Theses and Dissertations   Student Graduate Works

9-2007

Common Aero Vehicle Autonomous Reentry Trajectory Optimization Satisfying Waypoint and No-Fly Zone Constraints

Timothy R. Jorris

Common Aero Vehicle
Autonomous Reentry Trajectory Optimization
Satisfying Waypoint and No-Fly
Zone Constraints

DISSERTATION

Timothy R. Jorris, Major, USAF

AFIT/DS/ENY/07-04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

COMMON AERO VEHICLE
AUTONOMOUS REENTRY TRAJECTORY OPTIMIZATION
SATISFYING WAYPOINT AND NO-FLY
ZONE CONSTRAINTS

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Timothy R. Jorris, BS, MS
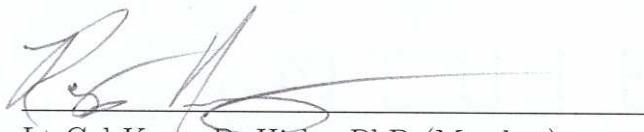
Major, USAF

September 2007

Common Aero Vehicle
Autonomous Reentry Trajectory Optimization
Satisfying Waypoint and No-Fly
Zone Constraints

Timothy R. Jorris, BS, MS

Major, USAF

Approved:

_____   _____
Dr. Richard G. Cobb (Chairman)   15 Aug 07
                                                                          Date

_____   _____
Lt Col Kerry D. Hicks, PhD (Member)   15 Aug 07
                                                                          Date

_____   _____
Dr. William P. Baker (Member)   15 Aug 07
                                                                          Date

_____   _____
Dr. Alan W. Johnson (Dean's Representative)   15 Aug 07
                                                                          Date

Accepted:

_____   _____
Dr. Marlin U. Thomas   31 Aug 07
Dean, Graduate School of   Date
Engineering and Management

AFIT/DS/ENY/07-04

Abstract

To support the Air Force's Global Reach concept, a Common Aero Vehicle is being designed to support the Global Strike mission. "Waypoints" are specified for reconnaissance or multiple payload deployments and "no-fly zones" are specified for geopolitical restrictions or threat avoidance. Due to time critical targets and multiple scenario analysis, an autonomous solution is preferred over a time-intensive, manually itera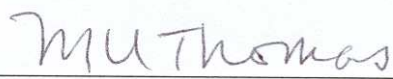tive one. Thus, a real-time or near real-time autonomous trajectory optimization technique is presented to minimize the flight time, satisfy terminal and intermediate constraints, and remain within the specified vehicle heating and control limitations. This research uses the Hypersonic Cruise Vehicle (HCV) as a simplified two-dimensional platform to compare multiple solution techniques. The solution techniques include a unique geometric approach developed herein, a derived analytical dynamic optimization technique, and a rapidly emerging collocation numerical approach. This up-and-coming numerical technique is a direct solution method involving discretization then dualization, with pseudospectral methods and nonlinear programming used to converge to the optimal solution. This numerical approach is applied to the Common Aero Vehicle (CAV) as the test platform for the full three-dimensional reentry trajectory optimization problem. The culmination of this research is the verification of the optimality of this proposed numerical technique, as shown for both the two-dimensional and three-dimensional models. Additionally, user implementation strategies are presented to improve accuracy and enhance solution convergence. Thus, the contributions of this research are the geometric approach, the user implementation strategies, and the determination and verification of a numerical solution technique for the optimal reentry trajectory problem that minimizes time to target while satisfying vehicle dynamics and control limitation, and heating, waypoint, and no-fly zone constraints.

Acknowledgments

I want to thank Dr. Richard Cobb for pitting my wingman David Irvin and me against each other; without such fierce competition I don't think either of us would have graduated. I'd also like to thank Vince Chioma for helping me get jump started in the program, and Vanessa Bond for being there to provide motivation whenever necessary. The bottom line, however, is that this dissertation would really not be possible without the encouragement and support of my loving wife.

Timothy R. Jorris

Table of Contents

List of Figures

List of Tables

xiii

List of Symbols

List of Abbreviations

Common Aero Vehicle
Autonomous Reentry Trajectory Optimization
Satisfying Waypoint and No-Fly
Zone Constraints

I. Introduction

1.1 Motivation

Global Strike (GS) and Global Persistent Attack (GPA) are two of the seven United States Air Force Concepts of Operations [104:pg.10]. Various hypersonic and reentry vehicle technologies are being pursued to enable such a prompt global reach capability [118]. The government competition for the Force Application Launch from the Continental United States (FALCON) [19:pg.4] program and the National Aeronautics and Space Administration (NASA) Next Generation Launch Technology (NGLT) Program Office [42] validate both the interest and need for future research. The Common Aero Vehicle (CAV) is an Unmanned Aerial Vehicle (UAV) supporting these technologies with the role of "Striking from Space" [76]. One goal is a global strike mission where targets within a 9,000 nautical mile range can be reached in less than two hours with 12,000 lb payload capabilities [121]. To analyze integration and mission effectiveness, the Control Sciences Division of the Air Force Research Laboratory/Air Vehicles Directorate (AFRL/VA), Aerospace Vehicles Technology Assessment and Simulation Branch at Wright-Patterson AFB is developing the Space Access Vehicles Mission and Operations Simulation (SAVMOS) [47; 117]. SAVMOS is a computer simulation environment designed for modeling a Military Spaceplane (MSP), Common Aero Vehicle (CAV) [57:pg.7; 32], and its operations system. The core trajectory generation for reentry is the Integrated Development and Operations System (IDOS) [57; 47:pg.3]. This allows for simulation runs from reentry to the target, with the trajectory satisfying vehicle dynamics and heat/structural load constraints. The cur-

rent CAV implementation within SAVMOS does not allow for rapid mission/target modifications, nor is this a current operational capability for Global Strike.

1.2 Problem Description

The desire to compute a survivable reentry trajectory has been around since the dawn of manned space flight. The Space Shuttle introduced a more difficult challenge by including a larger lift capacity reentry vehicle with mission needs to precisely navigate to a desired landing site, thus requiring appropriate guidance [39]. Since then, autonomous reentry vehicles such as the X-33 and X-37 required both autonomy and the ability to accommodate off-nominal reentry conditions produced from modeling errors or vehicle faults. A summary of the on-going reentry trajectory methods is featured in [22]. One article of interest [36] compared five of the proposed trajectory generation techniques and tested them with the high-fidelity X-33 Marshall Aerospace Vehicles Representation in C (MAVERIC) simulation [38]. These results were re-presented two years later [37]. Clearly the civilian objective of reentry through non-hostile environments, with the single mission of vehicle recovery, has been addressed.

For military applications to vehicles such as CAV, additional requirements are introduced. First of all, geopolitical limitations may be imposed, thus eliminating feasible trajectories that would overfly these restricted areas. Also, reconnaissance or multiple payload deployments may dictate flying through specified waypoints. Waypoints may also be used to adjust the reentry trajectory to coordinate with other military objectives or air traffic avoidance. The target is assumed to be time-critical; therefore, achieving these objectives in minimum time is required.

Therefore, the desired capability and the overall goal of this research is:

Given a time-sensitive target, impose multiple intermediate waypoints, add threat avoidance or no-fly zones, and then autonomously generate a minimum time flyable and survivable trajectory for a hypersonic reentry vehicle such as the Common Aero Vehicle.

2

The final research objective is to prove the optimality of the solution to the above problem. The methodology will be to pose it as a minimization problem having an objective function with equality and inequality constraints. The objective, or cost, function includes time to target; however, it could also be any function of the final states, e.g. terminal velocity or energy. The waypoints and no-fly zones are considered rigid constraints, meaning waypoint exact passage must be achieved, and no-fly zones shall not be violated. A survivable trajectory imposes a maximum allowable spike heating constraint. Lastly, the optimality of a numerically generated solution must be verified. Some methods enforce optimality in the solution process, others will need to be verified after the solution is generated. This is addressed in Chapters III and IV.

1.2.1   *Terminology.*   The overall mission objective is to fly from an *initial* point to a *final/terminal* point or *target*, in minimum time. The generic start and finish point of a trajectory are called *endpoints*. *Waypoints* are specified intermediate coordinates to fly over to satisfy payload delivery or reconnaissance mission requirements. The vehicle must fly directly over each waypoint, also called waypoint *passage*; however, time, altitude, bank angle, flight path angle, velocity, and heading are not constrained. Some control law solutions are called *bang-bang* or *bang-level-bang*, meaning the control takes on the value of minimum, zero, or maximum. For a bang-bang controller for bank angle, a *turnpoint* is defined as a discrete point where there is a change in the current constant control; therefore, the vehicle rolls into a turn or completes a turn at a turnpoint. There is no predetermined limit on the number of discrete changes in control, i.e. turnpoints. If waypoint passage occurs at some point within a constant control turn, the waypoint and turnpoints are not coincident. A *no-fly zone* is a region with a boundary that the vehicle may contact but must not violate. The entire trajectory can be broken up into sequential *legs*, *segments*, or *phases*; these terms may be used interchangeably. The breaks between phases may occur at waypoints, turnpoints, or other ending criteria such as no-fly zone contact.

The problem starts at the initial time t_0. Since there can be multiple waypoints in a mission, each is numbered i, and the time of passage is denoted t_i. Similarly, each no-fly zone is numbered j; however, the time of interest for each no-fly zone is the time of no-fly zone boundary contact t_j. The target is reached at the final or *terminal* time t_f and is called target *intercept*.

1.2.2 Hypersonic Cruise Vehicle (HCV) Overview. The Hypersonic Cruise Vehicle (HCV) is a proposed hypersonic vehicle to meet specified global reach objectives. The HCV objective is to deliver 12,000 lb of payload at a range of 9,000 nautical miles (nmi) in 2 hours [20; 105], at a notional altitude of 100,000 feet. The HCV has thrust thus validating a simplifying constant altitude assumption as used herein. The HCV provides a platform with a large turn radius, making it ideal to validate optimal trajectories in the two-dimensional horizontal plane. The HCV is of additional interest because it may have the mission of deploying a CAV [121]. More details of the HCV specifications and limitations are provided in Section 3.3.

1.2.3 Common Aero Vehicle (CAV) Overview. The following abstract provides a succinct introduction to the Common Aero Vehicle:

> The Common Aero Vehicle (CAV) is a concept which describes a space reentry aeroshell launched into space on a suitable vehicle, which then survives atmospheric reentry, reduces its speed to low Mach numbers or even sub-Mach, and dispenses a cargo, payload or weapon in the Earth's atmosphere. The conceptual CAV might be propelled into space by any of a number of present or future launch platforms and dispense a wide variety of cargoes, payloads or weapons. Development of the CAV capability will satisfy future requirements enunciated in numerous national visions, future studies, and military plans and will ultimately be necessary to fully realize the opportunities inherent in operating from, through, and in space and give true meaning to the phrases global reach, global power projection, and global engagement. [75]

1.2.4 Launch Scenario. The launch profile of study is a sub-orbital reentry. Launch may be achieved via an expendable launch vehicle, expendable air launch

rocket, the notional reusable Space Operations Vehicle (SOV) [75; 86:pg.4], or the Hypersonic Cruise Vehicle (HCV) [121]. A benefit of sub-orbital launch is that the launch vehicle can be selected uniquely based on the range to the target or other mission dependent variables. Additionally, development can be performed incrementally, meaning a lesser performance launch vehicle can support the CAV now, and later advancements in launch vehicle technologies can be directly applied to future CAV missions. HCV trajectory optimization is performed as a simplified, constant altitude, hypersonic launch vehicle. This analysis is applicable to a launch vehicle/CAV combined mission, as computed in [121], or as an initial guess to a CAV only reentry trajectory.

1.2.5 Computational Objectives. The objective of this research effort is to provide a solution using the computational capacity that could be aboard an autonomous reentry vehicle such as the CAV. The current belief is this would be comparable to the computing power of a desktop personal computer. The uplink capability of the vehicle would then only require receiving mission specific details, and not an entire mission solution.

1.2.5.1 Real-time updates. Even though the time from separation to target may be minimal, enroute mission updates may still be a possibility. Additional work in this field is being performed and is termed Real-Time Optimal Control (RTOC) [10; 79]. Therefore, rapid onboard computing is also justified for real-time updates synergistic with the vehicle inner-loop control system. The inner-loop system is assumed to have the capability to adjust for minor deviations due to wind while maintaining the computed trajectory profile. The research herein is compatible with, but does not address, RTOC and inner-loop guidance.

1.2.5.2 Autonomous Pre-conflict Analysis. Autonomy and minimal computation power are key elements to enabling hypothetical pre-conflict scenario analysis. A wargaming capability is a key element of the Course of Action (COA)

Analysis phase and COA Comparison phase of the Joint Air Estimate Process (JAEP) [103:pg.III-4] and [102:pg.1]. This research provides an enabling technology towards providing this wargaming objective.

1.3 Technology Advancement

In order to advance current technology and contribute uniquely to the engineering field, a problem or scenario must be identified that has not been previously achieved and has a target audience or customer. Section 1.1 introduced reentry trajectory generation work being done but also pointed out that the work is catered to commercial applications and neglects unique military concerns such as threat regions or geopolitical no-fly zones. With the military utility of the Global Positioning System (GPS), many UAVs such as Global Hawk and Predator [112:pg.2], achieve route definition through waypoints. Various trajectory optimization strategies navigating through waypoints have been studied [3; 46; 106; 113; 119] and are addressed in Section 2.2. The simulation environment SAVMOS, mentioned in Section 1.1, integrates one of the leading trajectory generation software packages [17; 29], but fails to provide autonomous trajectory generation, waypoint satisfaction, and threat avoidance. Therefore, the proposed enabling technology is to develop an autonomous trajectory generation algorithm to satisfy vehicle dynamics, survivability, waypoint, and threat or no-fly zone constraints as stated in Section 1.2. Autonomy allows for onboard real-time computations as well as military capability analysis, i.e. wargaming as addressed in Section 1.2.5.2. The case study vehicles for the work proposed herein are the HCV and the CAV described in Sections 1.2, 1.2.3, and 1.2.4 due to the HCV's cruise capability and the CAV's global strike capability; commensurate with the research interest expressed by AFRL, Air Vehicles Directorate (AFRL/VA) and AFRL, Space Vehicles Directorate (AFRL/VS).

1.4 Document Outline

The objectives of this research are to determine and verify a viable optimal solution technique that computes the minimum time reentry trajectory which strikes the target while satisfying the vehicle dynamics, heating limitation, intermediate way-points, and no-fly zone constraints. A unique aspects of this research is the combination of these multiple constraints into one solution, and verifying the optimality of a numerically generated solution. Before approaching the problem as a whole, each one of the independent components is researched. The previous research on any single task is summarized in Chapter II. In addition to the research summary is a description of how well that solution technique may contribute to the combined problem. The end of the chapter includes the decision to pursue the dynamic optimization solution technique and the justification of its application to this problem. Chapter III contains a generic dynamic optimization problem definition, followed by vehicle and mission specific constraints. The problem complexity begins with a simpler two-dimensional model and then increases to a more complex three-dimensional model. The simplest problem setup is the HCV at constant altitude and constant speed. Some complexity is then introduced by allowing the speed of the HCV to decrease throughout the trajectory. The CAV is the final vehicle considered with altitude no longer confined; thus with a change in altitude comes the inclusion of the heating constraint. Chapter IV contains the solution development, both analytical and numerical. In spite of the initial focus on the dynamic optimization technique, an advantageous 2-D geometric solution is also derived and presented. Following this contribution is the full 3-D numerical solution. Subsequently, the optimality of these numerical results is verified using the previously derived analytical criteria. An elaborate problem setup is required to prove optimality to the level of detail described above; however, once the numerical solution process is verified, a simpler setup is implemented. Before finishing this chapter, the numerical results from a streamlined, user representative, problem setup is compared to the previous, more rigorous, results. Chapter V provides the

conclusions to be drawn from the research herein, contributions, and comments on future research.

The appendices provide additional details to complement the material in the main body. These include specifics on the CAV aerodynamic model (Appendix A), nondimensionalization of the equations of motion (Appendix B), derivation of the shortest path between points (Appendix C), determination of the geometrically optimal waypoint passage point (Appendix D), important solution aspects of the pseudospectral methods (Appendix E), and strategies for efficient user implementations of the numerical solvers (Appendix F).

II. Previous Research

This research provides a synergistic solution encompassing three key technologies; trajectory generation, waypoint satisfaction, and threat or no-fly zone avoidance. The following addresses the research in each area separately and discusses the applicability to a combined solution.

2.1 Reentry Trajectory Generation

2.1.1 Space Shuttle. The foundation of much of the current lifting body reentry vehicle research [24; 55; 61; 81; 84; 89; 90; 91; 92; 122] is based on or at least baselined from the Space Shuttle [39]. The Shuttle trajectory solution was based on a drag acceleration envelope which represented the various vehicle limitations; such as surface temperature, dynamic pressure, load limits, and glide limitations as seen in Figure 1(a). The trajectory, as a function of velocity, is broken up into phases such as temperature control, equilibrium glide, and constant drag to traverse through the specified envelope, as seen in Figure 1(b).

(a) Flight Corridor   (b) Reference Drag Profile
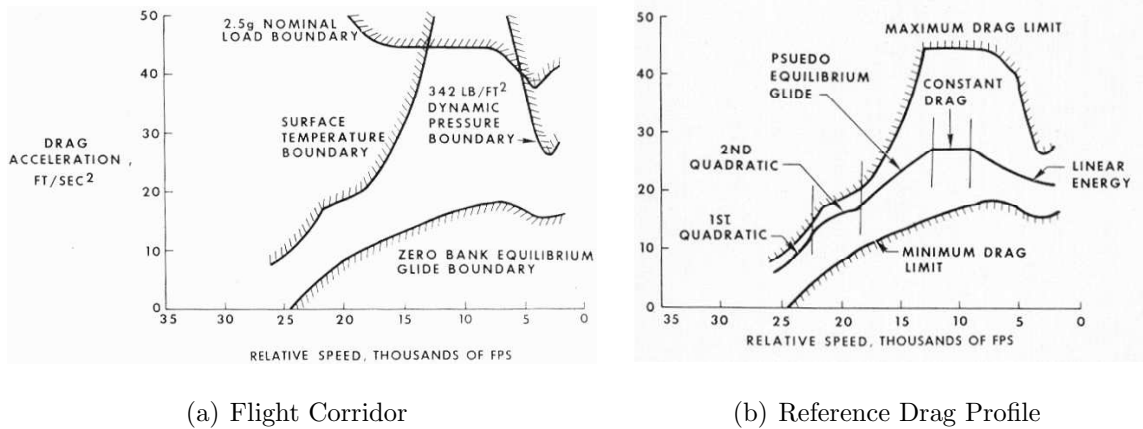
Figure 1:   Shuttle Entry - Operational Angle-of-Attack Profile. [39]

2.1.2 X-33 and X-37. The list of research referenced in the previous paragraph and [82] uses the X-33 as the test vehicle but others [1; 16; 28; 45; 60; 63] include the X-37 as an applicable vehicle. For reusable launch vehicles [83; 84], specifically X-40A [85], fault mitigation is another pursued topic. One paper [37]
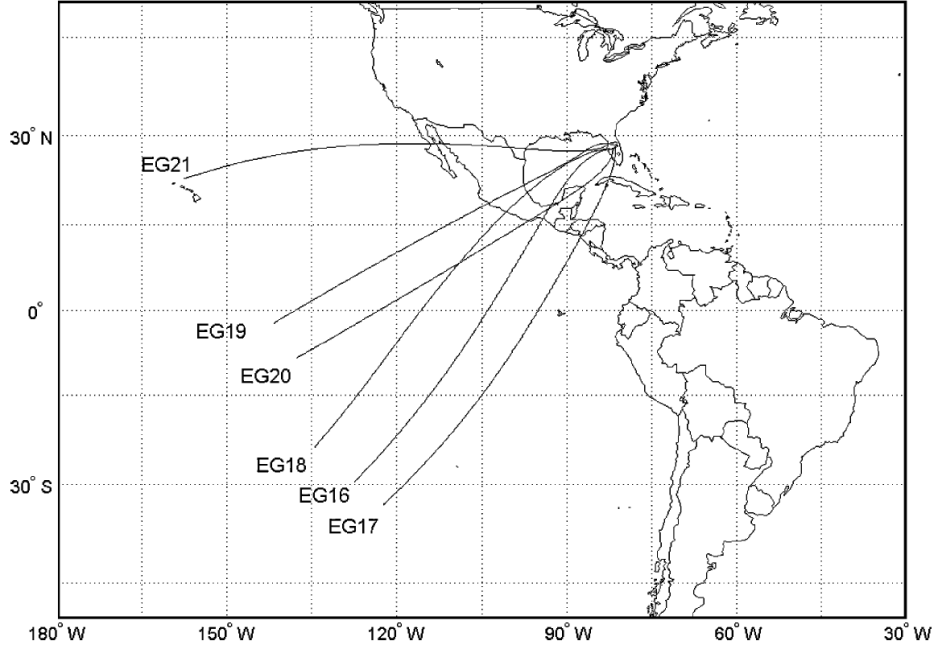
Figure 2:   X-33 - Ground Track for Six Analyzed Entries [81]

analyzes multiple reentry trajectory generation techniques and provides test results comparing five methods [24; 35; 81; 90; 122]. The core techniques are the baseline guidance method [35] like the Shuttle, the linear quadratic method [24], the numerical predictor-corrector method [122], the drag-energy three-dimensional method [81], and the quasi-equilibrium glide method [90]. The algorithms producing the best results were the quasi-equilibrium glide [90] and the Evolved Acceleration Guidance Logic for Entry (EAGLE) [54; 81] which are shown in Figure 2. The quasi-equilibrium glide, having apparent roots from pseudo-equilibrium glide [39], appears to be very promising; however, the EAGLE program has had more extensive testing and simulation work for validation. Furthermore, this is the algorithm currently incorporated in SAVMOS with `Matlab`® implementation. The Program To Optimize Simulated Trajectories II (POSTII) [71] is trajectory optimization and simulation tool [21; 65]. It has a two-point boundary value problem solver; however, there is not a readily available means to incorporated waypoints or no-fly zone constraints. The Optimal Trajectories by Implicit Simulation (OTIS) [59; 67] software by Glenn Research Center uses the newly emerging collocation method for satisfying the differential equations

10
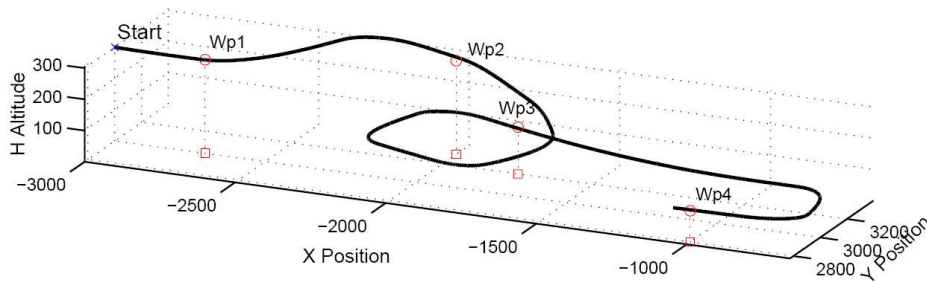
Figure 3:   Fuzzy Logic - Simulation of 4 Waypoints Trajectory [3]

and solving the two-point boundary value problem. This software solves for a set of parameters to optimize the specified trajectory criteria. One critic [62] comments that OTIS does not enforce continuity over phase transitions; however, the software package used herein can accept user defined functions to remain continuous across a phase break. This is described more in Section 3.5 and Appendix F.

2.1.3 Feedback Linearization.   Feedback linearization transforms a nonlinear system into a fully or partially linear system, and then uses linear design techniques to complete the control design [93:pg.204,207]. Feedback linearization can also be used to provide an entry tracking law [9]. Feedback linearization is most applicable when linearizing about a nominal point or trajectory; however, a nominal trajectory does not yet exist. Thus this approach is too susceptible to the non-linearities of this reentry trajectory problem.

2.2 Waypoints

2.2.1 Fuzzy Logic.   A non-traditional approach to waypoint control is a five-dimensional waypoint-based fuzzy guidance system (FGS) for unmanned aircraft vehicles [3; 46]. The heading and altitude for each waypoint are specified, either as a restriction or an enhancement. Figure 3 shows a spiral descent was correctly identified when two consecutive waypoints were otherwise non-flyable. This approach is valid, but the other facets of the proposed problem can not readily be incorporated into an overarching analytical formulation.

11
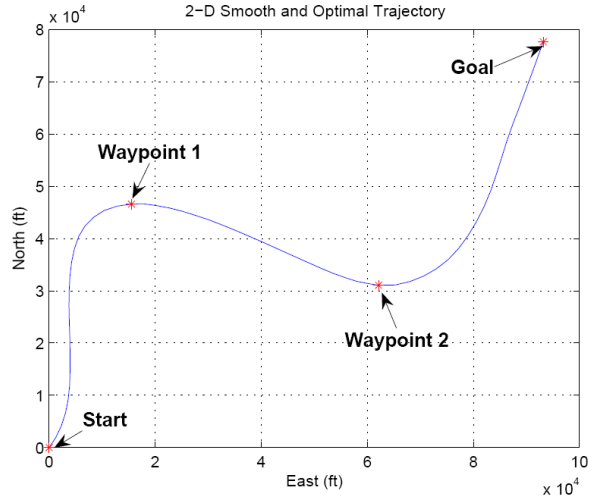
Figure 4:   Riccati Equation - Time Optimal Trajectory of the Horizontal Flight with Constant Speed [119]

2.2.2   Discontinuous Lagrange Multipliers.   A discontinuous Lagrange multiplier is one that has a jump, or discontinuity, at an interior-point constraint, e.g. a waypoint.   A discontinuous Lagrange multiplier arises when trying to solve a two-point boundary problem that now has one or more interior-point constraints [14:pg.101; 44:pg.275]. One author [119] combined discontinuous Lagrange multipliers with linearization about an initial guess and then propagated the Riccati differential equation to compute components for the Lagrange multipliers. This approach is appealing to the problem posed herein because analytical gradients can be directly computed since a method for computing the Lagrange multipliers exists. This approach is used for the two-dimensional analytical results and for validation of the three-dimensional numerical results.

2.2.3   Linearization and Propagating Riccati Equation.   Linearizing the system allows use of established linear optimization and control techniques. One example is establishing line segments between waypoints and then designing a linear quadratic regulator (LQR) for line-following guidance [113]. Bryson introduces a solution by Riccati equation [12:pg.206] and expands it to following a desired output [12:pg.217]. In another text [13:pg.93], Bryson presents the algorithm as a backward information

12

filter-smoother and this is the process used in Figure 4 [119]. A receding horizon planning strategy using Mixed-Integer Linear Programming (MILP) [106] is used to command a UAV through waypoints and threat zones. This appears very similar to the proposed research; however, this technique takes advantage of loiter circles and implements intermediate courses based on limited current information. The linearity of the system in the local region and the step size during an iteration can drastically affect the performance of a linearization technique; therefore, a linearized solution will not be used to find a solution for this research.

2.2.4 Steepest Descent. Steepest descent is one of the simplest gradient based algorithms, but it is sensitive to scaling parameters [13:pg.28,125,154] and is notoriously slow to converge for problems with long narrow valleys [58:pg.3-5]. For these reasons, alternate gradient based methods are pursued for this research.

2.2.5 Sequential Quadratic Programming. Sequential Quadratic Programming (SQP) is also gradient based, and SQP methods represent the state-of-the-art in nonlinear programming methods [58:pg.3-29]. By computing the Hessian, typically indirectly via the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) method [107:pg.293], second-order gradient information is used that dramatically improves performance over the first-order steepest decent method. As an example, a minimum-time-to-turn flight algorithm is computed for an F-4 fighter aircraft using SQP [66]. Due to the fast convergence, this is the core algorithm embedded in many numerical solution techniques. For example, an industry standard from Stanford, Sparse Nonlinear Optimizer (SNOPT), is based on SQP.

2.3 No-Fly Zones

Threat or no-fly zones are important in multiple contexts. The military threat zone is typically understood as a threat severity being proportional to the distance from the threat [49; 56; 100; 108], as seen in Figure 5. Other instances of no-fly zones or regions are obstacle avoidance [120] and air traffic avoidance [70]. Incorporation of
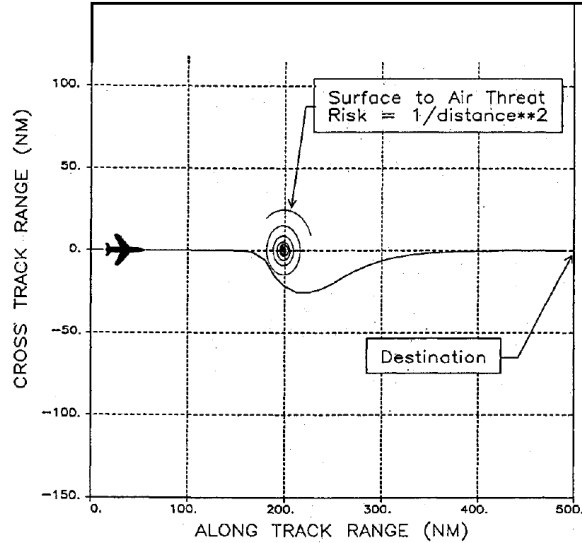
Figure 5:   Threats - Ground Track for Horizontal-Plane Optimization [108]

the no-fly zones, within the trajectory planning with waypoints is the subject of the research.

2.3.1 Voronoi Diagram.   A threat avoidance grid around threat sources is first segmented into equal threat distance segments, or more accurately described as a Voronoi Diagram [49], as shown in Figure 6. This approach allows for incrementally solving the solution, meaning a local solution exists before completing the terminal phase computation. The authors [49] use a spline algorithm, with results shown in Figure 6(b), to convert the sharp corners of the Voronoi diagram solution in Figure 6(a) to smooth flyable trajectories. Algorithms for creating these Voronoi diagrams are available in `Matlab`® [25; 64:pg.65]. This works well for threats, modeled as sources, but does not readily incorporate waypoints, which may be sinks in cost. Therefore, this technique is not incorporated within this research.

2.3.2 Barrier and Interior Point Methods.   This approach may be used for either threat/terrain avoidance [100] as shown in Figure 7, or air traffic avoidance [70] as shown in Figure 8. The barrier and interior point methods [5:pg.380] increase the cost as the constraint boundary is approached. The solution process starts on

14

(a) Initial Coarse Path   (b) Smoothing Algorithm Finished
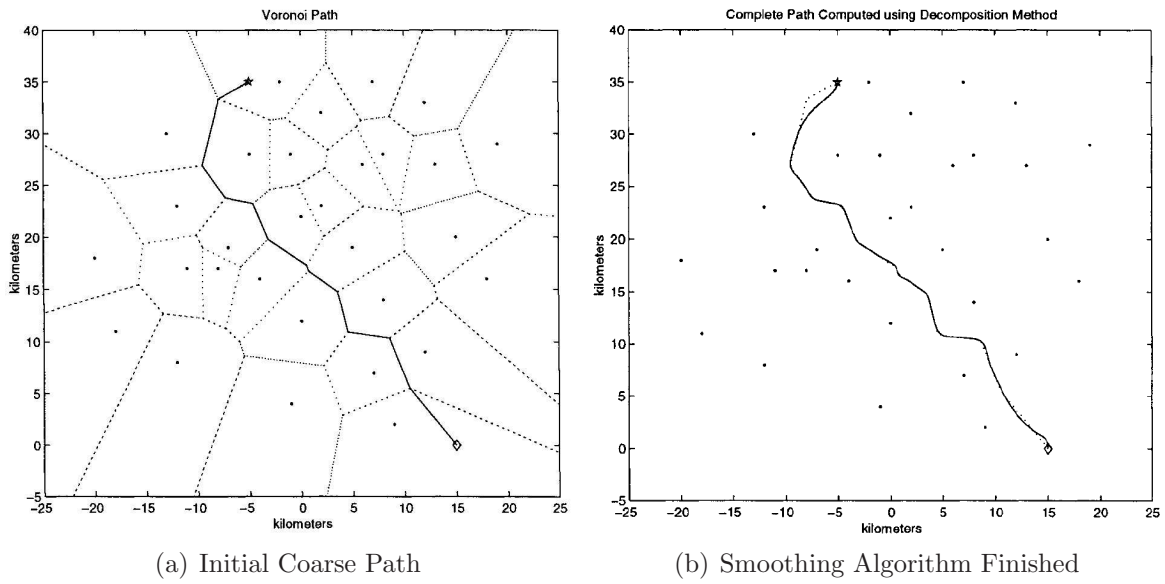
Figure 6:   Voronoi Diagram [49]
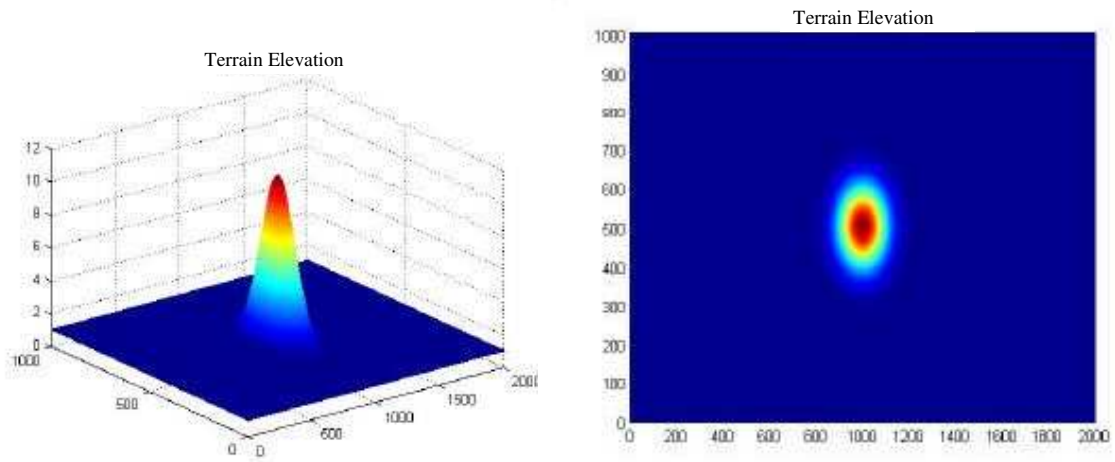
Figure 7:   Barrier - Terrain with Threats Formulated as an Exponential Function. On the Right is an Overhead View [100].
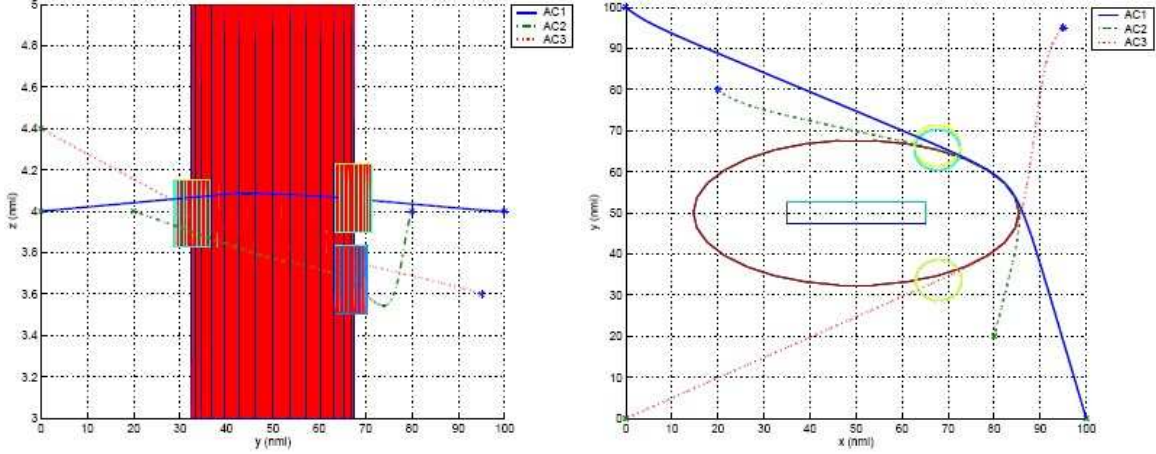
Figure 8: Interior Point - Projection onto yz and xy Plane of the Aircraft Trajectories when No-Fly Zone is Modeled as an Ellipse [70]

the interior, i.e. within the feasible region, and attempts to improve the cost. An exponential or $1/x$ function is typically applied that approaches infinity at the constraint boundary; therefore, such high cost at the boundary drives the solution to remain within the interior. A downside to this method is the cost function is either undefined or infinity for constraint violations; consequently, incidental iteration steps within the constraint could cause a solution algorithm to diverge. A similar approach uses a penalty function that increases cost for violating the constraint, for example $f = A\exp(-r^2/b)$ is the penalty shown in Figure 7. Here, A is amplitude, r is radius, and b is a scaling factor to adjust the width. Other functions such as a step or polynomials [64:pg.441] can also be used to penalize departure from the feasible region, i.e. violating the constraint. A function representing the constraint that has continuous first derivatives [94:pg.99] is conducive to analytical gradients. The analytical development herein does not use penalty functions; however, such weights for feasible violations are internal to the numerical algorithms in the form or merit functions [6; 8].

2.3.3 Receding Horizon and Artificial Intelligence. Several authors [52; 53; 106] approach the UAV using obstacle problems with a receding horizon controller (RHC). This technique is based on the most current information, i.e. waypoints and

16
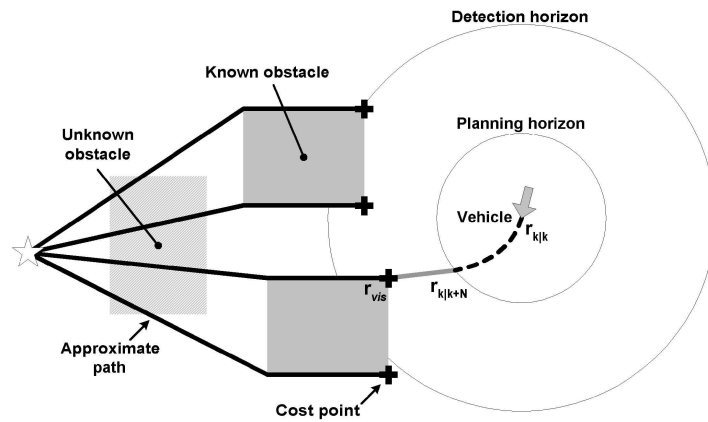
Figure 9:   Receding Horizon - Representation of a Cost-to-Go Function [53; 74]
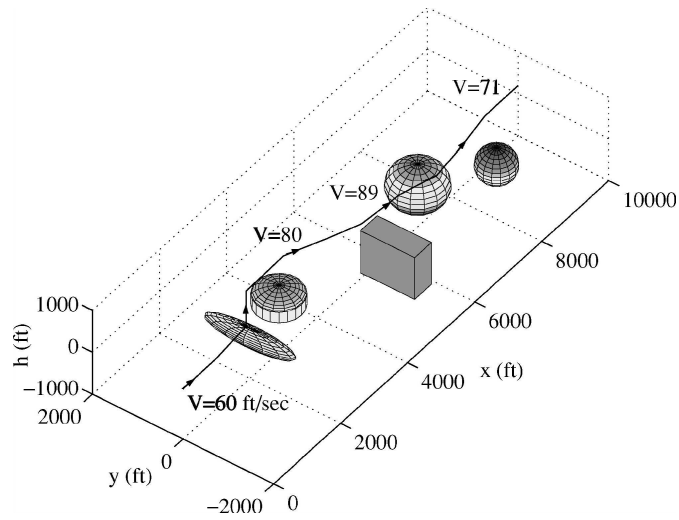
Figure 10:   A* Search-Optimal Flight Trajectory in Stationary Obstacles [120]

constraints, which may only extend out radially to a limited "detection horizon", to find a solution out to the "planning horizon" shown in Figure 9. The use of an admissible set of allowed solutions allows the vehicle to negotiate/reject feasible solutions but may be trapped by obstacles. Ensuring goal attainment, even in a disturbance environment, is the objective of these planners. The shortest path from each node and cost-to-go are similar to a weighted Voronoi diagram. This avoidance of block obstacles seems most applicable to small maneuverable UAVs, such as helicopters, in an obstacle rich urban environment. This is exactly the case in Figure 10 for trajectory planning using an A* search scheme in artificial intelligence [120]. For the research herein, the waypoints and threats are known throughout the trajectory; therefore, the receding horizon approach will not be used since the entire trajectory will be computed initially.

2.4 Pseudospectral Methods

The pseudospectral methods are generic numerical solution techniques, not specific to any one class of problems addressed in the previous sections. It's one piece of solving the optimal control problem via the direct method instead of the indirect method. Two authors [12; 14; 44] demonstrate many examples of the indirect approach; meaning solving a series of optimality conditions to then indirectly arrive at the optimal control. Except for very simple problems that have a purely analytical solution, discretization will certainly be part of the solution process, e.g. using numerical integration. Defining the optimality criteria has been called "dualization" [27] since it creates a set of costates or dual variables. The problem arise where the states are known at the initial time, and the costates are specified at the final time. The shooting method is one indirect approach that integrates the states forward and the costates backwards; continuing until converging to a solution that satisfies the initial and final conditions for the states and costates, respectively. This entails Runge-Kutta numerical integration which is a form of discretization. There are other ways to solve the problem by converting the unknown quantities of the indirect optimiza-

tion problem into a parameter optimization problem [43]; thus solving for a discrete number of variables. In summary, the indirect method specifies a set of optimality conditions first, then discretizes the problem to find the optimal control. Conversely, the direct method interchanges the dualization and discretization steps as described next.

There are at least five authors that use the direct approach to solve the optimal control problem, each have developed their own software package. Professor O. von Stryk's papers [95; 96] discuss the direct versus indirect method and provide example problems solved with his direct collocation software (DIRCOL). DIRCOL is also used in [50] to solve a Space Shuttle heating problem. John Betts [6; 7; 8] also converts the problem into a nonlinear program (NLP) and created the Sparse Optimal Control Software (SOCS) as a solver. The software package called Nonlinear Trajectory Generation (NTG) [68] focuses on solving a lower dimensional problem, also satisfying conditions at "knots" or collocation points. All of these software packages rely on the NLP solvers Nonlinear Program Solver (NPSOL) or SNOPT from the Stanford Systems Optimization Laboratory (SOL). I. Michael Ross [27; 77; 79] and David Benson et al. [4] use the Legendre-Gauss-Lobatto (LGL) and Legendre-Gauss (LG) pseudospectral methods, respectively. The pseudospectral method computes the states, controls, and derivatives at discrete nodes; then uses the Karush-Kuhn-Tucker (KKT) conditions to enforce optimality. The benefit to the pseudospectral method is that the solution to the differential equations is computed simultaneously across all nodes rather than sequentially via Runge-Kutta integration; therefore alleviating the potentially debilitating instability issue of the forward and backward integration in the shooting method. These authors' respective software packages are DIDO and Gauss Pseudospectral Optimal Control Software (GPOCS) [72]. One of the challenges to the direct method is obtaining the costate information. These software packages also have costate information; specifically, DIDO uses the Covector Mapping Theorem and GPOCS uses Costate Mapping Theorem to convert the discretized KKT multipliers to the continuous Lagrange multipliers, see Figure 11. Figure 11 also shows the

19

similarity between these latter two approaches. The Lagrange multipliers are the key to verifying optimality used within this research.

(a) DIDO [27]

(b) GPOCS [4]
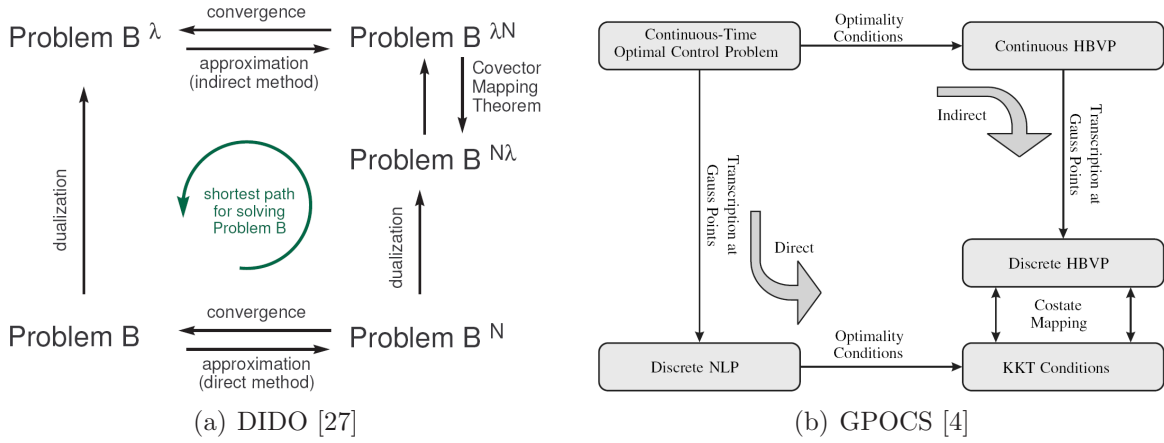
Figure 11:   Computation of the Continuous Lagrange Multipliers

2.5   Concluding Remarks

Various methods are shown to solve any single research objective; however, the challenge is finding a method to combine the vehicle dynamics, heating, waypoints, and no-fly zone constraints. Dynamic optimization can combine these constraints into a single problem, but many numerical solution techniques have pitfalls that hinder convergence. The direct method, using pseudospectral methods, avoids some of the numerical pitfalls, and thus is extremely appealing to solving the optimal control problem. Obtaining the costate information from the proposed direct method is critical to verifying optimality. Programs such as DIRCOL, DIDO, and GPOCS provide such costate information. The challenge within this research is to formulate the dynamic optimization problem, derive the optimality criteria, and verify that the numerical results match the derived optimality criteria. The problem structure is derived in Chapter III, followed by the numerical results and comparisons in Chapter IV.

III. Problem Definition and Assumptions

This chapter documents the generic dynamic optimization problem as well as the specific constraints used throughout this research. The simpler HCV model and mission are described first. This is used to build up to the more complex mission using the CAV vehicle. With the problem and assumptions clearly defined, the analysis and results follow in Chapter IV.

3.1 Generic Problem Statement

The overall objective of this research is to create an optimal trajectory, for a given vehicle, to strike a final target in minimum time. The cost or objective function defines optimality, which is mission dependent and typically defined by the user. For example, the user may want to arrive in minimum time or maximum energy. Additionally, equality and inequality constraints are required to define the given mission; thus, allowing incorporation of waypoints and no-fly zones. Other path limitations, such as thermal environment, may also exist. The vehicle itself is represented by the dynamics or equations of motion and any control limitations. These generic definitions are formulated into equations and used to solve for the optimal control, leading to the optimal trajectory. Therefore, the first step is to introduce the dynamic optimization nomenclature used throughout this research.

3.1.1 Dynamic Optimization. The continuous time, dynamic optimization problem with open final time is the particular problem of interest. The open final time allows time to be in the cost function, which is the intent for a minimum time problem. The advantage to the continuous time formulation, over discrete time formulation, is it allows the direct incorporation of the differential equations of motion. The equations of motion in Equation (1) are a function of the states, $\mathbf{x}(t) \in \mathbb{R}^n$, the controls $\mathbf{u}(t) \in \mathbb{R}^m$, and the independent variable $t \in \mathbb{R}$; where n is the number of states and m is the number of controls.

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) \tag{1}$$

In Bolza form [12], the objective or cost function, J, has two components. The first component $\phi(\mathbf{x}(t_d), t_d)$ is a cost defined at a discrete time t_d, or the sum of costs at discrete times, $\phi(\mathbf{x}(t_{d_1}), t_{d_1}) + \phi(\mathbf{x}(t_{d_2}), t_{d_2})$. For example, for final time t_f and final state $\mathbf{x}(t_f) \equiv \mathbf{x}_f$ the first term is $\phi(\mathbf{x}_f, t_f)$. The second component is the integrand cost L, taken from the initial time t_0 to final time t_f. The combined cost function is:

$$J = \phi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} L(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t \tag{2}$$

Additionally, there may be a terminal constraint ψ which is a function of the final state and the final time, and must satisfy:

$$\psi(\mathbf{x}_f, t_f) = \mathbf{0} \tag{3}$$

Certain path equality constraints C may also exist within the trajectory [14:pg.99; 44:pg.294]. Notice here that the equality constraint C is a function of the states and the controls:

$$C(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{0} \tag{4}$$

Inequality constraints are also important to this research. The inequality may be a function of either the states or the controls [14:pg.108-118]. The inequality constraints only appear in the solution during the times they are active, i.e. equal to zero. The state inequality constraint is $S(\mathbf{x}(t), t) \leq 0$. When the control constraint is active, the control inequality constraint in Equation (5) is a special case of Equation (4), i.e. $C(\mathbf{u}(t), t) = 0$. Equation (7) illustrates how active and inactive constraints are handled. The technique being formulated herein requires the equality constraint to be a function of the control; therefore, the nomenclature for C has been maintained whereas the distinction is made for S since it is not a function of the control. The separate state and control inequality constraints are:

$$S(\mathbf{x}(t), t) \leq 0 \qquad C(\mathbf{u}(t), t) \leq 0 \tag{5}$$

22

The goal of optimization is to determine a control to minimize the cost while satisfying the dynamics and constraints. In order to ensure the constraints are satisfied they are adjoined to the cost function using Lagrange multipliers; ν, λ, and μ. Notice that the following adjoined cost function $\bar{J}$ maintains the same value as Equation (2) if all of the constraints are satisfied. To simplify notation, the dependence of each variable is omitted, e.g. $\phi(\mathbf{x}_f, t_f)$ is simply written as ϕ. Thus the optimization problem is to determine the control $\mathbf{u}$ that minimizes the scalar cost $\bar{J}$:

$$\min_{\mathbf{u}} \bar{J} = \phi + \nu^T \psi + \int_{t_0}^{t_f} \left[L + \lambda^T (f - \dot{\mathbf{x}}) + \mu^T C \right] dt \tag{6}$$

To activate the constraints when $C = 0$, the applicable components of the multiplier μ take on the following values:

$$\mu \begin{cases} > 0, & C = 0, \\ = 0, & C < 0 \end{cases} \tag{7}$$

The state inequality constraint S appears to be missing; however, a method to incorporate this constraint is shown later. The calculus of variation dictates that the variations of the adjoined cost, taken with respect to each variable, must be zero in order for the solution to possibly be a minimum. Thus, the variation of the adjoined cost function must be:

$$\delta \bar{J} = 0 \tag{8}$$

Before presenting the results, some intermediate variables are defined. The Hamiltonian H is a scalar defined as:

$$H \equiv L + \lambda^T f + \mu^T C \tag{9}$$

and Φ is the scalar defined as:

$$\Phi \equiv \phi + \nu^T \psi \tag{10}$$

The time derivative of Φ is defined as:

$$\dot{\Phi} \equiv \Phi_{\mathbf{x}} \, \dot{\mathbf{x}} + \Phi_{t_f} \tag{11}$$

The following summarizes the necessary optimality conditions for the continuous time, free final time problem; the intermediate steps can be found in [12:pg.159]. The dependence notation is returned here to clarify the conditions that are functions of time versus those that apply only at the final time. The following shortened notation has been maintained; $\mathbf{x}$ is a simplification for $\mathbf{x}(t)$ and $\mathbf{u}$ is a simplification for $\mathbf{u}(t)$. Lastly, the subscripts represent a partial derivative with respect to that variable. The costates, λ, differential equations are:

$$\dot{\lambda}^T(t) = -H_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t) = -L_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t) - \lambda^T(t) f_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t) - \mu^T(t) C_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, t) \tag{12}$$

The final condition on the costates are:

$$\lambda^T(t_f) = \Phi_x(\mathbf{x}_f, t_f) = \phi_x(\mathbf{x}_f, t_f) + \nu^T \psi_x(\mathbf{x}_f, t_f) \tag{13}$$

The optimality criterion:

$$H_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, t) = L_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, t) + \lambda^T f_u(\mathbf{x}, \mathbf{u}, t) + \mu^T(t) C_u(\mathbf{x}, \mathbf{u}, t) = 0 \; \forall \; t \geq t_0 \tag{14}$$

and finally the transversality condition:

$$\dot{\Phi}(\mathbf{x}_f, t_f) + L(\mathbf{x}_f, \mathbf{u}_f, t_f) = 0 \tag{15}$$

Notice that S does not explicitly appear in these optimality criteria; however, Section 3.1.3 covers how S is to be included.

3.1.2 Discontinuous Lagrange Multipliers. A unique situation occurs when there are interior-point state constraints, i.e. constraints that apply at a single un-

specified time within the trajectory. This brings about the concept of discontinuous Lagrange multipliers. These discontinuities will also be called "jumps" in the costates [14:pg.101]. Assume there is only one interior-point constraint, occurring at the unspecified time t_1, as a function of the states and time:

$$N(\mathbf{x}(t_1), t_1) = \mathbf{0} \tag{16}$$

The time t_1 now represents the final time of the previous segment and the initial time of the next segment; thus, let t_1^- represent just before t_1 and t_1^+ just after t_1. Similar to the final condition on the costates in Equation (13), there is now a jump to accommodate the intermediate condition in Equation (16) [14:pg.103; 44:pg.278]:

$$\lambda^T(t_1^-) = \lambda^T(t_1^+) + \pi_n^T \frac{\partial N}{\partial \mathbf{x}(t_1)} \tag{17a}$$

$$H(t_1^-) = H(t_1^+) - \pi_n^T \frac{\partial N}{\partial t_1} \tag{17b}$$

The vector of multipliers π_n must be solved for to satisfy Equations (17a) and (17b). The finite number of interior-point constraints may be greater than 1; therefore, Equation (17) can be generalized to apply to any one of the jumps simply by changing t_1 to the applicable time of the interior-point constraint, e.g. t_i or t_j.

3.1.3 Path Inequality and Equality Constraints. The path inequality constraint is introduced in Equation (5) as a function of the state and time; it is repeated here for convenience:

$$S(\mathbf{x}(t), t) \leq 0 \tag{18}$$

Since the control does not explicitly appear in Equation (18), the time derivative of S is taken until the control does appear. This is demonstrated in the Breakwell problem in Appendix F. If it takes q derivatives for the control to appear, this time derivative is written as $S^{(q)}$. For the special case where this is the only constraint, then $C = S^{(q)}$

and Equation (9) becomes:

$$H(\mathbf{x}, \mathbf{u}, t) = L(\mathbf{x}, \mathbf{u}, t) + \lambda^T f(\mathbf{x}, \mathbf{u}, t) + \mu^T S^{(q)}(\mathbf{x}, \mathbf{u}, t) \tag{19}$$

This process adds interior-point constraints for all of the $q - 1$ derivatives which each must equal zero. Thus, the interior-point constraint vector in Section 3.1.2 becomes:

$$N(\mathbf{x}(t_1), t_1) = \begin{bmatrix} S(\mathbf{x}(t_1), t_1) \\ S^{(1)}(\mathbf{x}(t_1), t_1) \\ \vdots \\ S^{(q-1)}(\mathbf{x}(t_1), t_1) \end{bmatrix} \tag{20}$$

This interior-point constraint occurs at some time t_1, which for the path constraint, is the point of boundary contact. This is also characterized by a jump in the costates occurring when the constraint becomes active at time t_1. Since $S^{(q)}$ is a function of the control, it may be possible to maintain $S^{(q)} = 0$; this represents being on the constrained arc. Unlike entering the constrained arc, there is no jump in the costates upon exiting the constrained arc. Also, since $S^{(q)}$ is a function of the states, the partial derivative $S_x^{(q)}$ appears in the propagation of the costate equations, Equation (12). The optimal control continues to be solved from Equation (14). If the computed optimal control would lead to violating the constraint, then the control to remain on the constraint is used instead. Therefore, the optimal trajectory may follow the constraint, or simply touch the constraint, based on whichever minimizes the Hamiltonian without violating the constraint.

This section defined the continuous time dynamic optimization problem with free final time. Equality and inequality constraints were addressed as a function of the controls or the states. The next step is to specify the particular mission requirements in terms of constraints to be incorporated into a combined dynamic optimization problem for this research. The following sections address the mission, the simpler HCV model, and then progress to the more complex CAV model.

3.2 Mission Assumptions

The overall problem has mission components and vehicle components. The mission components include the target, waypoints, and no-fly zones; whereas the vehicle components include the equations of motion, control limitations, and aerodynamic heating limitations. The following mission assumptions are presented first to begin to scope the problem:

1. The waypoints are specified in the desired sequence.

2. Waypoint passage must be directly overhead.

3. Inner-loop control is available. Only the outer-loop or trajectory generation is addressed.

4. Waypoints are sufficiently spaced such that no two are within the turn radius of the vehicle.

5. Altitude for waypoint passage is not specified.

6. No-fly zones are specified as circular exclusion zones with infinite altitude.

7. No-fly zones must not be violated.

8. Target coordinates and final altitude are specified.

Item 1 ensures the user's mission definition is not altered. Item 2 avoids modifying the solution by manipulating the penalty of a near miss. Item 3 scopes this research effort to focus on the primary trajectory optimization objective. Item 4 characterizes the expected performance of a hypersonic vehicle. Item 5 attempts to avoid over specifying the problem; furthermore, waypoint altitude does not enhance the mission objectives. Item 6 simplifies the derivations; however, other shapes could alteratively be incorporated. Item 7, like item 2, avoids modification of the solution by manipulating the penalty of a slight penetration of a no-fly zone. Item 8 ensures the solution does not place the vehicle at such an excessive altitude that spike heating becomes a factor during a rapid final descent, or leave the vehicle with insufficient altitude to

perform any required terminal maneuvers. The following sections cover the vehicle specific assumptions and individual vehicle descriptions.

3.3 Hypersonic Cruise Vehicle (HCV)

3.3.1 Vehicle Description. As mentioned in the Introduction the HCV is a proposed hypersonic global strike vehicle. For this study its thrust provides the ability to maintain altitude to simplify the equations of motion by removing the variation in altitude. Thrust is also used to maintain constant speed or maintain constant deceleration. For this research thrust is not a control variable, it is merely used conceptually to validate the constant altitude, speed, and deceleration simplifications. The maximum available thrust will dictate the achievable flight envelope, i.e. altitude and velocity. To determine the validity of a trajectory, operational limitations are derived from a similar aircraft. The National Aerospace Plane (NASP) (UX-30) [121] may be used due to its unclassified data; however, the SR-71 is used here since it is a proven platform and the data is readily available [101].

The stated objective of 9,000 nautical miles (nmi) in 2 hours only partially defines the flight envelope; therefore, some addition assumptions are made and Mach numbers are computed for comparison to other aircraft. A representative flight altitude of 100,000 ft is chosen since it was the approximate test altitude for the unmanned hypersonic Hyper-X (X-43A) test vehicle [98]. The range and time objectives equate to 2.315 km/s, which at flight level (FL1000) is Mach 7.66. In the Operating Limitations section of the SR-71 Flight Manual [101], equivalent airspeed, expressed in knots equivalent airspeed (KEAS), is used to specify a minimum acceptable airspeed over a range of altitudes and Mach numbers. This minimum is used to verify sufficient airspeed to sustain level flight or engine performance. This minimum acceptable airspeed is $V_{min} = 310$ KEAS, which equates to Mach 4.47 at FL1000. If a trajectory reaches the target at an airspeed less than V_{min} it is considered invalid since the constant altitude assumption has been violated. For example, a flight starting at an initial airspeed of 2.315 km/s at FL1000, as described above, with a deceleration of 0.1

m/s^2 over 2 hours has a final airspeed of 366 KEAS or Mach 5.28, thus the level flight assumption remains valid. A deceleration profile is chosen to demonstrate the effects of non-constant velocity which leads to non-constant turn radii. Also, deceleration will be a component of the more complex CAV model. The bank angle σ is limited to $\pm 20°$ due to wing loading, stability, and/or inlet airflow tolerances. A constant altitude is maintained by ensuring the vertical component of lift equals the vehicle weight i.e. $Lift = mg/\cos\sigma$. The thrust used to counteract the drag associated with the additional lift due to a non-zero bank angle is $Thrust = m\left(a + g/(\cos\sigma C_L/C_D)\right)$. These derived parameters are used as a representative HCV for demonstrating results.

3.3.2 Mission Profile. For the analysis, a particular mission is specified in Table 1 by its initial location, intermediate waypoints, no-fly zone(s), and final destination or target. The waypoints represent specified mission critical locations that must be flown over precisely, either for reconnaissance or for multiple payload deliveries. It's arguable that a vehicle may be limited to level flight for imaging or payload deployment; however, due to the long durations of the turns, a short level segment within the turn is approximated here as a continuous turn. Therefore, in addition to no heading angle constraint, there is also no bank angle constraint at the waypoints. The no-fly zone coordinates are the center of the keep out circle with radius specified.

Table 1:   HCV Mission Description

Descriptor	Latitude	Longitude	Radius
Initial	N 28° 34′	W 80° 38′	
Waypoint 1	N 52° 11′	W 43° 46′	
Waypoint 2	N 17° 01′	W 20° 33′	
No-Fly Zone	N 22° 39′	E 11° 03′	840 nmi
Target	N 33° 46′	E 29° 34′	
h_0=100,000 ft=30.48 km, V_0=2.315 km/s, θ_0=0°			

The velocity is V; and the heading angle is θ, measured positive going counter-clockwise from the x-axis.

3.3.3 HCV Assumptions. The assumptions made for the HCV simplify the analytic derivation of the optimality conditions. This simplification makes it easier to follow the derivation, and helps add intuition into the final results. The assumptions are:

1. Altitude is maintained regardless of velocity.

2. Deceleration is either zero or constant.

The first assumption enforces constant altitude; however, post run analysis assesses the validity of the flight profile based on an acceptable minimum velocity. The second assumption decouples the bank angle and changes in velocity. Realistically, an increase in bank angle would decrease the vertical component of lift and the angle-of-attack would have to increase to provide enough lift to maintain altitude. The increase in angle-of-attack would then result in an increase in drag resulting in a decrease in velocity. Again, the intent of this simpler model is to gain insight into the dynamic optimization problem, not to produce a high fidelity HCV model.

3.3.4 Cost Function. The objective is to minimize flight time. The cost can therefore be expressed as t_f or as $\int_0^{t_f} 1 \, \mathrm{d}\, t$. The former seems to produce slightly faster results; possible because the integral of 1 internally requires an additional state. The results should obviously be the same with either expression, so for this research the cost function is:

$$J = t_f \tag{21}$$

3.3.5 2-D Equations of Motion. The 2-D equations of motion confine the flight profile to a constant altitude, horizontal plane. Also, the control u is a function of the bank angle normalized by the tangent of the maximum bank angle (σ_{max}), $u = \tan \sigma / \tan \sigma_{max}$. The purpose for the normalization is to easily identify the minimum

and maximum, i.e. ∓ 1. The units for the states and time are nondimensional in order to facilitate numerical stability. The conversion from dimensional to nondimensional is in Appendix B as Equation (B.17). The 2-D HCV equations of motion are:

$$\dot{x} = V \cos \theta \tag{22a}$$

$$\dot{y} = V \sin \theta \tag{22b}$$

$$\dot{\theta} = \frac{\tan \sigma_{max}}{V} u \tag{22c}$$

$$\dot{V} = a \tag{22d}$$

Where a is a constant. Now define

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} & \dot{V} \end{bmatrix}^T = f(\mathbf{x}, \mathbf{u}, t) \tag{23}$$

3.3.6 Control Constraints.   For this model of the HCV the only control is the normalized bank angle u. As described in Section 3.3.1, the maximum bank angle is $\pm 20°$. In Section 3.3.5 the control is normalized to produce an acceptable range for u, i.e. $-1 \le u \le 1$. To define an equality constraint as a function of the control, as described in Section 3.1.1, the control constraint is:

$$C(\mathbf{u}(t), t) = \begin{bmatrix} u - 1 \\ -u - 1 \end{bmatrix} \le \mathbf{0} \tag{24}$$

3.3.7 Terminal Constraints.   The terminal constraint for the HCV are merely the specified final coordinate $[x_f, y_f]$, i.e. the target. Altitude does not change so it does not need to be included:

$$\psi(\mathbf{x}(t_f), t_f) = \begin{bmatrix} x(t_f) - x_f \\ y(t_f) - y_f \end{bmatrix} \tag{25}$$

3.3.8 Path Inequality Constraints.   The no-fly zone assumption in Section 3.2 states that the no-fly zones are circular exclusion zones. Since there can be multiple no-fly zones each one is indexed j. Each no-fly zone is characterized by the coordinate of its center, $[x_{cj},\ y_{cj}]$, and its radius, R_j. Thus, to remain outside the no-fly zone the current distance from the center of the no-fly zone must be equal to or greater than the no-fly zone radius. Let the displacement in the x-direction and y-direction be, $\Delta x_j = x(t) - x_{cj}$ and $\Delta y_j = y(t) - y_{cj}$, respectively. In equation form, consistent with Section 3.1.3, each no-fly zone inequality constraint is:

$$\begin{matrix} S(\mathbf{x}(t),t) \\ j = 1, 2, ..., j_{\text{end}} \end{matrix} \quad = \quad \frac{1}{2}\left(R_j^2 - \Delta x_j^2 - \Delta y_j^2\right) \leq 0 \qquad (26)$$

The value j_{end} is the user specified finite number of no-fly zones. This constraint is in effect for all time t; however, the next section shows there is also an unspecified discrete time t_j associated with each no-fly zone.

3.3.9 Interior-Point Constraints.   The waypoints and no-fly zones both create interior-point[†] constraints. The waypoints are already the definition of an interior-point constraint; that is a specified function of the states at a discrete unspecified time that must equal zero. Since there may be multiple waypoints, each one is indexed i, thus the individual waypoint passage occurs at time t_i. The constraint to fly over the waypoint at time t_i implies the vehicle coordinates, $[x(t_i),\ y(t_i)]$, are equal to the specified waypoint coordinates, $[x_i,\ y_i]$. The interior-point constraints, for the finite number of waypoints i to i_{end}, are thus:

$$\begin{matrix} N(\mathbf{x}(t_i),t_i) \\ i = 1, 2, ..., i_{\text{end}} \end{matrix} \quad = \quad \begin{bmatrix} x(t_i) - x_i \\ y(t_i) - y_i \end{bmatrix} = \mathbf{0} \qquad (27)$$

[†]An interior-point within this research corresponds to a time along the trajectory between the initial and final time, thus creating a multi-*point* boundary value problem. Hence, as referenced herein, an interior-point constraint is not associated with the interior point method used within the field of linear programming.

Section 3.1.3 demonstrates how the no-fly zone inequality constraint can produce interior-point equality constraints. Specifically, the time derivative of the inequality constraint, Equation (26), must be taken until a control in $\mathbf{u}$ appears. For the HCV, u appears in the second derivative; therefore, the interior-point constraint associated with each no-fly zone consists of the zero derivative S and the first time derivative $S^{(1)}$. These are tangency requirements, meaning the boundary must be approached at a tangent or a violation will occur. This takes the same form as the waypoint constraints N but M will be used for distinction:

$$
\begin{array}{c}
M(\mathbf{x}(t_j), t_j) \\
j = 1, 2, ..., j_{\text{end}}
\end{array}
=
\left[
\begin{array}{c}
S(\mathbf{x}(t_j), t_j) \\
S^{(1)}(\mathbf{x}(t_j), t_j)
\end{array}
\right]
=
\left[
\begin{array}{c}
\frac{1}{2} \left(R_j^2 - \Delta x_j^2 - \Delta y_j^2 \right) \\
-\Delta x_j V \cos \theta - \Delta y_j V \sin \theta
\end{array}
\right]
= \mathbf{0} \quad (28)
$$

Also, the multipliers in Equation (17) are renamed π_m:

$$
\lambda^T(t_j^-) = \lambda^T(t_j^+) + \pi_m^T \frac{\partial M}{\partial \mathbf{x}(t_j)} \tag{29a}
$$

$$
H(t_j^-) = H(t_j^+) - \pi_m^T \frac{\partial M}{\partial t_j} \tag{29b}
$$

This equality constraint S differs from the inequality constraints in Equation (26) since this S only applies at the time of no-fly zone boundary contact t_j. The next section handles the second derivative of the no-fly zone constraint which must be zero to remain on the no-fly zone.

3.3.10   Path Equality Constraints.   As derived in Section 3.1.1, a state equality constraint is adjoined to the Hamiltonian, Equation (19). For this problem a state equality constraint, generically expressed as $C(\mathbf{x}, \mathbf{u}, t)$ in Section 3.1.1, is required to remain on the no-fly zone constraint arc until returning to the optimal bang-level-bang control law; specifically, the second derivative of the no-fly zone must remain

zero until departing the no-fly zone boundary:

$$
\begin{aligned}
C(\mathbf{x}, \mathbf{u}, t) &= S^{(2)}(\mathbf{x}, \mathbf{u}, t) \\
&= -V^2 + u(\Delta x_j \sin\theta - \Delta y_j \cos\theta)\tan\sigma_{max} + (-\Delta x_j \cos\theta - \Delta y_j \sin\theta)a \\
&= 0
\end{aligned}
\tag{30}
$$

The partial derivative $S_{\mathbf{x}}^{(2)}$ is used to propagate the costates.

3.3.11 HCV Summary. The cost, dynamics, control limitations, waypoints, and no-fly zones have all been expressed in terms of a dynamic optimization problem. Three types of state equality constraints have been shown; interior-point constraints, path equality constraints or tangency requirements, and maintaining boundary contact. The principal difference is interior-point constraints and tangency requirements occur at a single time, whereas the adjoined q time derivative path constraints apply the entire time the particular constraint is active, i.e. equal to zero. This distinction dictates the form of the solution as either a jump in the costates, as in Equation (17), or a propagation of the costates, as in Equation (12). The analytical solution to this simpler HCV problem is found in Chapter IV, and the more complex CAV problem setup is presented in the next section.

3.4 Common Aero Vehicle (CAV)

The CAV is an operationally representative vehicle for this research. It represents a hypersonic reentry vehicle with crossrange capability necessary to acquire waypoints and avoid no-fly zones. It has no thrust; therefore, the reentry profile must avoid exceeding a specified limit of the heating rate at the stagnation point. The following describes the vehicle model and expresses the CAV mission in terms of dynamic optimization constraints.

3.4.1 Vehicle Description. Reference [69] describes a low-lift CAV and a high-lift CAV. The high-lift CAV, CAV-H, is modeled here to extend the crossrange

capability. The full aerodynamic database and characteristic parameters are presented in Appendix A. The control and heating limitations are realistic, but are chosen to force boundary contact in order to verify different forms of the solution.

3.4.2 Mission Profile. A target nearly halfway around the globe is used to represent a global strike mission. A waypoint in the middle of the Atlantic represents a telemetry station that can validate the vehicle navigation and control to terminate a rogue vehicle if necessary. The radius of the first no-fly zone is intentionally chosen to be much smaller than the turn capability of the CAV, thus forcing the solution to only be able to contact the boundary at a single point. The next waypoint represents a secondary mission target for reconnaissance or payload delivery. The last no-fly zone has a large enough radius to allow the boundary to be followed if optimal, and is large enough to force full control authority in order to clear the no-fly zone and still make it to the target. The heating constraint limitation is set low enough such that the unconstrained heating problem exceeds the value, thus the constrained solution must incorporate the limitation. The initial conditions are approximately those presented in [69]. The specified final altitude is the approximate altitude that the Space Shuttle uses to complete its entry guidance phase [39]. Table 2 provides the mission objectives:

Table 2: CAV Mission Description

Descriptor	Latitude	Longitude	Radius
Initial	N 28° 35.286′	W 80° 40.194′	
Waypoint 1	N 34° 2.810′	W 27° 18.430′	
No-Fly Zone 1	N 20° 15.513′	W 3° 27.588′	960 nmi
Waypoint 2	N 33° 13.298′	E 41° 41.266′	
No-Fly Zone 2	N 55° 43.849′	E 58° 33.688′	1500 nmi
Target	N 31° 36.653′	E 65° 42.016′	
h_0=122 km, V_0=24,000 ft/s=7.3152 km/s, γ_0=-1.5° θ_0=4°			

Altitude is h. The flight path angle γ, as depicted in Figure 12 [41], is the angle with respect to the local horizontal, measured positive away from the Earth.
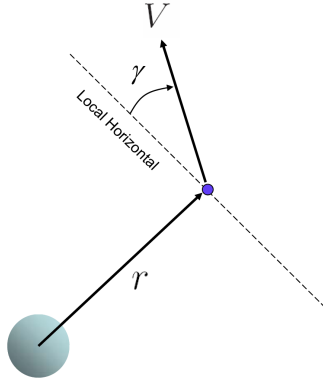
Figure 12:   Flight Path Angle Defined [41]

3.4.3   CAV Assumptions.   The following assumptions simplify the 3-D equations of motion and maintain the x, y, and θ states presented for the 2-D case. These simplifications do not indicate limitations to this solution process, they are intended to make the derived analysis more intuitive and easier to follow. The CAV 3-D model assumptions are:

1. Flat, non-rotating Earth

2. Gravity is constant

3. Flight path angle is small

4. Drag is the dominate deceleration term

5. Coefficient of Lift (C_L) and Coefficient of Drag (C_D) are only a function of angle-of-attack

6. Control is bank angle and angle-of-attack, both limited

7. Atmospheric density is modeled as a simple exponential

8. Heat flux at the stagnation point is limited

9. G-loading and total heat are not addressed

Ignoring the rotation of the Earth is a common assumption seen in [41; 97; 109; 110]. The flat Earth approximation is most applicable to lifting bodies with a low flight path angle as seen in [114:pg.248; 87; 73:pg.204]. The flat Earth model is used as a proof of concept and to provide commonality with the 2-D derivation; however, a spherical Earth model would be more appropriate for computing operationally representative results. The small flight path angle leads to the small angle approximations. For hypersonic vehicles, C_L and C_D are often assumed independent of Mach Number, i.e. simply a function of angle-of-attack [111]. The estimated fit to the CAV aerodynamic data is in Appendix A.

3.4.4 Cost Function. The minimum time objective function from the HCV in Section 3.3.4 remains the cost function for the CAV:

$$J = t_f \tag{31}$$

3.4.5 3-D Equations of Motion. The equations of motion represent a flat Earth model with zero Earth rotation. A small angle approximation for the flight path angle implies $\cos\gamma \approx 1$ and $\sin\gamma \approx \gamma$. It is also assumed that the drag term is the dominate term in the $\dot{V}$ equation, i.e. the component of gravity in the velocity direction is negligible compared to the aerodynamic drag. This last assumption is consistent with the small angle approximation for γ. The nondimensional equations of motion originating from [114:pg.249; 87] and derived in Appendix B are:

$$\dot{x} = V\cos\theta \tag{32a}$$

$$\dot{y} = V\sin\theta \tag{32b}$$

$$\dot{h} = V\gamma \tag{32c}$$

$$\dot{V} = -\frac{BV^2 e^{-\beta r_0 h}\left(1 + c_\ell^2\right)}{2E^*} \tag{32d}$$

$$\dot{\gamma} = BVe^{-\beta r_0 h}c_\ell\cos\sigma - \frac{1}{V} + V \tag{32e}$$

$$\dot{\theta} = BVe^{-\beta r_0 h}c_\ell\sin\sigma \tag{32f}$$

37

The constant B is a function of the vehicle parameters, atmospheric constant, and initial conditions. The constant E^* is the maximum lift-to-drag ratio. The controls are bank angle, σ, and the fraction of C_L^*, c_ℓ. The coefficient C_L^* is the coefficient of lift that produces the maximum lift-to-drag ratio. More details are contained in Appendix B.

3.4.6 Control Constraints. The controls are bank angle, σ, and normalized coefficient of lift, c_ℓ. The bank angle is limited to $\pm 60°$ to represent stability limitations, and to force the control to reach the imposed limit for this research. The maximum limitation of $c_\ell = 2$ is derived from the aerodynamic data in Appendix A. The minimum $c_\ell = 0$ maintains proper orientation for the thermal protection system. Thus the control limitations are:

$$C(\mathbf{u}, t) = \begin{bmatrix} \sigma - \pi/3 \\ -\sigma - \pi/3 \\ c_\ell - 2 \\ -c_\ell \end{bmatrix} \leq \mathbf{0} \tag{33}$$

3.4.7 Terminal Constraints. The terminal constraints for the CAV are the target coordinates and the specified final altitude h_f:

$$\psi(\mathbf{x}(t_f), t_f) = \begin{bmatrix} x(t_f) - x_f \\ y(t_f) - y_f \\ h(t_f) - h_f \end{bmatrix} \tag{34}$$

3.4.8 Path Inequality Constraints. The path inequality constraints for the CAV are the no-fly zones and the maximum stagnation point heat rate. The no-fly zone discussed in Section 3.3.8 is repeated here for convenience:

$$\begin{matrix} S(\mathbf{x}(t), t) \\ j = 1, 2, ..., j_{\text{end}} \end{matrix} = \frac{1}{2} \left(R_j^2 - \Delta x_j^2 - \Delta y_j^2 \right) \leq 0 \tag{35}$$

The function for the heating constraint is found in [41; 109], which has its origin from [15], and is also discussed in [34]. The dimensional form of the equation for $\dot{q}_{s\,dim}$ is:

$$\dot{q}_{s\,dim} = \frac{k_{dim}}{\sqrt{r_{\text{nose}}}} \left(\frac{\rho}{\rho_{sl}}\right)^{1/2} \left(\frac{V_{dim}}{\sqrt{g_0\,r_0}}\right)^3 \tag{36}$$

The dimensions are based on the converted units of the constant k_{dim}, which is 17,000 Btu ft$^{-3/2}s^{-1}$ [15], and vehicle dependent radius of the nose, r_{nose}. The dimensional maximum heat flux, $\dot{q}_{s\,max}$, is used to nondimensionalize Equation (36). The nondimensional altitude h and velocity V are defined in Appendix B. The nondimensional heating rate at stagnation point, $\dot{q}_s$, in terms of the nondimensional states h and V is:

$$\dot{q}_s = K e^{-\beta r_0 h/2} V^3 \tag{37}$$

where $K = k_{dim} e^{-\beta(r_0-R_\oplus)/2}/(\sqrt{r_{\text{nose}}}\,\dot{q}_{s\,max})$ making the maximum heat flux $\dot{q}_s = 1$. The heating path inequality constraint will have the same form as S in Equation (35); however, Q is used for distinction. There is no minimum heating limitation; however, from Equation (37) $\dot{q}_s$ is always ≥ 0. Thus the heating inequality constraint is:

$$Q(\mathbf{x}(t),t) = [\dot{q}_s - 1] = \left[K e^{-\beta r_0 h/2} V^3 - 1\right] \leq 0 \tag{38}$$

This Q is not related to total heating which is often represented by this symbol. For this research the total heating is considered a vehicle design parameter and not a trajectory optimization parameter.

3.4.9 Interior-Point Constraints. Since the flat Earth model maintains the states x and y, the waypoint interior-point constraint from Equation (27) still applies:

$$\begin{array}{c} N(\mathbf{x}(t_i), t_i) \\ i = 1, 2, ..., i_{\text{end}} \end{array} = \left[\begin{array}{c} x(t_i) - x_i \\ y(t_i) - y_i \end{array}\right] = \mathbf{0} \tag{39}$$

39

Similarly, the derived no-fly zone interior-point constraint, also from Section 27, is still:

$$M(\mathbf{x}(t_j), t_j) \atop j = 1, 2, ..., j_{\text{end}} = \begin{bmatrix} S(\mathbf{x}(t_j), t_j) \\ S^{(1)}(\mathbf{x}(t_j), t_j) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \left(R_j^2 - \Delta x_j^2 - \Delta y_j^2 \right) \\ -\Delta x_j V \cos \theta - \Delta y_j V \sin \theta \end{bmatrix} = \mathbf{0} \quad (40)$$

It can be seen that the heating constraint, Equation (38), is not a function of either control. Therefore, as described in Section 3.1.3, time derivatives must be taken until a control explicitly appears. This only requires the first time derivative, thus the interior-point constraint derived from the heating constraint Q up to the $q - 1$ derivative is simply:

$$Q(\mathbf{x}(t_k), t_k) \atop k = 1, 2, ..., k_{\text{end}} = [\dot{q}_s(t_k) - 1] = \left[K e^{-\beta r_0 h(t_k)/2} V(t_k)^3 - 1 \right] = 0 \quad (41)$$

This equation only requires satisfaction upon contact with the heating constraint at time t_k, where k represents the number of contacts with the heating constraint.

3.4.10 Path Equality Constraints.   The path equality constraints apply while a constraint is active, i.e. while the inequality constraint *equals* zero. The control to remain on the constraint arc is used if the optimal control cannot be maintained due to constraint violation. Therefore, if on the no-fly zone constraint, the second time derivative of the inequality constraint must equal zero:

$$
\begin{aligned}
C(\mathbf{x}, \mathbf{u}, t) &= S^{(2)}(\mathbf{x}, \mathbf{u}, t) \\
&= -V^2 + (\Delta x_j \sin \theta - \Delta y_j \cos \theta) \, \dot{\theta} + (-\Delta x_j \cos \theta - \Delta y_j \sin \theta) \, \dot{V} \\
&= 0 \quad (42)
\end{aligned}
$$

The form of this constraint differs from the HCV problem since the CAV equations of motion are different. The term in front of $\dot{V}$ is a multiple of $S^{(1)}$, thus will be zero; however, it is retained in order to take the partial derivative with respect to $\mathbf{x}$ to propagate the costates. Next, while on the heating constraint, the first time

derivative of the heating inequality constraint must equal zero:

$$Q^{(1)}(\mathbf{x}, t) = \beta r_0 \gamma + \frac{3B}{E^*} e^{-\beta r_0 h}(1 + c_\ell^2) = 0 \qquad (43)$$

Once again, contact with a constraint does not dictate remaining on the constraint. A boundary arc is only traversed if the optimal control would result in violation of the constraint. This will be analyzed in the following Analysis chapter.

3.4.11 CAV Summary. The CAV model increases the model complexity with additional degrees-of-freedom; namely, altitude and flight path angle. The variable altitude leads to a stagnation heating constraint. While the waypoint and no-fly zone constraints have the same form, their analytical solutions are different than the HCV. The differences arise from the derived constraints being functions of the state derivatives; therefore, since the equations of motion have changed, the derived constraint equations change. This will be covered further in the Analysis chapter to follow.

3.5 Numerical Methods Implementation Techniques

The next chapter will show that the simpler HCV problem has an analytical solution; contrarily, the complexity of the CAV requires numerical techniques to efficiently compute the solution. A numerical technique typically represent the continuous problem via some form of discretization. The discretization method, i.e. the spacing of the discretized points, is going to be based on the dynamics. Thus, the discretization may need to be different for different portions of the modeled dynamics. For example, Runge-Kutta integration is often run with variable step size to allow for an increase in the number of time steps during rapid state changes. This methodology remains true when attempting to model the original states plus the added dimensionality of the costates; therefore, an increased number of time steps is required for rapid changes in the states as well as rapid changes in the costates. Section 3.1.2 discusses discontinuous Lagrange multipliers, i.e. jumps in the costates.

Thus, in order to properly model the system, any numerical technique must be able to model these discontinuities accurately. This modeling can be achieved by creating a dense set of time steps, or by actually allowing for a discontinuity. The dense set of time steps is used when the numerical method forces continuity, thus the dense set of continuous points replicates a discontinuity. Alternatively, a model can allow for discontinuities for certain parameters while maintaining continuity for others. For example, the controls or costates may have permissible discontinuities whereas the states may be forced to be continuous. These difference in discretization are characteristic of the software packages DIDO and GPOCS. DIDO increases the density of the time steps, called nodes, by specifying intermediate "knots", while GPOCS allows the user to break the problem up into phases. The phased approach of GPOCS allows for a multi-point boundary value problem (MPBVP) by specifying terminal constraints at the end of each phase, i.e. interior-point constraints. Additionally, each phase is discretized individually; therefore, the Chebyshev spacing described in Appendix E creates the desired situation of higher node density at the beginning and end of each phase. The choice of phase breakpoints dictates where desired higher accuracy occurs or where discontinues are allowed. This Chebyshev spacing can have the adverse affect of leaving sparse spacing in the middle of the phase, even with an increase in the specified number of nodes. If this middle region requires higher accuracy, it may be necessary to force another phase change in this region. In summary, the CAV numerical trajectory solution is broken up into phases for three purposes; to specify interior-point constraints, to allow for discontinuities, and/or to increase the accuracy at a region of high state or costate change.

One of the challenges of a phased trajectory solution or approach is determining the required number of phases. To maximize accuracy, every jump in the costates should occur at a phase breakpoint in order to allow for such discontinuities. The number of jumps is dictated by the number of times the path constraints are contacted; unfortunately, this number is typically part of the solution and is not known a priori. It is possible to run a lower number of phases, then add phase breaks at each path

constraint contact; however, it may also be possible to allow the numerical solution to model these discontinuities with continuous costates, i.e. without a phase break. For simplified end user implementation, it is therefore desirable to minimize the number of phases, i.e. eliminate the need to add more breakpoints. The ability to still converge on the optimal solution even with a minimal number of phases is therefore one of the topics of this research. Section 4.10 provides a comparison between a trajectory broken up into a higher number of phases, i.e. increased accuracy, and a trajectory broken up into a minimal number of phases, i.e. more desirable user implementation.

3.6   Solution Comments

This section provided vehicle and mission descriptions for both the HCV and the CAV. It also outlined some of the required analytical and numerical techniques to finally compute a solution for the optimal reentry trajectory. The following chapter provides the detailed analysis and numerical results for both vehicles and their respective missions. The results are not merely presented, but are broken down to ensure they match the expected analytical derivations. User strategies for repeatable solution convergence are also described. The outcome is a verified optimal solution with detailed user implementation techniques.

IV.   Analysis and Results

The culmination of this research effort lies in the analysis and results presented in this chapter. Chapter III developed the fundamental theory; however, the problem specific solutions remain to be solved. For each vehicle and mission, the goal is to *analytically* compute the optimal solution to the maximum extent possible. For the two-dimensional constant speed HCV, a complete geometric solution is found requiring no integration. When deceleration is added to the HCV problem, a solution is determined using an indirect method by satisfying an analytically derived set of optimality criteria [48]. Lastly, a proposed direct numerical solution technique is used to solve the optimal three-dimensional CAV reentry trajectory problem. A two-step process is implemented to validate these numerical results. First, the solution to the HCV problem is recomputed using this new numerical solution technique and the results are compared to those previously computed via the geometric and analytical approach. Second, the candidate numerical result is tested to verify it matches the derived analytical form of the solution. Once this exhaustive process of verifying the optimality of the results is complete, a minimal set of user representative steps is outlined for future implementation. This will fulfil the research objective of determining a solution process capable of rapidly converging on the optimal reentry trajectory solution that satisfies the vehicle dynamic, heating limitation, and the added military constraints of waypoints and no-fly zones; all within the confines of the available control authority.

4.1   2-D Baseline

In order to access the improvement achievable by optimizing a trajectory, a non-optimal baseline trajectory is computed for comparison. This baseline trajectory requires very little computation since it's a simple steer and point approach to navigation. The baseline controller is the optimal bang-level-bang controller for each segment separately [23; 26; 88]; however, this segment by segment technique does not ensure optimality over the entire trajectory. Specifically, due to the non-optimal

incoming heading to each waypoint, it is likely to overshoot thus extending the trajectory. The amount of time to be gained by optimizing over this baseline is proportional to the amount of time the vehicle is in a turn compared to the total range. For a subsonic aircraft, with relatively small turn radii, the ratio of time within a turn is low and this optimization may produce a negligible time savings. However, the HCV has huge turn radii in comparison to its designed range and thus the time savings can be significant. This minimum time solution will likely also benefit a fighter aircraft in confined terrain or a reconnaissance UAV with a small maximum bank angle. The initial point, waypoints, no-fly zone, and target from Table 1 are shown in Figure 13. This figure also shows the baseline trajectory for both the constant speed case and decelerating flight. In either baseline case, the control law is to make a maximum bank turn toward the next waypoint or no-fly zone contact, level out, and then fly straight to the next destination. This process continues until target intercept. Later comparisons will illustrate the inefficient overshoots mentioned earlier.
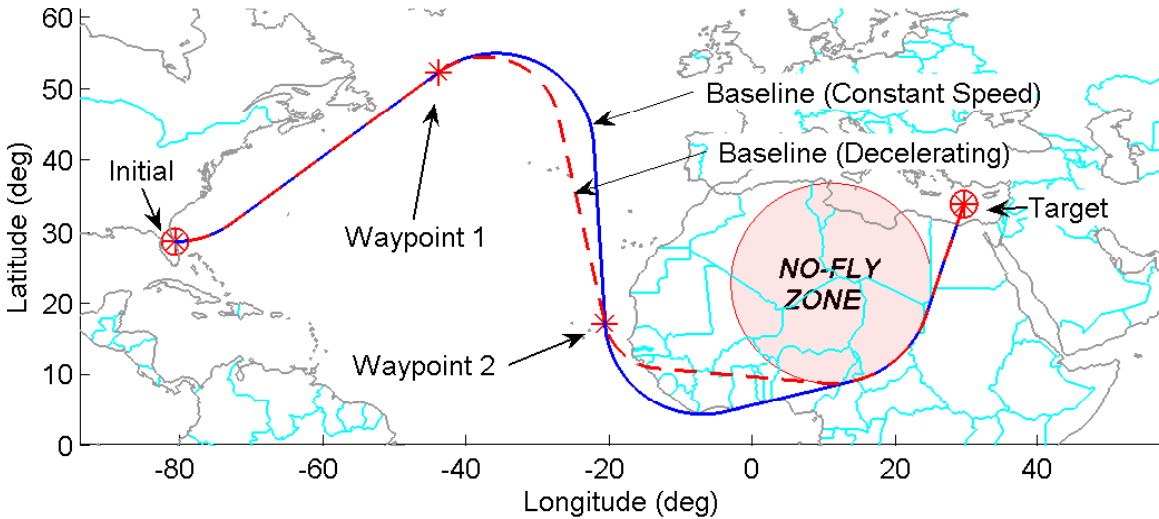
Figure 13:   Baseline Trajectories: Constant Speed and Decelerating Flight

4.2   2-D Geometric

A geometric approach is taken for the special case of the HCV at constant speed. A geometric technique is applicable since the vehicle traverses a constant turn radius

while maintaining a constant bank angle. A bang-level-bang controller [23; 26; 88] is optimal, but *when* to turn remains to be optimized. Since a bang-level-bang trajectory consist only of straight legs and turns, waypoint passage must occur either during level flight or somewhere in a turn. If the waypoint passage occurs along a straightaway, there's no time of passage to optimize; therefore, the point of waypoint passage along a turn is what is to be optimized.

Figure 14 shows the simplest waypoint intercept case involving three non-collinear points; two endpoints, P_1 and P_2, and one intermediate waypoint, W_P. This figure depicts several pilot techniques for flying from P_1, directly over the waypoint W_p, and then onto P_2. To fly the profile in Figure 14(a), the pilot aims directly for the waypoint and upon waypoint passage initiates a maximum bank turn towards endpoint P_2. This is the baseline technique from Section 4.1. For Figures 14(b) and 14(c), the pilot initiates a heading to the left of W_p, then upon approaching W_P rolls into a maximum bank turn which is timed to ensure passage over the waypoint at some fraction along the turn, and finally rolls out once aligned with endpoint P_2. The last case in Figure 14(d), where the pilot initiates a heading even further to the left of W_P, commences a maximum bank turn such that waypoint passage occurs exactly at the roll out to endpoint P_2. From these examples, the waypoint passage can occur at the very beginning of the turn, at some fraction along the turn, or at the very end of the turn. The fraction of turn achieved before waypoint passage is dictated by the orientation of the turn radius with respect to the waypoint, as controlled by angle χ in Figure 14; therefore, the geometric solution aims to optimize angle χ.

For this analysis a constant speed is assumed; therefore, minimizing the flight path will minimize the flight time. The cost becomes the flight path length given by:

$$J = d_1(\chi) + R\,\Delta\theta(\chi) + d_2(\chi) \tag{44}$$

Here R is the specified turn radius and the values for d_1, d_2, and $\Delta\theta$ are all geometrically a function of angle χ. The turn radius is rotated about the waypoint, thus

(a) Baseline, Waypoint on Incoming Tangent

(b) $\chi > 0$, Intermediate Value

(c) $\chi = 0$, Optimal Orientation

(d) $\chi < 0$, Waypoint on Outgoing Tangent
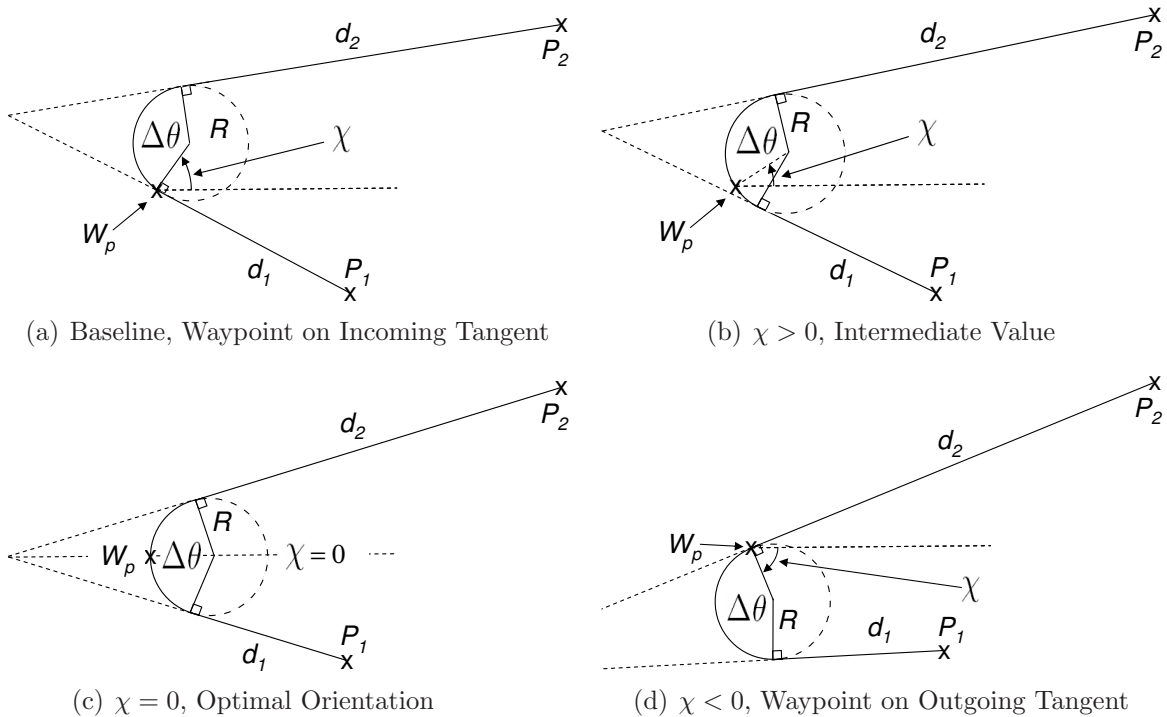
Figure 14: Turn Radius Orientation Rotated about Intermediate Waypoint (W_p) by Angle χ. Radius (R) is constant. Endpoints (P_1 and P_2), and Intermediate Waypoint (W_p) are fixed.

ensuring passage. Also, the initial orientation ($\chi = 0$) is such that the intersection of the tangent lines from P_1 and P_2 is collinear with the waypoint and the center of the turn circle as shown in Figure 14(c). The angle χ is anchored at the waypoint and is measured counter-clockwise to the center of the turn circle as shown in Figure 14. A minimum of the cost J with respect to χ may exist when:

$$\partial J / \partial \chi = 0 \tag{45}$$

Using purely geometric relations, the solution to Equation (45) can be shown to exist at $\chi = 0$, for any values of the initial setup for P_1, P_2, W_p, and R. This orientation implies that waypoint passage occurs at the halfway point along the turn, i.e. the midpoint of the turn. The geometric results are shown in Figure 15. As expected, graphically the optimal geometric solution does appear shorter than the baseline trajectory; the supporting numerical comparison is presented in Section 4.4.

47
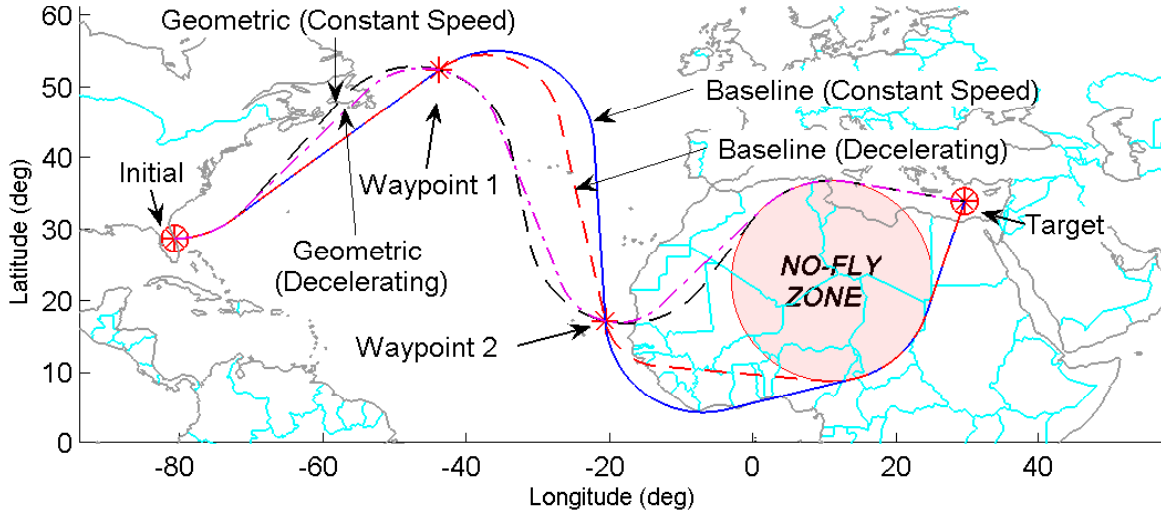
Figure 15:   Baseline and Geometric Trajectories: Constant Speed and Decelerating

The significance of this result is that the minimum time trajectory can be ana-
lytically computed for any given initial, final, intermediate waypoint, and turn radius.
Furthermore, at a constant velocity, the equations of motion do not need to be inte-
grated, so long as a relationship between maximum bank angle and minimum turn
radius exist. For an air vehicle to produce enough lift in a banked turn to maintain
level flight, the centripetal acceleration a_c is $Lift \sin \sigma / m$ where $Lift$ must be $mg/ \cos \sigma$.
Thus for a non-skidding turn:

$$a_c = g \tan \sigma = \frac{V^2}{R} \tag{46}$$

$$\text{thus} \quad R = \frac{V^2}{g \tan \sigma} = \text{Minimum Turn Radius} \tag{47}$$

In this geometric approach, the no-fly zones can be treated just like a minimum turn
radius. If the no-fly zone radius is greater than the minimum turn radius, a tangent
to the no-fly zone is flown. If the no-fly zone is smaller than the minimum turn radius,
the no-fly zone is only contacted in one place since flight along the boundary cannot
be maintained.

To support the claim of midpoint optimality, the optimal configuration shown
in Figure 14(c) can also be viewed geometrically as a circle inscribed within a vertex

48

of a triangle. The center of an inscribed circle bisects the angle of each vertex [30]; therefore, the line from the vertex to the center of the circle is the angle bisector. This also implies, with respect to the center of the circle, that the angle from one tangent to the angle bisector is equal to the angle from the other tangent to the angle bisector. Thus, since the optimal waypoint in Figure 14(c) lies on the angle bisector, the waypoint occurs halfway along the arc from the incoming tangent to the outgoing tangent. The optimality of this halfway point is proven in Appendix C.

For an application with multiple waypoints and no-fly zones, a simple iterative technique is used to compute the optimal geometry. Initially select a value for χ, then compute the point of intersection of the two tangent lines as outlined in Appendix D. This provides a measure as to whether the intersection point is collinear with the waypoint and center of the turn circle. This is an iterative process since a tangent is made from the previous and following turn radii or no-fly zones. Since the intermediate waypoint turn radius orientation will change the points of tangency, the process continues until no more changes are required. Since no integration is required for the constant speed case, each iteration is fast, and convergence requires only a few iterations.

4.3   2-D Analytical, Bryson's Method

The next solution method is an analytical approach using the dynamic optimization problem defined in Chapter III. This section references Bryson's method because in Section 4.5 it will be seen that the purely numerical results produce different costate histories than those derived here. Studying Bryson's method provides insight into why the solution must hold certain properties, this intuition is lost when the numerical method in Section 4.5 simply produces a solution. The Breakwell problem, presented by Bryson [14:pg.120] and Ross et al. [78] illustrates the differences on a simpler problem than the research herein. Reference [78] mentions the alternate form of the Lagrangian [40], but does not provide an explanation. Section 4.6 explains the fundamental difference in problem statement and compares the different methods.

49

Appendix F specifically addresses the Breakwell problem. It will be shown in Section 4.6 that Bryson's method and the alternate Covector/Costate method are both correct; however, the differences must be understood in order to verify the numerical results. Bryson's method is presented first because it provides more insight into the expected solution.

The previous chapter provides the general framework for the dynamic optimization solution for the problem presented herein. In order to solve analytically, additional steps are required. The vectors ν, μ, π_n, and π_m are required in order to determine the optimal control and to propagate the costates using Equation (12).

4.3.1 Control.   The first task is to determine an optimal control u. The Hamiltonian in Equation (9) can be multiplied out to show that there is only one term that is a function of u when not on a no-fly zone, i.e. when $\mu = 0$.

$$H = \lambda_x V \cos\theta + \lambda_y V \sin\theta + \lambda_\theta \frac{\tan\sigma_{max}}{V} u + \lambda_V a \tag{48}$$

Pontryagin minimum principle yields that u must minimize H [2]. Furthermore, since the velocity V and the normalizing parameter $\tan\sigma_{max}$ are always positive, u must be the maximum absolute value and opposite sign of λ_θ to minimize H. This defines λ_θ as the switching function for u in a bang-level-bang controller [18]. The optimal control specified below is only when the vehicle is not on the no-fly zone constraint.

$$u^*(t) = \begin{cases} = & 1, & \lambda_\theta < 0 \\ = & 0, & \lambda_\theta = 0 \\ = & -1, & \lambda_\theta > 0 \end{cases} \quad \boxed{\begin{array}{l} \text{This } u^* \text{ is only applicable when} \\ \text{off the no-fly zone constraints} \end{array}} \tag{49}$$

To determine the control while on a no-fly zone constraint, first assume that the no-fly zone is larger than the minimum turn radius. This is not a limitation, it just implies

50

that if the no-fly zone is smaller than the minimum turn radius, then the optimal control would still be Equation (49). The tangent entry requirement is specified as constraint M in Equation (28), i.e. $S = 0$ and $S^{(1)} = 0$. To remain on the no-fly zone constraint, the $S^{(2)} = 0$ condition must remain satisfied. For convenience, Equation (26) is repeated here as Equation (50). Starting with Equation (50), the chain rule and state derivatives in Equation (22) are used to compute the time derivatives of S. Setting Equation (52) equal to zero, the optimal control, u^*, along a no-fly zone boundary is computed in Equation (53).

$$S = \frac{1}{2}\left(R_j^2 - \Delta x_j^2 - \Delta y_j^2\right) \tag{50}$$

$$S^{(1)} = -\Delta x_j V \cos\theta - \Delta y_j V \sin\theta \tag{51}$$

$$S^{(2)} = -V^2 + u(\Delta x_j \sin\theta - \Delta y_j \cos\theta)\tan\sigma_{max}$$
$$+ (-\Delta x_j \cos\theta - \Delta y_j \sin\theta)a = 0 \tag{52}$$

$$u^*(t) = \frac{V^2}{(\Delta x_j \sin\theta - \Delta y_j \cos\theta)\tan\sigma_{max}} \boxed{\begin{array}{l}\text{This } u^* \text{ is only applicable while} \\ \text{on a no-fly zone constraint}\end{array}} \tag{53}$$

The term in front of a in Equation (52) is zero, since it is a multiple of Equation (51), which is identically zero when tangent to the no-fly zone boundary. It is not eliminated from Equation (52) since the partial derivative of $S^{(2)}$ with respect to the state vector is still required to propagate the costates.

4.3.2 Costate Propagation. The derivative of the costates must be determined in order to propagate the costates backwards in time. The costate time history is used to compute the optimal control time history. Substituting the partial deriva-

tive of Equations (23) and (52) into the transpose of Equation (12) produces:

$$\dot{\lambda} = \left[-\lambda^T f_{\mathbf{x}}\right]^T - \mu \left[S_{\mathbf{x}}^{(2)}\right]^T \tag{54}$$

$$\begin{bmatrix} \dot{\lambda}_x \\ \dot{\lambda}_y \\ \dot{\lambda}_\theta \\ \dot{\lambda}_V \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \lambda_x V \sin\theta - \lambda_y V \cos\theta \\ -\lambda_x \cos\theta - \lambda_y \sin\theta + \lambda_\theta \dfrac{\tan\sigma_{max}}{V^2} u \end{bmatrix}$$

$$- \mu \begin{bmatrix} u \sin\theta \tan\sigma_{max} - a\cos\theta \\ -u\cos\theta \tan\sigma_{max} - a\sin\theta \\ (\Delta x_j \sin\theta - \Delta y_j \cos\theta)\, a \\ -2V \end{bmatrix} \tag{55}$$

If not on a no-fly zone boundary, then $\mu = 0$ and only the first term in Equation (55) needs to be propagated. However, if on a no-fly zone constraint, then $\mu \neq 0$ and its value must be computed. Solving for $H_u = 0$ as the partial derivative of H in Equation (19) with respect to u yields:

$$\mu = \frac{-\lambda_\theta}{V(\Delta x_j \sin\theta - \Delta y_j \cos\theta)} \tag{56}$$

4.3.3 Final Costates. The costates are initialized at the final time, t_f, to begin the backwards in time integration. The vector ν contains two elements, ν_x and ν_y, since there are two elements of the final constraint ψ. From the final costate equation, Equation (13), with $\phi_{\mathbf{x}} = 0$ and ψ only a function of x and y, the final costates can simply be defined as:

$$\begin{bmatrix} \lambda_x(t_f) & \lambda_y(t_f) & \lambda_\theta(t_f) & \lambda_V(t_f) \end{bmatrix} = \begin{bmatrix} \nu_x & \nu_y & 0 & 0 \end{bmatrix} \tag{57}$$

If it is assumed that at target intercept the vehicle is not in a turn, then $u(t_f) = 0$, $\lambda_\theta(t_f) = 0$, and $\dot{\lambda}_\theta(t_f) = 0$. Substitution of ν_x and ν_y into the transversality condition,

Equation (15), and $\dot{\lambda}_\theta(t_f) = 0$ into Equation (55), leads to:

$$\lambda_x(t_f) \quad = \quad \nu_x \quad = \quad \frac{-\cos\theta_f}{V_f} \tag{58a}$$

$$\lambda_y(t_f) \quad = \quad \nu_y \quad = \quad \frac{-\sin\theta_f}{V_f} \tag{58b}$$

4.3.4 Jump Conditions.   Next, the boundary conditions at the end of each segment must be satisfied. To satisfy the intermediate boundary conditions N and M, there is a jump in the Hamiltonian and the costates at times t_i and t_j. The jumps occur at each waypoint passage, and entry onto the tangent of each no-fly zone constraint.

The continuity of the Hamiltonian is determined by the jump criteria in Equation (17b), the interior-point constraint N for the waypoints, and the interior-point constraint M for the no-fly zones. The following is derived from the definition of N in Equation (27) and from the definition of M in Equation (28):

$$\partial N/\partial t_i = 0 \tag{59a}$$

$$\partial M/\partial t_j = 0 \tag{59b}$$

Since the jump in the Hamiltonian is zero, the Hamiltonian is continuous across the interior-point constraints:

$$H(t_i^-) \quad = \quad H(t_i^+) = H(t_i) \tag{60a}$$

$$H(t_j^-) \quad = \quad H(t_j^+) = H(t_j) \tag{60b}$$

Also, since the Hamiltonian is not explicitly a function of time it must be a constant; therefore, solving the transversality condition in Equation (15) and substituting $\lambda(t_f)$ from Equation (13) into H at t_f in Equation (9), the constant for H is completely

determined as:

$$H(t_f) = H(t) = H(t_i) = H(t_j) = -1 \tag{61}$$

The costate discontinuities at each waypoint are determined from N and the jump criteria in Equation (17a). The control is continuous if waypoint passage at t_i occurs along the turn, i.e. $u(t_i^-) = u(t_i^+)$. Since the assumption is that the waypoints are sufficiently spaced, the control is zero just before starting the turn at time t_{si}, $u(t_{si}^-) = 0$, and non-zero immediately upon commencing the turn, $u(t_{si}^+) \neq 0$. Similarly for the no-fly zone, the control is zero immediately before entering the no-fly zone boundary, $u(t_j^-) = 0$, and non-zero immediately after contact, $u(t_j^+) \neq 0$. From the waypoint optimal control, $u^*(t)$, defined in Equation (49), any time the control is zero the switching costate (λ_θ) is zero. As derived from Equation (17), the costate jump at each waypoint is given in terms of the unknown constants, π_{nx} and π_{ny}:

$$\lambda(t_i^-)^T = \lambda(t_i^+)^T + [\ \pi_{nx} \quad \pi_{ny} \quad 0 \quad 0\ ] \tag{62}$$

The costate propagation in Equation (55) shows $\dot{\lambda}_x = \dot{\lambda}_y = 0$; therefore, the costates λ_x and λ_y do not change from the start of the waypoint turn at time t_{si}, to immediately before waypoint passage t_i (referencing forward in time):

$$\lambda_x(t_{si}) = \lambda_x(t_i^-) = \lambda_x(t_i^+) + \pi_{nx} \tag{63a}$$

$$\lambda_y(t_{si}) = \lambda_y(t_i^-) = \lambda_y(t_i^+) + \pi_{ny} \tag{63b}$$

Using Equation (63) and the continuity of the Hamiltonian, there exists a relationship between heading at the start of the turn, θ_{si}, and the heading at the jump, θ_i. Using Equations (55) and (63) with $\dot{\lambda}_\theta(t_{si}) = 0$ and $\lambda(t_i^+)$ obtained by integrating backwards

in time from t_f, the derived constants are:

$$
\pi_{nx} = \frac{-\sin\theta_i \left(\lambda_x(t_i^+)\sin\theta_{si} - \lambda_y(t_i^+)\cos\theta_{si} \right)}{\cos(\theta_i - \theta_{si})}
$$
$$
+ \frac{\cos\theta_{si}\lambda_\theta(t_i^+)\tan\sigma_{max}\left(u(t_i^+) - u(t_i^-)\right)}{V_{si}^2} \tag{64a}
$$

$$
\pi_{ny} = \frac{\cos\theta_i \left(\lambda_x(t_i^+)\sin\theta_{si} - \lambda_y(t_i^+)\cos\theta_{si} \right)}{\cos(\theta_i - \theta_{si})}
$$
$$
+ \frac{\sin\theta_{si}\lambda_\theta(t_i^+)\tan\sigma_{max}\left(u(t_i^+) - u(t_i^-)\right)}{V_{si}^2} \tag{64b}
$$

When waypoint passage occurs along the turn $u(t_i^-) = u(t_i^+)$ and only the first terms in Equation (64) are non-zero. Note that the baseline trajectory defined in Section 4.1 is a special case, where the passage of each waypoint occurs at the start of the turn, i.e. $t_{si} = t_i$ and $\theta_{si} = \theta_i$, but the turn just commenced so $u(t_i^-) \neq u(t_i^+)$; thus, the terms at the end of Equation (64) are non-zero. Recall the baseline is a simple non-optimized steer and point control, so dynamic optimization is not required; therefore, this special case is only included for completeness to allow the Hamiltonian to be computed for the baseline to see how it compares to the optimal Hamiltonian.

For the no-fly zone constraint, the components of M in Equation (28) must be satisfied:

$$
M = \begin{bmatrix} S \\ S^{(1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\left(R_j^2 - \Delta x_j^2 - \Delta y_j^2\right) \\ -\Delta x_j V \cos\theta - \Delta y_j V \sin\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{65}
$$

The control is zero approaching the no-fly zone, which implies λ_θ has been zero, thus the entry has the property that $\dot{\lambda}_\theta(t_j^-) = 0$. Also, like the waypoint jump, the Hamiltonian is constant $H(t_j^-) = H(t_j^+)$. Taking the partial derivatives of M in Equation (28) and solving for the constants using $\lambda_\theta(t_j^-) = 0$, $\dot{\lambda}_\theta(t_j^-) = 0$, and

$H(t_j^-) = H(t_j^+)$ leads to the following:

$$\Delta x_j \;=\; x(t_j) - x_{cj} \quad , \quad \Delta y_j = y(t_j) - y_{cj} \tag{66}$$

$$\begin{bmatrix} \lambda_x(t_j^-) \\ \lambda_y(t_j^-) \\ \lambda_\theta(t_j^-) \\ \lambda_V(t_j^-) \end{bmatrix} = \begin{bmatrix} \lambda_x(t_j^+) - \pi_{m0}\Delta x_j - \pi_{m1}V_j\cos\theta_j \\ \lambda_y(t_j^+) - \pi_{m0}\Delta y_j - \pi_{m1}V_j\sin\theta_j \\ \lambda_\theta(t_j^+) + \pi_{m1}\left(\Delta x_j \sin\theta_j - \Delta y_j \cos\theta_j\right)V_j \\ \lambda_V(t_j^+) \end{bmatrix} \tag{67}$$

$$\pi_{m0} \;=\; \frac{\lambda_x(t_j^+)\sin\theta_j - \lambda_y(t_j^+)\cos\theta_j}{\Delta x_j \sin\theta_j - \Delta y_j \cos\theta_j} \tag{68}$$

$$\pi_{m1} \;=\; \frac{\lambda_\theta(t_j^+)u(t_j^+)\tan\sigma_{max}}{V_j^3} = \frac{-\lambda_\theta(t_j^+)}{V_j\left(\Delta x_j \sin\theta_j - \Delta y_j \cos\theta_j\right)} \tag{69}$$

Since the states at time t_j are continuous, the plus and minus superscripts are unnecessary. The control $u(t_j^+)$ in Equation (69) is replaced with the u required to maintain $S^{(2)} = 0$ from Equation (53). Within the derivation of Equation (69), additional terms appear that are multiples of $S^{(1)}$; however, since $S^{(1)}=0$ these terms are not shown for simplification.

Having now defined all the criteria for the optimal trajectory, the next step is to connect all the solution pieces. The geometric solution provides an initial guess of the ten solution times; the waypoint passage times (2), the no-fly zone entry time (1), the final time (1), the end of the initial turn toward the first waypoint (1), the start and end time of each waypoint turn (4), and the no-fly zone exit time (1). These times, and the associated optimal control, are used to propagate the states forward. From the forward integration, the accuracy of hitting the final target, Equation (25); passing through each waypoint, Equation (27); and satisfying the tangency conditions, Equation (28), are all used as the measure of solution success. At this point, for the mission as defined in Table 1, the solution has eight solution criteria. The forward states are interpolated during the backward integration for the costates. During the backward integration, the appropriate derivatives and jumps are computed. The accuracy of achieving each $\lambda_\theta(t_{si}^-) = 0$ are additional solution criteria; two for this

56

mission since there are two waypoints. This increases the total number of solution criteria to ten. Thus, a search vector of ten times is used to satisfy ten success criteria, which should each be zero. A root solver is used to manipulate a search vector to drive the solution vector to zeros. For this research, the solution or search times are input into the Matlab® root solver, fsolve, to drive the solution criteria towards zero. A related indication of success (optimality) is to observe the value of the Hamiltonian. During this derivation, $H(t)$ is assumed continuous, but only the optimal solution will yield $H(t) = -1$ from Equation (61) for all t. The results from this process are presented in the next section.

4.4   2-D Comparison, Geometric versus Analytical

This comparison includes the baseline, geometric, and analytical dynamic optimization approach for constant speed and deceleration. It is presented prior to the numerical results because the numerical results converge to the same dynamic optimization solution, just the technique has changed. This section will therefore illustrate the time savings possible using optimal control. Several HCV trajectories, based on the mission described in Table 1, are presented in Figure 16.
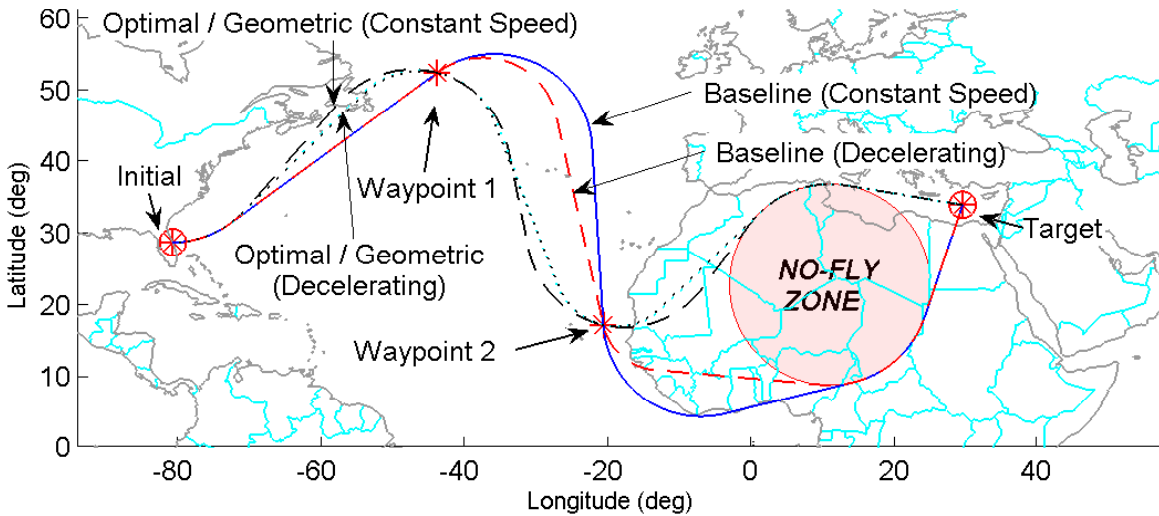
Figure 16:   Baseline, Geometric, and Analytical Dynamic Optimization Trajectories; both Constant Speed and Decelerating

For this problem the initial heading is fixed and the final heading is not constrained; therefore, the optimization is the orientation angle for the turn radii of the first and second intermediate waypoints, and the point entering the no-fly zone boundary. The baseline, geometric, and dynamic optimization numerical results are presented in Table 3. For constant speed, the geometric and dynamic optimization results are identical, as derived in Appendix C. With deceleration, the geometric and dynamic optimization results are nearly the same, and are thus indistinguishable in Figure 16.

The control u is bang-level-bang based on the value of the costate λ_θ. The costates and control are plotted in Figure 17(a). Throughout the rest of this document, the labeling and scaling for some variables may be along the right side of the figure, e.g. λ_x, λ_θ, and u in Figure 17(a). In Figure 17(a) the value of u switches with the sign of λ_θ. The exception to the bang-level-bang control is while the trajectory is on the no-fly zone boundary. The value of u is less than the maximum to avoid violating the constraint; however, optimality is maintained since $H_u = 0$ is maintained by Equation (56).

In Figure 17(b) the effect of a non-optimal solution is demonstrated by observing the value of the Hamiltonian. Here, the optimal solution correctly maintains a Hamiltonian value of -1; however, with deceleration the geometric solution is seen to be non-optimal. Non-optimality is expected with the geometric solution since it forces constant turn radii; however, this no longer represents a maximum bank angle turn while decelerating. The baseline Hamiltonian, not plotted, is even further from optimal reaching a Hamiltonian value of 1.04 (that is 2.04 from the optimal value of -1).
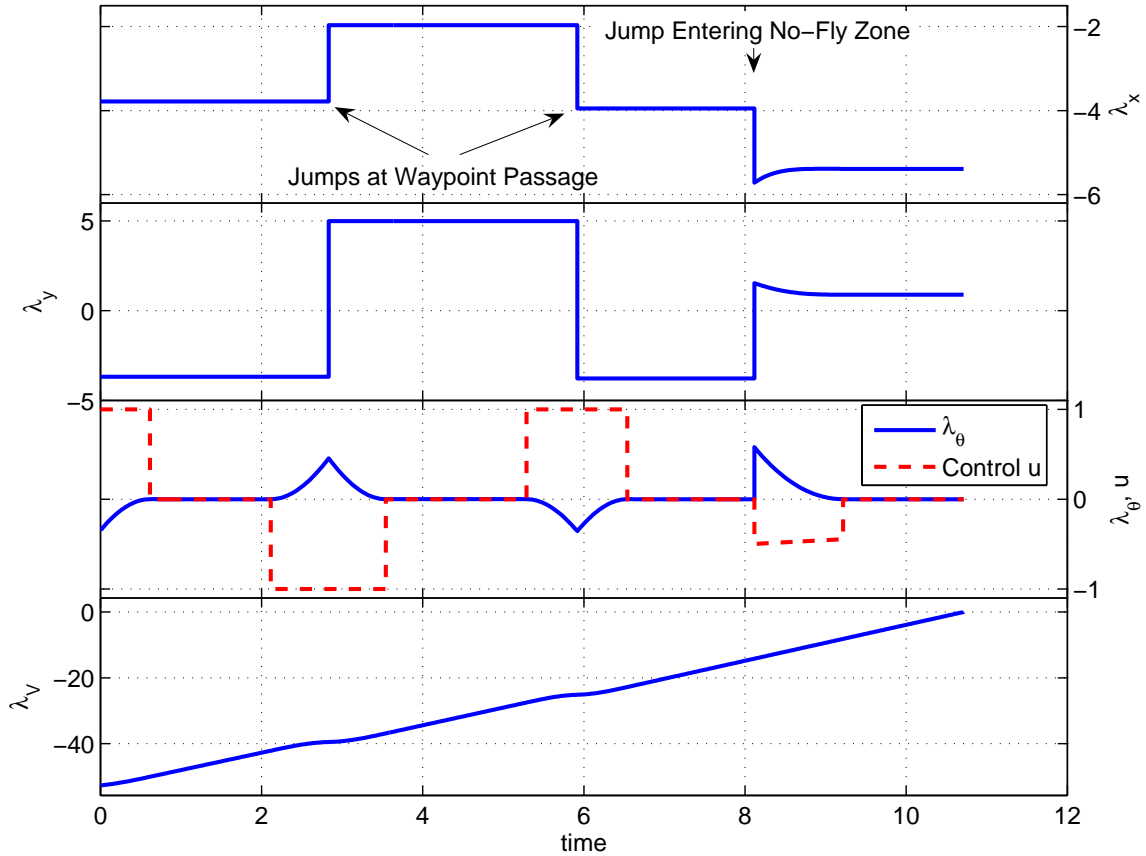
(a) Costates and Control u Time History for Dynamic Optimal Control of Decelerating Flight
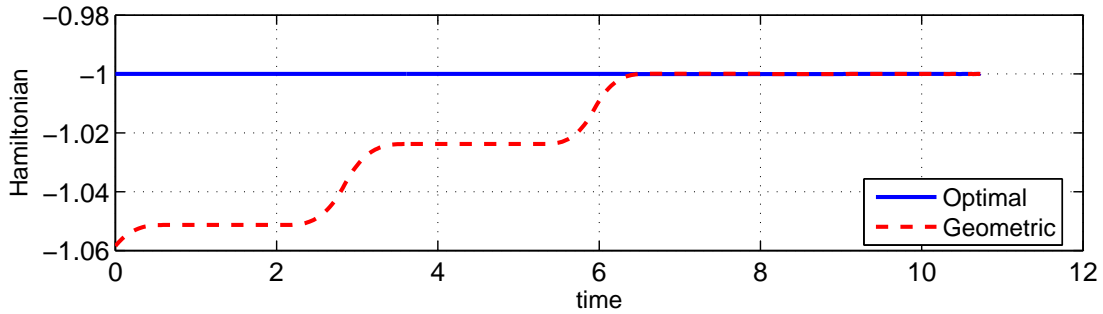
(b) Optimal Hamiltonian $H(t) = -1$, and Non-Optimal $H(t) \neq -1$.

Figure 17:   Nondimensional Costates, Control, and Hamiltonian Time Histories

In Table 3 there are some notable trends beyond just minimizing flight time. For the constant speed baseline case, the vehicle does not meet the design specifications of 9,000 nmi in 2 hrs. So, not only does the baseline take 14% longer, it fails the mission. A mission failure may be characterized by consuming all fuel prior to reaching

$V_{min} = 310$ KEAS, Initial Conditions: $\theta_0 = 0$ deg, $M_0 = 7.66$					
$V_0 = 2.315$ km/s, $r_0 = 6408.48$ km, $g_0 = 9.7169$ m/s^2					
Technique	Decel (m/s^2)	V_f KEAS	M_f	Distance (nmi)	Time (hh:mm:ss)
Baseline	0	531.69	7.66	10285	02:17:08
Geometric	0	531.69	7.66	9000	02:00:00
Optimal	0	531.69	7.66	9000	02:00:00
Baseline[1]	0.1	301.31	4.34	9822	02:47:11
Geometric	0.1	331.24	4.78	8853	02:25:28
Optimal	0.1	331.89	4.78	8831	02:24:59

[1] Invalid solution since minimum airspeed for level flight (V_{min}) is not maintained.

Table 4:   Nondimensional Times for Turns and Target Arrival

Technique	Initial end	Waypoint 1			Waypoint 2			No-Fly Zone		Target arrive
		start	pass	end	start	pass	end	enter	exit	
Baseline	0.507	2.616	2.616	4.339	5.819	5.819	7.228	8.328	9.157	10.132
Geometric	0.688	1.843	2.738	3.633	4.487	5.415	6.344	6.984	7.894	8.866
Optimal	0.688	1.843	2.738	3.633	4.487	5.415	6.344	6.984	7.894	8.866
Baseline	0.502	2.748	2.748	4.149	6.323	6.323	7.099	9.031	10.713	12.352
Geometric	0.641	2.084	2.842	3.624	5.276	5.940	6.623	8.129	9.250	10.747
Optimal	0.615	2.114	2.834	3.542	5.288	5.918	6.539	8.116	9.218	10.712

All trajectories $t_0 = 0$. First three entries are constant speed, last three are decelerating.

the target. A similar time savings is seen with the decelerating case; however, the baseline is unsatisfactory since it has reached an airspeed too slow to sustain level flight ($V_{min} = 310$ KEAS for level flight), thus this trajectory is invalid. Once again, the optimized trajectory has not only saved time, but enabled mission success. Table 4 is provided as a means of verifying duplication of the presented results.

4.5   2-D Numerical

This section is an introduction to the numerical technique required to solve the more complex 3-D CAV problem. These numerical results are computed using the software package GPOCS, and are compared to the analytical results previously computed in Section 4.3.

Figure 18 shows that the numerical results are approximately equal to the analytically derived results. It also illustrates the numerical technique of separate phases, where the waypoints and target are specified as terminal boundary conditions for the appropriate phase.
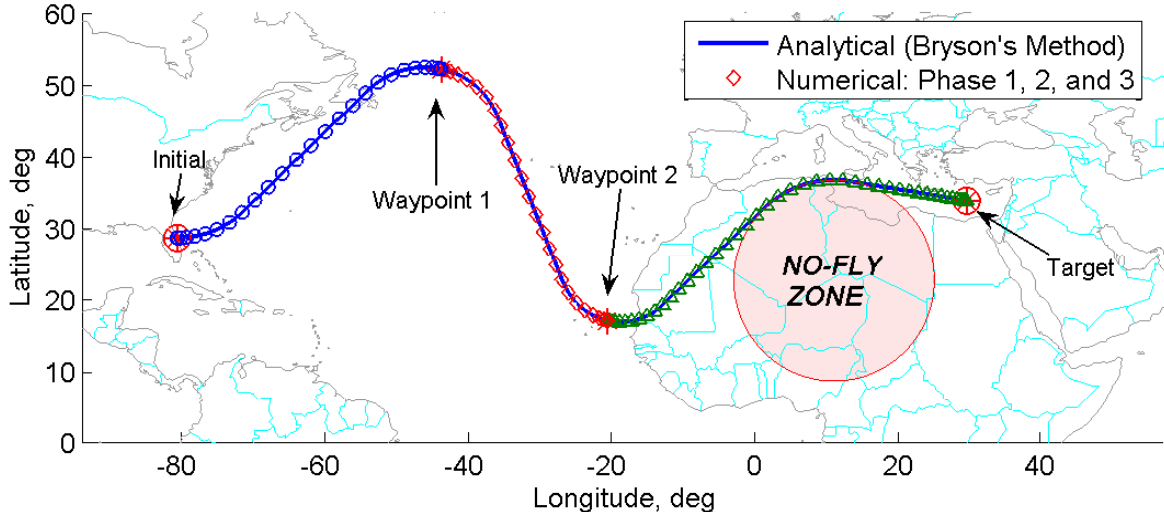
(a) Map
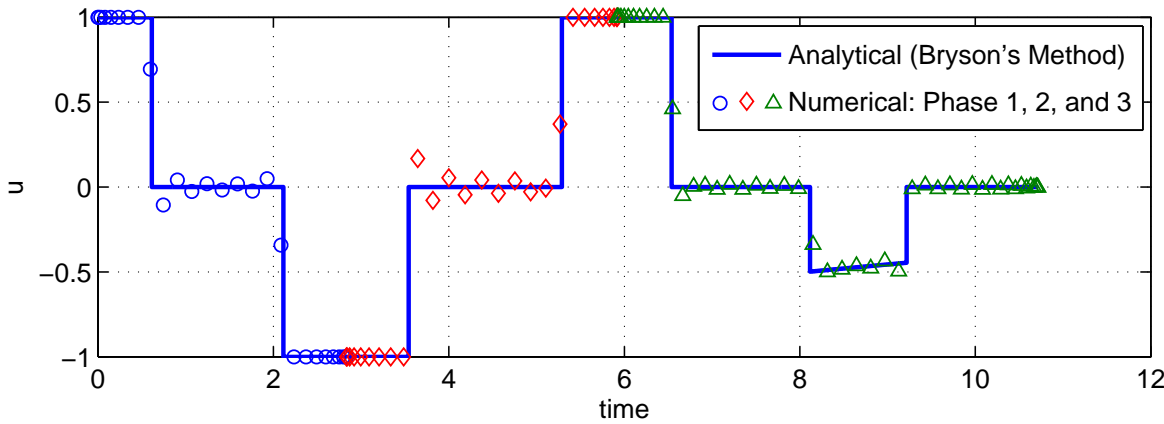
(b) Control

Figure 18:   Analytical (Bryson's method) versus Numerical Results

The costates in Figure 19 match for most of the time history; however, the numerical results have different jump values and derivatives while the trajectory is on the no-fly constraint. In comparison to Bryson's method, the GPOCS software package uses an alternate path constraint method that is compatible with the Gauss pseu-

dospectral costate mapping theorem [4], shown in Figure 11(b). To understand the differences, this alternate method is used to derive another set of analytical equations for the costates; the derivation and comparison is presented next in Section 4.6.
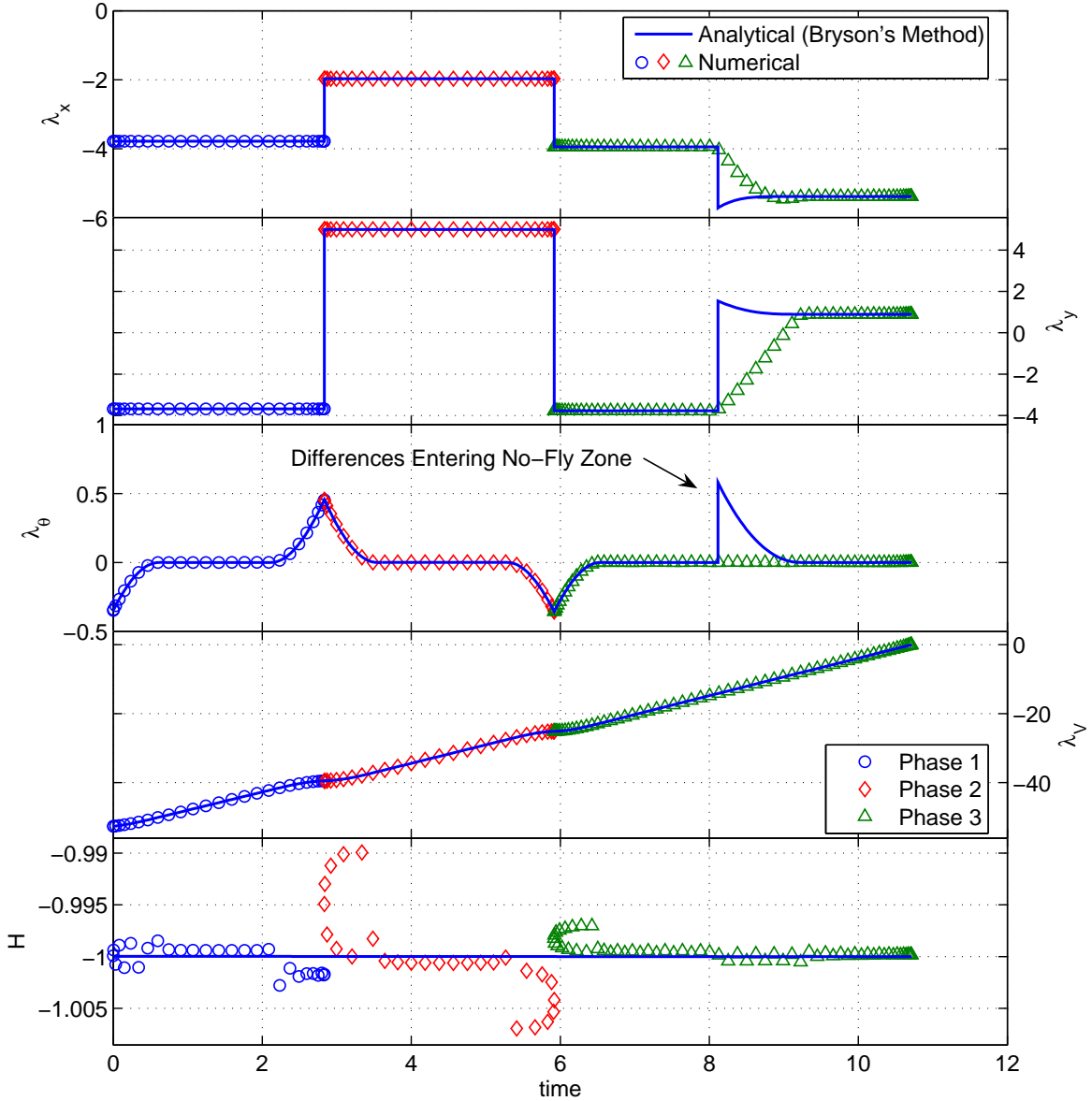
Figure 19:   Costates: Analytical (Bryson's method) versus Numerical Results

4.6   2-D Analytical, Alternate Method

The Gauss pseudospectral costate mapping theorem is used within GPOCS to convert the discrete KKT multipliers into the continuous costates. The user supplied

62

path function provided to GPOCS is simply S, the no-fly zone constraint. Thus S, not $S^{(2)}$, is adjoined to the Hamiltonian:

$$H = L + \lambda^T f + \mu S \tag{70}$$

This change in Hamiltonian implies a change in the costate propagation equations, as compared to Equation (54):

$$
\dot{\lambda} = \left[-\lambda^T f_{\mathbf{x}}\right]^T - \mu \left[S_{\mathbf{x}}\right]^T
$$

$$
\begin{bmatrix} \dot{\lambda}_x \\ \dot{\lambda}_y \\ \dot{\lambda}_\theta \\ \dot{\lambda}_V \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \lambda_x V \sin\theta - \lambda_y V \cos\theta \\ -\lambda_x \cos\theta - \lambda_y \sin\theta + \lambda_\theta \dfrac{\tan\sigma_{max}}{V^2} u \end{bmatrix} - \mu \begin{bmatrix} -\Delta x_j \\ -\Delta y_j \\ 0 \\ 0 \end{bmatrix} \tag{71}
$$

This change in adjoined path constraint has created non-zero derivatives for λ_x and λ_y. Furthermore, the previously derived interior-point constraint, M, is not imposed so there is no jump in the costates. From the new definition of $\dot{\lambda}_\theta$ in Equation (71) there is zero change in its derivative occurring at no-fly zone contact; therefore, without a change in derivative it must remain its previous value of $\dot{\lambda}_\theta = 0$. This criteria leads to the following relationship along the no-fly zone boundary:

$$\lambda_x V \sin\theta - \lambda_y V \cos\theta = 0 \tag{72}$$

Since V is always positive it can be divided out of this equation. The next step is to take the time derivative of both sides of the equation:

$$\sin\theta \dot{\lambda}_x + \lambda_x \cos\theta \, \dot{\theta} - \cos\theta \dot{\lambda}_y + \lambda_y \sin\theta \, \dot{\theta} = 0 \tag{73}$$

Substituting in the values of $\dot{\lambda}_x = \mu \Delta x_j$ and $\dot{\lambda}_y = \mu \Delta y_j$, from Equation (71), leads to an expression for μ:

$$\mu = \frac{-\left(\lambda_x \cos\theta + \lambda_y \sin\theta\right) V}{\left(\Delta x_j \sin\theta - \Delta y_j \cos\theta\right)^2} \tag{74}$$

Upon no-fly zone contact the costates are propagated using Equations (71) and (74) to create the time history presented in Figure 20. The control time history is unchanged since $u^*(t)$ in Equation (53) is not a function of the costates.
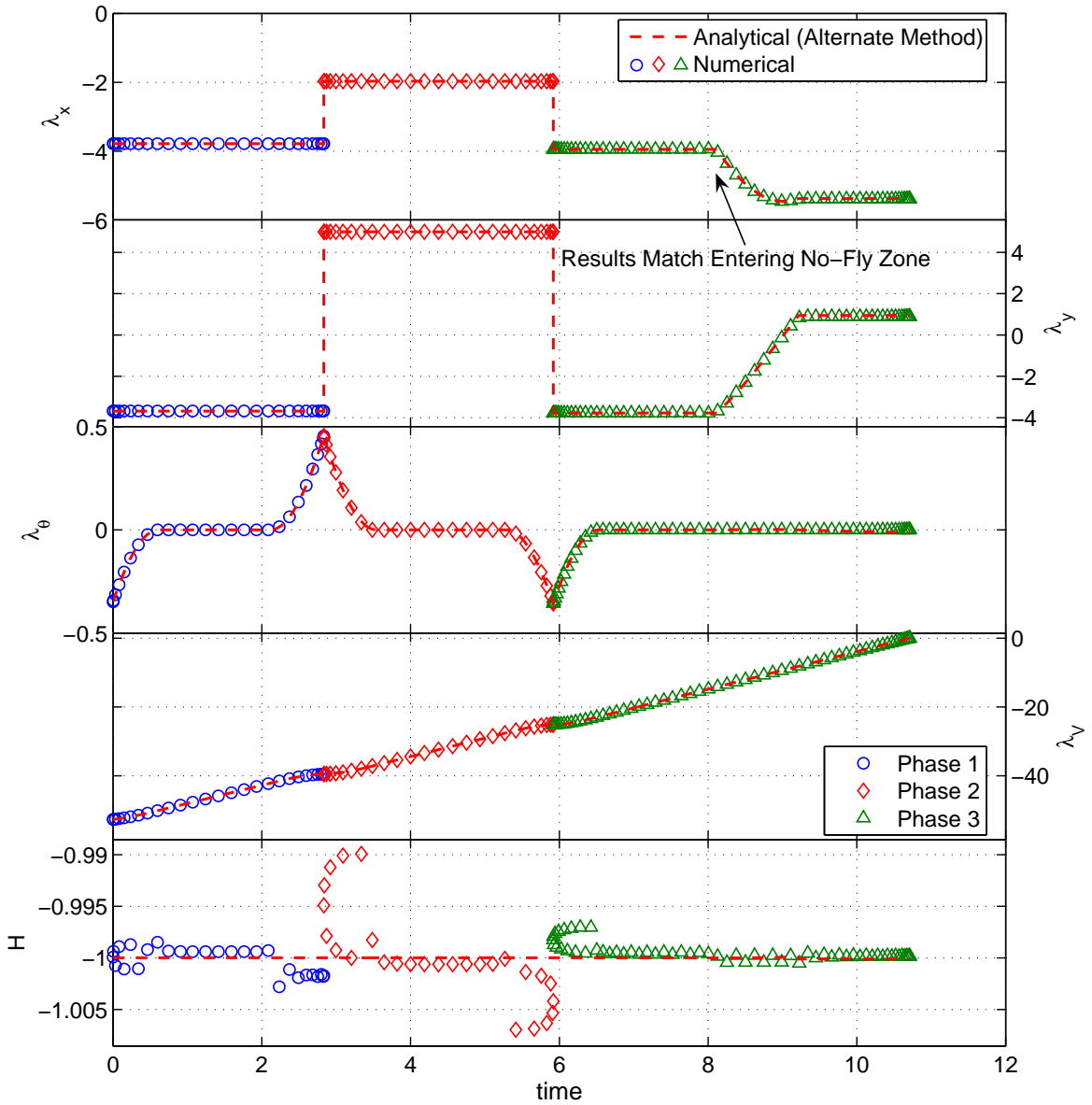
Figure 20: Analytical (Alternate Method) versus Numerical Results

64
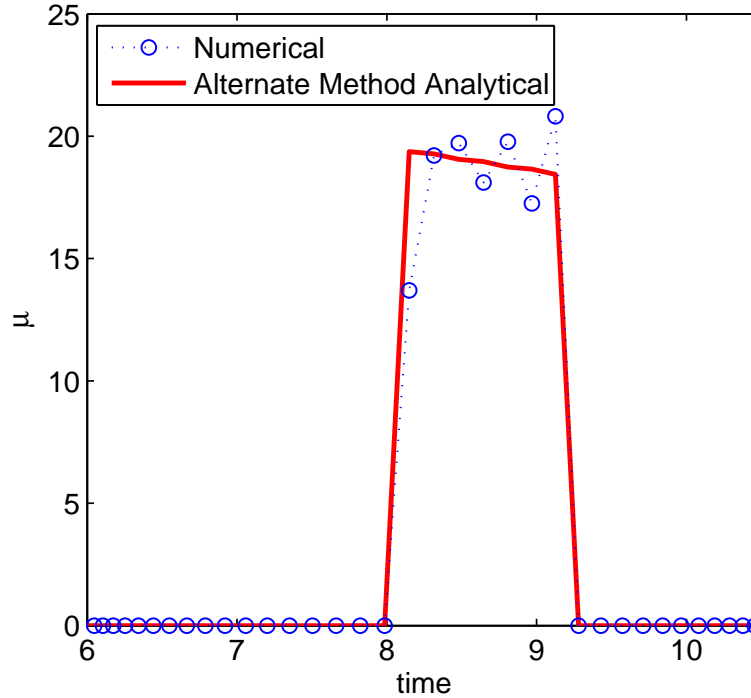
Figure 21:   Numerical and Analytical Path Multiplier, μ

The correlation in the numerical results, as shown in Figure 20, with the alternate derivation of the costate propagation equations clearly shows that numerical computation is conforming to the optimality criteria. This understanding of what path constraint is being adjoined to the Hamiltonian to create the time history of the costates is used in the CAV 3-D analysis. Part of the numerical solution by GPOCS is the value of the path multiplier, μ. The value of μ from Equation (74) is plotted in Figure 21, along with the GPOCS path constraint multiplier output. The GPOCS values have not been altered; however, the computed μ values are set to zero for all times off the no-fly zone boundary constraint. The 3-D model is analyzed next using this understanding gained from the simpler 2-D results.

4.7   3-D Analytical

The analytical solution is derived first to determine the form of the optimal control. The first step is to derive the solution with no path constraints. Then, with

path constraints, determine how the propagation of the costates changes with the inclusion of the path multipliers, μ^T.

4.7.1 3-D Optimal Unconstrained Control. First assume the Hamiltonian is only a function of the integrand and the costates, i.e. $\mu = 0$ since the path constraints are assumed inactive:

$$
\begin{aligned}
H & = L + \lambda^T f \\
& = \lambda_x V \cos\theta + \lambda_y V \sin\theta + \lambda_h V\gamma - \lambda_V \frac{BV^2 e^{-\beta_{r_0}h}(1+c_\ell^2)}{2E^*} \\
& \quad + \lambda_\gamma \left[BVe^{-\beta_{r_0}h}c_\ell \cos\sigma - \frac{1}{V} + V \right] + \lambda_\theta BVe^{-\beta_{r_0}h}c_\ell \sin\sigma \qquad (75)
\end{aligned}
$$

The first-order optimality condition requires $H_{\mathbf{u}} = 0$ and the second-order criteria requires $H_{\mathbf{uu}} > 0$ [12:pg.66; 50], i.e. positive definite. These represent the following matrices:

$$
\mathbf{u} = \begin{bmatrix} \sigma & c_\ell \end{bmatrix}^T \tag{76}
$$

$$
H_{\mathbf{u}} = \begin{bmatrix} H_\sigma & H_{c_\ell} \end{bmatrix} = 0 \tag{77}
$$

$$
H_{\mathbf{uu}} = \begin{bmatrix} H_{\sigma\sigma} & H_{\sigma c_\ell} \\ H_{c_\ell\sigma} & H_{c_\ell c_\ell} \end{bmatrix} > 0 \tag{78}
$$

Starting with Equation (75) the partial derivative of the Hamiltonian with respect to the controls are:

$$
H_\sigma = -\lambda_\gamma BVe^{-\beta_{r_0}h}c_\ell \sin\sigma + \lambda_\theta BVe^{-\beta_{r_0}h}c_\ell \cos\sigma = 0 \tag{79a}
$$

$$
H_{c_\ell} = -\lambda_V \frac{BV^2 e^{-\beta_{r_0}h}}{E^*}c_\ell + \lambda_\gamma BVe^{-\beta_{r_0}h}\cos\sigma + \lambda_\theta BVe^{-\beta_{r_0}h}\sin\sigma = 0 \tag{79b}
$$

66

The second partial derivative is taken to ensure $H_{\mathbf{uu}}$ is positive definite:

$$H_{\mathbf{uu}} = \begin{bmatrix} -c_\ell BV e^{-\beta r_0 h}(\lambda_\gamma \cos\sigma + \lambda_\theta \cos\sigma) & BV e^{-\beta r_0 h}(-\lambda_\gamma \sin\sigma + \lambda_\theta \cos\sigma) \\ BV e^{-\beta r_0 h}(-\lambda_\gamma \sin\sigma + \lambda_\theta \cos\sigma) & -\lambda_V \dfrac{BV^2 e^{-\beta r_0 h}}{E*} \end{bmatrix} > 0 \tag{80}$$

The off diagonals of $H_{\mathbf{uu}}$ are zero since they are simply H_σ/c_ℓ, where $H_\sigma = 0$ from Equation (79a). Thus, the eigenvalues of $H_{\mathbf{uu}}$, which must be > 0 [12:pg.66], are equal to the terms on the main diagonal:

$$-c_\ell BV e^{-\beta r_0 h}(\lambda_\gamma \cos\sigma + \lambda_\theta \cos\sigma) \;>\; 0 \tag{81a}$$

$$-\lambda_V \frac{BV^2 e^{-\beta r_0 h}}{E*} \;>\; 0 \tag{81b}$$

In analyzing Equation (81b); the terms B, V^2, $e^{-\beta r_0 h}$, and E^* are all positive, therefore λ_V must be negative:

$$\lambda_V < 0 \tag{82}$$

In order to satisfy the conditions in Equation (79), two equations must be solved simultaneously:

$$-\lambda_\gamma \sin\sigma + \lambda_\theta \cos\sigma \;=\; 0 \tag{83a}$$

$$\lambda_V c_\ell/E^* + \lambda_\gamma \cos\sigma + \lambda_\theta \sin\sigma \;=\; 0 \tag{83b}$$

To solve Equation (83a), assume $\lambda_\gamma = k\cos\sigma$ and $\lambda_\theta = k\sin\sigma$ where k is a constant to be determined. From the assumed values for λ_γ and λ_θ, and the trigonometric identity $\sin^2\sigma + \cos^2\sigma = 1$ [51:pg.A52], the following must also be true:

$$\lambda_\gamma^2 + \lambda_\theta^2 = k^2 \tag{84}$$

Thus, an intermediate solution is:

$$k = \pm\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \tag{85a}$$

$$\lambda_\gamma = \pm\cos\sigma\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \qquad \text{implies} \qquad \cos\sigma = \pm\lambda_\gamma/\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \tag{85b}$$

$$\lambda_\theta = \pm\sin\sigma\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \qquad \text{implies} \qquad \sin\sigma = \pm\lambda_\theta/\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \tag{85c}$$

The sign of k must still be determined from Equation (81a); enforcing the control limitation $c_\ell \geq 0$, implies $\lambda_\gamma\cos\sigma + \lambda_\theta\cos\sigma < 0$. Using the solution from Equation (85) leads to:

$$(\pm\lambda_\gamma^2 \pm \lambda_\theta^2)/\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} < 0 \tag{86}$$

To satisfy Equation (86) implies k must be negative; therefore, the solution for the control σ is:

$$\cos\sigma = -\lambda_\gamma/\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \tag{87a}$$

$$\sin\sigma = -\lambda_\theta/\sqrt{\lambda_\gamma^2 + \lambda_\theta^2} \tag{87b}$$

$$\sigma = \texttt{atan2}(\sin\sigma, \ \cos\sigma) \tag{87c}$$

Equation (87) is now plugged into Equation (83b) to solve for c_ℓ as:

$$c_\ell = \frac{-E^*\sqrt{\lambda_\gamma^2 + \lambda_\theta^2}}{V\lambda_V} \tag{88}$$

The `atan2` function in Equation (87c) solves for the correct quadrant for the angle σ. This may seem unnecessary since σ is confined to $-60° \leq \sigma \leq 60°$; however, the sign of σ is still determined by the unbounded solution. Thus, using this methodology maintains the correct sign for σ, and avoids incorrectly jumping from one bound to the next when the unbounded solution exceeds $\pm 90°$.

The numerically solved values, from the next section, for λ_V, λ_γ, and λ_θ will be input into Equations (87) and (88) to integrate the states forward to verify the numerical results are adhering to the optimality criteria enforced in this derivation.

4.7.2 3-D Optimal Constrained Control.   The CAV model allows for two controls, σ and c_ℓ, which have specified limitations. It will be shown that when one control is restricted to its limit, the other control law must be modified to maintain optimality. In other words, if a control is at its limit, the control laws derived in Equations (87) and (88) may no longer be valid.

The control limitations, Equation (33), are adjoined to the Hamiltonian via the multiplier μ, as defined in Equation (9), repeated here:

$$H = L + \lambda^T f + \mu^T C \tag{89}$$

As an example, assume the bank angle has reached its maximum positive value, thus the bank angle constraint is active. For optimality, H_σ and H_{c_ℓ} must still equal zero; however, since μ is now non-zero the solutions in Equations (87) and (88) are no longer valid. The new equations for H_σ and H_{c_ℓ} are:

$$H_\sigma = -BVe^{-\beta r_0 h}\lambda_\gamma c_\ell \sigma + BVe^{-\beta r_0 h}\lambda_\theta c_\ell \cos\sigma + \mu = 0 \tag{90a}$$

$$H_{c_\ell} = \lambda_V \frac{BV^2 e^{-\beta r_0 h} c_\ell}{E^*} + \lambda_\gamma BVe^{-\beta r_0 h}\cos\sigma + \lambda_\theta BVe^{-\beta r_0 h}\sin\sigma = 0 \tag{90b}$$

Unlike the previous section, σ is now known to be at its maximum. Thus, when bank angle is at its maximum positive value the optimal c_ℓ is:

$$c_\ell = \frac{(\lambda_\gamma \cos\sigma + \lambda_\theta \sin\sigma)E^*}{V\lambda_V} \tag{91}$$
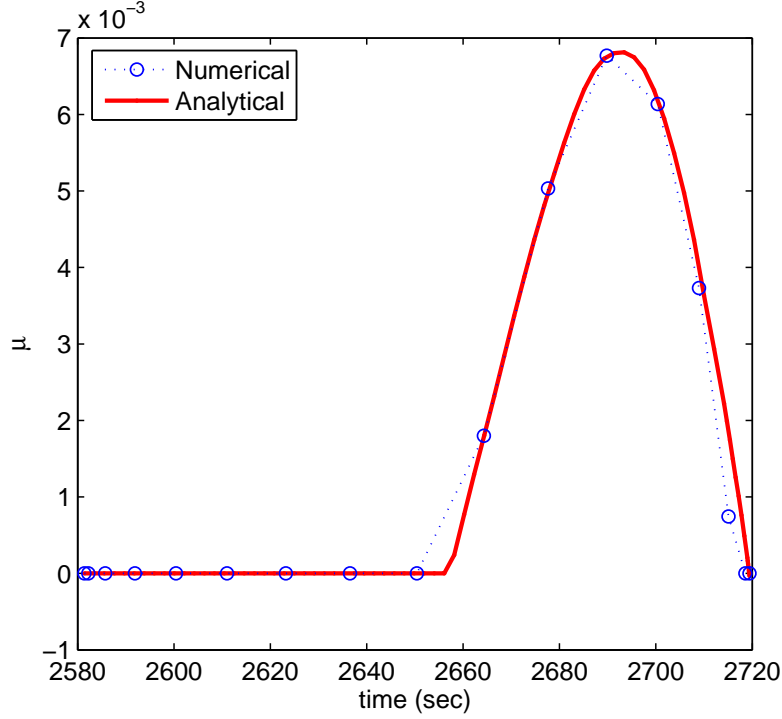
Figure 22: Control Multiplier μ for Constrained Bank Angle ($\sigma = 60°$)

To compare this derivation to the numerical output, Equation (91) is used in Equation (90a) to define μ:

$$\mu = BVe^{-\beta r_0 h}(\lambda_\gamma \sin \sigma - \lambda_\theta \cos \sigma)c_\ell \tag{92}$$

The comparison between the numerically generated results and this analytical form is presented in Figure 22. The time span in Figure 22 is at the end of the trajectory where maximum bank angle is required to achieve the optimal solution. The matching results demonstrate that the numerical results are satisfying the optimality criteria. For the case of maximum c_ℓ, the optimality criteria in Equations (87) and (88) remain valid since Equation (79a) is not influenced by c_ℓ (c_ℓ is present but it is divided out thus leaving the originally derived solution).

70
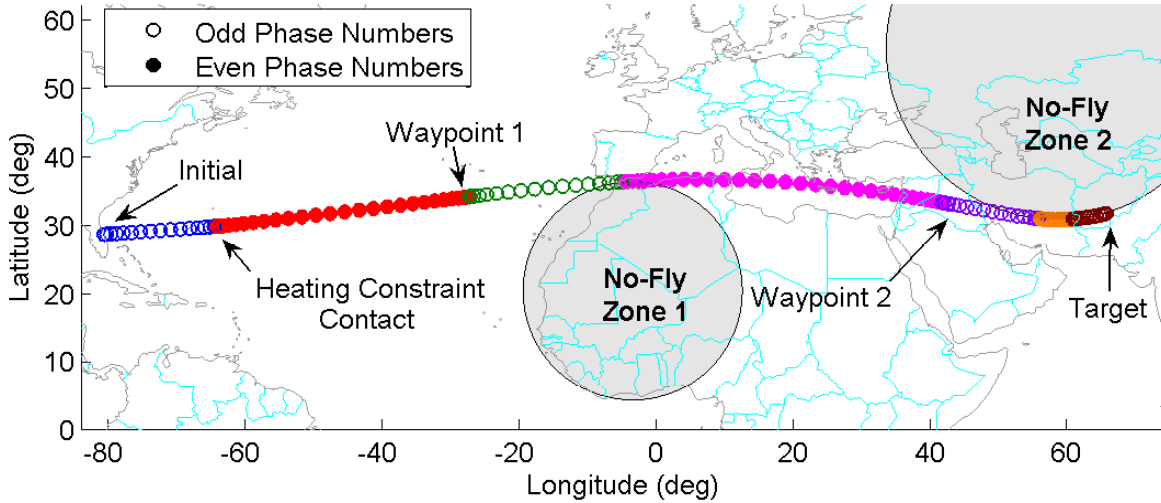
Figure 23:   Map: Seven Phase Numerical Results

4.8   3-D Numerical Results

To maximize the accuracy of the numerical results the trajectory is broken up such that every jump in the costates occurs at the end of a phase. This requires an iterative approach to produce these more accurate results; however, the more accurate results are used to verify optimality and to test the solution convergence with fewer phases, i.e. less iterations from previous solutions.

The costate jumps occur at contact with the heating and no-fly zone constraints. Preliminary numerical results, with fewer phases, are used to identify the number of path constraint contacts. Next, additional phases are specified to more accurately capture these events and verify the original results. Thus, this trajectory is broken into seven phases; t_0 to heating constraint contact, heating constraint contact to waypoint 1, waypoint 1 to no-fly zone 1, no-fly zone 1 to waypoint 2, waypoint 2 to first contact of no-fly zone 2, first contact of no-fly zone 2 to second contact of no-fly zone 2, second contact of no-fly zone 2 to target, as seen on the map in Figure 23.   Figure 24 shows the states throughout the seven phases. One important factor is that the states remain continuous across the phase breaks. These numerical results are using collocation techniques, not integration, so verifying the results satisfy the equations of motion is presented in the next section. The magnitude of flight path angle, γ,
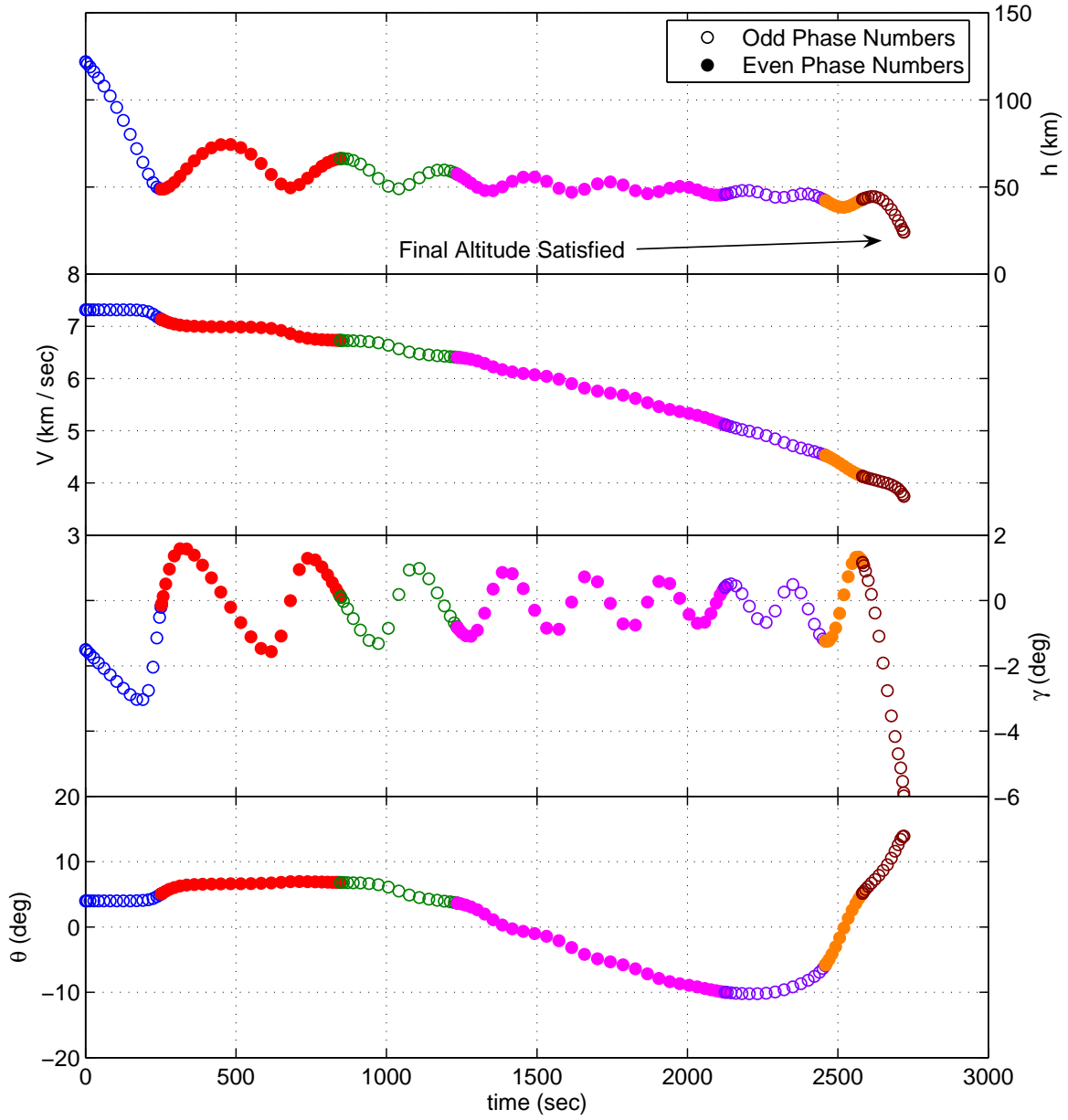
Figure 24:   States: Seven Phase Numerical Results

remains small throughout the trajectory; therefore, the small angle approximation is still valid. Also, the specified final altitude is seen to be satisfied in the top graph of Figure 24.

The costates are presented in Figure 25. The jumps in the costates are at the end of each phase, as expected. The zero derivatives of λ_x and λ_y are clearly seen. The terminal boundary conditions for λ_V, λ_γ, and λ_θ are all met by equalling zero, thus satisfying Equation (13) using Equation (34). Finally, the Hamiltonian is constant and continues to satisfy $H(t) = -1$.

The controls are plotted in Figure 26. The controls are seen to remain within their specified limitations; for example, c_ℓ remains on the limit of $c_\ell = 2$ at approximately 200 seconds into the trajectory, and bank angle remains at the limit of $\sigma = 60°$ at approximately 2700 seconds. The path constraint multiplier for this maximum bank angle constraint contact was previously plotted in Figure 22.

The path constraints are shown in Figure 27, with the no-fly zone constraints S labeled on the left and the heating constraint Q labeled on the right. Expanded views of these path constraints are plotted in Figure 28. Contact with a path constraint occurs when the value of the constraint equals zero, i.e. $S = 0$ from Equation (35) or $Q = 0$ from Equation (41). For each path constraint there is contact without travel along the constrained arc. In Figure 28(c) there are two separate contacts on no-fly zone 2, hence, the two sets of costate jumps labeled in Figure 25.

The next section takes the numerical costate histories in Figure 25, computes the analytical controls from Sections 4.7.1 and 4.7.2, and integrates the states. The intent is to verify that the numerical results are properly estimating the optimal solution.
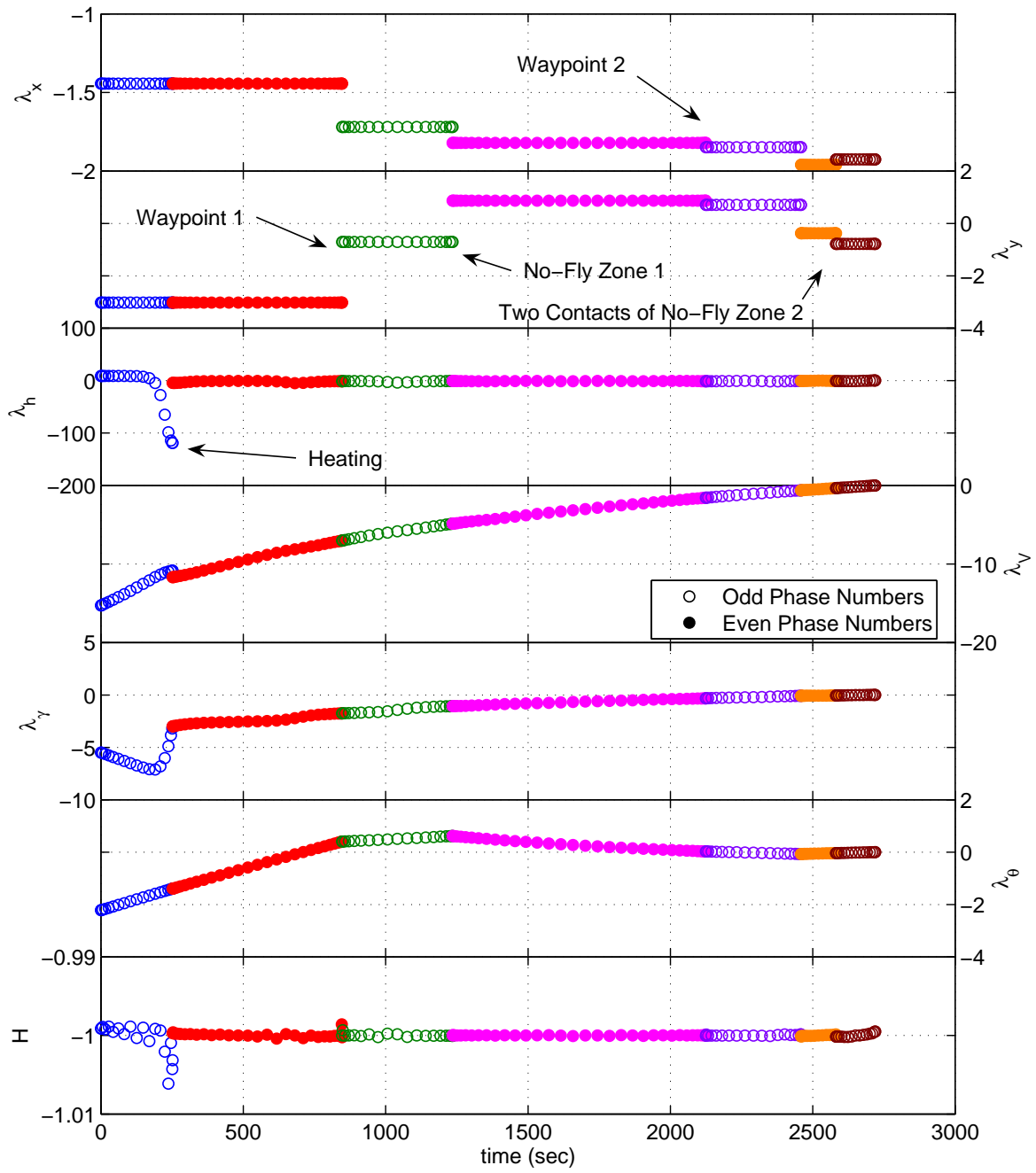
Figure 25: Costates: Seven Phase Numerical Results
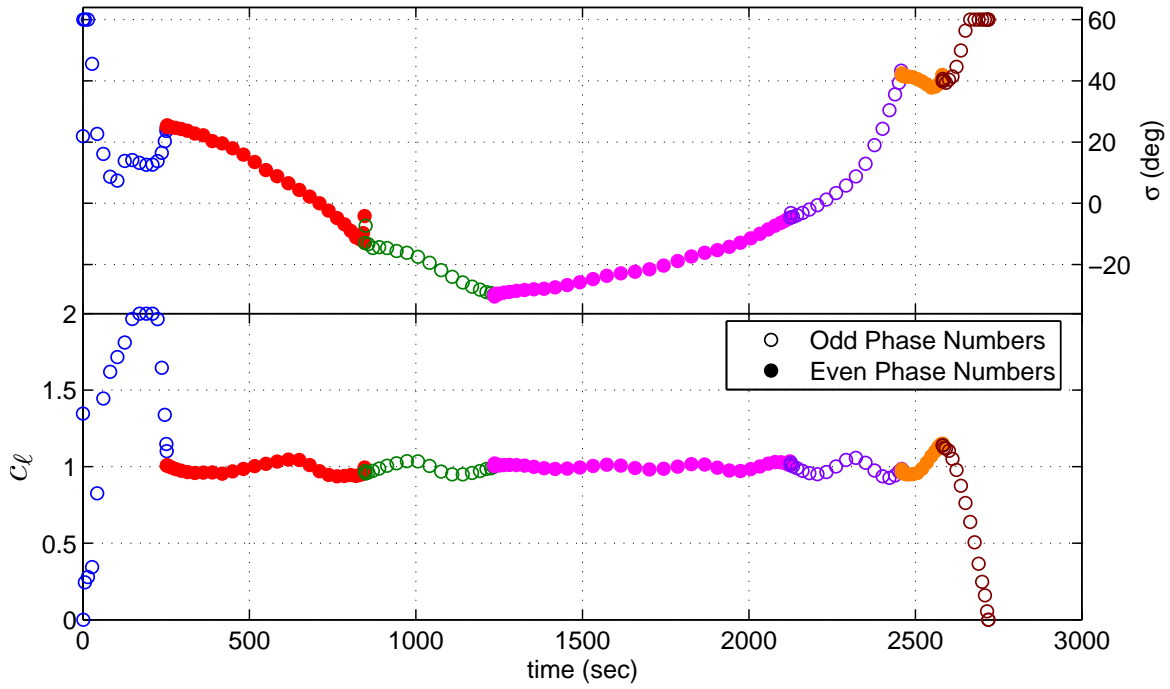
Figure 26:   Controls: Seven Phase Numerical Results
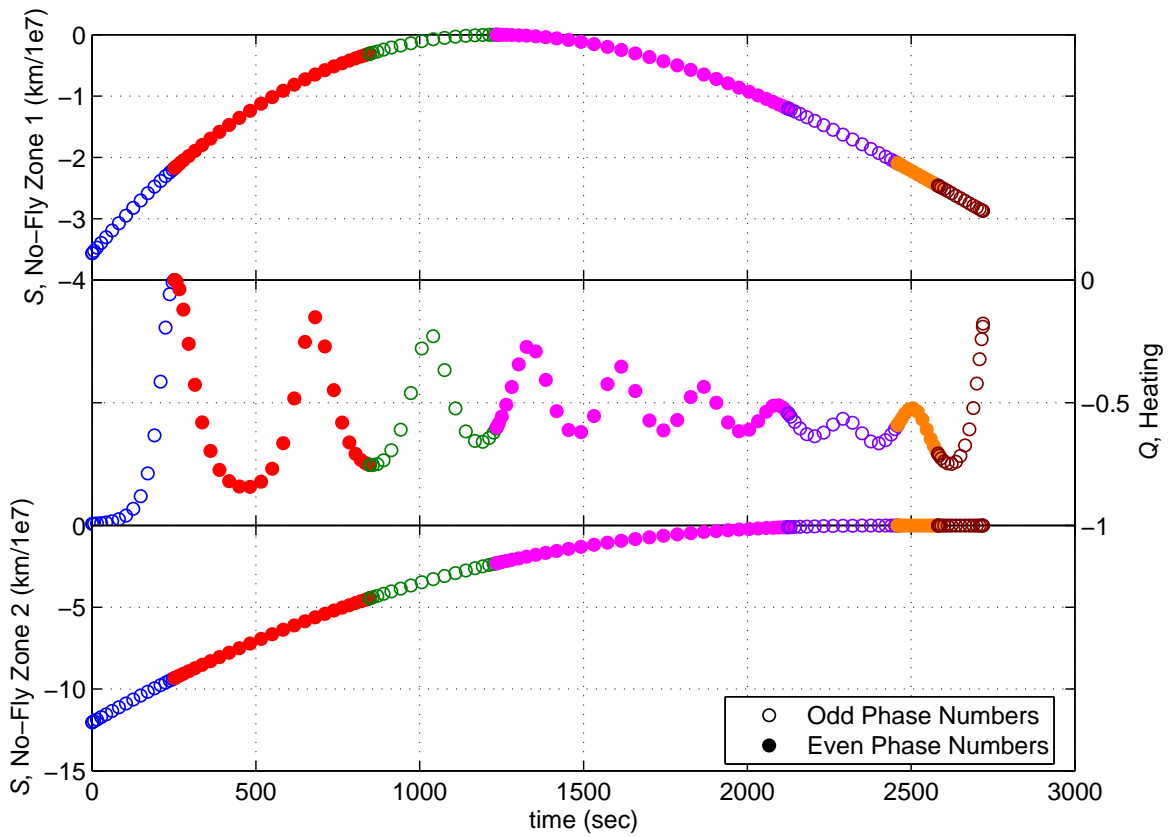
Figure 27:   Path Constraints: Seven Phase Numerical Results

75

(a) Q, Heating   (b) S, No-Fly Zone 1   (c) S, No-Fly Zone 2
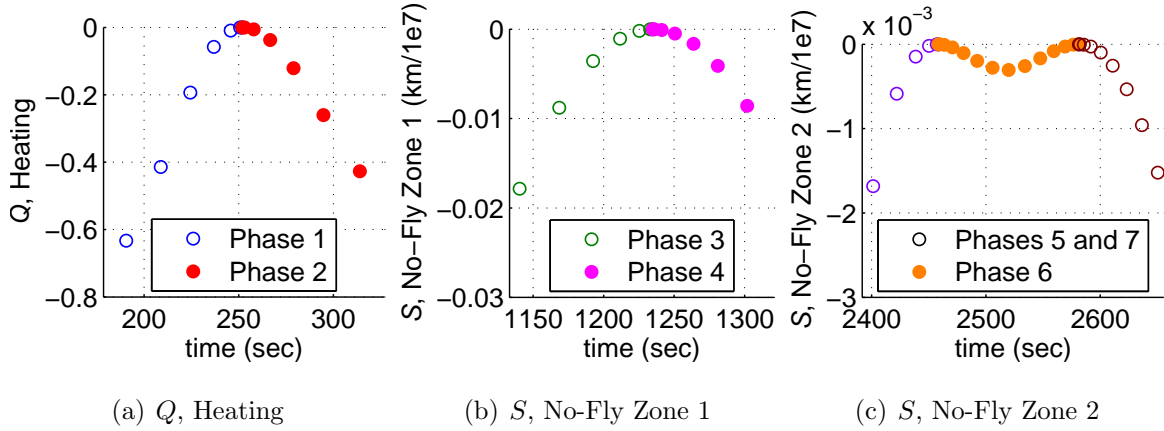
Figure 28:   Path Constraints Expanded: Seven Phase Numerical Results

4.9   3-D Comparison, Analytical versus Numerical

In Figures 29 through 33 the numerical results are compared to the form of the solution analytically derived in Sections 4.7.1 and 4.7.2. The states and constraints are nearly a perfect match. For the controls in Figure 31 the analytical results appear smoother, especially at the beginning of the trajectory. This is attributed to costate interpolation used to compute the controls. The derivations in Section 4.7.2 discussed the change in optimal control law when a control constraint is reached. The correlation in the results at the end of the trajectory in Figure 30 demonstrate the successful implementation of the revised control law. The data point jump at the last point in Figure 31 is caused from the singularities that arise as the costates λ_V, λ_γ, and λ_θ approach the analytically derived values of zero. Lastly, in the expanded view in Figure 33 there are differences in constraint contact points. Herein lies the instability with the integration in the shooting method, the points of contact and departure must be exact or the propagated costates may diverge.
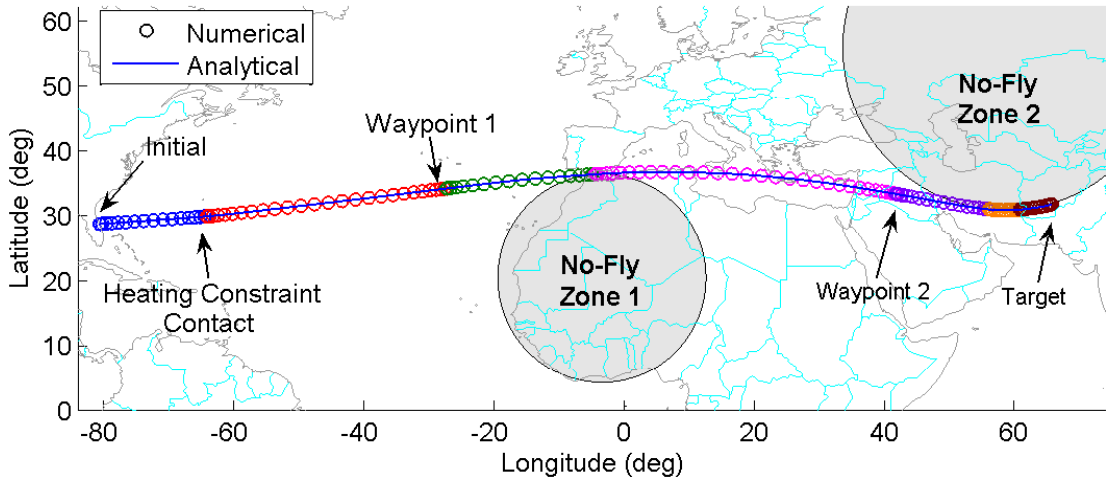
Figure 29:   Map: Seven Phase Numerical and Analytical
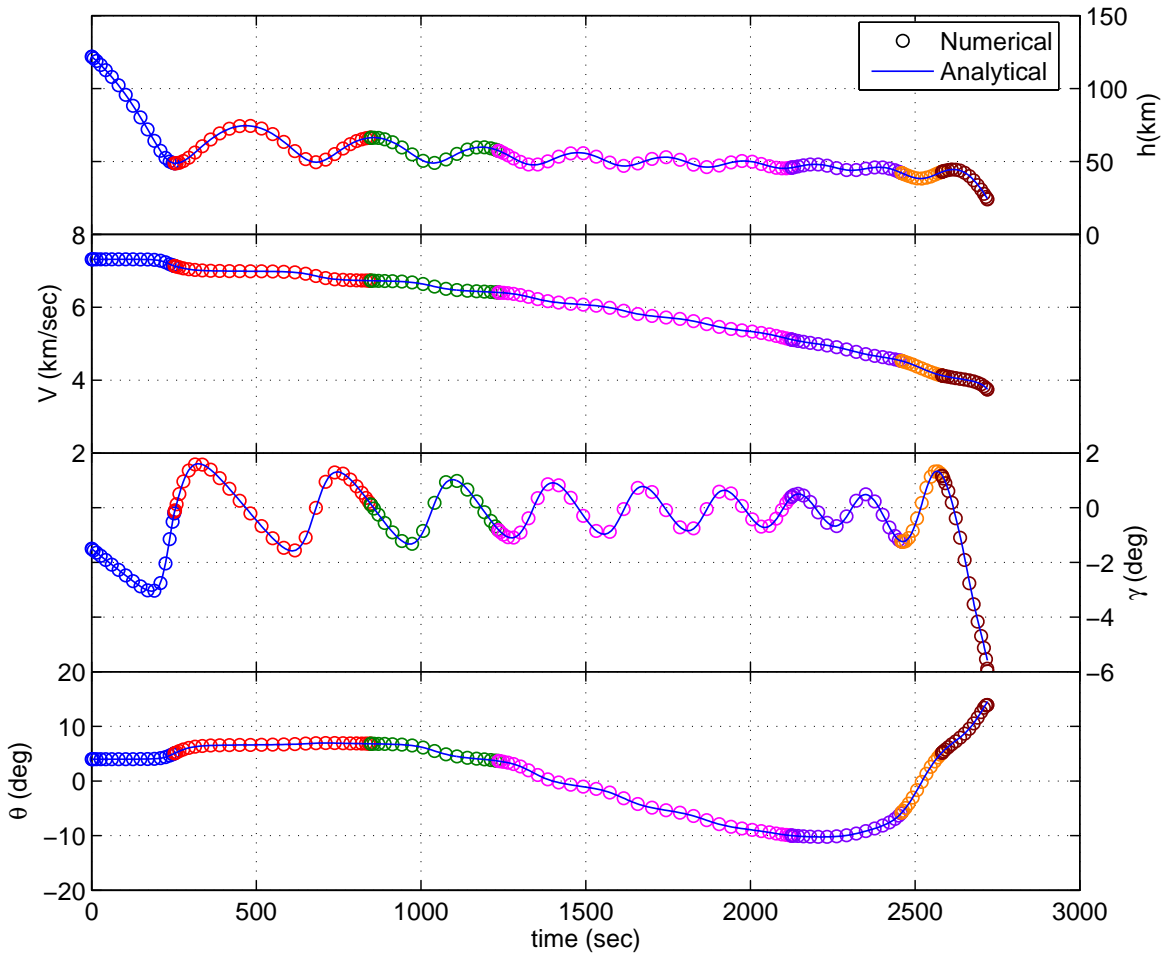
Figure 30:   States: Seven Phase Numerical and Analytical

77
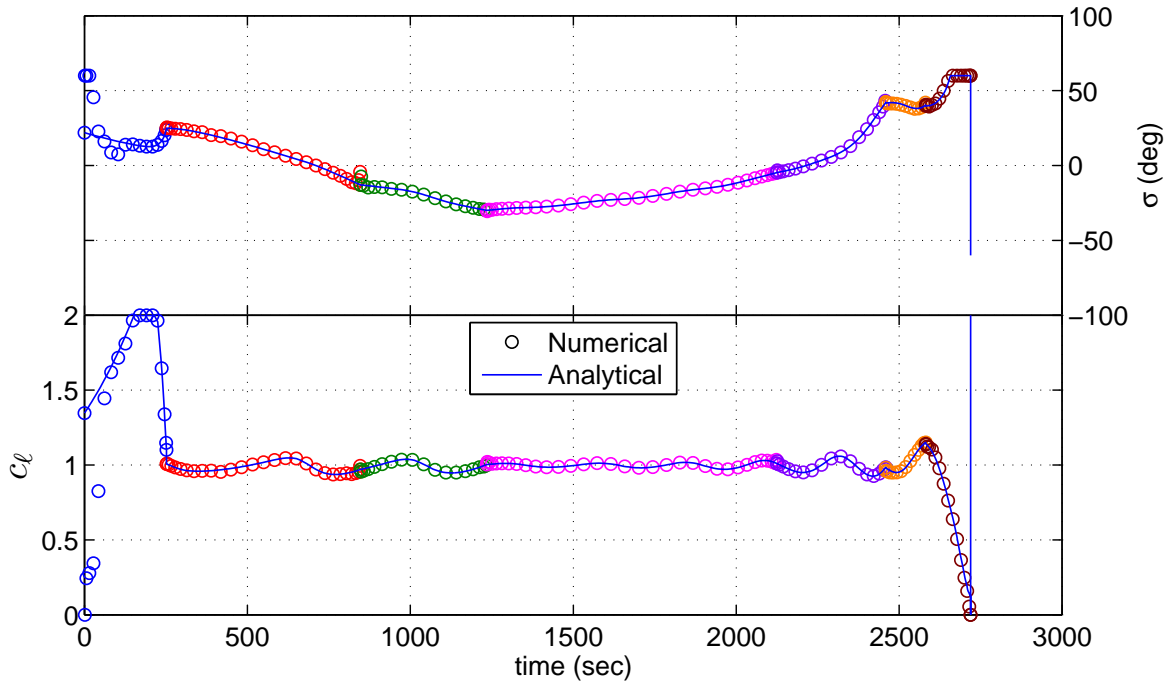
Figure 31:   Controls: Seven Phase Numerical and Analytical
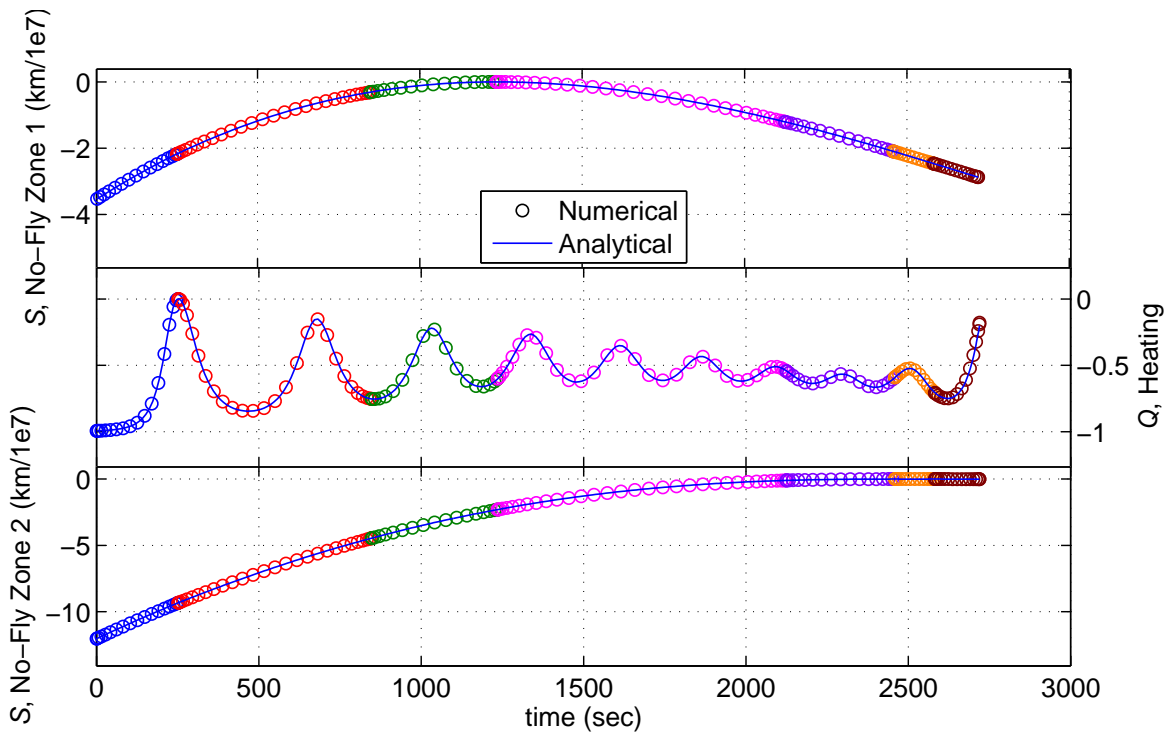
Figure 32:   Path Constraints: Seven Phase Numerical and Analytical

(a) Q, Heating   (b) S, No-Fly Zone 1   (c) S, No-Fly Zone 2
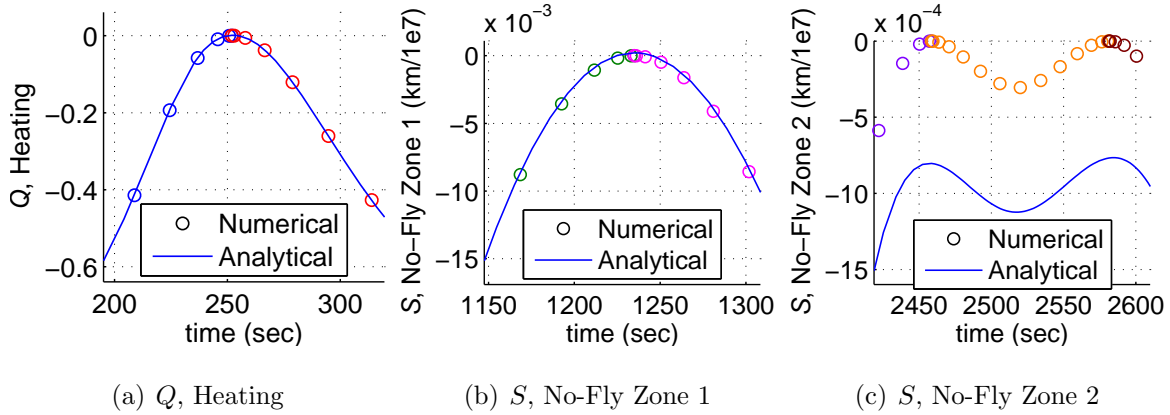
Figure 33:   Path Constraints Expanded: Seven Phase Numerical and Analytical

4.10   Numerical Comparison of Phase Breakpoints

This rigorous analysis forces this problem to be broken up into many phases, several required advanced knowledge of the solution, i.e. the number of boundary contacts.   The next objective is to evaluate the solution effectiveness using fewer phases. The map results are seen in Figure 34.
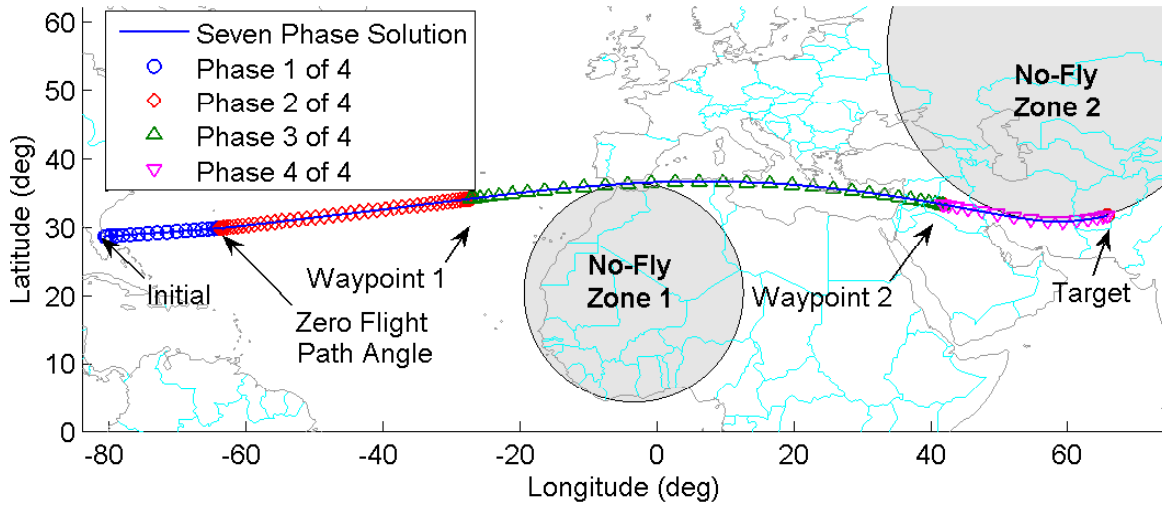
Figure 34:   Map: Number of Phases Comparison

For this mission, there are two known intermediate waypoints; therefore, there are at least three phases; t_0 to $t_{i=1}$, $t_{i=1}$ to $t_{i=2}$, and $t_{i=2}$ to t_f. Through some solution knowledge, but without confining the solution, a fourth phase is added. This phase

79

ends at the first point the flight path angle passes through zero. The rational is the first assault on the heating limit requires the most accuracy; therefore, by placing a phase transition here, near the potential heating constraint contact, additional solution accuracy is achieved. The rigorous seven phase results are compared to the much simpler four phase results in Figures 34 through 39.
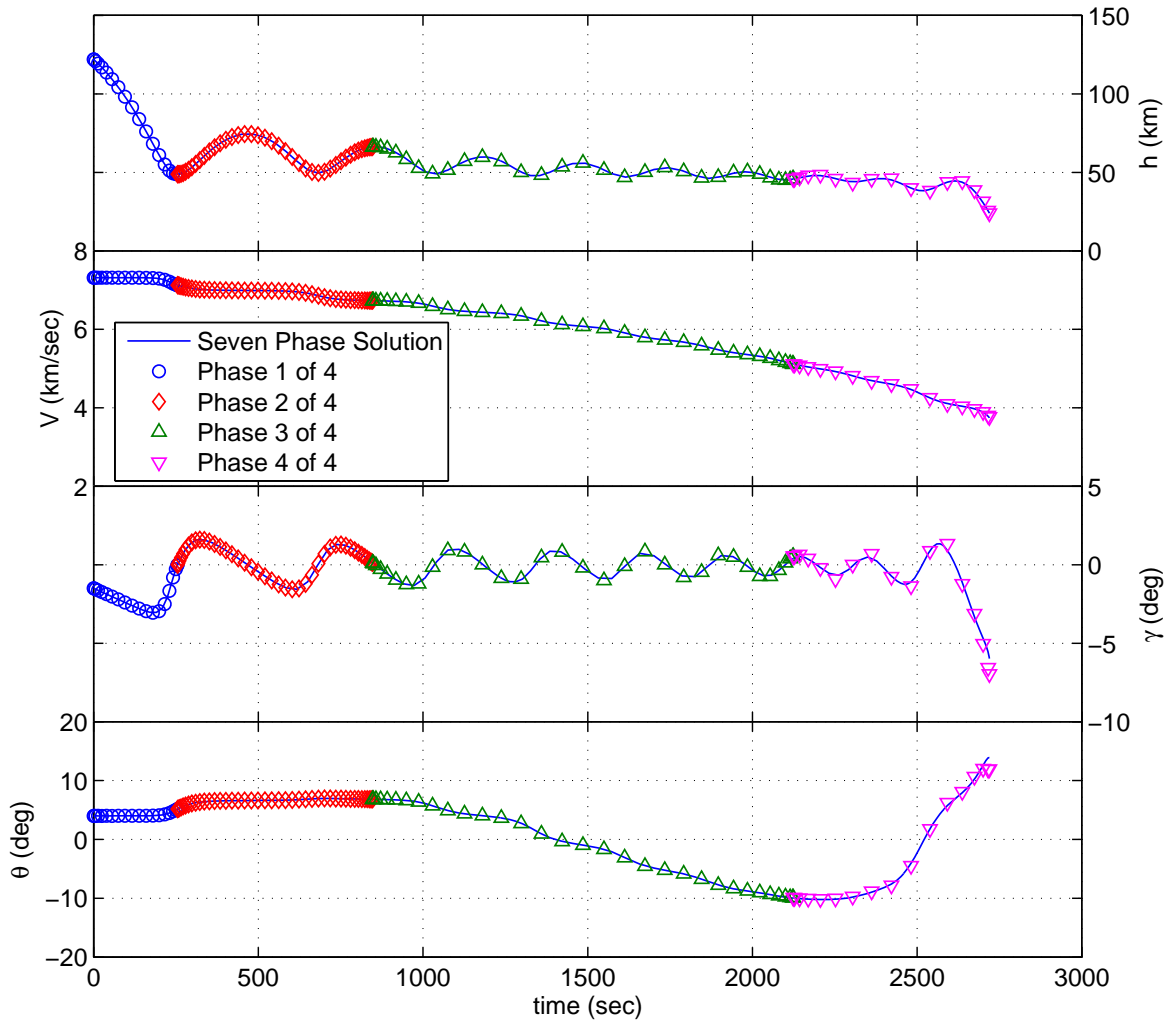
Figure 35:   States: Number of Phases Comparison
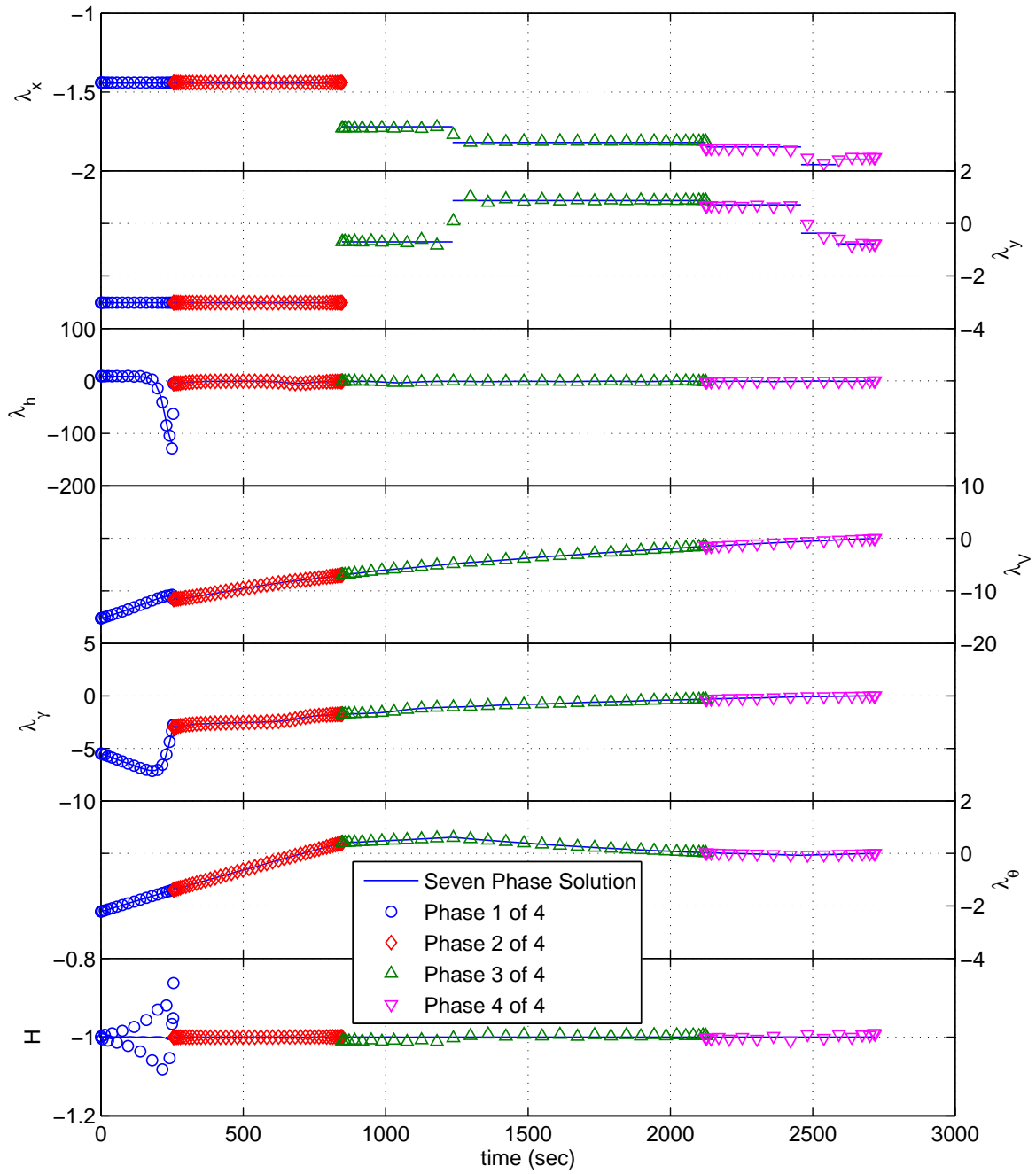
Figure 36:   Costates: Number of Phases Comparison
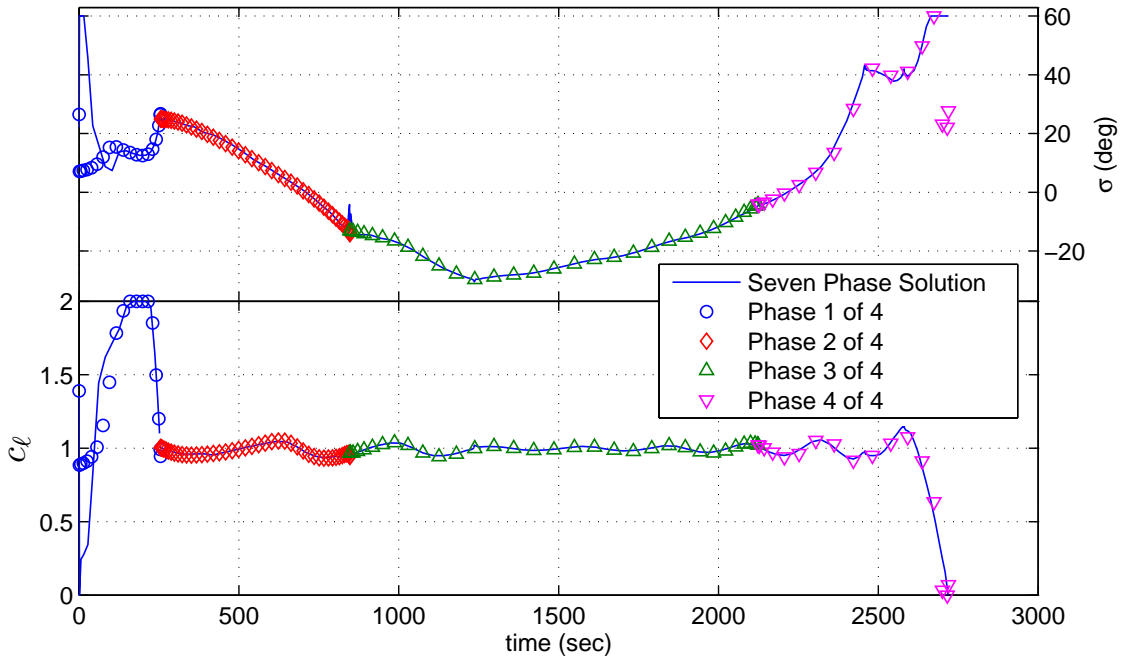
Figure 37:   Controls: Number of Phases Comparison
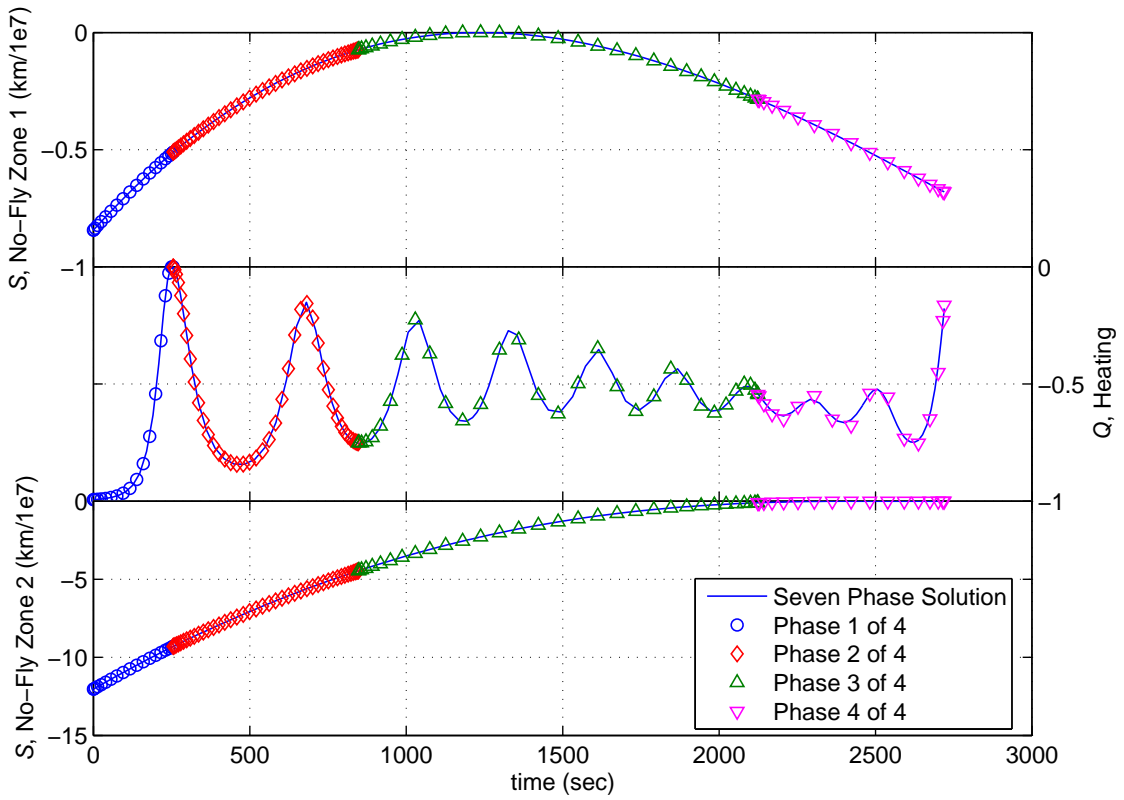
Figure 38:   Path Constraints: Number of Phases Comparison

(a) Q, Heating (b) S, No-Fly Zone 1 (c) S, No-Fly Zone 2
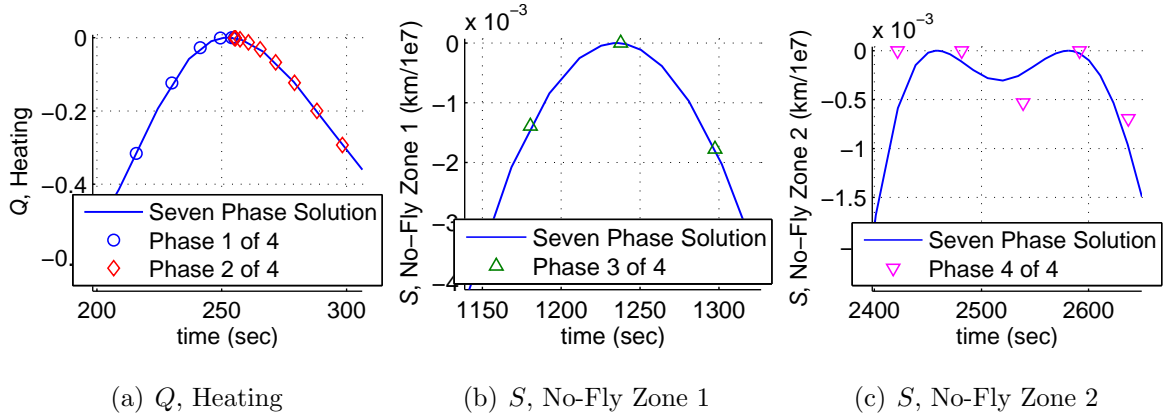
Figure 39:   Path Constraints Expanded: Number of Phases Comparison

The closeness in these results demonstrates that the jump in costates due to path constraint contact can be estimated as continuous. Quantitatively, the final times for the seven phase and four phase solutions are 2719.42 sec and 2719.18 sec, respectively; thus, well within the accuracy of the model. Future analysis is required to generalize this small error to higher numbers of waypoints, no-fly zones, and heating constraint contacts. The setup for the four phase problem is much simpler and it does not require advanced knowledge of the number of times the path constraints are contacted. Due to the direct connection between phases and waypoint, regardless of heating or no-fly zones, it makes it possible to automate the setup based on the mission objectives.

4.11   Summary of Analysis

The development herein built upon itself to eventually determine and verify a viable optimal reentry trajectory solution technique. The simpler 2-D HCV case is used as a foundation to understand and introduce the analytical and numerical techniques. The geometric solution provides an independently developed optimal solution to validate both the analytical and numerical solutions. Bryson's path constraint method is seen to differ from that used in the numerical collocation software. Understanding this difference provides insight into which costates are expected to have discontinuities; therefore, provides a means of replicating the results using analytical methods. The more complex 3-D CAV model demonstrates the ability to incorporate

all of the vehicle and mission constraints. Using dynamic optimal control theory an analytical solution is derived to validate these numerical results. After verifying that the numerical results are indeed satisfying the optimality criteria, the next step is to simplify implementation. By replicating the numerical results using just four phases instead of the full seven phases demonstrates that a priori knowledge of path constraint contact is unnecessary; therefore, an automated problem setup technique can be developed primarily dependent on the number of waypoints. The following section explores the importance of these results.

V. Conclusions, Contributions, and Future Work

Having presented the problem, analysis, and results the next step is to address the conclusions to be drawn, how they affect the user and the research community, and what future work lies ahead.

5.1 Conclusions

This research successfully determined and verified a numerical technique capable of generating an optimal reentry trajectory satisfying all of the specified vehicle and mission constraints. The suitability for satisfying the Global Strike mission also includes the autonomy and user implementation of the solution technique. The solution is considered autonomous due to several factors; manual manipulation of variables is not required, the problem has a repeatable setup structure, and the solution can converge even with a very simple "point-to-target" initial guess. The repeatability is the structured setup based on the number of waypoints and convergence independent of the number of no-fly zones or heating constraint contacts. The desirable user implementation is commensurate with the autonomy; namely repeatability and no a priori solution requirements. The next measure of mission suitability is the adaptability to alternate missions. Due to the ability to set up independent phases, the number of waypoints should not hinder solution performance. Also, since the knowledge of the number of no-fly zone and heating constraint contacts is not required in the problem setup, this technique dynamically determines the optimal solution. Lastly, user implementation does not require extensive training or excessive intervention to converge to a solution. Knowledge of optimal control theory is not required, and the described problem setup minimized user intervention or iterations. Thus, once the cost and dynamics are defined, the phases can be automatically separated based on the given waypoints and if a feasible solution exist, observed convergence has been extremely fast. Addressing infeasibility is discussed in Section 5.3, while the contributions of this research are covered in the next section.

5.2 Contributions

The contributions arise from both the 2-D model research and the 3-D model research. The optimal geometric solution is derived within the 2-D model research. The geometric solution is invaluable since it provides the optimal solution with no integration for the constant speed case; and provides a near-optimal solution for the deceleration case. Thus, any autonomous unmanned vehicle with limited computing capacity could benefit greatly from such a computationally efficient algorithm. Furthermore, any hypersonic vehicle with increased crossrange capability could benefit by using the geometric solution as an initial guess satisfying the waypoint and no-fly zone constraints; however, the "point-to-target" initial guess in this research was sufficient thus the geometric solution was not required. The application of the numerical collocation solution technique to this highly constrained reentry trajectory problem is an outcome of the 3-D model research. Applying the solution technique to the constrained reentry trajectory problem was one contribution, while verifying the optimality was another equally important contribution. Furthermore, detailing an efficient user implementation strategy strengthened the applicability of this numerical technique. For example, the proposed user implementation solution took minutes on a current day desktop computer, as compared to hours of manually coding additional phase breakpoints and interpreting multiple solution iterations. The autonomy of the proposed user strategy also contributes to rapid war-gaming analysis for assessing mission capability and anticipating mission success. Overall, this research developed enabling solution methods and implementation techniques to enhance Global Strike mission success.

5.3 Recommendations for Future Research

This research problem statement intentionally included a narrow scope of constraint definitions; motivated by existing user requests and compatibility to detailed analytical techniques. Increasing the model fidelity, broadening the constraint definitions, or increasing the depth of analysis of the existing problem, can open up a

wide band of future research. The flat Earth model should be expanded to a rotating spherical Earth model to produce more operationally representative results. Also, the total heating should be constrained or verified to be within the vehicle design tolerance. Other topics could include quantifying the ability to accommodate additional waypoints and no-fly zones, or analyzing the effects of raising or lowering the heating constraint. Also, research could investigate techniques to counter infeasibilities; such as creating "soft", or cost based, waypoints and/or no-fly zones; or adding the ability to increase initial altitude/velocity/energy if the target is unreachable. The problem setup in this research can easily be modified to maximize velocity for a set terminal altitude, or maximize terminal energy at a given coordinate. Altitude could be added to no-fly zones by modeling as hemispheres or cylinders with a top altitude. Other shapes for no-fly zones could be specified, e.g. ellipses or a generic shape with smooth corners, $(x/a)^p + (y/b)^r = 1$. Additional state constraints at waypoint passage could be added, e.g. zero bank angle for a reconnaissance platform. Terminal flight path or heading angle may be confined to accommodate the terminal guidance of another reentry vehicle. Performing a study of the number of nodes as a function of computational time, versus accuracy and convergence may also be of interest. Dynamic meshing (number of nodes) may lead to faster solutions; e.g. start with a course grid, then use that as the initial guess to a finer more accurate grid. Dynamic phase breakpoints could be implemented, i.e. adding phases at identified areas of interest from a previous solution. For a vehicle with greater crossrange, the geometric approach may contribute significantly to convergence when used as an initial guess. Mission analysis could be performed; e.g. target footprints, time-to-respond, and maximum accessible intermediate targets. No-fly zones could be treated more like threat zones with likelihood of damage proportional to threat proximity, or probability of survival could be calculated. Bank angles in excess of 90 degrees and/or negative angle-of-attack ($c_\ell < 0$) could be used to define new missions, such as closer range with higher kinetic kill capability. Also, pop-up threats or real-time waypoint tasking can be investigated. This is commensurate with the ongoing RTOC research. The definition of

"soft" waypoints could include achieving the highest number of some, yet completely ignoring others; e.g. maximizing the number of reconnaissance targets per mission. The inner-loop guidance could be implemented; i.e. compute the optimal trajectory then perform closed-loop tracking of that trajectory. In conclusion, the above list represents yet another contribution of this research to enable a whole realm of Global Strike mission enhancements.

Appendix A.   Common Aero Vehicle (CAV) Aerodynamics

A.1   CAV Aerodynamic Data

The following is taken from [69] and is used to create a Mach independent model for use in this research.

Table A.1:   CAV-H Aero Data Base

AOA	Mach 3.5	Mach 5	Mach 8	Mach 10	Mach 15	Mach 20	Mach 23
Lift to Drag Ratio (L/D)							
10°	2.2000	2.5000	3.1000	3.5000	3.3846	3.2692	3.2000
15°	2.5000	2.6616	2.9846	3.2000	3.0846	2.9692	2.9000
20°	2.2000	2.3616	2.6846	2.9000	2.7846	2.6692	2.6000
Coefficient of Lift (C_L)							
10°	0.4500	0.4250	0.4000	0.3800	0.3700	0.3600	0.3500
15°	0.7400	0.7000	0.6700	0.6300	0.6000	0.5700	0.5570
20°	1.0500	1.0000	0.9500	0.9000	0.8500	0.8000	0.7800
Coefficient of Drag (C_D)							
10°	0.2045	0.1700	0.1290	0.1090	0.1090	0.1090	0.1090
15°	0.2960	0.2630	0.2240	0.1970	0.1950	0.1920	0.1920
20°	0.4770	0.4230	0.3540	0.3100	0.3050	0.3000	0.3000

CAV-H Aero Reference Area $S_{ref} = 750$ in^2

CAV-H Mass $m = 2000$ lbs $= 907.186$ kg (assuming g $= 32.174$ ft/s^2)

A.2   CAV Aerodynamic Model

The coefficient of drag (C_D) is assumed to be a parabolic function of the coefficient of lift (C_L) [111]:

$$C_D = C_{D_0} + KC_L^2 \tag{A.1}$$

Taking this definition of C_D leads to an expression for lift-over-drag, L/D:

$$\frac{L}{D} = \frac{C_L}{C_D} = \frac{C_L}{C_{D_0} + KC_L^2} \tag{A.2}$$

In order to find the C_L that will produce the maximum L/D, take the partial derivative of Equation (A.2) with respect to C_L:

$$\frac{\partial(L/D)}{\partial C_L} = \frac{1}{C_{D_0} + KC_L^2} - \frac{2KC_L^2}{(C_{D_0} + KC_L^2)^2} \tag{A.3}$$

Set Equation (A.3) equal to zero and solve for the optimal C_L, C_L^*:

$$C_{D_0} + KC_L^2 - 2KC_L^2 = 0 \tag{A.4}$$

Thus

$$C_L^* = \sqrt{\frac{C_{D_0}}{K}} \tag{A.5}$$

Input this equation into Equation (A.1) to find the corresponding C_D, C_D^*:

$$C_D^* = C_{D_0} + K\left(\sqrt{\frac{C_{D_0}}{K}}\right)^2 \tag{A.6}$$

$$C_D^* = 2C_{D_0} \tag{A.7}$$

Then the maximum L/D is:

$$E^* = \frac{C_L^*}{C_D^*} \tag{A.8}$$

The normalized coefficient of lift is defined as:

$$c_\ell = \frac{C_L}{C_L^*} \tag{A.9}$$

The values in Table A.1 are plotted in Figure A.1 with the curve fit using:

$$C_L^* = 0.45 \tag{A.10a}$$

$$E^* = 3.24 \tag{A.10b}$$

Assuming $C_{L_{max}} = 0.9$ from Figure A.1

$$c_{\ell_{max}} = \frac{C_{L_{max}}}{C_L^*} = 2.0 \tag{A.10c}$$

90
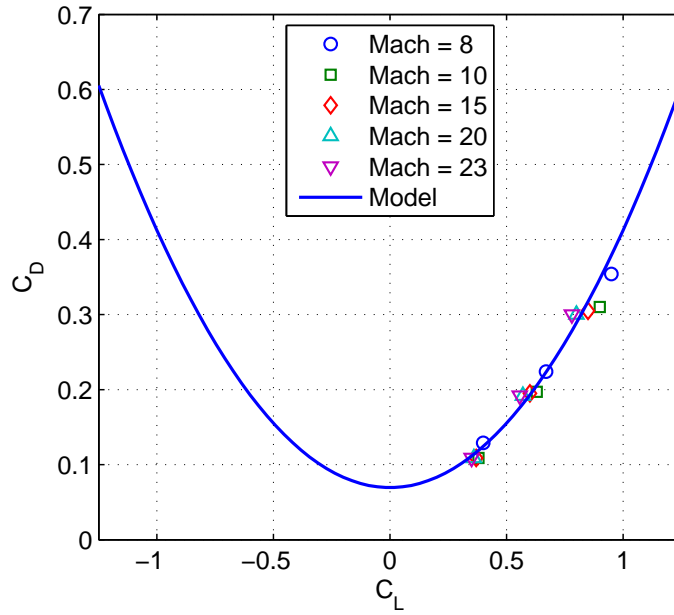
Figure A.1:   CAV-H Aerodynamic Data and Model

From Figure A.1 the Mach independence appears valid; therefore, for this research Equation (A.10) is used to represent the CAV model. In Figure A.2 the new nondimensional variable c_ℓ is plotted to verify the $c_\ell = 1$ corresponds to the maximum L/D.
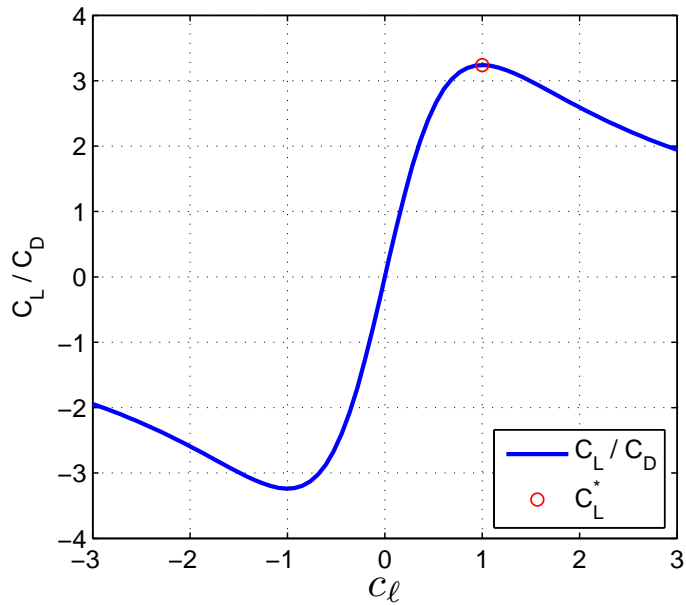
Figure A.2:   CAV-H Nondimensional Variable c_ℓ

91

Appendix B. Nondimensionalization

Nondimensionalization simplifies the equations of motion and ensures all states, paths, and cost are the same order of magnitude. Scaling is critical for successful implementation of numerical solution techniques. This research uses a flat Earth model; however, a radius is used for normalization to be consistent with other references [41; 109; 110], and to make an easy transition to a spherical Earth model for future research. This radius is the radius of the Earth $R_\oplus$ plus the flat Earth altitude. The nondimensionalization herein starts with the spherical rotating Earth equations of motion and simplifies these equations, and changes notation, to arrive at the flat Earth equations of motion used in this research.

B.1 Spherical Earth 3-D Reentry Equations of Motion - Rotating Earth

Assume thrust (T) is aligned with vehicle velocity vector. The states are radius (r) from the center of the Earth, longitude (θ_S), latitude (ϕ_S), velocity (V_S), flight path angle (γ_S) as defined in Figure B.1, and heading (ψ_S) measured from East counterclockwise. Equation (B.1) are the *Kinematic Equations* and *Force Equations*.

$$\dot{r} = V_S \sin \gamma_S \tag{B.1a}$$

$$\dot{\theta}_S = \frac{V_S \cos \gamma_S \cos \psi_S}{r \cos \phi_S} \tag{B.1b}$$

$$\dot{\phi}_S = \frac{V_S \cos \gamma_S \sin \psi_S}{r} \tag{B.1c}$$

$$\dot{V}_S = \frac{(T - D_S)}{m} - g \sin \gamma_S + r\omega_\oplus^2 \cos \phi_S \left(\cos \phi_S \sin \gamma_S - \sin \phi_S \sin \psi_S \cos \gamma_S\right) \tag{B.1d}$$

$$\begin{aligned}
\dot{\gamma}_S = \frac{1}{V_S} \Bigg[& \frac{L_S}{m} \cos \sigma - g \cos \gamma_S + \frac{V_S^2}{r} \cos \gamma_S \\
& + 2V_S \omega_\oplus \cos \phi_S \cos \psi_S + r\omega_\oplus^2 \cos \phi_S \left(\cos \phi_S \cos \gamma_S + \sin \phi_S \sin \psi_S \sin \gamma_S\right) \Bigg]
\end{aligned} \tag{B.1e}$$

$$\begin{aligned}
\dot{\psi}_S = \frac{1}{V_S} \Bigg[& \frac{L_S \sin \sigma}{m \cos \gamma_S} - \frac{V_S^2}{r} \cos \gamma_S \cos \psi_S \tan \phi_S \\
& + 2V_S \omega_\oplus \left(\sin \psi_S \cos \phi_S \tan \gamma_S - \sin \phi_S\right) - \frac{r\omega_\oplus^2}{\cos \gamma_S} \sin \phi_S \cos \phi_S \cos \psi_S \Bigg]
\end{aligned} \tag{B.1f}$$

Here velocity (V_S) is relative, i.e. velocity is not inertial. When the kinematic equations are integrated, they provide the position of the vehicle as seen from the rotating planet [110]. The Earth's rotation is $\omega_\oplus$, lift is L_S, drag is D_S, bank angle is σ, and mass is m. The subscript S refers to spherical, thus alleviating confusion with previously used variables. The first term in Equation (B.1d) is the net acceleration from the difference between the thrust T and the drag D_S, whereas the second term is the component of gravity in the velocity direction. The terms involving the Earth's rotation $\omega_\oplus$ are later eliminated with the zero Earth rotation assumption. The first three terms of Equation (B.1e) are the change in flight path angle due to the vertical component of lift, the component of gravity perpendicular to the velocity vector, and the component of centripetal force perpendicular to the velocity vector. The first two terms of Equation (B.1f) are the component of lift perpendicular to the r-V plane, and centripetal force component perpendicular to the r-V plane. The r-V plane is shown in Figure B.2.
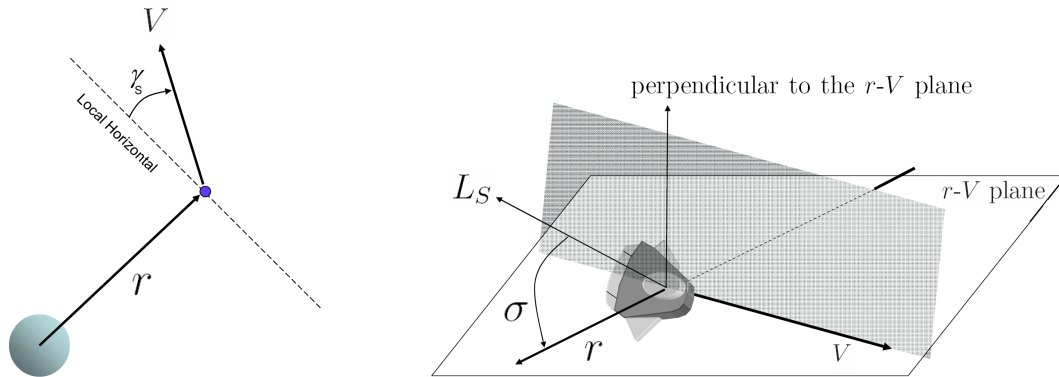
Figure B.1:   Fight Path Angle [41]   Figure B.2:   Schematic of the r-V plane [41]

B.2   Spherical Earth 3-D Reentry Equations of Motion - Non-Rotating Earth

The first step is to assume no rotation of the Earth, $\omega_\oplus = 0$, thus simplifying Equations B.1 [41; 109; 110].

$$
\begin{align}
\dot{r} &= V_S \sin \gamma_S \tag{B.2a}\\
\dot{\theta}_S &= \frac{V_S \cos \gamma_S \cos \psi_S}{r \cos \phi_S} \tag{B.2b}\\
\dot{\phi}_S &= \frac{V_S \cos \gamma_S \sin \psi_S}{r} \tag{B.2c}\\
\dot{V}_S &= \frac{(T - D_S)}{m} - g \sin \gamma_S \tag{B.2d}\\
\dot{\gamma}_S &= \frac{1}{V_S} \left[\frac{L_S}{m} \cos \sigma - g \cos \gamma_S + \frac{V_S^2}{r} \cos \gamma_S \right] \tag{B.2e}\\
\dot{\psi}_S &= \frac{1}{V_S} \left[\frac{L_S \sin \sigma}{m \cos \gamma_S} - \frac{V_S^2}{r} \cos \gamma_S \cos \psi_S \tan \phi_S \right] \tag{B.2f}
\end{align}
$$

The states are radius from the center of the Earth (r), longitude (θ_S), latitude (ϕ_S), velocity (V_S), flight path angle (γ_S), and heading (ψ_S) measured from East counterclockwise. The flight path angle is defined in Figure B.1. The radius r is the sum of the Earth radius $R_\oplus$ and the altitude above the Earth.

B.3   Flat Earth 3-D Reentry Equations of Motion - Non-Rotating Earth

The following equations of motion are from [114:pg.249]; however, with appropriate assumptions they can also be derived from Equation (B.2). The following assumptions are used to obtain the dimensional flat Earth equations of motion:

1. Flight path angle remains small, thus $\cos \gamma_S \approx 1$ and $\sin \gamma_S \approx \gamma_S$.

2. Thrust is zero, $T = 0$.

3. Drag is the dominate deceleration term, $D_S \gg mg \sin \gamma_S$. The term $mg \sin \gamma_S$ is the component of vehicle weight in the velocity direction. This component is assumed small compared to the drag D_S since the flight path angle is small.

4. Tangential centripetal acceleration is zero, same affect as being at the equator, $\phi_S = 0$.

5. Change from initial to final altitude is negligible compared to the radius of the Earth, $r \approx r_0$

6. Initial gravity remains constant, $g \approx g_0$.

7. Horizontal motion is only Northing y and Easting x from initial position, thus $\dot{x}_F = V_F \cos \theta_F$ and $\dot{y}_F = V_F \sin \theta_F$.

The constants r_0 and g_0 are initial altitude and gravity, respectively. The new variable θ_F within the assumptions is no longer longitude, it is heading angle measured counterclockwise from East, thus analogous to ψ_S if at the equator. Let $h_F = r - R_\oplus$ where $R_\oplus$ is the radius of the Earth, thus $\dot{h}_F = \dot{r}$. Also, the subscript F refers to flat to alleviating confusion with previous variables. Lastly, the subscript on certain variables are changed for consistency; $V_F = V_S$, $L_F = L_S$, $D_F = D_S$, and $\gamma_F = \gamma_S$. The following are the dimensional flat Earth equations of motion with the assumptions applied:

$$\dot{x}_F = V_F \cos \theta_F \tag{B.3a}$$

$$\dot{y}_F = V_F \sin \theta_F \tag{B.3b}$$

$$\dot{h}_F = V_F \gamma_F \tag{B.3c}$$

$$\dot{V}_F = -\frac{D_F}{m} \tag{B.3d}$$

$$\dot{\gamma}_F = \frac{L_F}{mV_F} \cos \sigma - \frac{g_0}{V_F} + \frac{V_F}{r_0} \tag{B.3e}$$

$$\dot{\theta}_F = \frac{L_F}{mV_F} \sin \sigma \tag{B.3f}$$

B.4   Nondimensionalization

The dynamics, controls, and path constraints must be nondimensionalized. The dynamics and path constraint nondimensionalization is covered in the following sections, and the controls σ and c_ℓ are already nondimensional.

B.4.1   Equations of Motion.   The initial dimensional radius and gravity are r_0 and g_0, respectively [41; 109]. From [111] $h = 0$ is defined as the initial nondimensional altitude; therefore, $h = (r_S - r_0)/r_0$ or $h = (h_F - h_{F0})/r_0$. The atmospheric density ρ is assumed to be a simple exponential, i.e. not broken up into multiple layers. The density is ρ_{sl} at the surface ($r_S = R_\oplus$), and is ρ_0 at the initial altitude $h = 0$, thus:

$$\rho = \rho_{sl}e^{-\beta h_F} = \rho_{sl}e^{-\beta(r_0\,h + h_{F0})} = \rho_{sl}e^{-\beta(r_0 - R_\oplus)}e^{-\beta r_0 h} \tag{B.4}$$

$$\rho_0 = \rho_{sl}e^{-\beta h_{F0}} \text{ or } \rho_{sl} = \rho_0 e^{\beta h_{F0}} \tag{B.5}$$

$$\rho = \rho_0 e^{\beta h_{F0}}e^{-\beta h_F} = \rho_0 e^{-\beta(h_F - h_{F0})} = \rho_0 e^{-\beta r_0 h} \tag{B.6}$$

The density decay constant is $\beta = 0.14 \text{ km}^{-1}$ for this research. The next step is to nondimensionalize L_F and D_F. Let nondimensional V be defined as $V = V_F/\sqrt{g_0 r_0}$ [41; 109]. This normalization factor $\sqrt{g_0 r_0}$ represents the circular orbit speed at the initial conditions. From Appendix A $c_\ell = C_L/C_L^*$, $E^* = C_L^*/C_D^*$, $C_D^* = 2C_{D_0}$, and $C_L^* = \sqrt{C_{D_0}/K}$ gives:

$$C_L = c_\ell\, C_L^* \tag{B.7a}$$

$$
\begin{aligned}
C_D &= C_{D_0} + KC_L^2 \\
&= C_D^*/2 + Kc_\ell^2\,(C_L^*)^2 \\
&= C_D^*/2 + Kc_\ell^2\,\frac{C_{D_0}}{K} \\
&= C_D^*/2 + c_\ell^2\,C_D^*/2 \\
&= C_D^*(1 + c_\ell^2)/2 \\
&= \frac{C_L^*(1 + c_\ell^2)}{2E^*} \tag{B.7b}
\end{aligned}
$$

Using Equations (B.6) and (B.7) in the equations for lift and drag:

$$
\begin{aligned}
L_F &= \frac{1}{2}\rho V_F^2 C_L S_{ref} \\
&= \frac{1}{2}\rho_0 e^{-\beta r_0 h} g_0 r_0 V^2 c_\ell C_L^* S_{ref} \tag{B.8} \\
D_F &= \frac{1}{2}\rho V_F^2 C_D S_{ref} \\
&= \frac{1}{2}\rho_0 e^{-\beta r_0 h} g_0 r_0 V^2 \frac{C_L^*(1+c_\ell^2)}{2E^*} S_{ref} \tag{B.9}
\end{aligned}
$$

The constant S_{ref} is vehicle dependent reference area, specified in Table A.1 for the CAV. The nondimensional constant $B = (\rho_0 r_0 S_{ref} C_L^*)/(2m)$ is used to simplify the first terms in Equations B.3d, B.3e, and B.3f. In computing B the variables ρ_0, r_0, S_{ref}, and m must all be expressed in consistent units in order to cancel out the dimensionality. The term D_F/m is an acceleration, and the term $L_F/(mV_F)$ has units of 1/time.

$$
\begin{aligned}
\frac{D_F}{m} &= \frac{1/2\,\rho_0 e^{-\beta r_0 h} g_0 r_0 V^2 C_L^*(1+c_\ell^2)}{2E^* m} S_{ref} \\
&= \frac{BV^2 e^{-\beta r_0 h}(1+c_\ell^2)}{2E^*} g_0 \tag{B.10a} \\
\frac{L_F}{mV_F} &= \frac{\rho_0 e^{-\beta r_0 h} g_0 r_0 V^2 c_\ell C_L^* S_{ref}}{2mV\sqrt{g_0 r_0}} \\
&= BV e^{-\beta r_0 h} c_\ell \sqrt{\frac{g_0}{r_0}} \tag{B.10b}
\end{aligned}
$$

$$\tag{B.10c}$$

Substituting the quantities from Equation (B.10) into Equation (B.3) leads to the dimensional equations below; however, they are now in terms of the nondimensional

variables V, h, and θ:

$$\dot{x}_F = V\sqrt{g_0 r_0}\cos\theta \tag{B.11a}$$

$$\dot{y}_F = V\sqrt{g_0 r_0}\sin\theta \tag{B.11b}$$

$$\dot{h}_F = V\sqrt{g_0 r_0}\gamma \tag{B.11c}$$

$$\dot{V}_F = -\frac{BV^2 e^{-\beta r_0 h}\left(1 + c_\ell^2\right)}{2E^*}g_0 \tag{B.11d}$$

$$\dot{\gamma}_F = BVe^{-\beta r_0 h}c_\ell\cos\sigma\sqrt{\frac{g_0}{r_0}} - \frac{g_0}{V\sqrt{g_0 r_0}} + \frac{V\sqrt{g_0 r_0}}{r_0} \tag{B.11e}$$

$$\dot{\theta}_F = BVe^{-\beta r_0 h}c_\ell\sin\sigma\sqrt{\frac{g_0}{r_0}} \tag{B.11f}$$

The constants g_0 and r_0 are used to nondimensionalize Equation (B.3), using length/r_0, acceleration/g_0, speed/$\sqrt{g_0 r_0}$, and time/$\sqrt{r_0/g_0}$. This normalization factor $\sqrt{g_0 r_0}$ represents the circular orbit speed at the initial conditions. The angles remain in radians. The derivatives as a function of states and time must also be normalized, e.g. dx/dt must be multiplied by $1/r_0$ to normalize the state and $\sqrt{r_0/g_0}$ to normalize time in the denominator, which combines to $1/\sqrt{g_0 r_0}$. Using a similar technique for the remaining derivatives in Equation (B.11) leads to the following nondimensional equations of motion used in this research:

$$\dot{x} = V\cos\theta \tag{B.12a}$$

$$\dot{y} = V\sin\theta \tag{B.12b}$$

$$\dot{h} = V\gamma \tag{B.12c}$$

$$\dot{V} = -\frac{BV^2 e^{-\beta r_0 h}\left(1 + c_\ell^2\right)}{2E^*} \tag{B.12d}$$

$$\dot{\gamma} = BVe^{-\beta r_0 h}c_\ell\cos\sigma - \frac{1}{V} + V \tag{B.12e}$$

$$\dot{\theta} = BVe^{-\beta r_0 h}c_\ell\sin\sigma \tag{B.12f}$$

The product βr_0 is dimensionless and is maintained as two variables to be consistent with previous research [41; 109; 110]. For example, for typical values of $\beta = 0.14$ km^{-1} and $r_0 = 6500$ km the product equals 910.

98

B.4.2 Path Constraint. The heating constraint is nondimensionalized using the maximum allowable heating; however, it must be expressed in terms of the nondimensional units in order to be calculated. Equation 36 uses V_{dim} to generically refer to a dimensional variable such as V_F. Using Equation (B.4), the dimensional heat rate at stagnation is [15; 41; 109; 114]:

$$
\begin{aligned}
\dot{q}_{s\,dim} &= \frac{k_{dim}}{\sqrt{r_{\text{nose}}}} \left(\frac{\rho}{\rho_{sl}}\right)^{1/2} \left(\frac{V_F}{\sqrt{g_0\, r_0}}\right)^3 \\
&= \frac{k_{dim}}{\sqrt{r_{\text{nose}}}} \left(e^{-\beta(r_0-R_\oplus)}e^{-\beta r_0 h}\right)^{1/2} (V)^3 \\
&= \frac{k_{dim}e^{-\beta(r_0-R_\oplus)/2}}{\sqrt{r_{\text{nose}}}} \left(e^{-\beta r_0 h}\right)^{1/2} (V)^3
\end{aligned}
\tag{B.13}
$$

The constant k_{dim} is based on the heating model, and r_{nose} is the vehicle nose radius. Let $K = \dfrac{k_{dim}e^{-\beta(r_0-R_\oplus)/2}}{\sqrt{r_{\text{nose}}}\dot{q}_{smax}}$ complete the expression of the nondimensional heating constraint, as found in Chapter III as Equation (37) repeated below:

$$
\dot{q}_s = Ke^{-\beta r_0 h/2}V^3
\tag{B.14}
$$

B.4.3 2-D Equations of Motion. The two-dimensional equations of motion are a subset of the dimensional flat Earth equations in Equation (B.11). Since the altitude is constant, $\dot{h} = 0$ and $\dot{\gamma} = 0$. Also, the vertical component of lift must be equal to the vehicle weight, thus $L_F = mg/\cos\sigma$. The dimensional acceleration $\dot{V}_F$ is simply represented as a_{dim}. The 2-D dimensional equations of motion are thus:

$$
\dot{x}_F = V_F \cos\theta_F
\tag{B.15a}
$$

$$
\dot{y}_F = V_F \sin\theta_F
\tag{B.15b}
$$

$$
\dot{\theta}_F = \frac{g}{V_F}\tan\sigma
\tag{B.15c}
$$

$$
\dot{V}_F = a_{dim}
\tag{B.15d}
$$

At constant altitude, g in Equation (B.15c) remains equal to the initial gravity defined as g_0. Let the normalized control u be defined as:

$$u = \frac{\tan \sigma}{\tan \sigma_{max}} \tag{B.16}$$

Using this definition for u and the same nondimensionalization variables as in Section B.4.1, the nondimensional 2-D (HCV) equations of motion are:

$$\dot{x} = V \cos \theta \tag{B.17a}$$

$$\dot{y} = V \sin \theta \tag{B.17b}$$

$$\dot{\theta} = \frac{\tan \sigma_{max}}{V} u \tag{B.17c}$$

$$\dot{V} = a \tag{B.17d}$$

These are the equations of motion defined in Section 3.3.5.

Appendix C. Shortest Path

An extension of [23] is to include a secondary point, thus optimizing a path from an initial point, through an intermediate waypoint, and onto a final destination. A solution to the two point problem in given first, it will then be applied to solve the three point problem.

C.1 Shortest Path between Two Points

The following are specified: an initial coordinate, $[x_0, y_0]$, an initial heading, θ_0, and a final coordinate $[x_f, y_f]$. The time history within the initial turn is from $0 \leq t \leq T_a$ where T_a is the total time in the turn:

$$
\begin{aligned}
x(t) &= x_0 - \frac{1}{u} \sin \theta_0 + \frac{1}{u} \sin(\theta_0 + ut) \\
y(t) &= y_0 + \frac{1}{u} \cos \theta_0 - \frac{1}{u} \cos(\theta_0 + ut)
\end{aligned}
$$

Following the turn is a straight leg lasting T_b, thus time is $T_a \leq t \leq (T_a + T_b)$:

$$
\begin{aligned}
x(t) &= x(T_a) + \cos \theta_f (t - T_a) \\
y(t) &= y(T_a) + \sin \theta_f (t - T_a)
\end{aligned}
$$

So,

$$
x_f = x_0 - \frac{1}{u} \sin \theta_0 + \frac{1}{u} \sin \theta_f + \cos \theta_f T_b \tag{C.1a}
$$

$$
y_f = y_0 + \frac{1}{u} \cos \theta_0 - \frac{1}{u} \cos \theta_f + \sin \theta_f T_b \tag{C.1b}
$$

Multiply Equation (C.1a) by $\sin \theta_f$ and subtract Equation (C.1b) multiplied by $\cos \theta_f$ to get:

$$
(x_f - x_0 + \frac{1}{u} \sin \theta_0) \sin \theta_f - (y_f - y_0 + \frac{1}{u} \cos \theta_0) \cos \theta_f = \frac{1}{u} \tag{C.2}
$$
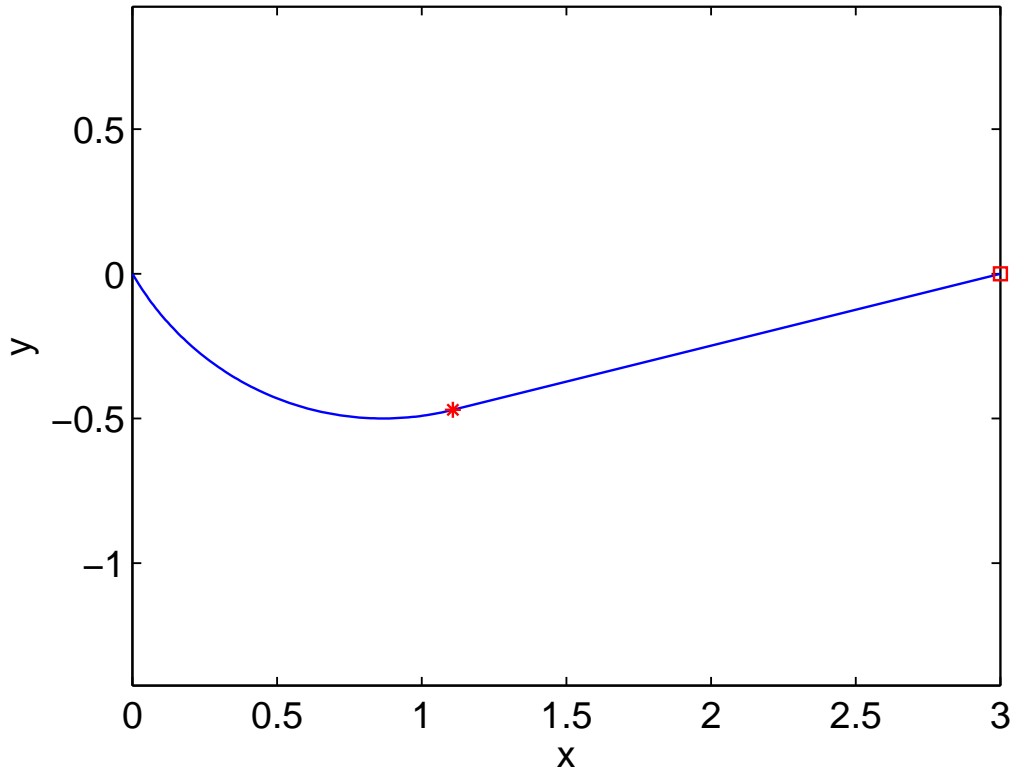
Figure C.1:   Two Point Turn from Initial Coordinate and Heading

Let the following be used as shorthand notation:

$$a = x_f - x_0 + \frac{1}{u} \sin \theta_0 \tag{C.3}$$

$$b = -\left(y_f - y_0 - \frac{1}{u} \cos \theta_0 \right) \tag{C.4}$$

$$c = \frac{1}{u} \tag{C.5}$$

This shortens to $a \sin \theta_f + b \cos \theta_f = c$, thus combining the right hand side gives:

$$\sqrt{a^2 + b^2} \sin \left(\theta_f + \tan^{-1} \left(\frac{b}{a} \right) \right) = c \tag{C.6}$$

So the final heading (θ_f), time along initial arc (T_a), straight leg time (T_b), and total time (T_t) are:

$$\theta_f = \sin^{-1}\left(\frac{c}{\sqrt{a^2+b^2}}\right) - \mathrm{atan2}\,(b,a) \tag{C.7}$$

$$T_a = \frac{\theta_f - \theta_0}{u} \tag{C.8}$$

$$T_b = a\cos\theta_f - b\sin\theta_f \tag{C.9}$$

$$T_t = T_a + T_b \tag{C.10}$$

C.2   Derive Midpoint as Optimal Waypoint Position

For the constant speed case, the optimal turn radius orientation is stated as the waypoint occurring halfway between the initiation and completion of the turn, see Figure 14(c) in Section 4.2. The geometric derivation of $\chi = 0$ is rather laborious; however, the following are two alternative derivations.

Dynamic optimization shows a bang-level-bang solution, meaning the control is either zero or at a maximum left or right bank angle. At constant velocity, a bang-level-bang control trajectory is seen in Figure C.2. The initial, waypoint, and final points are fixed; $[x_0, y_0]$, $[x_w, y_w]$, and $[x_f, y_f]$ respectively. The initial angle θ_0 and final angle θ_f are free. The fraction of time along the turn that waypoint passage occurs is also free, and must be determined for the optimal solution. Let the waypoint passage occur at some fraction (k) between starting the turn $(k = 0)$ and exiting the turn $(k = 1)$; therefore, the heading angle at waypoint passage is $\theta_0 + k(\theta_f - \theta_0)$. For example in 4.2, Figure 14(a) is $k = 0$, Figure 14(b) is $k \approx 1/4$, Figure 14(c) is $k = 1/2$, and Figure 14(d) is $k = 1$. Let the time of flight from the initial point to the start of the turn be T_0 and the time of flight from the exit of the turn to the final point be T_f. During the level portions of flight the control is zero, and during a turn the control is ± 1 which is retained generically as u.
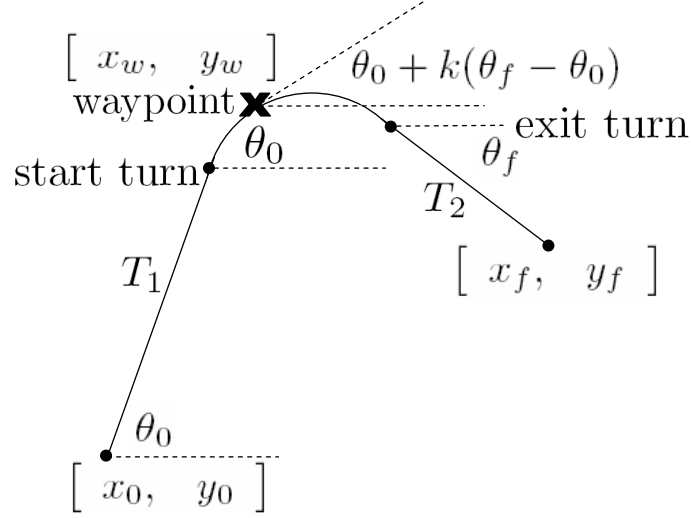
Figure C.2:   A constant speed trajectory from an initial point at a heading θ_0, passing though an intermediate waypoint, and completing the turn at a heading θ_f to intercept a final point.

C.2.1   Proof Method 1 - Equations of Motion.   The following derivation integrates the equations of motion from the initial point, to the waypoint depicted in Figure C.2, then to the final point. The nondimensional equations of motion are Equation (22) with $\dot{V} = 0$. These equations can be integrated from the initial point to the waypoint:

$$x_w = x_0 + T_0\, V \cos \theta_0 - \frac{V^2}{u \tan \sigma_{max}} \sin \theta_0 + \frac{V^2}{u \tan \sigma_{max}} \sin[\theta_0 + k(\theta_f - \theta_0)]$$

(C.11a)

$$y_w = y_0 + T_0\, V \sin \theta_0 + \frac{V^2}{u \tan \sigma_{max}} \cos \theta_0 - \frac{V^2}{u \tan \sigma_{max}} \cos[\theta_0 + k(\theta_f - \theta_0)]$$

(C.11b)

104

The second set of equations specifies the path from the waypoint to final:

$$x_f = x_w - \frac{V^2}{u \tan \sigma_{max}} \sin[\theta_0 + k(\theta_f - \theta_0)] + \frac{V^2}{u \tan \sigma_{max}} \sin \theta_f + T_f V \cos \theta_f$$

(C.12a)

$$y_f = y_w + \frac{V^2}{u \tan \sigma_{max}} \cos[\theta_0 + k(\theta_f - \theta_0)] - \frac{V^2}{u \tan \sigma_{max}} \cos \theta_f + T_f V \sin \theta_f$$

(C.12b)

The leg time T_0 is solved from Equation (C.11) and the leg time T_f is solved from Equation (C.12):

$$T_0 = \frac{1}{V} \left\{ (x_w - x_0) \cos \theta_0 + (y_w - y_0) \sin \theta_0 - \frac{V^2}{u \tan \sigma_{max}} \sin [k(\theta_f - \theta_0)] \right\}$$

$$T_f = \frac{1}{V} \left\{ (x_f - x_w) \cos \theta_f + (y_f - y_w) \sin \theta_f - \frac{V^2}{u \tan \sigma_{max}} \sin [(1 - k)(\theta_f - \theta_0)] \right\}$$

(C.13)

The time in the turn is $V(\theta_f - \theta_0)/(u \tan \sigma_{max})$, so the total cost ($J$) is the time from the initial point, through the waypoint in Figure C.2, to the final point:

$$J = T_0 + V(\theta_f - \theta_0)/(u \tan \sigma_{max}) + T_f$$

(C.14)

To minimize the cost with respect to the waypoint passage fraction (k), the partial derivative of J with respect to k must be zero, i.e. $\partial J/\partial k = 0$. Assume $\theta_f \neq \theta_0$ since that would imply a straight line or no turn. Performing the partial derivative:

$$\frac{\partial J}{\partial k} = -\frac{V}{u \tan \sigma_{max}}(\theta_f - \theta_0) \cos[k(\theta_f - \theta_0)]$$

$$+ \frac{V}{u \tan \sigma_{max}}(\theta_f - \theta_0) \cos[(1 - k)(\theta_f - \theta_0)] = 0$$

(C.15)

105

While in the turn the control u, the speed V, the maximum bank angle θ_{max}, and the term $(\theta_f - \theta_0)$ are all non-zero, so division by these terms remains valid:

$$\cos[k(\theta_f - \theta_0)] = \cos[(1 - k)(\theta_f - \theta_0)]$$
$$k(\theta_f - \theta_0) = (1 - k)(\theta_f - \theta_0)$$
$$k = 1/2$$

This shows that the optimal waypoint passage occurs halfway through the turn, which has the geometric properties shown in Figure 14(c).

C.2.2 Proof Method 2 - Dynamic Optimization. Dynamic optimization uses dynamics, terminal, and intermediate state constraints to formulate the problem. With only one waypoint, and without considering a no-fly zone, this is a simplified version of the problem generically derived in Section 3.1.1. The cost to minimize is final time (t_f) and the heading to determine for the optimal solutions is θ_w, which occurs at the waypoint passage time (t_w). The nondimensional equations of motion are Equation (22) with $\dot{V} = 0$. The terminal and waypoint constraint are ψ and N respectively:

$$\psi = \begin{bmatrix} x(t_f) - x_f \\ y(t_f) - y_f \end{bmatrix} \qquad N = \begin{bmatrix} x(t_w) - x_w \\ y(t_w) - y_w \end{bmatrix} \tag{C.16}$$

The Hamiltonian is:

$$H = \lambda_x V \cos\theta + \lambda_y V \sin\theta + \lambda_\theta \frac{\tan\sigma_{max}}{V} u$$

The following are Equations (55) and (58) without the velocity dynamics and with $\mu_S = 0$ (outside no-fly zones):

$$\dot{\lambda}_x = 0 \qquad\qquad \lambda_x(t_f) = -\cos\theta_f/V$$
$$\dot{\lambda}_y = 0 \qquad\qquad \lambda_y(t_f) = -\sin\theta_f/V$$
$$\dot{\lambda}_\theta = \lambda_x V \sin\theta - \lambda_y V \cos\theta \qquad \lambda_\theta(t_f) = 0$$

106

Equation (64) is the solution for the jump in costates. Since there is only one waypoint, $\theta_i = \theta_w$ and $t_i^+ = t_w^+$. For waypoint passage between initial and final, the control is constant over the jump, i.e. $u(t_w^-) = u(t_w^+)$. Since λ_x and λ_y are constants (excluding the jumps), $\lambda_x(t_w^+) = \lambda_x(t_f)$ and $\lambda_y(t_w^+) = \lambda_y(t_f)$. At the start of the turn (t_{si}^-), the heading is $\theta_{si} = \theta_0$. The jump in costates λ_x and λ_y are:

$$
\begin{aligned}
\pi_{nx} &= \frac{-\sin\theta_w \left(\lambda_x(t_w^+)\sin\theta_{si} - \lambda_y(t_w^+)\cos\theta_{si} \right)}{\cos(\theta_w - \theta_{si})} \\
&= \frac{-\sin\theta_w \left(-\cos\theta_f \sin\theta_0 + \sin\theta_f \cos\theta_0 \right)}{V\cos(\theta_w - \theta_0)} \\
&= -\sin\theta_w \frac{\sin(\theta_f - \theta_0)}{V\cos(\theta_w - \theta_0)} \tag{C.17a} \\
\pi_{ny} &= \frac{\cos\theta_w \left(\lambda_x(t_w^+)\sin\theta_{si} - \lambda_y(t_w^+)\cos\theta_{si} \right)}{\cos(\theta_w - \theta_{si})} \\
&= \frac{\cos\theta_w \left(-\cos\theta_f \sin\theta_0 + \sin\theta_f \cos\theta_0 \right)}{V\cos(\theta_w - \theta_0)} \\
&= \cos\theta_w \frac{\sin(\theta_f - \theta_0)}{V\cos(\theta_w - \theta_0)} \tag{C.17b}
\end{aligned}
$$

Therefore the costates behind the jump at t_w^-, going backwards in time, are:

$$
\begin{aligned}
\lambda_x(t_w^-) &= -\frac{\cos\theta_f}{V} - \sin\theta_w \frac{\sin(\theta_f - \theta_0)}{V\cos(\theta_w - \theta_0)} \\
\lambda_x(t_w^-) &= -\frac{\sin\theta_f}{V} + \cos\theta_w \frac{\sin(\theta_f - \theta_0)}{V\cos(\theta_w - \theta_0)}
\end{aligned}
$$

Substitute these values into the Hamiltonian at the start of the turn (t_{si}^-), where $u(t_{si}^-) = 0$, $\dot{x}(t_{si}^-) = V\cos\theta_0$, and $\dot{y}(t_{si}^-) = V\sin\theta_0$; then set the Hamiltonian equal to -1. The Hamiltonian and the states are continuous so the plus and minus superscript

107

notation is unnecessary:

$$H(t_{si}) = \left(-\cos\theta_f - \sin\theta_w \frac{\sin(\theta_f - \theta_0)}{\cos(\theta_w - \theta_0)} \right)\cos\theta_0$$
$$+ \left(-\sin\theta_f + \cos\theta_w \frac{\sin(\theta_f - \theta_0)}{\cos(\theta_w - \theta_0)} \right)\sin\theta_0 = -1$$
$$-(\cos\theta_f\cos\theta_0 + \sin\theta_f\sin\theta_0) - (\sin\theta_w\cos\theta_0 - \cos\theta_w\sin\theta_0)\frac{\sin(\theta_f - \theta_0)}{\cos(\theta_w - \theta_0)} = -1$$
$$\cos(\theta_f - \theta_0)\cos(\theta_w - \theta_0) + \sin(\theta_f - \theta_0)\sin(\theta_w - \theta_0) = \cos(\theta_w - \theta_0)$$
$$\cos(\theta_f - \theta_0 - \theta_w + \theta_0) = \cos(\theta_w - \theta_0)$$
$$\theta_f - \theta_w = \theta_w - \theta_0$$
$$\theta_w = \frac{\theta_f + \theta_0}{2} = \theta_0 + \frac{1}{2}(\theta_f - \theta_0) \tag{C.18}$$

Since θ_w is of the form $\theta_0 + k(\theta_f - \theta_0)$, the optimal solution is again shown to be $k = 1/2$. This solution for θ_w can be plugged back into the jump equation, Equation (C.17), to solve for π_{nx} and π_{ny}. With a solution for the jumps, the value of the costates at t_w^- can be computed. For one waypoint this provides the value of the initial costates at t_0; $\lambda_x(t_0) = -\cos\theta_0/V$, $\lambda_y(t_0) = -\sin\theta_0/V$, and $\lambda_\theta(t_0) = 0$.
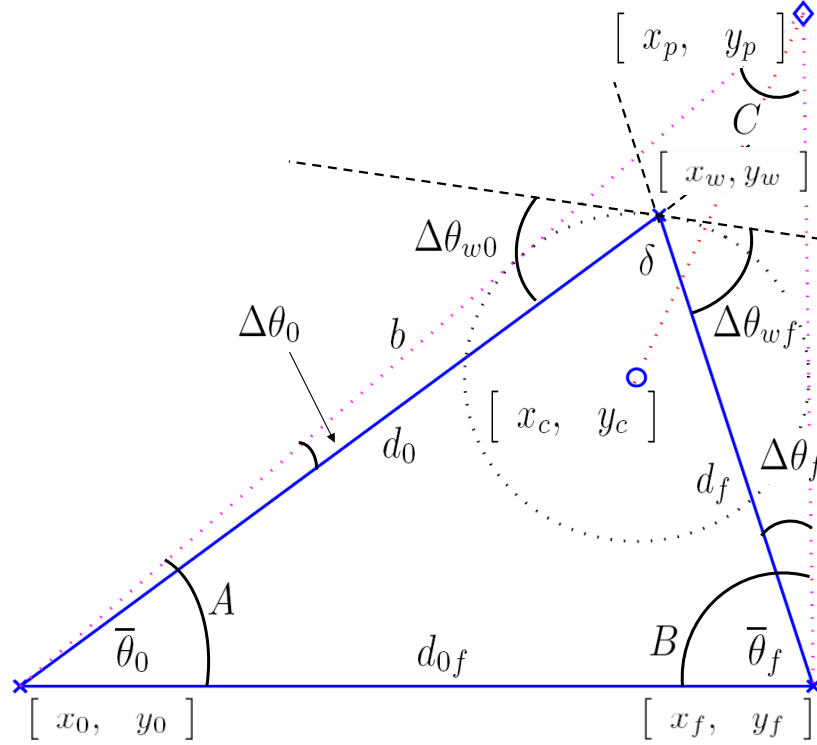
Figure D.1:   Nomenclature for Iterative Solution

The data to be given are initial position $[x_0,\ y_0]$, intermediate waypoint $[x_w,\ y_w]$, final position $[x_f,\ y_f]$, radius (r), and direction of turn (u). Control (u) positive implies increasing heading angle (θ) thus counterclockwise rotation. Heading is measured counterclockwise from the x-axis. The given points will be normalized by the radius so radius (r) is assumed 1. Also, the given geometry is rotated such that the x-axis is aligned along the initial and final points. Lastly, the problem is translated such that the initial point is $[0,\ 0]$. This can all be reversed following the computations to recuperate the original configuration.

In addition to the given values, other computed values are fixed. The angles within the triangle between initial and the waypoint $(\bar{\theta}_0)$ and final and the waypoint $(\bar{\theta}_f)$, and the distances from initial to waypoint (d_0), waypoint to final (d_f), and initial to final (d_{0f}). The size of $\Delta\theta_0$ will be adjusted until the optimal orientation is

achieved. The iteration loop is initiated with a guess. A good initial guess is described at the end of the section.

$$\Delta\theta_0 = \text{guess} > 0$$

First, the relationship between the change in $\Delta\theta_0$ and the corresponding value of $\Delta\theta_{w0}$ must be derived. Figure D.2 shows the initial leg; however, it is translated to start at $[0, 0]$ and rotated to be aligned with the x-axis. The location of the center of the circle rotated $\Delta\theta_{w0}$ about the waypoint is:
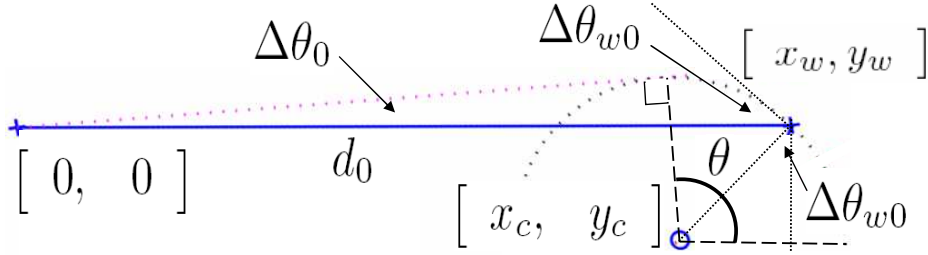
Figure D.2:   Initial Leg of Trajectory

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} d_0 - \cos(\frac{\pi}{2} - \Delta\theta_{w0}) \\ -\sin(\frac{\pi}{2} - \Delta\theta_{w0}) \end{bmatrix} = \begin{bmatrix} d_0 - \cos(\Delta\theta_{w0} - \frac{\pi}{2}) \\ \sin(\Delta\theta_{w0} - \frac{\pi}{2}) \end{bmatrix} = \begin{bmatrix} d_0 - \sin\Delta\theta_{w0} \\ -\cos\Delta\theta_{w0} \end{bmatrix}$$

$$(D.1)$$

Originating from the center of the circle $[x_c, y_c]$, at a perpendicular distance of radius $r = 1$, there exist an angle θ such that the line passes through the initial point $[0, 0]$. The normal form of the straight line equation for any position $[x, y]$ is:

$$(x - x_c)\cos\theta + (y - y_c)\sin\theta = 1 \tag{D.2}$$

In Figure D.2 the right triangle containing $\Delta\theta_0$ has the complementary angle equal to $\frac{\pi}{2} - \Delta\theta_0$. Since, it is on the x-axis which is parallel to the horizontal line at the center

of the circle then it is also equal to $\pi - \theta$. Combining these relationships provides:

$$\frac{\pi}{2} - \Delta\theta_0 \;=\; \pi - \theta \tag{D.3}$$

$$\theta \;=\; \Delta\theta_0 + \frac{\pi}{2} \tag{D.4}$$

Now substituting $[x,\ y] = [0,\ 0]$, Equations (D.1) and (D.4) into Equation (D.2) produces:

$$(0 - d_0 + \sin\Delta\theta_{w0})\cos(\Delta\theta_0 + \frac{\pi}{2}) + (0 + \cos\Delta\theta_{w0})\sin(\Delta\theta_0 + \frac{\pi}{2}) \;=\; 1 \tag{D.5}$$

$$(d_0 - \sin\Delta\theta_{w0})\sin\Delta\theta_0 + \cos\Delta\theta_{w0}\cos\Delta\theta_0 \;=\; 1 \tag{D.6}$$

Equation (D.6) establishes the relationship between $\Delta\theta_0$ and $\Delta\theta_{w0}$. Since the iterations will start with $\Delta\theta_0$ the expression for $\Delta\theta_{w0}$ in terms of $\Delta\theta_0$ must be found:

$$d_0 \sin\Delta\theta_0 + \cos\Delta\theta_{w0}\cos\Delta\theta_0 - \sin\Delta\theta_{w0}\sin\Delta\theta_0 \;=\; 1$$

$$\cos(\Delta\theta_{w0} + \Delta\theta_0) \;=\; 1 - d_0 \sin\Delta\theta_0$$

$$\boxed{\Delta\theta_{w0} = \cos^{-1}(1 - d_0 \sin\Delta\theta_0) - \Delta\theta_0} \tag{D.7}$$

In Figure D.1, at the waypoint, the relationship between $\Delta\theta_{w0}$ and $\Delta\theta_{wf}$ is defined as:

$$\Delta\theta_{w0} + \Delta\theta_{wf} + \delta \;=\; \pi$$

$$\delta \;=\; \pi - \bar\theta_0 - \bar\theta_f$$

$$\Delta\theta_{w0} + \Delta\theta_{wf} + \pi - \bar\theta_0 - \bar\theta_f \;=\; \pi$$

$$\boxed{\Delta\theta_{wf} = \bar\theta_0 + \bar\theta_f - \Delta\theta_{w0}} \tag{D.8}$$

Now the process needs to be reversed for the final leg. The same setup as in Figure D.2 can be used, except all of the 0 subscripts become f subscripts. The terms will

be regrouped differently in order to get $\Delta\theta_f$ in terms of the computed $\Delta\theta_{wf}$:

$$
\begin{bmatrix} x_{c_f} \\ y_{c_f} \end{bmatrix} = \begin{bmatrix} d_f - \sin\Delta\theta_{wf} \\ -\cos\Delta\theta_{wf} \end{bmatrix}
$$

$$
x_{c_f}\sin\Delta\theta_f - y_{c_f}\cos\Delta\theta_f = 1
$$

$$
d_c = \sqrt{x_{c_f}^2 + y_{c_f}^2}
$$

$$
d_c \sin\left(\Delta\theta_f + \tan^{-1}\left(\frac{-y_{c_f}}{x_{c_f}}\right)\right) = 1
$$

$$
\sin\left(\Delta\theta_f - \tan^{-1}\left(\frac{y_{c_f}}{x_{c_f}}\right)\right) = \frac{1}{d_c}
$$

$$
\boxed{\Delta\theta_f = \sin^{-1}\left(\frac{1}{d_c}\right) + \tan^{-1}\left(\frac{y_{c_f}}{x_{c_f}}\right)}
\tag{D.9}
$$

Using the Law of Sines, the exterior triangle can be computed from the computed interior angles and known side length d_{0f}:

$$
A = \bar{\theta}_0 + \Delta\theta_0
$$

$$
B = \bar{\theta}_f + \Delta\theta_f
$$

$$
C = \pi - A - B
$$

$$
\frac{b}{\sin B} = \frac{d_{0f}}{\sin C}
$$

$$
b = \frac{d_{0f}}{\sin C}\sin B
$$

$$
\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} b\cos A \\ b\sin A \end{bmatrix}
$$

The optimal solution occurs when the line from the center of the circle $[x_c,\ y_c]$ to the peak of the triangle $[x_p,\ y_p]$ passes through the waypoint $[x_w,\ y_w]$. The center of the

circle in Equation (D.1) must be rotated to realign with the initial leg:

$$
\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \cos\bar{\theta}_0 & -\sin\bar{\theta}_0 \\ \sin\bar{\theta}_0 & \cos\bar{\theta}_0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix}
\tag{D.10}
$$

$$
\begin{aligned}
\theta_{c_p} &= \tan^{-1}\left(\frac{y_p - y_c}{x_p - x_c}\right) \\
\theta_{c_w} &= \tan^{-1}\left(\frac{y_w - y_c}{x_w - x_c}\right) \\
\text{error} &= \theta_{c_w} - \theta_{c_p}
\end{aligned}
\tag{D.11}
$$

The iteration continues with a new value for $\Delta\theta_0$

$$
\Delta\theta_0 = \Delta\theta_0 + \text{error}
\tag{D.12}
$$

This iterative loop was compared to satisfying Equation (D.11) using `Matlab`®'s `fsolve`. Accuracy was driven to 1e-10 radians. Although the loop took 20 to 30 iterations compared to only 2 to 4 with `fsolve`, the iteration loop took orders of magnitude less cpu time. The other advantage is `fsolve` sometime took excessively large jumps in $\Delta\theta_0$ which created infeasible constructions, i.e. trigonometric function errors. The final step is to convert the iteration results to vehicle headings. For this configuration:

$$
\begin{aligned}
\theta_0 &= A \\
\theta_f &= -B \\
\theta_w &= \frac{\theta_f + \theta_0}{2}
\end{aligned}
$$

Equation (D.9) can be used to compute an initial guess for $\Delta\theta_0$. A good guess is that the radius will be aligned with the bisect of angle δ. In many cases this initial guess is within a degree of the final answer; therefore, a non-iterative, near optimal, solution

exists to minimize computational time:

$$\Delta\theta_{w0_{\text{guess}}} = \frac{\pi}{2} - \frac{\delta}{2} = \frac{\bar{\theta}_0 + \bar{\theta}_f}{2}$$

$$\begin{bmatrix} x_{c_{\text{guess}}} \\ y_{c_{\text{guess}}} \end{bmatrix} = \begin{bmatrix} d_o - \sin\Delta\theta_{w0_{\text{guess}}} \\ -\cos\Delta\theta_{w0_{\text{guess}}} \end{bmatrix}$$

$$d_{c_{\text{guess}}} = \sqrt{x_{c_{\text{guess}}}^2 + y_{c_{\text{guess}}}^2}$$

$$\Delta\theta_{0_{\text{guess}}} = \sin^{-1}\left(\frac{1}{d_{c_{\text{guess}}}}\right) + \tan^{-1}\left(\frac{y_{c_{\text{guess}}}}{x_{c_{\text{guess}}}}\right)$$

Appendix E.   Pseudospectral Method

The purpose of pseudospectral methods is to approximate the continuous solution to a set of differential equations using polynomial interpolation through discrete points or nodes. The motivation is to avoid sequential integration which can lead to divergence and may prohibit determining a solution. Legendre or Chebyshev polynomials, as shown in Figure E.1 [31], may be used to satisfy the differential equations at a discrete number of nodes N. Computing the solution at the nodes is also termed collocation [11; 31; 50; 99]. These collocation techniques satisfy all the nodes simultaneously, thus avoiding the pitfalls of integration, especially the forward and backward integration within the shooting method. The discretization and node placement is fundamental to the various pseudospectral techniques [4; 33; 80; 115; 116]. Equally space nodes is the simplest arrangement; however, for polynomial interpolation equispacing will lead to large errors as the number of nodes is increased. This large error, especially at the endpoints, is called the Runge Phenomenon as seen in Figure E.2(a).
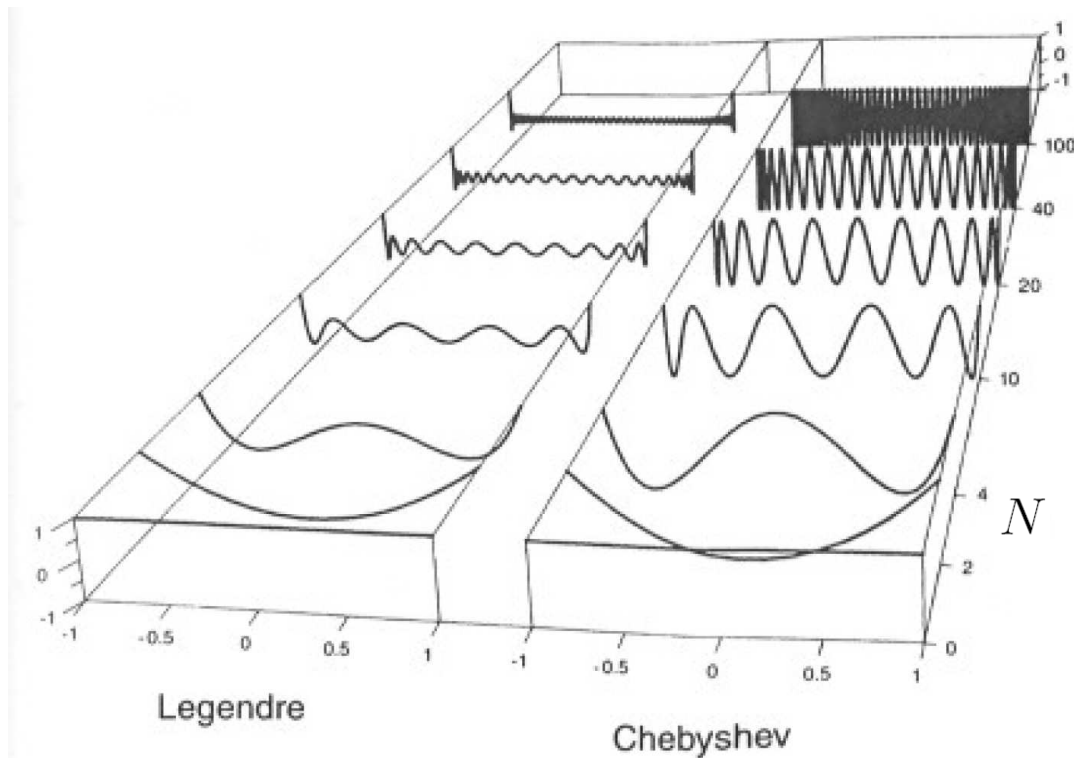
Figure E.1:   Comparison of Legendre and Chebyshev Polynomials [31]
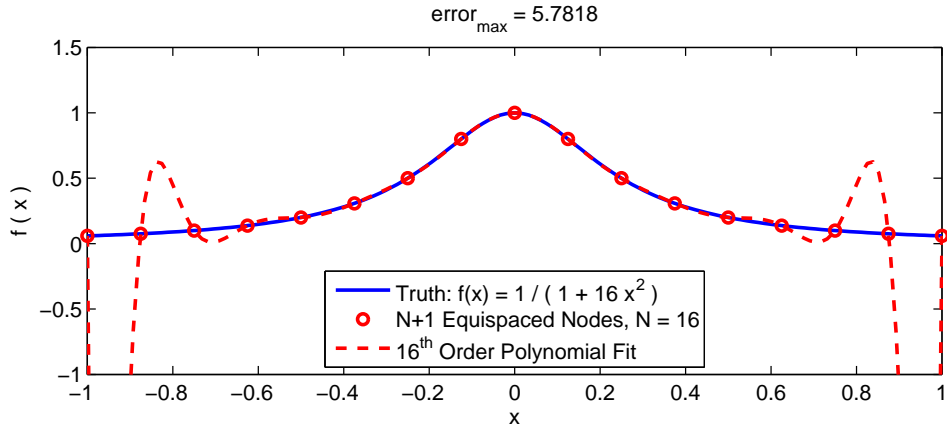
(a) Evenly Spaced Nodes
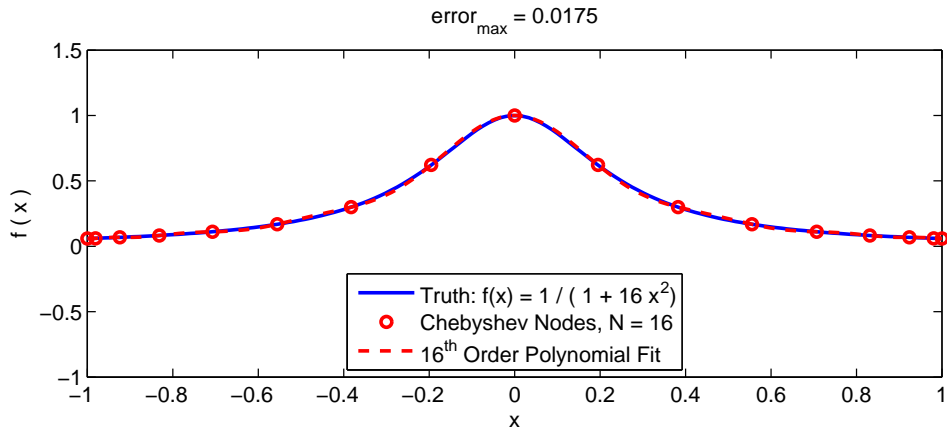
(b) Chebyshev Nodes

Figure E.2:   Runge Phenomenon is Avoided Using Chebyshev Points

The avoid these errors, the density of the nodes at the endpoints is increased using Chebyshev points. The Chebyshev point placement in Equation (E.1) is shown in Figure E.3.

$$x_j = \cos(j\pi/N), \qquad j = 0, 1, \ldots, N \tag{E.1}$$

Using Chebyshev points, the polynomial interpolation accuracy improves with an increase in the number of nodes as seen in Figure E.2(b). An important observation is that the node placement is more sparse at the middle; therefore, simply increasing the number of nodes may not produce more accurate results if the region of interest is near the midpoint. Software packages such as DIDO and GPOCS allow for regions of

more dense node placement. This is implemented using "knots" in DIDO or "phases" in GPOCS. By specifying these intermediate events, or boundary conditions, a new endpoint is created; thus, adding nodes and reducing errors.
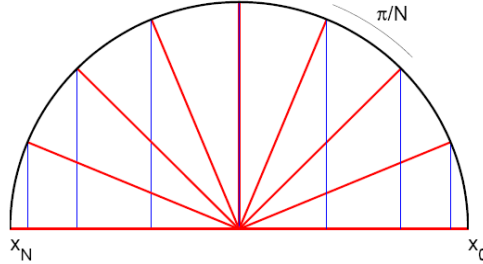
Figure E.3:   Chebyshev Points Projected onto the x-axis [99]

Increasing the number of nodes does improve accuracy; however, it may also extend the time to compute the solution. Also, it may be more efficient to compute a solution with a reduced number of nodes first, then use that solution as an initial guess for a subsequent software run with additional nodes. This dynamic meshing technique is mentioned as potential additional research in Section V.

The pseudospectral method requires that the differential equation be exactly satisfied at the "collocation" points [11:pg.12]. This may be accomplished using Chebyshev differential matrices [99:pg.54]. An example from [99:pg.54] is recreated and presented in Figure E.4. These methods use state values at discrete nodes to construct differential matrices [31], to then compute the derivative as shown in Figure E.4. The derivative matrix used to compute the results in Figure E.4 is computed using $N = 7$ and $N = 20$. This demonstrates the initial accuracy possible even with a lower number of nodes, and shows how the accuracy increasing with an increased number of nodes.

The nodes included in the solution to the differential equations may differ between software packages. DIDO uses the Legendre pseudospectral method which is based on interpolating functions on Legendre-Gauss-Lobatto (LGL) quadrature nodes [78]. This technique does include the boundary conditions whereas the GPOCS

Gauss pseudospectral technique using Legendre-Gauss nodes [4] specifically does not use the endpoints. This variation in solution technique also varies the discrete KKT multiplier mapping to the continuous costates. The Covector Mapping Principle [27] is used within DIDO; however, the Costate Mapping Theorem [4] is used within GPOCS. The placement of this mapping is shown in Figure E.5.
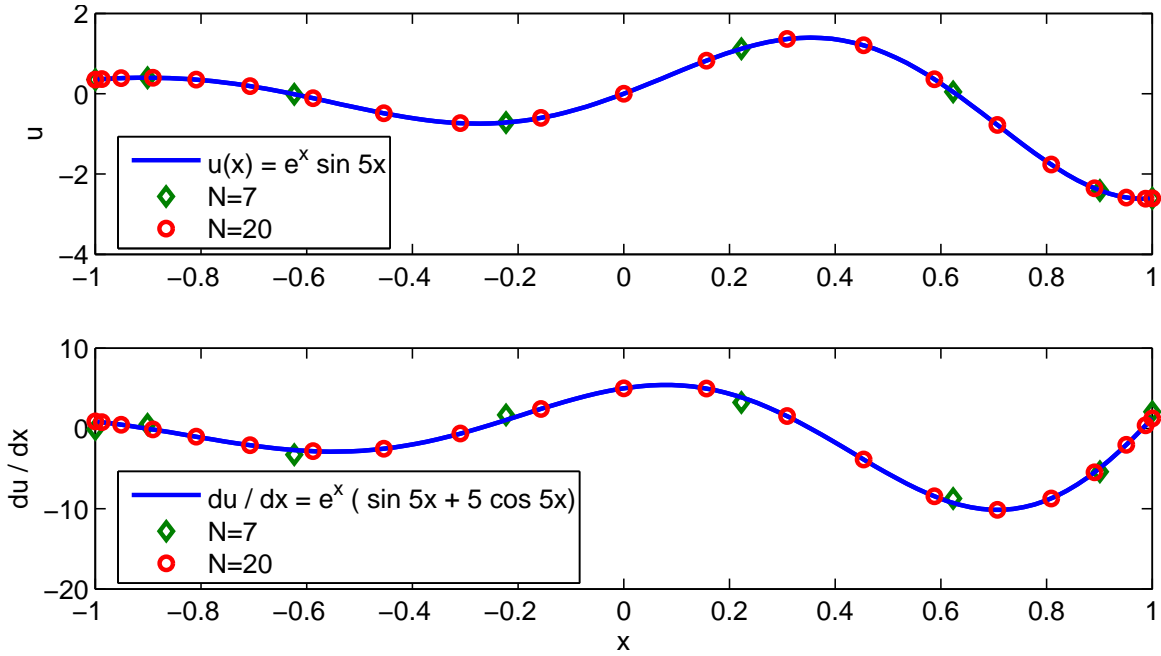
Figure E.4:   Derivative Computed Using Chebyshev Differentiation Matrices

(a) DIDO [27]

(b) GPOCS [4]
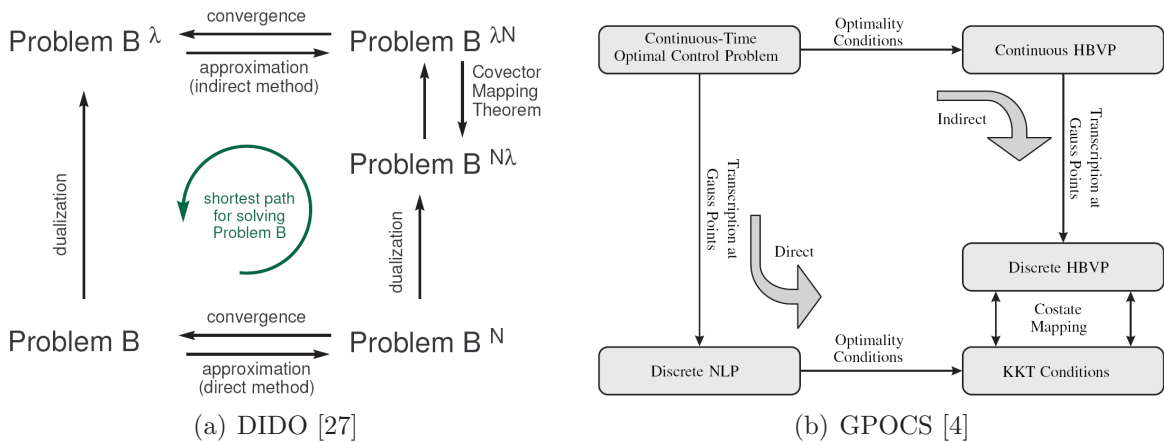
Figure E.5:   Computation of the Continuous Lagrange Multipliers

Appendix F. Software Implementation Strategies

The software coding techniques used to compute the numerical results are presented in this appendix. This is included to help any follow-on research duplicate the results presented herein; however, it not intended to replace or replicate the software's user's manual [72].

F.1   Capture Jumps in Costates

The waypoints and final altitude are specified as matching lower and upper bounds on the initial and terminal state bounds for the applicable phase, i.e. they are not computed as separate events. The no-fly zones are specified as S in Equation (26); however, to force the software to find the exact location of the discontinuities the definitions for $S = 0$ and $S^{(1)} = 0$ are specified to be satisfied at the end of the applicable phases, i.e. these are event conditions as a function of the terminal states for that phase. If higher accuracy is required to force a sharp discontinuity, instead of one that has a partially continuous corner, the $S^{(1)}$ function is multiplied by a scaling factor such as 100. The heating constraint $Q = 0$ and $Q^1(c_\ell = 0.8) \leq Q^1 \leq Q^1(c_\ell = 1.2)$ are set as events to force a discontinuity at the heating constraint contact. In looking at Equation (43) it can be seen that it is a function of the control, which is unknown; hence, the band selection of $0.8 \leq c_\ell \leq 1.2$. Again, these were tricks to force and exact solution, final implementation is much simpler by merely setting the event $\gamma = 0$. This doesn't force a jump as accurate as above, but it does provide sufficient nodes to properly characterize the jump to converge on the correct solution.

F.2   Control Multipliers

The simplest method for control limitations is to specify the upper and lower bounds, per phase. If, however, the multipliers associated with an active control constraint are of interest, the control can be input as a path constraint. The control multiplier is then contained in the output path multipliers. This is the method used to compare the numerical and analytical results within this research.

F.3   Events and Cost

There were a few techniques that provided better results; however, I did not run any formal tests. First, the minimum time cost performed better when it was specified as the terminal time of the final phase. This seemed to work better than having an integrand cost of 1. Also, be sure to use the intended variable since initial, terminal, and intermediate values are sometimes available in the same function. Typically, for events and linkages the intended variable is initial or terminal. Inserting control limitations and state constraints worked best when inserted directly as part of the state or control bounds; therefore, use the bounds instead of a separate path or event if possible.

F.4   Singularities and Results

Always ensure any singularities are avoided; therefore, set state or control bounds to avoid undefined trigonometric functions or a divide by zero. Integrating the dynamics is a beneficial way to ensure the answer is feasible. Also, ensure the maximum step size of the integrator is low enough to avoid divergence. The magnitude of the costates and the value of the Hamiltonian are valuable in identifying an optimal solution. If the costate have huge magnitudes, or the Hamiltonian does not remain constant, a non-optimal solution is likely. The Hamiltonian will only be constant if it is not explicitly a function of time.

F.5   Nodes and Iterations

It may be best to start a problem with fewer nodes and a low maximum number of major iteration steps. This technique allows for an initial look at the results to verify it is approaching an expected or reasonable solution. It also can be used to identify contact with a path constraint. Such contact may require another phase breakpoint to increase the number of nodes at that area of interest. This may also be a detriment, too few nodes with high optimality tolerance and/or feasibility tolerance can produce an "infinite" loop. The controls may be so spaced out that a small

change is held for such a long time that convergence is not possible, or at least is very time consuming. Typically this behavior can be identified by monitoring the merit function and looking for bounded oscillations.

F.6   Breakwell Example

The Breakwell problem is presented in two publications, but with different costate histories [14; 78]. The results in [78] are reproduced using GPOCS and are plotted in Figure F.1. This difference with respect to the research herein is addressed in Sections 4.3 and 4.6. This example demonstrates how to produce the same results using the analytical method and numerical method. The basis of confusion is the analytical method [14; 44] adds interior-point constraints to provide more criteria for a solution. This adds intuition into the solution; however, it is not required for the numerical results. Although these additional steps are not required, they will be implemented in the numerical method to force comparable results. This comparison will verify that even though these methods produce different costates, they are both correct, and both lead to the same optimal control and state history.

F.6.1   Analytical Derivation.   The differences in costates occur when adjoining the path constraint S, or one of its time derivatives, to the Hamiltonian. For the Breakwell problem the constraint is $x \leq \ell$, thus the path constraint is:

$$S(\mathbf{x}(t), t) = [x(t) - \ell] \leq 0 \tag{F.1}$$

For the analytical method, the next step is to define the interior-point constraint that is applied upon entering the constraint boundary at time t_1. This is derived by taking the time derivative of S until the control a appears. For this problem $a = \dot{v} = \ddot{x}$; therefore, the second time derivative contains the control, i.e. $S^{(2)} = a$. Thus, the

121

lower time derivatives become the interior-point constraint:

$$N(\mathbf{x}(t_1), t_1) = \begin{bmatrix} S \\ S^{(1)} \end{bmatrix} = \begin{bmatrix} x(t_1) - \ell \\ v(t_1) \end{bmatrix} = \mathbf{0} \tag{F.2}$$

From the definition of N there will be a jump in λ_x and λ_v, as shown in Figure F.1(b) for the analytical case. Typically, GPOCS does not compute the costates in the manner just described since this is more derivation than simply programming the path constraint S.

F.6.2 GPOCS Implementation. The GPOCS software package allows the problem to be broken up into phases. To force GPOCS to replicate the Bryson derivation, the problem is split into three phases. Since the Bryson problem setup is going to be three phases, the initial comparison is also broken into three phases, as shown in Figure F.1. This is not required for a solution, but it does provide a better comparison of the computed cost.

Figure F.1 uses S as the adjoined path constraint. For this particular numerical solution, the problem setup also requires an interior-point constraint of $S = 0$ at the end of phases 1 and 2. Thus, since S is only a function of x, there is only a jump in the λ_x costate. The next step is to replicate the Bryson results derived in Section F.6.1.

Figure F.2 uses $S^{(2)}$ as the adjoined path constraint. From the previous derivation there is also an interior-point constraint N, from Equation (F.2), at the end of phase 1, i.e. entering the path constraint boundary. The linkage between phases is programmed to force state continuity, but the costates and control may be discontinuous. The path constraint for the second phase is $S^{(2)} = a = 0$. Since the interior-point constraint N is a function of x and v, there is a jump in λ_x and λ_v, as shown in Figure F.2(b). This GPOCS problem setup replicates the analytical Bryson derivation; therefore, the GPOCS and Bryson solutions now match, as seen in Figure F.2(b).

(a) States and Control a

(b) Costates and Hamiltonian H
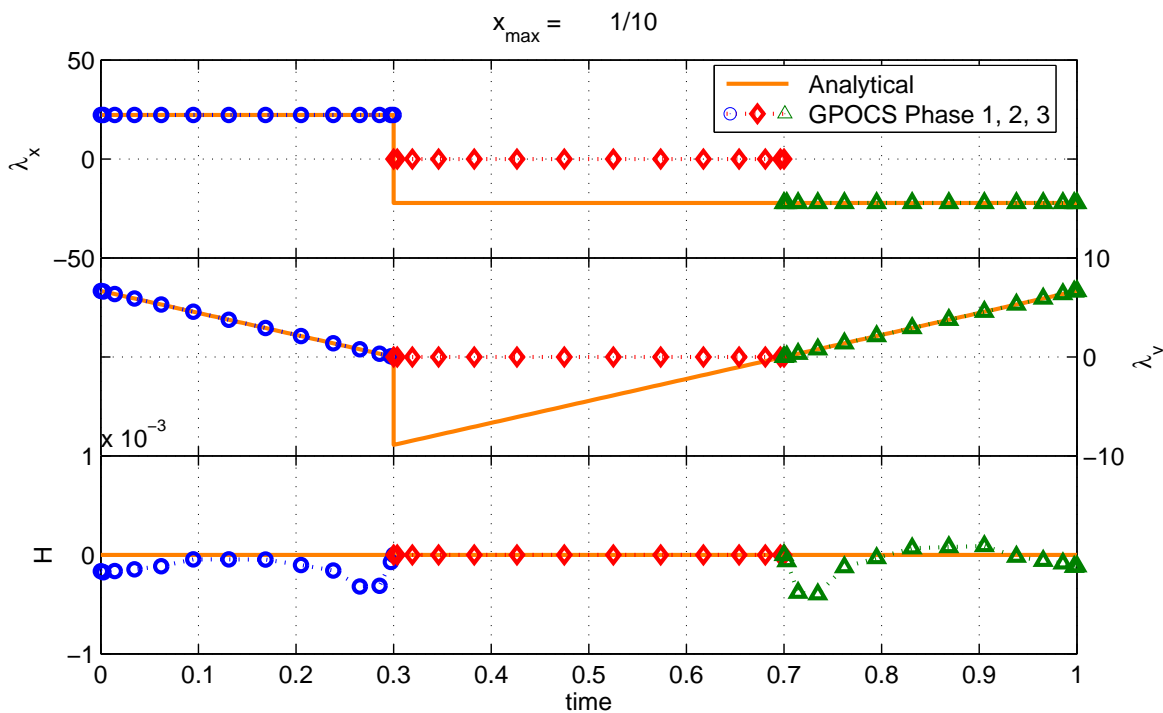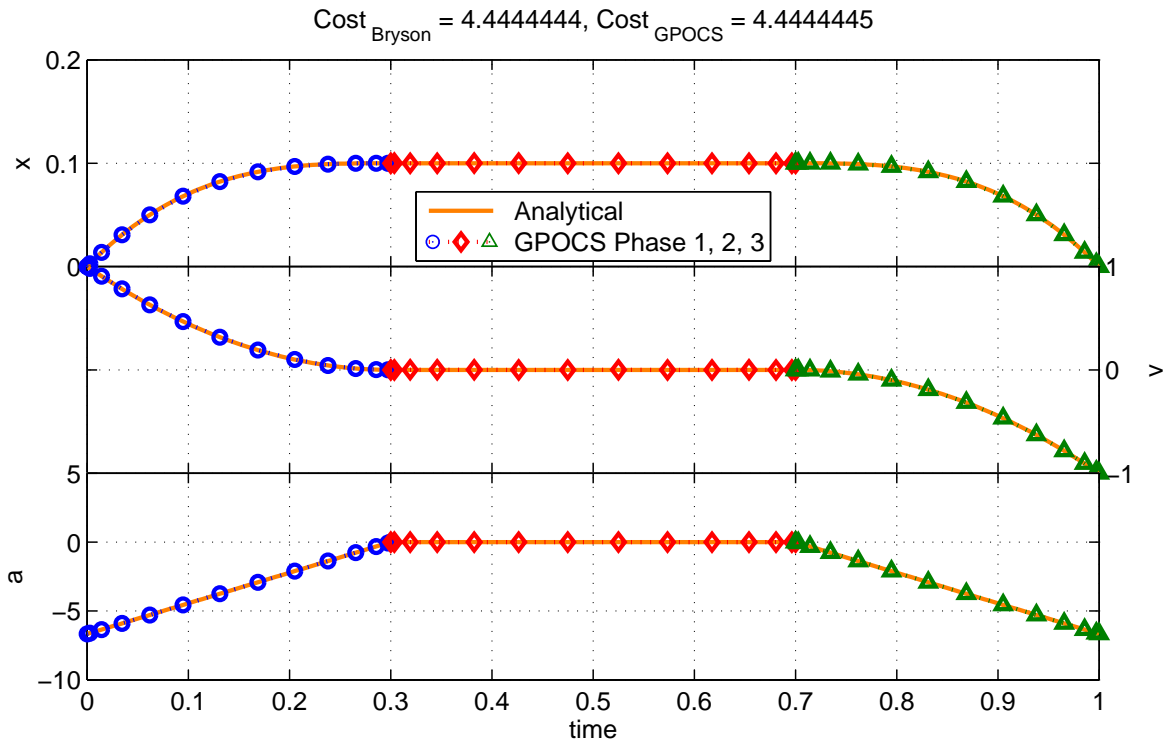
Figure F.1:   Breakwell Problem: Differences in Bryson and GPOCS Results

Cost$_{Bryson}$ = 4.4444444, Cost$_{GPOCS}$ = 4.4444445

(a) States and Control a

x_{max} = 1/10

(b) Costates and Hamiltonian H
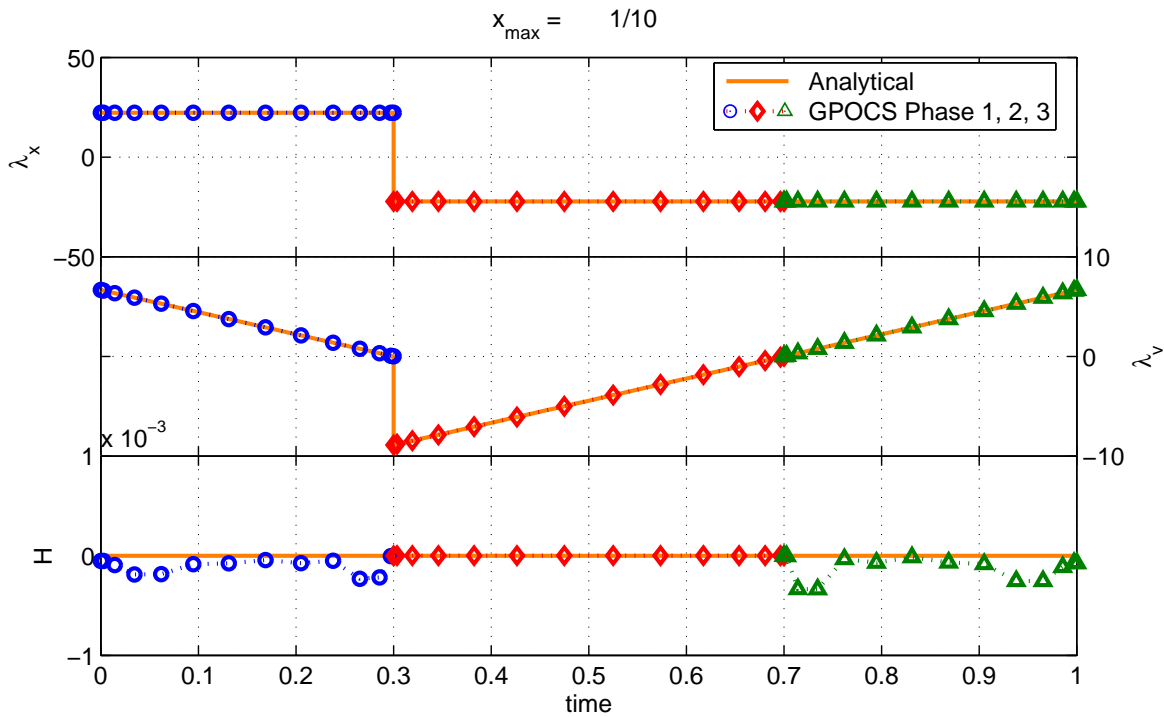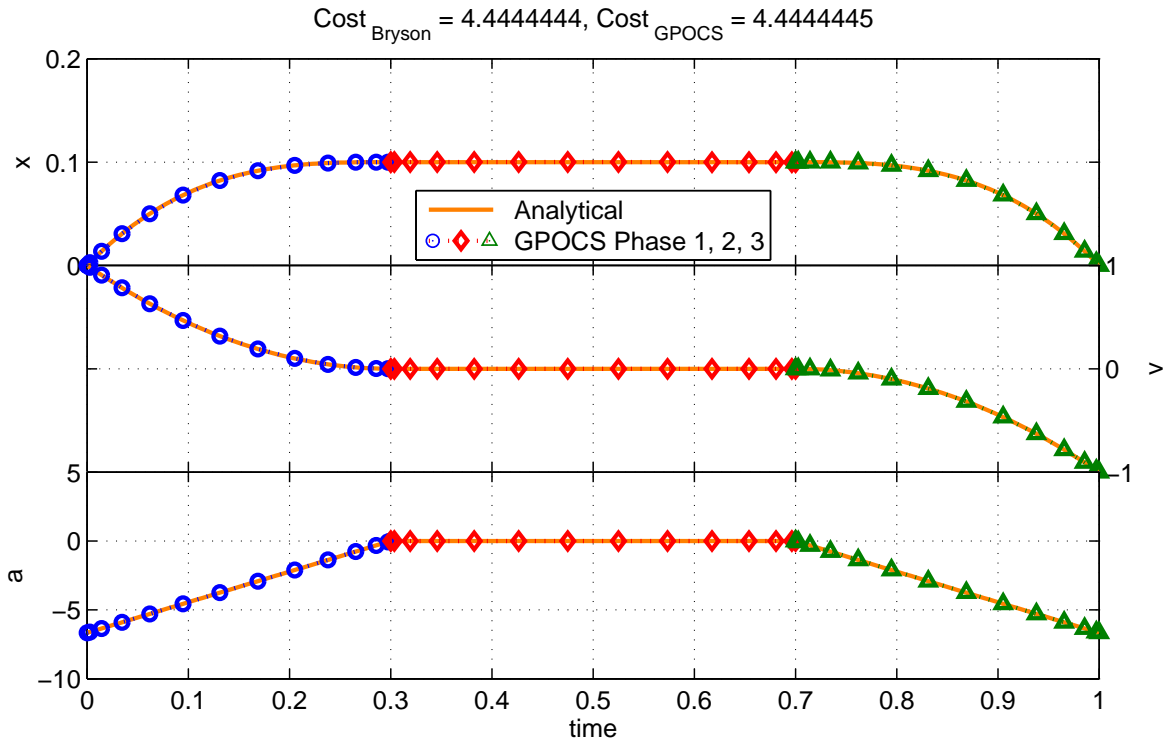
Figure F.2:   Breakwell Problem: Matching Bryson and GPOCS Results

F.6.3 Matlab *Code.*   The following code contains the GPOCS solution, the Bryson and Ho solution, and plotting commands. The GPOCS main program is `Breakwell_Problem`. The main program calls the cost function (`Cost`), the equations of motion (`Dynamics`), the path constraint for the second of three phases (`Path`), and the linkage between multiple phases (`Link`). The support functions are provided as function handles using the `@` call, thus allowing all of these functions to be contained in a single file. For comparison, the analytical Bryson solution is computed in `Bryson_Ho`. The plotting routine, `Plot_Results`, is provided to demonstrate how to extract the cost, time, states, costates, Hamiltonian, and path constraint multipliers from the GPOCS results structure.

```matlab
function Breakwell_Problem % Multiple Methods , Multiple Phases
%% written by: Maj Tim Jorris , AFIT/ENY, DSY-07S, Grad: Sep 2007
clear global ,  global CONST % Remove previous global values
%% Constants
CONST.ell          = 1/10;           % maximium , x(t) <= ell
%% Method
CONST.bryson_method = true;          % true or false
%% Functions
problem.FUNCS.cost     = @Cost;         % Cost
problem.FUNCS.ode      = @Dynamics;    % Equations of Motion
problem.FUNCS.path     = @Path;        % Path Constraint
problem.FUNCS.link     = @Link;         % Linkage between Phases
%% Number of Phases Required
N=36;    % Total Number of Nodes
if CONST.ell > 1/4       % Unconstrained Problem
    %% Nodes
    N=N; Np=length(N); CONST.Np=Np;
    for i=1:Np, ph(i).nodes=N(i); end
    %% Time
    tf=1; tbreak=[0 ; tf]; dt=[0 0];  % breakpoints , +/- amount
    problem.FUNCS=rmfield(problem.FUNCS,'path'); % not required
elseif CONST.ell >= 1/6  % IP Constraint between Phases 1 and 2
    %% Nodes
```

```matlab
    Ni=round(N/2); N=[Ni,Ni]; Np=length(N); CONST.Np=Np;
    for i=1:Np, ph(i).nodes=N(i); end
    %% Time
    tf=1; tbreak=[0 0.5 tf]; dt=[0 0 0];% breakpoints, +/- amount
    problem.FUNCS=rmfield(problem.FUNCS,'path'); % not required
else                       % 3 Phases,  Path Constraint for Phase 2
    %% Nodes
    Ni=round(N/3); N=[Ni,Ni,Ni]; Np=length(N); CONST.Np=Np;
    for i=1:Np, ph(i).nodes=N(i); end
    %% Time
    tf=1; tbreak=[0 3*CONST.ell 1-3*CONST.ell tf]; dt=[0 0 0 0];
end
%% Time of Phase Breakpoints
for i=1:Np
    ph(i).time.min=[tbreak(i)-dt(i) tbreak(i+1)-dt(i+1)];
    ph(i).time.max=[tbreak(i)+dt(i) tbreak(i+1)+dt(i+1)];
end
%% States
x0=0; v0= 1; xf=0; vf=-1; xnames={'x','v'}; unames={'a'};
for i=1:Np, ph(i).states.names=xnames;
    ph(i).states.min=-2*ones(2,3);
    ph(i).states.max= 2*ones(2,3);
    if i==1
        % fixed initial
        ph(i).states.min(:,1)=[x0;v0];
        ph(i).states.max(:,1)=[x0;v0];
        % fixed phase endpoint (if multiple phases)
        if Np >= 2
            ph(i).states.min(:,end)=[CONST.ell;0];
            ph(i).states.max(:,end)=[CONST.ell;0];
        end
    end
    if i==2 & i~=Np % only if multiple phases
        % fixed phase initial
        ph(i).states.min(:,1)=[CONST.ell;0];
```

```matlab
            ph(i).states.max(:,1)=[CONST.ell;0];
            if ~CONST.bryson_method
                % fixed phase endpoint
                ph(i).states.min(:,end)=[CONST.ell;0];
                ph(i).states.max(:,end)=[CONST.ell;0];
            end
        end
        if i==Np
            if ~CONST.bryson_method & Np >= 2
                % fixed phase initial (if multiple phases)
                ph(i).states.min(:,1)=[CONST.ell;0];
                ph(i).states.max(:,1)=[CONST.ell;0];
            end
            % fixed terminal
            ph(i).states.min(:,end)=[xf;vf];
            ph(i).states.max(:,end)=[xf;vf];
        end
    end
%% Controls
for i=1:Np, ph(i).controls.names=unames;
    ph(i).controls.min=-10;  ph(i).controls.max= 10;
end
%% Path
if isfield(problem.FUNCS,'path')
    ph(2).path.min    =[0];  ph(2).path.max    =[0];
end
%% Guess
if sum(N) <= 36 % Initial low number of nodes for estimated sol'n
    t_guess=tbreak([1;end]);   x_g=[ x0 v0; xf vf ];
    for i=1:Np
        guess(i).time    =[tbreak(i)    ;tbreak(i+1)];
        guess(i).controls=interp1(t_guess, x_g ,guess(i).time);
        guess(i).states  =interp1(t_guess, x_g ,guess(i).time);
    end
else    % Increased number of nodes for higher accuracy
```

127

```matlab
        Prev=load([mfilename,'_Results']);  % Previous estimated sol'n
        for i=1:Np
            guess(i).time      = Prev.results.solution(i).time;
            guess(i).controls  = Prev.results.solution(i).controls;
            guess(i).states    = Prev.results.solution(i).states;
        end
    end
%% Linkage
for i=1:(Np-1) % Always one less link than phases
    % Want to enforce state continuity
    linkages(i).left.phase_number=i;
    linkages(i).right.phase_number=i+1;
    linkages(i).min=0*ph(1).states.min(:,1); % gets right size
    linkages(i).max=linkages(i).min;
    problem.linkages=linkages;
end
%% Problem
problem.phases=ph; problem.guess =guess; problem.name=mfilename;
problem.independent_variable='increasing';
problem.Derivatives='automatic'; problem.autoscale='off';
problem.FeasTol=1e-6; problem.OptTol=1e-6; PrintOutput=true;
%% Run
results=gpocs('snopt7',problem,PrintOutput);
save([mfilename,'_Results'])    % Used as estimate for later runs
Plot_Results(results)
%% Cost Function
function [phi, L] = Cost(sol)
phi  = 0; L = 1/2*sol.controls.^2;
%% Equations of Motion
function Xdot = Dynamics(sol)
dx = sol.states(:,2); dv = sol.controls(:,1); Xdot=[dx,dv];
%% Path Constraint
function pth=Path(sol)
global CONST  % phase_number is not necessary
if sol.phase_number==2 & CONST.bryson_method
```

```matlab
            pth=sol.controls;
        else            % This function is only called for Phase 2
            pth=sol.states(:,1)-CONST.ell;
        end
%% Linkage
function link=Link(breakpoint)
xminus=breakpoint.right.state; xplus=breakpoint.left.state;
link=xminus(1:end)-xplus(1:end); % continuous states
%% End of GPOCS Code
%% Bryson and Ho Solution, pg. 121-123
function [results]=Bryson_Ho(ell)
N=201; % Define number of time steps
if ell > 1/4, t=linspace(0,1,N);
elseif ell >= 1/6
    n=round(N/2); % ensures 1/2 is included
    t=[linspace(0,1/2,n)'; linspace(1/2+eps,1,n)']';
else
    n=round(N/3); t=[linspace(0,3*ell,n)';...
        linspace(3*ell+eps,1-3*ell,n)';linspace(1-3*ell+eps,1,n)']';
end
% Define states and costates, p.121 Bryson and Ho
if ell > 1/4
    u=0*t; x=u; v=u; lamx=u; lamv=u; H=u; L=u;
    u=-2+u; x=t.*(1-t); v=1-2*t; lamv=-u; J=2; H=-2+0*t;
elseif ell >= 1/6
    half1=find(t<=1/2); half2=find(t>1/2);
    t1=t(half1); t2=1-t(half2);
    u=0*t; x=u; v=u; lamx=u; lamv=u; H=u;
    u(half1)=-8*(1-3*ell)+24*(1-4*ell)*t1;
    u(half2)=-8*(1-3*ell)+24*(1-4*ell)*t2;
    x(half1)= t1-4*(1-3*ell)*t1.^2+4*(1-4*ell)*t1.^3;
    x(half2)= t2-4*(1-3*ell)*t2.^2+4*(1-4*ell)*t2.^3;
    v(half1)= 1-8*(1-3*ell)*t1+12*(1-4*ell).*t1.^2;
    v(half2)=-1+8*(1-3*ell)*t2-12*(1-4*ell).*t2.^2;
    lamx(half1)=24*(1-4*ell); lamx(half2)=-24*(1-4*ell);
```

```matlab
        lamv=-u;  J=2+6*(1-4*ell)^2;
        Hnum=-8*(1-6*ell)^2;  H(half1)=Hnum;  H(half2)=Hnum;
    else
        s1=find(t<=3*ell);  s2=find(t>3*ell&t<(1-3*ell));
        s3=find(t>=(1-3*ell));  s4=find(t>3*ell);
        ta=1-t(s1)/3/ell;  tb=(1-(1-t(s3))/3/ell);
        u(s1)=-2/3/ell*ta;  u(s2)=0*s2;  u(s3)=-2/3/ell*tb;
        x(s1)=ell*(1-ta.^3);  x(s2)=ell+0*s2;  x(s3)=ell*(1-tb.^3);
        v(s1)=ta.^2;  v(s2)=0*s2;  v(s3)=-tb.^2;
        lamx(s1)=2/9/ell.^2;  lamx(s4)=-2/9/ell^2;
        lamv(s1)=2/3/ell*ta;  lamv(s4)=2/3/ell*(1-(1-t(s4))/3/ell);
        J=4/9/ell;    Hnum=0;  H(s1)=Hnum;  H(s2)=Hnum;  H(s3)=Hnum;
    end
results.cost=J;  results.solution.controls=u;
results.solution.states=[x;v];  results.solution.Hamiltonian=H;
results.solution.costates=[lamx;lamv];  results.solution.time=t;
%% Plotting
function Plot_Results(results)
% Assign Constants
J=results.cost;  soln=results.solution;global CONST,ell=CONST.ell;
rBH=Bryson_Ho(ell);  B=rBH.cost;  sBH=rBH.solution;  % Bryson and Ho
Np=length(soln);  % soln is a structure array containing each phase
props={'defaultAxesFontSize','defaultLineLineWidth',...
        'defaultAxesXGrid','defaultAxesYGrid'};  % plot properties
old=get(0,props);  set(0,props,{12,2,'on','on'})  % reset defaults
% Plot  States and Control
figure(1),clf,  mkr={'bo:','rd:','g^:'};
tb=sBH.time;  xb=sBH.states;  ub=sBH.controls;
for i=1:Np,  t=soln(i).time;  X=soln(i).states;  U=soln(i).controls;
    h1=subplot(3,1,1);  if i==1,plot(tb,xb(1,:),'m-'),  end,  hold on
    hl=plot(t,X(:,1),mkr{i});  ylabel('x'),  if i==1,  hleg=hl;  end
    h2=subplot(3,1,2);  if i==1,plot(tb,xb(2,:),'m-'),  end,  hold on
    plot(t,X(:,2),mkr{i}),  ylabel('v'),
    h3=subplot(3,1,3);  if i==1,plot(tb,ub(1,:),'m-'),  end,  hold on
    plot(t,U(:,1),mkr{i}),  ylabel('a'),  hold on
```

```matlab
end , xlabel('time') % Compute Cost from Bryson & Ho, pg. 122-123
title(h1,sprintf('Cost_{Bryson} = %.7f, Cost_{GPOCS} = %.7f',B,J))
hbry=findobj(h1,'Color','magenta');
hleg=legend([hbry,hleg],'  Bryson & Ho','  GPOCS','Location','S');
hll=findobj(hleg,'LineStyle',':');dat1=get(hll,{'Xdata','Ydata'});
hlm=findobj(hleg,'Marker','o');   dat2=get(hlm,{'Xdata','Ydata'});
hold(hleg,'on'), plot(hleg,dat1{1}(1),dat1{2}(1),'bo')
plot(hleg,dat2{1}(1),dat2{2}(1),'rd',dat1{1}(2),dat1{2}(2),'g^')
set(findall(gcf,'Col',[0,1,0]),'Color',[0,.5,0]) % darker green
% Plot Costates and Hamiltonian
figure(2),clf, pb=sBH.costates; Hb=sBH.Hamiltonian;
for i=1:Np, yl={'\lambda_x','\lambda_v'};
    t=soln(i).time; L=soln(i).costates; H=soln(i).Hamiltonian;
    h1=subplot(3,1,1); if i==1,plot(tb,pb(1,:),'m-'), end, hold on
    hl=plot(t,L(:,1),mkr{i}); ylabel(yl{1}),if i==1, hleg=hl; end
    h2=subplot(3,1,2); if i==1,plot(tb,pb(2,:),'m-'), end, hold on
    plot(t,L(:,2),mkr{i}),ylabel(yl{2})
    h3=subplot(3,1,3); if i==1,plot(tb,Hb(1,:),'m-'), end, hold on
    plot(t,H(:,1),mkr{i}), ylabel('H') , hold on
end , xlabel('time'), title(h1,sprintf('x_{max} = %s',rats(ell)))
bry=findobj(h1,'Color','magenta');
hleg=legend([bry,hleg],'  Bryson & Ho','  GPOCS','Location','NE');
hll=findobj(hleg,'LineStyle',':');dat1=get(hll,{'Xdata','Ydata'});
hlm=findobj(hleg,'Marker','o');   dat2=get(hlm,{'Xdata','Ydata'});
hold(hleg,'on'), plot(hleg,dat1{1}(1),dat1{2}(1),'bo')
plot(hleg,dat2{1}(1),dat2{2}(1),'rd',dat1{1}(2),dat1{2}(2),'g^')
set(findall(gcf,'Col',[0,1,0]),'Color',[0,.5,0]) % darker green
% Plot the Path Mutliplier (if a path constraint is assigned)
if isfield(results.FUNCS,'path') && Np>=2
    figure(3),plot(soln(2).time,soln(2).multipliers.paths(:,1),...
    'rd:'),xlabel('time'), ylabel('\mu'), title('Path Multiplier')
end , set(0,props,old) % return to previous defaults
%% End of Breakwell Problem
```

Bibliography

1. Allwine, Daniel A., Joseph E. Fisher, and Jeremy A. Strahler. "On-Line Trajectory Generation for Hypersonic Vehicles." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 15-18 August 2005.

2. Athans, Michael and Peter L. Falb. *Optimal Control: An Introduction to the Theroy and its Application.* Dover Publications, Inc., 2007.

3. Baralli, Francesco, Lorenzo Pollini, and Mario Innocenti. "Waypoint-Based Fuzzy Guidance for Unmanned Aircraft a New Approach." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 5-8 August 2002. AIAA-2002-4993.

4. Benson, David A., Geoffrey T. Huntington, Tom P. Thorvaldsen, and Anil V. Rao. "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, 29(6):1435–1440, November-December 2006.

5. Bertsekas, Dimitri P. *Nonlinear Programming.* Athena Scientific, Belmont, MA, second edition, 1999. ISBN 1-886529-00-0.

6. Betts, John T. "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, March-April 1998.

7. Betts, John T. "A Direct Approach to Solving Optimal Control Problems," *CSE in Industry*, May-June 1999.

8. Betts, John T. and William P. Huffman. "Path Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming." *AIAA-91-2739-CP.* 1991.

9. Bharadwaj, Sanjay, Anil V. Rao, and Kenneth D. Mease. "Entry Trajectory Tracking Law via Feedback Linearization," *Journal of Guidance, Control, and Dynamics*, 21(5):726–732, September-October 1998.

10. Bollino, Kevin P., L.Ryan Lewis, Pooya Sekhavat, and I. Michael Ross. "Pseudospectral Optimal Control: A Clear Road for Autonomous Intelligent Path Planning." *AIAA 2007 Conference and Exhibit.* 7-10 May 2007.

11. Boyd, John P. *Chebyshev and Fourier Spectral Methods.* Dover Publications, Inc., second edition, 2000. University of Michigan.

12. Bryson, Arthur E., Jr. *Dynamic Optimization.* Addison Wesley Longman, 1999.

13. Bryson, Arthur E., Jr. *Applied Linear Optimal Control: Examples and Algorithms.* Cambridge University Press, 2002.

14. Bryson, Arthur E., Jr. and Yu-Chi Ho. *Applied Optimal Control.* Taylor & Francis, 1975.

15. Chapman, Dean R. *An Approximate Analytical Method for Studying Entry Into Planetary Atmospheres.* Technical Note 4276, National Advisory Committee for Aeronautics, 1958.

16. Chaudhary, A., V. Nguyen, H. Tran, D. Poladian, and E. Falangas. "Dynamics and Stability and Control Characteristics of the X-37." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 6-9 August 2001. AIAA-2001-4383.

17. Chen, D. T. "Using IDOS to Develop EAGLE into "Real" Flight Software to Support Responsive Missions." *AIAA 2nd Responsive Space Conference.* 2004.

18. Clements, John C. "Minimum-Time Turn Trajectories to Fly-To Points," *Optimal Control Applications & Methods*, 11, 1990.

19. DARPA. *FALCON Force Application and Launch from CONUS Task 1 Small Launch Vehicle (SLV) - Phase II*, 2004. Program Solicitation Number 04-05.

20. DARPA/USAF. "Northrop Grumman Takes Aim at Hypersonic Weapon Delivery System," 25 November 2003. `http://www.is.northropgrumman.com/media_news/2003_archive.html`.

21. Desai, Prasun N., Robert D. Braun, and Richard W. Powell. *Aspects of Parking Orbit Selection in a Manned Mars Mission*. Technical Report TP-3256, NASA Langley Research Center, December 1992.

22. Doman, David B. "Introduction: Reusable Launch Vehicle Guidance and Control," *Journal of Guidance, Control, and Dynamics*, 27(6):929, November-December 2004.

23. Dubins, L. E. "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, 79:497–516, 1957.

24. Dukeman, Greg A. "Profile-Following Entry Guidance Using Linear Quadratic Regulator Theory." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 5-8 August 2002. AIAA-2002-4457.

25. ElGindy, Hossam and Lachlan Wetherall. "A Simple Voronoi Diagram Algorithm for a Reconfigurable Mesh," *IEEE Transactions on Parallel and Distributed Systems*, 8(11):1133–1142, November 1997.

26. Erzberger, Heinz and Homer Q. Lee. "Optimum Horizontal Guidance Techniques for Aircraft," *Journal of Aircraft*, 8(2):95–101, February 1971.

27. Fahroo, Fariba and I. Michael Ross. "On Discrete-Time Optimality Conditions for Pseudospectral Methods." *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. 21-24 August 2006.

28. Finzi, A.E., M. Lavagna, and A. Di Gregorio. "Atmospheric Re-Entry Trajectory Tracking and Control for an Unmanned Space Vehicle with a Lyapunov Approach." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 11-14 August 2003. AIAA-2003-5441.

29. Fisher, Joseph E., Tim Bevacqua, Douglas A. Lawrence, J. Jim Zhu, and Michael Mahoney. "Integrated G&C Implementation Within IDOS - A Simulink Based Reusable Launch Vehicle Simulation." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 11-14 August 2003.

30. Fogiel, M. *The Geometry Problem Solver.* Research and Education Association, 1977.

31. Fornberg, Bengt. *A Practical Guide to Pseudospectral Methods.* Cambridge University Press, 1998.

32. Fuhrmann, Henri D. et al. "Conceptual Design of a Reusable Access to Space Technology Demonstrator," *Space 2003*, 23-25 September 2003. AIAA-2003-6395.

33. Hager, William W. "Runge-Kutta Methods in Optimal Control and the Transformed Adjoint System," *Numerische Mathematik*, 87(2):247282, 2000.

34. Hankey, Wilbur L. *Re-Entry Aerodynamics.* American Institute of Aeronautics and Astronautics, 1988.

35. Hanson, John M., Dan J. Coughlin, Gregory A. Dukeman, John A. Mulqueen, and James W. McCarter. "Ascent, Transition, Entry, and Abort Guidance Algorithm Design for the X-33 Vehicle," *AIAA*, 1998. AIAA-98-4409.

36. Hanson, John M. and Robert E. Jones. "Advanced Guidance and Control Methods for Reusable Launch Vehicles: Test Results." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 5-8 August 2002. AIAA-2002-4561.

37. Hanson, John M. and Robert E. Jones. "Test Results for Entry Guidance Methods for Space Vehicles," *Journal of Guidance, Control, and Dynamics*, 27(6):960–966, November-December 2004.

38. Hanson, JohnM. "Advanced Guidance and Control Project for Reusable Launch Vehicles." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 14-17 August 2000.

39. Harpold, Jon C. and Claude A. Graves, Jr. "Shuttle Entry Guidance," *The Journal of the Astronautical Sciences*, 27(3):239–268, July-September 1979.

40. Hartl, Richard F., Suresh P. Sethi, and Raymond G. Vickson. "A Survey of the Maximum Principles for Optimal Control Problems with State Constraints," *SIAM Review*, 37(2):181–218, June 1995.

41. Hicks, Kerry D., Lt Col. "Introduction to Astrodynamic Reentry," May 2007. Air Force Institute of Technology, Wright-Patterson AFB, OH, Edition 0.7, (unpublished).

42. Hueter, Uwe and John J. Hutt. "NASA's Next Generation Launch Technology Program - Next Generation Space Access Roadmap." *12th AIAA International Space Planes and Hypersonic Systems and Technologies.* 15-19 December 2003.

43. Hull, David G. "Conversion of Optimal Control Problems into Parameter Optimization Problems," *Journal of Guidance, Control, and Dynamics*, 20(1):57–60, January-February 1997.

44. Hull, David G. *Optimal Control Theory for Applications.* Springer-Verlag, 2003.

45. Hull, Jason R., Neha Gandhi, and John D. Schierman. "In-Flight TAEM-Final Approach Trajectory Generation for Reusable Launch Vehicles." *Infotech@Aerospace.* 26-29 September 2005. AIAA-2005-7114.

46. Innocenti, Mario, Lorenzo Pollini, and Demetrio Turra. "Guidance of Unmanned Air Vehicles Based on Fuzzy Sets and Fixed Waypoints," *Journal of Guidance*, 27(4):715–720, 2002.

47. Jacobs, Thomas H., Elan T. Smith, and Michael W. Garrambone. "Space Access Vehicles Mission and Operations Simulation (SAVMOS) For Simulating Reusable Launch Vehicles," 2005.

48. Jorris, Timothy R. and Richard G. Cobb. "2-D Trajectory Optimization Satisfying Waypoints and No-Fly Zone Constraints." *AAS/AIAA Space Flight Mechanics Meeting.* 28 Jan - 1 Feb 2007. AAS 07-114.

49. Judd, Kevin B. and Timothy W. McLain. "Spline Based Path Planning for Unmanned Air Vehicles." *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA-2001-4238. 2001.

50. Kreim, H., B. Kugelmann, H. J. Pesch, and M. H. Breitner. "Minimizing the Maximum Heating of a Re-Entering Space Shuttle: An Optimal Control Problem with Multiple Control Constraints," *Optimal Control Applications & Methods*, 17:45–69, 1996.

51. Kreyszig, Erwin. *Advanced Engineering Mathematics*. John Wiley & Sons, Ltd., 8th edition, 1999.

52. Kuwata, Yoshiaki and Jonathan How. "Three Dimensional Receding Horizon Control for UAVs." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 16-19 August 2004. AIAA-2004-5144.

53. Kuwata, Yoshiaki, Tom Schouwenaars, Arthur Richards, and Jonathan How. "Robust Constrained Receding Horizon Control for Trajectory Planning." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 15-18 August 2005. AIAA-2005-6079.

54. Leavitt, J. A., A. Saraf, D. T. Chen, and K. D. Mease. "Performance of Evolved Acceleration Guidance Logic for Entry (EAGLE)." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 5-8 August 2002.

55. Lu, Ping. "Regulation about Time-Tarying Trajectories: Precision Entry Guidance Illustrated," *AIAA*, 1999. AIAA-99-4070.

56. Lu, Ping and Nguyen X. Vinh. "Optimal Control Problems with Maximum Functional," *Journal of Guidance*, 14(6):1215–1223, November-December 1991.

57. Mahoney, Michael J. and Layne Cook. "Development and Operations Of Flight Systems for Responsive Missions." *2nd Responsive Space Conference*. 19-22 April 2004.

58. MathWorks. *Optimization Toolbox: For Use with MATLAB*, 2006. User's Guide Version 3.

59. Mease, K. D., D. T. Chen, S. Tandon, D. H. Young, and S. Kim. "A Three-Dimensional Predictive Entry Guidance Approach," *American Institute of Aeronautics and Astronautics*, 2000. AIAA-2000-3959.

60. Mease, K. D., D. T. Chen, P. Teufel, and H. Schönenberger. "Reduced-Order Entry Trajectory Planning for Acceleration Guidance," *Journal of Guidance, Control, and Dynamics*, 25(2):257–266, March-April 2002.

61. Mease, Kenneth D., P. Teufel, H. Schönenberger, D.T. Chen, and S. Bharadwaj. "Re-entry Trajectory Planning for a Reusable Launch Vehicle," *American Institute of Aeronautics and Astronautics*, 1999. AIAA-99-4160.

62. Nelson, Doug. *Qualitative and Quantitative Assessment of Optimal Trajectories by Implicit Simulation (OTIS) and Program to Optimize Simulated Trajectories (POST)*. Technical Report, Georgia Institute of Technology, 2001. AE 8900 Individual Research Project.

63. Ngo, Anhtuan D. and William B. Blake. "Longitudinal Control and Footprint Analysis for a Reusable Military Launch Vehicle." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 11-14 August 2003. AIAA-2003-5738.

64. Okabe, Atsuyuki, Barry Boots, and Kokichi Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Ltd., 1992.

65. Olds, John R. and Irene A. Budianto. "Constant Dynamic Pressure Trajectory Simulation with POST," *AIAA*, (AIAA-1998-302), 1998.

66. Ong, Shaw Y. and Bion L. Pierson. "2-D and 3-D Minimum-Time-To-Turn Flights via Parameter Optimization," *American Institute of Aeronautics and Astronautics*, 1992. AIAA-92-0731.

67. Paris, S.W. and S. R. Hargraves. *Optimal Trajectories by Implicit Simulation (OTIS) Volume II - User's Manual.* Boeing Defense and Space Group, 1996.

68. Petit, Nicolas, Mark B. Milam, and Richard M. Murray. "Inversion Based Constrained Trajectory Optimization," California Institute of Technology, 2000.

69. Phillips, Terrry H. *A Common Aero Vehicle (CAV) Model, Description, and Employment Guide.* Technical Report, Schafer Corporation for AFRL and AFSPC, 27 January 2003.

70. Raghunathan, Arvind U., Vipin Gopal, Dharmashankar Subramanian, Lorenz T. Biegler, and Tariq Samad. "3D Conflict Resolution of Multiple Aircraft via Dynamic Optimization." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 11-14 August 2003.

71. Raiszadeh, Ben and Eric M. Queen. *Partial Validation of Multibody Program to Optimize Simulated Trajectories II (POST II) Parachute Simulation with Interacting Forces.* Technical Report, NASA, Langley Research Center, Hampton, Virginia, 2002. TM-2002-211634.

72. Rao, Anil V. *User's Manual for GPOCS$^{©}$ Version 1.0: A MATLAB$^{®}$ Implementation of the Gauss Pseudospectral Method for Solving Multiple-Phase Optimal Control Problems.* TOMLAB^{TM}, Gainesville, FL 32607, July 2007.

73. Regan, Frank J. and Satya M. Anandakrishnan. *Dynamics of Atmospheric Re-Entry.* American Institute of Aeronautics and Astronautics, 1993.

74. Richards, A., Y. Kuwata, and J. How. "Experimental Demonstrations of Real-time MILP Control." *Proceedings of the AIAA Guidance, Navigation, and Control Conference.* Austin, TX, August 2003.

75. Richie, George. "The Common Aero Vehicle: Space Delivery System of the Future." *AIAA Space Technology Conference & Exposition.* 28-30 September 1999. A99-42026.

76. Richie, George and John Haaren. "Striking from Space - The Future of Space Force Applications," *American Institute of Aeronautics and Astronautics, Inc.*, 1998.

77. Ross, I. Michael and Christopher N. DSouza. "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, 28(4):686–697, July-August 2005. AIAA-8285-973.

78. Ross, I. Michael and Fariba Fahroo. "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Sciences*, 295, 2003.

79. Ross, I. Michael and Fariba Fahroo. "A Unified Framework for Real-Time Optimal Control." *IEEE Conference on Decision and Control*. December 2003.

80. Ross, I. Michael and Fariba Fahroo. "Discrete Verification of Necessary Conditions for Switched Nonlinear Optimal Control Systems." *Proceeding of the 2004 American Control Conference*. June 30-July 2 2004.

81. Saraf, A., J. A. Leavitt, D. T. Chen, and K. D. Mease. "Design and Evaluation of an Acceleration Guidance Algorithm for Entry AIAA-11015-306," *Journal of Spacecraft and Rockets*, 41(6):986–996, November-December 2004.

82. Schierman, J. D., J. R. Hull, and D. G. Ward. "Adaptive Guidance with Trajectory Reshaping for Reusable Launch Vehicles." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 5-8 August 2002. AIAA-2002-4458.

83. Schierman, J. D., D. G. Ward, J. F. Monaco, and J. R. Hull. "A Reconfigurable Guidance Approach for Reusable Launch Vehicles." *American Institute of Aeronautics and Astronautics*. 2001. AIAA-2001-4429.

84. Schierman, John D. and Jason R. Hull. "In-Flight Entry Trajectory Optimization for Reusable Launch Vehicles." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 15-18 August 2005. AIAA-2005-6434.

85. Schierman, John D., David G. Ward, Jason R. Hull, Neha Gandhi, Michael W. Oppenheimer, and David B. Doman. "Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight Test Results," *Journal of Guidance, Control, and Dynamics*, 27(6):November–December, 2004.

86. Scholz, Edwin, John Duffey, Steven Sasso, Gregory Peralta, and Lt. Lee Jackson. "Overview of SOV Concepts and Technology Needs." *Joint Propulsion Conference and Exhibit*. 20-23 July 2003.

87. Shaffer, Patrick J., I. Michael Ross, Michael W. Oppenheimer, and David B. Doman. "Optimal Trajectory Reconfiguration and Retargeting for a Reusable Launch Vehicle." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 15-18 August 2005.

88. Shapira, Ilana and Joseph Z. Ben-Asher. "Near-Optimal Horizontal Trajectories for Autonomous Air Vehicles," *Journal of Guidance, Control, and Dynamics*, 20(4):735–741, July-August 1997.

89. Shen, Zuojun and Ping Lu. "On-Board Entry Trajectory Planning Expanded to Sub-Orbital Flight." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 11-14 August 2003.

90. Shen, Zuojun and Ping Lu. "Onboard Generation of Three-Dimensional Constrained Entry Trajectories," *Journal of Guidance, Control, and Dynamics*, 26(1):111–121, January-February 2003.

91. Shen, Zuojun and Ping Lu. "Dynamic Lateral Entry Guidance Logic," *Journal of Guidance, Control, and Dynamics*, 27(6):949–959, November-December 2004. AIAA-8008-728.

92. Shen, Zuojun and Ping Lu. "On-Board Entry Trajectory Planning for Sub-Orbital Flight," *Acta Astronautica*, 56(6):573–591, 2005.

93. Slotine, Jean-Jacques E. and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

94. Stakgold, Ivar. *Green's Functions and Boundary Value Problems.* Wiley-Interscience, 1998. 2nd Edition.

95. von Stryk, Oskar. "Numerical Solution of Optimal Control Problems by Direct Collocation," *International Series of Numerical Mathematics*, 111:129–143, 1993.

96. von Stryk, Oskar and R. Bulirsch. "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, 37:357–373, 1992.

97. Tewari, Ashish. *Atmospheric and Space Flight Dynamics.* Birkhäuser, 2007.

98. Thompson, Elvia, Keith Henry, and Leslie Williams. "Guinness Recognizes NASA Scramjet," 20 June 2005. `http://www.nasa.gov/audience/formedia/archives/2005-all-archives.html`.

99. Trefethen, Lloyd N. *Spectral Methods in MATLAB.* Society for Industrial and Applied Mathematics, 2000.

100. Twigg, Shannon, Anthony Calise, and Eric Johnson. "On-Line Trajectory Optimization Including Moving Threats and Targets." *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA 2004-5139. 2004.

101. USAF. *SR-71A Flight Manual (U)*, October 1989. `http://www.sr-71.org`. Change 2.

102. USAF. "The Joint Air Estimate Planning Process," 2003.

103. USAF. *Joint Publication 3-30 Command and Control for Joint Air Operations*, 2003.

104. USAF. "Air Force Handbook - 109th Congress," Department of the Air Force, Washington, DC, 2005.

105. USAF/AEDC. "Tunnel 9 Completes Vehicle Testing," 1 November 2005. `http://www.arnold.af.mil/aedc/highmach/2005/oct21/tunnel9.htm`.

106. Valenti, M., T. Schouwenaarsy, Y. Kuwataz, E. Feronx, and J. How. "Implementation of a Manned Vehicle - UAV Mission System." *AIAA Guidance, Navigation, and Control Conference and Exhibit.* 16-19 August 2004. AIAA-2004-5142.

107. Venkataraman, P. *Applied Optimization with MATLAB Programming.* John Wiley & Sons, Ltd., New York, 2002. ISBN 0-471-34958-5.

108. Vian, John L. and John R. Moore. "Trajectory Optimization with Risk Minimization for Military Aircraft," *Journal of Guidance*, 12(3):311–317, May-June 1989.

109. Vihn, Nguyen X., Adolf Busemann, and Robert D. Culp. *Hypersonic and Planetary Entry Flight Mechanics.* University of Michigan Press, 1980.

110. Vinh, Nguyen X. *Optimal Trajectories in Atmospheric Flight.* Elsevier, 1981.

111. Vinh, Nguyen X. and Der-Ming Ma. "Optimal Multiple-Pass Aeroassisted Plane Change," *Acta Astronautica*, 21(11/12/2007):749–758, 1990.

112. Weatherington, Dyke and Allen Wilson. "The Office of the Secretary of Defense (OSD) Unmanned Aerial Vehicles (UAV) Common Mission Planning Architecture (CMPA) - An Overview." *2nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations.* 15-18 September 2003. AIAA-2003-6522.

113. Whang, Ick Ho and Tae Won Whang. "Horizontal Waypoint Guidance Design Using Optimal Control," *IEEE Transactions on Aerospace and Electric Systems*, 38(3):1116–1120, July 2002.

114. Wiesel, William E. *Spaceflight Dynamics.* The McGraw-Hill Companies, Inc., 2nd edition, 1997.

115. Williams, P. "A Gauss-Lobatto Quadrature Method for Solving Optimal Control Problems." Andrew Stacey, Bill Blyth, John Shepherd, and A. J. Roberts (editors), *Proceedings of the 7th Biennial Engineering Mathematics and Applications Conference, EMAC-2005*, volume 47 of *ANZIAM J.*, C101–C115. July 24 2006. `http://anziamj.austms.org.au/V47EMAC2005/Williams2`.

116. Williams, Paul. "Jacobi Pseudospectral Method for Solving Optimal Control Problems," *Journal of Guidance, Control, and Dynamics*, 27(2):293–297, 2004.

117. Withrow, Melissa. "Directorate Demonstrates Simulation Environment," news@afrl, October/November 2004.

118. Wolf, Amy F. "Conventional Warheads for Long-Range Ballistic Missiles: Background and Issues for Congress," CRS Report for Congress, 19 June 2007. Order Code RL33067.

119. Yang, Hong and Yiyuan J. Zhao. "Efficient Trajectory Synthesis Through Specified Waypoints," *AIAA 3rd "Unmanned Unlimited" Technical Conference*, 20-23 September 2004.

120. Yang, Hong Iris and Yiyuan J. Zhao. "Trajectory Planning for Autonomous Aerospace Vehicles amid Known Obstacles and Conflicts," *Journal of Guidance, Control, and Dynamics*, 27(6):997–1008, Nov-Dec 2004.

121. Youssef, Jussein and Rajiv Chowdhry. "Hypersonic Global Reach Trajectory Optimization." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Providence, Rhode Island, 16-19 August 2004.

122. Zimmerman, Curtis, Greg Dukeman, and John Hanson. "Automated Method to Compute Orbital Reentry Trajectories with Heating Constraints," *Journal of Guidance, Control, and Dynamics*, 26(4):523–529, July-August 2003.

Index

The index is conceptual and does not designate every occurrence of a keyword. Page numbers in bold represent concept definition or introduction.

REPORT DOCUMENTATION PAGE

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD–MM–YYYY)	2. REPORT TYPE		3. DATES COVERED (From — To)
11–09–2007	Doctoral Dissertation		Sep 2004 — Sep 2007

4. TITLE AND SUBTITLE

Common Aero Vehicle Autonomous Reentry Trajectory Optimization Satisfying Waypoint and No-Fly Zone Constraints

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jorris, Timothy, R., Maj, USAF

5d. PROJECT NUMBER

07-298

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/DS/ENY/07-04

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/VS
Attn: Russell E. Partch
3550 Aberdeen Ave
Kirtland AFB, NM 87117-5776   DSN: 246-8929

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT   To support the Global Strike mission, an autonomous trajectory optimization technique is presented to minimize the flight time, satisfy terminal and intermediate constraints, and remain within the specified vehicle heating and control limitations. "Waypoints" are specified for reconnaissance or multiple payload deployments and "no-fly zones" are specified for geopolitical restrictions or threat avoidance. The Hypersonic Cruise Vehicle (HCV) is used as a simplified two-dimensional platform to compare multiple solution techniques. The solution techniques include a unique geometric approach, an analytical dynamic optimization technique, and a numerical approach. This numerical technique is a direct solution method involving pseudospectral methods and nonlinear programming to converge to the optimal solution. The Common Aero Vehicle (CAV) is used as the test platform for the full three-dimensional reentry trajectory optimization problem. The culmination of this research is the verification of the optimality of this proposed numerical technique, as shown for both the two-dimensional and three-dimensional models. Lastly, user implementation strategies are presented to improve accuracy and enhance solution convergence.

15. SUBJECT TERMS

reentry vehicles, hypersonic flight, suborbital trajectories, optimization, first strike capability, Global Strike, Common Aero Vehicle, Hypersonic Cruise Vehicle, waypoint, no-fly zone, pseudospectral, collocation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Richard G. Cobb (ENY)
U	U	U	UU	166	19b. TELEPHONE NUMBER (include area code) (937) 255–3636, ext. 4559