3-10-2010

Simulation and Application of GPOPS for Trajectory Optimization and Mission Planning Tool

Danielle E. Yaple

SIMULATION AND APPLICATION OF GPOPS FOR A TRAJECTORY OPTIMIZATION AND MISSION PLANNING TOOL

THESIS

Danielle E. Yaple, 2nd Lieutenant, USAF

AFIT/GAE/ENY/10-M29

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/10-M29

SIMULATION AND APPLICATION OF GPOPS FOR A TRAJECTORY OPTIMIZATION AND MISSION PLANNING TOOL

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

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 Master of Science in Aeronautical Engineering

Danielle E. Yaple, BS

2nd Lieutenant, USAF

March 2010

AFIT/GAE/ENY/10-M29

SIMULATION AND APPLICATION OF GPOPS FOR A TRAJECTORY OPTIMIZATION AND MISSION PLANNING TOOL

Danielle E. Yaple, BS

2nd Lieutenant, USAF

Approved:

/signed/	
Dr. Richard G. Cobb (Chairman)	Date
/signed/	
Lt Col Frederick G. Harmon (Member)	Date
/signed/	
Lt Col Douglas D. Decker (Member)	Date

Abstract

Rapid trajectory generation is crucial to prompt global warfare. To meet the USAF's objective of *Persistent and Responsive Precision Engagement*, a rapid mission planning tool is required. This research creates the framework for the mission planning tool and provides a sample optimal trajectory which is solved using the GPOPS software package. GPOPS employs a Gaussian pseudospectral method to solve the non-linear equations of motion with both end conditions and path constraints. By simultaneously solving the entire trajectory based on an initial guess and small number of nodes, this method is ideal for generating rapid solutions. The sample case is a multi-phase minimum time, optimal control problem which is used to validate the planning tool. The developed framework includes different atmospheric models, gravity models, inclusion of no-flyzones and waypoints, and the ability to create a library of sample cases. This versatile tool can be used for either trajectory generation or mission analysis.

The results of this research show the complexities in solving an optimal control problem with states that change from one phase of the problem to another. At the conclusions of this research multiple phases were successfully connected and solved as a single optimal control problem. However, the entire trajectory solution from launch to impact solved simultaneously, is still an objective yet to be demonstrated. The results found should be a solid foundation for a future mission planning tool.

Acknowledgments

I would like to thank Dr. Cobb for his support, Billy Karasz for helping understand the inner workings of GPOPS, and my family and friends for putting up with me after little sleep and a lot of stress.

Danielle Yaple

Table of Contents

Page

List of Figures

Page

Page

Page

List of Tables

Page

List of Symbols

ϕ	latitude or Mayer cost
θ	longitude
γ	flight path angle
ψ	heading angle
β	atmospheric scaling height
μ	gravitational constant
x(t)	state vector
u(t)	control vector
J	cost function
H	Hamiltonian
C_L	Coefficient of Lift
C_D	Coefficient of Drag
ω_e	angular rotation rate of the earth
ρ	density of atmosphere
g	gravity
L	Lagrange
$\dot{x}$	EOMs
t	time
C	constraints
h	altitude

P	pressure
R	universal gas constant
T	temperature or thrust or kinetic energy
T_h	temperature gradient
r	radius
D	drag
r_p	non-dimensional radius
${}^{R}V_p$	non-dimensional relative velocity
σ	bank angle
$\dot{\bar{q}}_s$	stagnation heating
η	altitude
S	reference area

List of Abbreviations

AFGSC	Air Force Global Strike Command
AFRL	Air Force Research Laboratory
DAE	Differential Algebraic Equations
EGM96	Earth Gravity Model 1996
EOM	Equation of Motion
FLTC	Focused Long-Term Challenges
GPOPS	General Pseudospectral OPtimal control Software
GPS	Global Positioning System
GUI	Graphic User Interface
KLC	Kodiak Launch Center
KSC	Kennedy Space Center
NLP	Non-Linear Programming
SQP	Sequential Quadratic Programming
USAF	United States Air Force
VAFB	Vandenberg Air Force Base
WFF	Wallops Flight Facility
WGS84	World Geodetic System 1984

SIMULATION AND APPLICATION OF GPOPS FOR TRAJECTORY OPTIMIZATION AND MISSION PLANNING TOOL

I. Introduction

Hollywood has romanticized the concept of the strategic super computer and war. Movies imply that the all knowing machines of artificial intelligence have their own sense of awareness and eventually can take over the world. In fact, computers are more capable than humans of sorting through vast amount of information in performing repetitive calculations. With the current advances in computing power, the ability to rapidly conduct complex strategic and tactical war plans, like in Hollywood storylines, is only a question of when.

1.1 Motivation

A mission planning tool is very applicable to today's Air Force mission. A tool, which can rapidly produce optimal trajectories has both operational and research potential. Operationally, this tool can be put to use in support the new Global Strike Command mission. For research, such a tool could demonstrate the utility and impact of new vehicle technologies and guidance systems.

1.1.1 Operational Applications

In September of 2009 the Air Force's new command, Air Force Global Strike Command, began operations to support the Global Strike mission. This command has taken over missions for the B-2 Spirit, B-52 Stratofortress and Minuteman III (LGM-30)

[1]. The mission of Global Strike is to be able to hit any desirable target in an advantageous time. The exact criterion is still flexible as it is more dictated by our vehicle limitations than flight planning limits. Though the Global Strike Command has specific units attached to its control, the concepts behind Global Strike are Air Force wide. The goal is to hit the enemy faster and harder from further away.

An easy to use mission planning tool would support this mission by allowing for new trajectories to be created rapidly. This would help the mission directors in planning attacks quickly and also permit them to make unlimited adjustments to meet demands. Being able to see numerous trajectory possibilities allows leaders to make well informed decisions quickly and match the appropriate weapon to the appropriate target with confidence.

1.1.2 Air Force Research Lab's Application

The Air Force Research Laboratory (AFRL) is broken into groups called Focused Long-Term Challenges (FLTC). FLTC 4 is dedicated to *Persistent and Responsive Precision Engagement*, and within a subset of FLTC 4 lies challenge 4.2, *Globally Deliver Full Spectrum of Kinetic Energy*. The short-term goal for this challenge is to hit a target from anywhere in theatre subsonically. The mid-range goal is to hit a target 600 nm in 20 min or 2000 nm in less than 2 hours at speeds greater than or equal to Mach 2. In the long-term, AFRL would like to be able to hit a target 9000 nm away in a time period of 2 hours or less. Eventually they would like to incorporate hypersonic speed and time periods of 1 hr or less to hit the desired target [8].

As envisioned, a mission planning tool should help to facilitate the afore mentioned AFRL goals through its intended ability to set constraints and leave other variables free. If reaching a desired range is the goal other constraints can be allowed to change to see what is required to achieve that range. The same can be done with the time, speed and error. The speed at which trajectories can be calculated would allow for a greater depth of investigation in the same amount of time.

1.2 Problem Statement

The concept of a tool that can map trajectories seems simple enough, yet in reality the dynamics and constraints pose an intense problem to solve. This problem involves non-linear differential equations, two point boundary value problems, and path constraints. The top level problem description is to design an optimized trajectory to get a missile from a launch location to a prescribed target location.

In order to accomplish the task of creating the optimized trajectory many subtasks need to be accomplished. Below are a few goals to solving this problem:

1. Break the problem up into phases
2. Define the reference frames and derive the equations of motion for each phase
3. Connect the phases
4. Set up constraints for the problem
5. Develop a specific case to test and support the above goals.

In order to solve the complicated problem of optimizing a trajectory, the optimal control problem needs to be broken up into 5 phases: 3 at launch (due to 3 stage vehicle),

a re-entry phase and a terminal phase. These phases are then optimized to minimize overall time in support of the Global Strike mission.

After defining the phases, the reference frame for each phase has to be defined to allow for the phases to be tied together. The first section of phases (launch) is defined in an earth centered inertial reference frame and the other two phases are both in earth centered relative reference with the re-entry states defined in spherical coordinates, and the terminal phase in x, y, and z coordinates.

The equations of motion for each section are based on the reference frame that the section is in. The first section has thrust, drag and gravity and has thrust control. The second section equations are the re-entry equations that include drag and lift and has bank angle as a control. The final section is assumed to be ballistic and has no control and is driven by gravity and the initial conditions for that phase.

After all the phases are defined by their equations and constraints, the complete problem is given to the solver. The solver being used is General Pseudospectral OPtimization control Software (GPOPS). This solver provides a solution for all of the above given information. This solver will output a trajectory, the control and Hamiltonian for the optimal control problem.

1.3 Preview

The rest of this thesis is broken down into four chapters; Chapter II will provide a literature review and background for the research. It will include a history of dynamic optimization and psuedospectral methods. It also includes information needed to design a mission planning tool such as: possible vehicles, atmospheric models and launch or target

sites. Finally, it includes a brief description of previous tools used to accomplish this goal and research that has been done in this area. Chapter III is the methodology and it covers the problem statement, assumptions crucial to the mission, how the problem is broken up and the method in which it is to be solved. Also included in this chapter is the break down as to how the mission planning tool will work when all the pieces are integrated correctly. Chapter IV is the results and analysis of the research. It includes the final trajectory and graphics which illustrate how different inputs impacted the trajectory. Finally in Chapter V contains the conclusions and suggestions for future research.

II. Literature Review

2.1 Dynamic Optimization

The concepts of dynamic optimization date back to the 17th century when Johann Bernoulli posed a problem to his fellow scholars, such as Leibniz, l'Hopital, Newton and Bernoulli's brother, Daniel. This problem was called the brachistochrone problem which is Greek for 'shortest time'. This problem was very true to its name, the goal was to find the minimum time for a bead to move down a string. From the idea of the brachistochrone problem a new field of mathematics was born, optimization theory. In 1733 Euler wrote *Elmenta Calculi Variatonum*, translated meaning Calculus of Variations and thus the founding mathematics behind optimization were established [12].

In the 1950's Bellman made a contribution to Dynamic programming in updating the work of Hamilton and Jacobi thus formulating the Hamiliton-Jacobi-Bellman equation. In 1962 Pontryagin's max(min) principle revolutionalized optimization theory in providing a method to determine optimal control in a constrained problem. It is the underlying principle behind a bang-bang optimal control solution.

Equation 1 shows the generic problem statement for a current day, optimal control problem, breaking it up into a cost equation, dynamics, boundary conditions and constraints.

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

subject to the system dynamics:

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

the boundary conditions: (1)

$$t_0 \text{ and } \mathbf{x}_0$$

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

and constraints (equality and inequality):

$$C(\mathbf{x}_f, t_f, \mathbf{x}_0, t_0) \leq 0$$

where J is the cost function, L is the Lagrange, and ϕ is the Mayer, x is the state vector, $\dot{x}$ is the dynamic vector (derivative of the states), t is the time, t_0 and x_0 are the initial conditions, ψ contains the function for the final conditions and C is the constraints.

Analytical solutions to optimal control problems exist for simple cases. The majority of real-world problems would be too complex to solve using analytical methods and a numerical approach is needed. There are a numerous numerical methods in use today.

Numerical methods can be either indirect or direct. Indirect methods use the first order optimality condition and solve the Hamiltonian for a boundary value problem. To solve problems indirectly, Calculus of Variations or the Method of Lagrange Multipliers are used. Two Point Boundary Value Problems can be solved with multiple standard methods. Small radii of convergence is a method that integrates states forward and the co-states backwards to solve for a solution. Another method, Steepest Decent, looks at the gradients of the dynamics and performance indexes to solve the differential equations [25].

Direct methods are based on parameterizing functions with approximations and then transcribing them to a nonlinear program. These methods are the basis of pseudospectral methods.

2.2 Pseudospectral Methods

Pseudospectral methods are a subset of numerical methods. In order to explain Pseudospectral solution methods one must first understand nonlinear programming (NLP). NLP is the mathematical process of solving a system of equations incorporating both equalities and inequalities with a cost or objective function that is to be minimized or maximized. The nonlinear part is in the constraint equations or cost function. In equation form an NLP looks similar to the generic case below:

$$\begin{aligned} &\text{Min } f(z) \\ &\text{s.t. } g(z) \leq 0 \\ &h(z) = 0 \\ &z \in Z \end{aligned} \tag{2}$$

where g is inequality constraint, h is the equality constraint, f is the objective/cost function, and z is a vector of parameters that exist in the domain of Z to include a set of solutions which satisfy all constraints.

There are three different NLP solvers used regularly with MATLAB®: SNOPT, NPSOL and IPOPT. SNOPT "employs a sparse SQP algorithm with limited-memory quasi-Newton approximation to the Hessian or Lagrangian" whereas NPSOL employs a dense SQP algorithm. [2] Both SNOPT and NPSOL are effective on large scale

problems, but they have computationally expensive functions and gradients. IPOPT is also a large scale solver, but uses an interior point method. All of the NLP solvers use a nodal method to solve the problem. The problem in Equation 2 is broken up into phases and then discretized to a finite number of points (nodes) assigned to each phase and then the problem is solved at those points. An example of a multiple-phase problem is shown in Figure 1.

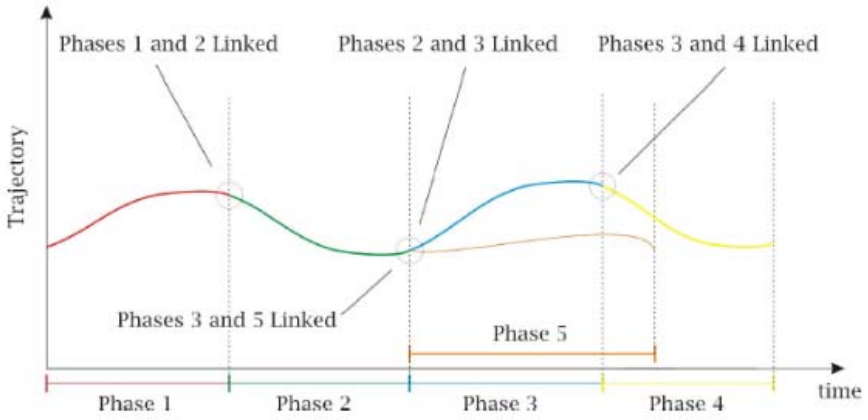


Figure 1: Linkage for Multiple-Phase Optimal Control Problem [34]

Next, a collocation scheme is used to capture the dynamics at each of the nodes. The approximations at each point come from 1 of 3 collocation schemes, Lobatto, Radau, or Gauss. A Lobatto scheme enforces conditions on the endpoint of the area of interest. An example of this method would be the trapezoid rule or Hermite-Simpson method. A Radau scheme enforces conditions only at one end of the area of interest. An example of this would be the Runge-Kutta method. Finally a Gauss scheme has conditions enforced strictly on the interior of the interval. An example of this method is the midpoint rule.

When a collocation scheme is combined with an NLP solver, a tool can be developed to solve nonlinear optimization problems. [33]

There are multiple software packages that are used to solve optimal control problems. Built into MATLAB®, is a toolbox, 'optimtool' which will solve both constrained and unconstrained problems as well as controlled and uncontrolled problems. However there are limitations to this software and it would be hard to set up a multiple phase problem.

Software packages have implemented the pseudospectral method for solving optimal control problems. These are programs such as DIDO and GPOPS (General Pseudospectral OPtimal Control Software). Both of these will solve nonlinear optimal control problems with path constraints. GPOPS is a Gaussian Pseudospectral Method (GPM) program written in MATLAB® and solves complicated multiple phase optimal control problems. It's structured in multiple levels to allow for each phase to have user defined specifications. GPOPS resembles other software in its structure that were written in FORTRAN as opposed to MATLAB®.

2.3 Environmental Models

In setting up any problem, including any of the methods explained in the previous section, the conditions need to be defined in which the problem is solved. Thus the environmental models used in this method need to be explored. The atmospheric and earth models are described below.

2.3.1 Atmospheric Models

A basic atmospheric model would be a simple exponential model. This would have density and pressure as an exponential function of altitude. The most commonly used atmospheric model however is the standard atmospheric model.

The standard atmosphere model is based off the hydrostatic equation. Assuming that the velocity is zero (no winds) a pressure, temperature and density model are developed as a function of altitude. The Mean Sea Level (MSL) values are from data collections:

$$T_0 \equiv 288.16\ K \equiv 518.69°R$$

$$P_0 \equiv 101.325\ kPa \equiv 2116.2\ lb/ft^2$$

$$\rho_0 \equiv 1.225\ kg/m^3 \equiv .002377\ slug/ft^3$$

The temperature model can be seen in Figure 2. It is approximated into areas of constant temperature (isothermal layers) and areas of varying temperature (gradient layers) based off of scientific data.

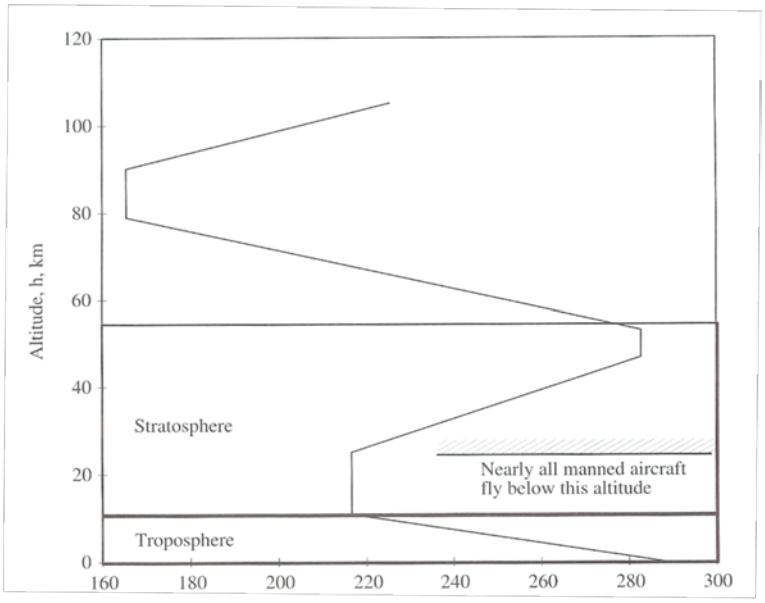


Figure 2: Temperature Profile for Standard Atmosphere [6]

The equations for pressure and density are based off of the layer they fall in since they are reliant on the temperature. The equations for isothermal layers are equal because temperature is constant. Equation 3 shows the isothermal layers equation for density and pressure varying over altitude for a constant temperature.

$$\frac{\rho}{\rho_1} = \frac{P}{P_1} = e^{-\left(g/RT\right)(h-h_1)} \tag{3}$$

where h is the current altitude and h_1 is the reference altitude for the isothermal layer. ρ is the density at the current altitude and ρ_1 is the reference density for the isothermal layer, P is the pressure at the current altitude and P_1 is the reference pressure for the isothermal

layer, R is the universal gas constant, g is the gravity acceleration at the current altitude and T is the current temperature for the layer.

For the gradient layers however temperature, pressure and density all have to be defined for the given altitude. Equations 4, 5, and 6 show the variance over altitude for the gradient layers.

$$T = T_1 + T_h(h - h_1) \tag{4}$$

$$\frac{P}{P_1} = \left(\frac{T}{T_1}\right)^{-\left(g/T_h R\right)} \tag{5}$$

$$\frac{\rho}{\rho_1} = \left(\frac{T}{T_1}\right)^{-\left[\left(g/T_h R\right)+1\right]} \tag{6}$$

with all the terms defined as in Equation 3 but T is the temperature at the current altitude and T_1 is the reference temperature for the gradient layer, T_h is the temperature gradient.

For convenience the standard atmosphere model has been calculated and placed into tables. These tables have temperature, density, pressure and other properties verses altitude. The common used standard was published in 1976 which is contained in a table in the Appendix. Originally published in 1958 by the U.S. Committee on Extension to the Standard Atmosphere and updated in 1962, 1966 and finally 1976. Figure 3 shows temperature, density and pressure verses altitude plots.

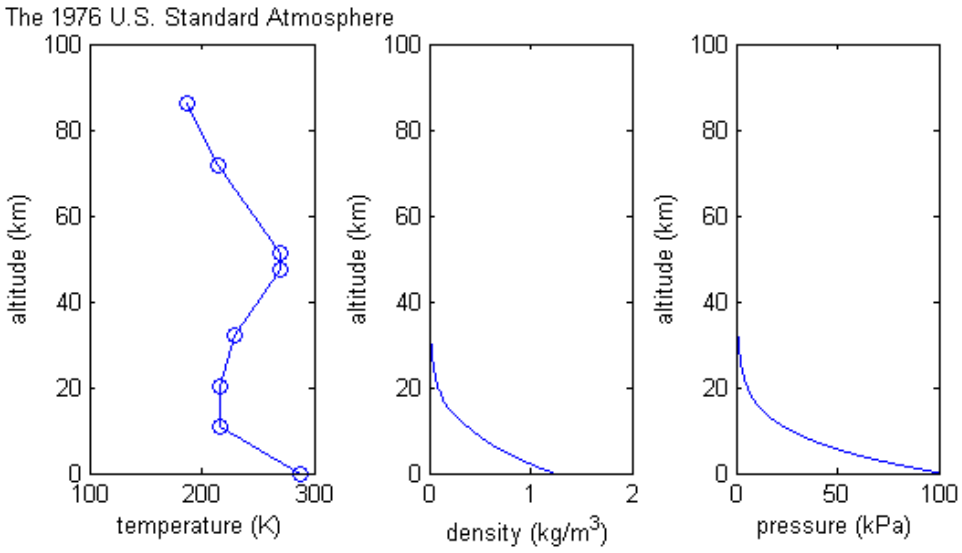


Figure 3: 1976 Standard Atmosphere Profiles [36]

2.3.2 Gravity and Earth Model

The two main models for the earth are a spherical one or an oblate spheroid one. Each of these models has a gravity model associated with it. The simplest model is that the earth is spherical. The spherical model means that the earth has a constant radius and has no flattening making the surface uniform. This means that gravity can be measured as a function of radius from the center of the earth.

$$g = \frac{\mu}{r^2} \tag{7}$$

where μ is the gravitational constant of earth and r is the radius from the center of the earth to the vehicle.

This model assumes that the mass of the vehicle is negligible in comparison to the mass of the earth and gives a uniform density across a set altitude regardless of placement on the earth.

The other model having an elliptical surface was most recently defined by WGS-84 (World Geodetic System – 1984) [26], which is the currently valid scheme used by the Global Positioning System (GPS). This model has an oblate earth meaning that the poles are flattened. It has a major and minor axis for the earth which is placed into the model. MATLAB® currently has a built in function to use this model. The gravitation model associated with WGS 84 is the 1996 Earth Gravitational Model (EGM96). This model is designed as a group effort by the National Imagery and Mapping Agency, NASA Goddard Space Flight Center and the Ohio State University. Below is an image showing the geoid heights across the surface of the earth.

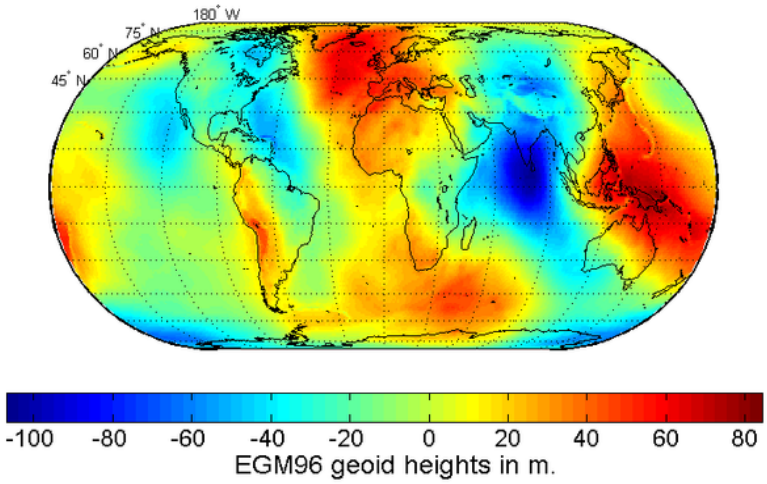


Figure 4: EGM96 Gravity Model [26]

2.4 Highlighted Vehicles

Along with multiple environmental models there are also multiple vehicles that could be used in this problem. These vehicles are explored below.

Figure 5: Minotaur III Launch [27]

The Minotaur is an US rocket transformed from the LGM-118 Peacekeeper Missile. Its mission is to carry heavy payloads over a long range by a suborbital launch.

Table 1: Minotaur III Data [28]

	Stage 1	Stage 2	Stage 3
	SR118	**SR119**	**SR120**
Burn Time (sec)	83	54	62
Thrust (kN)	1607	1365	329
Mass Initial (kg)	53070	27800	8200
Mass Empty (kg)	4086	2900	1400
Diameter (m)	2.36	2.35	2.35
Length (m)	7.72	5.4	2.3
Shroud (stage 3) – 338 kg Payload – 1248 kg Nose Radius – 1 m			

Figure 6: Space Shuttle Launch [35]

NASA's Space Shuttle went operational in the 1980's and is now schedule to be retired in 2010. This vehicle's missions include reaching an orbital trajectory and a re-entry process on its return, usually carrying people and supplies to and from the International Space Station.

Though the Space Shuttle's trajectory is not of great interest to the Air Force currently, the amount of data on its trajectories and heating profile is vast. This will allow for validations of the tool to be completed by making comparisons to the Shuttle model that will be discussed in Section 2.6

Table 2: Space Shuttle Data [35]

	Boosters	**External Tank**	**Orbiter**
	2 Boosters	3 SSMEs	2 OME
Thrust	2,800,000 lbf each	1,225,704 lbf total sea level	53.4 kN (12,000lbf) vaccum
Specific Impulse	269 s	455 s	316 s
Burn Time	124 s	480 s	1250s
Fuel	Solid	LOX/LH2	MMH/N_2O_4
Height	184 ft		
Diameter	28.5 ft		
Mass	4,470,000 lb		

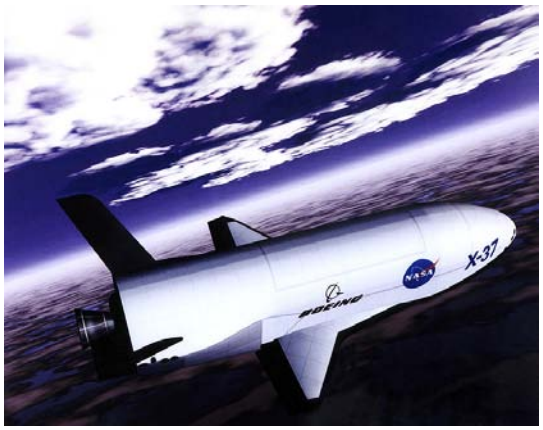


Figure 7: Early artist's rendition of X-37 [38]

Designed by Boeing to test future launch technology for orbit and re-entry, the X-37 is a reusable robotic spacecraft originally designed to launch from the Shuttle cargo bay was redesigned for a Delta IV or comparable rocket.

Table 3: X-37 Data [38]

X-37	
Length	27 ft 6 in
Wingspan	15 ft
Weight (Loaded)	12,000 lb
Power	Rocketdyne (Pratt & Whitney)

Figure 8: Simulated in-flight View of X-33 [37]

Designed by Lockheed Martin as a tool to test capabilities required of a single stage to orbit reusable launch vehicle, the X-33 project was cancelled in 2001 but Lockheed Martin continues research on it.

Table 4: X-33 Data [37]

X-33	
Height	20 m
Mass	285,000 lb
Engine	2 J-2S Linear Aerospikes
Thrust	410,000 lbf
Fuel	LOX/LH_2

Figure 9: Minuteman Launch [29]

Currently the main missile for Global Strike Command, the Minuteman carries a nuclear warhead from a silo, using 3 stages to get to the target.

Table 5: Minuteman Data [29]

	Stage 1	Stage 2	Stage 3
Propulsion	TU-122	SR-19-AJ-1	SR73-AJ/TC-1
Weight	78,000lb		
Length	59 ft 9.5 in		
Diameter	5 ft 6 in		
Range	8100 miles		
Ceiling	700 miles		
Speed	15000 mph		

2.5 Launch Sites

Different vehicles are launched from different areas, thus it is important to examine diverse launch sites. It is important to have multiple launch sites in order to reach targets efficiently. Table 6 lists the most likely launch sites used by the United States. Figure 10 shows a map of all of the potential launch sites. The launch data came from previous cases ran by older tools.

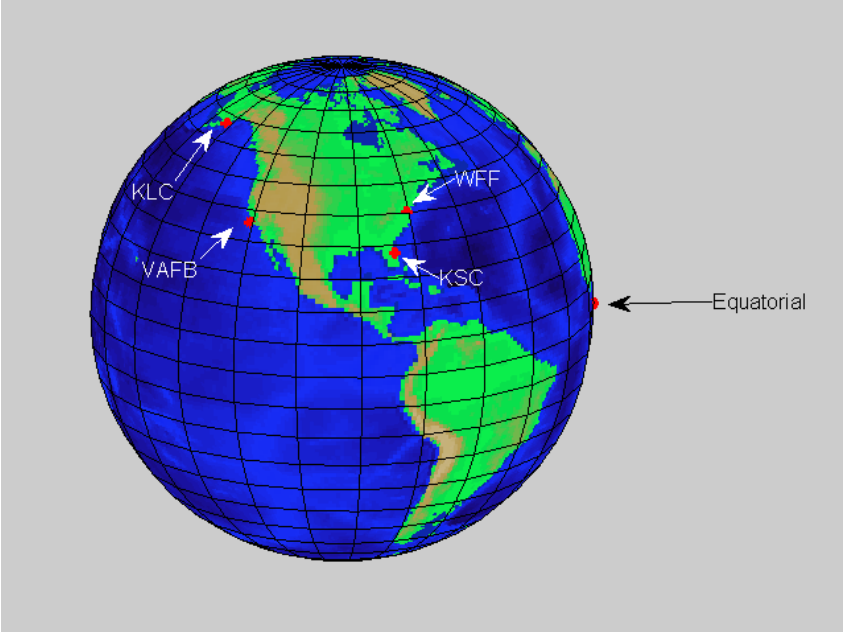


Figure 10: Map of Potential Launch Sites

Table 6: Launch Site Data

Launch	Latitude	Longitude	Azimuth	
VAFB	34.58	-120.63	170	270
KSC	28.39	-80.6	35	110
WFF	37.84	284.51	90	160
KLC	57.44	-152.34	--	--
Equator	0.00	0.00	170	270

Vandenberg Air Force Base – VAFB

VAFB is a military installation responsible for satellite launches for both military and commercial operations. It is also a test site for ICBMs (Intercontinental Ballistic Missiles) to include the Minuteman.

Kennedy Space Center – KSC

KSC is NASA's main launch facility best known for housing and launching the Space Shuttle. Connected to KSC is Cape Canaveral Air Force Station. This has been and currently is the launch site for the Delta, Atlas and Titan rockets. In the future it will also be the launch site of SpaceX's Falcon 9.

Wallops Flight Facility – WFF

WFF is located on the eastern shore of Virginia and is controlled by NASA Goddard as primary a research launch site, launching sounding rockets and small suborbital and orbital rockets.

Kodiak Launch Complex - KLC

KLC is located on Kodiak Island in Alaska and is controlled by the Alaska Aerospace Development Corporations. It is a commercial launch facility for suborbital and orbital launch vehicles.

Equatorial Launch

An equatorial launch site is ideal for putting satellites or orbiters into an equatorial orbit due to the fact that the vehicle can be directly launched into that orbit as opposed to having to perform a maneuver once in space and takes advantage of earth's rotation minimizing fuel requirements.

2.6 Previous Models

There are multiple methods that have been used in the past as well as current methods that generate trajectories for various vehicles. Some create optimized trajectories, others show feasible solutions.

2.6.1 Shuttle Model

The Space Shuttle model described below is applicable to this research because a lot of previous research for lifting bodies comes from Space Shuttle data. In order to compare the data the model must be understood.

The trajectory is calculated from the drag acceleration envelope seen in Figure 11, which included limits such as: surface temperature, dynamic pressure, load, and glide limitations. [16]

Along with implementing constraints in this model the trajectory is broken up into phases of flight. The three phases are: Temperature control, Equilibrium glide and Constant drag.

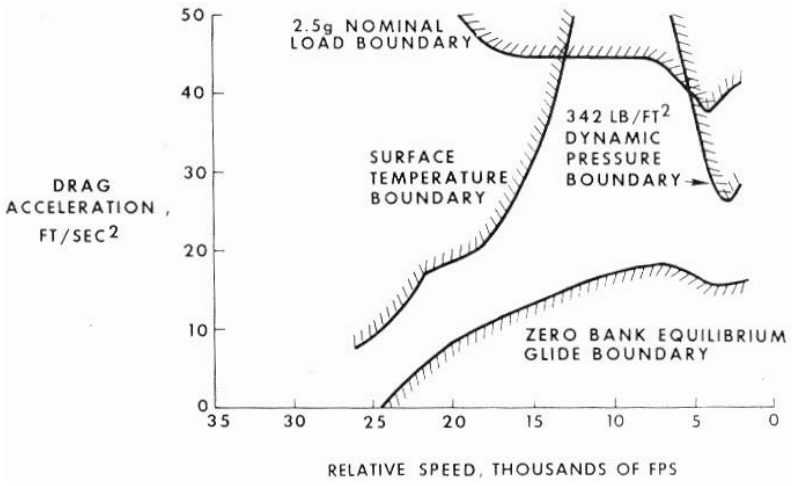


Figure 11: Shuttle Trajectory Envelopes [16]

2.6.2 X-33 and X-37 Model

Five main methods for trajectory calculations were tested and compared [15]. Their algorithms include: Baseline guidance [14], Linear quadratic method [11], Numerical predictor-corrector method [39], Drag-energy 3 dimensional method [30] and Quasi-equilibrium glide method [34]. The best being Quasi-equilibrium glide which led to Evolved Acceleration Guidance Logic for Entry (EAGLE) is shown in Figure 12.

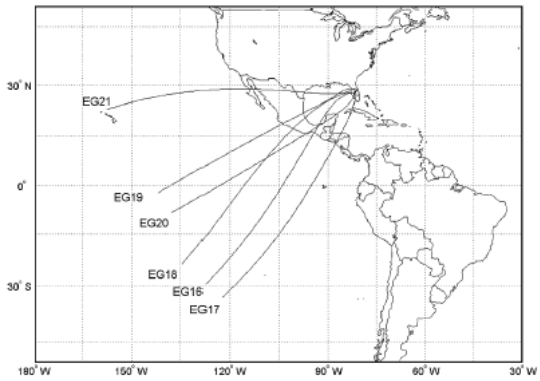


Figure 12: EAGLE Ground Track [33]

Figure 13 shows how some of the previous models can use waypoint guidance to calculate the trajectory. The problem is broken into sections and guided from one section to the next solved not as a continuous problem but as a segmented optimization.

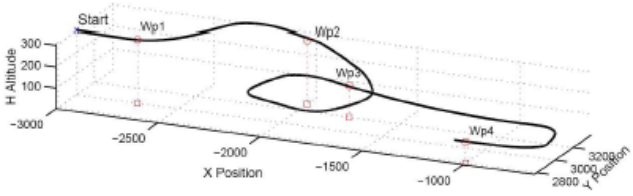


Figure 13: Waypoint Guidance [30]

2.6.3 Currently Used Methods

There are currently two methods used by the government to optimize trajectories, POST (Program to Optimize Simulated Trajectories) and OTIS (Optimal Trajectories by Implicit Simulation). POST was written by Lockheed Martin Astronautics in a joint effort with NASA Langley Research Center. POST is a two point boundary value solver that uses a direct shooting method to solve the equations of motion for the trajectory. This means that the solver takes a set of initial conditions and integrates forward in time to see if the final conditions are met. If they aren't, then the initial conditions are changed and the integration is done again. This method is very computationally expensive thus taking a long time to find an optimized trajectory. POST is also not easily capable of incorporating waypoints or no-fly zones into trajectories.

OTIS was written in the late 1980's by Boeing and NASA Glenn Research Center. It has two methods for solving trajectories. It can do a direct shooting method similar to that of POST where it integrates forward in time from initial conditions. It also has a collocation method, where it fits a polynomial to the problem, allowing the trajectory to be solved at nodes along the trajectory as opposed to just at the endpoints.

2.7 GPOPS

The tool used for this research is GPOPS (General Pseudospectral OPtimal control Software). This tool uses a nonlinear problem solver (SNOPT or IPOPT) to solve a set of differential equations. It has a collocation method allowing the problem to be solved at a user specified number of nodes. Though not specifically designed to solve trajectories, it is designed to solve multiple phase optimal control problems. The tool has

the user define a cost function, the dynamics or differential equations, connections between phases, and sets up limits and guesses for each of the states and controls. [4]

Each GPOPS case is setup as a group of scripts. The four foundational scripts are: a main script; a DAE script (dynamics); a connect script; and a cost script. A brief description of each script is given below since these are key elements to the mission planning tool.

Main Script

The main script holds the framework for the case. It is setup into sections based off the number of phases a problem contains. A phase is defined by a distinction in the states, controls, equations of motion or cost functions. This structure allows the user to implement different phases to be used in one optimal control problem. Each phase has a select number of nodes assigned to it. Nodes are the number of points that will be passed into the NLP solver. This number is crucial because it must be large enough to catch the general solution but not too large to cause numerical divergence or have unneeded calculations.

Each phase has a definition section in which minimums and maximums are defined for time, states, and controls. Both time and control allow for a minimum and maximum value for both the initial and final value. The states had minimum and maximum inputs for the beginning, along the section, and the end. A path constraint will be defined separately in the DAE script (described next), but the minimum and maximum values are set in the main script.

In order to solve the solution to the dynamics that will be defined in DAE, an educated guess is needed to input as a rough solution. This guess includes time, states and controls. It consists of as many points as the user has. The guesses need to be within the bounds previously set and their validity highly impacts the run time of the code to find a solution. Between each phase a connection matrix is defined which is passed into the Connection script and connects the states and controls of one phase to the next phase. The equations for this connection are in the Connection script. The Main script just sets the minimum and maximum values for the equations.

DAE (Differential algebraic equations) Script

The DAE script consists of the dynamics for the optimal control problem. Each phase can have its own dynamics or all of the phases could be under one set of differential equations. Each state requires a first-order differential equation to be provided to the NLP solver. These equations of motion can be very complicated or very simple and GPOPS can handle both. Along with the dynamics the DAE script includes any path constraints. There isn't a limit to the number of constraints. These constraints can be written in terms of both states and controls.

Connect Script

The Connect script contains the final states from the phase before and the initial states for the phase after. From the before and after states, equations are written to transform one state to another or to allow for gaps in the stages between phases. The easiest example of this is in a multiple stage launch problem. At the end of each phase a

portion of the rocket is dropped and thus the mass will have a difference between the end of stage 1 or phase 1 and stage 2 or phase 2.

Cost Script

The Cost script creates the cost function for the optimal control problem. It includes 2 terms: the Lagrange and Mayer functions as defined in Equation 1. There can be a separate cost function for each phase, or a sum of weighted cost functions for each phase to create a total cost function.

2.8 Previous AFIT Research

Major Timothy Jorris completed a dissertation entitled "Common Aero vehicle Autonomous Re-entry Trajectory Optimization Satisfying Waypoint and No-Fly Zone Constraints" [22]. His research included trajectory generation, waypoint satisfaction, threat or no-flyzone avoidance and used a GPM. William Karasz continued Major Jorris' research with "Optimal Re-entry Trajectory Terminal State Due to Variation in Waypoint Locations". [24] This research will be an expansion upon their work. Using Karasz's non-dimensional equations of motion for the re-entry section allows all of his research on waypoints to be directly applied to that section of the mission planning tool. Their work on the application of GPOPS is also directly applied in this research.

2.9 Summary

Since the 17th century optimal control problems have been posed in all shapes and sizes and this field has been shaped by famous mathematicians over the years. The advancements in this field have greatly been formed by the technological advancements

of the day. Starting with problems that can only be solved by hand to advanced nonlinear programming solvers, computerized tools are driving the current field of optimal control. The application of optimal control theory to trajectories seems natural. Optimizing trajectories ties with optimizing missile systems and a more optimized military. In civilian operations, optimized equates to less money. That's why from the background research provided in the previous chapter, the mission planning tool objectives which will be described in the next chapter is a forward progression in optimal control.

III. Methodology

3.1 Generic Problem Statement

The overall objective of this research is to set up the foundations for a tool to compute the optimal trajectory. This foundation will calculate an optimized trajectory with settable parameters to allow for manipulation of the trajectory constraints. The trajectory will be broken up into phases with different equations of motion to get from a launch site to a target site. The goal is to obtain a rapid method for generating optimized trajectories. This will also show how the Gaussian psuedospectral solver, GPOPS, can be used on real life problems.

This solver is set up in a modular setting in order to allow for multiple vehicles, launch and target sites, earth and atmospheric models and different constraints to be placed. The trajectory created will be optimized for minimum flight time.

3.2 Mission Assumptions

The assumptions for this mission are broken up into three categories: overall assumptions which apply to each phase for any configuration; phase assumptions that vary based on the phase the mission is in; and environmental assumptions. These are assumptions that are specific to an atmospheric or earth model. Listed below are the specific assumptions for each category.

3.2.1 Overall Assumptions

1. Vehicle is a point mass
2. All mass separations are instantaneous

3. Lift and Drag Coefficients are constant
4. Launch and Target Site Specified

3.2.2 Phase Assumptions

Launch

1. Thrust is Constant – No throttling
2. Only Control – 3 angles for thrust vectoring

Re-entry

1. No Thrust
2. Bank angle only control

Terminal

1. No Thrust
2. No Control

Model Assumptions

1. There are only two atmospheric possibilities, the simple exponential and standard atmospheric model
2. There are only two possible earth models, spherical and WSG-84
3. The gravity models go with the earth model assumed (spherical to inversely proportional to radius squared and WSG 84 to EGM96)

3.3 Phases

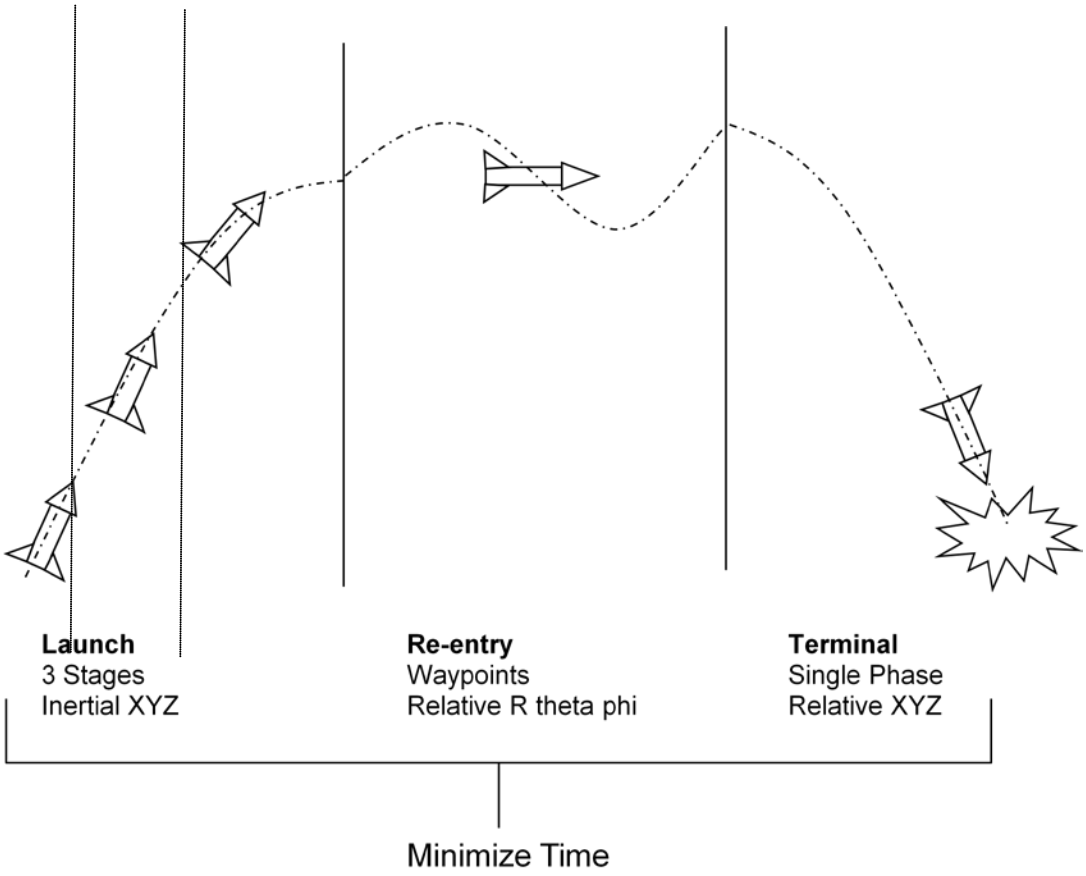


Figure 14: Phase Description

The problem is broken up into 5 phases, 3 launch phases (1 for each stage), a re-entry phase and a terminal phase as shown in Figure 14. An exoatmospheric phase was considered, but initially to keep the problem simple, it is assumed re-entry conditions are met at the end of the launch phase. Thus, there are 3 sections in this problem launch, re-entry and terminal. Each section has a different set of dynamics and is in a different reference frame. The launch section could be set up as 1 or 3 phases but 3 phases allow for easier transition between mass separations at each phase. There is one unifying cost function for the entire problem which is across all the phases, minimizing time.

3.3.1 Reference Frame

There are four reference frames used in this mission.

1. $\hat{e}$ is the inertial planet fixed reference frame
2. $\hat{e}_1$ is the relative planet fixed reference frame
3. $\hat{e}_2$ is the relative vehicle pointing reference frame
4. $\hat{e}_{,,}$ is the relative velocity reference frame

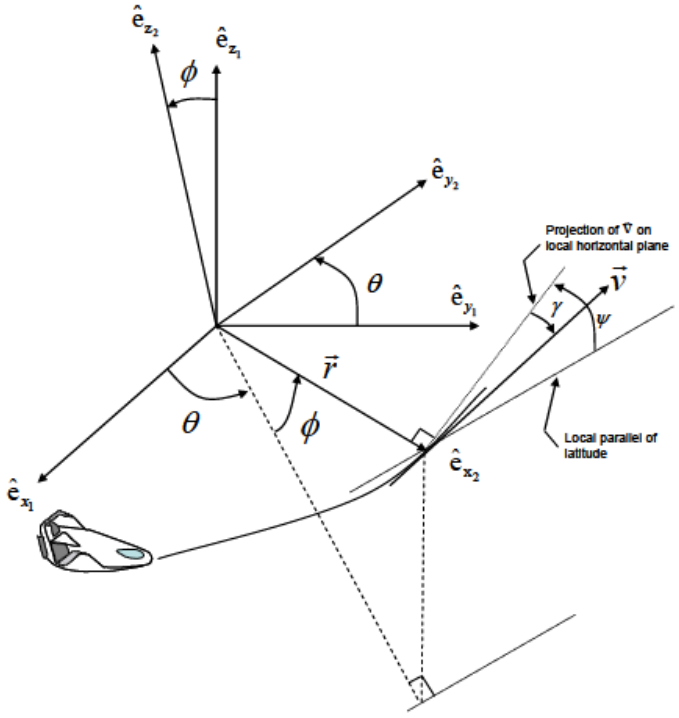


Figure 15: Reference Frame Graphic [17]

Reference frame 1, stays fixed in one spot regardless of time, thus the earth rotates around this reference frame. Reference frame 2 incorporates the rotation of the earth. This frame

stays fixed to the surface of the earth and is used to show the results. Reference frame 3 and 4 are show the angles for the vehicle and velocity directions.

The phases are not all set in the same reference frame. The initial 3 launch phases are set in an inertial earth centered reference frame. The re-entry phase is in a relative earth centered but is in spherical coordinates and the terminal phase is also in relative earth centered coordinates but Cartesian x, y, and z. Below are the equations to transform each reference frame into a relative earth fixed frame ($\hat{e}_1$).

$$\hat{e}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \omega_e \Delta t \end{bmatrix} \hat{e} \tag{8}$$

$$\hat{e}_1 = \begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -\cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \hat{e}_2 \tag{9}$$

$$\hat{e}_1 = \begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -\cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\cos(\psi)\sin(\gamma) & \cos(\psi)\cos(\gamma) & -\sin(\psi) \\ -\sin(\psi)\sin(\gamma) & \sin(\varphi)\cos(\gamma) & \cos(\psi) \end{bmatrix} [\hat{e}''] \tag{10}$$

where θ, ϕ, γ, and ψ are angles defined in Figure 15. $\omega_e$is the rotational rate of the earth and Δt is the time elapsed since $\hat{e}_1$ equals $\hat{e}$.

3.3.2 Equations of Motion

One of the required MATLAB® scripts for GPOPS is DAE. DAE includes the dynamics of the problem which in this case would be the equations of motion (EOM). Each section has a different set of equations because they are in different reference frames and have different forces acting on them. Below are the 3 sections EOMs.

Launch EOMs

$$\dot{r} = v \tag{11}$$

$$\dot{v} = \frac{D + T}{m} - g \tag{12}$$

$$\dot{m} = \text{constant} \tag{13}$$

The states are: radius, r which consists of x, y and z, velocity, v which consists of v_x v_y and v_z and mass, m. D is the drag, g is gravity and T is thrust.

After the launch is completed the body has a constant mass, meaning mass is no longer a state in the equations of motion. In order to make the scaling for the problem easier for the solver to handle the dynamics are non-dimensionalized by the radius of earth, gravity at the surface or a combination of the two. The re-entry equations of motion have spherical state variables and are listed below.

Re-entry EOMs [24]

$$\dot{r_p} = {}^{R}V_p \sin\gamma \tag{14}$$

$${}^{R}\dot{V_p} = -D_p - \frac{g\sin\gamma}{{r_p}^2} + r_p {\omega_e}^2 \cos\phi(\cos\phi\sin\gamma - \sin\phi\sin\psi\cos\gamma) \tag{15}$$

$$\dot{\theta_p} = \frac{{}^{R}V_p \cos\gamma\cos\psi}{r_p \cos\phi} \tag{16}$$

$$\dot{\phi_p} = \frac{{}^{R}V_p \cos\gamma\sin\psi}{r_p} \tag{17}$$

$$\dot{\gamma_p} = \frac{L_p r_p}{V_p m}\cos\sigma - \frac{g\cos\gamma}{{r_p}^2 V_p} + \frac{{}^{R}V}{r_p}\cos\gamma + 2\omega_e(\cos\phi\cos\psi) + \frac{r_p}{V_p}{\omega_e}^2\cos\phi(\cos\phi\cos\gamma + \sin\phi\sin\psi\sin\gamma) \tag{18}$$

$$\dot{\psi_p} = \frac{L_p \sin\sigma}{V_p \cos\gamma} - \frac{{}^{R}V_p}{r_p V_p}\cos\gamma\cos\psi\tan\phi + 2\omega_e(\sin\psi\cos\phi\tan\gamma - \sin\phi) - \frac{r{\omega_e}^2}{V_p\cos\gamma}\sin\phi\cos\phi\cos\psi \tag{19}$$

The p subscript indicates that the value is non-dimensional. The states are non-dimensional radius, r_p and non-dimensional relative velocity, ${}^{R}V_p$. ω_e is also non-dimensionalized. The remaining states are angles; longitude θ, latitude ϕ, heading angle ψ and flight path angle γ. D_p is non-dimensional drag, L_p is non-dimensional lift, ω_e is the rotation of the earth and the control is bank angle, σ.

After the re-entry phase the missile takes a ballistic dive to the surface for detonation. Those equations of motion follow.

Terminal EOMs

$$\dot{r} = v \tag{20}$$

$$\dot{v} = \frac{D}{m} - g \tag{21}$$

The states are: radius, r is a vector in the Cartesian coordinate frame consisting of x, y and z, and velocity, v which consists of v_x v_y and v_z. D is the drag, m is the mass and g is gravity.

3.3.3 Constraints

In addition to the equations of motion, path constraints can be defined for each phase that the solution must satisfy. Maximum heating and acceleration are crucial to a mission planning tool as structures are only capable of handling certain temperature and structural loads. Below are the equations for heating and acceleration.

$$\dot{\overline{q_s}} = \eta^{1/2} T^{3/2} \tag{22}$$

where $\dot{\overline{q_s}}$ is the stagnation heating, T equals kinetic energy and η equals non-dimensionalize altitude and are defined below.

$$T = \frac{1}{2}\left(\frac{^{R}V^2}{g_0 r_0}\right) \tag{23}$$

where V is velocity, g_0 is the reference gravity and r_0 is the reference radius.

$$\eta = \frac{\rho S C_D}{2m\beta} \tag{24}$$

where ρ is the density, S is the reference area, C_D is the coefficient of drag, m is the mass and

β is the scale height.

The acceleration equations also use both the kinetic energy and the scale height listed in Equations 23 and 24. The acceleration is important for both structural loading of the missile itself and sensitivity of the payload.

$$a_{decel} = 2\beta g_0 r_0 \eta T \sqrt{1 + \left(\frac{C_L}{C_D}\right)^2} \tag{25}$$

where β is the scale height, g_0 is the reference gravity, r_0 is the reference radius, T is the kinetic energy, η is the altitude, C_D is the coefficient of drag and C_L is the coefficient of lift. [17]

3.4 Optimal Control Problem

The total optimal control problem ties the cost function with the equations of motion to get a total problem statement. GPOPS takes the cost function, dynamics and constraints and solves the optimal control problem. The cost function for the problem of a prompt strike is to minimize time, thus the complete optimal control problem can be written as follows:

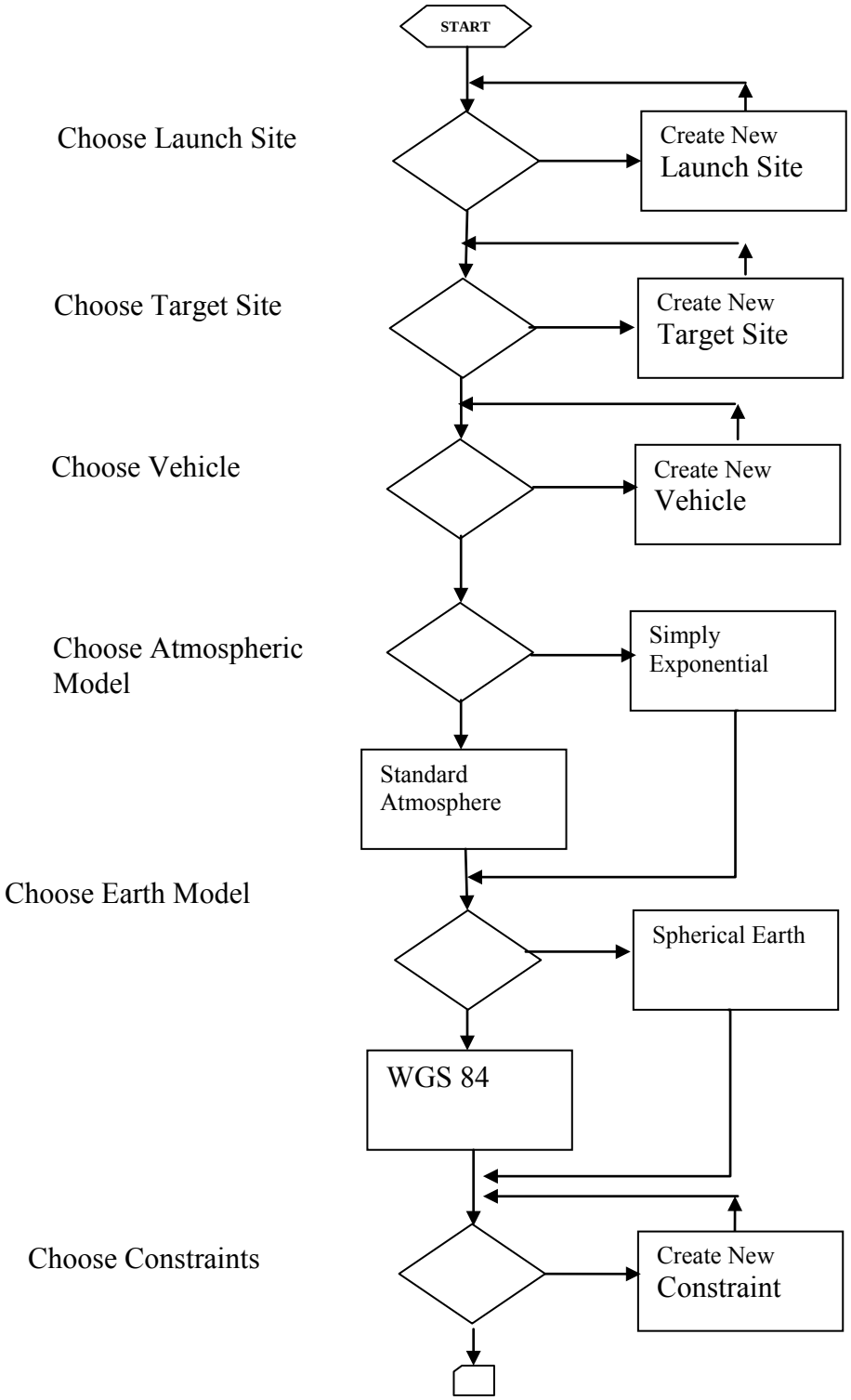
$$\begin{gathered} \text{Min } J = t_f \\ \text{s. t.} \\ \dot{x} = f(x(t), u(t), t) \\ C(x(t_0), t_0, x(t_f), t_f) \leq 0 \end{gathered} \tag{26}$$

where $\dot{x}$ represents the first order derivative of the state vector, x is the states, u is the control, t is time t_0 is the initial time and t_f is the final time. C represents the path constraints.

3.5 Specific Case

To test the capability of this tool, a simple specific case is selected. This hypothetical case is launched from Kennedy Space Center and the desired target is Timbuktu (3° Longitude, 16.5° Latitude). A spherical earth model is used with an inverse square law gravity model and the atmosphere is a simple exponential model. This case is a fairly simplified case but it is a building block to show future capabilities. The specific inputs for each phase are discussed in Chapter IV.

3.6 Mission Planning Tool



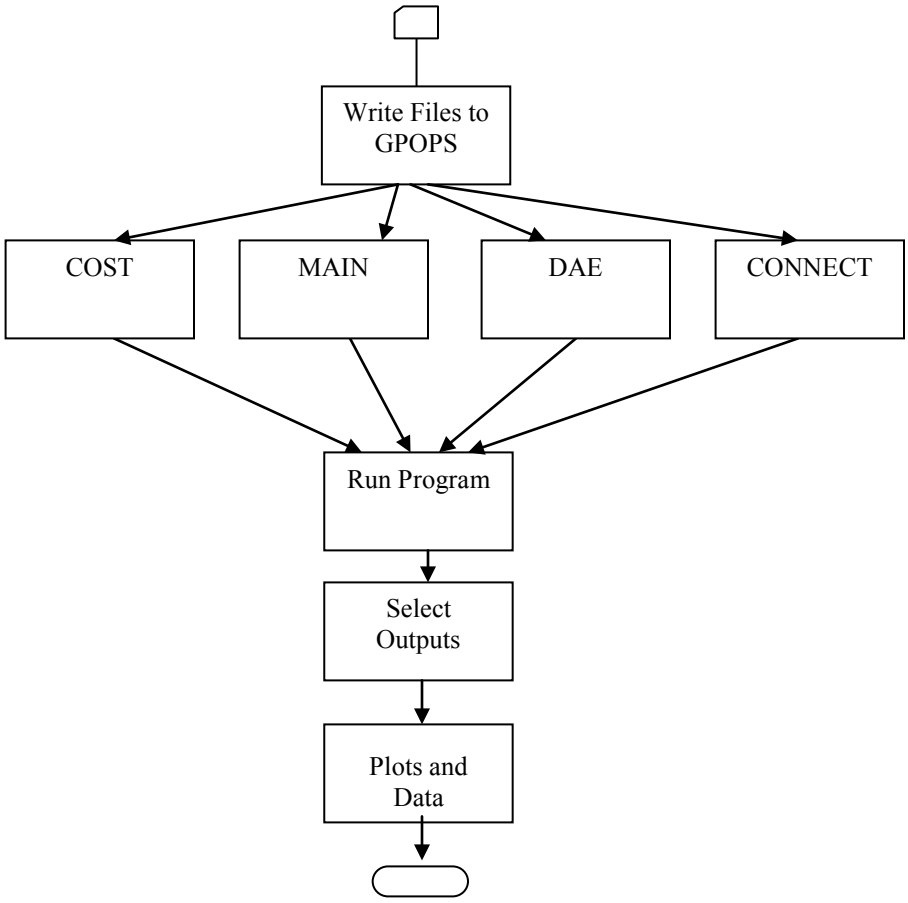


Figure 16: Mission Planning Tool Flowchart

Above is the flowchart for the mission planning tool. It shows the options the user has in the models and allows for future options to easily be added in. This program consists of a frontend that has the model, launch, target and vehicle selection along with any constraint placements. The backend allows for the user to select the data output they want as figures or variables and then the core section is the GPOPS code that the frontend automatically formats.

Figure 17, Figure 18, and Figure 19 show examples of the Graphic User Interface (GUI). This GUI follows the algorithm shown in Figure 16. Figure 17 is the main screen which allows the user to select or create the launch site, target, vehicle and constraints.

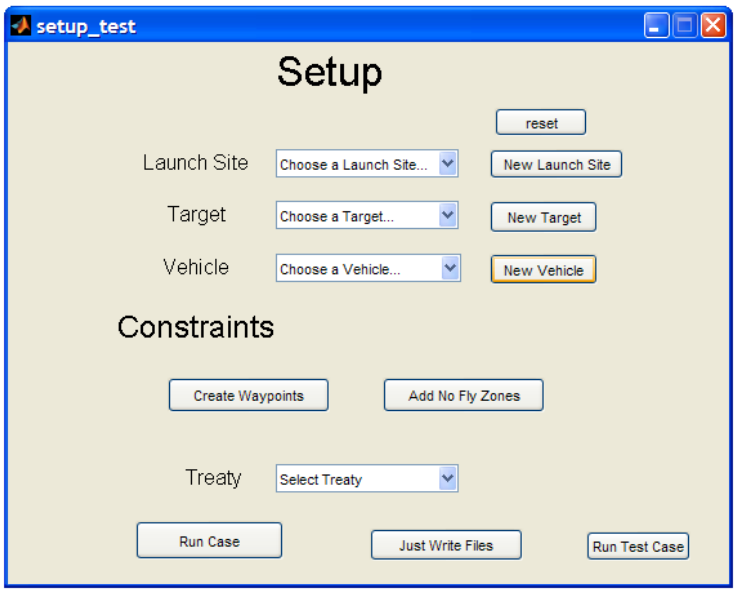


Figure 17: GUI for Inputs

Figure 18 is one of the subset menus. It allows the user to create a new launch site and when run, sets up all of the launch inputs for the GPOPS. Figure 19 shows the backend of the GUI. This is where the user selects what outputs they would like to be displayed to the screen and a filename for the data file.

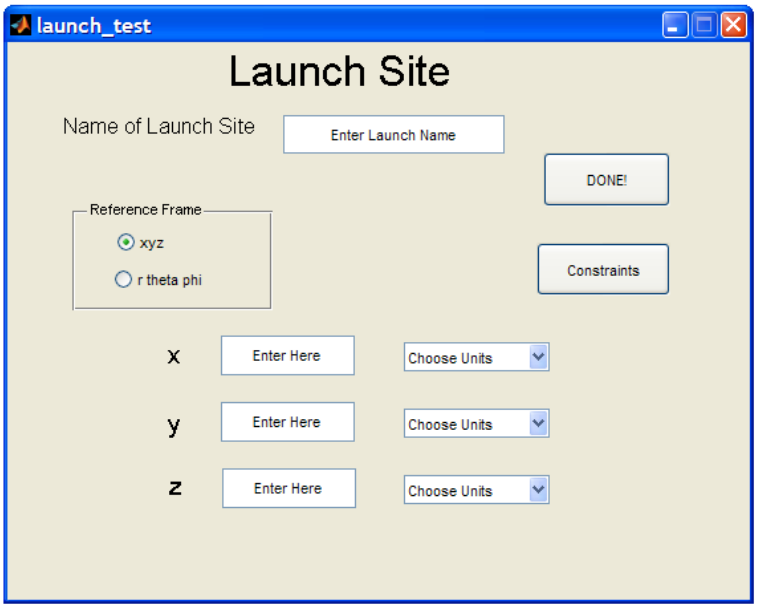


Figure 18: GUI for Launch Site Inputs

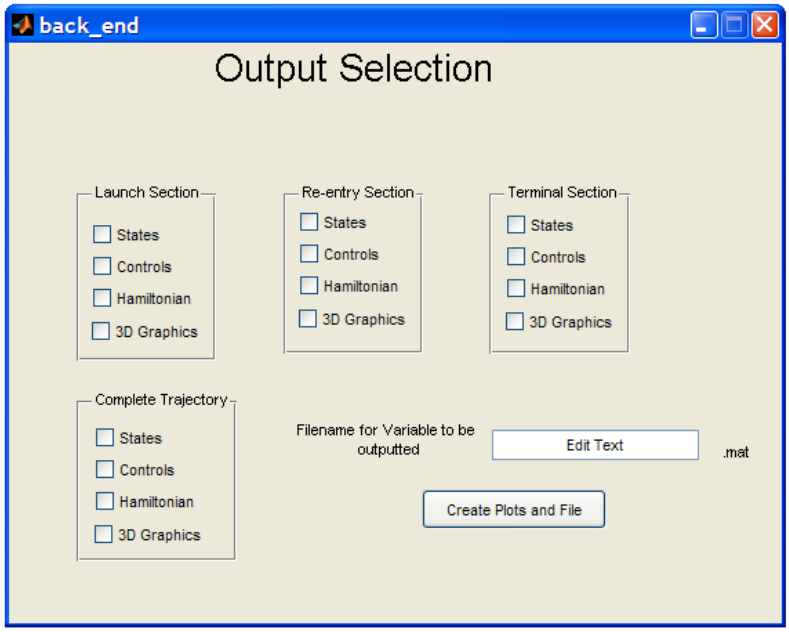


Figure 19: GUI for Output Selection

3.7 Summary

This chapter provided an outline for the methodology followed to solve the optimal control problem presented. Analyzing the dynamics to evaluate the cost function over the set of constraints will give an optimized solution. Tying all of the methods together to apply to the specific case will give the desired results. Finally, the format of a mission planning tool was discussed so the application of this specific case to this model can be seen. The next chapter will go into the results for the specific case described in Section 3.5.

IV. Analysis and Results

4.1 Changes in Specific Case

Originally this research was set up to have a unified cost function for connecting phases. This method ran into a lot of difficulties with finding a convergent solution. The procedure to run an optimal control problem in GPOPS with varying states between phases had yet to be examined (it is accomplished later in the results). This problem has both time and non-dimensionalize time and states in both spherical and Cartesian coordinates. There are also multiple reference frames to be converted between. There are numerous possible problems that could be incurred in connecting phases with different states, for example scaling between could be a big problem (which is why non-dimensionalize dynamic equations were used).

The current method used with re-entry problems in GPOPS to get a converged solution is to vary state constraints to allow for a "close" solution to be found and then iterating closer to the actual constraint. This method is hard to implement on 5 phases, especially when the states vary. More about how this could be accomplished is discussed in Chapter V.

Due to the complications in the unified method, a sectional method was implemented to get solutions. This sectional method breaks the problem into 3 separate problems: launch, re-entry and terminal. The re-entry section still has the same cost function and setup as previously mentioned. The launch section has a new cost function which minimizes the distance to target seen in Equation 27. The terminal phase is solved by stepping forward in time to the target solution from the final value of the re-entry phase.

$$\text{Min J} = \left(\theta_f - \theta_{\text{target}}\right)^2 + \left(\phi_f - \phi_{\text{target}}\right)^2 \tag{27}$$

where θ_f is the longitude at the end of the launch phase and θ_{target} is the target's longitude and ϕ as latitude.

As can be seen in Figure 20 the 3 sections are similar to the phases with different problem statements. Each section is still connected to the section before and after it, this is just accomplished outside of the optimal control problem.

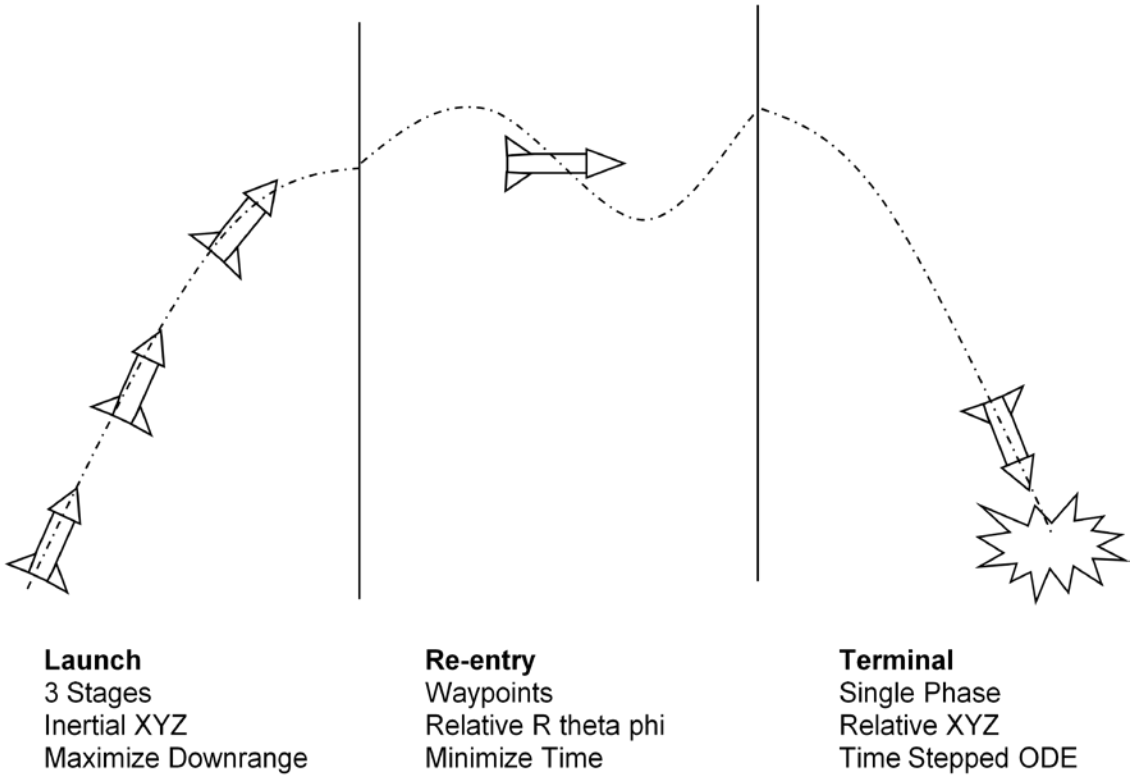


Figure 20: New Section Break Down

Each phase of the GPOPS code requires minimum and maximum values for the states and controls and a guess for them as well. The guess can be the end points or any points along the path. It could even be an entire trajectory. The tables to follow show those values for this problem.

Table 7: Launch States (Section 1)

State	Min	Max	Guess
x	-12756 km	12756 km	916.3 km
y	-12756 km	12756 km	-5535.1 km
Z	-12756 km	12756 km	3033.5 km
V_x	-10 km/s	10 km/s	0.4036 km/s
V_y	-10 km/s	10 km/s	0.0668 km/s
V_z	-10 km/s	10 km/s	0 km/s
m	90589 kg	1194 kg	90589 kg

The guess, min and max values were the same for all three stages of the launch. The guesses are the inertial, initial conditions, the x, y, and z are the position of the launch site and V_x, V_y and V_z are the rotation of the earth seen in Table 7.

Table 8: Re-entry States (Section 2)

State	Min	Max	Guess	Guess
r_p	1	1.2	1.0191	1.02
V_p	.01	1	0.6953	.9
Θ	-1.3064	.0524	-1.3064	0.0472
ϕ	0.2880	0.4853	0.4853	0.2906
Ψ	-6.2832	6.2832	-0.1304	-0.4952
Γ	-1.5533	1.5533	0.3289	0.0007

The guess in Table 8 was generated from the launch and terminal conditions. The values for Re-entry are earth centered relative. The max and min were decided by previous experience

with GPOPS on what conditions allowed for converged solutions and stayed away from singularities in the equations of motion.

Table 9: Terminal Initial Conditions (Section 3)

State	IC
X	6113.16 km
Y	289.02 km
Z	1830.53 km
V_x	.103 km/s
V_y	-.958 km/s
V_z	-.492 km/s

Since the terminal phase is a forward propagation of the equations of motion, the only values needed to solve the problem are the initial conditions shown in Table 9. The values for the Terminal phase are also earth centered relative.

After finding converged solutions for the different sections, the re-entry and terminal section were later linked in one minimum time problem. Creating only 2 sections for this problem a launch section and a re-entry thru termination section. These results would not have been possible without first finding a possible solution in the 3 section method.

4.2 Results of Simulation Scenarios

The new method has been explained in the Chapter 4.1, this section will explore the results of that method.

4.2.1 Launch Results

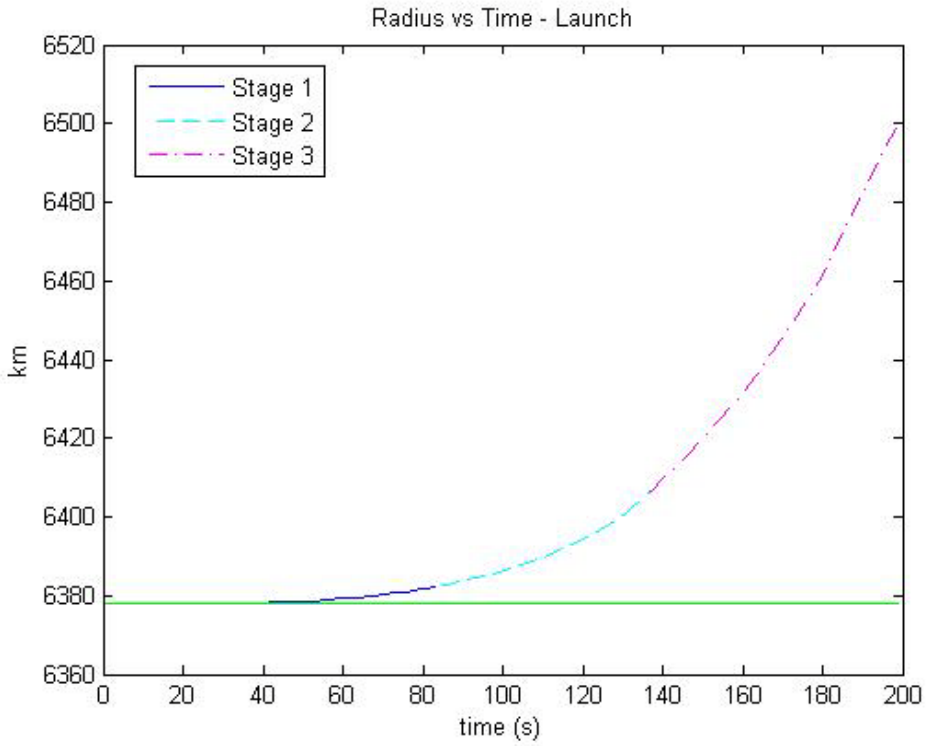


Figure 21: Radius vs. Time for Launch Section

The majority of the energy expended in launch is used to counter the weight of the vehicle and gain altitude. Figure 21 shows the climb in the launch phase. It can be infered from Figure 22 and Figure 23 that the first two stages are primarly used to get the vehicle off the ground. The manuevering for the desired direction is not clear until the third stage. Figure 21 also shows that most of the altitude is gained in the third stage which would also show that it has the most flexibility to be manuevered.

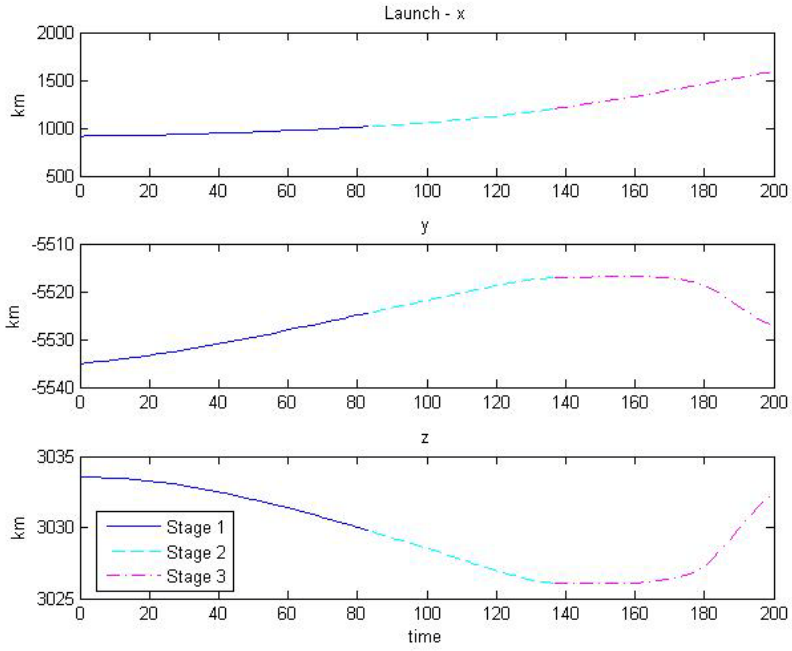


Figure 22: Position for Launch Section

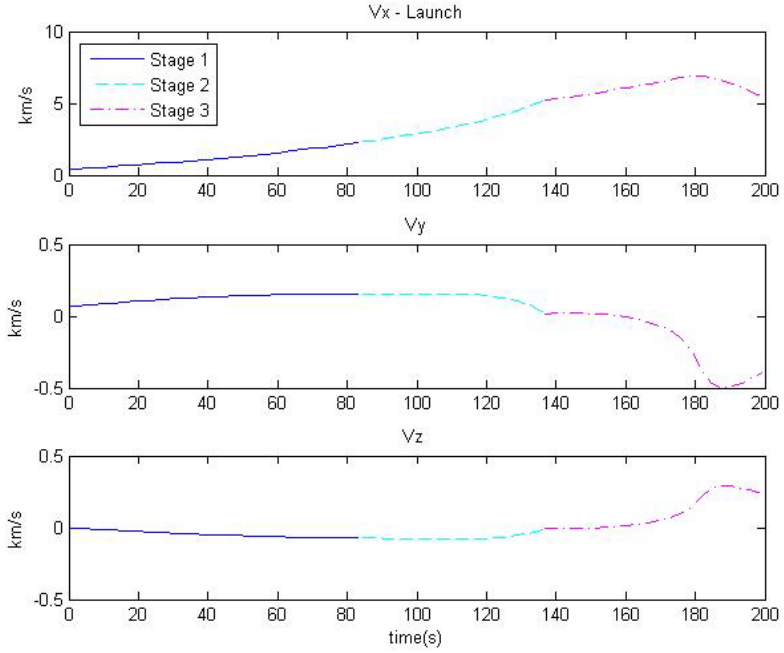


Figure 23: Velocity for Launch Section

The controls for the launch section are the x, y and z components of the thrust unit vector. Thus providing the thrust vectoring for each stage. Figure 24 shows that the majority of the thrust was in the x direction for the largest portion of the launch. This seems logical because most of the thrust is required to get the vehicle to altitude as compared to changing its direction. Figure 25 and Figure 26 show the y and z components of the thrust. The most drastic thrust vectoring changes occur in the 3rd stage.

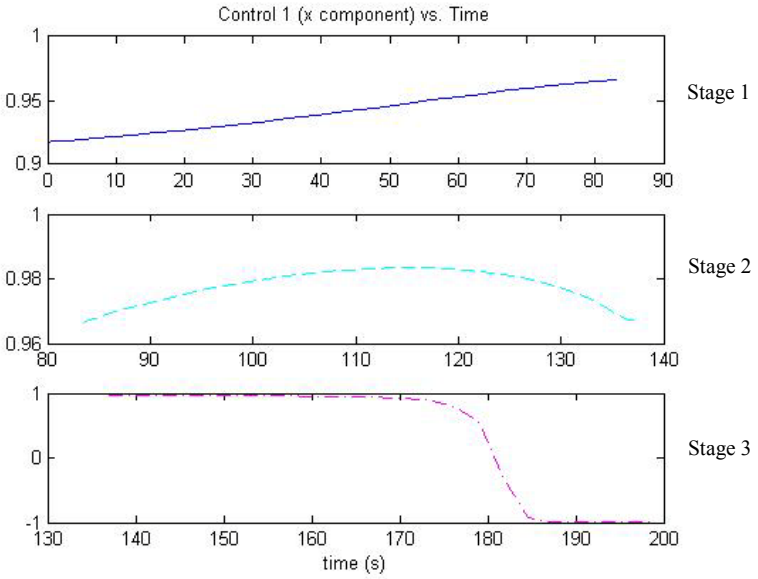


Figure 24: Control 1 Launch

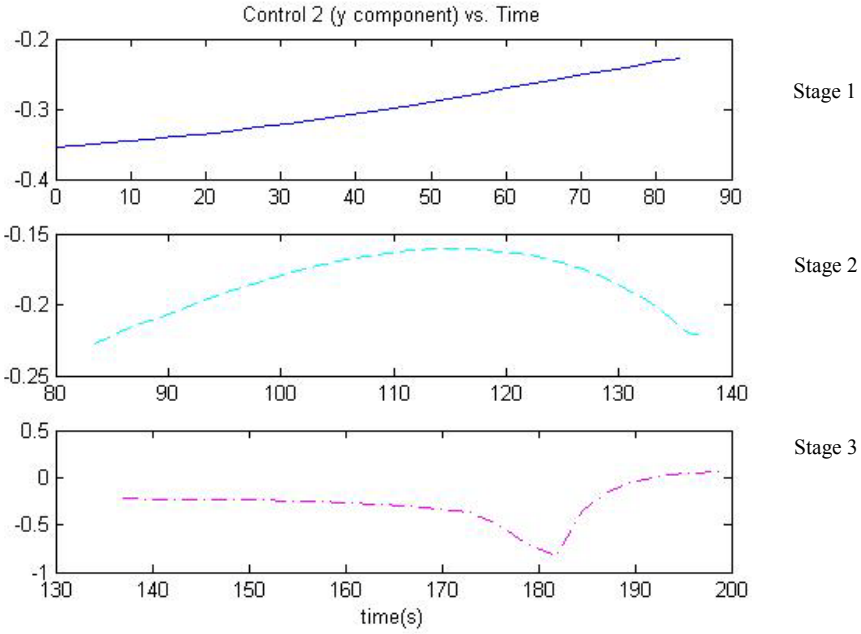


Figure 25: Control 2 Launch

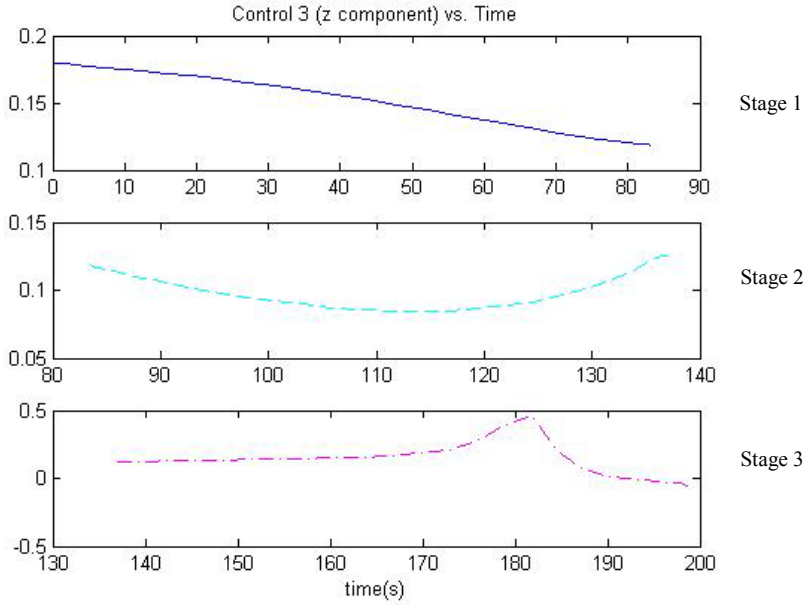


Figure 26: Control 3 Launch

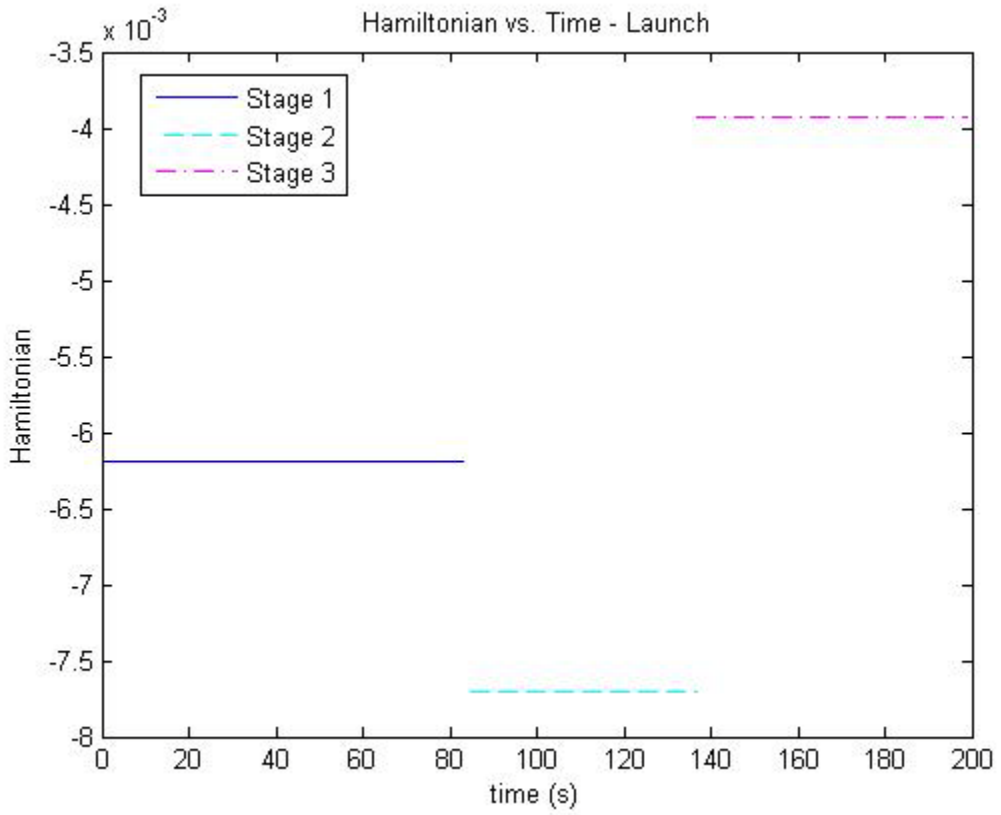


Figure 27: Hamiltonian for Launch Section

Figure 27 shows the Hamiltonian for the launch section. The Hamiltonian for the launch is approximately zero for all three phases which means that the controls and states for this section are feasible. For a fixed final time the Hamiltonian should be zero.

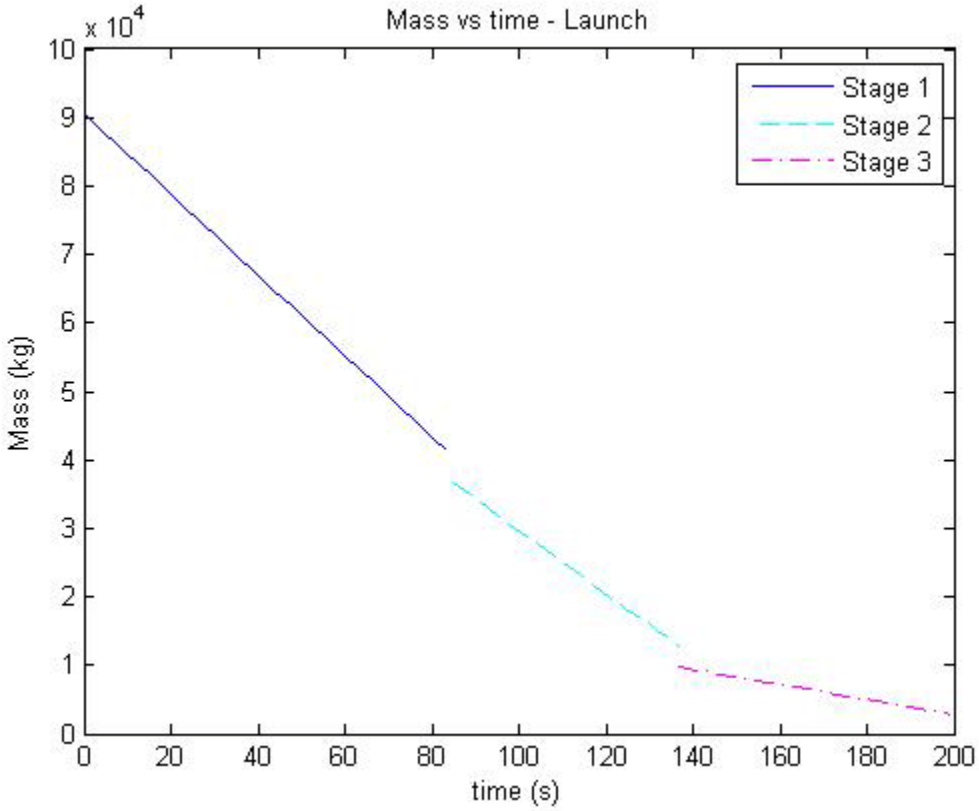


Figure 28: Mass Profile

Figure 28 shows the mass profile for the launch section. The mass state is a set profile regardless of the control or initial and final conditions.

4.2.2 Re-entry Results with Bank Control

With the launch completed the next section is re-entry. The complete re-entry trajectory is show in Figure 29. As can be seen, the re-entry section of the trajectory is the longest portion by both time and distance. Re-entry trajectories tend to be oscillatory, which can be shown in the Figure 30. The velocity also oscillates with the dives and climbs of the trajectory which can be seen in Figure 31.

The states appear feasible and the solution is converged. When the control is examined later in this section along with the Hamiltonian the feasibility of re-entry trajectory is questioned.

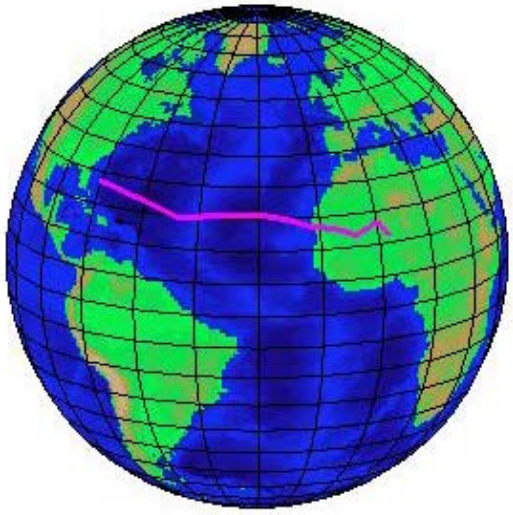


Figure 29: Globe Trajectory – Re-entry Section

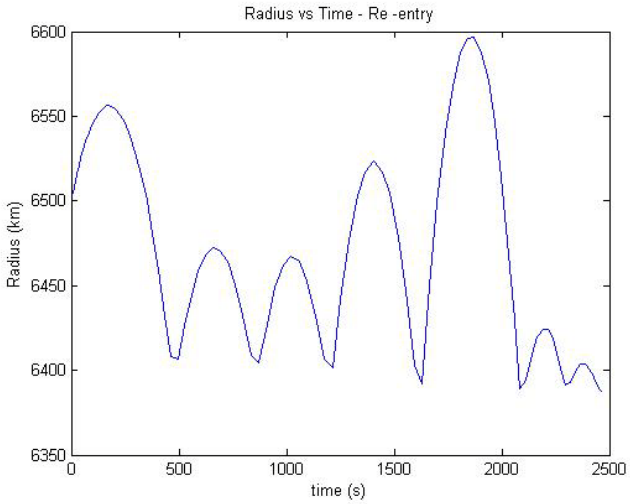


Figure 30: Radius for Re-entry Section

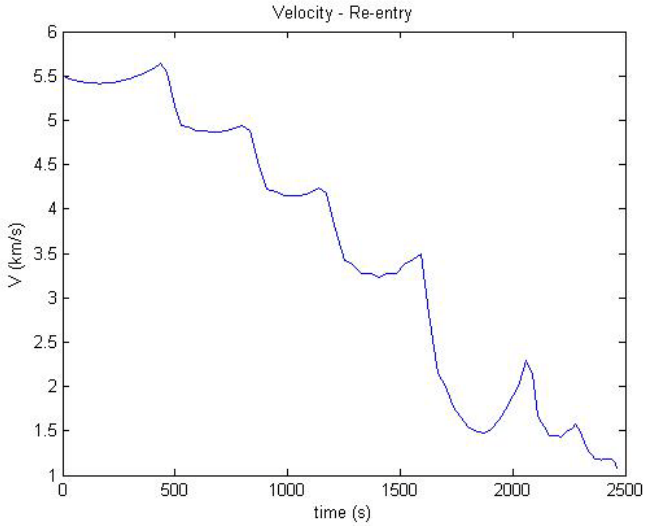


Figure 31: Velocity for Re-entry Section

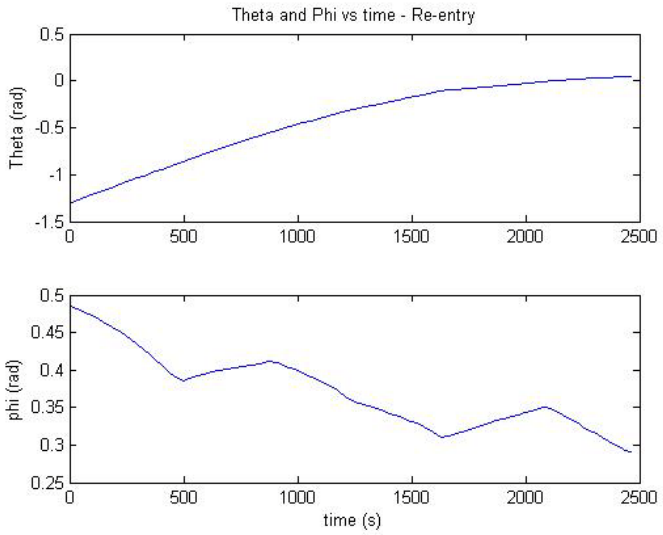


Figure 32: Heading Angles for Re-entry Section

The heading angles shown in Figure 32 have a smooth profile. The velocity angles, shown in Figure 33, presents the velocity angles, showing that the velocity is constantly changing direction in order to meet the requirements for the trajectory. This is due to solving the trajectory at a set number of points and then interpolating between them.

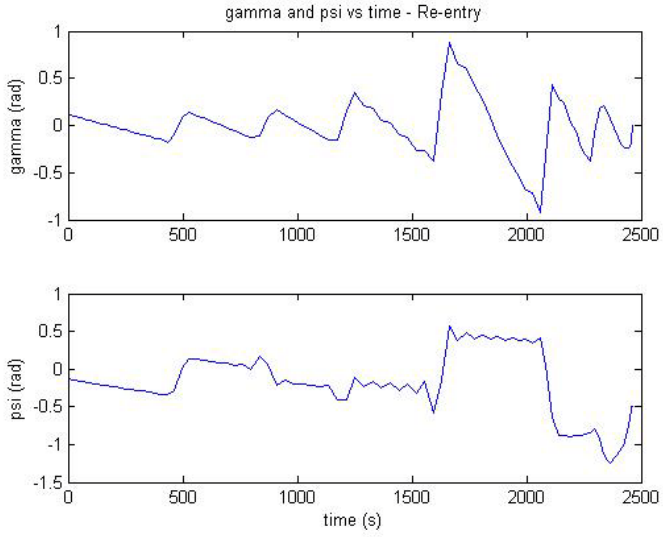


Figure 33: Velocity Angles for Re-entry

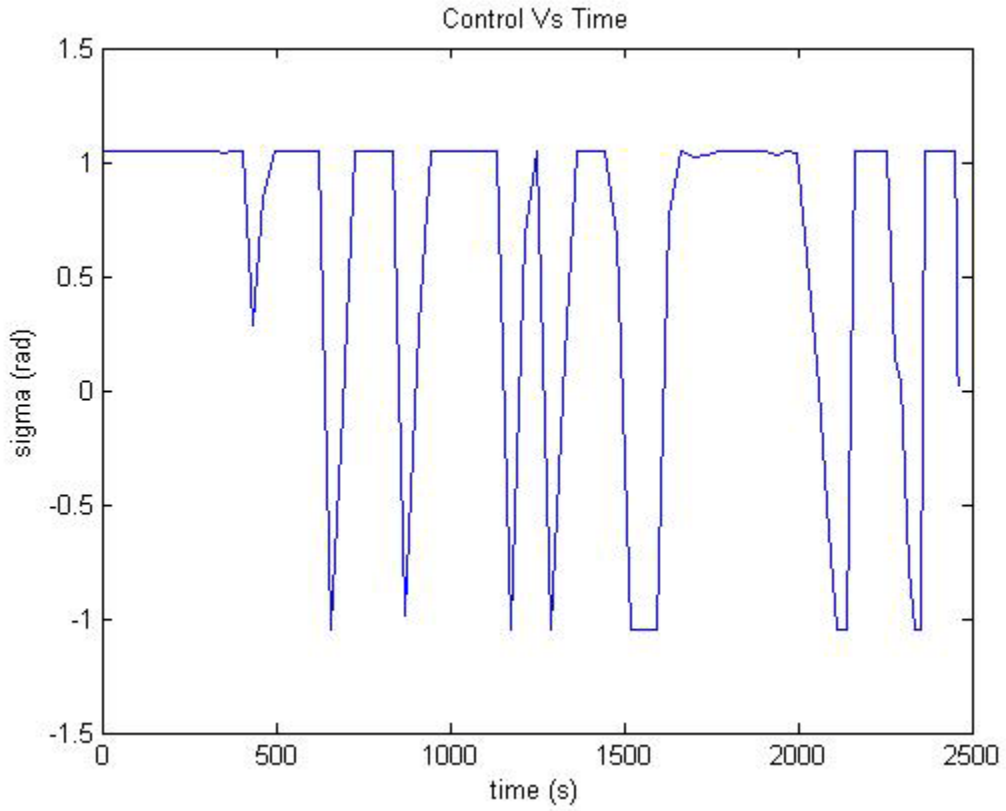


Figure 34: Control for Re-entry

Figure 34 shows an oscillating control profile. Since the only control was bank angle in this case and there wasn't a bank rate constraint, the bank angle was able to jump between positive and negative 180 degrees. Physically the vehicle cannot be at a bank angle of negative 180 degrees one second and then positive 180 degrees the next. There has to be a reasonable transitional period. Figure 35 supports the conclusions drawn from the control. The Hamiltonian doesn't appear stable thus showing that the solution converged upon isn't feasible. This means that the control is solved at select nodes for given values but isn't a continuous profile.

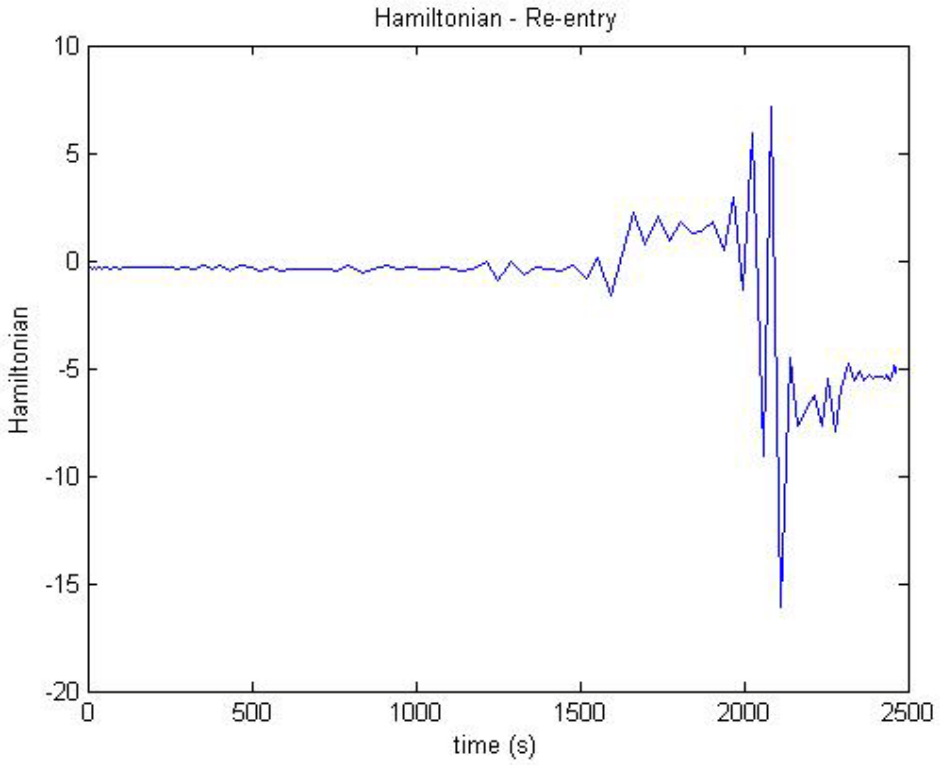


Figure 35: Hamiltonian Re-entry

4.2.3 Re-entry Results with Bank Rate Control

Since the Hamiltonian didn't converge to -1 and the control, bank angle wasn't physically feasible it seemed necessary to increase the order or derivative of the control. The results to follow have a control of bank rate instead of bank angle. Figure 36 shows the new trajectory on a globe. To follow is a comparison of the bank angle control re-entry phase data and a 245 node bank rate control data.

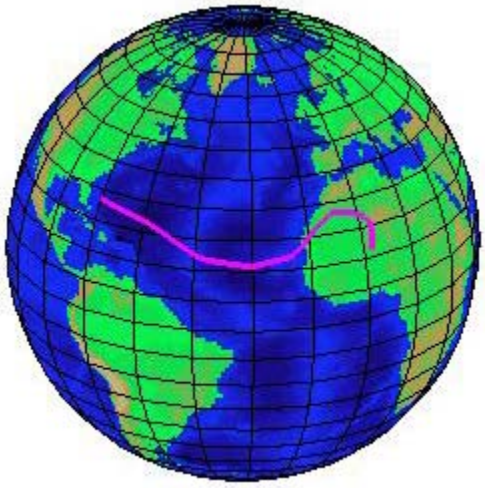


Figure 36: Globe Result for Bank Rate Control

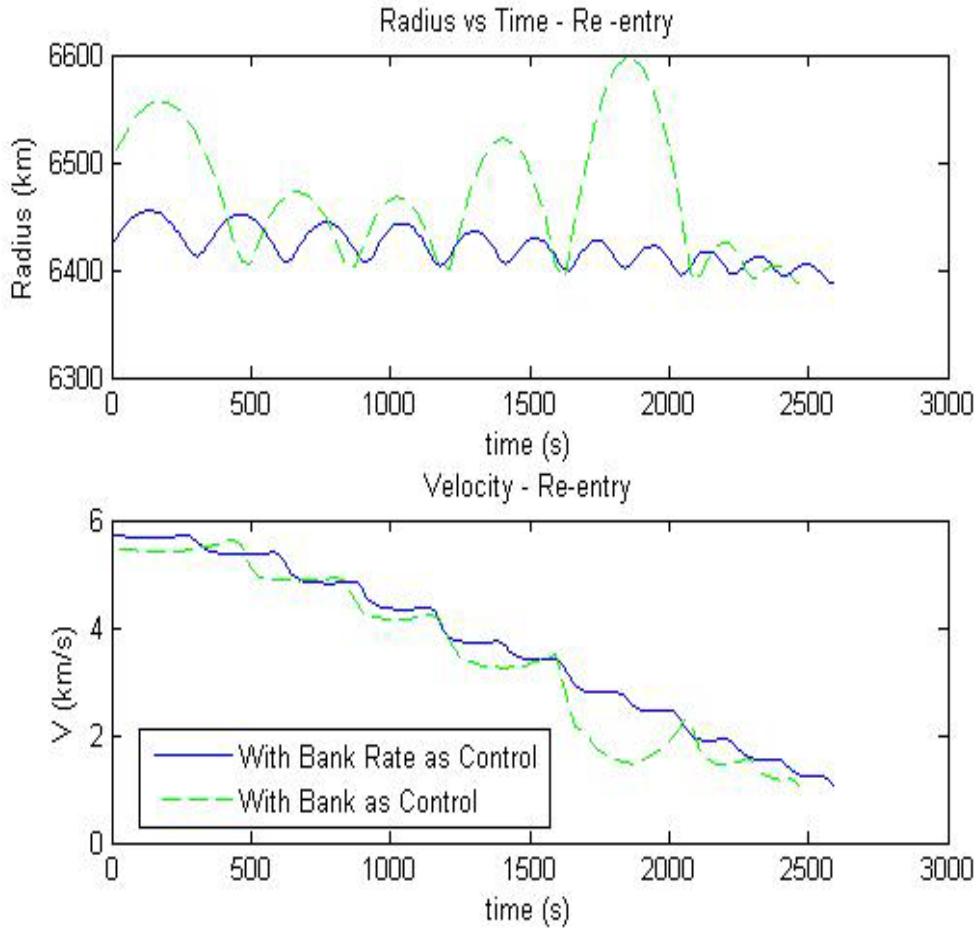


Figure 37: Radius and Velocity Difference in Controls

As can be seen in Figure 37, both the velocity and radius profiles for a bank rate control have milder oscillations than that with only bank angle as the control. A converged solution could not be found for the bank angle control at as high of a nodal count as the bank rate solution which may account for some of the differences but the more stable profile appearance is likely due to the added degree of control.

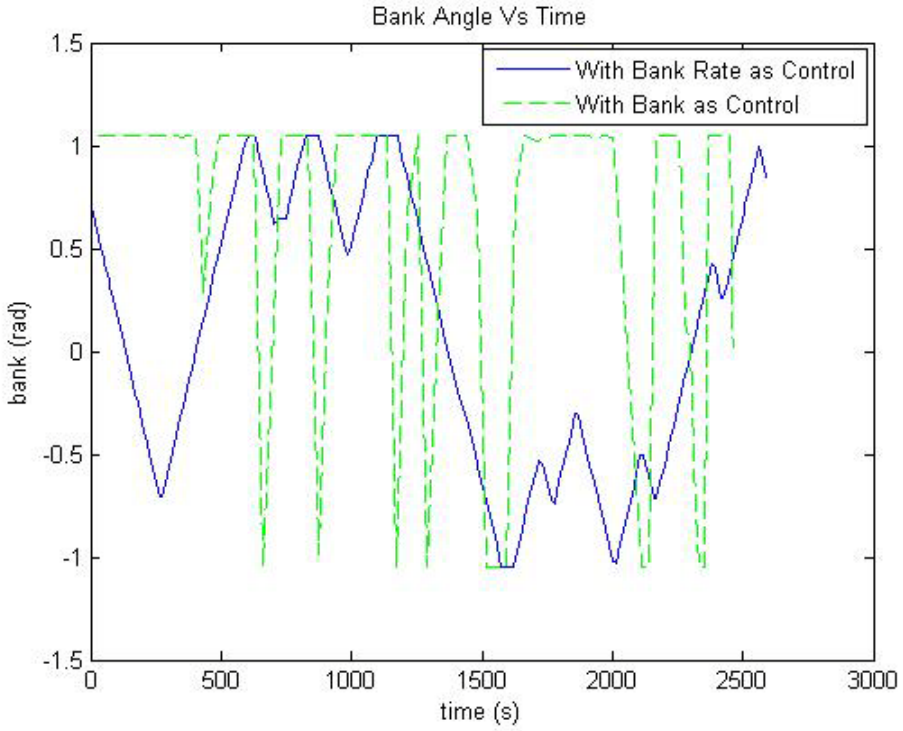


Figure 38: Bank Angle Differences

Figure 38 shows how different having a rate as a control is to having the angle itself as the control. The bank angle appears to be more sinusoidal and the bank rate is more bang bang control as opposed to a bang bang control being the bank angle. The profile for the bank rate control is physically feasible for the missile to fly, thus, making this total trajectory more realistic.

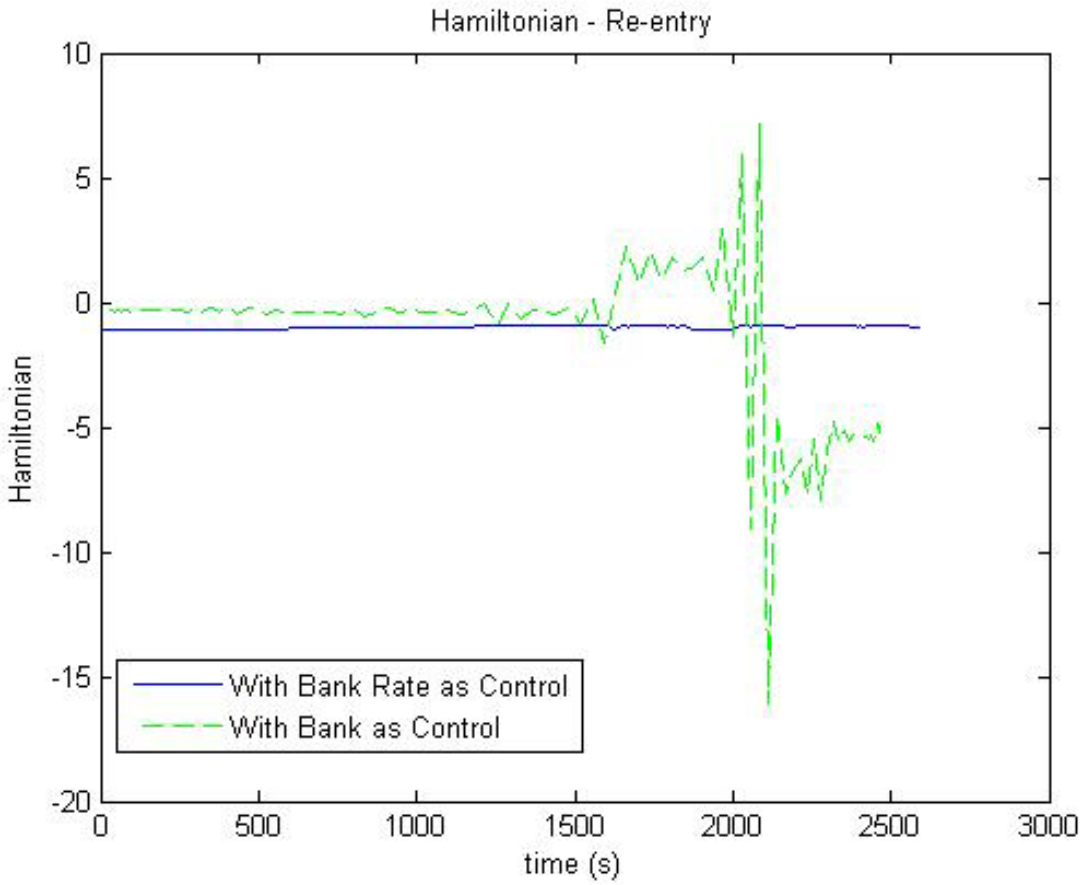


Figure 39: Difference in Hamiltonian

Figure 39 shows a Hamiltonian that is a lot closer to -1 for the bank rate control than the bank angle control. A close look at the Hamiltonian for the bank angle rate is shown in Figure 40.

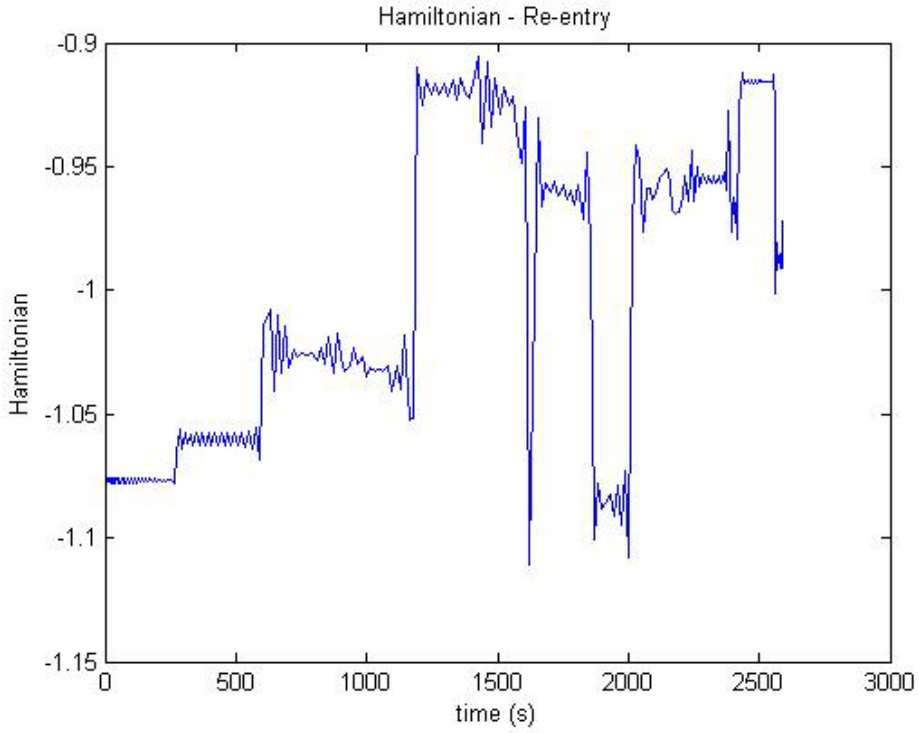


Figure 40: Hamiltonian for Bank Rate

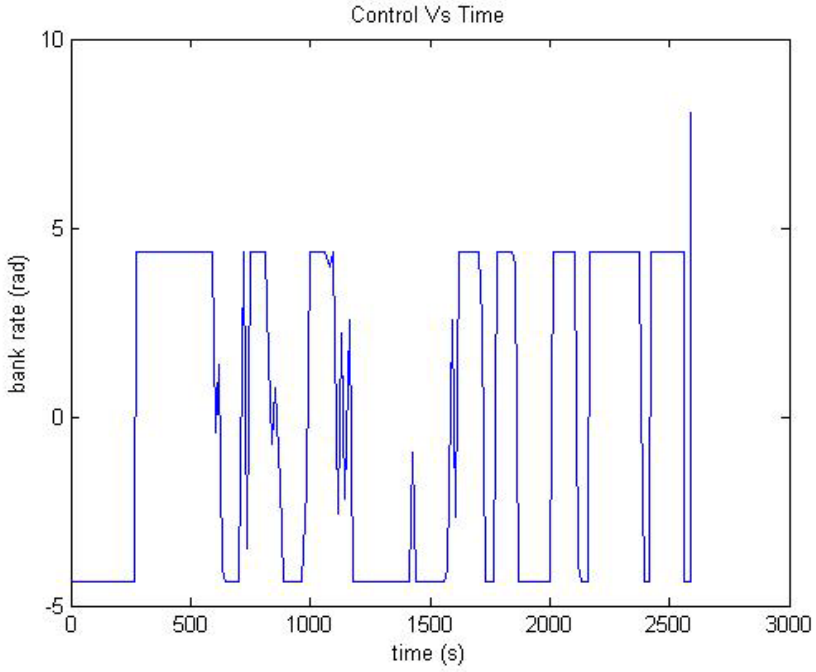


Figure 41: Bank Rate vs. Time

The data compared in the comparison graphs for the bank rate control verses bank angle control are for 245 nodes. The number of nodes a problem uses drastically impacts the solution. A converged solution is identified by the same solution being presented for a various number of nodes. This leads me to believe that the Hamiltonian for the bank rate control could reach a more stable value as its convergence to -1 can be shown in Figure 46. Both Figure 42 and Figure 43 show small difference in the different nodes the same is also true for Figure 44 the position. The bank angle has slightly greater differences because it is directly tied to the control bank rate. These results show a converging solution so one could say the re-entry phase is feasible.

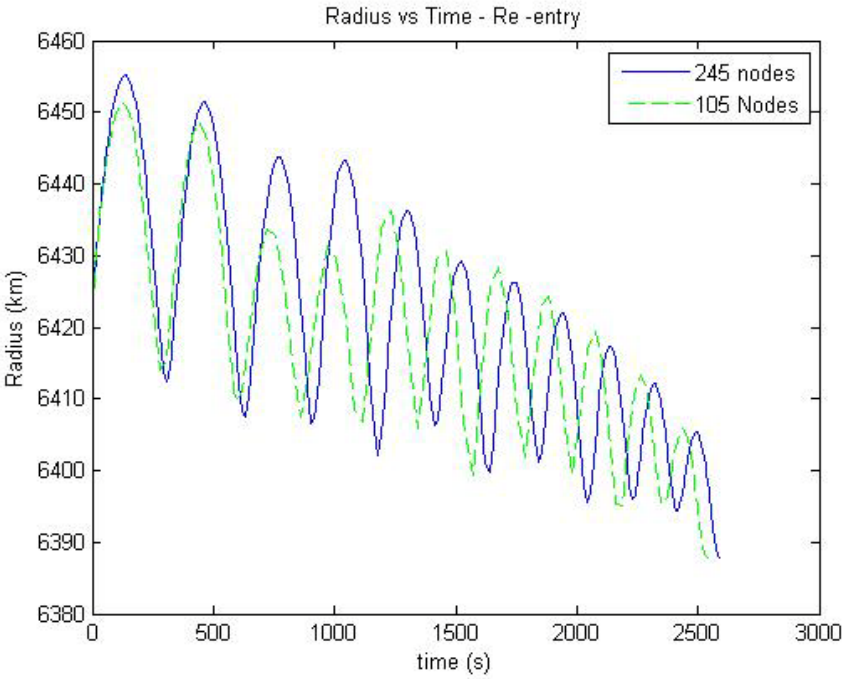


Figure 42: Radius for Different Nodes

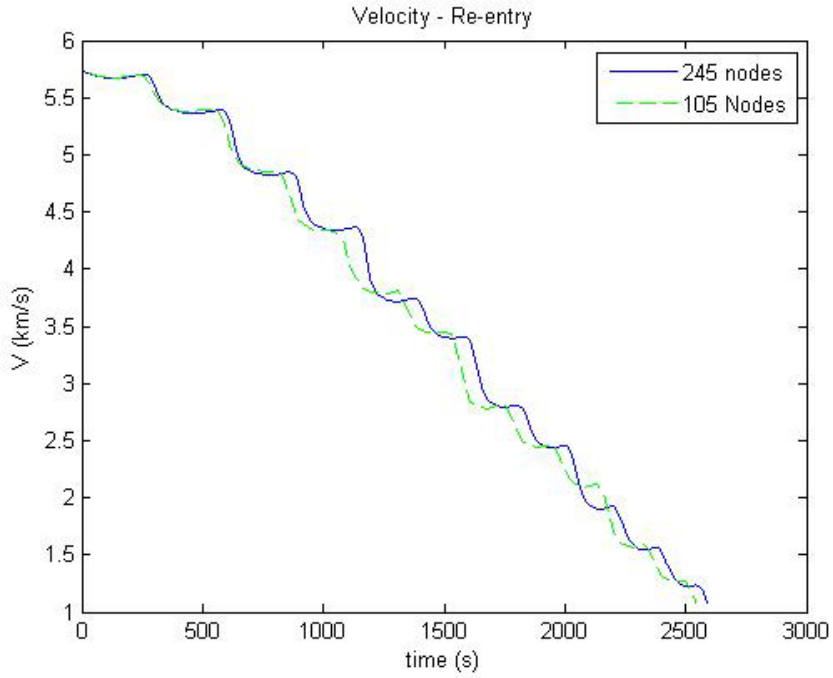


Figure 43: Velocity for Different Nodes

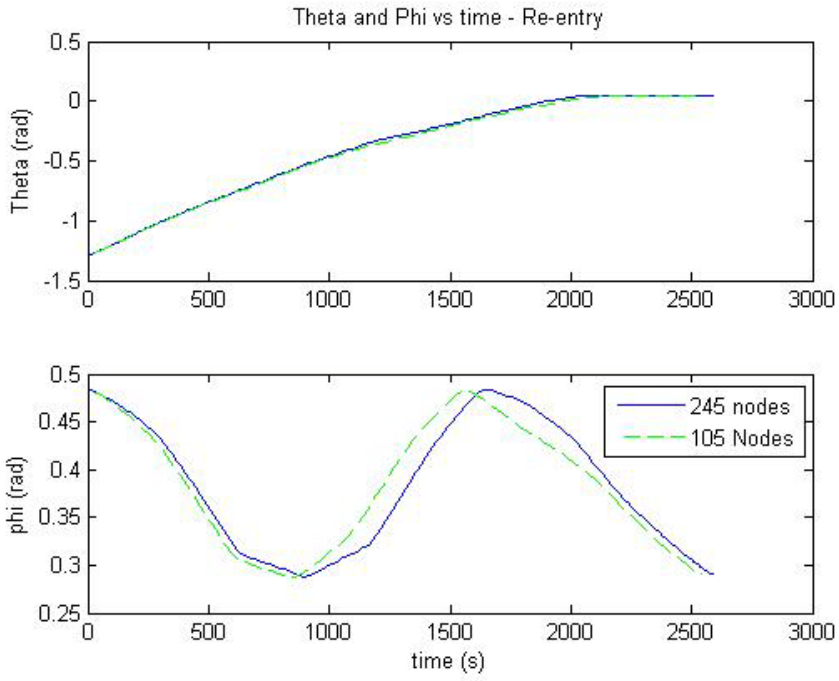


Figure 44: Position for Different Nodes

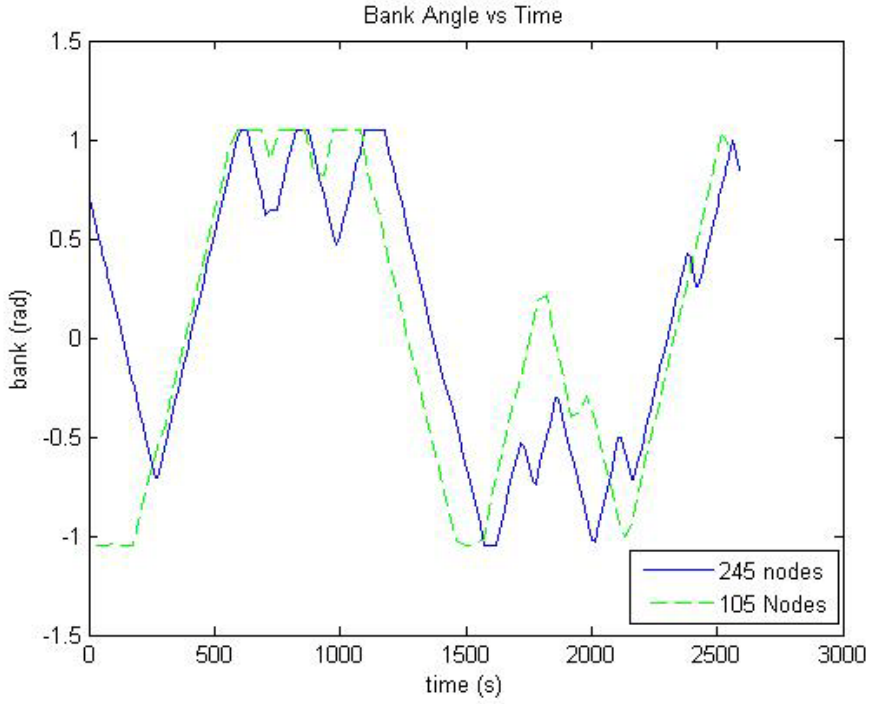


Figure 45: Bank Angle for Different Nodes

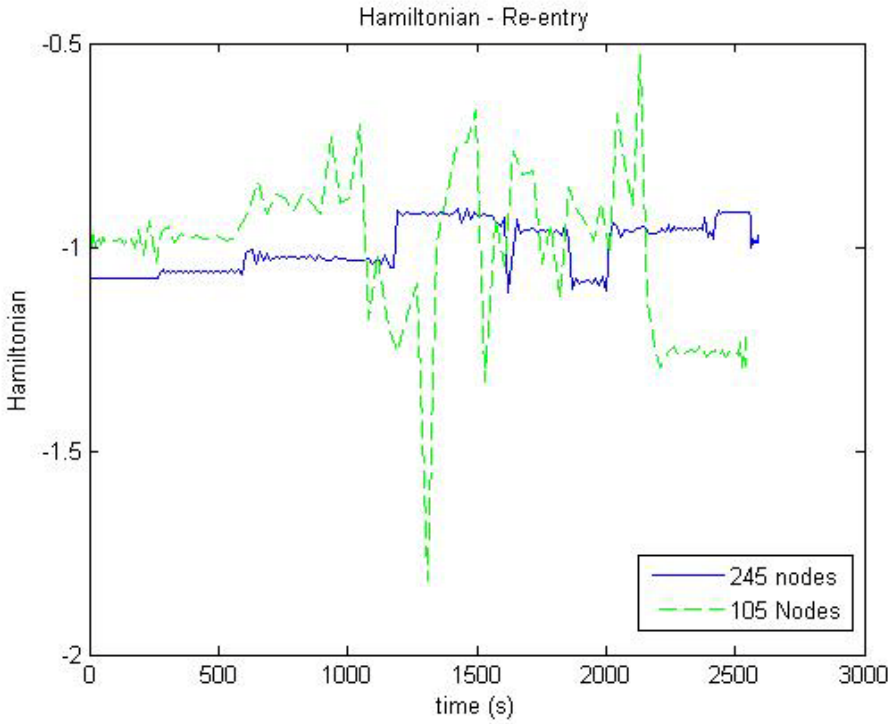


Figure 46: Hamiltonian for Different Nodes

4.2.4 Terminal Results

The terminal results are purely created from a forward propagation of the initial conditions. Figure 47 shows a 3D representation of this section of the trajectory.

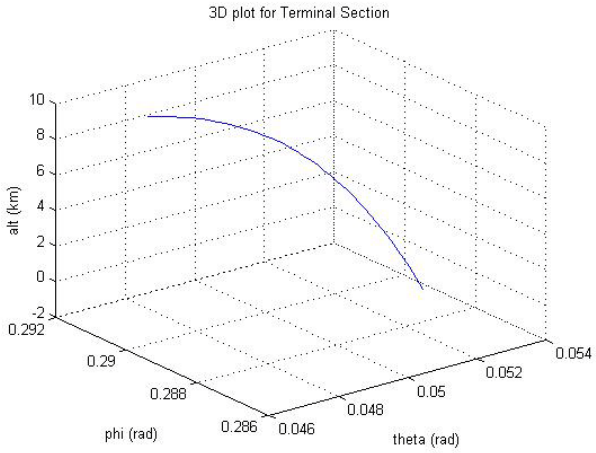


Figure 47: 3D position plot for Terminal Section

The states are shown in Figure 48 and Figure 50 with Figure 49 showing the magnitude of the velocity.

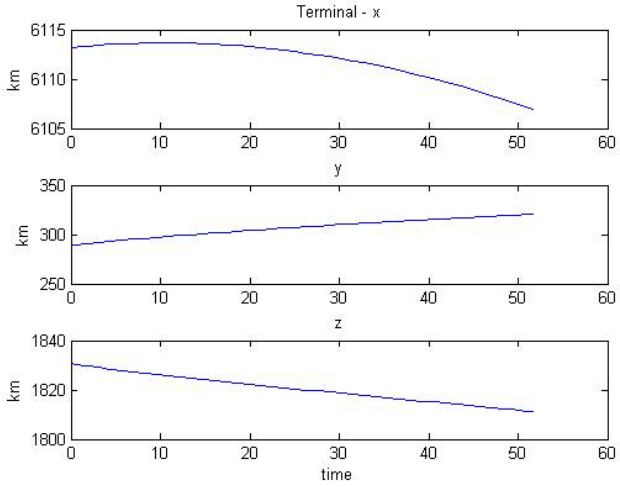


Figure 48: Position for Terminal Section

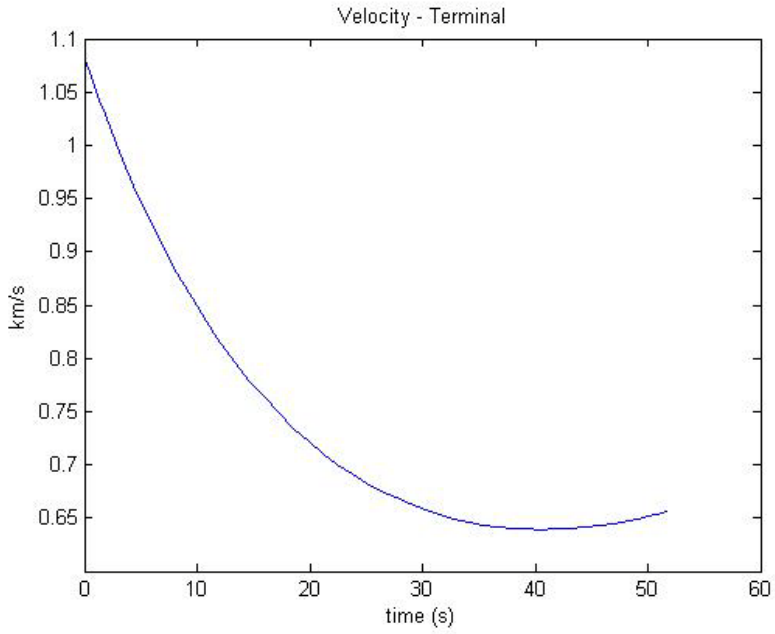


Figure 49: Velocity Profile for Terminal Section

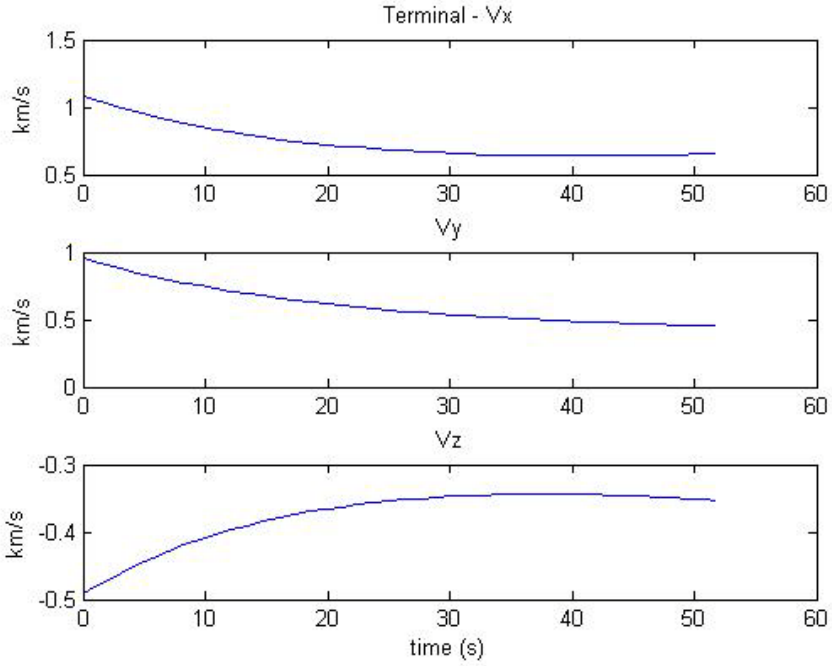


Figure 50: Velocity Breakdown for Terminal Section

4.2.5 Total Trajectory Results

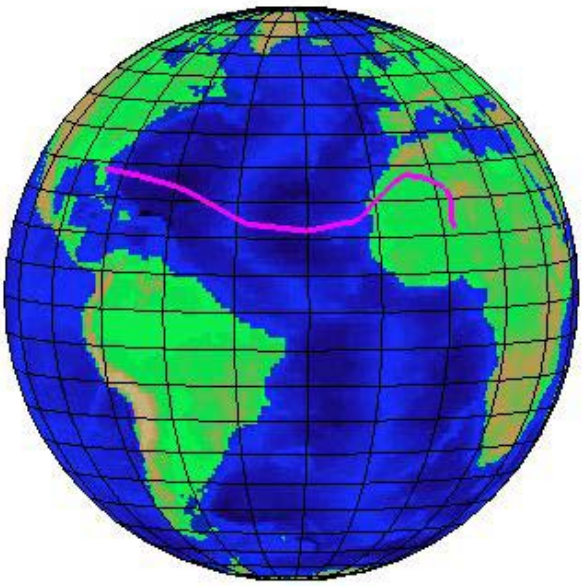


Figure 51: Globe Complete Trajectory

Figure 51 shows the entire trajectory on earth from KSC to Timbuktu. Figure 52 and Figure 53 show the radius and velocity for the total trajectory showing smooth connection from one section to the next. There are no discontinuities in the plots. These plots use the bank rate control thus having a feasible solution. Yet it can be seen by the amount of maneuvering at the end of the trajectory that this solution is not optimal.

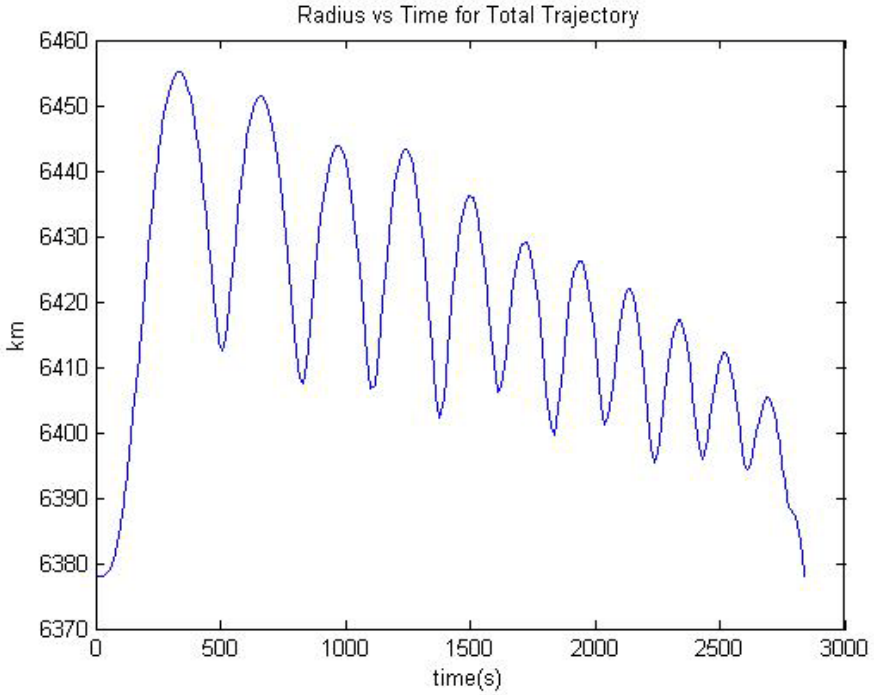


Figure 52: Radius for Complete Trajectory

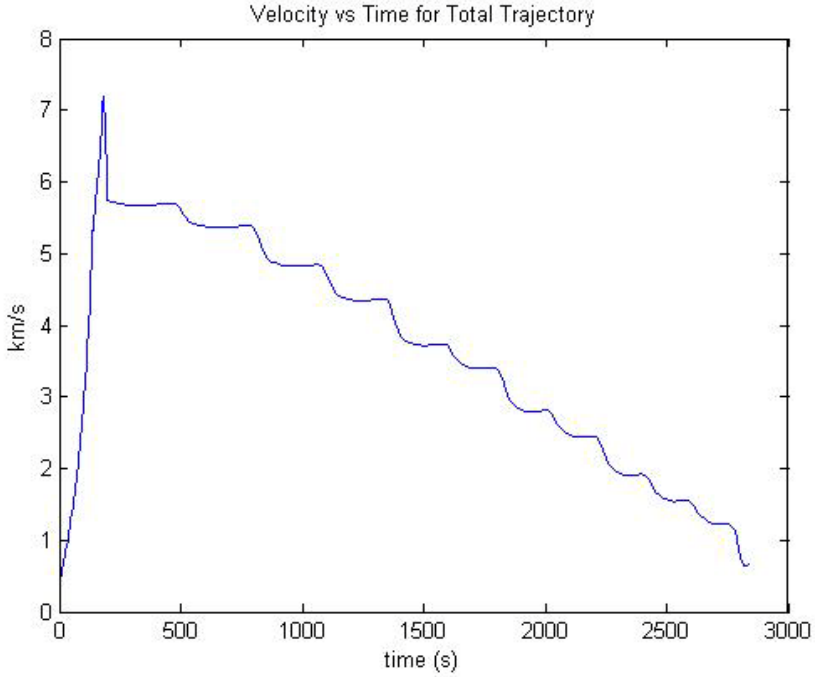


Figure 53: Velocity for Complete Trajectory

4.2.6 Connected Results

Due to the lack of optimality in the 3 separate solution method it seemed crucial to return to a connected approach. Instead of attacking the entire problem again, Only the re-entry phase and the terminal phase are connected. An initial converged solution took about 20hrs to achieve. The results create a clearly more optimal trajectory as can be seen in the figures to follow. Figure 54 shows the connected radius and velocity for the re-entry and terminal phases. The radius and velocity both connect smoothly between phases and still trend towards similar patterns, but the total time is now around 1600 seconds for re-entry and terminal phase as opposed to over 2500 seconds for just the re-entry.

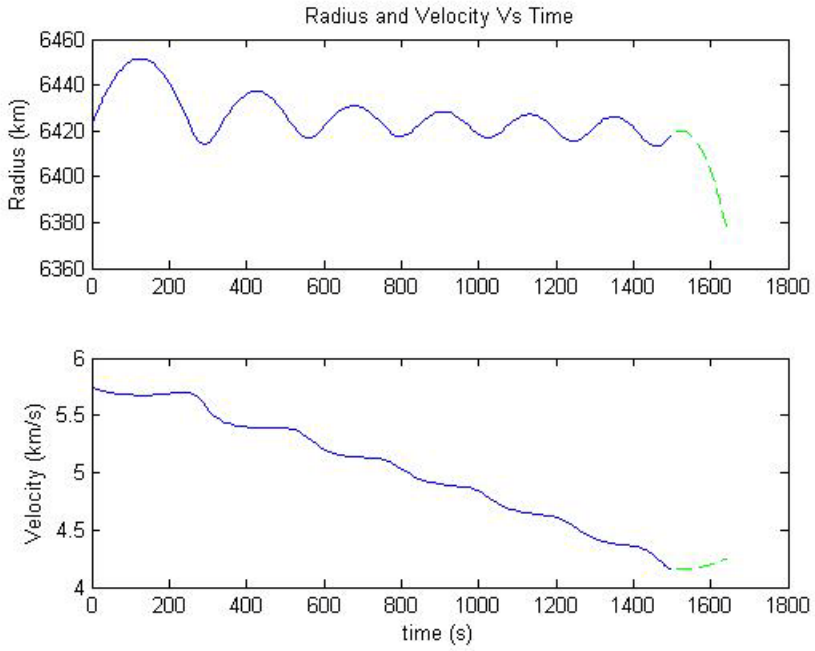


Figure 54: Radius and Velocity for Connect Solution

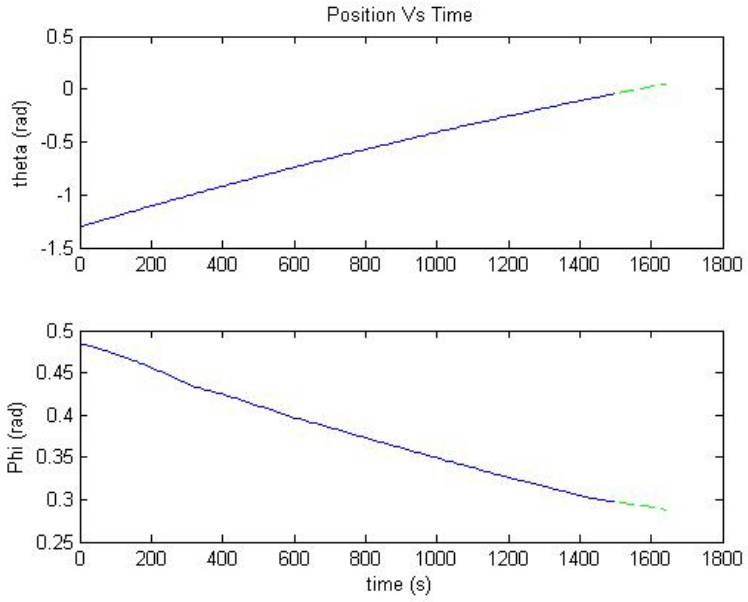


Figure 55: Longitude and Latitude for Connected Solution

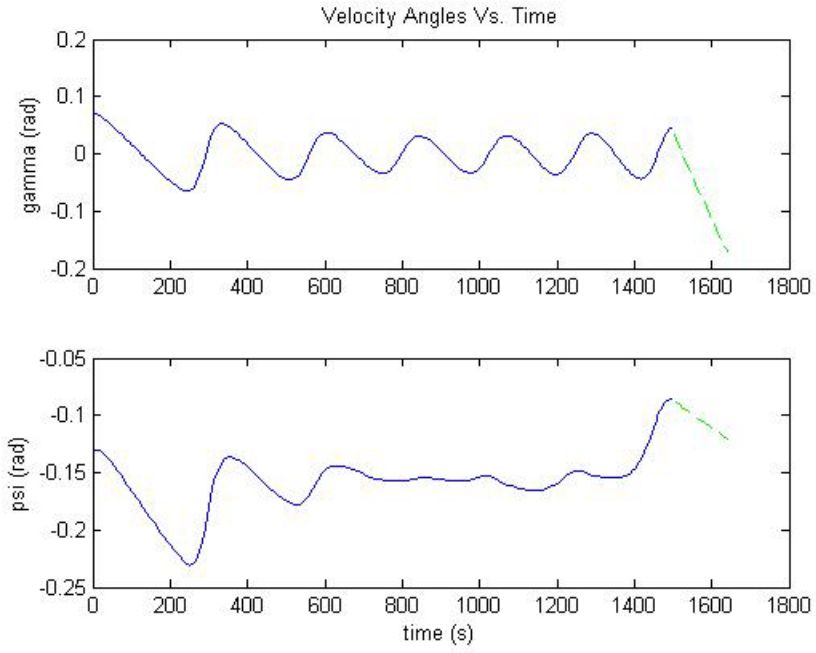


Figure 56: Gamma and Psi for Connected Solution

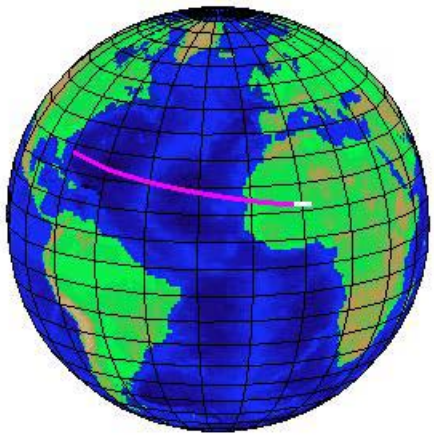


Figure 57: Globe for Connected Solution

Figure 55 shows a direct path in both longitude and latitude to the target site. This seems appropriate for a minimum time problem between two points to create a straight line; this can also be seen in the globe trajectory plot, Figure 57. Figure 56 shows a small variance in the velocity angles.

Figure 58 shows the bank angle and bank rate for the re-entry portion of the connected solution. A saw tooth wave is the bank angle solution with a square wave as the control or bank angle. The other solutions in the re-entry phase appeared to be trending towards this solution but connecting the phases allowed for the solution to completely converge. Figure 59 shows the Hamiltonian is within .01 on either side of -2 as opposed to the pervious Hamiltonians which varied at least 10 times that.

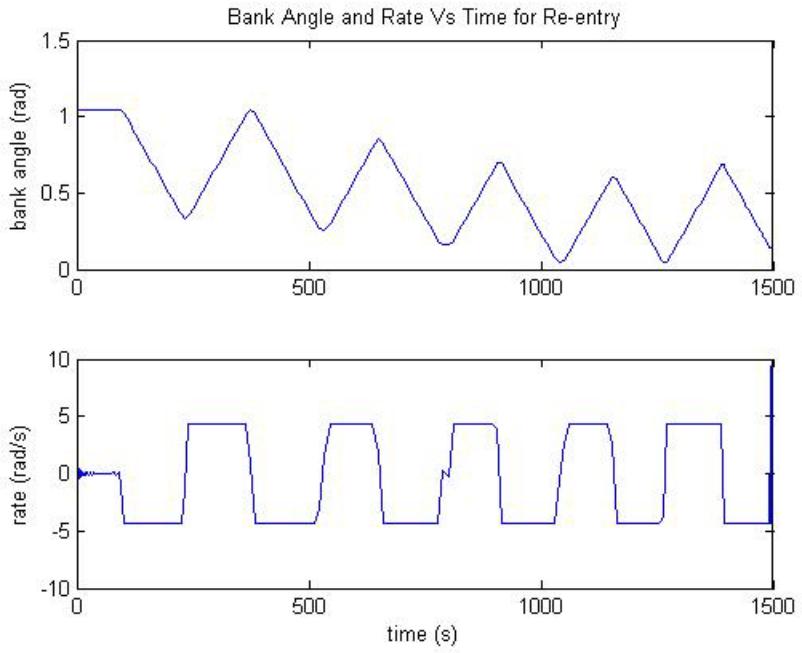


Figure 58: Bank Angle and Rate for Re-entry Connected Solution

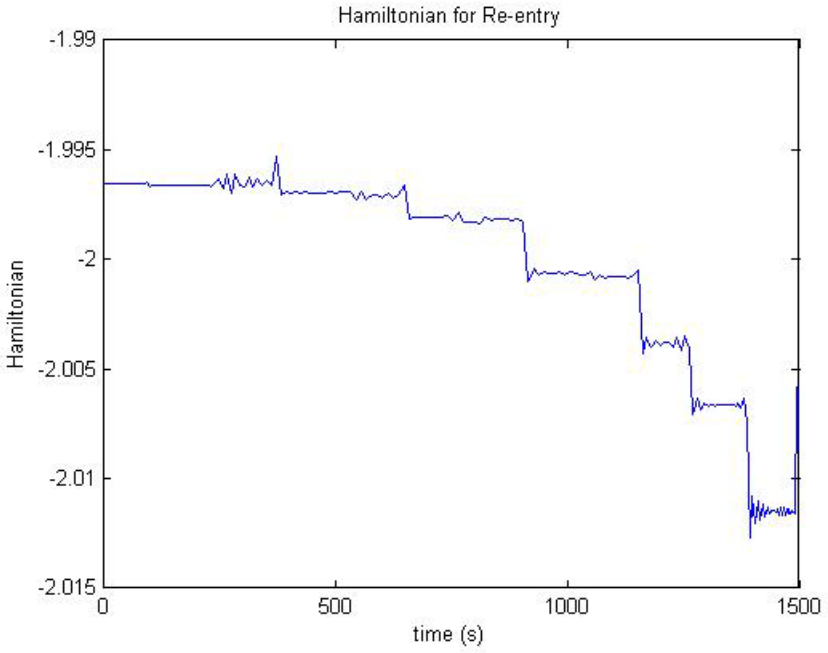


Figure 59: Hamiltonian for Connected Solution

4.3 Investigative Questions Answered

Though the methodology did change in the process of this experiment, the original goals were still achieved. Below are a recap of the original goals and a brief description of how they were achieved.

4.3.1 Break the problem into phases

The problem was broken up into 3 main sections and then additional phases are added for stages in launch or waypoints in the re-entry section. Thus 5 total phases are assigned, for an efficient transition between different dynamics and states in the problem.

4.3.2 Define the reference frames and derive the equations of motion for each phase

The reference frames for each section are clearly defined. The process for converting from one reference frame to another is in place for any future cases. Each of the 3 sections has their own equation of motion defined for the reference frame the state will be in.

4.3.3 Connect the phases

The code to switch between reference frames has successfully been tested outside of the optimal control solver, showing that the method behind the connection equations is valid. The derivation for the connection equations can be found in Appendices A and B.

4.3.4 Set up constraints for the problem

All the constraints or formats for potential additional constraints have been set up for this problem. Constraints for the states and controls have been put into place by establishing the minimum and maximum values which are written in the Main script. The path constraints which would include acceleration and heating have not yet been introduced but the structure is there once minimums and maximums for the case are determined.

4.3.5 Develop specific case to test and support the above goals

Section 4.2 provided a description of a trajectory for a specific case. All of the results are based upon this case.

4.4 Summary

Due to complication in the implementation of a single solution method, a sectional method was employed breaking the problem into 3 smaller problems: launch, re-entry, and termination. The solutions for the section method are shown in Section 4.2. Due to infeasibility in the original setup, an additional degree of control was added to the problem, bank rate. These results then allowed for the re-entry and terminal phases to be connected creating now only 2 separate parts of the problem. Though not fully meeting the overall objective of designing and implementing a single solution method, all of the goals for setting up a mission planning tool were met, shown in Section 4.3.

V. Conclusions and Recommendations

5.1 Conclusions of Research

The results obtained demonstrate that to get a complete optimal trajectory for the enter mission as opposed to a piece-wise optimal solution for each section, the 3 separate trajectory solutions need to be tied together into a single cost and then solved by the GPOPS code. The current sing section trajectory solution can be used as a good guess to place into a complete GPOPS code at a later stage. The research show that for the re-entry section an addition state for a bank angle was needed for the control variable to keep a realistic bank angle. Overall, it was shown that GPOPS is capable of generating piece-wise missile trajectories and has potential for use as a mission planning tool.

5.2 Significance of Research

There is a substantial requirement for a new trajectory optimization tool in the Air Force. Computational abilities have drastically improved since 1988 when OTIS was contracted. The idea of rapid trajectory generation has changed over the years from days to minutes to the potential of seconds to create a trajectory. GPOPS has the potential to solve the complicated equations of motion faster than its predecessors thus making it ideal for the mission planning tool. This research is a great first step to a user friendly mission planning tool. It shows the mathematical capability of optimizing the stages of the problem in a rapid solution and setting up for future models to be tested.

5.3 Recommendations for Action

The research provided in this paper leads to instant actions, or actions that can take place without much further research. The three main recommendations for action are: Integrate new GPOPS version 3.0; Compare case with POST or OTIS data; and run other cases of interest. GPOPS version 3.0 includes a new mesh grid algorithm which should no longer require the user to figure out the correct number of node points needed to solve the trajectory. This should allow for a more automatic process in solving trajectories.

Since POST and OTIS are both currently being used today an immediate action to be performed would be to run similar scenarios on both GPOPS and POST or OTIS and compare the results and run time. GPOPS should be able to process an answer quicker but the differences between the solutions would need to be further investigated.

This research was sponsored by AFRL with the specific case being a research case they have ran with their systems before, but simplified. Increasing the model to match AFRL's cases and getting specific targets of significant to place in the model would make the trajectories easier to compare to historical data. In addition to actions that can currently be completed there are many additions to this research that would be worthy of future investigation. The next section covers those research endeavors.

5.4 Recommendations for Future Research

The main motivation behind this research is to create a mission planning tool so future research should be in support of that goal. There are many steps that are needed in order to get a user friendly mission planning tool that will rapidly generate trajectories. The future areas of

research for a mission planning tool can be broken up into five categories: Vehicle Specifications; Model; Path; Code and Far Future additions.

Vehicle Specifications

In regards to the aerodynamics information there is a lot of future work that can be done in this area. First, implementing an angle of attack dependency for the lift and drag coefficients. Also, not assuming that the mass is a point mass and thus has a changing center of mass and pressure with appropriated lift and drag and moment calculations. Since a lot of aerodynamic data is experimental and thus tabular, having a table lookup option for aerodynamics would be a nice addition to the vehicle models.

In the propulsion section of the vehicle model, having an option for varying types of propulsion systems would be a needed expansion. Allowing for throttled thrust, variable mass flow rates or a specific thrust profile to be input would create more realistic and versatile models.

The controls in the original set up are fairly simplistic having only angles or angular rates as the control. Creating options for more complicated controls and feedback loops into the dynamics that allow for reactions to onboard sensors or GPS data would create a more realistic solution. Also having a library of vehicles and data available for the tool would be very appealing to users to be able to select a vehicle as opposed to finding all the data for any new case one wanted to run.

Model

In addition to the atmospheric and gravity models, having wind models would increase the accuracy of the final trajectory. Though a lot of tools neglect wind to simplify their models, there are multiple standard wind models that could be apply to this tool to be selected by the

user. Also the number of models for both the atmosphere and gravity could be increase to offer the user of the tool more possibilities for research. One example would be to add a nonstandard atmospheric model.

An interesting study that could possibly help make the overall tool run faster would be to analyze the speed of convergence for each additional level of fidelity in the model. Future research could determine whether or not running a case with simplifier models first to determine a guess solution would be faster than trying to get the complicated models to converge from a basic guess.

Path

Studying No-fly zones and Waypoints have been the primary research emphasis for many of my predecessors in this field. Applying what they have done for a mission planning tool and expanding on it would create highly developed path constraints. An ideal user function for the mission planning tool would be to have the user capable of selecting a treaty they would like to follow which would include a set of no-fly zones.

Another path changing constraint that should be added to this tool is angle requirements for both launch and detonation. A target may need to be attacked from a specific direction in order to minimize civilian causalities, for example to avoid a hospital the missile made need to attack from a direction other than the optimized one. As could be seen with the launch site data, most launch sites have angle restrictions on the azimuth angle. Future research could be to determine the impact changing these angles has on the entire trajectory.

Code

The areas of coding that would benefit from future research would be to unify the sections created into a single case like originally planned. Preliminary steps would need to be taken to get a single code to work with different states. There may be scaling issues with the original set up of this problem. Future investigations may need to figure out the best method to create a single code with a unified cost function.

A guess generator is also a great need for the mission planning tool. In order for the code to run the fastest, which speed is one of GPOPS greatest selling points for this tool, a better guess is needed than initial and final conditions. This guess generator could use an ordinary differential equation solver to gain a rough trajectory to be input into the guess for GPOPS.

The current phase connections equations have to call multiple functions and sometimes calculate values that aren't needed as a check for validity. A more efficient phase connection may also help to connect the entire code. Also increasing the control in the re-entry section to bank acceleration should provide smoother bank angle plots and may make the solution converge at a faster rate.

Far Future

Along with finding the optimal trajectory, it is important to have information on how variations in the parameters will cause the mission to fail. The capability to do failure analysis would be a great future addition to the mission planning tool. Keeping the final conditions to within a set error tolerance minimum and maximum variations at waypoints or at launch could be calculated. Thus the tool would be creating an optimal trajectory with a success envelope around it.

This tool should be versatile enough to be adapted for many missions in the future and not just a missile trajectory. Orbital missions should not be too complicated to add on the mission. In addition to the transformation between reference frames that is already in place, orbital parameters would need to be added to the mix.

Appendix A: Rpg2Xyz

The derivation below shows how position is converted into an earth fixed relative frame from a vehicle fixed frame.

$$[\hat{e}_2] = R_{y_2}(-\phi)R_{z_1}(\theta)[\hat{e}_1] \tag{A1}$$

$$\hat{e}_2 = \begin{bmatrix} \cos(-\phi) & 0 & -\sin(-\phi) \\ 0 & 1 & 0 \\ \sin(-\phi) & 0 & \cos(-\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{e}_1$$

$$\hat{e}_2 = \begin{bmatrix} \cos(\phi) & 0 & \sin(-\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{e}_1$$

$$\begin{bmatrix} \cos(\phi) & 0 & \sin(-\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}^{-1} \hat{e}_2 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{e}_1$$

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos(\phi) & 0 & \sin(-\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}^{-1} \hat{e}_2 = \hat{e}_1 \tag{A2}$$

$$R_{z_1}(-\theta)R_{y_2}(\phi)[\hat{e}_2] = [\hat{e}_1] \tag{A3}$$

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \hat{e}_2 = \hat{e}_1$$

$$\begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \hat{e}_2 = \hat{e}_1 \tag{A4}$$

$$\begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{A5}$$

$$x = \cos(\theta)\cos(\phi)\, r \tag{A6}$$

$$y = \sin(\theta)\cos(\phi)\, r \tag{A7}$$

$$z = \sin(\phi)\, r \tag{A8}$$

Appendix B: Vgp2Vxyz

The derivation below shows how velocity is converted into an earth fixed relative frame from the velocity frame.

$$[\hat{e}_2] = R_{y_2}(-\phi)R_{z_1}(\theta)[\hat{e}_1] \tag{B1}$$

$$\hat{e}_2 = \begin{bmatrix} \cos(-\phi) & 0 & -\sin(-\phi) \\ 0 & 1 & 0 \\ \sin(-\phi) & 0 & \cos(-\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{e}_1 \tag{B2}$$

$$\hat{e}_2 = \begin{bmatrix} \cos(\phi) & 0 & \sin(-\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{e}_1$$

$$\begin{bmatrix} \cos(\phi) & 0 & \sin(-\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}^{-1} \hat{e}_2 = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{e}_1$$

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \cos(\phi) & 0 & \sin(-\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}^{-1} \hat{e}_2 = \hat{e}_1$$

$$R_{z_1}(-\theta)R_{y_2}(\phi)[\hat{e}_2] = [\hat{e}_1] \tag{B3}$$

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \hat{e}_2 = \hat{e}_1$$

$$\begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \hat{e}_2 = \hat{e}_1 \tag{B4}$$

$$[\hat{e}''] = R_{z'}(-\gamma)R_{x_2}(\psi)[\hat{e}_2] \tag{B5}$$

$$[\hat{e}_2] = R_{x_2}(-\psi)R_{z'}(\gamma)[\hat{e}''] \tag{B6}$$

$$[\hat{e}_2] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} [\hat{e}'']$$

$$[\hat{e}_2] = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\cos(\psi)\sin(\gamma) & \cos(\psi)\cos(\gamma) & -\sin(\psi) \\ -\sin(\psi)\sin(\gamma) & \sin(\varphi)\cos(\gamma) & \cos(\psi) \end{bmatrix} [\hat{e}''] \quad \text{(B7)}$$

$$\begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\cos(\psi)\sin(\gamma) & \cos(\psi)\cos(\gamma) & -\sin(\psi) \\ -\sin(\psi)\sin(\gamma) & \sin(\varphi)\cos(\gamma) & \cos(\psi) \end{bmatrix} [\hat{e}''] = \hat{e}_1 \quad \text{(B8)}$$

For velocity $[\hat{e}''] = \begin{bmatrix} 0 \\ V \\ 0 \end{bmatrix}$ e1 $= \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\cos(\psi)\sin(\gamma) & \cos(\psi)\cos(\gamma) & -\sin(\psi) \\ -\sin(\psi)\sin(\gamma) & \sin(\varphi)\cos(\gamma) & \cos(\psi) \end{bmatrix} \begin{bmatrix} 0 \\ V \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\phi) & -\sin(\theta) & -cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) & \cos(\theta) & -\sin(\theta)\sin(\phi) \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} V\sin(\gamma) \\ V\cos(\psi)\cos(\gamma) \\ V\sin(\psi)\cos(\gamma) \end{bmatrix}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} V\sin(\gamma)\cos(\theta)\cos(\phi) - V\cos(\psi)\cos(\gamma)\sin(\theta) - V\sin(\psi)\cos(\gamma)cos(\theta)\sin(\phi) \\ V\sin(\gamma)\sin(\theta)\cos(\phi) + V\cos(\psi)\cos(\gamma)\cos(\theta) - V\sin(\psi)\cos(\gamma)\sin(\theta)\sin(\phi) \\ V\sin(\gamma)\sin(\phi) + V\sin(\psi)\cos(\gamma)\cos(\phi) \end{bmatrix} \quad \text{(B9)}$$

Appendix C: 1976 Standard Atmosphere Table Example [33]

Table II

Geometric Altitude, Metric Units

Altitude		Accel. due to gravity	Pressure scale height	Number density	Particle speed	Collision frequency	Mean free path	Molecular weight
Z (m)	H (m)	g (m/s^2)	H_p (m)	n (m^{-3})	V (m/s)	ν (s^{-1})	L (m)	M (kg/kmol)
27000	26886	9.7239	6598.9	6.8916 +23	484.23	1.4575 + 8	2.7734 - 6	28.964
27100	26985	9.7236	6602.0	5.9973	484.32	1.4353	2.8170	28.964
27200	27084	9.7233	6605.1	5.9046	484.41	1.4134	2.8613	28.964
27300	27183	9.7230	6608.3	5.8133	484.50	1.3918	2.9062	28.964
27400	27282	9.7227	6611.4	5.7235	484.58	1.3706	2.9518	28.964
27500	27382	9.7223	6614.5	5.6351	484.67	1.3498	2.9981	28.964
27600	27481	9.7220	6617.7	5.5481	484.76	1.3292	3.0451	28.964
27700	27580	9.7217	6620.8	5.4625	484.85	1.3090	3.0928	28.964
27800	27679	9.7214	6624.0	5.3783	484.94	1.2891	3.1413	28.964
27900	27778	9.7211	6627.1	5.2954	485.03	1.2695	3.1905	28.964
28000	27877	9.7208	6630.2	5.2138 +23	485.12	1.2502 + 8	3.2404 - 6	28.964
28100	27976	9.7205	6633.4	5.1335	485.21	1.2312	3.2911	28.964
28200	28075	9.7202	6636.5	5.0545	485.30	1.2126	3.3425	28.964
28300	28175	9.7199	6639.6	4.9767	485.39	1.1942	3.3948	28.964
28400	28274	9.7196	6642.8	4.9002	485.48	1.1761	3.4478	28.964
28500	28373	9.7193	6645.9	4.8248	485.57	1.1582	3.5016	28.964
28600	28472	9.7190	6649.0	4.7507	485.66	1.1407	3.5562	28.964
28700	28571	9.7187	6652.2	4.6777	485.75	1.1234	3.6117	28.964
28800	28670	9.7184	6655.3	4.6059	485.84	1.1064	3.6680	28.964
28900	28769	9.7181	6658.4	4.5353	485.93	1.0897	3.7252	28.964
29000	28868	9.7178	6661.6	4.4657 +23	486.01	1.0732 + 8	3.7832 - 6	28.964
29100	28967	9.7175	6664.7	4.3973	486.10	1.0570	3.8421	28.964
29200	29066	9.7172	6667.8	4.3299	486.19	1.0410	3.9019	28.964
29300	29166	9.7169	6671.0	4.2636	486.28	1.0253	3.9625	28.964
29400	29265	9.7166	6674.1	4.1983	486.37	1.0098	4.0241	28.964
29500	29364	9.7163	6677.3	4.1341	486.46	9.9468 + 7	4.0867	28.964
29600	29463	9.7160	6680.4	4.0709	486.55	9.7960	4.1501	28.964
29700	29562	9.7157	6683.5	4.0088	486.64	9.6484	4.2146	28.964
29800	29661	9.7153	6686.7	3.9474	486.73	9.5031	4.2800	28.964
29900	29760	9.7150	6689.8	3.8871	486.82	9.3608	4.3463	28.964
30000	29859	9.7147	6692.9	3.8278 +23	486.91	9.2192 + 7	4.4137 - 6	28.964
30100	29958	9.7144	6696.1	3.7694	486.99	9.0805	4.4821	28.964
30200	30057	9.7141	6699.2	3.7119	487.08	8.9448	4.5515	28.964
30300	30156	9.7138	6702.3	3.6553	487.17	8.8095	4.6219	28.964
30400	30255	9.7135	6705.5	3.5996	487.26	8.6772	4.6935	28.964
30500	30354	9.7132	6708.6	3.5448	487.35	8.5469	4.7660	28.964
30600	30453	9.7129	6711.8	3.4908	487.44	8.4187	4.8397	28.964
30700	30552	9.7126	6714.9	3.4377	487.53	8.2924	4.9145	28.964
30800	30651	9.7123	6718.0	3.3854	487.62	8.1681	4.9904	28.964
30900	30751	9.7120	6721.2	3.3340	487.71	8.0457	5.0674	28.964
31000	30850	9.7117	6724.3	3.2833 +23	487.79	7.9251 + 7	5.1456 - 6	28.964
31100	30949	9.7114	6727.4	3.2335	487.88	7.8065	5.2249	28.964
31200	31048	9.7111	6730.6	3.1844	487.97	7.6897	5.3055	28.964
31300	31147	9.7108	6733.7	3.1361	488.06	7.5746	5.3872	28.964
31400	31246	9.7105	6736.9	3.0885	488.15	7.4614	5.4702	28.964
31500	31345	9.7102	6740.0	3.0417	488.24	7.3499	5.5544	28.964
31600	31444	9.7099	6743.1	2.9956	488.33	7.2401	5.6398	28.964
31700	31543	9.7096	6746.3	2.9502	488.42	7.1320	5.7265	28.964
31800	31642	9.7093	6749.4	2.9056	488.50	7.0255	5.8146	28.964
31900	31741	9.7090	6752.6	2.8616	488.59	6.9207	5.9039	28.964
32000	31840	9.7087	6755.7	2.8183 +23	488.68	6.8175 + 7	5.9945 - 6	28.964
32100	32038	9.7080	6764.0	2.7330	488.92	6.6149	6.1818	28.964
32400	32236	9.7074	6780.8	2.6478	489.41	6.4147	6.3825	28.964
32600	32434	9.7068	6797.6	2.5640	489.91	6.2209	6.5892	28.964
32800	32632	9.7062	6814.4	2.4838	490.40	6.0335	6.8020	28.964
33000	32838	9.7054	6831.2	2.4062	490.90	5.8522	7.0212	28.964
33200	33027	9.7050	6848.1	2.3313	491.39	5.6767	7.2469	28.964
33400	33225	9.7044	6864.9	2.2589	491.88	5.5070	7.4793	28.964
33600	33423	9.7038	6881.7	2.1889	492.37	5.3426	7.7185	28.964
33800	33621	9.7032	6898.5	2.1212	492.86	5.1836	7.9647	28.964
34000	33819	9.7026	6915.4	2.0558 +23	493.35	5.0297 + 7	8.2182 - 6	28.964
34200	34017	9.7020	6932.2	1.9925	493.84	4.8807	8.4791	28.964
34400	34215	9.7014	6949.0	1.9313	494.33	4.7365	8.7477	28.964
34600	34413	9.7008	6965.8	1.8722	494.82	4.5968	9.0240	28.964
34800	34611	9.7002	6982.7	1.8150	495.31	4.4616	9.3084	28.964
35000	34808	9.6995	6999.5	1.7597	495.79	4.3307	9.6010	28.964
35200	35006	9.6989	7016.3	1.7062	496.28	4.2040	9.9021	28.964
35400	35204	9.6983	7033.2	1.6544	496.77	4.0812	1.0212 - 5	28.964
35600	35402	9.6977	7050.0	1.6043	497.25	3.9623	1.0531	28.964
35800	35599	9.6971	7066.8	1.5559	497.74	3.8471	1.0858	28.964
36000	35797	9.6965	7083.7	1.5093 +23	498.22	3.7356 + 7	1.1196 - 5	28.964
36200	35995	9.6959	7100.5	1.4637	498.71	3.6276	1.1542	28.964
36400	36193	9.6953	7117.3	1.4198	499.19	3.5228	1.1899	28.964
36600	36390	9.6947	7134.2	1.3773	499.67	3.4214	1.2266	28.964
36800	36588	9.6941	7151.0	1.3362	500.15	3.3231	1.2643	28.964
37000	36786	9.6935	7167.8	1.2965	500.63	3.2278	1.3031	28.964
37200	36984	9.6929	7184.7	1.2579	501.11	3.1355	1.3430	28.964
37400	37181	9.6923	7201.5	1.2207	501.59	3.0461	1.3841	28.964
37600	37379	9.6917	7218.4	1.1846	502.07	2.9594	1.4262	28.964
37800	37577	9.6911	7235.2	1.1496	502.55	2.8753	1.4696	28.964

Appendix D: Code

Example of Main Code

```
% ---------------------------------------
% Main File
% Re-entry Section
% ---------------------------------------
% ---------------------------------------------
%
% Danielle Yaple
% 28 December 2009
% ---------------------------------------------

function [Re_entry] =Reentry_Main()
global CONSTANTS
a = 100;
nodenum = a;
run MOD_basic
run VEH_minotaur
run TAR_timbuktu
run LAU_KSC

CONSTANTS.m = 1194.8; %kg
% r0 =  6.5000e+003;
% V0max = 5.5;
% theta0 =   -1.3685;
% phi0 =   0.4950;
% gamma0 =    -.6632;
% psi0 =     -0.0279;

r0 =  6.500e+003;
V0 = 5.5;
theta0 =  -1.306357339705933;
phi0 =      0.485335659325143;
gamma0 =   0.328910761317162;
psi0 =      -0.130375293982734;
sigma_guess = 0;
gs = 9.81/1000; %km/s^2
rs = CONSTANTS.rs;
CONSTANTS.wp_earth = 7.2722E-5*sqrt(rs/gs); % rad/s
rp0 = r0/rs;
Vp0 = V0/sqrt(rs*gs);
thetaf = CONSTANTS.TARGET.theta; %rad
phif = CONSTANTS.TARGET.phi; %rad

%extremes
tp_min = 500/sqrt(CONSTANTS.rs/gs);
tp_max = 15000/sqrt(CONSTANTS.rs/gs);
rpmin = 1;
rpmax = 1.2;
```

```
Vpmin = 0.01;
Vpmax = 1;
thetamin = theta0;
thetamax = thetaf;
phimin = phif;
phimax = phi0;
gammamin = -deg2rad(89);
gammamax = deg2rad(89);
psimin = -2*pi;
psimax = 2*pi;
sigmamin = deg2rad(-60); %rad
sigmamax = deg2rad(60); %rad

rp_guess1 = 1.02;
Vp_guess1 = .9;
gamma_guess1 = 0;
psi_guess1 = deg2rad(0);

%from ode
rf = 6.387892378315309e+003;
rfmax = rf;
rpf = rf/CONSTANTS.rs;
rpfmax = rfmax/CONSTANTS.rs;
phif =  0.290637756866697;
thetaf =  0.047243562179519;
Vf = 1.082251805453242;
Vfmax = Vf +1;
Vpf = Vf/sqrt(CONSTANTS.rs*CONSTANTS.gs);
Vpfmax = Vfmax/sqrt(CONSTANTS.rs*CONSTANTS.gs);
% Vpfmin = Vfmin/sqrt(CONSTANTS.rs*CONSTANTS.gs);
% Vpfmax = Vfmax/sqrt(CONSTANTS.rs*CONSTANTS.gs);
gammaf = 7.051945056714285e-004;
psif =  -0.495156679466100;

iphase = 1;
limits(iphase).nodes = nodenum;
limits(iphase).time.min = [0 tp_min];
limits(iphase).time.max = [0 tp_max];
limits(iphase).state.min(1,:)   = [rp0 rpmin rpf];
limits(iphase).state.max(1,:)   = [rp0 rpmax rpf];
limits(iphase).state.min(2,:)   = [Vp0 Vpmin Vpf];
limits(iphase).state.max(2,:)   = [Vp0 Vpmax Vpf];
limits(iphase).state.min(3,:)   = [theta0 thetamin thetaf];
limits(iphase).state.max(3,:)   = [theta0 thetamax thetaf];
limits(iphase).state.min(4,:)   = [phi0 phimin phif];
limits(iphase).state.max(4,:)   = [phi0 phimax phif];
limits(iphase).state.min(5,:)   = [gammamin gammamin gammaf];
limits(iphase).state.max(5,:)   = [gammamax gammamax gammaf];
limits(iphase).state.min(6,:)   = [psi0 psimin psif];
limits(iphase).state.max(6,:)   = [psi0 psimax psif];
limits(iphase).control.min(1,:) = sigmamin;
limits(iphase).control.max(1,:) = sigmamax;
```

```
limits(iphase).parameter.min    = [];
limits(iphase).parameter.max    = [];
limits(iphase).path.min    = [];
limits(iphase).path.max    = [];
limits(iphase).event.min   = [];
limits(iphase).event.max   = [];

if exist([mfilename,'.mat'],'file')
    load(mfilename);
    guess(iphase).time = output.solution(iphase).time;
    guess(iphase).state = output.solution(iphase).state;
    guess(iphase).control = output.solution(iphase).control;
    guess(iphase).parameter = output.solution(iphase).parameter;
else
    guess(iphase).time =  [0; tp_max];
    guess(iphase).state(:,1) = [rp0;rp_guess1];
    guess(iphase).state(:,2) = [Vp0;Vp_guess1];
    guess(iphase).state(:,3) = [theta0;thetaf];
    guess(iphase).state(:,4) = [phi0; phif];
    guess(iphase).state(:,5) = [gamma0; gamma_guess1];
    guess(iphase).state(:,6) = [psi0; psi_guess1];
    guess(iphase).control(:,1) = [sigma_guess; sigma_guess];
    guess(iphase).parameter = [];
end

setup.name  = 'Re-entry';
setup.funcs.cost = 'Reentry_Cost';
setup.funcs.dae = 'Reentry_Dae';
% setup.funcs.link = 'Reentry_Connect';
setup.limits = limits;
setup.guess = guess;
% setup.linkages = linkages;
setup.derivatives = 'automatic';
setup.direction = 'increasing';
setup.autoscale = 'on';

output = gpops(setup);
solution = output.solution;
status = output.SNOPT_info;
Re_entry.sol = solution;
if status == 1
    save(mfilename,'output');
    r = output.solution.state(:,1)*rs;
    V = output.solution.state(:,2)*sqrt(rs*gs);
    theta = output.solution.state(:,3);
    phi = output.solution.state(:,4);
    gamma = output.solution.state(:,5);
    psi = output.solution.state(:,6);
    time = output.solution.time*sqrt(rs/gs);

    linestyles = ['m-'];
    plot_on_sphere(phi,theta,r,21,linestyles);
```

```
    figure(22);
    subplot(7,1,1); plot(time,r-CONSTANTS.rs,linestyles);
ylabel({'altitude','[km]'}); axis tight; hold on;
    subplot(7,1,2); plot(time,V,linestyles); ylabel({'velocity','[km/s]'});
axis tight; hold on;
    subplot(7,1,3); plot(time,(theta).*180/pi,linestyles);
ylabel({'longitude','[deg]'}); axis tight; hold on;
    subplot(7,1,4); plot(time,(phi).*180/pi,linestyles);
ylabel({'latitude','[deg]'}); axis tight; hold on;
    subplot(7,1,5); plot(time,(gamma).*180/pi,linestyles); ylabel({'flight-
path-angle','[deg]'}); axis tight; hold on;
    subplot(7,1,6); plot(time,(psi).*180/pi,linestyles);
ylabel({'heading','[deg]'}); axis tight; hold on;
    subplot(7,1,7); plot(theta.*180/pi,phi.*180/pi,linestyles);
ylabel('latitude [deg]'); xlabel('longitude [deg]'); axis tight; hold on;

    l = length(time);
    time(l);
    rf = r(l);
    Vf = V(l);
    thetaf = theta(l);
    phif = phi(l);
    gammaf = gamma(l);
    psif = psi(l);
    [x,y,z] = rpt2xyz(rf, phif, thetaf);
    [Vx,Vy,Vz] = Vgp2Vxyz(Vf,gammaf,psif,thetaf,phif);


else
   fprintf('boo\n')

end
```

Example of Connect Code

```
function [connect Dconnect] = Sim1Connect(sol);

global CONSTANTS

xf_left = sol.left.state;
p_left = sol.left.parameter;
left_phase = sol.left.phase;
x0_right = sol.right.state;
p_right = sol.right.parameter;
right_phase = sol.right.phase;

connect = x0_right-xf_left;


% avoid calc of derivs in not necessary
if nargout == 2
    Dconnect = [-eye(length(xf_left)), eye(length(x0_right))];
end
```

Example DAE Code

```
function XDOT = Sim1Dae(sol)
% These are the equations of motion for re-entry based on Hicks
 % t = solode.time;
rp = sol.state(:,1); % altitude ; rp = radius prime (prime to denote
nondimensional)
Vp = sol.state(:,2); % velocity-like ; Vp = velocity prime
%theta = solode.states(:,3); % longitude
phi = sol.state(:,4); % latitude
gamma = sol.state(:,5); % flight-path angle
psi = sol.state(:,6); % heading
% sigma = sol.state(:,7);
% bnkrate = sol.control(:,1); % bank angle
sigma = sol.control(:,1);
u1 = cos(sigma);
u2 = sin(sigma);

% constants ; the calling function should have a global variable named
CONSTANTS with the below variables.
global CONSTANTS
rs = CONSTANTS.rs; % radius at the surface of the Earth [km]
rhos = CONSTANTS.rhos; % density of atmosphere at surface of the Earth
beta = CONSTANTS.beta; % scaling constant for atmosphere
m = CONSTANTS.m;
Cl = CONSTANTS.Cl;
Cd = CONSTANTS.Cd;
S = CONSTANTS.S;
wp_earth = CONSTANTS.wp_earth;

Lp = .5.*rhos.*exp(-beta.*rs.*(rp-1)).*Cl.*S.*Vp.^2.*rs./m;
Dp = .5.*rhos.*exp(-beta.*rs.*(rp-1)).*Cd.*S.*Vp.^2.*rs./m;

drpdot = Vp.*sin(gamma);
dVpdot = -sin(gamma)./rp.^2 +
rp.*wp_earth.^2.*cos(phi).*(cos(phi).*sin(gamma)-
sin(phi).*sin(psi).*cos(gamma)) - Dp;
dthetapdot = Vp./rp.*cos(gamma).*cos(psi)./cos(phi);
dphipdot = Vp./rp.*cos(gamma).*sin(psi);
dgammapdot = .5.*rhos.*exp(-beta.*rs.*(rp-1)).*Cl.*S.*Vp.*rs.*u1./m -
cos(gamma)./(rp.^2.*Vp) + Vp.*cos(gamma)./rp +
2.*wp_earth.*cos(phi).*cos(psi) +
rp./Vp.*wp_earth.^2.*cos(phi).*(cos(phi).*cos(gamma)+sin(phi).*sin(psi).*sin(
gamma));
dpsipdot = Lp.*u2./Vp./cos(gamma) - Vp./rp.*cos(gamma).*cos(psi).*tan(phi) +
2.*wp_earth.*(sin(psi).*cos(phi).*tan(gamma)-sin(phi)) -
rp.*wp_earth.^2./Vp.*sin(phi).*cos(phi).*cos(psi)./cos(gamma);

XDOT = [drpdot dVpdot dthetapdot dphipdot dgammapdot dpsipdot];
end
```

Example of Cost Code

```
function [Mayer,Lagrange]=Reentry_Cost(sol)


t0 = sol.initial.time;
x0 = sol.initial.state;
tf = sol.terminal.time;
xf = sol.terminal.state;
t  = sol.time;
x  = sol.state;
u  = sol.control;

%Method 1 Minimize Time

Mayer = tf;
Lagrange = zeros(size(t));
```

Bibliography

1. Air Force Global Strike Command Home Page. http://www.afgsc.af.mil/. October 2009

2. Axelsson, O., "Global Integration of Differential Equations Through Lobatto Quadrature," BIT, Vol. 4 (1964) pp 69-86

3. Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method, *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1435-1440.

4. Benson, D. A., *A Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, MIT, November 2004.

5. Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method, *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, November-December 2006, pp. 1435-1440.

6. Bertin, John J., Brandt, Stiles, Whitford, *Introduction to Aeronautics: A Design Perspective* (2nd Edition). American Institute of Aeronautics and Astronautics, 2004

7. Biblarz, Oscar , Sutton, *Rocket Propulsion Elements* (7th Edition). John Wiley & Sons, 2000.

8. Bowman, Keith, Dr., "Air Force Research Laboratory 2009 Focused Long Term Challenge 4 – Persistent and Responsive Precision Engagement" Address to Attendees of Air Force Research Laboratory's Focused Long Term Challenge Workshop. Colonial Williamsburg, VA. 5-7 October 2009.

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

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

11. 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.

12. Euler, L., Elementa Calculi variationum, Originally published in Novi Commentarii academiae scientiarum Petropolitanae 10, 1766, pp 51-93.

13. Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "A Unified Framework for the Numerical Solution of Optimal Control Problems Using Pseudospectral Methods," *Automatica*, Provisionally accepted for publication, December 2009.

14. 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.

15. 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.

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

17. Hicks, Kerry D., "Introduction to Astrodynamic Reentry" Summer 2009. Air Force Institute of Technology, Wright-Patterson AFB, OH, Edition 1.3, (unpublished).

18. Huntington, G. T., Benson, D. A., and Rao, A. V., Design of Optimal Tetrahedral Spacecraft Formations, *The Journal of the Astronautical Sciences*, Vol. 55, No. 2, April-June 2007, pp. 141-169.

19. Huntington, G. T., *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*, Ph.D. Thesis, Dept. of Aeronautics and Astronautics, MIT, May 2007.

20. Huntington, G. T. and Rao, A. V., "Optimal Reconfiguration of Spacecraft Formations Using a Gauss Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, May-June 2008, pp. 689-698.

21. Huntington, G. T., Benson, D. A., How, J. P., Kanizay, N., Darby, C. L., and Rao, A. V., "Computation of Boundary Controls Using a Gauss Pseudospectral Method," *2007 Astrodynamics Specialist Conference*, Mackinac Island, Michigan, August 2007.

22. Jorris, Timothy R. Common "Aero Vehicle Autonomous Reentry Trajectory Optimization Satisfying Waypoint and No-Fly Zone Constraints". Air Force Institute of Technology (AU), Wright-Patterson AFB OH.

23. 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.

24. Karasz, William. “Optimal Re-entry Trajectory Terminal State Due to Variations in Waypoint Locations”. Air Force Institute of Technology (AU), Wright-Patterson AFB OH.

25. Kirk, Donald E., *Optimal Control Theory: An Introduction*, Dover Publications, 1998

26. Lemoine, F.G. and others. "The Development of the Joint NASA GSFC and the National Imagery and Mapping Agency (NIMA) Geopotential Model EGM96", NASA/TP-1998-206861, Goddard Space Flight Center, Greenbelt, Maryland, July 1998.

27. “Minotaur”. http://www.astronautix.com/lvs/minotaur.htm. February 2010

28. “Minotaur”. http://en.wikipedia.org/wiki/Minotaur_III. February 2010

29. “Minuteman”. http://en.wikipedia.org/wiki/Minuteman_missile. February 2010

30. "OptControlCentre." 10 January 2010. http://www.optcontrolcentre.com/html/nlp_solvers.html.

31. Rao, A. V., Benson, D. A., Darby C. L., Patterson, M. A., Francolin, C., Sanders, I., and Huntington, G. T., “*GPOPS*: A MATLAB Implementation of the Gauss Pseudospectral Method for Solving Multiple-Phase Optimal Control Problems,” *ACM Transactions on Mathematical Software,* submitted for publication (October 2008).

32. Rao, A. V., Benson, D. A., Darby C. L., Patterson, M. A., Francolin, C., Sanders, I., and Huntington, G. T., “*GPOPS*: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method,” *ACM Transactions on Mathematical Software,* Vol. 37, No. 2, to appear.

33. 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.

34. Shen, Zuojun and Ping Lu. “Onboard Generation of Three-Dimensional Con-strained Entry Trajectories," Journal of Guidance, Control, and Dynamics,26(1):111-121, January-February 2003.

35. “Space Shuttle”. http://en.wikipedia.org/wiki/Space_shuttle. February 2010

36. U.S. Standard Atmosphere, 1976, U.S. Government Printing Office, Washington, D.C., 1976.

37. “X-33”. http://en.wikipedia.org/wiki/X-33. February 2010

38. “X-37”. http://en.wikipedia.org/wiki/X-37. February 2010

39. 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.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
25-03-2010	Master's Thesis	August 2008 – March 2010

TITLE AND SUBTITLE

Simulation and Application of GPOPS for Trajectory Optimization and Mission Planning Tool

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Yaple, Danielle E., 2nd Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENY)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/10-M29

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

Undisclosed Sponsor

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

Rapid trajectory generation is crucial to prompt global warfare. To meet the USAF's objective of *Persistent and Responsive Precision Engagement*, a rapid mission planning tool is required. This research creates the framework for the mission planning tool and provides a sample optimal trajectory which is solved using the GPOPS software package. GPOPS employs a Gaussian psuedospectral method to solve the non-linear equations of motion with both end conditions and path constraints. By simultaneously solving the entire trajectory based on an initial guess and small number of nodes, this method is ideal for generating rapid solutions. The sample case is a multi-phase minimum time optimal control problem which is used to validate the planning tool. The developed framework includes different atmospheric models, gravity models, inclusion of no-flyzones and waypoints and the ability to create a library of sample cases. This versatile tool can be used for either trajectory generation or mission analysis.

The results of this research show the complexities in solving an optimal control problem with states that change from one phase of the solution to another. The final resulting trajectory is calculated from a sectioned method, allowing changes in states to be done outside of the optimal control problem. This method should be the foundation for a state varying complete optimal control problem and the mission planning tool.

15. SUBJECT TERMS

Optimal Control; Trajectory; GPOPS; Psuedospectral Methods

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Richard Cobb, ADVISOR
U	U	U	UU	114	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 4559 (Richard.Cobb@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18