propagated along each trajectory given in Fig. 3-7 with combined parametric bias and process noise in which $c_p = 1$ and $w_p = 2$. The top figure shows the open-loop propagations, while the bottom figure shows the closed-loop propagations with unity-gain-matrix feedback. As expected, the final open-loop variance is smallest along the optimal open-loop trajectory (given in black), and the final closed-loop variance is smallest along the optimal unity-gain-matrix trajectory (given in green).

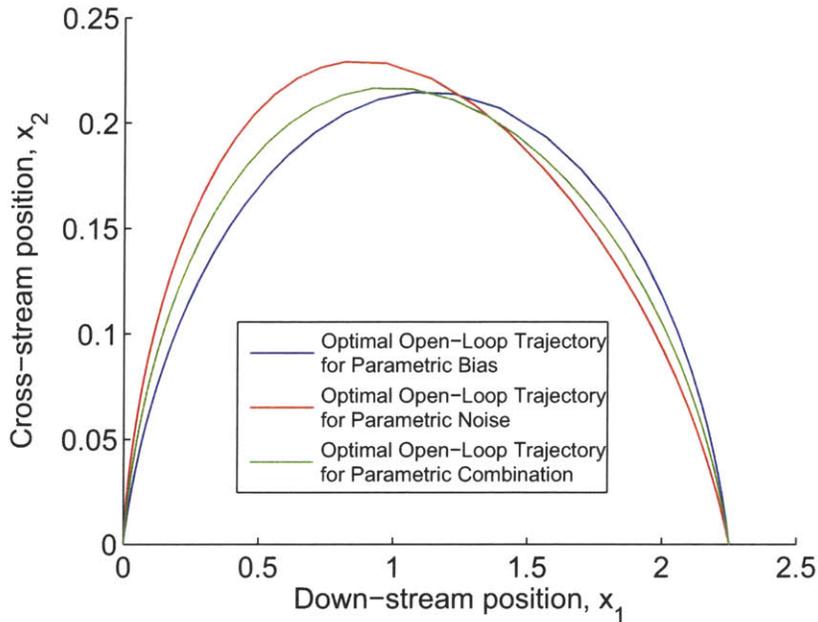


Figure 3-9: Optimal open-loop Zermelo trajectories for each of the three parametric uncertainty models investigated in this chapter. The blue trajectory is optimal for a parameter bias, the red trajectory is optimal for parameter process noise, and the green trajectory is optimal for a parameter bias-noise combination in which the process noise variance is specifically twice as large as the bias variance.

3.2.1.4 Parametric Uncertainty Comparison

At this point, three separate parametric uncertainty models have been investigated, but the results for each model have been shown individually. The optimal open-loop and unity-gain-matrix trajectories have been compared under each feedback scheme, but always with the same uncertainties for which they were specifically designed. When the optimal open-loop trajectories for each of the three parametric uncertainty scenarios are plotted together, their relationships become more apparent. Figure 3-9 plots the optimal open-loop trajectories for each of the three parametric uncertainty models investigated. The optimal open-loop trajectory for a constant parameter bias is equivalent to the minimum integrated cross-stream trajectory, which is symmetric about the vertical line passing through its maximum cross-stream point. The optimal open-loop trajectory for parametric process noise travels farther away from shore early in the trajectory, causing a noticeable asymmetry about the vertical line passing through its maximum cross-stream point. The shape of the optimal open-loop trajectory for the combined parametric uncertainty model appears to blend the shapes of the pure bias and noise solutions. Recalling the formulation of the sensitivity metrics, the set of combined parametric uncertainty solutions can be visualized as a continuum bounded by the pure bias and pure noise cases in the extremes. As the magnitude of the bias grows larger relative to the magnitude of the process noise, the

Table 3.4: Comparison of optimal open-loop trajectories for each parametric uncertainty case. The first column lists the optimal open-loop trajectories for each parametric uncertainty scenario, while the second, third, and fourth columns provide their final position variance sums when subjected to each of the three parametric uncertainty models individually with zero feedback. As expected, trajectories that were optimized for the bias, noise, and combination, respectively, each produce smaller final position variances when applied to the uncertainty models for which they were designed, as emphasized in bold.

Trajectory	Final OL Variance with Bias	Final OL Variance with Noise	Final OL Variance with Combo
Optimal OL Trajectory with Bias	0.01899	0.007105	0.03320
Optimal OL Trajectory with Noise	0.01980	0.006687	0.03318
Optimal OL Trajectory with Combo	0.01918	0.006791	0.03278

bias dominates the error propagation until the uncertainty model approaches a pure bias. To account for a shift in this direction, the relative value of α within the sensitivity metric grows until the relative size of β becomes zero, and the metric reduces to J_{S1} . The opposite is true when the magnitude of the process noise increases and the uncertainty model approaches the pure noise case — β grows until α is diminished to zero, leaving J_{S2} as the metric. The optimal trajectories for the combined uncertainty are directly coupled to these shifts in uncertainty, and the solutions slide along the continuum linking the optimal bias and optimal noise solutions. When the open-loop covariance is propagated along each of these three trajectories using each of the three investigated parametric uncertainty models, the optimality of each solution for its intended uncertainty model is verified.

A similar continuum is visible for the optimal unity-gain-matrix trajectories, as demonstrated in Fig. 3-10, but the established bounds are now the optimal unity-gain-matrix solutions for the pure bias and pure noise cases. When the covariance is propagated along each of these three trajectories using each of the three investigated uncertainty models with unity-gain-matrix feedback to verify the optimality of each solution.

Along the open-loop and unity-gain-matrix continua, the trajectories that are most robust to changing sources of uncertainty lie somewhere in the middle. Table 3.4 lists the final open-loop position variance sums propagated along the optimal open-loop and unity-gain-matrix trajectories with each of the investigated parametric uncertainties, and Table 3.5 lists the unity-gain-matrix feedback variances. The trajectories

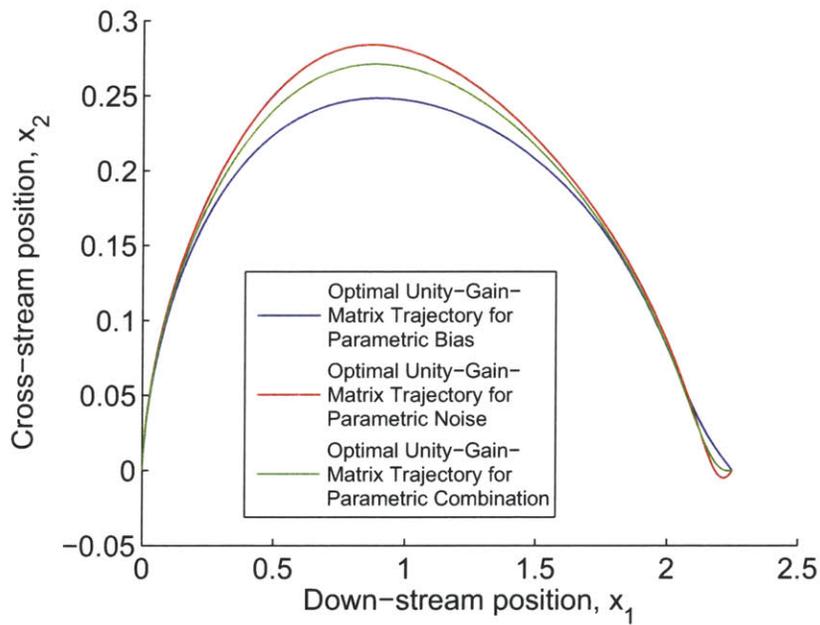


Figure 3-10: Comparison of optimal unity-gain-matrix trajectories for each parametric uncertainty case. The blue trajectory is optimal for a parameter bias, the red trajectory is optimal for parameter process noise, and the green trajectory is optimal for a parameter bias-noise combination in which the process noise variance is specifically twice as large as the bias variance.

Table 3.5: Closed-loop final position variance sums with unity-gain-matrix feedback. The first column lists the optimal unity-gain-matrix trajectories for each parametric uncertainty scenario, while the second, third, and fourth columns provide their final position variance sums when subjected to each of the three parametric uncertainty models individually with unity-gain-matrix feedback. As expected, trajectories that were optimized for the bias, noise, and combination, respectively, each produce smaller final position variances when applied to the uncertainty models for which they were designed, as emphasized in bold.

Trajectory	Final CL Variance with Bias	Final CL Variance with Noise	Final CL Variance with Combo
Optimal CL Trajectory with Bias	0.0007451	0.0005426	0.001830
Optimal CL Trajectory with Noise	0.0007866	0.0005030	0.001793
Optimal CL Trajectory with Combo	0.0007617	0.0005086	0.001779

that optimize a bias-noise combination always accumulate final position variances that fall somewhere between the variances of the pure bias- and noise-optimal trajectories. Practically speaking, when shaping trajectories to reduce susceptibility to errors, the designer must decide upon the uncertainty types and magnitudes for which to account in the optimization process. Any trajectory will be specifically tuned to minimize the uncertainty model assumed in its generation, but it will be sub-optimal for any deviations in that uncertainty climate. Thus, if the designer is unsure exactly which uncertainties are likely to manifest themselves during implementation, these results suggest that including an average of each uncertainty type during the trajectory optimization will at least guarantee that the trajectory is partially suited to combat each particular uncertainty type.

3.2.2 State and Parametric Uncertainty Combinations

The final uncertainty model demonstrates the versatility and effectiveness of both trajectory-shaping techniques by combining the parametric uncertainty types investigated above with state initialization errors and process noise. In the context of the Zermelo problem, initialization errors mean that the initial location of the boat is not precisely known, and state process noise means that the dynamics governing the position rates of the boat contain inaccuracies or perturbations along the trajectory. If state perturbations along the trajectory are assumed to be uncorrelated in time with zero mean, they can be modeled stochastically with AWGN acting on the dynamics.

Beginning with the covariance formulation, if the noise and initialization errors both have zero correlation coefficients, this uncertainty model can be represented as

$$P_0 = \begin{bmatrix} c_{x_1} & 0 & 0 \\ 0 & c_{x_2} & 0 \\ 0 & 0 & c_p \end{bmatrix}, \quad Q = \begin{bmatrix} w_{x_1} & 0 & 0 \\ 0 & w_{x_2} & 0 \\ 0 & 0 & w_p \end{bmatrix}, \quad (3.23)$$

where the off-diagonal terms are zero. Here, c_{x_1} and c_{x_2} represent the variances of the expected down-stream and cross-stream initialization errors, respectively, c_p gives the variance of the parameter bias, w_{x_1} and w_{x_2} provide the state process noise variances in the down-stream and cross-stream rate equations, respectively, and w_p sets the variance of the parametric process noise. For the covariance method, J_{P_1} is once again the most appropriate covariance metric, since the design goal still seeks to minimize the final position variance sum.

Implementing the sensitivity formulation requires an extended application of the logic used to develop the J_{S_1} , J_{S_2} , and J_{S_3} sensitivity metrics. A new metric must be crafted that considers how each component of the present uncertainty model impacts the intended design goal, and penalizes each term according to the relative magnitudes among them. Since the design goal seeks to minimize the unweighted sum of the final position variances, the metric should include the sensitivities of the final position states, and since this model assumes that initial errors and process noise exist for each variable, these sensitivities should be taken with respect to perturbations in both the initial values and time-varying values of each state. Like J_{S_3} , the new metric should be able to penalize the initial condition uncertainties independently of the process noise uncertainties. Applying this logic, the most appropriate sensitivity metric for this scenario becomes

$$\begin{aligned} J_{S_4} = & \alpha \left(\left(\frac{\partial x_1(t_f)}{\partial x_1(t_0)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial x_1(t_0)} \right)^2 \right) \\ & + \beta \left(\left(\frac{\partial x_1(t_f)}{\partial x_2(t_0)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial x_2(t_0)} \right)^2 \right) \\ & + \gamma \left(\left(\frac{\partial x_1(t_f)}{\partial p(t_0)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial p(t_0)} \right)^2 \right) \\ & + \zeta \int_{t_0}^{t_f} \left(\left(\frac{\partial x_1(t_f)}{\partial x_1(t)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial x_1(t)} \right)^2 \right) dt \\ & + \eta \int_{t_0}^{t_f} \left(\left(\frac{\partial x_1(t_f)}{\partial x_2(t)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial x_2(t)} \right)^2 \right) dt \end{aligned} \quad (3.24)$$

$$+ \theta \int_{t_0}^{t_f} \left(\left(\frac{\partial x_1(t_f)}{\partial p(t)} \right)^2 + \left(\frac{\partial x_2(t_f)}{\partial p(t)} \right)^2 \right) dt,$$

where the values of each weight α , β , γ , ζ , η , and θ can be adjusted to account for different relative magnitudes among the initial condition and process noise uncertainty strengths.

The sensitivity and covariance formulations produce identical optimal trajectories when

$$\begin{aligned} \alpha &= c_{x_1}, & \beta &= c_{x_2}, & \gamma &= c_p, \\ \zeta &= w_{x_1}, & \eta &= w_{x_2}, & \text{and } \theta &= w_p. \end{aligned}$$

Choosing a particular case with unity-gain-matrix feedback, in which

$$\begin{aligned} \alpha &= c_{x_1} = 1.5, & \beta &= c_{x_2} = 1.5, & \gamma &= c_p = 0.5, & (3.25) \\ \zeta &= w_{x_1} = 3, & \eta &= w_{x_2} = 2, & \text{and } \theta &= w_p = 1, \end{aligned}$$

the resulting trajectory is significantly different from any of the optimal unity-gain-matrix trajectories for the previous parametric uncertainty models. The general shape and performance of the trajectory more resembles those associated with the nominal cost metrics J_{N1} and J_{N2} than any others. Figure 3-11 show the optimal unity-gain-matrix trajectories for every uncertainty model investigated to this point, as well as the original J_{N1} and J_{N2} trajectories. When the closed-loop dispersion statistics are propagated along each trajectory with the unity gain matrix and the uncertainty magnitudes specified in Eq. (3.25), the newfound trajectory indeed minimizes the final position variance sums, providing improvements greater than 9.95% over the other optimal unity-gain-matrix trajectories, though only 0.96% and 0.24% better than the J_{N1} and J_{N2} trajectories, respectively.

3.2.3 Sensitivity vs. Covariance Summary

The preceding uncertainty models were applied to the simple Zermelo problem in an effort to better understand the capabilities, limitations, strengths, and weaknesses of each method. Ultimately, it was discovered that as long as measurements are excluded, the sensitivity and covariance methods can achieve equivalent results when applied correctly. Both methods can directly desensitize trajectories to any combination of parametric biases, parametric process noise, state initialization errors, and state process noise. Even for such a simple system without much flexibility in feasible trajectories, the improvements can be dramatic. Both methods can also directly shape trajectories to take full advantage of specific linear feedback controller dynamics. Nevertheless, the implementations of each method contain significant differences.

First, for a system with a total of n nominal states, the sensitivity matrix adds n^2 extra variables and differential constraints, while the covariance matrix adds only $\frac{n(n+1)}{2}$ extra variables and differential constraints due to its symmetry. The simple Zermelo problem only requires three more variables than the covariance method, but

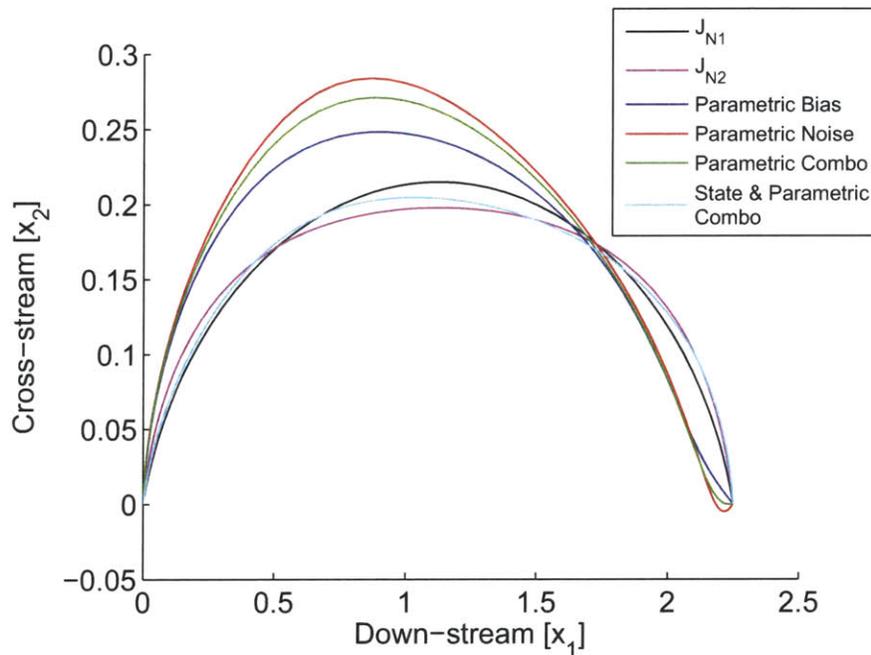


Figure 3-11: Optimal unity-gain-matrix trajectories for each uncertainty model investigated in this chapter. The blue trajectory is optimal for a parameter bias, the red trajectory is optimal for parameter process noise, the green trajectory is optimal for a parameter bias-noise combination in which the process noise variance is specifically twice as large as the bias variance, and the cyan trajectory is optimal for the state-parameter uncertainty combination specified in Eq. (3.25). The black trajectory optimizes the cross-stream integral, while the magenta trajectory optimizes the cross-stream square integral.

as systems grow more complex with greater numbers of nominal states and updated parameters, this difference becomes much more substantial and greatly impacts the viability of many numerical optimization techniques.

Second, the sensitivity method does not provide a statistical measure of the expected dispersions along a trajectory as part of its solution. As demonstrated above, dispersion statistics can be calculated ex post facto along any trajectory, but this requires formulating and propagating the covariance dynamics. Thus, if the trajectory is solved using the sensitivity method, its associated dispersions are analyzed using the covariance dynamics. On the other hand, the covariance method accomplishes both tasks at once, without using the sensitivity matrix at all. In the covariance realm, the Riccati equation is used in both the design and analysis phases, and responses to different uncertainties can be propagated along trajectories by simply tweaking the P_0 , Q , and R matrices.

Third, although the sensitivity matrix does not require statistical uncertainty models in the evaluation of its dynamics or cost metrics, they must still be known in order to format sensitivity metrics in the most appropriate manner. The sensitivity method requires a significantly different cost metric for each individual uncertainty model – each metric must be crafted to include sensitivities of the design goal with respect to perturbations in the uncertain states and parameters. Even if the design goal remains fixed, the metric must be rederived when the uncertainty model changes, and appropriate weights for each term must be founded on statistical knowledge of the uncertainties. On the other hand, the covariance method only requires a single metric for a given design goal, no matter which uncertainty models are used. The sensitivity method only provides equivalent results when the sensitivity penalties match the relative variance magnitudes in the initial covariance and process noise matrices. Both methods require statistical knowledge of the uncertainty models, but the sensitivity metric formulation process is significantly more complicated. As shown in the final example, when state insertions errors, parameter biases, and process noise are all combined, the sensitivity metric can grow very large and complex.

Lastly, while the covariance method can account for measurement updates within the Riccati equation dynamics, the sensitivity matrix cannot account for measurements, at least not when applied as shown here.

For these reasons, the covariance method is selected as the preferred trajectory-shaping technique, and is used exclusively throughout the remainder of this thesis.

3.3 Maximum Capability Confidence

Now, rather than fixing the boat’s final location, a new trajectory design goal seeks to characterize the boat’s maximum down-stream capability. The boat is still constrained to return to shore in fixed unity time, but the down-stream coordinate of the final landing site is left open, and the endpoint cost metric

$$J_{N3} = -x_1(t_f) \tag{3.26}$$

is used to maximize the boat’s down-stream capability.

In reality, however, system uncertainties induce final down-stream dispersions about the optimal landing site. Standard deviation is defined as the square root of variance, so the standard deviation of the final down-stream dispersion can be calculated by propagating the covariance matrix to the final conditions and taking the square root of the first diagonal element. The standard deviation of the final down-stream dispersion measures the expected distance from the nominal landing site inside which the boat is expected to land. The confidence in the final dispersions captured within one standard deviation depends on the probability distribution of the final dispersions. Although the system dynamics are nonlinear, the covariance is propagated along the *linearized* trajectory. The linearization preserves the Gaussian nature of the uncertainties throughout the propagation, although the covariance only provides an approximation of the dispersion statistics subject to linearity assumptions. For a Gaussian distribution, one standard deviation (σ) defines a confidence region about the nominal state that encloses 68.27% of the expected dispersions, and three standard deviations (3σ) provides a 99.73% confidence range. Thus, while the nominal maximum down-stream trajectory characterizes the boat’s maximum capability if no uncertainties are realized during its journey, the *confident* or *useful* down-stream capability only extends to the worst-case, inner boundary of the boat’s downstream confidence region.

Using Eq. (3.26) as the metric, the boat’s nominal maximum down-stream distance is 2.781. Assuming, for example, that a constant parametric bias exists in which $c_p = 1$, the covariance is propagated with unity-gain-matrix feedback to produce a final down-stream standard deviation of 0.05667. Scaling to 3σ and subtracting from the nominal down-stream landing site, the boat’s 99.73% confident down-stream capability is actually only 2.611. However, the covariance trajectory-shaping technique can be used to solve for different trajectories with smaller expected final position dispersions. A trade between intended landing site and final position error can be conducted by combining the maximum down-stream metric with a final position covariance penalty, in which a weighting factor α can be used to scale the covariance penalty relative to the down-stream penalty. Thus when $\alpha = 0$, the original maximum down-stream metric J_{N3} is recovered, but as α grows, the resulting trajectories aim for closer landing sites which tend to generate smaller final position errors. As the final position covariance shrinks, so does the confidence region surrounding the intended landing location. Figure 3-12 shows a series of maximum down-stream trajectories for different values of α , and plots the 3σ confidence range surrounding the nominal landing sites. Table 3.6 compiles the landing data, including the nominal site, the size of the 3σ confidence regions, and the resulting useful ranges for each. In fact, as the figure and table demonstrate, for certain values of α , the resulting 3σ confidence region lies completely within the original maximum down-stream confidence region, so the useful range of the boat is actually extended.

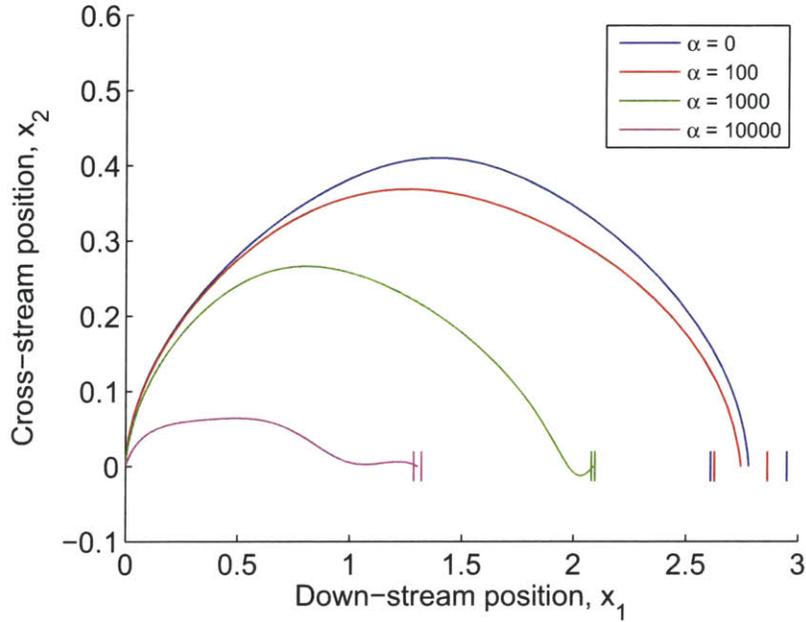


Figure 3-12: Maximum down-stream trajectories with various weights on the final position variance, shown with their 3σ down-stream confidence regions: The blue trajectory purely maximizes the intended down-stream landing site, while the red, green, and magenta trajectories respectively increase the penalty on the final position covariance trace. When the weights are chosen appropriately, the confidence region of shaped trajectories can lie completely within the original confidence region, depicted here by the red confidence region within the blue.

Table 3.6: Maximum down-stream data for optimal trajectories generated with various penalties on final position covariance: The first column lists various penalty weights on the final position covariance trace, the second column shows the targeted down-stream landing site locations for each solution, the third column gives the size of the 3σ down-stream confidence region surrounding each intended landing site, and the fourth and final column lists the 3σ maximum range of the boat as determined by each case, calculated by subtracting the 3σ down-stream error from the intended landing location. Note that for $\alpha = 100$, the boat's maximum range is actually extended.

α	Targeted Landing Site	3σ down-stream Error	3σ Maximum Range
0	2.781	0.17	2.611
100	2.746	0.117	2.629
1000	2.089	0.008	2.081
10000	1.305	0.016	1.289

3.4 Simultaneous Feedback Optimization

The results in Sections 3.2 and 3.3 demonstrate the ability of trajectory-shaping techniques to optimize both open-loop targeting precision and closed-loop targeting precision with predefined feedback control laws. For both of these scenarios, although the reference design depends upon a specific control law, the controller design remains independent of the reference. To further merge the reference trajectory and feedback controller design processes, the optimization problem can be augmented to choose optimal feedback gains in conjunction with the reference trajectory. This process specifically tailors the design of the controller to the reference trajectory, offering two key advantages. First, by increasing the synergy of the reference and controller designs, even greater dispersion reductions can be achieved. Second, and perhaps more importantly, performance characteristics of the feedback controller can be constrained to ensure the closed-loop system behaves in reasonable manners. Revisiting the fixed-target Zermelo boat problem, the covariance-trajectory shaping technique is used to demonstrate the optimization of feedback gains and reference trajectories simultaneously, as well as the ability to ensure realistic controller performance. Although this technique applies to any uncertainty scenario, the combined parametric uncertainty model investigated in Section 3.2.1.3 will be employed in this demonstration.

In order to select the feedback control gains as part of the trajectory optimization process, the problem formulation must be slightly expanded to include the elements of the feedback gain matrix as additional design parameters. This expansion can take several forms, depending on whether or not the designer wishes to utilize constant or variable gain profiles. Constant gains can be included in the problem formulation as states with trivial dynamics, which reduces them to constant parameters. The augmented state vector for the Zermelo problem with the covariance matrix and constant linear feedback gains thus becomes

$$\mathbf{x}_{aP} = \begin{bmatrix} x_1 \\ x_2 \\ p \\ K_1 \\ K_2 \\ \text{symvec}(P) \end{bmatrix}, \quad (3.27)$$

where the trivial dynamics for the gains are written as

$$\dot{K}_1 = 0 \quad \text{and} \quad \dot{K}_2 = 0, \quad (3.28)$$

and the rest of the dynamics are defined as before. Cost metrics, event constraints, and path constraints can now include the feedback gains as well as the nominal states, controls, and covariance, which ultimately allows the designer to tune the system's closed-loop response characteristics.

Section 3.2.1.3 shows the reference trajectory that minimizes the final position variance with a combined parametric uncertainty model and constant unity-gain-matrix feedback. When the constant gains are allowed to vary anywhere between ± 10

and the trajectory is reshaped, a different trajectory and set of gains are discovered which reduce the final position variance 97.5% more than the optimal unity-gain-matrix solution. The optimal gains for the reshaped trajectory are

$$K_1 = -10 \quad \text{and} \quad K_2 = -6.319. \quad (3.29)$$

Although arbitrary for now, the bounds on the gains were selected with the knowledge that higher gains typically lead to faster feedback corrections, which require increased control authority to execute. In this problem, the control variable is the boat's heading angle, so feedback corrections correspond to additional changes in the boat's course. Changing the feedback gains adjusts the system's closed-loop poles that dictate the rates of the angular correction commands. Natural limits exist on the speed at which a boat can command a new course, and the gain bounds here are set to ± 10 to simply allow an increase in control authority by roughly a factor of 10. As expected, the optimization utilizes the ability to increase the gains and system response time, but it is worth noting that both gain bounds are not active in the optimal solution.

Although the covariance trajectory-shaping and simultaneous feedback design processes have garnered vast improvements in expected targeting precision, the closed-loop performance characteristics of each controller along the designed trajectories have not been carefully considered to this point. The optimization routine exploits the performance allowed by the gain bounds, but the gain themselves do not sufficiently guarantee system stability or performance. Since the linearized Zermelo dynamics are time-varying, rigorous stability and closed-loop performance analysis warrants the use of Lyapunov's direct stability methods. The definition of stability for a time-varying system contains subtleties that cannot be fully captured its instantaneous eigenvalues. However, it is assumed for now that the instantaneous stability and closed-loop performance characteristics defined by the time-varying eigenvalues provide a useful approximation of system stability and performance, though they guarantee neither. With this fact in mind, the time-varying eigenvalues of the linearized closed-loop dynamics are evaluated along each trajectory with their respective controller designs. When unity-gain-matrix feedback is applied along the optimal trajectory for that control law, the eigenvalues of the linearized closed-loop system are shown to have positive real parts for the first 0.25 sec of the 1.0 sec journey. When the optimal constant feedback gain matrix is applied along its matching optimal trajectory, the linearized system displays instantaneous instability for the first 0.31 seconds. Although instantaneous instability does not guarantee time-varying system instability, it suggests unfavorable closed-loop stability characteristics along the trajectory and controller pairs designed to this point.

To rectify unfavorable closed-loop performance, additional constraints can be applied to the constant gain problem to ensure that the closed-loop responses meet reasonable specifications. The linearized closed-loop Zermelo dynamics conveniently reduce to a second-order system with two non-trivial poles, so its closed-loop responses can be approximated with simple instantaneous second-order damping ratio

and response time calculations. The two poles of the system can be written as

$$s = -\zeta\omega_0 \pm \omega_0\sqrt{\zeta^2 - 1}, \quad (3.30)$$

where ζ is the damping ratio and ω_0 is the natural frequency. When $\zeta < 1.0$, the poles are complex conjugates, and the system exhibits oscillatory underdamped closed-loop responses. When $\zeta \geq 1.0$, both poles are real, and the system displays critically or overdamped closed-loop responses. The response time can be examined by

$$\bar{t} = \frac{2\zeta}{\omega_0}, \quad (3.31)$$

which physically represents the time at which an applied impulse would cause a step in the system [4]. This calculation provides a measure of the quickness of a system's response time, meaning lower values of \bar{t} correspond to faster systems, and it approximates the 63% rise time with 20% accuracy for damping ratios greater than 0.7 [4].

When the instantaneous system is unstable along the current trajectories, the damping ratios and response times are actually negative, and when the instantaneous system is stable, their damping ratios are positive but relatively small to create fast responses that reduce dispersions more quickly. Along the instantaneously stable portion of the unity-gain-matrix solution, the damping ratio varies between 0 and 0.25, and along the instantaneously unstable portion of the optimal constant gain trajectory, the damping ratio remains between 0 and 0.7. Even during their stable portions, both solutions possess highly oscillatory tendencies, and in both cases, the response times never exceed 0.2 sec. To dictate slower, less oscillatory behavior, path constraints can be placed on the damping ratios and response times along the entire trajectory. When the damping ratio along the entire trajectory is constrained to remain above 0.7, the optimization finds a different trajectory along with the constant gains

$$K_1 = -2.815 \quad \text{and} \quad K_2 = -10 \quad (3.32)$$

Due to the interdependence among the closed-loop performance characteristics, the damping ratio constraint ensures the response times naturally remain above 0.28 along the entire trajectory. Figure 3-13 shows the reference trajectories generated with simultaneous constant feedback optimization. Unlike the other optimal trajectories, the peak cross-stream point of the newly constrained trajectory occurs more than halfway down-stream towards the target. Figure 3-14 plots the damping ratios along each trajectory, showing the significant impact of the performance constraint.

As expected, the increased damping and slower response times dictated by the feedback performance constraints significantly hurt the achievable final position variances. With the constrained trajectory-feedback pair, the improvement in final position variance over the optimal open-loop scenario is only 86.5%, while the original constant-gain trajectory-feedback pair provides a 99.9% improvement. Figure 3-15 and Table 3.7 compare the final position variances of each trajectory with the feedback controllers for which they were designed. With minimal constraints on closed-loop

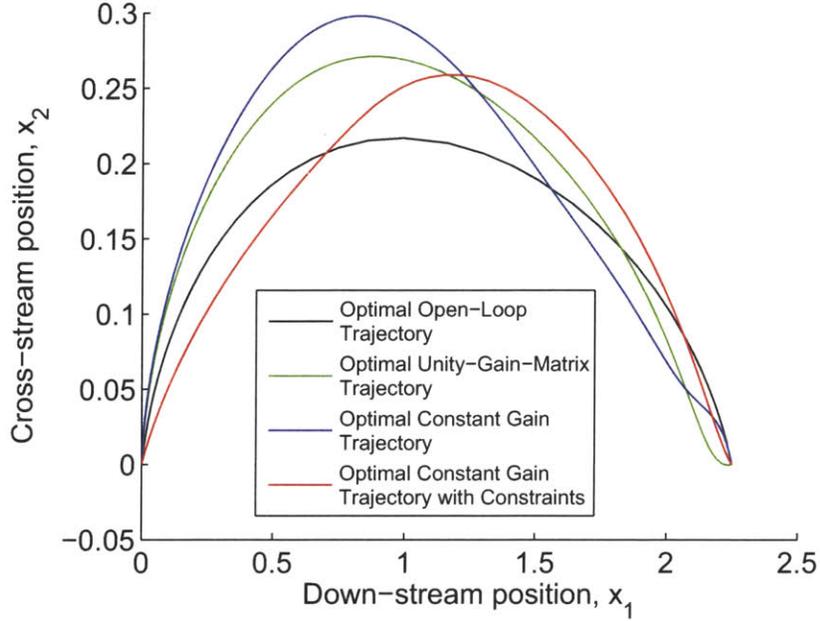


Figure 3-13: Optimal trajectories for various feedback schemes with a combined parametric uncertainty model in which $c_p = 1$ and $w_p = 2$. The black trajectory is optimal without feedback, the green trajectory is optimal for constant unity-gain-matrix feedback, the blue trajectory is optimal for constant feedback gains bounded by ± 10 , and the red is optimal for constant feedback gains bounded by ± 10 and damping ratios greater than 0.7.

Table 3.7: Final position variance sums for each trajectory with its respective feedback control scheme and combined parametric uncertainty in which $c_p = 1$ and $w_p = 2$. The first column lists the optimal trajectory-controller pairs investigated, the second column provides the final expected position variances for each specific pair, and the third column provides the improvement percentages with respect to the optimal open-loop trajectory applied without feedback.

Trajectory	Final Variance Sum	Improvement over Open-Loop Variance Sum
Optimal Open-Loop	0.03278	N/A
Optimal Unity-Gain-Matrix	0.001779	94.6%
Optimal Constant Gain	0.00004361	99.9%
Optimal Constant Gain with Constraints	0.004414	86.5%

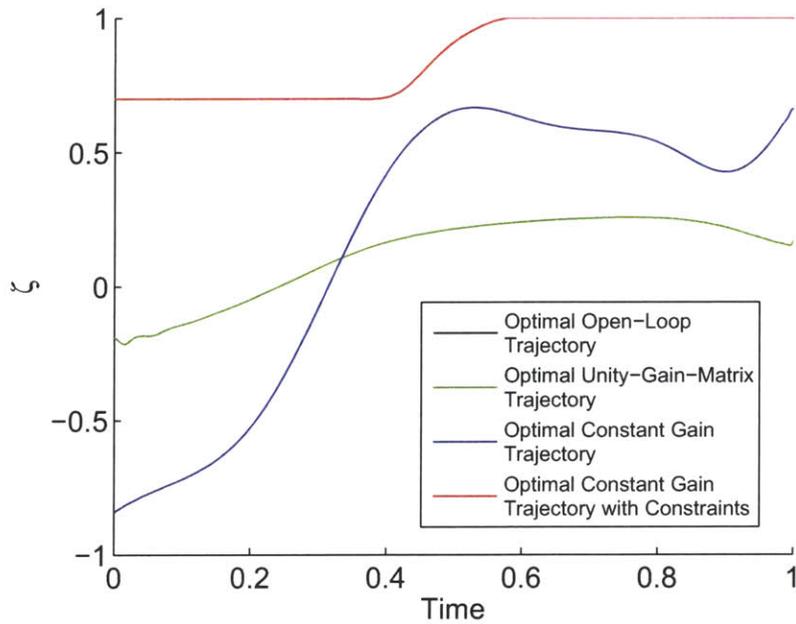


Figure 3-14: Damping ratios along each trajectory optimized with various feedback schemes and a combined parametric uncertainty in which $c_p = 1$ and $w_p = 2$. Without applying a constraint on damping ratio, the optimization provides reference-controller pairs that are unstable for a significant interval of the trajectory, and maintain relatively small damping ratios during the stable portions to produce rapid closed-loop responses.

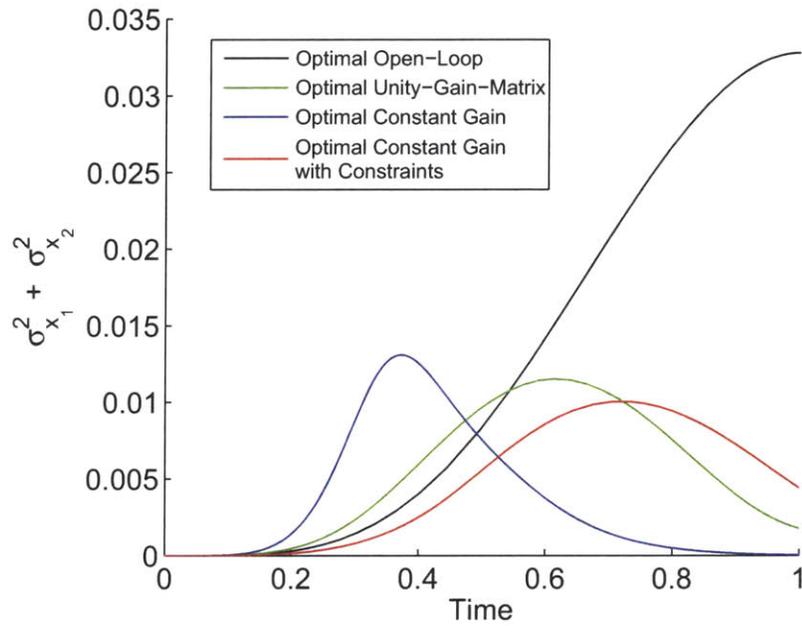


Figure 3-15: Position variance sums propagated along each trajectory with the controller for which it was optimized and combined parametric uncertainty in which $c_p = 1$ and $w_p = 2$. When compared against the other trajectory-controller pairs, the constrained closed-loop performance case provides higher errors than the pairs without closed-loop performance constraints, but still significantly improves upon the open-loop scenario, and demonstrates the ability of the trajectory-shaping techniques to incorporate realistic feedback response constraints into the design process.

performance and control authority, the closed-loop responses along the original unity-gain-matrix and optimal constant-gain trajectories may be too unrealistic with their associated feedback controllers, and the real system may actually perform significantly *worse* if it cannot keep up with the feedback commands. For the established feedback performance constraints, the newest reference trajectory provides the optimal final position variances, and presents a much more realistic and applicable solution. Without constraints on the closed-loop performance, the optimization process simply finds controllers that ask for the fastest and most aggressive corrections, without any regard to the system's actual ability to execute those commands.

These results specifically demonstrate the ability of the investigated trajectory-shaping techniques to not only combine the reference and feedback controller design process but to allow the designer to incorporate closed-loop performance constraints on the optimization process, ensuring realistic control authority specifications. Although results are only demonstrated here for a constant-gain feedback law, this technique can easily be extended to account for variable-gain laws. Rather than including the elements of the feedback gain matrix as additional states with trivial dynamics, the variable gain problem simply requires including the gains as additional control variables so they can vary freely with time.

Chapter 4

Hypersonic Reentry Vehicle Problem

This chapter introduces a mission profile for a recoverable hypersonic reentry vehicle. The investigated flight profile allows the vehicle to collect valuable aerodynamic performance data during the supersonic portion of its flight, before deploying a guided parachute for a safe recovery. Undurti conducted research on the maximum capabilities of axisymmetric skid-to-turn (STT) reentry vehicles with lift-to-drag ratios (L/Ds) of approximately one and two [17], and Abrahamson looked at axisymmetric STT vehicles with L/Ds of approximately two [1]. Recent research in reentry capabilities, however, has gravitated towards asymmetric vehicles that can achieve even greater L/Ds and maneuverability [6, 11]. Performance data gathered from testing such vehicles will be useful for future reentry missions of all types, both manned and unmanned

Undurti and Abrahamson used nominal trajectory optimization techniques to generate footprints characterizing the maximum range capabilities of their vehicles. However, neither included uncertainty considerations in the trajectory design process. Following the recent research trend and extending their work, this problem poses a generic recoverable vehicle with a slightly higher L/D of approximately 2.5, and generates a footprint to exhibit the vehicle's nominal maximum range capabilities. Then, the covariance trajectory-shaping technique is applied to the footprint border trajectories to more accurately represent and even extend the vehicle's confident range. Although knowledge of the vehicle's maximum capabilities is useful for mission planning purposes, test flight trajectories are likely to target recovery zones well within the footprint. Following the maximum range study, a fixed recovery site is selected, and the covariance trajectory-shaping technique is used to accomplish two fixed-target trajectory design goals. In the first, the particular path taken by the vehicle is not important, but it must reach the recovery site with as little expected error as possible. In the second, the vehicle must gather flight test data by performing the largest turn possible, but it must still reach the site within a specified precision tolerance to ensure satisfactory recovery. This particular study demonstrates these trajectory design concepts with a generic small-scale asymmetric vehicle model, but the problem can easily be extended to account for many different reentry scenarios by simply exchanging the particular vehicle model and updating trajectory constraints appropriately.

The first section in this chapter presents the nominal reentry problem. The plant dynamics contain equations for describing the 3 degree-of-freedom (3 DOF) motion of a body above the Earth’s surface, and uses analytic expressions for atmospheric density and asymmetric vehicle aerodynamic coefficients. Physical constraints are imposed upon the trajectory optimization to ensure the vehicle remains controllable throughout the trajectory, and to ensure it attains satisfactory final altitude and velocity conditions for safe, recoverable parachute deployments. The second section uses the nominal problem setup to characterize the maximum downrange and cross-range capabilities by generating a footprint without taking into account any system uncertainties. The third section characterizes probable uncertainties in the nominal reentry model, chooses appropriate models for each, and augments the nominal reference design process to include the covariance matrix. The fourth section applies the covariance trajectory-shaping technique to more accurately represent the vehicle’s useful range subject to confidence constraints. The fifth and final section selects a fixed recovery location within the vehicle’s footprint and generates trajectories for both fixed-target design goals. The covariance trajectory-shaping technique provides the machinery for directly solving these problems, and results are presented.

4.1 Nominal Trajectory Design

The nominal equations governing the six degree-of-freedom (6 DOF) translation and rotation of any body traveling above the Earth’s surface are derived by Bollino [9] and Abrahamson [1]. Each uses several important Cartesian coordinate frames to derive the translational dynamics of the system in spherical coordinates. These frames also provide convenient alternatives for expressing position and velocity vectors, and the transformation matrices given here allow the designer to transform spherical coordinate covariances into Cartesian coordinate covariances in any desired frame. The applicable frames are summarized here, but visual representations of each frame and more detailed explanations can be found in Chapter 2 of Abrahamson’s thesis [1].

4.1.1 Coordinate Frame Summary

4.1.1.1 Earth-Centered-Inertial (ECI)

The ECI frame provides the fundamental inertial basis for the body’s motion above the Earth’s surface. The ECI frame originates at the center of the Earth, with the x_{ECI} direction pointing to the location of the Sun during the Vernal Equinox (also known as the Vernal Point), the z_{ECI} direction pointing through the North pole, and the y_{ECI} direction completing the right-handed orthogonal system. The ECI frame provides the inertial reference for the remaining frames because the direction of the Vernal Point remains fixed in space despite the Earth’s rotations and revolutions.

4.1.1.2 Earth-Centered-Earth-Fixed (ECEF)

The ECEF frame also originates at the center of the Earth, but unlike the ECI frame, it rotates with the Earth. The x_{ECEF} direction points to 0 deg latitude and 0 deg longitude at all times, the z_{ECEF} direction points through the North pole, and the y_{ECEF} direction completes the right-handed orthogonal system. The z_{ECI} and z_{ECEF} axes are always aligned, and approximately every 24 hours when the Earth completes a rotation, the x_{ECI} and x_{ECEF} axes become aligned. Thus the transformation between these two frames is a single rotation about the z axis which depends on the Earth's angular rotation rate Ω_E and the amount of time t that has passed since the last frame alignment. This transformation from the ECI to the ECEF frame can be written as

$$T_{\text{ECI} \rightarrow \text{ECEF}} = \begin{bmatrix} \cos(\Omega_E t) & \sin(\Omega_E t) & 0 \\ -\sin(\Omega_E t) & \cos(\Omega_E t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.1)$$

where the Earth's angular rotation rate is given by the constant

$$\Omega_E = 7.292115 \times 10^{-5} \text{ rad/s}. \quad (4.2)$$

4.1.1.3 Up-Downrange-Crossrange (UDC) and Up-East-North (UEN)

The models derived by Abrahamson and Bollino diverge slightly in the definition of the next frame – the Up-Downrange-Crossrange (UDC) frame. For both authors, the UDC frame originates at the body's center of mass, with the x_{UDC} direction pointing upwards along the line connecting the center of the Earth to the body, the y_{UDC} direction pointing downrange with respect to a specified reference plane, and the z_{UDC} direction pointing crossrange with respect to the same reference plane. However, Abrahamson allows the UDC frame to specify downrange and crossrange directions with respect to any desired reference plane, while Bollino simply references the equatorial plane. For applications in this chapter, the reference plane will always be defined as the equator, so the UDC frame becomes the Up-East-North (UEN) frame, in which the x_{UEN} direction still points upward away from the center of the Earth, but the y_{UEN} direction now points due East, and the z_{UEN} direction points due North. The UEN frame is obtained from the ECEF frame by a sequence of two rotations:

1. Positive rotation about the z_{ECEF} axis by the body's longitude μ
2. Positive rotation about the East direction by the body's spherical latitude λ

This transformation can be combined as

$$T_{\text{ECEF} \rightarrow \text{UEN}} = \begin{bmatrix} \cos(\lambda) \cos(\mu) & \cos(\lambda) \sin(\mu) & \sin(\lambda) \\ -\sin(\mu) & \cos(\mu) & 0 \\ -\sin(\lambda) \cos(\mu) & -\sin(\lambda) \sin(\mu) & \cos(\lambda) \end{bmatrix}. \quad (4.3)$$

4.1.1.4 Velocity (V)

The velocity (V) frame also originates at the body's center of mass, with the x_V direction pointing along the velocity vector, the y_V direction pointing perpendicular to both the body's position and velocity vectors, and the z_V direction completing the right-handed orthogonal system. The V frame is obtained from the UEN frame by a sequence of four rotations:

1. Positive rotation about the y_{UEN} axis by 90 deg
2. Positive rotation about the radial direction by 90 deg.
3. Positive rotation about the radial direction by the body's heading angle ψ
4. Negative rotation about the y_V axis by the body's flight path angle γ

The radial direction is defined as the direction pointing from the center of the Earth upward through the body's center of mass, the heading angle is measured counter-clockwise from the East direction to the local horizontal component of the velocity vector, and the flight path angle is measured from the local horizontal to the velocity vector itself. This transformation can be combined to write

$$T_{UEN \rightarrow V} = \begin{bmatrix} \sin(\gamma) & \cos(\gamma) \cos(\psi) & \cos(\gamma) \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \\ \cos(\gamma) & -\sin(\gamma) \cos(\psi) & -\sin(\gamma) \sin(\psi) \end{bmatrix}. \quad (4.4)$$

4.1.1.5 Non-Rolling Body (B)

Finally, the non-rolling body (B) frame originates at the body's center of mass, with the x_B direction fixed along the vehicle's nose. The body y_B and z_B definitions depend on the particular method used to control the vehicle's motion. The asymmetric body used in this study has a preferred zero-roll orientation for maintaining straight and level flight in which the entire lift vector acts in the local vertical plane. For such vehicles, the y_B axis is fixed to the body's side direction in the preferred zero-roll orientation and the z_B axis is fixed to the body's vertical direction in the preferred zero-roll orientation. The B frame is obtained from the V frame by a sequence of two rotations which depend explicitly on the vehicle's attitude:

1. Positive rotation about the z_V axis by the body's yaw angle ϕ
2. Negative rotation about the y_B axis by the body's pitch angle θ

The yaw angle is the angle between the velocity direction and the component of the body's nose direction (x_B axis) in the x_V - y_V plane, while the pitch angle is the angle

between the x_V - y_V plane and the body's nose direction (x_B axis). This transformation can be combined and expressed as

$$T_{V \rightarrow B} = \begin{bmatrix} \cos(\theta) \cos(\phi) & \cos(\theta) \sin(\phi) & \sin(\theta) \\ -\sin(\phi) & \cos(\phi) & 0 \\ -\sin(\theta) \cos(\phi) & -\sin(\theta) \sin(\phi) & \cos(\theta) \end{bmatrix}. \quad (4.5)$$

While the body's attitude in any zero-roll orientation can be represented by the pitch and yaw angles, it can also be expressed in terms of total angle of attack and bank angle. Total angle of attack α is defined as the direct angle between the body's nose direction (x_B axis) and the velocity vector (x_V axis), and the bank angle σ is defined as the angle between the lift vector and the velocity-vertical plane (z_V - x_V plane). In order to fully characterize the attitude of a body, however, a third aerodynamic angle β must be used to represent sideslip. However, in this particular set of dynamics, the body is assumed to always perform perfectly coordinated turns so that $\beta = 0$ and no sideslip occurs. Pitch and yaw are related to angle of attack and bank angle by purely trigonometric functions. These relationships can be manipulated many different ways using trigonometric identities, with one such representation defined by

$$\sin(\theta) = \cos(\sigma) \sin(\alpha) \quad (4.6)$$

and

$$\cos(\theta) \sin(\phi) = \sin(\sigma) \sin(\alpha). \quad (4.7)$$

4.1.2 3 DOF Dynamics

The 6 DOF model provided by Abrahamson and Bollino can be reduced to 3 DOF by simply omitting the rotational dynamics, which reduces the state vector to

$$\mathbf{x} = \begin{bmatrix} r \\ \mu \\ \lambda \\ v \\ \gamma \\ \psi \end{bmatrix}. \quad (4.8)$$

Each of the states describe the position and velocity vectors of the body in spherical coordinates, so they inherently neglect the oblateness of the Earth's surface. Here, r represents the distance from the body's position to the center of the Earth, μ represents the body's longitude, λ represents the body's spherical latitude, v denotes the body's speed within the ECEF and UEN coordinate frames, γ denotes the body's flight path angle relative to the local spherical horizontal, and ψ denotes the body's heading angle counterclockwise from the East direction. The dynamics governing this system, as derived by Abrahamson and Bollino for an equatorial reference plane, are

written as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{r} \\ \dot{\mu} \\ \dot{\lambda} \\ \dot{v} \\ \dot{\gamma} \\ \dot{\psi} \end{bmatrix}, \quad (4.9)$$

where

$$\begin{aligned} \dot{r} &= v \sin(\gamma), \\ \dot{\mu} &= \frac{v \cos(\gamma) \cos(\psi)}{r \cos(\lambda)}, \\ \dot{\lambda} &= \frac{v \cos(\gamma) \sin(\psi)}{r}, \\ \dot{v} &= -\frac{D}{m} - \frac{G_m}{r^2} \sin(\gamma) + \Omega_E^2 r F_1, \\ \dot{\gamma} &= \frac{L \cos(\sigma)}{mv} + \left(\frac{v}{r} - \frac{G_m}{r^2 v} \right) \cos(\gamma) + 2\Omega_E C_2 + \Omega_E^2 \frac{r}{v} F_2, \\ \dot{\psi} &= \frac{L \sin(\sigma)}{mv \cos(\gamma)} - \frac{v}{r} \cos(\gamma) \cos(\psi) \tan(\lambda) - \frac{2\Omega_E C_3}{\cos(\gamma)} - \frac{\Omega_E^2 r F_3}{v \cos(\gamma)}. \end{aligned}$$

While the position rates are solely trigonometric functions of the states themselves, the velocity rates contain terms that describe several different forces acting on the body. First, the speed, flight path angle, and heading rate equations all contain terms which account for the Coriolis and centrifugal accelerations acting on the body due to the Earth's rotation. The representation of these forces within the equations of motion are simplified by the coefficients

$$\begin{aligned} F_1 &= \cos^2(\lambda) \sin(\gamma) - \cos(\lambda) \sin(\lambda) \cos(\gamma) \sin(\psi), \\ F_2 &= \cos^2(\lambda) \cos(\gamma) + \cos(\lambda) \sin(\lambda) \sin(\gamma) \sin(\psi), \\ F_3 &= \cos(\lambda) \sin(\lambda) \cos(\psi), \\ C_2 &= \cos(\lambda) \cos(\psi), \\ C_3 &= \sin(\lambda) \cos(\gamma) - \cos(\lambda) \sin(\gamma) \sin(\psi). \end{aligned}$$

Second, the speed and flight path rate equations each contain terms which describe the effects of gravitational accelerations acting on the body, utilizing the gravitational constant

$$G_m = 3.9860064 \times 10^{14} \text{ m}^3/\text{s}^2. \quad (4.10)$$

Lastly, the speed dynamics depend explicitly on accelerations due to drag, while the flight path and heading angle dynamics depend explicitly on accelerations due

to lift. Although the 3 DOF model solely describes translational dynamics, the lift and drag forces still depend on the attitude of the body, as well as its particular aerodynamic capabilities. The lift and drag forces acting on any body are proportional to the local atmospheric density, the body's speed squared, the reference area exposed to the relative wind, and the particular lift and drag coefficients, respectively. For all cases investigated here, the total angles of attack are constrained to remain relatively small (≤ 15 deg), meaning the vehicle's nose remains pointed nearly in the direction of the velocity vector. This constraint helps avoid stalls and unstable supersonic behavior, while also allowing the body's reference area S to be modeled adequately as a constant. When the local atmospheric density and the body's speed squared are combined to express dynamic pressure as

$$q = \frac{1}{2} \rho v^2, \quad (4.11)$$

the lift and drag forces can be written as

$$L = qSC_L \quad (4.12)$$

and

$$D = qSC_D, \quad (4.13)$$

respectively. C_L and C_D represent the lift and drag coefficients which are resolved from the vehicle's axial and normal body force coefficients using the trigonometric relationships

$$C_L = C_N \cos(\alpha) - C_X \sin(\alpha) \quad (4.14)$$

and

$$C_D = C_N \sin(\alpha) + C_X \cos(\alpha). \quad (4.15)$$

The axial force coefficient C_X and the normal force coefficient C_N reflect the aerodynamic performance characteristics for the vehicle. Tables of coefficients are typically gathered through empirical performance testing and generally given in terms of Mach number and angle of attack. Undurti and Abrahamson provide analytic approximations that capture the basic functionalities of tabular data for representative axisymmetric hypersonic reentry bodies trimmed to stabilize at $\alpha = 0$ during flight. George extends this model slightly to account for asymmetric bodies with different trim angles and lift characteristics [5], so the models for axial and normal force coefficients become

$$C_X = C_{Xa} e^{-C_{Xb}(M - C_{Xc})} + C_{Xd} + C_{Xk}(\alpha)^2 \quad (4.16)$$

and

$$C_N = C_{N0} + C_{Na}\alpha. \quad (4.17)$$

The constant parameters C_{Xa} , C_{Xb} , C_{Xc} , C_{Xd} , C_{Xk} , C_{N0} , and C_{Na} are selected to appropriately model the vehicle's desired L/D and trim characteristics. Mach number

M is a function of the body's speed and the speed of sound s_s , which can be calculated as

$$M = \frac{v}{s_s}. \quad (4.18)$$

The speed of sound depends on local atmospheric conditions including temperature and density, but for the purpose of this reentry scenario it will be assigned the constant value

$$s_s = 330 \text{ m/s}. \quad (4.19)$$

Since dynamic pressure depends on atmospheric density, this problem also requires an atmospheric density model. Many types of atmospheric density models exist, with varying degrees of accuracy and simplicity. MATLAB provides the `atmoscoesa` program which computes atmospheric density values as a function of geopotential altitude based on the work of the 1976 Committee on Extension to the Standard Atmosphere (COESA) [16]. This model is valid for the range of controllable altitudes in a typical reentry problem, but the execution of `atmoscoesa` is relatively slow, and in a direct collocation optimization method, the atmospheric density must be recalculated every time the system dynamics are evaluated. To speed up the process, an analytic fit of the COESA density data is created to avoid direct calls to `atmoscoesa` itself. The density data resembles an exponential decay as altitude grows, so a polynomial fit is applied to the logarithm of the density output for the range of valid geopotential altitudes, and the analytic density model becomes

$$\rho = e^{a_0 + a_1 h_p + a_2 h_p^2 + \dots + a_n h_p^n}, \quad (4.20)$$

where n denotes the order of the polynomial fit and h_p represents geopotential altitude.

It is important to note that although geopotential altitude approximates geometric altitude, these two quantities are not equivalent. This difference is intuitively explained in part by describing what it means to *fix* each respective height. Fixing geometric height corresponds to fixing the physical altitude above the Earth's surface, as detected by any distance-measuring device, while fixing geopotential height equates to fixing the gravitational potential energy per unit mass [16]. This distinction is important in the context of atmospheric modeling because in actuality, the Earth's oblateness means that gravitational forces acting on the surface along the equator are weaker than gravitational forces acting on the surface at the poles. Due to this phenomenon, gravitational forces compress the atmosphere differently around the Earth, and make geopotential altitude a more appropriate functionality for atmospheric density. Geopotential and geometric altitude are analytically related by

$$h_p = \frac{h R_0}{h + R_0}, \quad (4.21)$$

where h is the geometric altitude, and R_0 represents the radius of the Earth at 45

deg latitude, approximated as

$$R_0 = 6356766 \text{ m} . \quad (4.22)$$

The geopotential altitude calculation requires geometric altitude above the ellipsoidal Earth, and this quantity can be approximated from the spherical radius r and spherical latitude λ by

$$h = r + R_e \left(\frac{1}{2} f (1 - \cos(2\lambda)) + f^2 \left(\frac{R_e}{r} - \frac{1}{16} \right) (1 - \cos(4\lambda)) - 1 \right) , \quad (4.23)$$

where R_e is the Earth's mean equatorial radius and f is the Earth's flattening coefficient. These two constants are approximated to seven significant figures as

$$R_e = 6378137 \text{ m} \quad (4.24)$$

and

$$f = 0.003352811 . \quad (4.25)$$

Using Eqs. 4.9-4.25, the 3 DOF dynamics for a hypersonic body traveling above the surface of the ellipsoidal Earth are completely defined.

4.1.3 Vehicle Model

As mentioned above, this scenario will investigate an asymmetric reentry vehicle with a maximum L/D of 2.5. To demonstrate the ability to handle vehicle asymmetries, the angle of attack trim condition will be set squarely between the upper and lower bounds at 7 deg. To model these particular L/D and trim characteristics, the axial and normal force coefficient fitting constants are chosen as

$$\begin{aligned} C_{Xa} &= 0.317 , \\ C_{Xb} &= 0.550 , \\ C_{Xc} &= 1.00 , \\ C_{Xd} &= 0.0830 , \\ C_{Xk} &= 1.00 , \\ C_{N0} &= 0.0200 , \\ C_{Na} &= 3.00 . \end{aligned}$$

Hypersonic reentry vehicle sizes and shapes span a very wide spectrum, but this scenario arbitrarily chooses a small-scale 500 lb model so that

$$m = 500 \text{ lb} = 226.8 \text{ kg} . \quad (4.26)$$

For the purpose of the 3 DOF dynamic calculations, the dimensions of the vehicle do not matter except in the calculation of an appropriate reference area. However, rather than explicitly defining a vehicle shape, a reference area is selected so that

the body's nominal ballistic coefficient at the reentry condition is approximately 3000 lb²/ft², or 14,650 kg²/m² in metric units. The ballistic coefficient can be calculated as

$$\beta = \frac{m}{C_d S}, \quad (4.27)$$

and using Eqs. 4.15 and 4.16 to find C_d at the initial conditions given by Eq. (4.35) in the next section, an appropriate reference area can be calculated approximately as

$$S = 1.969 \text{ ft}^2 = 0.1829 \text{ m}^2. \quad (4.28)$$

4.1.4 Nominal Optimization Problem

To make the numerical values for radial distance a little more intuitive, the distance r from the center of the Earth to the body will be replaced by the altitude a of the body above the spherical Earth model. These two quantities are related simply by

$$r = a + R_e. \quad (4.29)$$

In this relationship,

$$dr = da, \quad (4.30)$$

which means

$$\dot{r} = \dot{a}, \quad (4.31)$$

so this change of variables does not affect the dynamic constraints given by Eq. (4.9). Using the spherical position and velocity components as the state vector, the optimization's control variables become the body's total angle of attack and bank angle. However, while control bounds applied according to Eq. (2.8) restrict the magnitudes of the control variables along the trajectory, they do not restrict their rates of change. In the pure 3 DOF model, the body's rotational dynamics are unconstrained, so the optimization has the ability to change the aerodynamic angles instantaneously. In reality, the vehicle's attitude is governed by the rotational dynamics included in the 6 DOF model, which naturally imposes limits on the rates at which the vehicle can rotate. 6 DOF limitations can be partially addressed by expanding the state vector to include total angle of attack and bank angle, and defining the control variables as their respective rates. The state vector then becomes

$$\mathbf{x} = \begin{bmatrix} a \\ \mu \\ \lambda \\ v \\ \gamma \\ \psi \\ \alpha \\ \sigma \end{bmatrix} \quad (4.32)$$

with the control vector

$$\mathbf{u} = \begin{bmatrix} \dot{\alpha} \\ \dot{\sigma} \end{bmatrix}. \quad (4.33)$$

The dynamic constraints given by Eq. (4.9) are also expanded as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{a} \\ \dot{\mu} \\ \dot{\lambda} \\ \dot{v} \\ \dot{\gamma} \\ \dot{\psi} \\ \dot{\alpha} \\ \dot{\sigma} \end{bmatrix}, \quad (4.34)$$

4.1.4.1 Initial and Terminal Constraints

This scenario assumes that the reentry vehicle has already been lofted to an altitude of 150,000 ft and begins its descent traveling along the equator at 13,000 ft/s and 0 deg flight path angle. The spherical and ellipsoidal Earth models are equivalent at the equator, so the initial altitude condition does not need to be adjusted for the Earth's oblateness. Similarly, the ellipsoidal Earth model assumes the Earth is flattened along the polar axis so that spherical and ellipsoidal longitudes are always equivalent. In fact, μ does not appear anywhere in the 3 DOF dynamic constraint calculations. For all practical purposes, the initial longitude does not matter because the longitude state solutions can be shifted ex post facto without corrupting the rest of the trajectory. In metric units, the initial conditions become

$$\mathbf{x}_0 = \begin{bmatrix} 150,000 \text{ ft} \\ 0 \text{ deg} \\ 0 \text{ deg} \\ 13,000 \text{ ft/s} \\ 0 \text{ deg} \\ 0 \text{ deg} \\ 0 \text{ deg} \\ 0 \text{ deg} \end{bmatrix} = \begin{bmatrix} 45,720 \text{ m} \\ 0 \text{ rad} \\ 0 \text{ rad} \\ 3962.4 \text{ m/s} \\ 0 \text{ rad} \\ 0 \text{ rad} \\ 0 \text{ rad} \\ 0 \text{ rad} \end{bmatrix}. \quad (4.35)$$

This scenario assumes that the reentry vehicle will deploy a guided parachute for safe recovery. Many chutes require relatively slow subsonic speeds for safe deployment, and to ensure that this is possible, this problem requires the vehicle to pitch straight up at the end of its flight, reduce its speed, and trigger a parachute at an appropriate velocity as it falls back down toward Earth. Unfortunately, the analytic expressions for axial and normal lift coefficients given by Eqs. 4.16 and 4.17 are derived explicitly to model the aerodynamic characteristics of a hypersonic vehicle at supersonic speeds, and they provide poor approximations when the vehicle's speed drops much below the speed of sound. To retain the fidelity of the model, the trajectory design will cease

near the subsonic transition, leaving the remaining ascent and subsequent parachute descent for other simulations. In the Apollo program, drogue chutes were typically released at an altitude of 24,000 ft although they were designed for deployments as high as 40,000 ft [10]. To give this vehicle a similar altitude cushion, the reference trajectory will cease when the vehicle is traveling in the upward direction at a speed of 1000 ft/s and an altitude of 15,000 ft. The heading rate equation has a singularity when the flight path angle is 90 deg, so the terminal condition will be near-vertical with an 85 deg flight path angle. Propagating from these terminal conditions along the equator with a simple drag model at zero angle of attack, the vehicle can be expected to reach an apogee of just over 25,000 ft before falling back down to Earth, meeting the desired altitude cushion.

The terminal altitude condition describes the final height of the body above the Earth's surface, but the states only provide the vehicle's position in terms of spherical coordinates. Since the final recovery location may be located off the equator where the spherical and ellipsoidal Earth models are no longer equivalent, the final altitude condition must be expressed as an event constraint using Eq. (4.23) to calculate the final ellipsoidal altitude. The ellipsoidal altitude h is a function of a and λ , so the altitude event constraint can be evaluated as

$$e_h = h(a_f, \lambda_f) , \quad (4.36)$$

with the upper and lower bounds pinned to the desired altitude, so that in meters

$$e_h^L = e_h^U = 15,000 \text{ ft} = 4572 \text{ m} . \quad (4.37)$$

Although the flight path angle γ actually refers to the local sphere, the difference between γ and the flight path angle relative to the local ellipsoid at any given location is equivalent to the difference between the spherical and ellipsoidal (geodetic) latitudes at that location. Since the trajectories in this scenario stay near the equator, the differences in flight path angle are considered small enough to be neglected for this problem. Thus, the final speed and flight path angle constraints can be written simply as final state bounds, where

$$v_f = 1000 \text{ ft/s} = 304.8 \text{ m/s} , \quad (4.38)$$

and

$$\gamma_f = 85 \text{ deg} = 1.4835 \text{ rad} . \quad (4.39)$$

The vehicle's time of flight is measured relative to the beginning of its trajectory, so the initial time is fixed to zero. The final time, however, is left unconstrained to allow as much maneuvering as desired before reaching the terminal state conditions.

4.1.4.2 Interior Constraints

The 3 DOF dynamics given by Eq. (4.9) contain singularities at

$$\begin{aligned}
 r &= 0, \\
 \lambda &= \pm \frac{\pi}{2} \text{ rad}, \\
 v &= 0, \\
 \gamma &= \pm \frac{\pi}{2} \text{ rad}, \\
 m &= 0,
 \end{aligned}$$

so spherical radius, latitude, speed, and flight path angle must each be bounded away from their respective singularities to prevent the numerical algorithm from attempting undefined calculations anywhere along the trajectory. Realistically, the smallest possible value of r would occur on the Earth's surface at each of the poles. Solving Eq. (4.23) for spherical altitude when $h = 0$ and $\lambda = \frac{\pi}{2}$ rad demonstrates that a can never decrease below -70,160 ft without dipping below sea level at the poles, so this value will be used as the lower bound for a . Values for the other states with singularities are chosen near but away from their unstable values to keep the problem as unconstrained as possible. Longitude, heading angle, and bank angle do not contain singularities, but since numerical optimization techniques typically require finite bounds for each variable to limit the scope of the search algorithms, their bounds are set to ± 360 deg to leave them essentially unconstrained. The total angle of attack is bounded by 0 and 15 deg, which forces the vehicle to roll over in order to pitch downwards. The complete lower interior state bounds can thus be written as

$$\mathbf{x}^L = \begin{bmatrix} -70,160 \text{ ft} \\ -360 \text{ deg} \\ -89 \text{ deg} \\ 5 \text{ ft/s} \\ -89 \text{ deg} \\ -360 \text{ deg} \\ 0 \text{ deg} \\ -360 \text{ deg} \end{bmatrix} = \begin{bmatrix} -21,384 \text{ m} \\ -2\pi \text{ rad} \\ -1.5533 \text{ rad} \\ 1.5240 \text{ m/s} \\ -1.5533 \text{ rad} \\ -2\pi \text{ rad} \\ 0 \text{ rad} \\ -2\pi \text{ rad} \end{bmatrix} \quad (4.40)$$

and the complete upper interior state bounds can be written as

$$\mathbf{x}^U = \begin{bmatrix} 10,000,000 \text{ ft} \\ 360 \text{ deg} \\ 89 \text{ deg} \\ 100,000 \text{ ft/s} \\ 89 \text{ deg} \\ 360 \text{ deg} \\ 15 \text{ deg} \\ 360 \text{ deg} \end{bmatrix} = \begin{bmatrix} 3,048,000 \text{ m} \\ 2\pi \text{ rad} \\ 1.5533 \text{ rad} \\ 30,480 \text{ m/s} \\ 1.5533 \text{ rad} \\ 2\pi \text{ rad} \\ 0.26180 \text{ rad} \\ 2\pi \text{ rad} \end{bmatrix} \quad (4.41)$$

While the final altitude is constrained by the event given in Eq. (4.36), a path constraint must be included to prevent the vehicle from hitting the Earth's surface at any other time during the trajectory. Such a constraint is particularly important in this scenario because the vehicle will be plunging toward the ground before pulling into a vertical climb near the trajectory's end. The path function is simply the body's altitude above the ellipsoid calculated using Eq. (4.23), so the path function can be written as

$$d_h = h(a(t), \lambda(t)) . \quad (4.42)$$

Neglecting terrain features on the surface of the Earth, a lower bound of zero prevents the vehicle from flying below sea level anywhere around the globe. The upper bound can be set to a large number to keep the vehicle's maximum altitude unconstrained, so that the path bounds become

$$d_h^L = 0 \quad (4.43)$$

and

$$d_h^U = 10,000,000 \text{ ft} = 3,048,000 \text{ m} .$$

Although the vehicle's maximum altitude is unconstrained, it is desirable to ensure the body remains controllable throughout its flight, and this reentry vehicle depends on aerodynamic lift and drag forces to control its trajectory. The lift and drag forces are proportional to dynamic pressure, reference area, and the aerodynamic coefficients. For the speeds and altitudes investigated in this problem, dynamic pressure dominates the lift and drag calculations governing the vehicle's controllability. Placing a lower bound on dynamic pressure along the trajectory provides a useful controllability metric linked to adequate airflow over the body. Intuitively, this makes sense, because if the vehicle is either traveling too slowly or too high above the atmosphere, it will not be able to utilize aerodynamic forces to control its motion. The dynamic pressure path bound can thus be written as

$$d_q = q(t) . \quad (4.44)$$

The dynamic pressure at the initial condition is just under $14,300 \text{ N/m}^2$, while the dynamic pressure at the final condition is just over $36,400 \text{ N/m}^2$. A dynamic pressure lower bound of $10,000 \text{ N/m}^2$ ensures that the vehicle satisfies the controllability requirement as it begins its unpowered glide, while still giving it the freedom to skip slightly higher in the atmosphere if desired. The minimum dynamic pressure guarantees that the vehicle remains in the presence of drag, and without a source of thrust, its total energy and speed will monotonically decrease. A sufficiently large upper bound must be set to keep the maximum dynamic pressure unconstrained. Although impossible, if the vehicle could somehow reach the ground at a speed of $13,000 \text{ ft/s}$, the dynamic pressure would be smaller than $9,700,000 \text{ N/m}^2$, so an upper bound of $10,000,000 \text{ N/m}^2$ ensures the dynamic pressure upper bound will never be active. The dynamic pressure path bounds can thus be written as

$$d_h^L = 10,000 \text{ N/m}^2 \quad (4.45)$$

and

$$d_h^U = 10,000,000 \text{ N/m}^2.$$

Finally, bounds must be chosen to limit the angle of attack and bank angle rates. In reality, the rate limits depend on the magnitudes of the vehicle's moments of inertia, as well as the forces generated by its control actuators. Since this problem utilizes a fictitious vehicle model to simply demonstrate concepts for any trajectory design application, the angle of attack rate limit is set arbitrarily to $\pm 10 \text{ deg/s}$, forcing the vehicle to take at least 1.5 sec to rotate between its minimum and maximum angle of attack orientations. The moment of inertia about the vehicle's longitudinal axis is assumed to be smaller than the moments about the other body axes, so the bank rate limit is arbitrarily set to $\pm 30 \text{ deg/s}$, allowing the bank angle to change three times more rapidly than the angle of attack. Applying these limits, the lower control bounds are written as

$$\mathbf{u}^L = \begin{bmatrix} -10 \text{ deg/s} \\ -30 \text{ deg/s} \end{bmatrix} = \begin{bmatrix} -0.17453 \text{ rad/s} \\ -0.52360 \text{ rad/s} \end{bmatrix} \quad (4.46)$$

and the upper control bounds are written as

$$\mathbf{u}^U = \begin{bmatrix} 10 \text{ deg/s} \\ 30 \text{ deg/s} \end{bmatrix} = \begin{bmatrix} 0.17453 \text{ rad/s} \\ 0.52360 \text{ rad/s} \end{bmatrix}.$$

Although this completes the set of constraints for this particular investigation, constraints can be altered or added to satisfy any particular mission scenario. For instance, when designing manned reentry trajectories, g-loading considerations become extremely important because the human body cannot handle as much strain as the vehicle structure itself. To account for human g-loading capacities, an additional path constraint could be formulated with bounds set to ensure g-loads remain within human tolerances for the duration of the flight. Heat loads, heating rates, and any other imaginable constraints can also be used to alter the trajectory application.

4.1.4.3 Scaling

As explained by Ross [12], the efficiency of numerical solution techniques depends highly on the scaling of the design parameters. Numerical methods rely on gradients that relate changes in the states and controls to deviations in the cost metric and constraints to produce search directions and step sizes for minimizing the metric and satisfying the constraints. When a problem is numerically well-conditioned, good gradients provide efficient paths to extremal solutions. However, when a problem is ill-conditioned, poor gradients generate paths that either take significantly more time to navigate or break the optimization algorithm altogether. As the number of design parameters grows in size, and the cost and constraints grow in complexity, the effects of poor scaling are magnified.

The states, controls, and time variables in any system possess natural units, and this system mixes distances, speeds, angles, angular rates, and time. Specifically, the spherical altitudes and ECEF speeds are given in meters and meters per second,

respectively, the rest of the states are angles given in radians, and the two control variables are given in radians per second. A typical trajectory, for instance, could very easily spend 100 sec at altitudes above 30,000 m and speeds above 8000 m/s, but never attain latitude and longitude values greater than 0.25 rad. When the design variables differ by orders of magnitude, the cost, constraint, and gradient magnitudes also vary greatly, and the numerical algorithm quickly becomes ill-conditioned. This disparity certainly occurs in the reentry problem, and attempting to solve it using a direct pseudospectral collocation method with natural units proves painfully slow and oftentimes impossible, particularly when the problem is broken into multiple segments. To address this issue, the states, controls, time, cost, and constraints can each be scaled separately so that the problem solved by the optimization routine is no longer in natural units. Different techniques exist, but applying constant linear scales to the states, controls, and time provides tremendous improvements in the solution speed and quality. Using this concept, the scaled states can be written as

$$\mathbf{x}_s = S_x^{-1} \mathbf{x}, \quad (4.47)$$

the scaled controls can be written as

$$\mathbf{u}_s = S_u^{-1} \mathbf{u}, \quad (4.48)$$

and the scaled time can be written as

$$t_s = S_t^{-1} t. \quad (4.49)$$

S_x is a constant $n \times n$ diagonal matrix with each diagonal element providing a unique scale for each individual state. Similarly, S_u is a constant $m \times m$ diagonal matrix containing scales for each control variable, and S_t is a scalar time scale.

Scaling the state, control, and time variables within the optimization routine requires equivalently scaling the initial state conditions, terminal state conditions, interior state bounds, control bounds, and time bounds. The optimization algorithm then processes the scaled variables and passes them into the cost and constraint evaluations. However, the cost and constraints are functions of the unscaled variables, so the design parameters must be unscaled before proceeding with the evaluations. After the evaluations, the cost and constraint functions pass unscaled cost and constraint values to the optimization machinery. Additional scales may be applied to the cost, event, and path outputs if desired, but they are not required. When the cost metric, event constraints, and path constraints remain unscaled, the optimization algorithm simply calculates gradients with respect to the original unscaled functions. However, the output from the dynamic constraint evaluation *cannot* be left unscaled. The state and time scales *implicitly* scale the state rates, so the rates passed back to the optimizer must be scaled to match the states and times visible to the optimizer. Using Eqs. 4.47 and 4.49, the relationship between the scaled and unscaled rates can be written

$$\dot{\mathbf{x}}_s = S_t S_x^{-1} \dot{\mathbf{x}}, \quad (4.50)$$

which must be applied at the end of the dynamic constraint evaluation.

Multiple concepts and opinions exist for selecting the most appropriate set of scales. For instance, Ross advises that the scales should be selected so that the scaled states and costates are roughly the same order of magnitude [12]. The costates represent the sensitivity of the cost metric to changes in the dynamic constraints, so this is one way to condition the gradients. For this problem, the scales are simply chosen to normalize the range of each variable as much as possible. First, S_t is chosen to normalize the vehicle's time of flight from 0 to 1, which requires that

$$S_t = t_f. \tag{4.51}$$

However, since t_f is free, S_t must also be allowed to vary. This can be accomplished by defining S_t as an additional constant parameter chosen within the optimization scheme. This parameter can then be scaled independently, choosing a constant to estimate the value of t_f for the particular problem.

4.2 Nominal Maximum Capabilities

Now that the nominal dynamics have been completely defined, cost metrics can be formulated to find trajectories that characterize the vehicle's maximum downrange and crossrange capabilities without accounting for any uncertainties in the system. Since the vehicle's initial velocity carries it eastward along the equator, and since the UEN frame is used to define the downrange and crossrange directions, metrics which penalize downrange distance and final longitude provide the same result. The metric

$$J = -\mu(t_f) \tag{4.52}$$

can be used to find the attainable recovery zone with the maximum downrange distance, and the metric

$$J = \mu(t_f) \tag{4.53}$$

can be used to find the attainable recovery site with the minimum downrange (or maximum uprange) distance. Similarly, metrics penalizing maximum crossrange distance and maximum final latitude provide equivalent results in this scenario, so the metric

$$J = -\lambda(t_f) \tag{4.54}$$

can be used to find the attainable recovery zone with the highest crossrange distance. Although spherical and geodetic latitudes are slightly different, geodetic latitude increases monotonically as a function of spherical latitude, and vice versa, so both metrics produce the same results.

The vehicle footprint is defined as the space of downrange-crossrange locations which the vehicle can reach from a particular initial condition while satisfying every en route and terminal constraint. The maximum downrange, uprange, and crossrange cases define three important points on the footprint boundary, and the rest of the boundary can be filled in using several methods. In one method, the northern half is

Table 4.1: Nominal values for maximum downrange, uprange, crossrange recovery zone locations: The first column lists each respective case, the second and third columns list the recovery zone locations in geodetic latitude and longitude coordinates, respectively, and the fourth and fifth columns list the recovery zone downrange and crossrange components in kilometers, respectively. Values are given with four significant figures.

Solution	Lat [deg]	Lon [deg]	Downrange [km]	Crossrange [km]
Max Downrange	0	13.88	1544	0
Max Uprange	0	1.848	205.7	0
Max Crossrange	3.325	8.232	916.4	367.6

solved using the maximum latitude metric while simultaneously fixing final longitudes between the maximum downrange and uprange solutions. Here, the final longitudes are selected as intermediate degree integers, so for this problem the set becomes

$$\mu_f = [2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13] \text{ deg.} \quad (4.55)$$

A separate solution is found for each individual μ_f by applying the metric given in Eq. (4.54). The same process can be used to solve for the southern half, but the sign of the metric is switched to minimize the terminal latitude of each solution.

Figure 4-1 shows the resulting footprint boundary in terms of latitude and longitude as well as in terms of downrange and crossrange distance. Table 4.1 summarizes the maximum downrange, uprange, and crossrange capabilities. For this particular set of initial conditions, the footprint is conveniently symmetric about the equator. This phenomenon occurs because the body's initial heading angle is both aligned with and located on the equatorial plane, and the Coriolis and centrifugal accelerations are symmetric about the Earth's plane of rotation. When the initial conditions do not meet these criteria, however, the footprint becomes slightly asymmetric. For this scenario, the footprint demonstrates that from these initial reentry conditions, the vehicle can reach recovery zones inside a swath nearly 735 km wide and 1338 km long, making its achievable downrange distances twice as long as its crossrange distances. This disparity arises because turns dissipate energy and speed more quickly than straight and level flight, and the vehicle has a finite energy budget. The pitch-up maneuver at the end places an additional tax on the system capabilities because the vehicle must perform an extra banking maneuver to roll upright before pitching once more, and must also save enough speed to survive this maneuver with the appropriate final conditions.

It is interesting to note that the maximum uprange recovery zone is over 200 km downrange of the insertion point – a phenomenon that is primarily a function of the 0 deg flight path angle insertion. The vehicle's initial velocity is pointed entirely downrange, and the vehicle must roll over before it can pitch downward and dive into the atmosphere. Due to the bank angle rate limit, the time required to roll over is not negligible, so the vehicle covers significant downrange distance before it can redirect

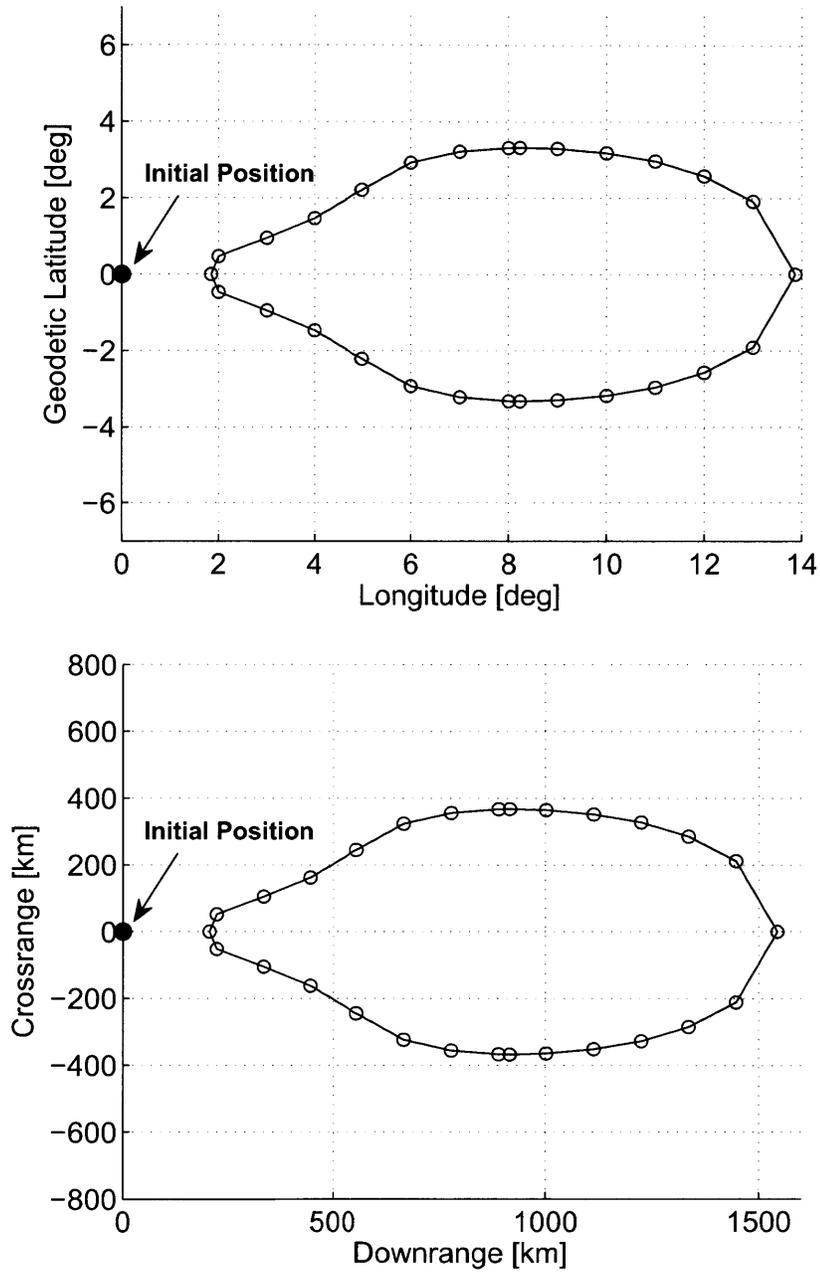


Figure 4-1: Nominal footprints without uncertainty considerations: The figure on top gives the footprint in terms of geodetic latitude and longitude, while the figure on bottom gives it in terms of downrange and crossrange position in kilometers.

its velocity vector downwards. In addition, while much of the footprint has a relatively ellipsoidal shape, the uprange quarter is constricted in a bottleneck shape. Similar behavior is observed in some of Undurti's footprints, particularly in his low- γ reentry scenarios without g-load and heating constraints [17]. However, the elongation is exaggerated here, again due to the vehicle's banking limitations and the extra energy required to perform the additional pitch-up maneuver. Widening the uprange portion of the footprint requires executing the banking maneuvers more quickly to reduce energy dissipation, but the rate limits and extra turning requirements constrain this performance.

The three fundamental trajectories for this footprint – the maximum downrange, uprange, and crossrange solutions – effectively encapsulate the range of characteristics displayed in the full set of footprint border solutions. The altitude profiles given in Fig. 4-2 each demonstrate different skipping behaviors. The maximum downrange trajectory is able to skip twice, staying as high in the atmosphere as possible to minimize energy and speed losses due to drag, and each successive skip reaches a slightly lower altitude as the vehicle bleeds more speed. As required, each trajectory stays above the Earth's surface. The maximum uprange trajectory digs deeper into the atmosphere before the final pitch-up than the maximum crossrange trajectory, and the maximum crossrange trajectory dips farther than the maximum downrange case. This phenomenon occurs because as the trajectory penalizes downrange distance, the dives become steeper to reduce the downrange groundtracks. Nevertheless, the maximum uprange trajectory remains more than 2700 ft above the ground.

Slower speeds activate the lower dynamic pressure limit at lower altitudes where the atmospheric densities are greater. Figure 4-3 plots the dynamic pressure profiles. The lower bounds are met at the peaks of the final skips in both the maximum downrange and uprange cases as the vehicle attempts to minimize drag and increase glide. As expected, the upper dynamic pressure bounds remain inactive. Figure 4-4 provides the speed and flight path angle profiles for each trajectory. Also as expected, the speed profiles decrease monotonically due to nonconservative drag forces acting on the unpowered vehicle. The flight path angle profiles prove that each trajectory meets the final 85 deg requirement. The maximum downrange and crossrange profiles remain near zero during the high-altitude glides, and then decrease to approximately -45 deg during the final dives before the pitch-up maneuver. Steeper dives result in quicker speed losses that curb the trajectory's groundtrack, so unlike the other two trajectories, the maximum uprange case achieves a minimum flight path angle of -65 deg to take advantage of this behavior. The vehicle could dive even more steeply if the aggressiveness of the final pitch-up maneuver were not constrained by angle of attack and angle of attack rate limits.

Figure 4-5 gives the angle of attack and bank angle profiles, which demonstrate the aerodynamic orientations of the vehicle required to fly the respective optimal trajectories. In the maximum downrange and crossrange cases, the vehicle maximizes its glide distance by oscillating gently about its 7 deg trim angle of attack during the high-altitude skips. In fact, for both of these trajectories, the angles of attack approach near-trim conditions as quickly as possible to maximize glide. However, in the maximum uprange case, the angle of attack remains near zero for the first 2.5

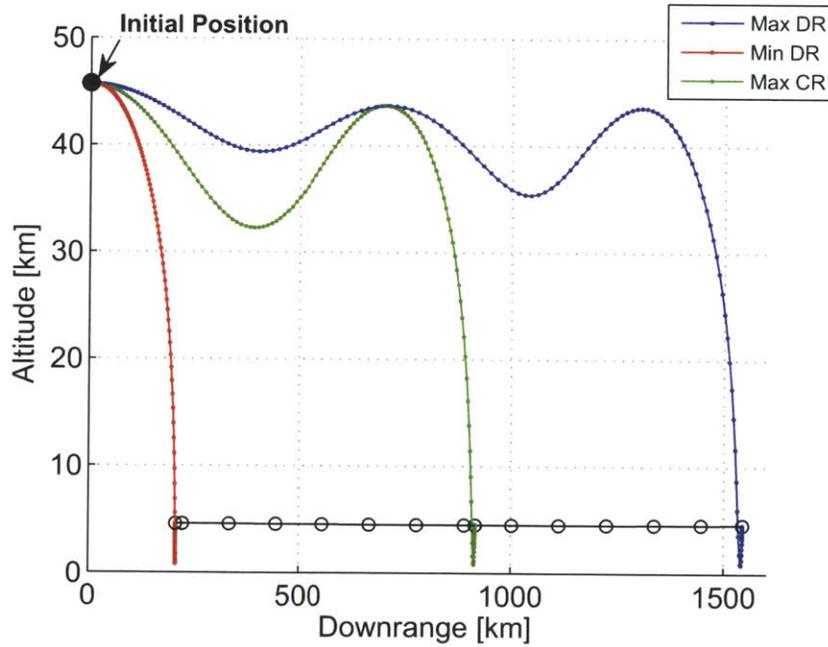
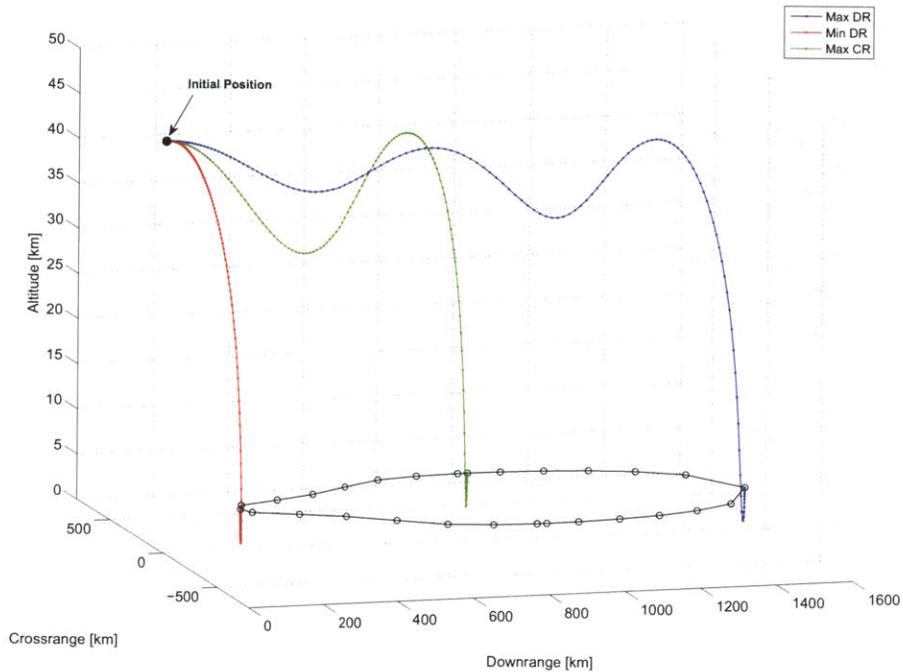


Figure 4-2: Altitude profiles for the maximum downrange, uprange, and crossrange trajectories shown with the footprint boundary: The figure on top displays an angled view in terms of downrange distance, crossrange distance, and altitude above the ellipsoid in kilometers. The figure on bottom shows the same data strictly in terms of downrange distance and altitude. Each trajectory displays different skipping behavior, with the magnitudes of each final peak limited by the dynamic pressure controllability constraint, and each trajectory pulls into the final pitch-up maneuver well before striking the ground.

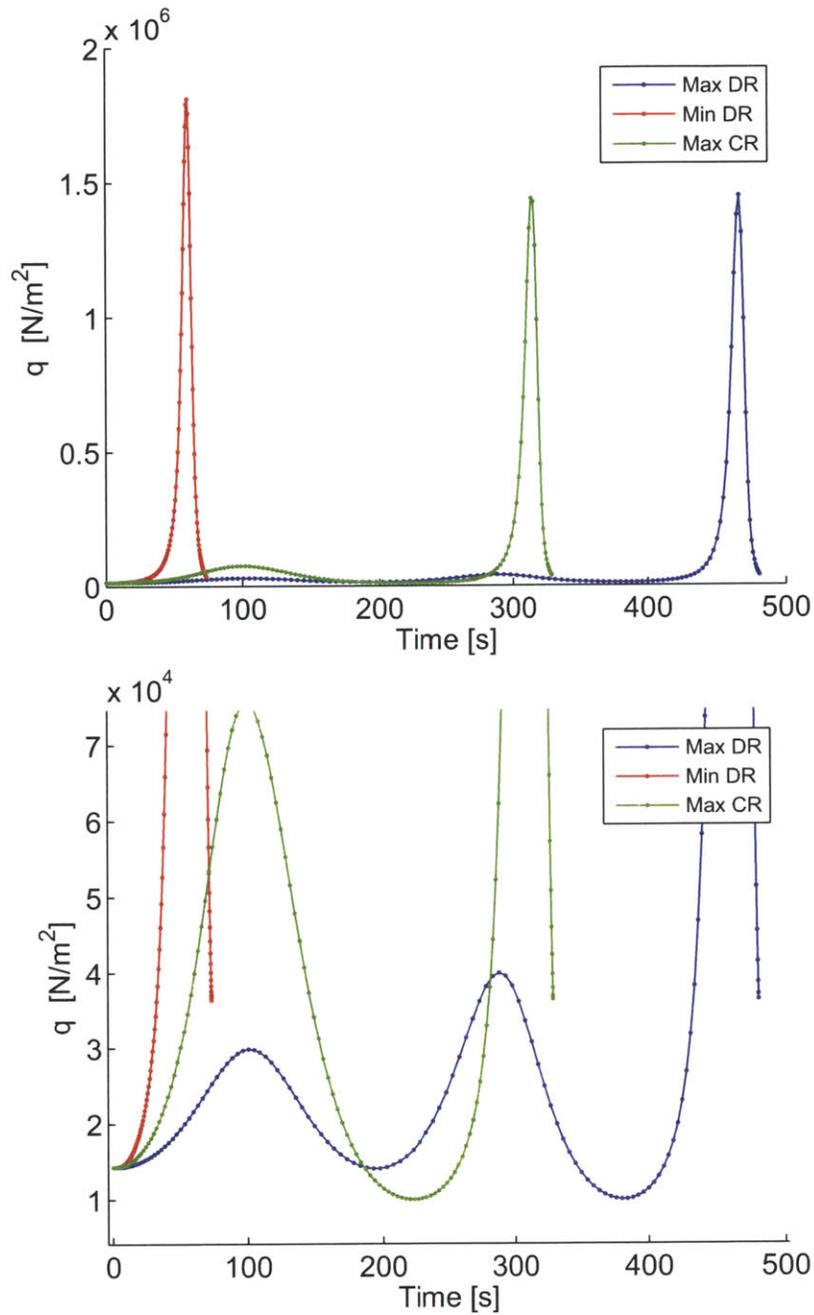


Figure 4-3: Dynamic pressure profiles for the maximum downrange, uprange, and crossrange trajectories: The figure on top displays the entire profile, proving the maximum bound remains inactive throughout the entire trajectories. The figure on bottom displays the same data but focuses on the regions of smaller dynamic pressures, demonstrating that each trajectory satisfies the controllability path constraint, but the lower bound is active at the peak of the final skips in the maximum downrange and crossrange cases.

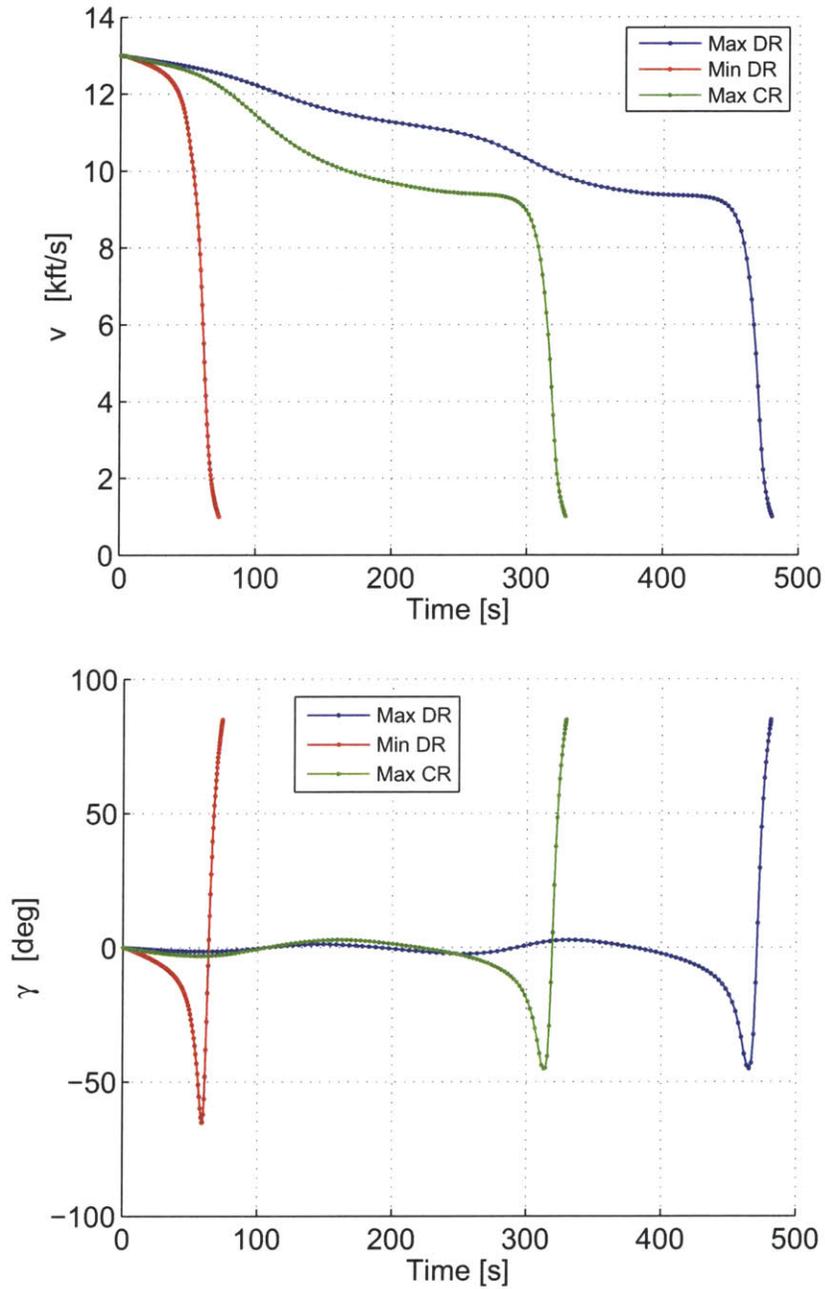


Figure 4-4: Speed and flight path angle profiles for the maximum downrange, uprange, and crossrange trajectories: The velocity profiles decrease monotonically to the terminal conditions, while the flight path angle profiles reflect the skipping, diving, and pitching-up characteristics.

seconds while the bank angle immediately increases to 180 deg. Once this flip occurs, the vehicle pitches into a downward dive. Reducing the angle of attack during the flip minimizes the lift generated in the upward and out-of-plane directions, which prevents unnecessary altitude gains and out-of-plane deviations. Although the maximum uprange trajectory executes this flip immediately, similar flips are observed at the ends of the other two trajectories. After spending significant time near trim during the high-altitude glides, the angles of attack plummet as the vehicle flips over. However, unlike the maximum uprange case, the angle of attack rate limit prevents the angles of attack from reaching zero during the flips along the other two trajectories. In all three cases, once the flip is executed and the dives begin, the angles of attack begin growing again to steepen the dives as much as possible. In the maximum uprange case, the angle of attack activates the maximum bound during much of the dive to maximize the steepness and minimize the downrange distance covered. The maximum downrange and crossrange trajectories also steepen their dives, but not quite as much as the maximum uprange case because downrange motion is still beneficial to them. Finally, for the final pitch-up maneuvers, all three trajectories force the angles of attack back to zero, perform efficient rolls, and then utilize the maximum angle of attack to climb as steeply as possible and reach the required 85 deg final flight path angle.

While the trajectories presented in this section exhibit the vehicle's maximum capabilities for the given plant dynamics, none of these trajectories incorporate uncertainty considerations in their designs. Before such considerations can be included in the designs, though, appropriate system uncertainties must first be identified.

4.3 Model Uncertainties

Of the myriad uncertainties that plague the reentry problem, several important ones will be considered in this demonstration. First, reentry vehicles reach the beginning of their glide phases by means of powered insertions, and the insertion methods are unlikely to deliver the vehicle exactly to the desired position with the intended velocity. Dispersions from the intended reentry conditions can be characterized as random processes with zero mean normal probability distributions, whose standard deviations describe how far from the desired initial conditions the vehicle tends to be delivered. For example, for normal distributions, pure 1σ standard deviation values describe the initial errors within which the vehicle will be delivered with 68.27% confidence, while 3σ values extend the dispersion boundary to demonstrate 99.73% confidence. The standard deviation values depend on the particular insertion method's uncertainties, but this scenario assumes the 3σ confidence regions for each of the initial reentry states are given by Table 4.2, in which the 3σ latitude and longitude values correspond to approximately 3000 m downrange distance error and 1000 m crossrange distance error. With the zero mean assumption, squaring the standard deviations provides the variances of each initial state, and assuming the initial state dispersions are uncorrelated, these variances can be placed along the diagonals of an 8×8 matrix to form an initial covariance matrix.

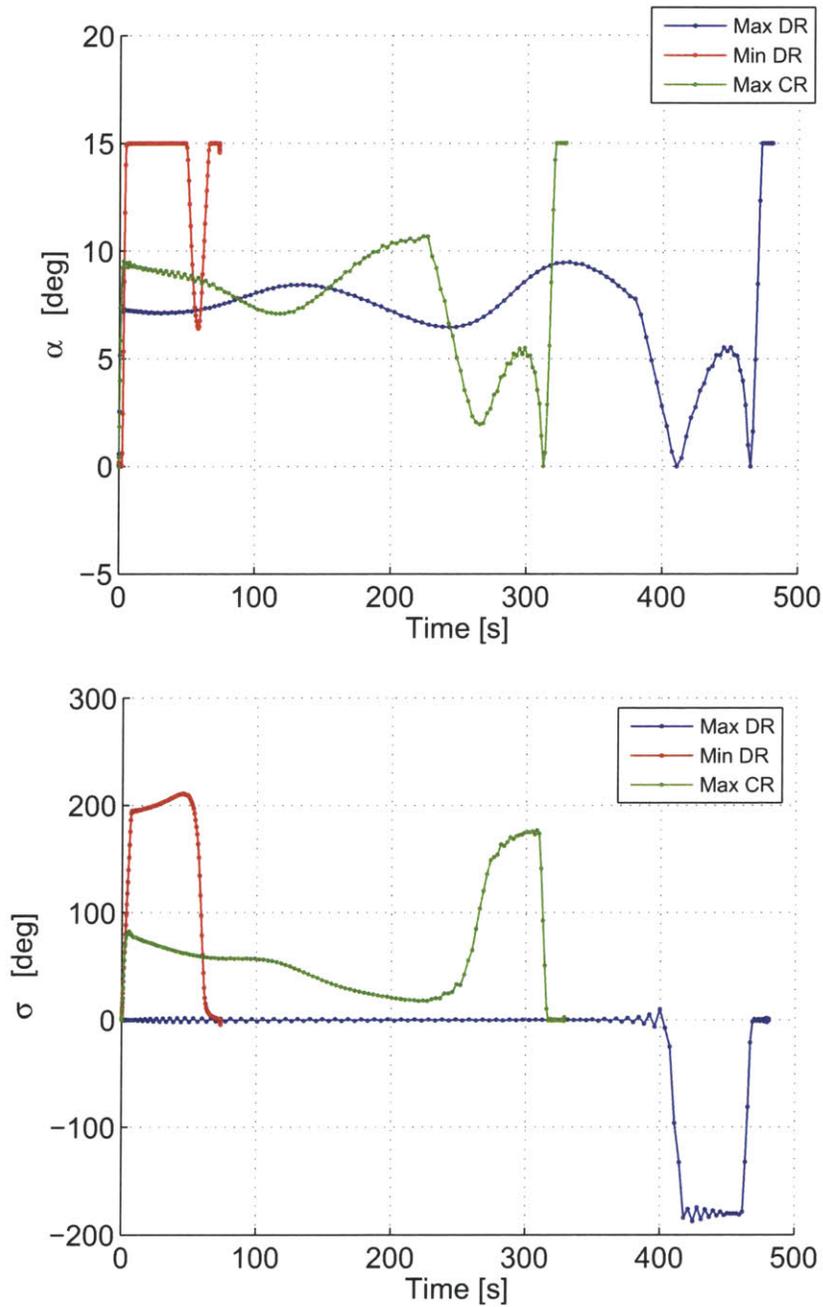


Figure 4-5: Angle of attack and bank angle profiles for the maximum downrange, uprange, and crossrange trajectories: The angle of attack profiles oscillate gently about the trim angle of attack when trying to maximize glide distances, but peg the upper bound when trying to maximize the dive and climb inclinations. The bank angle profiles for the maximum downrange and uprange cases stay as near to 0 and 180° as possible to reduce wasted energy from out-of-plane motion and minimize the banking required to flip the vehicle during dives and climbs, while the maximum crossrange maintains a positive bank to execute its turn.

Table 4.2: 3σ State Insertion Error Assumptions

State	3σ Insertion Errors
r	300 ft
μ	0.027 deg
λ	0.009 deg
v	600 ft/s
γ	3 deg
ψ	1.5 deg
α	0 deg
σ	0 deg

Second, the Earth’s atmosphere is continually expanding and contracting, which means the densities at any given location and altitude are not actually constant. Although models like COESA provide accurate fits to meticulous sets of empirical data, fixed density models never fully capture the atmosphere’s daily, seasonal, and geographic fluctuations. While density errors may be represented with simple bias and process noise models, significant research has shown that uncertainties in the atmosphere can be characterized much more realistically as functions of altitude. Like the COESA model, the code for NASA’s Earth Global Reference Atmospheric Model (GRAM) is too slow and cumbersome to evaluate during each iteration of the optimization routine. However, in addition to providing nominal density profile, this program can generate statistical dispersions about the nominal density model at various altitudes [8]. According to the statistical GRAM density outputs, the standard deviation of the dispersions rises in an exponential fashion until approximately 330,000 ft, after which it begins an exponential decay, and then flattens out at a constant value near 660,000 ft. Since the trajectories in this scenario stay well below 330,000 ft, a simple exponential fit can be applied to the GRAM density standard deviations between 0 and 330,000 ft. The resulting 1σ confidence model for atmospheric density dispersions, given in terms of percentage and as a function of altitude, is written as

$$\sigma_\rho(\mathbf{x}) = 0.003517 \exp\left(\frac{h}{87368 \text{ ft}}\right), \quad (4.56)$$

where h is altitude above the ellipsoid, as defined in Eq. (4.23). Figure 4-6 provides a plot of the atmospheric density 1σ confidence model. Atmospheric density is not a state in the original 3 DOF dynamics, however, so its uncertainties cannot be directly incorporated into the nominal covariance matrix. In addition, the technique explained in Section 2.4 augments a system with constant parameters, but ρ is not a constant in the current dynamics model. To circumvent this problem, the density calculation is rewritten as

$$\rho_{\text{act}} = (1 + C_\rho \sigma_\rho(\mathbf{x})) \rho, \quad (4.57)$$

in which ρ is the nominal atmospheric density given by the COESA fit, C_ρ is a constant parameter, σ_ρ is the standard deviation of the expected density dispersion given as

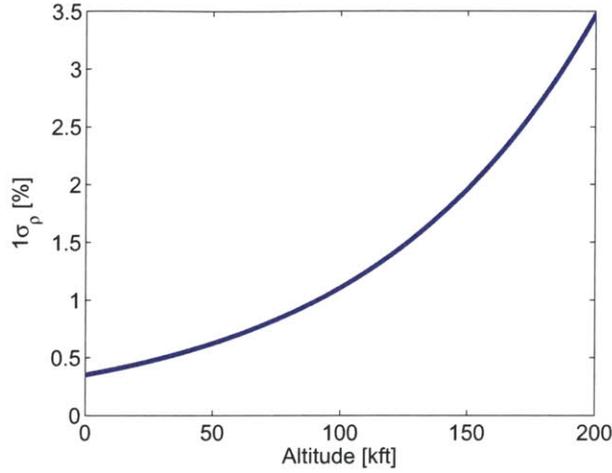


Figure 4-6: Standard deviation model for expected atmospheric density dispersions: An exponential fit is used to model the statistical GRAM density dispersions for the altitudes applicable to the current problem, and the dispersion standard deviations are given as percentages from the nominal values.

a percentage, and ρ_{act} is the realized density after applying a dispersion. Using this calculation, the variable ρ can be assumed to be deterministic and the uncertainty in atmospheric density can instead be captured by making C_ρ stochastic. Since C_ρ is a constant, it can be updated to a state with trivial dynamics. In addition, if C_ρ is given a normal probability distribution with mean

$$E[C_\rho] = 0 \quad (4.58)$$

and constant variance

$$E[C_\rho C_\rho] = 1, \quad (4.59)$$

the nominal atmospheric density calculation remains unchanged, but the atmospheric density uncertainty still gets incorporated into the covariance dynamics, where it impacts the rest of the state errors through correlations.

Third, it is extremely difficult – if not impossible – to develop an aerodynamic model that completely characterizes a vehicle’s performance envelope with perfect precision, despite extensive wind tunnel testing and other modeling efforts. Although vehicle models actually degrade over time as ablation disfigures the original structure and control surfaces, the aerodynamic errors in this scenario are assumed constant. For this particular scenario, it is assumed that the 3σ confidence in the vehicle model axial and normal force coefficients are each 5%, so that the standard deviations of each coefficient can each be written simply as

$$\sigma_{C_X}(\mathbf{x}) = \sigma_{C_N}(\mathbf{x}) = 0.01667 \quad (4.60)$$

However, as with atmospheric density, the axial and normal force coefficients used to

describe the vehicle's aerodynamic properties are not constants, so the axial coefficient calculation can be rewritten as

$$C_{X_{act}} = (1 + C_{C_X} \sigma_{C_X}(\mathbf{x})) C_X, \quad (4.61)$$

and the normal force coefficient can be rewritten as

$$C_{N_{act}} = (1 + C_{C_N} \sigma_{C_N}(\mathbf{x})) C_N, \quad (4.62)$$

where C_X and C_N are the nominal coefficients, C_{C_X} and C_{C_N} are constant parameters, σ_{C_X} and σ_{C_N} are the standard deviations of their expected dispersions given as percentages, and $C_{X_{act}}$ and $C_{N_{act}}$ are the realized force coefficients after applying dispersions. When the stochastic nature of the coefficients is shifted solely to C_{C_X} and C_{C_N} , with normal probability statistics given by

$$E[C_{C_X}] = 0, \quad E[C_{C_N}] = 0, \quad (4.63)$$

$$E[C_{C_X} C_{C_X}] = 1, \quad \text{and} \quad E[C_{C_N} C_{C_N}] = 1, \quad (4.64)$$

the nominal axial and normal force calculations remain unchanged, but the uncertainties given by σ_{C_X} and σ_{C_N} get incorporated into the covariance dynamics where they impact the rest of the state errors through correlations.

Lastly, factors such as ablation and unmodeled high frequency dynamics typically prevent vehicles from executing commanded changes in attitude with perfect precision throughout the trajectory. In this set of dynamics, the attitude rates are control variables, so unmodeled attitude accelerations can be treated as process noise in the angular rate dynamics. Although factors like ablation can easily degrade control authority as a function of time or control effort, this scenario assumes that the 3σ confidence in the angle of attack and bank angle rate dynamics are each approximately 5% of their respective maximum rate limits.

After augmenting the original system to include C_ρ , C_{C_X} , and C_{C_N} as additional states, the augmented state vector for the covariance trajectory-shaping problem becomes

$$\mathbf{x} = \begin{bmatrix} a \\ \mu \\ \lambda \\ v \\ \gamma \\ \psi \\ \alpha \\ \sigma \\ C_\rho \\ C_{C_X} \\ C_{C_N} \end{bmatrix}. \quad (4.65)$$

Applying the uncertainty models for insertion errors, atmospheric density, axial and normal force coefficients, and control error, the diagonal elements of the initial co-

variance matrix can be written as

$$\text{diag}(P_0) = \begin{bmatrix} 100^2 \text{ ft}^2 \\ 0.009^2 \text{ deg}^2 \\ 0.003^2 \text{ deg}^2 \\ 200^2 \text{ ft}^2/\text{s}^2 \\ 1^2 \text{ deg}^2 \\ 0.5^2 \text{ deg}^2 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 30.48^2 \text{ ft}^2 \\ 0.0001571^2 \text{ rad}^2 \\ 0.00005236^2 \text{ rad}^2 \\ 60.96^2 \text{ m}^2/\text{s}^2 \\ 0.01745^2 \text{ rad}^2 \\ 0.008727^2 \text{ rad}^2 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (4.66)$$

and the diagonal elements of the process noise matrix can be written as

$$\text{diag}(Q) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.15^2 \text{ deg}^2/\text{s}^2 \\ 0.5^2 \text{ deg}^2/\text{s}^2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.002618^2 \text{ rad}^2/\text{s}^2 \\ 0.008727^2 \text{ rad}^2/\text{s}^2 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (4.67)$$

Finally, in order to evaluate the covariance matrix dynamics, the system must be linearized along the nominal trajectory. For this problem, each partial derivative is derived analytically and then verified numerically for accuracy, as shown in Appendix A. Each required element of the Riccati equation is now available, so the covariance can be propagated along any nominal trajectory and added to the optimal control problem formulation to directly penalize and constrain expected errors along the reference.

4.3.1 Covariance Propagation vs. Discretization

While direct pseudospectral collocation methods generally exhibit very good convergence speeds and radii, large numbers of design variables easily overwhelm and render them useless. The DIDO direct collocation method employed here easily handles the nominal reentry problem, which contains 1 constant parameter (S_t), 2 control variables, 8 states, and 8 differential constraints. However, adding the covariance matrix requires the definition of 66 more states to capture each unique element of the full 11×11 covariance matrix, each with an extra differential constraint of its own. On top of that, each covariance state must be scaled appropriately, which proves extremely

difficult because the magnitudes of the covariance elements vary tremendously both over time and with respect to each other.

However, the structure of the direct collocation method provides a relatively simple solution to its own problem. Rather than discretizing each element of the covariance matrix and including the Riccati equation as a differential constraint, the covariance can simply be propagated forward during each cost or constraint evaluation [14]. The state dynamics do *not* depend on the covariance matrix, and while the covariance matrix *does* depend on the state and control values at each point in time, these values are easily obtained from the nominal problem formulation. While propagating the covariance makes each cost and constraint evaluation significantly more complicated and time-consuming, the problem can be solved without increasing the number of collocation design parameters, and this trade turns out to be supremely worthwhile for complicated problems. In Chapter 3, the Zermelo trajectories were each successfully shaped by discretizing and scaling the covariance and sensitivity states appropriately, but this proves highly unsuccessful for the much more complex reentry problem. Revisiting the Zermelo problem with propagated covariance metrics, the propagated and discretized methods are shown to in fact provide equivalent results. Thus, every reentry trajectory shaped with covariance considerations in the subsequent sections is solved by propagating the covariance within the cost and constraint evaluations.

At this stage, it is worth providing a few more details about the specific propagation algorithm used within the cost and constraint functions. MATLAB provides several variable-step integrators, which provide high accuracy but relatively long run times. Since the cost and constraint functions are often called thousands of times by DIDO before reaching an extremal solution, the time required for MATLAB's variable-step integrators to propagate the covariance matrix along an entire reentry trajectory proves unacceptable. Although not available as part of MATLAB's toolbox, simple 4th-order fixed-step Runge Kutta algorithms provide vastly superior speed, retain sufficient accuracy, and allow the designer to manually alter the time steps and trade one for the other if necessary. Given a general set of dynamics

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (4.68)$$

and an initial condition

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad (4.69)$$

the 4th order Runge Kutta method propagates the states forward in time according to

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{6}\Delta t (k_1 + 2k_2 + 2k_3 + k_4), \quad (4.70)$$

where Δt is the desired time step and

$$k_1 = \mathbf{g}(\mathbf{x}_n, \mathbf{u}_n, t_n), \quad (4.71)$$

$$k_2 = \mathbf{g}\left(\mathbf{x}_n + \frac{1}{2}k_1\Delta t, \mathbf{u}_n + \frac{1}{2}k_1\Delta t, t_n + \frac{1}{2}\Delta t\right),$$

$$k_3 = \mathbf{g} \left(\mathbf{x}_n + \frac{1}{2}k_2\Delta t, \mathbf{u}_n + \frac{1}{2}k_2\Delta t, t_n + \frac{1}{2}\Delta t \right),$$

and

$$k_4 = \mathbf{g} (\mathbf{x}_n + k_3\Delta t, \mathbf{u}_n + k_3\Delta t, t_n + \Delta t) .$$

For the covariance propagation problem, \mathbf{x} contains the elements of the covariance matrix, \mathbf{g} is the Riccati equation, and \mathbf{u} is the set of nominal states and controls. The nominal states and controls can be obtained at each necessary time in several ways, but the most effective way seems to be interpolating the discretized state and control solution, particularly with a spline. Propagating the nominal states simultaneously with the Runge Kutta method seems to encourage the optimization routine to select choppy control variables, degrade their interpolation, and corrupt the nominal state propagation in a way that artificially lowers the covariance metric. Interpolating the states rather than propagating them significantly reduces this behavior and more closely aligns the propagated covariance in each iteration to the current state and control solutions.

4.4 Maximum Range Confidence

Now that the reentry uncertainties and covariance dynamics have been established, the covariance can be propagated along each of the trajectories comprising the nominal footprint boundary. However, an appropriate representation of the final position dispersions must be extracted and linked to the range capabilities demonstrated by the nominal footprint. The circular error probable (CEP) provides an intuitive and useful representation.

4.4.1 Circular Error Probable (CEP)

At the terminal conditions, the 2×2 submatrix containing the spherical latitude and longitude covariances can be extracted from the full matrix. This submatrix defines the final latitude-longitude error ellipse at the terminal conditions, and its eigenvalues and eigenvectors determine the lengths and orientations of the principal axes, respectively. Although the ellipse itself provides a measure of the final two-dimensional position confidence at the terminal altitude, the circular error probable provides a more intuitive graphical representation of confidence regions, which proves particularly useful for depicting the nominal footprint regression. Given any two-dimensional error ellipse, the CEP calculation provides the equivalent circular radius which encloses 50% of the expected errors, and using an appropriate application of the chi-square probability distribution, this circle can be scaled to enclose any desired probability.

While the CEP can be calculated in terms of spherical latitude and longitude directly from the state covariance submatrix, it can also be desirable to calculate the CEP in terms of geodetic latitude and longitude as well as UEN position. However, these CEP calculations first require transforming the state covariance matrix into

covariances of the geodetic and UEN position variables by utilizing Property 4 defined in Section 2.3.2. Longitude is equivalent in the spherical and ellipsoidal Earth models, but geodetic latitude is approximated from spherical latitude according to

$$\lambda_{\text{geo}} = \lambda + f \frac{R_e}{r} \sin(2\lambda) + f^2 \frac{R_e}{r} \sin(4\lambda) \left(\frac{R_e}{r} - \frac{1}{4} \right). \quad (4.72)$$

The matrix for transforming the state covariance into geodetic latitude and longitude covariance is the 2×11 matrix containing the partial derivatives of geodetic latitude and longitude with respect to each of the spherical states. This matrix can thus be written as

$$A_{\text{geo}} = \begin{bmatrix} 0 & 1 & 0 & \vdots & \mathbf{0} \\ \frac{\partial \lambda_{\text{geo}}}{\partial r} & 0 & \frac{\partial \lambda_{\text{geo}}}{\partial \lambda} & \vdots & \mathbf{0} \end{bmatrix}, \quad (4.73)$$

where

$$\frac{\partial \lambda_{\text{geo}}}{\partial r} = -f \frac{R_e}{r^2} \sin(2\lambda) - f^2 \frac{R_e}{r^2} \sin(4\lambda) \left(\frac{R_e}{r} - \frac{1}{4} \right) - f^2 \frac{R_e}{r} \sin(4\lambda) \frac{R_e}{r^2} \quad (4.74)$$

and

$$\frac{\partial \lambda_{\text{geo}}}{\partial \lambda} = 1 + 2f \frac{R_e}{r} \cos(2\lambda) + 4f^2 \frac{R_e}{r} \cos(4\lambda) \left(\frac{R_e}{r} - \frac{1}{4} \right). \quad (4.75)$$

The geodetic latitude and longitude covariance matrix can finally be written as

$$P_{\text{geo}} = A_{\text{geo}} P A_{\text{geo}}^T, \quad (4.76)$$

where P is the full 11×11 spherical state covariance matrix. As expected, since the difference between spherical and geodetic latitudes are slight, numerical differences in the transformed covariance are small.

Similarly, the matrix for transforming the state covariance matrix into East and North position covariance is the 2×11 matrix containing the partial derivatives of the East and North position vector coordinates with respect to each of the spherical states. This calculation can be broken down into two steps, the first of which transforms the spherical state covariance into ECEF position covariance, and the second of which transforms the ECEF position covariance into UEN position covariance. First, the vehicle's ECEF position vector in terms of the spherical states can be written as

$$\begin{bmatrix} x_{\text{ECEF}} \\ y_{\text{ECEF}} \\ z_{\text{ECEF}} \end{bmatrix} = \begin{bmatrix} r \cos(\mu) \cos(\lambda) \\ r \sin(\mu) \cos(\lambda) \\ r \sin(\lambda) \end{bmatrix}, \quad (4.77)$$

so the 3×11 matrix for transforming the spherical covariance to ECEF position

covariance can be written as

$$A_{\text{ECEF}} = \begin{bmatrix} \cos(\mu) \cos(\lambda) & -r \sin(\mu) \cos(\lambda) & -r \cos(\mu) \sin(\lambda) & \vdots & \mathbf{0} \\ \sin(\mu) \cos(\lambda) & r \cos(\mu) \cos(\lambda) & -r \sin(\mu) \sin(\lambda) & \vdots & \mathbf{0} \\ \sin(\lambda) & 0 & r \cos(\lambda) & \vdots & \mathbf{0} \end{bmatrix}. \quad (4.78)$$

Second, the matrix for transforming the ECEF position covariance to UEN position covariance is simply the transpose of the rotation matrix given by Eq. (4.3), which is written as

$$A_{\text{UEN}} = \begin{bmatrix} \cos(\lambda) \cos(\mu) & -\sin(\mu) & -\sin(\lambda) \cos(\mu) \\ \cos(\lambda) \sin(\mu) & \cos(\mu) & -\sin(\lambda) \sin(\mu) \\ \sin(\lambda) & 0 & \cos(\lambda) \end{bmatrix}. \quad (4.79)$$

The 3×3 UEN position covariance matrix can finally be calculated as

$$P_{\text{UEN}} = A_{\text{UEN}} A_{\text{ECEF}} P A_{\text{ECEF}}^T A_{\text{UEN}}^T, \quad (4.80)$$

and the bottom right 2×2 submatrix of P_{UEN} provides the East-North position covariance. The CEP for geodetic latitude and longitude dispersions can now be calculated using P_{geo} , and the CEP for East and North position dispersions can be calculated using the bottom right 2×2 submatrix of P_{UEN} .

4.4.2 Shaping to Reduce CEP

In the Zermelo problem results, reference trajectories shaped with open-loop covariance still improved expected errors when applied in closed-loop environments, although the reverse did not appear to be true. For this reason, and also because a linear feedback control algorithm has yet not been designed for this reentry scenario, open-loop rather than closed-loop covariance is used to characterize and shape reentry trajectories.

Before shaping new trajectories, the open-loop covariance is propagated along each existing trajectory in the footprint boundary, and CEP radii are calculated and scaled to enclose 90% of the expected position dispersions at the final location. When the 90% circular confidence regions are superimposed along the original footprint, the band of rings intersect and form a new outer boundary enclosing all recovery zones achievable with 90% confidence, as shown in Fig. 4-7. As expected, the 90% confident footprint encompasses a significantly smaller area than the nominal footprint. In fact, the swath of recovery zones achievable with 90% confidence is approximately 190 km (14%) shorter in the downrange direction, and about 220 km (30%) shorter in the crossrange direction. In addition, while the nominal footprint's uprange bottleneck conveys the vehicle's limitations regarding quick crossrange maneuvers with steep dives, the CEPs demonstrate that these maneuvers also induce higher dispersion variances. In fact, the boundary trajectories with the largest CEPs are located in the uprange half of the footprint. Although the downrange trajectories have longer

flight times that allow the control command process noise to accumulate longer, quick banking maneuvers and steep descents generate larger dispersions. For instance, the maximum uprange trajectory has a shorter flight time than several of its neighboring boundary trajectories, but it executes a steeper banks and dives, and generates greater dispersion variances.

The nominal footprint perimeter is characterized by trajectories with various CEPs, but in reality it may be desirable to define a maximum allowable CEP and find the footprint that defines the vehicle's maximum capabilities while satisfying this condition. Discovering which recovery sites the vehicle can reach within certain error bounds can be instrumental in designing test flights that ensure safe and efficient vehicle recovery. When the problem is augmented to include the covariance matrix, the cost metrics and constraints may be any function of the covariance, including CEP. A new footprint can then be solved subject to the final dispersion constraint. Along the nominal footprint, the smallest and largest 90% CEPs, respectively, are 0.00995 deg and 0.0265 deg in geodetic latitude and longitude coordinates, or 63.35 km and 169.0 km in UEN coordinates. When the problem is augmented to include the covariance matrix and an additional event constraint is developed to limit the 90% CEP to 0.01 deg (approximately 63.37 km), the trajectory-shaping process produces a new footprint. As expected, the added constraint shrinks the new footprint completely inside the original footprint. However, the new 90% confident range actually extends *farther* than the nominal 90% confident range for much of the uprange portion – in some places up to 80 km. Figure 4-8 compares both footprints and their 90% confident vehicle ranges. By reining in the intended recovery sites and reshaping the trajectories, the useful range of the vehicle can be increased, and the covariance-trajectory shaping technique allows the designer to directly quantify and solve for range extensions that are impossible to discover with the nominal problem formulation.

4.5 Fixed-Recovery Site Precision

Now, rather than allowing the final recovery zone to remain open, a fixed site is selected within the vehicle's footprint, and two distinct goals are elaborated for reaching it. In the first, the vehicle must reach the recovery zone with the greatest precision possible. The particular trajectory flown does not matter, as long as it produces the smallest possible final position errors. In the second, the vehicle is tasked with performing the largest crossrange turning maneuver possible. However, a constraint is placed on the final position error so that the vehicle reaches the recovery zone within a specified confidence region. The recovery zone location is chosen at a geodetic latitude of 1.5 deg and a longitude of 9 deg, which correspond to a downrange distance of 1000 km and a crossrange distance of 167.8 km from the insertion point, well within the 90% confidence range of the vehicle. Converting the final latitude from geodetic to spherical at the terminal altitude, the final position states can be fixed to

$$\mu_f = 9 \text{ deg} = 0.1571 \text{ rad} \quad (4.81)$$

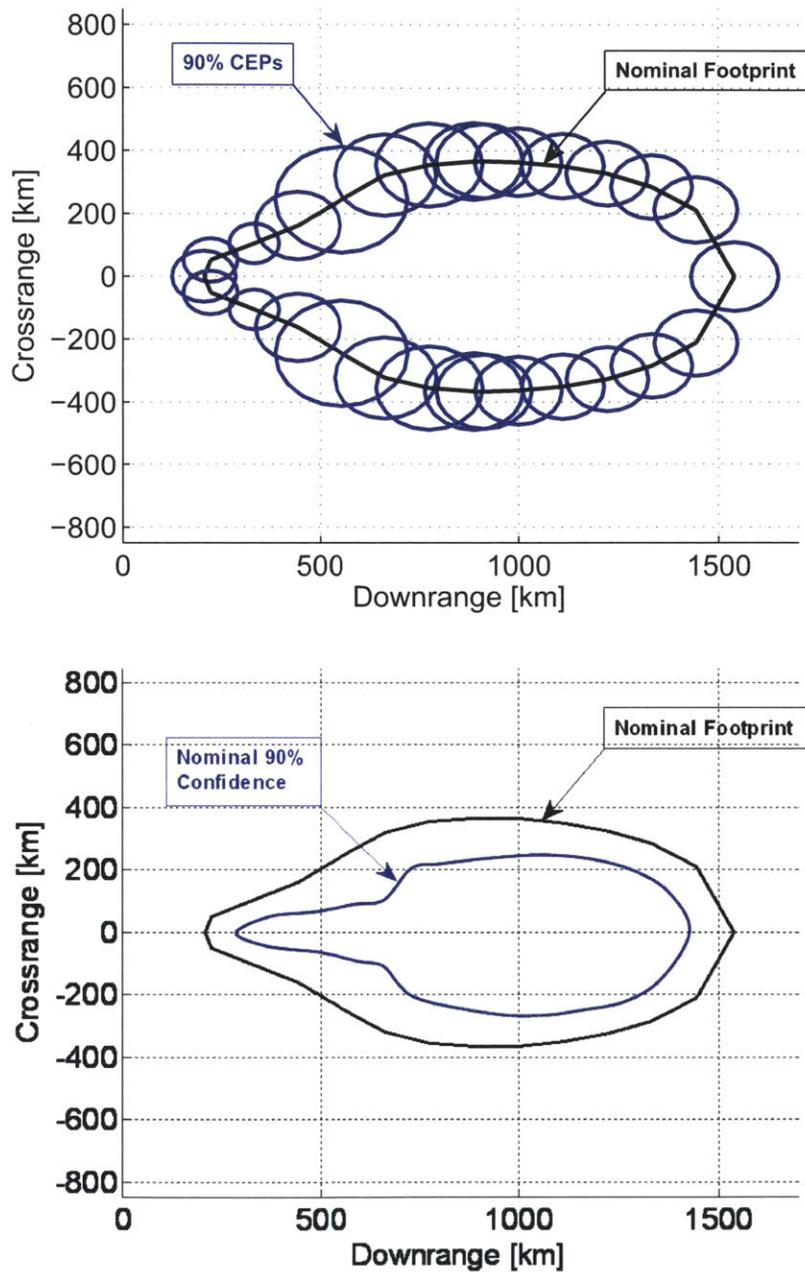


Figure 4-7: Nominal footprint with CEPs: The top figure shows the nominal footprint with 90% CEPs for each boundary trajectory depicted in blue. The bottom figure shows the nominal footprint once again as the outer boundary, with the inner boundary representing the 90% confident footprint obtained from the union of rings.

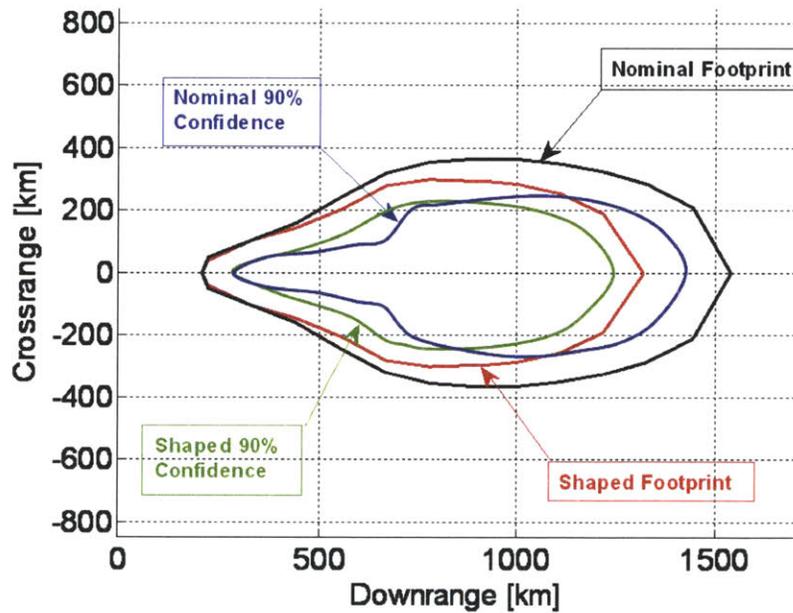
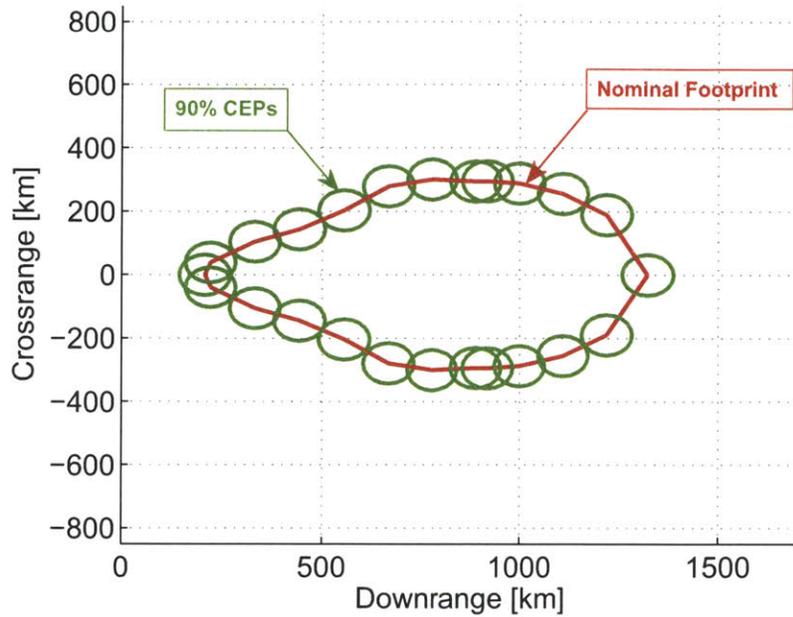


Figure 4-8: Nominal and shaped footprints with CEPs: The top figure shows the newly shaped footprint with the 90% CEP constrained to be below 63.67 km. The bottom figure compares the nominal and shaped footprints, along with their respective 90% confidence footprints. The nominal footprint is given in black with its 90% confidence region defined in black, while the newly shaped footprint is drawn in red with its 90% confidence region depicted in green. Even though a cost metric was not formulated specifically for this purpose, the CEP constraint actually extends the vehicle's 90% confident range in the uprange half, demonstrated where the green footprint lies outside the blue footprint.

and

$$\lambda_f = 1.490 \text{ deg} = 0.0260 \text{ rad}, \quad (4.82)$$

and these conditions will be used for both the maximum precision and maximum turning capability scenarios.

4.5.1 Maximum Precision

To form a basis from which to judge improvements in targeting precision given by the covariance trajectory-shaping technique, two reference trajectories are solved using purely nominal metrics. One reasonable cost metric seeks to minimize the accumulated control exertion, where control effort is defined as commanded angular rates and deviations from the trim angle of attack. Normalizing the penalties on each variable according to their upper limits, this metric can be written as an integral of squares such that

$$J = \int_{t_0}^{t_f} \left\{ \left(\frac{\alpha - \alpha_{trim}}{\alpha^U} \right)^2 + \left(\frac{\dot{\alpha}}{\dot{\alpha}^U} \right)^2 + \left(\frac{\dot{\sigma}}{\dot{\sigma}^U} \right)^2 \right\} dt, \quad (4.83)$$

where α^U , $\dot{\alpha}^U$, and $\dot{\sigma}^U$ represent the upper bounds on each respective variable. Omitting the penalty on deviations from the trim condition yields an alternative metric written as

$$J = \int_{t_0}^{t_f} \left\{ \left(\frac{\dot{\alpha}}{\dot{\alpha}^U} \right)^2 + \left(\frac{\dot{\sigma}}{\dot{\sigma}^U} \right)^2 \right\} dt, \quad (4.84)$$

which minimizes the vehicle's rotations by simply penalizing any changes in attitude.

Since these two nominal metrics do not directly penalize expected errors, it is unclear at first whether minimum control effort actually translates to minimum position errors. However, using the covariance trajectory-shaping technique, a cost metric is formulated which propagates the state covariance matrix to the terminal conditions, transforms the final state covariance into UEN position covariance and penalizes its trace to improve altitude, downrange, and crossrange errors simultaneously. In spherical coordinates, the latitude and longitude variances are given in rad^2 , so they have significantly smaller orders of magnitude than the altitude variance given in m^2 . The final position covariance is transformed to UEN coordinates so that the units are equivalent along each axis when the trace is taken.

The two trajectories produced with the nominal metrics have similar state and covariance profiles. In fact, their final CEPs are almost identical at the recovery site. Both trajectories skip once, meet the dynamic pressure limit at the peak, and then dive straight down until the final pitch-up maneuver for recovery. Their altitude, velocity, flight path angle, and heading profiles demonstrate only slight differences, although greater variations are noticed in the angle of attack and bank angle profiles. The trajectory generated with trim deviation penalties stays very near 7 deg for the duration of the flight until the dive and pitch-up maneuvers demand otherwise. This

is the only trajectory obtained so far that does not maximize its angle of attack during the final pitch-up maneuver. Instead, it stays closer to the trim condition, which means the vehicle plunges within 800 ft of the ground before climbing back upwards. The trajectory generated without trim deviation penalties demonstrates a more dynamic angle of attack profile, which slightly softens vehicle's maneuvers. For instance, during the final dive, its flight path angle stays above -35 deg, while the other nominal trajectory drops more steeply at -45 deg.

The trajectory which minimizes the trace of the final UEN position covariance demonstrates markedly different behavior, which reduces the final CEP from a radius of over 96 km to a radius of under 49 km – an improvement of more than 48%. Figures 4-8 through 4-11 compare the downrange-crossrange, altitude, speed, flight path angle, angle of attack, and bank angle profiles of the minimum covariance and minimum control effort trajectories. The altitude profile of the shaped trajectory still executes one major skip, but chooses to dig into the atmosphere sooner than the other trajectories, and remains below the dynamic pressure limit at its peak. When it dives toward the ground on its final approach, it does not take as direct a path to the recovery site. The vehicle pitches slightly near 30 km altitude, resulting in a notch in the altitude profile, before continuing a direct plunge until the pull-up recovery maneuver.

Since the vehicle dips into the atmosphere sooner, it loses speed earlier than the other trajectories, but still reaches the terminal conditions at the correct velocity. The flight path angle also reflects the different character of the skip, and the notch during the final dive is seen as a temporary increase in the flight path angle before it tips over to nearly -45 deg. The shaped downrange-crossrange groundtrack also exhibits slight heading angle oscillations as it travels toward the recovery zone, a behavior reflected in the bank angle profile. The bank angles of the two minimum effort trajectories remain positive throughout the entire flight, meaning the heading angle increases monotonically and consistently turns the vehicle counterclockwise. However, the bank angle of the covariance-shaped trajectory undulates, and actually goes negative twice during the flight, causing the vehicle to oscillate slightly between counterclockwise and clockwise turns. The shaped angle of attack profile also exhibits more dynamic behavior to reflect the rises and falls in the altitude and flight path angle profiles.

The covariance trajectory-shaping technique designs trajectories that take advantage of the natural system dynamics to reduce expected dispersions in ways that would be virtually impossible to infer from simply looking at the equations of motion.

4.5.2 Maximum Crossrange Turn

Since the vehicle's initial heading points along the equator, the size of its turn can be characterized by how far south the vehicle travels before heading back toward the intended recovery site. However, endpoint cost metrics may only be posed in terms of the initial and terminal states, and since the point of maximum turn may occur anywhere along the trajectory, the single-segment problem formulation is inadequate to handle this design. Instead, the reentry problem must be broken into two segments

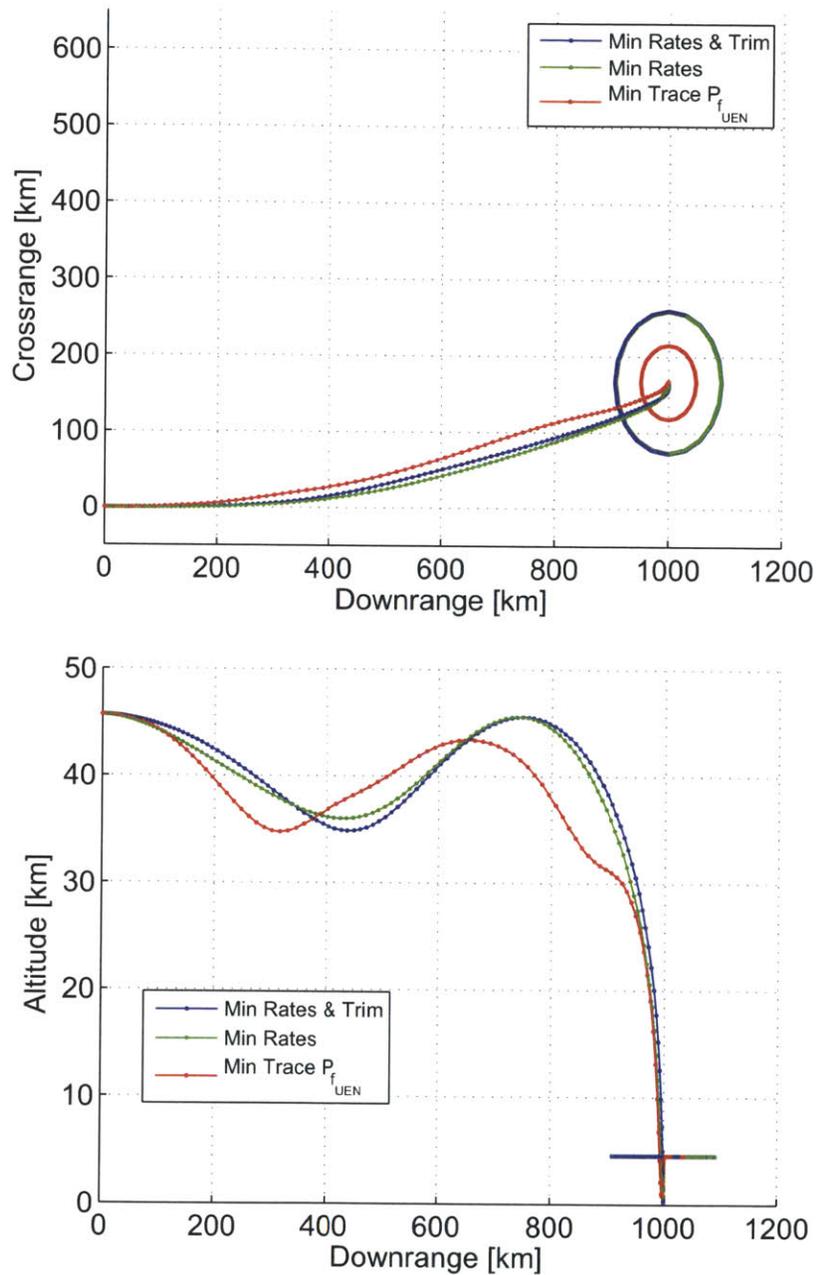


Figure 4-9: Downrange-crossrange and altitude profiles for the maximum precision fixed recovery zone scenario: The top figure shows downrange vs crossrange in kilometers, with the 90% CEPs plotted at the final location. The bottom figure shows altitude vs downrange in kilometers. The blue trajectory minimizes a combination of aerodynamic angular rates and deviations from trim, the green trajectory uses the same metric but omits the trim deviation penalty, and the red trajectory minimizes the trace of the final UEN position covariance.

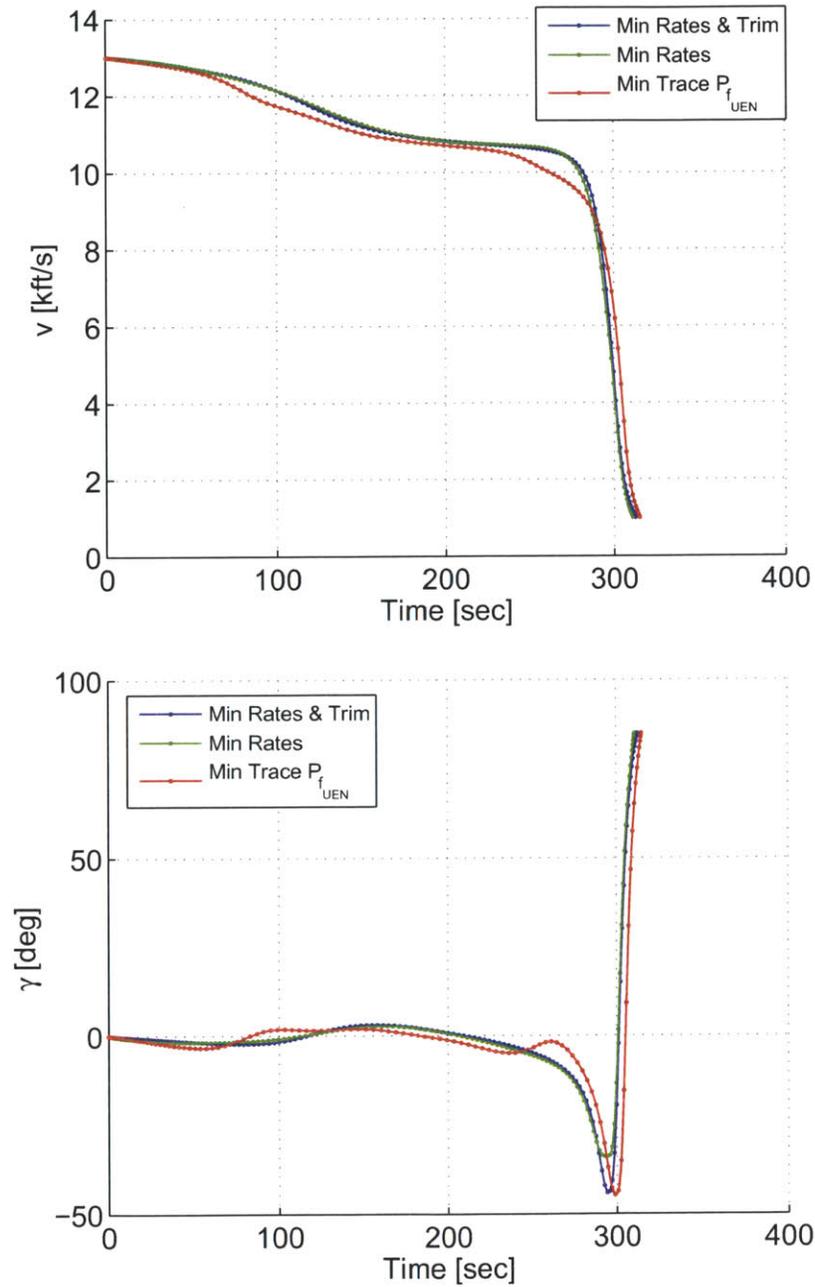


Figure 4-10: Speed and flight path angle profiles for the maximum precision fixed recovery zone scenario: The top figure shows speed vs time, while the bottom figure shows flight path angle vs time. The trajectory definitions remain from before.

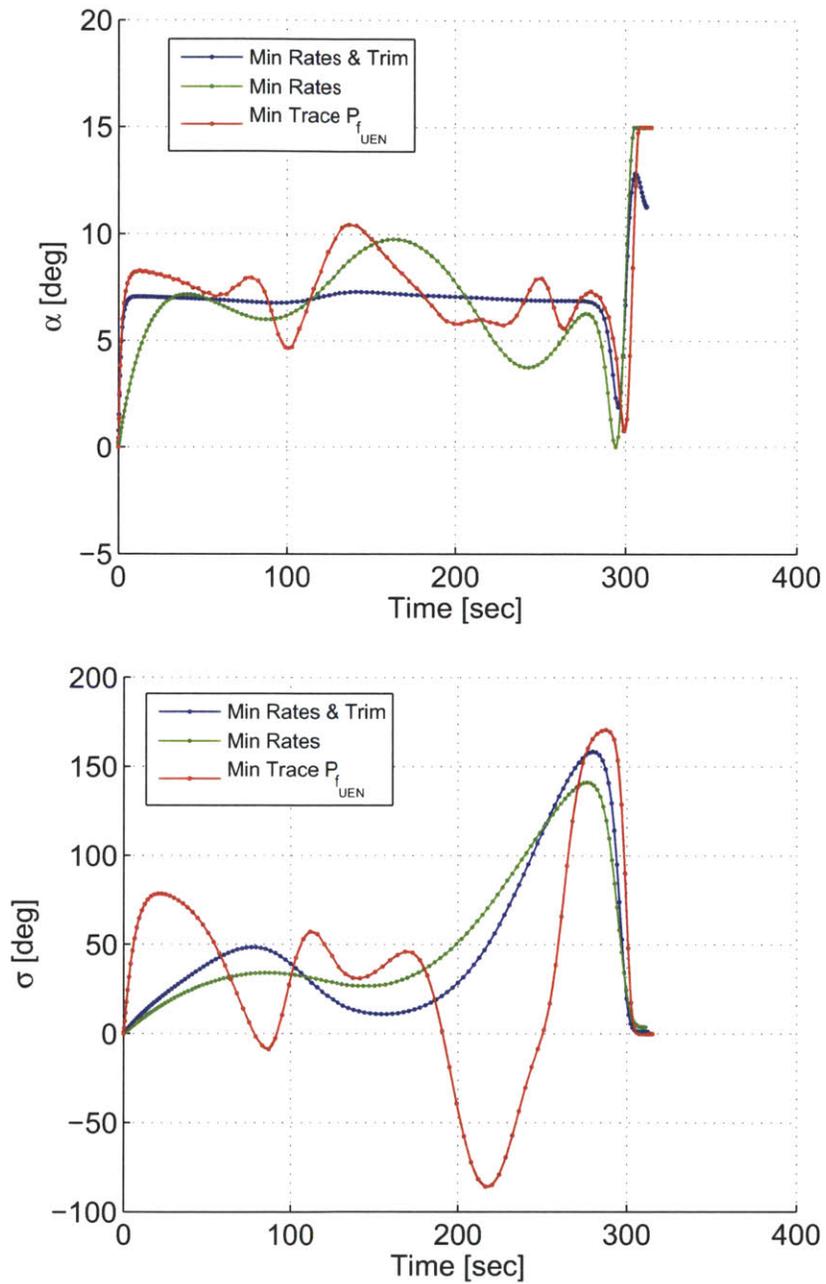


Figure 4-11: Angle of attack and bank angle profiles for the maximum precision fixed recovery zone scenario: The top figure shows angle of attack vs time, while the bottom figure shows bank angle vs. time. The trajectory definitions remain from before.

to be solved simultaneously. The first segment retains the initial reentry conditions defined for the single segment problem, but each final state is allowed to vary freely. Similarly, the second segment retains the fixed-site recovery conditions, but its initial states are permitted to vary freely. To ensure both segments properly connect a single continuous flight, a set of event constraints are generated to enforce continuity. The event is defined as

$$\mathbf{x}_1(t_f) - \mathbf{x}_2(t_0) = 0, \quad (4.85)$$

where \mathbf{x}_1 is the set of states describing the first segment, and \mathbf{x}_2 is the set of states describing the second segment. The dynamic constraints are equivalent for both segments, but the event ensures that the second segment begins where the first begins. In the two-segment problem, the maximum turn can be solved with the cost metric

$$J = \lambda_1(t_f), \quad (4.86)$$

which minimizes the latitude at the end of the first segment.

In the resulting solution, the vehicle's maximum turn carries it approximately 26.6 km south of the equator, before turning north towards the recovery zone. However, the 90% CEP radius at the end is over 107 km. When the same maximum turn metric is used with a maximum allowable CEP of 0.01 deg (corresponding to 63.37 km), the resulting solution only reaches 12.2 km south of the equator. However, the solution satisfies the required position error constraint, improves the 90% CEP by over 40%, and provides the largest possible turn given the problem. Figure 4-12 shows the downrange-crossrange and altitude plots for both trajectories.

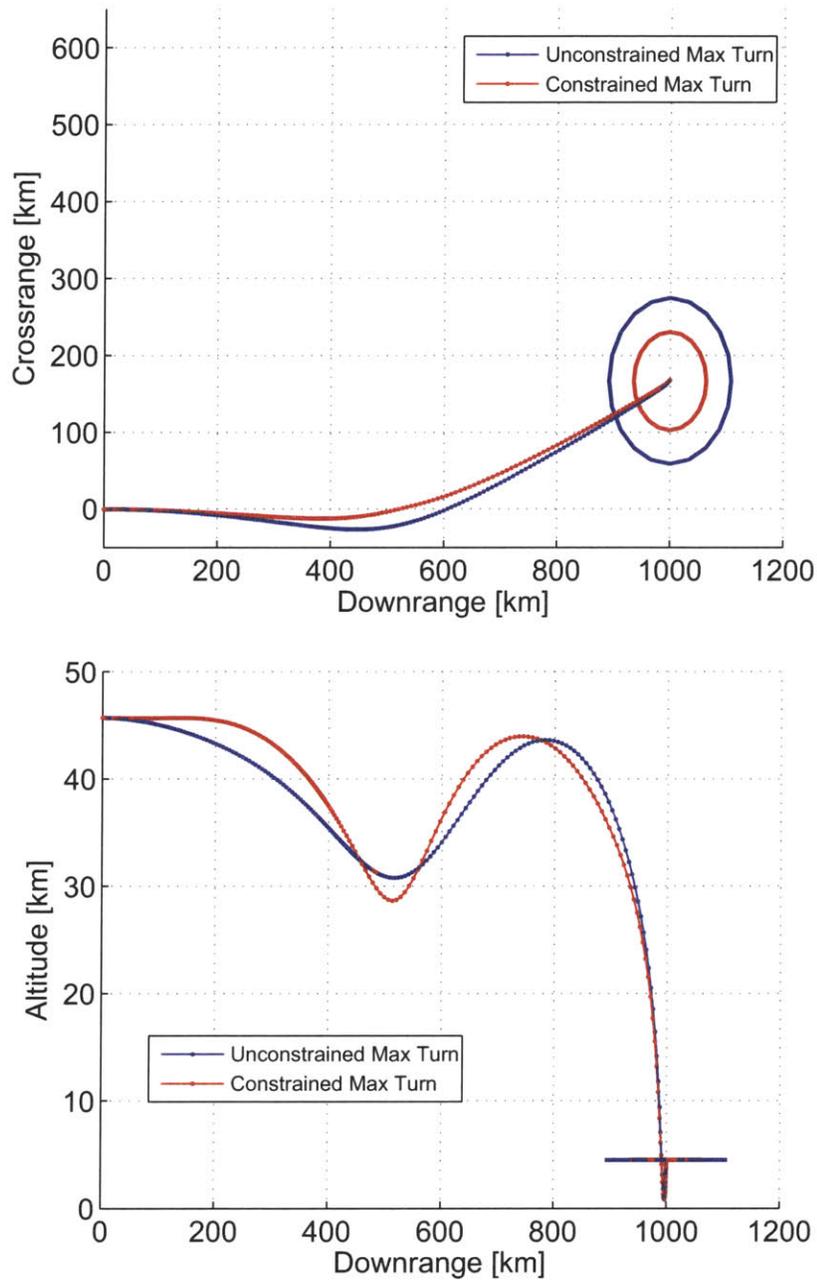


Figure 4-12: Downrange-crossrange and altitude profiles for the maximum turn fixed recovery zone scenario: The top figure shows downrange vs crossrange in kilometers, with the 90% CEPs plotted at the final location. The bottom figure shows altitude vs downrange in kilometers. The blue trajectory provides the maximum turn without incorporating uncertainty considerations, while the red trajectory uses the same metric but constrains the final CEP. The constrained solution does not turn as far south, but it represents the best solution given the requirements.

Chapter 5

Conclusions

Ultimately, the covariance trajectory-shaping technique successfully improved the robustness of hypersonic reentry trajectories to various types of uncertainty. For the posed reentry flight profile and recoverable vehicle, the covariance trajectory-shaping technique actually *extended* the performance confidence of the vehicle up to 80 km in some regions of the footprint, when subject to uncertainties in atmospheric density, axial and normal force coefficients, commanded attitude, and initial position and velocity. At a fixed recovery site within the footprint, and subject to the same uncertainties, the covariance trajectory-shaping technique improved CEP radius by nearly 50% when compared to the minimum control effort trajectories obtained with the nominal optimization process. Finally, covariance trajectory-shaping allowed the designer to balance penalties on final dispersions with maximum intermediate maneuvering capabilities. At the same recovery site, a 40% improvement in CEP required a sacrifice of 14 km in planned maximum out-of-plane turns. Using the same methodology, the vehicle model, trajectory constraints, and uncertainties can be recast to apply covariance trajectory-shaping to any desired reentry scenario.

Prior to the reentry application, the simple Zermelo boat problem was used to compare the methodologies of the sensitivity and covariance trajectory-shaping techniques. For a wide range of uncertainties including state process noise, parametric process noise and biases, as well as state insertion errors, the sensitivity and covariance techniques were formulated to produce equivalent results. However, the covariance technique possesses several significant advantages over the sensitivity technique. First, the covariance matrix is symmetric, so it requires fewer additional design parameters and dynamic constraints for the augmented optimal control problem. Second, its statistical representation of dispersions along a reference trajectory provides a more useful and intuitive measure of the impacts of uncertainty on a system. Third, the covariance matrix is able to address common design goals with much simpler performance metrics than those required for the sensitivity matrix. Finally, the covariance Riccati dynamics can include measurement updates and navigation error, while the sensitivity dynamics cannot easily do so. For these reasons, the covariance technique was defended as the preferred method.

Using the Zermelo application to prepare for the reentry problem, the trajectory-shaping techniques demonstrated improvements in both fixed-target precision and

maximum capability characterization. In the fixed-target scenario, covariance- and sensitivity-shaping techniques produced trajectories that reached the final conditions with optimal dispersion variances, and in the maximum capability scenario, metrics with appropriate penalty weights extended the confident down-stream range of the boat. The Zermelo problem demonstrated that both trajectory-shaping techniques can desensitize trajectories to both open-loop and closed-loop dispersions. Subsequently, the feedback controller design process was successfully merged with the reference trajectory design process to produce even greater synergistic improvements. More importantly, closed-loop performance constraints were implemented to ensure the trajectory-controller pairs stabilized the system and respected reasonable control authority limitations. However, it was ultimately determined that while shaping a reference trajectory for a specific feedback controller produces the greatest improvements, such trajectories do not necessarily respond well when the feedback environment changes from the one for which they were designed. On the other hand, trajectories designed using open-loop dispersion considerations may not be optimal in closed-loop settings, but they still produced significant improvements over trajectories obtained without uncertainty considerations at all. For this reason, the reentry problem was addressed using open-loop uncertainty considerations only.

5.1 Future Work

Although this thesis successfully implemented the trajectory-shaping techniques to improve fixed-target precision and maximum capability characterizations for a complex hypersonic reentry vehicle application, the following additional work would strengthen the results:

1. Test Variable-Gain Controller Design: Although the combined reference-controller design technique was demonstrated using the Zermelo problem, it was only implemented with constant gains. The extension to variable gain selection is not complicated theoretically, but should still be tested.
2. Independently Design Reentry Feedback Controller: The trajectories shaped in this work only took open-loop dispersions into consideration, so an appropriate next step would be to begin working feedback into the design process. This could begin by designing a realistic feedback controller independently of the reference design process, which could be used to evaluate closed-loop performance along the current trajectories, and even reshape new trajectories specifically accounting for the controller's presence.
3. Monte Carlo Analysis: The trajectory-shaping techniques reduce expected errors about the *linearized* system, but the expectations do not actually simulate a true nonlinear flyout of the trajectory with random processes imposed upon the system. Seywald applies Monte Carlo analysis to his nominal and desensitized trajectories in [13] to demonstrate the correspondence between the expected and realized improvements. Similar closed-loop Monte Carlo analysis should be

conducted along the reentry trajectories to prove that the trajectory-shaping techniques produce real improvements in flyability.

4. Merge Reentry Reference Trajectory and Feedback Designs: As predicted in the Zermelo results, this process may produce even greater synergistic improvements. However, constant gains could be unreasonable for full reentry trajectories, so variable gain schedules would have to be worked into the design process.

Appendix A

Reentry Jacobian Calculations

The Jacobian for the reentry problem dynamics is comprised of the partial derivatives of each rate equation with respect to each state. While finite differencing or automatic differentiation software can be used to numerically generate the values of the Jacobian everywhere along a nominal trajectory, analytic evaluations provide faster computation speeds in MATLAB. In the covariance trajectory-shaping problem, the Jacobian will be evaluated thousands of times by the direct collocation method, so increased speed is highly desirable. Since the plant dynamics are entirely analytic, so is its Jacobian.

The full state vector is given by Eq. (4.65), which includes C_{C_X} , C_{C_N} , and C_ρ used to account for uncertainties in axial force coefficient, normal force coefficient, and atmospheric density, respectively. The full 11×11 Jacobian is expressed as

$$G = \begin{bmatrix} \frac{\partial \dot{r}}{\partial r} & \frac{\partial \dot{r}}{\partial \mu} & \frac{\partial \dot{r}}{\partial \lambda} & \frac{\partial \dot{r}}{\partial v} & \frac{\partial \dot{r}}{\partial \gamma} & \frac{\partial \dot{r}}{\partial \psi} & \frac{\partial \dot{r}}{\partial \alpha} & \frac{\partial \dot{r}}{\partial \sigma} & \frac{\partial \dot{r}}{\partial C_\rho} & \frac{\partial \dot{r}}{\partial C_{C_X}} & \frac{\partial \dot{r}}{\partial C_{C_N}} \\ \frac{\partial \dot{\mu}}{\partial r} & \frac{\partial \dot{\mu}}{\partial \mu} & \frac{\partial \dot{\mu}}{\partial \lambda} & \frac{\partial \dot{\mu}}{\partial v} & \frac{\partial \dot{\mu}}{\partial \gamma} & \frac{\partial \dot{\mu}}{\partial \psi} & \frac{\partial \dot{\mu}}{\partial \alpha} & \frac{\partial \dot{\mu}}{\partial \sigma} & \frac{\partial \dot{\mu}}{\partial C_\rho} & \frac{\partial \dot{\mu}}{\partial C_{C_X}} & \frac{\partial \dot{\mu}}{\partial C_{C_N}} \\ \frac{\partial \dot{\lambda}}{\partial r} & \frac{\partial \dot{\lambda}}{\partial \mu} & \frac{\partial \dot{\lambda}}{\partial \lambda} & \frac{\partial \dot{\lambda}}{\partial v} & \frac{\partial \dot{\lambda}}{\partial \gamma} & \frac{\partial \dot{\lambda}}{\partial \psi} & \frac{\partial \dot{\lambda}}{\partial \alpha} & \frac{\partial \dot{\lambda}}{\partial \sigma} & \frac{\partial \dot{\lambda}}{\partial C_\rho} & \frac{\partial \dot{\lambda}}{\partial C_{C_X}} & \frac{\partial \dot{\lambda}}{\partial C_{C_N}} \\ \frac{\partial \dot{v}}{\partial r} & \frac{\partial \dot{v}}{\partial \mu} & \frac{\partial \dot{v}}{\partial \lambda} & \frac{\partial \dot{v}}{\partial v} & \frac{\partial \dot{v}}{\partial \gamma} & \frac{\partial \dot{v}}{\partial \psi} & \frac{\partial \dot{v}}{\partial \alpha} & \frac{\partial \dot{v}}{\partial \sigma} & \frac{\partial \dot{v}}{\partial C_\rho} & \frac{\partial \dot{v}}{\partial C_{C_X}} & \frac{\partial \dot{v}}{\partial C_{C_N}} \\ \frac{\partial \dot{\gamma}}{\partial r} & \frac{\partial \dot{\gamma}}{\partial \mu} & \frac{\partial \dot{\gamma}}{\partial \lambda} & \frac{\partial \dot{\gamma}}{\partial v} & \frac{\partial \dot{\gamma}}{\partial \gamma} & \frac{\partial \dot{\gamma}}{\partial \psi} & \frac{\partial \dot{\gamma}}{\partial \alpha} & \frac{\partial \dot{\gamma}}{\partial \sigma} & \frac{\partial \dot{\gamma}}{\partial C_\rho} & \frac{\partial \dot{\gamma}}{\partial C_{C_X}} & \frac{\partial \dot{\gamma}}{\partial C_{C_N}} \\ \frac{\partial \dot{\psi}}{\partial r} & \frac{\partial \dot{\psi}}{\partial \mu} & \frac{\partial \dot{\psi}}{\partial \lambda} & \frac{\partial \dot{\psi}}{\partial v} & \frac{\partial \dot{\psi}}{\partial \gamma} & \frac{\partial \dot{\psi}}{\partial \psi} & \frac{\partial \dot{\psi}}{\partial \alpha} & \frac{\partial \dot{\psi}}{\partial \sigma} & \frac{\partial \dot{\psi}}{\partial C_\rho} & \frac{\partial \dot{\psi}}{\partial C_{C_X}} & \frac{\partial \dot{\psi}}{\partial C_{C_N}} \\ \frac{\partial \dot{\alpha}}{\partial r} & \frac{\partial \dot{\alpha}}{\partial \mu} & \frac{\partial \dot{\alpha}}{\partial \lambda} & \frac{\partial \dot{\alpha}}{\partial v} & \frac{\partial \dot{\alpha}}{\partial \gamma} & \frac{\partial \dot{\alpha}}{\partial \psi} & \frac{\partial \dot{\alpha}}{\partial \alpha} & \frac{\partial \dot{\alpha}}{\partial \sigma} & \frac{\partial \dot{\alpha}}{\partial C_\rho} & \frac{\partial \dot{\alpha}}{\partial C_{C_X}} & \frac{\partial \dot{\alpha}}{\partial C_{C_N}} \\ \frac{\partial \dot{\sigma}}{\partial r} & \frac{\partial \dot{\sigma}}{\partial \mu} & \frac{\partial \dot{\sigma}}{\partial \lambda} & \frac{\partial \dot{\sigma}}{\partial v} & \frac{\partial \dot{\sigma}}{\partial \gamma} & \frac{\partial \dot{\sigma}}{\partial \psi} & \frac{\partial \dot{\sigma}}{\partial \alpha} & \frac{\partial \dot{\sigma}}{\partial \sigma} & \frac{\partial \dot{\sigma}}{\partial C_\rho} & \frac{\partial \dot{\sigma}}{\partial C_{C_X}} & \frac{\partial \dot{\sigma}}{\partial C_{C_N}} \\ \frac{\partial \dot{C}_\rho}{\partial r} & \frac{\partial \dot{C}_\rho}{\partial \mu} & \frac{\partial \dot{C}_\rho}{\partial \lambda} & \frac{\partial \dot{C}_\rho}{\partial v} & \frac{\partial \dot{C}_\rho}{\partial \gamma} & \frac{\partial \dot{C}_\rho}{\partial \psi} & \frac{\partial \dot{C}_\rho}{\partial \alpha} & \frac{\partial \dot{C}_\rho}{\partial \sigma} & \frac{\partial \dot{C}_\rho}{\partial C_\rho} & \frac{\partial \dot{C}_\rho}{\partial C_{C_X}} & \frac{\partial \dot{C}_\rho}{\partial C_{C_N}} \\ \frac{\partial \dot{C}_{C_X}}{\partial r} & \frac{\partial \dot{C}_{C_X}}{\partial \mu} & \frac{\partial \dot{C}_{C_X}}{\partial \lambda} & \frac{\partial \dot{C}_{C_X}}{\partial v} & \frac{\partial \dot{C}_{C_X}}{\partial \gamma} & \frac{\partial \dot{C}_{C_X}}{\partial \psi} & \frac{\partial \dot{C}_{C_X}}{\partial \alpha} & \frac{\partial \dot{C}_{C_X}}{\partial \sigma} & \frac{\partial \dot{C}_{C_X}}{\partial C_\rho} & \frac{\partial \dot{C}_{C_X}}{\partial C_{C_X}} & \frac{\partial \dot{C}_{C_X}}{\partial C_{C_N}} \\ \frac{\partial \dot{C}_{C_N}}{\partial r} & \frac{\partial \dot{C}_{C_N}}{\partial \mu} & \frac{\partial \dot{C}_{C_N}}{\partial \lambda} & \frac{\partial \dot{C}_{C_N}}{\partial v} & \frac{\partial \dot{C}_{C_N}}{\partial \gamma} & \frac{\partial \dot{C}_{C_N}}{\partial \psi} & \frac{\partial \dot{C}_{C_N}}{\partial \alpha} & \frac{\partial \dot{C}_{C_N}}{\partial \sigma} & \frac{\partial \dot{C}_{C_N}}{\partial C_\rho} & \frac{\partial \dot{C}_{C_N}}{\partial C_{C_X}} & \frac{\partial \dot{C}_{C_N}}{\partial C_{C_N}} \end{bmatrix}$$

First, C_ρ , C_{C_X} , and C_{C_N} have trivial dynamics, so the last three rows are full of zeros. Next, the 8×8 upper left submatrix represents the Jacobian of the original system. Since the Jacobian is evaluated along the nominal trajectory, and since C_ρ , C_{C_X} , and C_{C_N} have nominal values of zero, the additional parameters do not alter the Jacobian of the original system.

Treating this submatrix first, the partials of \dot{r} are

$$\begin{aligned}\frac{\partial \dot{r}}{\partial r} &= 0, \\ \frac{\partial \dot{r}}{\partial \mu} &= 0, \\ \frac{\partial \dot{r}}{\partial \lambda} &= 0, \\ \frac{\partial \dot{r}}{\partial v} &= \sin(\gamma), \\ \frac{\partial \dot{r}}{\partial \gamma} &= v \cos(\gamma), \\ \frac{\partial \dot{r}}{\partial \psi} &= 0, \\ \frac{\partial \dot{r}}{\partial \alpha} &= 0, \\ \frac{\partial \dot{r}}{\partial \sigma} &= 0.\end{aligned}$$

In the second row, the partials of $\dot{\mu}$ are

$$\begin{aligned}\frac{\partial \dot{\mu}}{\partial r} &= -\frac{v \cos(\gamma) \cos(\psi)}{r^2 \cos(\lambda)}, \\ \frac{\partial \dot{\mu}}{\partial \mu} &= 0, \\ \frac{\partial \dot{\mu}}{\partial \lambda} &= \frac{v \cos(\gamma) \cos(\psi) \sin(\lambda)}{r \cos^2(\lambda)}, \\ \frac{\partial \dot{\mu}}{\partial v} &= \frac{\cos(\gamma) \cos(\psi)}{r \cos(\lambda)}, \\ \frac{\partial \dot{\mu}}{\partial \gamma} &= -\frac{v \sin(\gamma) \cos(\psi)}{r \cos(\lambda)}, \\ \frac{\partial \dot{\mu}}{\partial \psi} &= -\frac{v \cos(\gamma) \sin(\psi)}{r \cos(\lambda)}, \\ \frac{\partial \dot{\mu}}{\partial \alpha} &= 0, \\ \frac{\partial \dot{\mu}}{\partial \sigma} &= 0.\end{aligned}$$

In the third row, the partials of $\dot{\lambda}$ are

$$\begin{aligned}\frac{\partial \dot{\lambda}}{\partial r} &= -\frac{v \cos(\gamma) \sin(\psi)}{r^2}, \\ \frac{\partial \dot{\lambda}}{\partial \mu} &= 0,\end{aligned}$$

$$\begin{aligned}
\frac{\partial \dot{\mu}}{\partial \lambda} &= 0, \\
\frac{\partial \dot{\mu}}{\partial v} &= \frac{\cos(\gamma) \sin(\psi)}{r}, \\
\frac{\partial \dot{\mu}}{\partial \gamma} &= -\frac{v \sin(\gamma) \sin(\psi)}{r}, \\
\frac{\partial \dot{\mu}}{\partial \psi} &= \frac{v \cos(\gamma) \cos(\psi)}{r}, \\
\frac{\partial \dot{r}}{\partial \alpha} &= 0, \\
\frac{\partial \dot{r}}{\partial \sigma} &= 0.
\end{aligned}$$

The next three rows contain the partials of \dot{v} , $\dot{\gamma}$, and $\dot{\psi}$, respectively, which become significantly more complicated because the lift, drag, centrifugal, and Coriolis acceleration terms require complex chain rule applications. The chain rule identities

$$\begin{aligned}
\frac{\partial D}{\partial r} &= \frac{\partial D}{\partial q} \frac{\partial q}{\partial \rho} \frac{\partial \rho}{\partial h_p} \frac{\partial h_p}{\partial h} \frac{\partial h}{\partial r}, \\
\frac{\partial D}{\partial \lambda} &= \frac{\partial D}{\partial q} \frac{\partial q}{\partial \rho} \frac{\partial \rho}{\partial h_p} \frac{\partial h_p}{\partial h} \frac{\partial h}{\partial \lambda}, \\
\frac{\partial L}{\partial r} &= \frac{\partial L}{\partial q} \frac{\partial q}{\partial \rho} \frac{\partial \rho}{\partial h_p} \frac{\partial h_p}{\partial h} \frac{\partial h}{\partial r}, \\
\frac{\partial L}{\partial \lambda} &= \frac{\partial L}{\partial q} \frac{\partial q}{\partial \rho} \frac{\partial \rho}{\partial h_p} \frac{\partial h_p}{\partial h} \frac{\partial h}{\partial \lambda},
\end{aligned}$$

are used to simplify the final expressions. The partial derivatives of each of the centrifugal and Coriolis variables are expressed as

$$\begin{aligned}
\frac{\partial F_1}{\partial \lambda} &= -2 \cos(\lambda) \sin(\lambda) \sin(\gamma) - \cos(\gamma) \sin(\psi) (\cos^2(\lambda) - \sin^2(\lambda)), \\
\frac{\partial F_1}{\partial \gamma} &= \cos^2(\lambda) \cos(\gamma) + \cos(\lambda) \sin(\lambda) \sin(\gamma) \sin \psi, \\
\frac{\partial F_1}{\partial \psi} &= -\cos(\lambda) \sin(\lambda) \cos(\gamma) \cos(\psi), \\
\frac{\partial F_2}{\partial \lambda} &= -2 \cos(\lambda) \sin(\lambda) \cos(\gamma) + \sin(\gamma) \sin(\psi) (\cos^2(\lambda) - \sin^2(\lambda)), \\
\frac{\partial F_2}{\partial \gamma} &= -\cos^2(\lambda) \sin(\gamma) + \cos(\lambda) \sin(\lambda) \cos(\gamma) \sin \psi, \\
\frac{\partial F_2}{\partial \psi} &= \cos(\lambda) \sin(\lambda) \sin(\gamma) \cos(\psi), \\
\frac{\partial F_3}{\partial \lambda} &= \cos(\psi) (\cos^2(\lambda) - \sin^2(\lambda)),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial F_3}{\partial \psi} &= -\cos(\lambda) \sin(\lambda) \sin(\psi) , \\
\frac{\partial C_2}{\partial \lambda} &= -\sin(\lambda) \cos(\psi) , \\
\frac{\partial C_2}{\partial \psi} &= -\sin(\lambda) \cos(\psi) , \\
\frac{\partial C_3}{\partial \lambda} &= \cos(\lambda) \cos(\gamma) + \sin(\lambda) \sin(\gamma) \sin(\psi) , \\
\frac{\partial C_3}{\partial \gamma} &= -\sin(\lambda) \sin(\gamma) - \cos(\lambda) \cos(\gamma) \sin(\psi) , \\
\frac{\partial C_3}{\partial \psi} &= -\cos(\lambda) \sin(\gamma) \sin(\psi) .
\end{aligned}$$

Other necessary partial derivative relationships include

$$\begin{aligned}
\frac{\partial \dot{v}}{\partial D} &= -\frac{1}{m} , \\
\frac{\partial \dot{\gamma}}{\partial L} &= \frac{\cos(\sigma)}{mv} , \\
\frac{\partial \dot{\psi}}{\partial L} &= \frac{\sin(\sigma)}{mv \cos(\gamma)} , \\
\frac{\partial D}{\partial q} &= SC_D , \\
\frac{\partial D}{\partial C_D} &= qS , \\
\frac{\partial L}{\partial q} &= SC_L , \\
\frac{\partial L}{\partial C_L} &= qS , \\
\frac{\partial q}{\partial \rho} &= \frac{1}{2}v^2 , \\
\frac{\partial q}{\partial v} &= \rho v , \\
\frac{\partial h_p}{\partial h} &= \frac{R_0^2}{R_0^2 + h^2} , \\
\frac{\partial h}{\partial r} &= 1 - \frac{R_e^2 f^2}{4r^2} (1 - \cos(4\lambda)) , \\
\frac{\partial h}{\partial \lambda} &= R_e f \sin(2\lambda) + R_e f^2 \left(\frac{R_e}{r} - \frac{1}{4} \right) \sin(4\lambda) , \\
\frac{\partial C_D}{\partial C_X} &= \cos(\alpha) , \\
\frac{\partial C_D}{\partial C_N} &= \sin(\alpha) ,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial C_D}{\partial \alpha} &= C_N \cos(\alpha) - C_X \sin(\alpha) , \\
\frac{\partial C_L}{\partial C_X} &= -\sin(\alpha) , \\
\frac{\partial C_L}{\partial C_N} &= \cos(\alpha) , \\
\frac{\partial C_L}{\partial \alpha} &= -C_N \sin(\alpha) - C_X \cos(\alpha) , \\
\frac{\partial C_X}{\partial \alpha} &= 2C_{Xk}\alpha , \\
\frac{\partial C_X}{\partial M} &= -C_{Xa}C_{Xb}e^{-C_{Xb}(M-C_{Xc})} , \\
\frac{\partial C_N}{\partial \alpha} &= C_{Na} , \\
\frac{\partial M}{\partial v} &= \frac{1}{s_s} , \\
\frac{\partial \rho}{\partial h_p} &= (a_1 + 2a_2h_p + \dots + na_nh_p^{n-1}) e^{a_0+a_1h_p+a_2h_p^2+\dots+a_nh_p^n} ,
\end{aligned}$$

where n is the order of the polynomial used to fit COESA atmospheric density to geopotential altitude.

Utilizing the chain rule in conjunction with these identities, the partial derivatives of \dot{v} are

$$\begin{aligned}
\frac{\partial \dot{v}}{\partial r} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial r} + \Omega_E^2 F_1 + \frac{2G_m}{r^2} \sin(\gamma) , \\
\frac{\partial \dot{v}}{\partial \mu} &= 0 , \\
\frac{\partial \dot{v}}{\partial \lambda} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial r} + \Omega_E^2 r \frac{\partial F_1}{\partial \lambda} , \\
\frac{\partial \dot{v}}{\partial v} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial q} \frac{\partial q}{\partial v} + \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial C_D} \frac{\partial C_D}{\partial C_X} \frac{\partial C_X}{\partial M} \frac{\partial M}{\partial v} , \\
\frac{\partial \dot{v}}{\partial \gamma} &= -\frac{G_m}{r^2} \cos(\gamma) + \Omega_E^2 r \frac{\partial F_1}{\partial \gamma} , \\
\frac{\partial \dot{v}}{\partial \psi} &= \Omega_E^2 r \frac{\partial F_1}{\partial \psi} , \\
\frac{\partial \dot{v}}{\partial \alpha} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial C_D} \left(\frac{\partial C_D}{\partial \alpha} + \frac{\partial C_D}{\partial C_X} \frac{\partial C_X}{\partial \alpha} + \frac{\partial C_D}{\partial C_N} \frac{\partial C_N}{\partial \alpha} \right) , \\
\frac{\partial \dot{v}}{\partial \sigma} &= 0 .
\end{aligned}$$

The partial derivatives of $\dot{\gamma}$ are

$$\begin{aligned}
\frac{\partial \dot{\gamma}}{\partial r} &= \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial r} + \left(-\frac{v}{r^2} + \frac{2G_m}{r^2 v} \right) \cos(\gamma) + \frac{\Omega_E^2 F_2}{v}, \\
\frac{\partial \dot{\gamma}}{\partial \mu} &= 0, \\
\frac{\partial \dot{\gamma}}{\partial \lambda} &= \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial \lambda} + 2\Omega_E \frac{\partial C_2}{\partial \lambda} + \Omega_E^2 \frac{r}{v} \frac{\partial F_2}{\partial \lambda}, \\
\frac{\partial \dot{\gamma}}{\partial v} &= -\frac{L \cos(\sigma)}{mv^2} + \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial q} \frac{\partial q}{\partial v} + \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial C_L} \frac{\partial C_L}{\partial C_X} \frac{\partial C_X}{\partial M} \frac{\partial M}{\partial v} \\
&\quad + \left(\frac{1}{r} + \frac{G_m}{r^2 v^2} \right) \cos(\gamma) - \Omega_E^2 F_2 \frac{r}{v^2}, \\
\frac{\partial \dot{\gamma}}{\partial \gamma} &= -\left(\frac{v}{r} - \frac{G_m}{r^2 v} \right) \sin(\gamma) + \Omega_E^2 \frac{r}{v} \frac{\partial F_2}{\partial \gamma}, \\
\frac{\partial \dot{\gamma}}{\partial \psi} &= 2\Omega_E \frac{\partial C_2}{\partial \psi} + \Omega_E^2 \frac{r}{v} \frac{\partial F_2}{\partial \psi}, \\
\frac{\partial \dot{\gamma}}{\partial \alpha} &= \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial C_L} \left(\frac{\partial C_L}{\partial \alpha} + \frac{\partial C_{DL}}{\partial C_X} \frac{\partial C_X}{\partial \alpha} + \frac{\partial C_L}{\partial C_N} \frac{\partial C_N}{\partial \alpha} \right), \\
\frac{\partial \dot{\gamma}}{\partial \sigma} &= -\frac{L \sin(\sigma)}{mv}.
\end{aligned}$$

The partial derivatives of $\dot{\psi}$ are

$$\begin{aligned}
\frac{\partial \dot{\psi}}{\partial r} &= \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial r} + \frac{v}{r^2} \cos(\gamma) \cos(\psi) \tan(\lambda) - \frac{\Omega_E^2 F_3}{v \cos(\gamma)}, \\
\frac{\partial \dot{\psi}}{\partial \mu} &= 0, \\
\frac{\partial \dot{\psi}}{\partial \lambda} &= \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial \lambda} - \frac{v \cos(\gamma) \cos(\psi)}{r \cos^2(\lambda)} - \frac{2\Omega_E}{\cos(\gamma)} \frac{\partial C_3}{\partial \lambda} - \frac{\Omega_E^2 r}{v \cos(\gamma)} \frac{\partial F_3}{\partial \lambda}, \\
\frac{\partial \dot{\psi}}{\partial v} &= -\frac{L \sin(\sigma)}{mv^2 \cos(\gamma)} + \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial q} \frac{\partial q}{\partial v} + \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial C_L} \frac{\partial C_L}{\partial C_X} \frac{\partial C_X}{\partial M} \frac{\partial M}{\partial v} \\
&\quad - \frac{\cos(\gamma) \cos(\psi) \tan(\lambda)}{r} + \frac{\Omega_E^2 r F_3}{v^2 \cos(\gamma)}, \\
\frac{\partial \dot{\psi}}{\partial \gamma} &= \frac{L \sin(\sigma) \sin(\gamma)}{mv \cos^2(\gamma)} + \frac{v}{r} \sin(\gamma) \cos(\psi) \tan(\lambda) - \frac{2\Omega_E C_3 \sin(\gamma)}{\cos^2(\gamma)} \\
&\quad - \frac{2\Omega_E}{\cos(\gamma)} \frac{\partial C_3}{\partial \gamma} - \frac{\Omega_E^2 r F_3 \sin(\gamma)}{v \cos^2(\gamma)}, \\
\frac{\partial \dot{\psi}}{\partial \psi} &= \frac{v}{r} \cos(\gamma) \sin(\psi) \tan(\lambda) - \frac{2\Omega_E}{\cos(\gamma)} \frac{\partial C_3}{\partial \psi} - \frac{\Omega_E^2 r}{v \cos(\gamma)} \frac{\partial F_3}{\partial \psi}, \\
\frac{\partial \dot{\psi}}{\partial \alpha} &= \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial C_L} \left(\frac{\partial C_L}{\partial \alpha} + \frac{\partial C_{DL}}{\partial C_X} \frac{\partial C_X}{\partial \alpha} + \frac{\partial C_L}{\partial C_N} \frac{\partial C_N}{\partial \alpha} \right),
\end{aligned}$$

$$\frac{\partial \dot{\psi}}{\partial \sigma} = \frac{L \cos(\sigma)}{mv \cos(\gamma)}.$$

Now, the partial derivatives of the 8 original states are taken with respect to the 3 additional uncertainty model parameters. Utilizing Eqs. (4.57), (4.61), and (4.62), the partial derivatives of each original state rate with respect to C_ρ are

$$\begin{aligned} \frac{\partial \dot{r}}{\partial C_\rho} &= 0, \\ \frac{\partial \dot{\mu}}{\partial C_\rho} &= 0, \\ \frac{\partial \dot{\lambda}}{\partial C_\rho} &= 0, \\ \frac{\partial \dot{v}}{\partial C_\rho} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial q} \frac{\partial q}{\partial \rho} \rho \sigma_\rho(\mathbf{x}), \\ \frac{\partial \dot{\gamma}}{\partial C_\rho} &= \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial q} \frac{\partial q}{\partial \rho} \rho \sigma_\rho(\mathbf{x}), \\ \frac{\partial \dot{\psi}}{\partial C_\rho} &= \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial q} \frac{\partial q}{\partial \rho} \rho \sigma_\rho(\mathbf{x}), \\ \frac{\partial \dot{\alpha}}{\partial C_\rho} &= 0, \\ \frac{\partial \dot{\sigma}}{\partial C_\rho} &= 0. \end{aligned}$$

The partial derivatives of each original state rate with respect to C_{C_x} are

$$\begin{aligned} \frac{\partial \dot{r}}{\partial C_{C_x}} &= 0, \\ \frac{\partial \dot{\mu}}{\partial C_{C_x}} &= 0, \\ \frac{\partial \dot{\lambda}}{\partial C_{C_x}} &= 0, \\ \frac{\partial \dot{v}}{\partial C_{C_x}} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial C_D} \frac{\partial C_D}{\partial C_x} C_x \sigma_{C_x}(\mathbf{x}), \\ \frac{\partial \dot{\gamma}}{\partial C_{C_x}} &= \frac{\partial \dot{\gamma}}{\partial L} \frac{\partial L}{\partial C_L} \frac{\partial C_L}{\partial C_x} C_x \sigma_{C_x}(\mathbf{x}), \\ \frac{\partial \dot{\psi}}{\partial C_{C_x}} &= \frac{\partial \dot{\psi}}{\partial L} \frac{\partial L}{\partial C_L} \frac{\partial C_L}{\partial C_x} C_x \sigma_{C_x}(\mathbf{x}), \\ \frac{\partial \dot{\alpha}}{\partial C_{C_x}} &= 0, \\ \frac{\partial \dot{\sigma}}{\partial C_{C_x}} &= 0. \end{aligned}$$

The partial derivatives of each original state rate with respect to C_{C_N} are

$$\begin{aligned}
\frac{\partial \dot{r}}{\partial C_{C_N}} &= 0, \\
\frac{\partial \dot{\mu}}{\partial C_{C_N}} &= 0, \\
\frac{\partial \dot{\lambda}}{\partial C_{C_N}} &= 0, \\
\frac{\partial \dot{v}}{\partial C_{C_N}} &= \frac{\partial \dot{v}}{\partial D} \frac{\partial D}{\partial C_D} \frac{\partial C_D}{\partial C_N} C_N \sigma_{C_N}(\mathbf{x}), \\
\frac{\partial \dot{\gamma}}{\partial C_{C_N}} &= \frac{\partial \dot{\gamma}}{\partial D} \frac{\partial D}{\partial C_D} \frac{\partial C_D}{\partial C_N} C_N \sigma_{C_N}(\mathbf{x}), \\
\frac{\partial \dot{\psi}}{\partial C_{C_N}} &= \frac{\partial \dot{\psi}}{\partial D} \frac{\partial D}{\partial C_D} \frac{\partial C_D}{\partial C_N} C_N \sigma_{C_N}(\mathbf{x}), \\
\frac{\partial \dot{\alpha}}{\partial C_{C_N}} &= 0, \\
\frac{\partial \dot{\sigma}}{\partial C_{C_N}} &= 0.
\end{aligned}$$

Now, each element of the full 11×11 Jacobian is expressed analytically, and the Jacobian can be coded and evaluated along nominal trajectories during covariance propagations.

Since the analytic derivation of the reentry problem Jacobian is relatively complex and prone to mistakes, the coded expressions are checked for accuracy with finite differencing. Forward-differencing and central-differencing are popular methods which provide first- and second-order derivative estimates, respectively. However, their accuracy can be highly dependent on step size. Large steps allow significant truncation errors, but small steps result in subtraction errors. Both problems can be avoided using the complex step derivative calculation method [7]. For any function $f(x)$, the complex step method calculates the derivative with respect to x by

$$\frac{df}{dx} = \frac{\text{imag}(f(x + i\Delta))}{\Delta}$$

where i is the unit imaginary number, Δ is the selected step size, and the `imag` operation extracts the imaginary portion of the complex number on which it operates. This finite difference method provides much more consistent results for any step size, and is easy to implement in MATLAB. For the reentry problem Jacobian calculation, the analytic and complex step derivatives along any given solution agree to within an order of magnitude of 1×10^{-13} .

Bibliography

- [1] Matthew J. Abrahamson. Boost Through Reentry Trajectory Planning for Maneuvering Reentry Vehicles. Master's thesis, MIT Department of Aeronautics and Astronautics, 2008.
- [2] John T. Betts. Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21:193–207, 1998.
- [3] Robert G. Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, 1997.
- [4] Bernard Friedland. *Control System Design: An Introduction to State-Space Methods*. Dover Publications, Inc., 1986.
- [5] Sean George. 6DOF Equations of Motion - Wind Axes. Charles Stark Draper Laboratory Internal Memorandum.
- [6] William Graham. First Minotaur IV Launches with Hypersonic Test Vehicle. *www.nasaspaceflight.com*, Apr 22, 2010.
- [7] I. M. Kroo J. R. R. A. Martins and J. J. Alonso. An Automated Method for Sensitivity Analysis using Complex Variables. *AIAA 38th Aerospace Sciences Meeting and Exhibit*, A00-16548, 2000.
- [8] C.G. Justus and F.W. Leslie. The NASA MSFC Earth Global Reference Atmospheric Model - 2007 Revision. 2008.
- [9] I. Michael Ross Kevin P. Bollino and D. B. Doman. Six-Degree-Of-Freedom Trajectory Optimization for Reusable Launch Vehicle Footprint Determination. *AIAA/AAS Astrodynamics Specialist Conference*, 2007.
- [10] T. W. Knacke. The Apollo Parachute Landing System. *AIAA Second Aerodynamic Decelerator Systems Conference*, Northrop Ventura TP-131, 1968.
- [11] Tariq Malik. Going Hypersonic: Flying FALCON for Defense. *www.space.com*, 2003.
- [12] I. Michael Ross. A Beginner's Guide to DIDO. 1998.
- [13] Hans Seywald. Desensitized Optimal Trajectories with Control Constraints. *AIAA/AAS Spaceflight Mechanics Meeting*, AAS Paper 03-147:737–743, 2003.

- [14] Hans Seywald. Desensitized Optimal Orbit Insertion. *AIAA/AAS Spaceflight Mechanics Meeting*, AAS Paper 04-215:1765–1780, 2004.
- [15] Hans Seywald and Renjith R. Kumar. Desensitized Optimal Trajectories. *AIAA/AAS Spaceflight Mechanics Meeting*, AAS Paper 96-107:103–115, 1996.
- [16] 1976 U.S. Standard Atmosphere. U.S. Government Printing Office, Washington, D.C. 1976.
- [17] Aditya Undurti. Optimal Trajectories for Maneuvering Reentry Vehicles. Master's thesis, MIT Department of Aeronautics and Astronautics, 2007.
- [18] Scott J. Zimmer. *Reducing Spacecraft State Uncertainty Through Indirect Trajectory Optimization*. PhD thesis, The University of Texas at Austin, 2005.