

THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF AEROSPACE ENGINEERING

TRAJECTORY OPTIMIZATION FOR AN IDEAL SOLAR SAIL MODEL USING
PSEUDOSPECTRAL OPTIMAL CONTROL

ALEX J. PINI

Spring 2010

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Aerospace Engineering
with honors in Aerospace Engineering

Reviewed and approved* by the following:

Robert Melton
Professor of Aerospace Engineering
Thesis Supervisor

David Spencer
Associate Professor of Aerospace Engineering
Faculty Reader

George Lesieutre
Professor of Aerospace Engineering
Department Head / Honors Advisor

*Signatures are on file in the Schreyer Honors College

Abstract

Solar sails are currently being analyzed to determine if they are a feasible alternative to more conventional means of space propulsion, such as chemical rockets. In this thesis, a minimum-time optimal solution is obtained for a sail traveling from Earth to Venus. This solution consists of the state and control histories of the sail, as well as the shape of the trajectory and the time of flight. Direct collocation is used, specifically a pseudospectral method, to numerically generate an optimal solution for the sail's governing equations of motion. The pseudospectral method is determined to be an effective means of solving these differential equations. The optimality of the solutions is verified by comparison of the Hamiltonian behavior to what is described in optimal control theory. The results are analyzed, noting some anomalies and offering possible explanations. Finally, the feasibility of the parameters used in the calculations is discussed based upon the current technology level of solar sails.

Table of Contents

TABLE OF CONTENTS	II
LIST OF FIGURES	IV
LIST OF TABLES	V
ACKNOWLEDGEMENTS	VI
CHAPTER 1: INTRODUCTION.....	1
1.1 Problem Statement	1
1.2 Thesis Objective.....	2
CHAPTER 2: PROBLEM FORMULATION	4
2.1 Coordinate Systems and Sail Parameters.....	4
2.2 Solar Sail Equations of Motion.....	7
2.3 Optimal Control Problem Statement.....	12
2.4 Scaling of Variables	14
CHAPTER 3: METHOD OF ANALYSIS.....	16
3.1 Direct Collocation with Nonlinear Programming.....	16
3.2 Pseudospectral Optimal Control	18
3.3 Conditions for Optimal Control	18
3.4 Problem Formulation in DIDO	19
CHAPTER 4: NUMERICAL EXAMPLES: RESULTS AND DISCUSSION	21
4.1 High Accuracy Time-Optimal Solution.....	21
4.2 The Effect of Sail Lightness Number on Optimal Results.....	26
4.3 Feasibility of Results.....	29
CHAPTER 5: CONCLUSIONS AND FUTURE WORK.....	31
5.1 Conclusions	31
5.2 Future Work Recommendations	32
APPENDIX A: MATLAB SOURCE – PERFECTLY REFLECTIVE.....	34
A.1 Problem File.....	34
A.2 Dynamics File	38
A.3 Events File.....	39
A.4 Path File	40

A.5 Cost File	41
APPENDIX B: VARIATION OF OPTIMAL RESULTS WITH SLN	42
REFERENCES.....	62

List of Figures

Figure 1: Solar Sail Position in a Newtonian Reference Frame	4
Figure 2: Solar Sail Position in a Cylindrical Coordinate System.....	5
Figure 3: The \hat{n} Vector and Associated Control Angles.....	9
Figure 4: Representation of Sail Position in a Rotating Frame	10
Figure 5: Direct Collocation Using a Piecewise Polynomial.....	17
Figure 6: Sail Trajectory with 81 Nodes for a SLF = 0.1	22
Figure 7: Hamiltonian as a Function of Time for 81 Nodes and SLF = 0.1	23
Figure 8: Control Angle (α) as a Function of Time for 81 Nodes and SLF = 0.1	24
Figure 9: Control Angle (γ) as a Function of Time for 81 Nodes and SLF = 0.1.....	25
Figure 10: Time of Flight as a Function of Sail Lightness Number Using 41 Nodes	27
Figure 11: Optimal Solution when SLF = 0.10	43
Figure 12: Optimal Solution when SLF = 0.15	44
Figure 13: Optimal Solution when SLF = 0.20	45
Figure 14: Optimal Solution when SLF = 0.25	46
Figure 15: Optimal Solution when SLF = 0.30	47
Figure 16: Optimal Solution when SLF = 0.35	48
Figure 17: Optimal Solution when SLF = 0.40	49
Figure 18: Optimal Solution when SLF = 0.45	50
Figure 19: Optimal Solution when SLF = 0.50	51
Figure 20: Optimal Solution when SLF = 0.55	52
Figure 21: Optimal Solution when SLF = 0.60	53
Figure 22: Optimal Solution when SLF = 0.65	54
Figure 23: Optimal Solution when SLF = 0.70	55
Figure 24: Optimal Solution when SLF = 0.75	56
Figure 25: Optimal Solution when SLF = 0.80	57
Figure 26: Optimal Solution when SLF = 0.85	58
Figure 27: Optimal Solution when SLF = 0.90	59
Figure 28: Optimal Solution when SLF = 0.95	60
Figure 29: Optimal Solution when SLF = 1.00	61

List of Tables

Table 1: Time of Flight as a Function of Sail Lightness Number Using 41 Nodes..... 28

Acknowledgements

First and foremost, I would like to thank Dr. Robert Melton for his assistance regarding all facets of this thesis. After countless meetings in his office to discuss the content described in this thesis, I am led to believe that his seemingly endless wisdom is surpassed only by his patience. His accessibility and general attitude towards students are two of the things that set him apart from any teacher or professor I've ever had, and it's a shame that many fail to recognize how important these "smaller" things can be. Throughout my time at Penn State, there has been no greater influence on my development as an engineer and resulting achievements, and he's been an outstanding professor, mentor, and friend. I am truly indebted to him.

I would also like to thank my parents for their multiple forms of support throughout my college career. One of the most important things that I've realized over the past four years is that parents with their level of dedication are hard to come by. I finally understand why they have pushed me throughout my life to try my hardest at everything I do. They are living proof of how much a person can accomplish if he is determined and willing to work for it, and I have learned a lot from them in this regard. I haven't yet fulfilled my dreams, but thanks to their help I am starting to see the fruits of my labor materialize.

Chapter 1: Introduction

In this chapter, a basic introduction of solar sails is provided. Also, the rationale for investigating the optimization problem addressed in this thesis is discussed. The objective of this thesis is then presented, along with key assumptions and points of consideration regarding the setup of the problem.

1.1 Problem Statement

A solar sail is a method of space propulsion that utilizes solar radiation pressure to produce thrust. The acceleration generated from photon collisions is relatively weak (on the order of mm/s^2), so solar sails are categorized as a form of “low-level thrust.” Solar sails are most effective over long distances, when their small constant accelerations allow for high velocities to accumulate over time. An additional advantage of solar sails is that there is no need for propellant, which often accounts for a large portion of a spacecraft’s mass. Since launch costs are directly proportional to the mass of the payload, a significant amount of money can be saved in this regard if a solar sail is utilized.

These are a few of the main reasons why solar sails have been investigated as a viable alternative to more conventional methods, namely rocket propulsion. One of the most critical metrics used to compare the two forms of propulsion is the time of flight required to reach a specified destination from Earth. Obtaining minimal-time solutions for sail

trajectories will provide some insight into whether or not implementation of a full-scale solar sail is worthwhile.

1.2 Thesis Objective

The primary objective of this research is to obtain the control history, which is the set of sail control angles at discrete points along the path, for the minimal-time optimal trajectory from Earth to Venus. The time of flight for the optimal trajectory and the shape of the trajectory are also points of interest for this problem. There are a few major assumptions/notes to mention when formulating this problem:

- In reality, a solar sail mission from Earth to Venus would typically start from an orbit around Earth and end in an orbit around Venus. For simplicity, the sail is assumed to start at Earth's center and end at the center of Venus.
- The orbits of Earth and Venus are perfectly circular, having an eccentricity of zero. In reality, the orbits of the two planets have eccentricities slightly above zero. Since the assumption does not stray too far from reality in this regard, it does not have a significant effect on the accuracy of the overall solution. Considering the eccentricities of the orbits in this problem increases the complexity of the problem formulation for only a marginal rise in the quality of the answer.

- The orbits of the two planets, although perfectly circular, will not be coplanar. This means that the inclinations of the two orbits will be exactly as they are in reality and the problem will be three-dimensional.
- The solar sail model used is perfectly flat and made of material that is perfectly reflective. This configuration is considered to be the ideal model for a solar sail because it gains the most momentum from collisions with photons. As a result, the ideal model has the best performance and the optimal results generated from using it can be used as a benchmark for more realistic non-ideal models.

The results described above will be obtained for ideal sails with different performance characteristics. With these results, conclusions can be drawn about the effect that certain parameters have on the dynamics of the sail, as well as the accuracy of the numerical method used to obtain the optimal solutions.

Chapter 2: Problem Formulation

In this chapter, the nonlinear differential equations of motion for a planar, perfectly reflecting solar sail in a heliocentric orbit will be derived, along with critical performance parameters for solar sails. Additionally, the equations of motion will be developed in a rotating reference frame and in cylindrical coordinates for convenience. The state and control variables for this configuration will also be established. Finally, the characteristic length and time units that will be used in this problem are introduced.

2.1 Coordinate Systems and Sail Parameters

In order to characterize the motion of a solar sail around the sun, a coordinate system must first be established. The position of a solar sail in a Newtonian, or inertial, reference frame can be defined as follows.

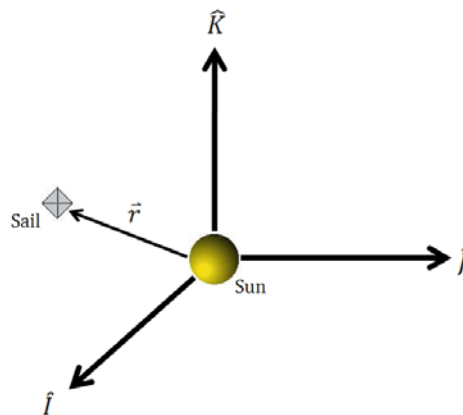


Figure 1: Solar Sail Position in a Newtonian Reference Frame

Since this optimization problem deals with circular motion around the sun, which is centered at the origin, polar coordinates are the most convenient candidate to use. However, the sail is not restricted to move in only two dimensions so a cylindrical coordinate system (with components ρ , θ , and z) will be utilized.

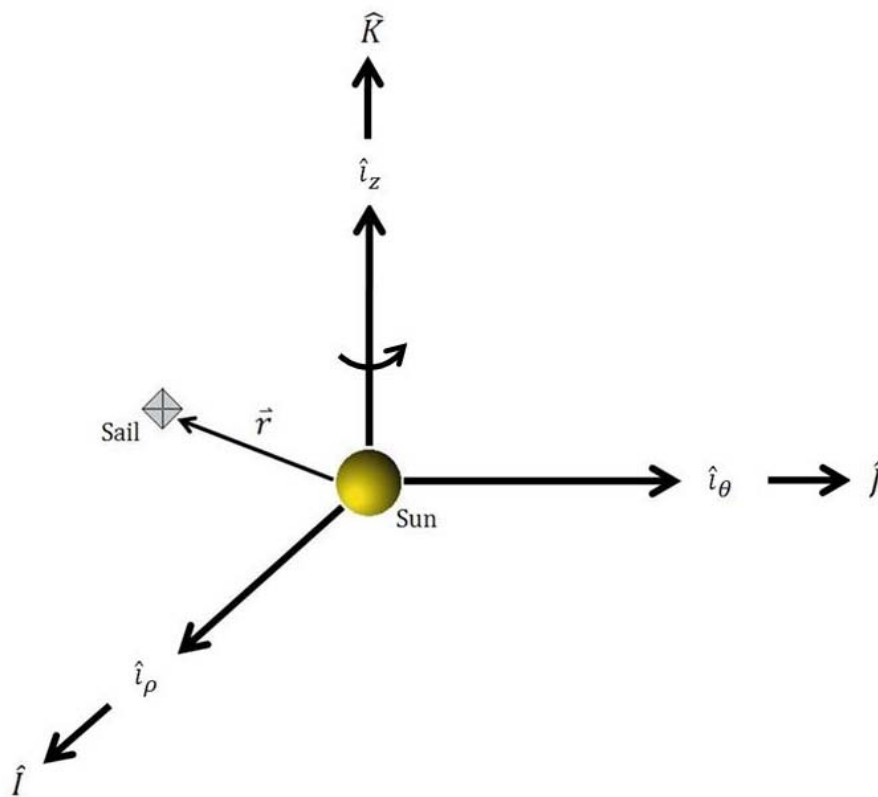


Figure 2: Solar Sail Position in a Cylindrical Coordinate System

There are a few parameters which have an influence on a solar sail's performance that should be introduced [1]. These parameters are constant for any given sail and are largely

dependent on the sail material used. This relationship will be explored in further detail in Chapter 4. The first such parameter is called the sail loading and is simply the total spacecraft mass per unit area.

$$\sigma = \frac{m}{A} \quad (2.1)$$

The sail loading is a useful tool to have when calculating the characteristic acceleration, another performance metric, and when determining the appropriate size of a solar sail. Sizing of solar sails is outside the scope of this thesis and will not be addressed, but the characteristic acceleration can be expressed as follows:

$$a_0 = \frac{9.12\eta}{\sigma[g * m^{-2}]} [mm * s^{-2}] \quad (2.2)$$

The variable η represents the efficiency of the solar sail and is used to account for any imperfections that may exist in a sail model. Since the sail model is assumed to be ideal for this problem, an efficiency of 1 will be used. The characteristic acceleration is defined as the acceleration that a sail experiences at 1 astronomical unit (AU) with the sail surface oriented perfectly normal to the sun-line. The characteristic acceleration does not have a direct effect on the dynamics of the sail, but is a more concrete way of evaluating performance than the sail loading and will also be revisited in Chapter 4.

The parameter that serves as the best indicator of performance is known as the sail lightness number (SLN), the sail lightness factor (SLF), or simply the lightness number. These three terms will be used interchangeably throughout the thesis. The lightness number

is simply the ratio of the repulsive acceleration on the sail provided by solar radiation pressure to the gravitational acceleration exerted on the sail by the sun.

$$\beta = \frac{a_{radiation}}{a_{gravitation}} \quad (2.3)$$

Since both accelerations vary with the sail's distance from the sun according to an inverse square relationship, the SLN is a dimensionless constant. As with the other parameters defined earlier, the lightness number is constant for any specific sail and is primarily dependent on the materials of which the sail is composed. Since the acceleration due to gravity at a given location will be the same for every sail, the lightness number can be considered a measure of a sail's ability to produce an acceleration from exposure to a given amount photons. Lightness numbers typically range from 0 to 1, with higher numbers correlating to higher performance. It will be shown in the next section that the lightness number appears in the equations that govern a sail's motion and has a significant influence on the dynamics of the sail as a result.

2.2 Solar Sail Equations of Motion

The equation of motion for a perfectly reflecting solar sail in a heliocentric orbit [1] can be shown to be:

$$\frac{d^2\vec{r}^N}{dt^2} + \frac{\mu}{r^2}\hat{r} = \beta \frac{\mu}{r^2}(\hat{r} \cdot \hat{n})^2\hat{n} \quad (2.4)$$

In this equation, $\frac{d^2\hat{r}^N}{dt^2}$ represents the acceleration of the sail with respect to the sun in the Newtonian frame of reference. The variable r is the radial distance between the sail and the sun and \hat{r} is the unit vector directed along the sun-sail line. As mentioned before, β is the sail lightness factor and μ is the gravitational parameter. The gravitational parameter is defined as:

$$\mu = G(m_{sun} + m_{sail}) \quad (2.5)$$

where G is the gravitational constant. However, the sail is assumed to have a much smaller mass than the sun, so m_{sail} can be considered negligible and the gravitational parameter becomes:

$$\mu = Gm_{sun} \quad (2.6)$$

The \hat{n} vector is the unit vector normal to the surface of the sail that is *not* exposed to photons. This is the direction in which the sail experiences the acceleration from the solar radiation pressure, since the acceleration can only be directed perpendicular to the surface of a planar sail. The \hat{n} vector is shown in Figure 3.

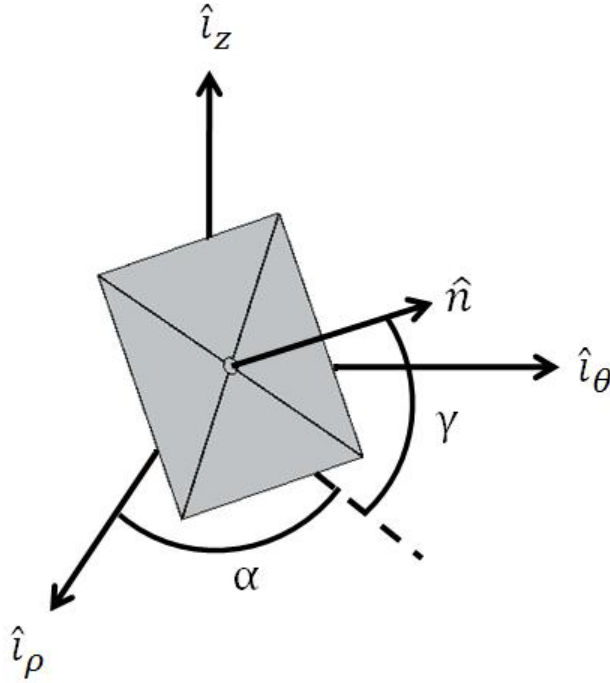


Figure 3: The n-hat Vector and Associated Control Angles

The angles α and γ are referred to as the control angles for the sail and their significance will be discussed in the next section. Regardless, the normal vector can be expressed in terms of the cylindrical coordinates and control angles, viz.

$$\hat{n} = \cos \gamma \cos \alpha \hat{i}_\rho + \cos \gamma \sin \alpha \hat{i}_\theta + \sin \gamma \hat{i}_z \quad (2.7)$$

Although Equation 2.4 is expressed in terms of the Newtonian reference frame, the problem at hand is most conveniently formulated in a rotating frame. The geometrical representation of this transformation can be established by altering Figure 2 to reflect a rotation through an arbitrary angle Θ :

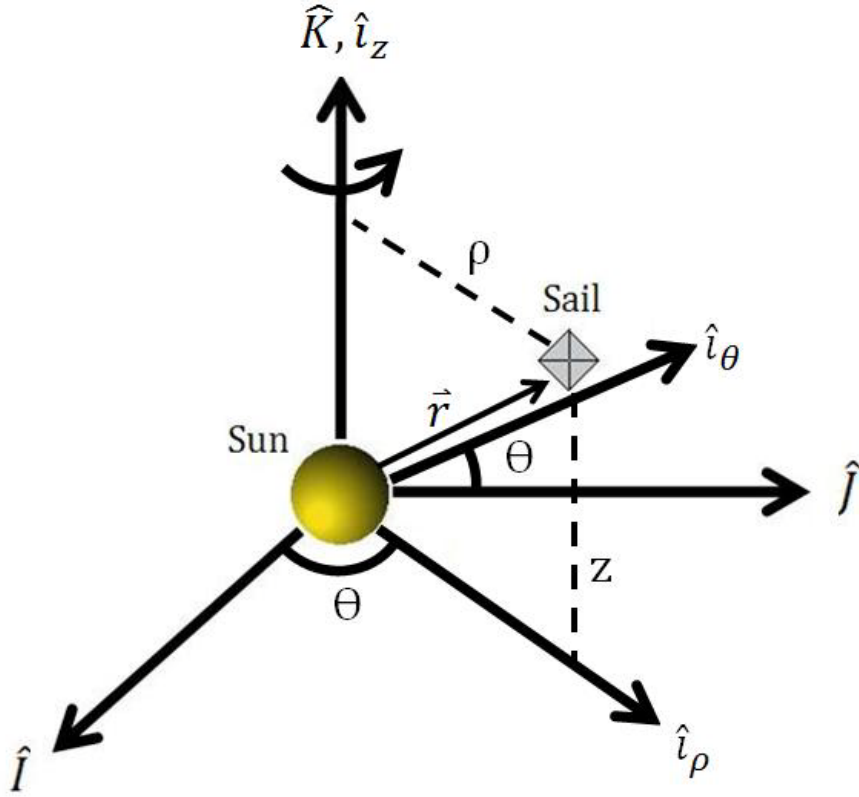


Figure 4: Representation of Sail Position in a Rotating Frame

Equation 2.4 can be transformed from the inertial to the rotating, or body, frame [2] by solving for the inertial acceleration term, $\frac{d^2\vec{r}^N}{dt^2}$, in the following equation.

$$\frac{d^2\vec{r}^B}{dt^2} = \frac{d^2\vec{r}^N}{dt^2} - (\dot{\vec{\omega}}^{B/N} \times \vec{r}) - \left(2\vec{\omega}^{B/N} \times \frac{d\vec{r}^B}{dt}\right) - \left[\vec{\omega}^{B/N} \times (\vec{\omega}^{B/N} \times \vec{r})\right] \quad (2.8)$$

In Equation 2.8, $\vec{\omega}^{B/N}$ represents the angular velocity of the body frame with respect to the Newtonian frame. By the same token, $\dot{\vec{\omega}}^{B/N}$ denotes the angular velocity of the rotating frame with respect to the Newtonian frame. Finally, $\frac{d\vec{r}^B}{dt}$ and $\frac{d^2\vec{r}^B}{dt^2}$ are the velocity and

acceleration of the sail in the rotating frame, respectively. The terms of Equation 2.8 take on the following values:

$$\vec{\omega}^{B/N} = \dot{\theta} \hat{i}_z \quad (2.9a)$$

$$\dot{\vec{\omega}}^{B/N} = \ddot{\theta} \hat{i}_z \quad (2.9b)$$

$$\vec{r} = \rho \hat{i}_\rho + z \hat{i}_z \quad (2.9c)$$

$$\frac{d\vec{r}^{B}}{dt} = \dot{\rho} \hat{i}_\rho + \dot{z} \hat{i}_z \quad (2.9d)$$

$$\frac{d^2\vec{r}^{B}}{dt^2} = \ddot{\rho} \hat{i}_\rho + \ddot{z} \hat{i}_z \quad (2.9e)$$

By substituting these terms into Equation 2.8 and performing a few operations, the expression for the inertial acceleration in terms of a rotating frame in cylindrical coordinates can be established.

$$\frac{d^2\vec{r}^N}{dt^2} = (\ddot{\rho} - \rho\dot{\theta}^2) \hat{i}_\rho + (\ddot{\theta}\rho + 2\dot{\theta}\dot{\rho}) \hat{i}_\theta + \ddot{z} \hat{i}_z \quad (2.10)$$

Finally, the results of Equation 2.10 can be inserted into Equation 2.4. By substituting the values for \hat{r} and \hat{n} , three nonlinear equations of motion in cylindrical component form emerge.

$$(\ddot{\rho} - \rho\dot{\theta}^2) \hat{i}_\rho + \left[\frac{\mu\rho}{(\rho^2 + z^2)^{3/2}} \right] \hat{i}_\rho - \beta \frac{\mu}{\rho^2 + z^2} \left(\frac{\rho \cos \gamma \cos \alpha + z \sin \gamma}{\sqrt{\rho^2 + z^2}} \hat{i}_\rho \right)^2 \cos \gamma \cos \alpha = 0 \quad (2.11a)$$

$$(\ddot{\theta}\rho + 2\dot{\theta}\dot{\rho}) \hat{i}_\theta - \beta \frac{\mu}{\rho^2 + z^2} \left(\frac{\rho \cos \gamma \cos \alpha + z \sin \gamma}{\sqrt{\rho^2 + z^2}} \hat{i}_\theta \right)^2 \cos \gamma \sin \alpha = 0 \quad (2.11b)$$

$$\ddot{z} \hat{i}_z + \left[\frac{\mu z}{(\rho^2 + z^2)^{3/2}} \right] \hat{i}_z - \beta \frac{\mu}{\rho^2 + z^2} \left(\frac{\rho \cos \gamma \cos \alpha + z \sin \gamma}{\sqrt{\rho^2 + z^2}} \hat{i}_z \right)^2 \sin \gamma = 0 \quad (2.11c)$$

Equations 2.11a-c are the three nonlinear equations of motion for a solar sail in a heliocentric orbit expressed in cylindrical coordinates for a rotating body frame.

2.3 Optimal Control Problem Statement

The objective of this problem is to determine the control history, $\vec{u}(t)$, that allows for a minimal time solution to the equations of motion (Equations 2.11a-c) subject to a set of constraints. The control history is the set of control angles, which were defined in Section 2.2, that are determined at each discrete point, or node, along the trajectory.

$$\vec{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} \alpha(t) \\ \gamma(t) \end{bmatrix} \quad (2.12)$$

To assist the optimal control software in converging towards a solution, the control angles were broken up into their trigonometric components. This gives the software more flexibility when determining the control history. Equation 2.12 can be altered so that the control history is now expressed as:

$$\vec{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ u_4(t) \end{bmatrix} = \begin{bmatrix} \cos \alpha(t) \\ \sin \alpha(t) \\ \cos \gamma(t) \\ \sin \gamma(t) \end{bmatrix} \quad (2.13)$$

However, an additional constraint is required to ensure that the trigonometric components for each control angle are compatible at each node in the trajectory.

$$\cos^2 \alpha(t) + \sin^2 \alpha(t) = 1 \quad (2.14a)$$

$$\cos^2 \gamma(t) + \sin^2 \gamma(t) = 1 \quad (2.14b)$$

To ensure that the trajectory of the sail coincides with Venus's orbit, constraints need to be placed on the final values of the sail's position, velocity and inclination [3]. This is accomplished through the following equations:

$$\rho_f^2 + z_f^2 = r_v^2 \quad (2.15a)$$

$$\dot{\rho}_f^2 + (\rho_f \dot{\theta}_f)^2 + \dot{z}_f^2 = \frac{\mu}{r_v} \quad (2.15b)$$

$$\frac{\rho_f^2 \dot{\theta}_f}{\sqrt{\rho_f^2 z_f^2 \dot{\theta}_f^2 + (\dot{\rho}_f z_f - \rho_f \dot{z}_f)^2 + \rho_f^4 \dot{\theta}_f^2}} = \cos i_v \quad (2.15c)$$

where the “f” subscript denotes the final values of the state variables. The state vector, \vec{x} , is composed of the cylindrical components represented in Figure 2 and their first time derivatives.

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \rho \\ \dot{\rho} \\ \theta \\ \dot{\theta} \\ z \\ \dot{z} \end{bmatrix} \quad (2.16)$$

Equations 2.11a-c are best formulated in terms of these state variables, as the optimal control software requires the equations of motion in the form $\dot{\vec{x}} = f(\vec{x}, \vec{u})$ [4]. By substituting the state variables from Equation 2.15 into the equations of motion represented in Equations 2.11a-c, the equations of motion in state variable form can be obtained after some manipulation.

$$\dot{\vec{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} x_1 x_3^2 - \frac{\mu x_1}{(x_1^2 + x_5^2)^{3/2}} + \beta \frac{x_2 \mu}{x_1^2 + x_5^2} \left(\frac{\rho u_3 u_1 + z u_4}{\sqrt{x_1^2 + x_5^2}} \right)^2 u_3 u_1 \\ \frac{1}{x_1} * \left\{ -2x_1 x_4 + \beta \frac{x_4 \mu}{x_1^2 + x_5^2} \left(\frac{\rho u_3 u_1 + z u_4}{\sqrt{x_1^2 + x_5^2}} \right)^2 u_3 u_2 \right\} \\ - \frac{\mu x_5}{(x_1^2 + x_5^2)^{3/2}} + \beta \frac{x_6 \mu}{x_1^2 + x_5^2} \left(\frac{\rho u_3 u_1 + z u_4}{\sqrt{x_1^2 + x_5^2}} \right)^2 u_4 \end{bmatrix} \quad (2.17)$$

Equation 2.17 marks the farthest that the equations of motion need to be developed in order for the optimal control software to solve for the minimal time solution. This set of equations fully describes the dynamics of a solar sail in a heliocentric orbit using cylindrical coordinates in a rotating reference frame.

2.4 Scaling of Variables

With many problems involving orbital mechanics, the numbers involved will be very large (in the case of distances) and very small (in the case of angular velocities) if conventional SI units are utilized to represent these values. The routines that are run by the optimal control

software are processor-intensive to begin with, but incorporating very large numbers into these calculations will result in unnecessarily long computation times. Also, the optimal control algorithm employs a particular representation of the variables in the problem such that they need to be scaled to approximately the same order of magnitude. Another problem associated with performing operations on very large and very small numbers is that there are high values of truncation and roundoff errors. To avoid these problems, canonical units are introduced.

First, the characteristic length unit will be an astronomical unit (AU), which is the mean distance from the earth to the sun. Also, the characteristic time unit will be a year. Finally, the gravitational parameter, μ , which is dependent on the length and time units via the equation for the orbit's period, can be determined to be $4\pi^2$. Using these units allows for the relevant variables to be expressed by more reasonable values, simplifying the necessary calculations [5].

Chapter 3: Method of Analysis

In this chapter, the general concept of direct collocation with nonlinear programming will be discussed, along with how it is used to generate approximate solutions to optimal control problems. The pseudospectral method, which is used to solve the problem that this thesis addresses, will also be examined. The necessary conditions for optimal control will then be defined in order to help validate the results obtained from the pseudospectral method. Finally, a summary of DIDO, the optimal control software utilized for this problem, will be provided.

3.1 Direct Collocation with Nonlinear Programming

Direct collocation is a method of implicitly solving the differential equations that were derived in Section 2.3 by approximating the optimal trajectory as a series of polynomials [6]. This trajectory is partitioned into segments that are separated by nodes of length T , where the polynomials, which are functions of the state and control variables defined in Chapter 2, must satisfy the differential equations of motion. In other words, the values at the left and right nodes for any segment must match those defined by the equations of motion such that there is a smooth, continuous curve connecting any two segments.

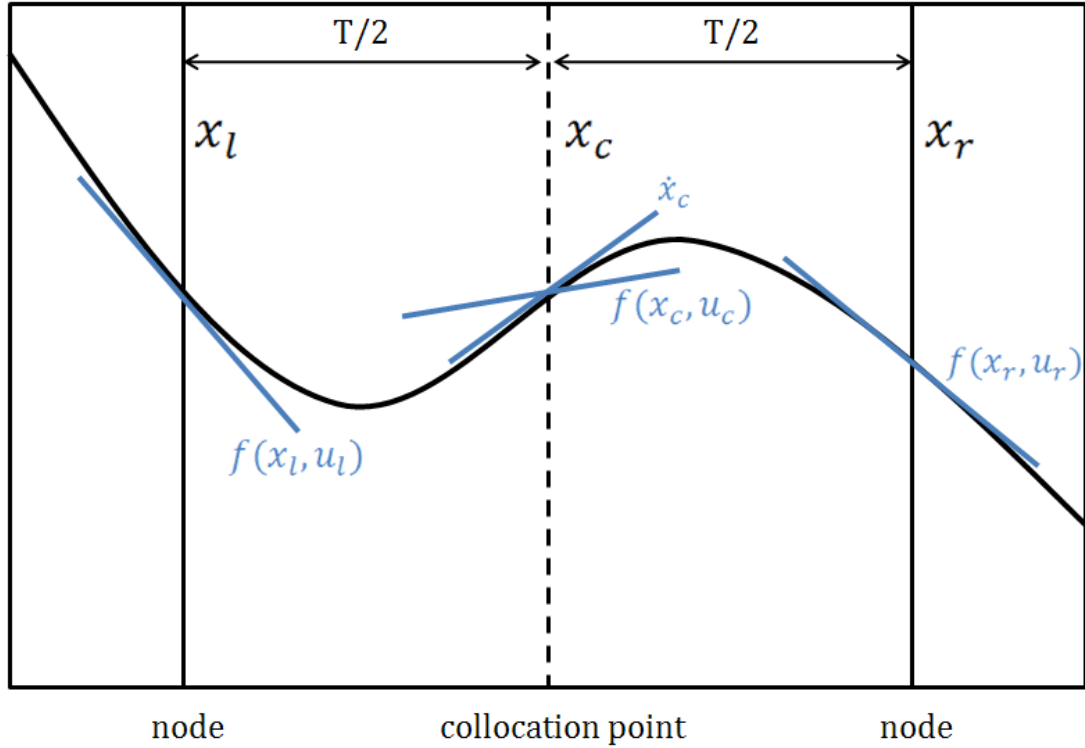


Figure 5: Direct Collocation Using a Piecewise Polynomial

In the middle of each segment, a collocation point is established and the derivative of the polynomials, \dot{x}_c , is compared to the derivative obtained from the equations of motion [7]. The defect, Δ , which is a measure of how well the polynomials represent the solution to the equations of motion, is defined as:

$$\Delta = f(x_c, u_c) - \dot{x}_c \quad (3.1)$$

When the state and control variables are determined such that the defect is below a specified tolerance, the polynomials are considered to be an accurate approximation of the solution to the differential equations.

3.2 Pseudospectral Optimal Control

The pseudospectral method is a type of direct collocation and will be the means of determining the optimal trajectory for this problem [4]. This method differs from the one described in Section 3.1 in that the optimal trajectory is a sum of Chebyshev polynomials with unknown coefficients. The optimal solution is found when the unknown polynomial coefficients are determined.

The number of nodes that the trajectory is divided into is equivalent to the number of polynomials that are used to represent the solution. Additionally, this number also denotes the order of these Chebyshev polynomials. Therefore, an approximation with many nodes will be represented by high-order polynomials and will produce a solution that has a relatively high accuracy.

3.3 Conditions for Optimal Control

In optimal control theory, the Hamiltonian is a useful tool for verifying that a solution approximately meets the necessary conditions for optimality. The Hamiltonian is composed of the system states, \vec{x} , and controls, \vec{u} , that were defined in Section 2.3 [8].

$$\mathcal{H} = 1 + \lambda_{x_n} f(\vec{x}, \vec{u}) \quad (3.2)$$

In the context of this problem, the Hamiltonian can be fully expressed as:

$$\mathcal{H} = 1 + \lambda_{x_1} \dot{x}_1 + \lambda_{x_2} \dot{x}_2 + \lambda_{x_3} \dot{x}_3 + \lambda_{x_4} \dot{x}_4 + \lambda_{x_5} \dot{x}_5 + \lambda_{x_6} \dot{x}_6 \quad (3.3)$$

where $\dot{x}_1, \dot{x}_2, \dots$ represent the equations of motion in state variable form that were established in Equation 2.17 and $\lambda_{x_1}, \lambda_{x_2}, \dots$ are referred to as their corresponding costates and are a form of Lagrange multipliers. To obtain a comprehensive idea of how accurate the approximate solution is, the Hamiltonian is generated at each node along the trajectory using Equation 3.3. For a minimum-time optimization problem such as this, the Hamiltonian should be constant over time and exactly equal to -1 [4].

3.4 Problem Formulation in DIDO

DIDO is a piece of optimal control software that runs through MATLAB. It accepts the equations of motion for the solar sail in state variable form, along with initial values and constraints that bound the problem and restrict the motion of the sail. It then employs the pseudospectral method described in Section 3.2 to produce a potentially optimal solution [4]. This solution consists of the optimal state variables and control variables at each node, along with the total time of flight. DIDO also calculates the costates at each node so the Hamiltonian can be evaluated. In order to input the appropriate data into DIDO, several files must be created, as shown below.

1. Dynamics File: This file contains the equations of motion for the sail in state variable form, as expressed in Equation 2.17.

2. Events File: This file contains the initial conditions for the state variables, along with the constraints that ensure the final position, velocity, and inclination of the sail's trajectory coincide with the corresponding values for Venus.

3. Path File: This file defines the constraints on the sail's control angles, as defined in Equations 2.14a-b.

4. Cost File: This file accepts a structure array from MATLAB containing the states, controls, and node locations. The output is the endpoint cost, which is essentially the time of flight for the optimal solution in the context of this problem.

5. Problem File: This is the main file that is in charge of calling the other files and is the only one that actually gets run. It also includes constraints on the maximum and minimum values that the state and control variables can have throughout the trajectory, as well as the maximum time of flight. The initial values for the state variables are also defined in this file, as well as parameters involving the pseudospectral method, such as providing initial guesses for the solution or setting the number of nodes to be used. Finally, all of the data processing is done at the end of the file.

The source code for these files is included in Appendix A.

Chapter 4: Numerical Examples: Results and Discussion

This chapter contains all of the optimal control solutions generated by DIDO. First, a solution with 81 nodes is obtained to display the software's capability to produce accurate results. Then, slightly less accurate solutions are calculated which investigate the effect that lightness number has on the dynamics of the sail, shape of the trajectory, and optimal time of flight. These results are analyzed and explanations are offered for any anomalies in the data. Finally, the feasibility of the key parameters used in the problem formulation is discussed.

4.1 High Accuracy Time-Optimal Solution

To demonstrate what a relatively accurate optimal solution looks like, DIDO was run with the trajectory separated by 81 nodes. As mentioned in Section 3.2, the number of nodes is equal to the order of the Chebyshev polynomials used to approximate the solution, so this trajectory is composed of a series of 81st order polynomials. Even with the use of canonical units, this solution took a relatively long time to generate and was intended to show DIDO's capability to produce accurate results. It should be noted that the z-axis in Figure 6 is scaled differently than the x and y axes, exaggerating the inclination of Venus. In reality, the two orbits are nearly coplanar, with Venus having an inclination of roughly 3.5° .

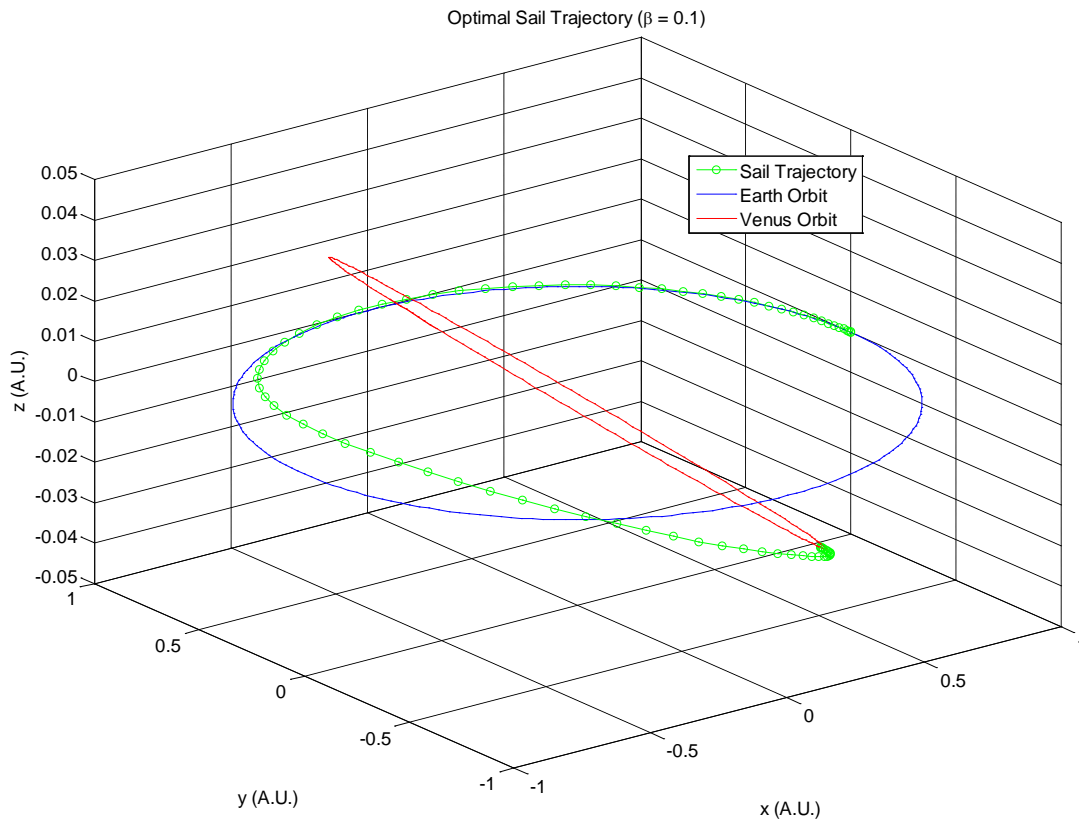


Figure 6: Sail Trajectory with 81 Nodes for a SLF = 0.1

Figure 7 shows how the Hamiltonian behaves over time for the trajectory shown above. As described in Section 3.3, a minimum-time optimal solution will have a Hamiltonian that is constant and equal to -1. Relatively speaking, the fluctuations that are present in the central portion of this plot are extremely small. While the solution generated by the pseudospectral method isn't exactly optimal, the behavior of the Hamiltonian indicates that the solution is extremely accurate.

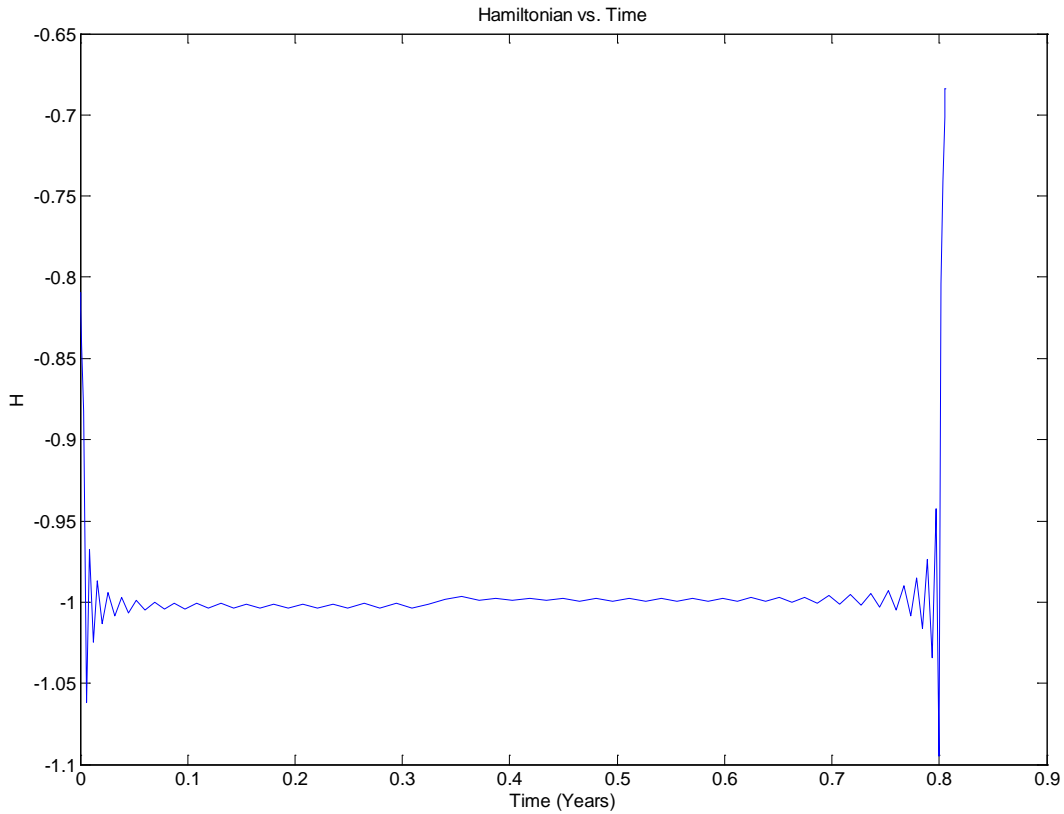


Figure 7: Hamiltonian as a Function of Time for 81 Nodes and SLF = 0.1

Another advantage of a solution with many nodes is that the control history looks smooth and continuous. Figure 8 is a plot of the required control angle α , which is the sail deflection within the instantaneous plane of motion. As shown in the figure, α assumes a negative value for the majority of the sail's flight. This corresponds to the sail being oriented so that the acceleration vector opposes the direction of the sail's motion, gradually reducing its velocity. As a result, the energy and radius of the sail's orbit are reduced to coincide with that of Venus.

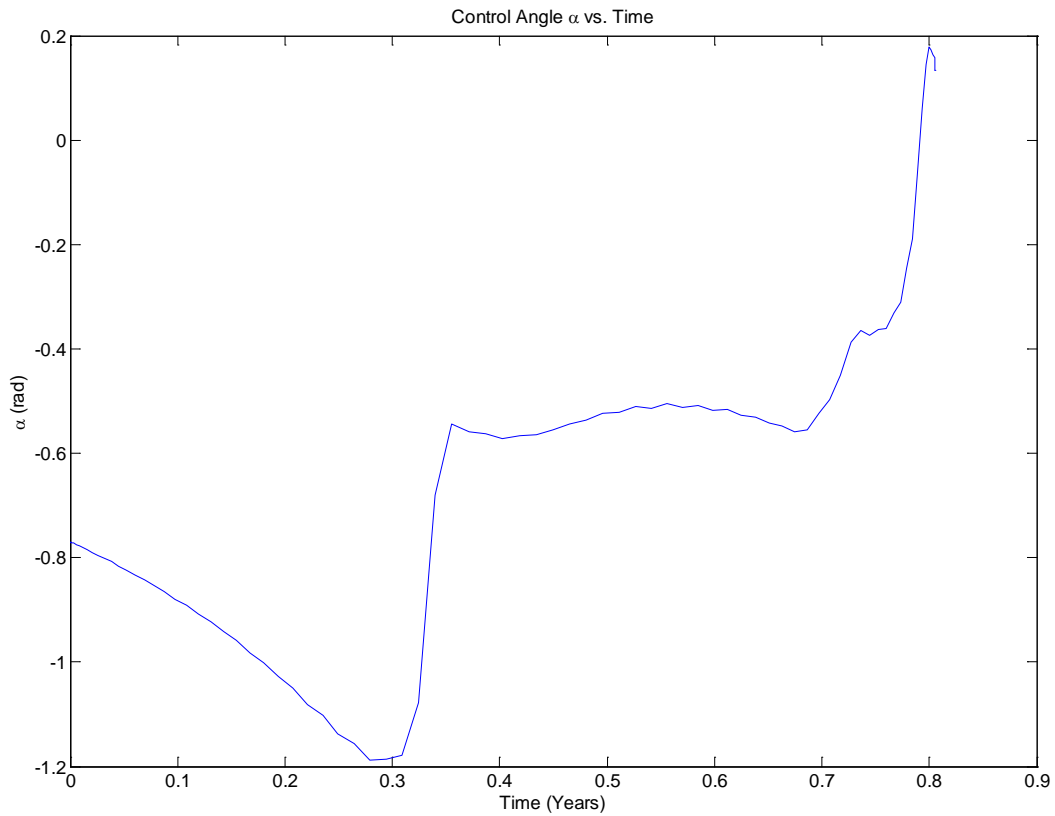


Figure 8: Control Angle (alpha) as a Function of Time for 81 Nodes and SLF = 0.1

Figure 9 shows the control history for γ , the angle that the sail makes in and out of the instantaneous plane of motion. The sail initially has a positive value for γ , which corresponds to a rise in the positive z direction above Earth's orbital plane. Roughly halfway through the trajectory, γ becomes negative and the sail starts to travel in the negative z direction to match the inclination of Venus's orbit.

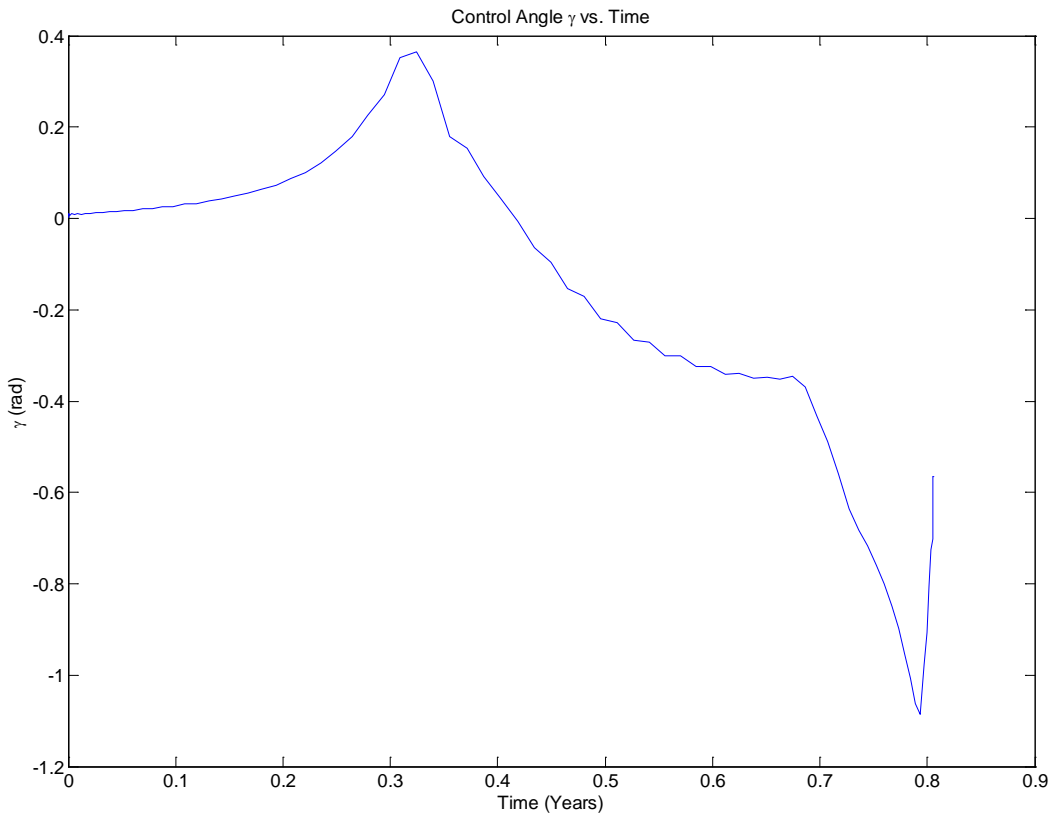


Figure 9: Control Angle (γ) as a Function of Time for 81 Nodes and SLF = 0.1

The reason why the optimal trajectory under these conditions needs to rise above the Earth’s orbital plane before dropping below it again is unclear. As shown in Section 4.2, this is not typically the case when a wide range of scenarios are considered. It is suspected that this trajectory is a product of the relatively low lightness number. Sails with lower lightness numbers will have less “control authority” than those with higher values. Control authority refers to the sail’s ability to maneuver through the forces in nature that try to govern its motion [9]. As a result, an optimal trajectory for a sail with a low lightness number might assume an unorthodox form that is less intuitive. In fact, the pseudospectral method utilized was unable to produce an optimal result for a value of β lower than 0.07.

4.2 The Effect of Sail Lightness Number on Optimal Results

As stated in Section 2.1, the sail lightness number is a measure of a solar sail's performance. To investigate the lightness number's influence on the dynamics of the sail, optimal solutions were generated for an ideal sail with a gradually increasing value of β . By holding all other parameters constant, the effect of lightness number on the optimal results becomes apparent. The solutions (listed in Appendix B) are the pseudospectral approximations using 41 nodes, which allows for faster computational time and a reasonable degree of accuracy.

The effect that the number of nodes has on the quality of the results is instantly noticeable. The optimal results generated using 81 nodes in Section 4.1 are far more smooth and continuous than the 41 node results. With the 41 node solutions, the control angles make sharp adjustments, which are not very practical. However, a considerable amount of accuracy is maintained, as shown by the fact that the Hamiltonian only fluctuates by ± 0.1 at the very most.

There are several trends that appear as the sail lightness factor is increased. The first trend deals with the actual sail trajectory. As mentioned in Section 4.1, the trajectory at low values of β rises fairly high above Earth's orbital plane before dropping back below it. With higher lightness numbers, this motion is less exaggerated or even nonexistent. This is because sails with higher values of β have more control authority and are able to take more direct routes to their destination.

Also, the optimal trajectories tended to become less smooth as lightness number increased. A jagged path is not very practical, especially for solar sails, because

instantaneous direction changes are only relevant when assuming impulsive maneuvers. The reason for this behavior is unknown, but it is speculated that it is a result of the pseudospectral method. High values of β tend to result in increased sensitivity to the control angles by DIDO, so changes in trajectory can be abrupt.

Another thing to note is that the orbit of Venus seems to rotate in a clockwise direction about the z-axis with increasing lightness number. In reality, Venus's orbit does not move in this fashion- this is a byproduct of the problem formulation. The initial state values for the sail's trajectory are fixed and there is no constraint that forces the path to coincide with the longitude of the ascending node for Venus. Therefore, this rotation of Venus's orbit is an indication of the optimal trajectory becoming shorter with increasing lightness number. As a result, the time of flight for the optimal trajectory decreases with growing values of β , as shown in Figure 10.

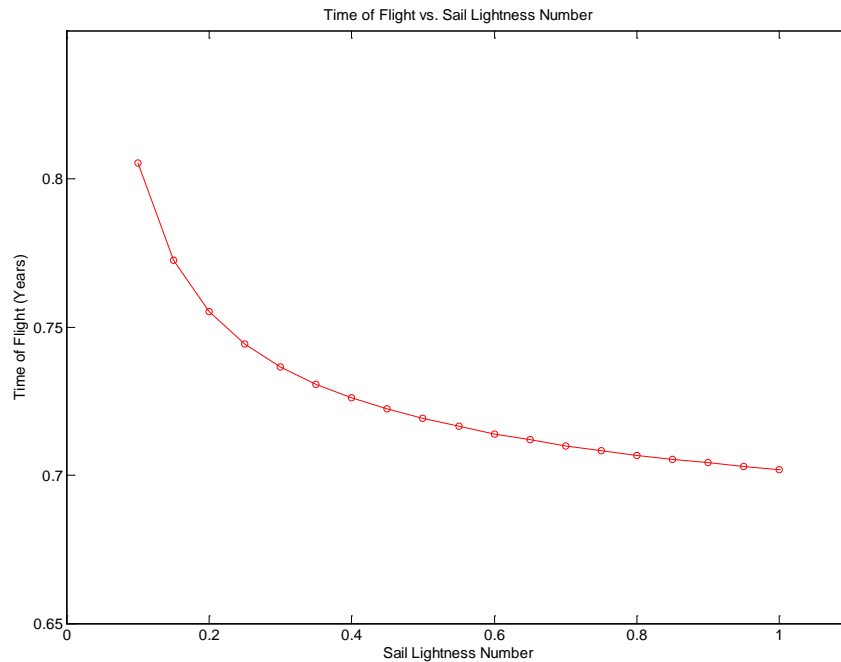


Figure 10: Time of Flight as a Function of Sail Lightness Number Using 41 Nodes

The actual times of flight and their corresponding lightness numbers are included in Table 1.

Table 1: Time of Flight as a Function of Sail Lightness Number Using 41 Nodes

Sail Lightness Number	Time of Flight (Years)
0.10	0.8053
0.15	0.7725
0.20	0.7552
0.25	0.7443
0.30	0.7366
0.35	0.7307
0.40	0.7261
0.45	0.7223
0.50	0.7191
0.55	0.7164
0.60	0.7140
0.65	0.7119
0.70	0.7100
0.75	0.7083
0.80	0.7068
0.85	0.7054
0.90	0.7042
0.95	0.7030
1.00	0.7019

As the lightness number increases, the optimal trajectory time of flight asymptotically approaches approximately 0.7 years. To put this number into perspective, a Hohmann transfer between two circular orbits with the same radius as Earth and Venus would take roughly 0.4 years. However, as mentioned in Section 1.1, the benefits of low-level thrust are seen over long distances, and the trajectories generated are relatively short. Also, the launch costs of the conventional spacecraft needed to perform the Hohmann transfer would be much higher than those of a solar sail.

4.3 Feasibility of Results

The optimal results have been produced for a range of lightness numbers, but as mentioned in Section 2.1, this parameter is largely dependent on the sail material. The ideal material for a solar sail has a low sail loading (σ) and a high reflectivity. The sail loading is the mass per unit area, so thin and lightweight materials prove to be best suited for solar sailing. A common technique involves coating such materials, like Mylar or Kapton, with a thin layer of aluminum to increase reflectivity [10]. The resulting material still has a low value of σ , which corresponds to a large characteristic acceleration via Equation 2.2. The sail lightness factor is directly related to the characteristic acceleration [11] through the following relationship:

$$\beta = 0.168a_0 \quad (4.1)$$

Solar sailing has garnered moderate interest from the scientific community and there have been a handful of missions involving sails over past few decades. However, the only successful missions were testing sail deployment methods. As of the writing of this thesis, there have been no successful missions conducted to test solar sails as a primary form of propulsion for a spacecraft. As a result, there is no experimental data regarding performance characteristics of solar sails in space.

For now, possible values of β must therefore be determined through purely theoretical methods. Sail materials and construction methods [11] have been proposed that could yield lightness numbers of 1. Adding small perforations in these sails that are smaller than the

wavelength of visible light would reduce the mass of the sail without allowing photons to pass through the material. In theory, this technique could raise the lightness number to as high as 10. Even though these numbers are theoretical, it seems as though sails with lightness numbers equivalent to the ones investigated in this thesis could easily be obtained using current manufacturing techniques.

However, these predictions will remain speculative until these sails are actually tested in space. There are currently two missions scheduled for 2010, IKAROS and Lightsail-1, that will obtain this critical performance data if successful.

Chapter 5: Conclusions and Future Work

This chapter provides some concluding remarks regarding the numerical method used to solve the optimal control problem in this thesis. Also, the software used to implement the pseudospectral method is discussed. Finally, the results obtained from this software are assessed.

5.1 Conclusions

In this thesis, it was determined that pseudospectral optimal control was an effective method of determining a minimal-time solution to the nonlinear equations of motion for a solar sail. DIDO proved to be an effective tool to implement this method, allowing for parameters to be changed and results to be output seamlessly. However, this was only the case once the program syntax was correctly formulated for the first time. When first using the software, it can be very confusing and it requires a fairly thorough reading of the user's manual. One of the biggest problems with DIDO is its inability to provide useful error messages, which often leave the user with no information about the location of any bugs in the code. Another area in which DIDO could improve upon is the computation time for more accurate solutions. However, much of this can be attributed to running through MATLAB, which is notoriously slow when compared to other languages.

The results obtained were quite useful, offering much insight into the dynamics of a solar sail. By generating optimal solutions for a range of sail lightness numbers, the idea that this

parameter is a strong indicator of a sail's performance was confirmed. Additionally, some information about the relationship between lightness factor and the pseudospectral method was discovered, as shown by the moderate drop in solution optimality with increasing values of β .

5.2 Future Work Recommendations

Suggested future work includes the utilization of non-ideal sail models for more realistic results. These results can be compared to the results presented in this thesis to determine how accounting for different imperfections affects the dynamics of the sail and its trajectory, as well as the impact on time of flight. Some aspects to consider in a non-ideal model include:

- The ability of a sail to billow from the solar radiation pressure. The ideal model assumes a perfectly flat and rigid sail, which does not occur in reality. Sail materials will be ultralight and ultrathin and subject to some form of deflection.
- The fact that photons do not reflect perfectly off the surface of the sail. There are phenomena such as re-radiation and photon dispersion that occur in reality and decrease the performance of the sail.

The optimal solutions for sails with a lightness number higher than 1 could also be established. If the upcoming sail missions described in Section 4.3 prove that these lightness numbers are attainable, then these new optimal trajectories will most likely yield better results.

Finally, other destinations could be considered, preferably ones that are a considerable distance away. Results for farther destinations will provide a better idea of how beneficial solar sail technology actually is.

Appendix A: MATLAB Source – Perfectly Reflective

A.1 Problem File

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Problem (script) file for the Perfectly Reflective Planar Sail
%
% Alex Pini
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%=====
% Problem variables:
%-----
% states = (rho, rhodot, theta, thetadot, z, zdot)
% controls = alpha, gamma
%=====

%-----
% bounds the state and control variables
%-----

rhoL = 0.6; rhoU = 1.2;
rhodotL = -0.5; rhodotU = 0.5;
thetaL = 0; thetaU = 3*pi;
thetadotL = pi; thetadotU = 4*pi;
zL = -0.2; zU = 0.2;
zdotL = -0.2; zdotU = 0.2;

bounds.lower.states = [rhoL; rhodotL; thetaL; thetadotL; zL; zdotL];
bounds.upper.states = [rhoU; rhodotU; thetaU; thetadotU; zU; zdotU];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Fixed limits on controls (sin and cos of the control angles)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bounds.lower.controls = [0; -1; 0; -1];
bounds.upper.controls = [1; 1; 1; 1];

%-----
% bound the horizon
%-----

t0 = 0;
tfMax = 1;
bounds.lower.time = [t0, t0];
bounds.upper.time = [t0, tfMax]; % Fixed time at t0 and a possibly free time
at tf
```

```

%-----
% Set up the bounds on the endpoint function
%-----

% See events file for definition of events function

mu = 4 * pi^2;

% Initial conditions for state variables

rho0 = 1;
rhodot0 = 0;
theta0 = 0;
thetadot0 = sqrt(mu/rho0^3);
z0 = 0;
zdot0 = 0;

bounds.lower.events = [rho0; rhodot0; theta0; thetadot0; z0; zdot0; 0;0;0];
bounds.upper.events = bounds.lower.events; % equality event function bounds

%% made path constraints into event constraints and commented out the path
%% constraint info in main problem file

bounds.lower.path = [0; 0];
bounds.upper.path = [0; 0];

%=====
% Define the problem using DIDO expressions:
%=====

IdealPlanar.cost = 'IdealPlanarCost';
IdealPlanar.dynamics = 'IdealPlanarDynamics';
IdealPlanar.events = 'IdealPlanarEvents';
IdealPlanar.path = 'IdealPlanarPath';
IdealPlanar.bounds = bounds;

%=====

% % Enable this section for first run to get a feasible solution
%
% ca = cos(-pi/4);
% sa = sin(-pi/4);
% cg = cos(0);
% sg = sin(0);
%
% % insert guesses for beginning, middle, and end times
%
% algorithm.guess.controls = [ca ca ca; sa sa sa; cg cg cg; sg sg sg];
%
% algorithm.guess.states = [rho0, 0.85 , 0.7233; 0 -0.2 0; 0 thetaU/2 thetaU;
%   thetadot0 (thetadot0+sqrt(mu/(0.7233^3)))/2 sqrt(mu/(0.7233^3));
%   0 0 0.01; 0 -0.01 0];
%
% algorithm.guess.time = [0 tfMax/2 tfMax];
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

algorithm.nodes = [21];
load 'ideal_planar_21_nodes_015' primal;
algorithm.guess = primal
% algorithm.mode = 'accurate' %enable for accurate solution
% Call dido
tStart= cputime; % start CPU clock
[cost, primal, dual] = dido(IdealPlanar, algorithm);
runTime = cputime-tStart
save 'ideal_planar_21_nodes_015'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OUTPUT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Calling values for state variables from primal structure

rho = primal.states(1,:);
rhodot = primal.states(2,:);
theta = primal.states(3,:);
thetadot = primal.states(4,:);
z = primal.states(5,:);
zdot = primal.states(6,:);
t = primal.nodes;

%Calling final values for state variables from primal structure

rhof = primal.states(1,end);
rhodotf = primal.states(2,end);
thetaf = primal.states(3,end);
thetadotf = primal.states(4,end);
z = primal.states(5,end);
zdotf = primal.states(6,end);
tf = primal.nodes(1,end)

%Converting position components in state vector to Cartesian coordinates
%for plotting purposes

[x,y,z] = pol2cart(theta(1,:), rho(1,:), z(1,:));

%Calculating the longitude of the ascending node for the optimal trajectory
%from the final values for the state variables

cLAN = (((rhof * zdotf - rhodotf * z) * cos(thetaf) + rhof * z * thetadotf
* sin(thetaf))/(sqrt((rhof * zdotf - rhodotf * z)^2 + (rhof)^2 * (z)^2 *
(thetadotf)^2)));

sLAN = (((rhof * zdotf - rhodotf * z) * sin(thetaf) - rhof * z * thetadotf
* cos(thetaf))/(sqrt((rhof * zdotf - rhodotf * z)^2 + (rhof)^2 * (z)^2 *
(thetadotf)^2)));

LAN = atan2(sLAN,cLAN);

```

```

%Plotting Earth and Venus Orbits

[ex,ey,ez] = cylinder(1,200);
[vx,vy,vz] = cylinder(0.723332,200);

i_v = 3.39471 * pi/180;

vx2 = vx .* cos(i_v);
vy2 = vy;
vz2 = -vx .* sin(i_v);

C_1 = [1, 0, 0; 0, cos(-i_v), sin(-i_v); 0, -sin(-i_v), cos(-i_v)];
C_3 = [cos(-LAN), sin(-LAN), 0; -sin(-LAN), cos(-LAN), 0; 0, 0, 1];

n = 1;

while n < 210

    node = (2 * pi /200) * n;

    V(1,n) = 0.723332 * cos(node);
    V(2,n) = 0.723332 * sin(node);
    V(3,n) = 0;

    n = n + 1;

end

V_3 = C_3 * C_1 * V;

%Outputting relevant figures

figure
plot3(x(1,:),y(1,:),z(1:,:), 'g-
o', ex(1,:),ey(1,:),ez(1:,:), 'b', V_3(1,:),V_3(2,:),V_3(3:,:), 'r')
grid on;
xlabel('x (A.U.)')
ylabel('y (A.U.)')
zlabel('z (A.U.)')
title('Optimal Sail Trajectory (\beta = 0.6)')
legend('Sail Trajectory', 'Earth Orbit', 'Venus Orbit')
figure
plot(primal.nodes,dual.Hamiltonian)
ylabel('H')
xlabel('Time (Years)')
title('Hamiltonian vs. Time')
figure
calpha = primal.controls(1,:);
salpha = primal.controls(2,:);
cgamma = primal.controls(3,:);
sgamma = primal.controls(4,:);
plot(primal.nodes,atan2(salpha,calpha))

```

```

title ('Control Angle \alpha vs. Time')
ylabel('\alpha (rad)')
xlabel('Time (Years)')
figure
plot(primal.nodes, atan2(sgamma,cgamma))
title ('Control Angle \gamma vs. Time')
ylabel('\gamma (rad)')
xlabel('Time (Years)')

```

A.2 Dynamics File

```

function XDOT = IdealPlanarDynamics(primal)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dynamics for the Perfectly Reflective Planar Sail
%
% Alex Pini
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mu = 4*pi^2;
SLF = 0.15;

XDOT = zeros(6,1);
rho = primal.states(1,:);
rhodot = primal.states(2,:);
theta = primal.states(3,:);
thetadot = primal.states(4,:);
z = primal.states(5,:);
zdot = primal.states(6,:);

calpha = primal.controls(1,:);
salpha = primal.controls(2,:);
cgamma = primal.controls(3,:);
sgamma = primal.controls(4,:);

%=====
% Equations of Motion:
%=====

r = sqrt(rho.^2 + z.^2);
rhatnsq = (( rho .* cgamma .* calpha + z .* sgamma)./r).^2;

xldot = rhodot;

x2dot = rho .* thetadot.^2 - (mu * rho)./(r.^3) + SLF .* mu./(r.^2) .*
rhatnsq .* cgamma .* calpha;

x3dot = thetadot;
x4dot = (1./rho) .* (-2 * rhodot .* thetadot + SLF .* mu./(r.^2) .*
rhatnsq .* salpha);

```

```

x5dot = zdot;

x6dot = -1 * (mu * z)./(r.^3) + SLF .* mu./(r.^2) .* rhatnsq .* sgamma;

%=====
XDOT = [x1dot; x2dot; x3dot; x4dot; x5dot; x6dot];
%=====

```

A.3 Events File

```

function endpointFunction = IdealPlanarEvents(primal)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Endpoint function for the Perfectly Reflective Planar Sail
%
% Alex Pini
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

rho0 = primal.states(1,1);      rhof = primal.states(1,end);
rhodot0 = primal.states(2,1);   rhodotf = primal.states(2,end);
theta0 = primal.states(3,1);    thetaf = primal.states(3,end);
thetadot0 = primal.states(4,1); thetadotf = primal.states(4,end);
z0 = primal.states(5,1);        zf = primal.states(5,end);
zdot0 = primal.states(6,1);     zdotf = primal.states(6,end);

% preallocate the endpointFunction evaluation for good MATLAB computing

endpointFunction = zeros(9,1); % t0 is specified in the problem file

%=====

mu = 4 * pi^2;

%Defining orbital properties of Venus so final states of optimal
%trajectory can be constrained

r_v = 0.7233;
i_v = 3.39471 * pi/180;
LAN_v = 45*pi/180;

endpointFunction(1) = rho0;
endpointFunction(2) = rhodot0;
endpointFunction(3) = theta0;
endpointFunction(4) = thetadot0;
endpointFunction(5) = z0;
endpointFunction(6) = zdot0;

%Equations constraining final states of optimal trajectory
%to correlate with those of Venus's orbit

```

```

endpointFunction(7) = (rhof)^2 + (zf)^2 - (r_v)^2;

endpointFunction(8) = (rhodotf)^2 + (rhof*thetadotf)^2 + (zdotf)^2 -
mu/(r_v);

endpointFunction(9) = (((rhof)^2 * thetadotf)/(sqrt((rhof)^2 * (zf)^2 *
(thetadotf)^2 + (rhodotf*zf - rhof*zdotf)^2 + (rhof)^4 * (thetadotf)^2)))-
cos(i_v);

%% endpointFunction(10) = (((rhof * zdotf - rhodotf * zf) * cos(thetaf) +
rhof * zf * thetadotf * sin(thetaf))/(sqrt((rhof * zdotf - rhodotf * zf)^2 +
(rhof)^2 * (zf)^2 * (thetadotf)^2)))-cos(LAN_v);

%% endpointFunction(11) = (((rhof * zdotf - rhodotf * zf) * sin(thetaf) -
rhof * zf * thetadotf * cos(thetaf))/(sqrt((rhof * zdotf - rhodotf * zf)^2 +
(rhof)^2 * (zf)^2 * (thetadotf)^2)))-sin(LAN_v);
%-----

```

A.4 Path File

```

function Constraints = IdealPlanarPath(primal)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Endpoint function for the Perfectly Reflective Planar Sail
%
% Alex Pini
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

calpha = primal.controls(1,:);
salpha = primal.controls(2,:);
cgamma = primal.controls(3,:);
sgamma = primal.controls(4,:);

%Constraining the trigonometric components of each control
%angle at each node so that the control angles
%are meaningful

conalpha = calpha.^2 + salpha.^2 - 1;
congamma = cgamma.^2 + sgamma.^2 - 1;

Constraints = [conalpha; congamma];

```

A.5 Cost File

```
function [EndpointCost, RunningCost] = IdealPlanarCost(primal)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Endpoint Cost for the Perfectly Reflective Planar Sail
%
% Alex Pini
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tf = primal.nodes(end);
EndpointCost = tf;
RunningCost = 0;
```


Appendix B: Variation of Optimal Results with SLN

In the following pages, the optimal results generated from DIDO are presented. The results consist of the shape of the trajectory, the control history, and the Hamiltonian behavior over time. With all other parameters held constant, the sail lightness number is incremented by 0.05 from 0.1-1.0.

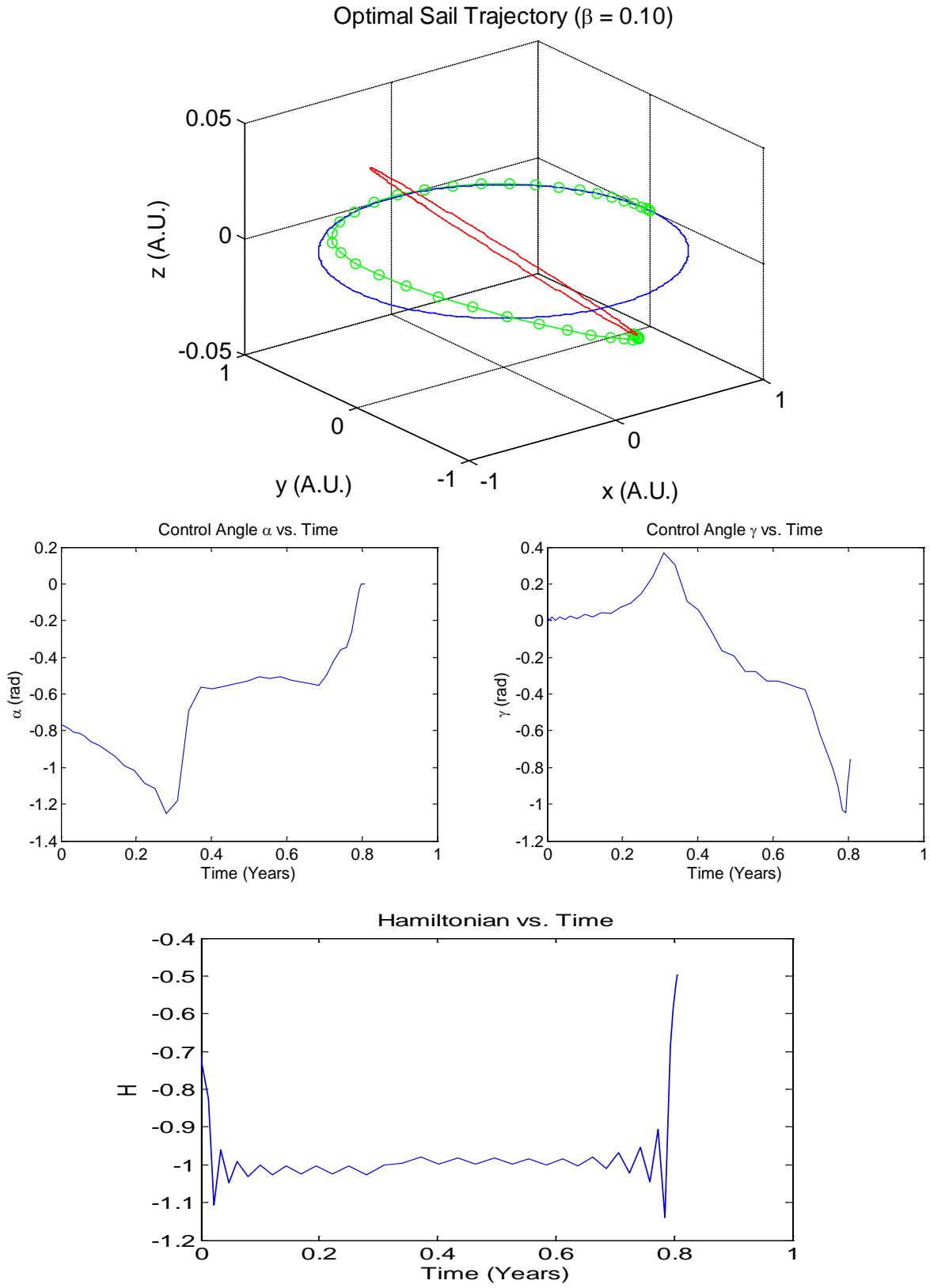


Figure 11: Optimal Solution when SLF = 0.10

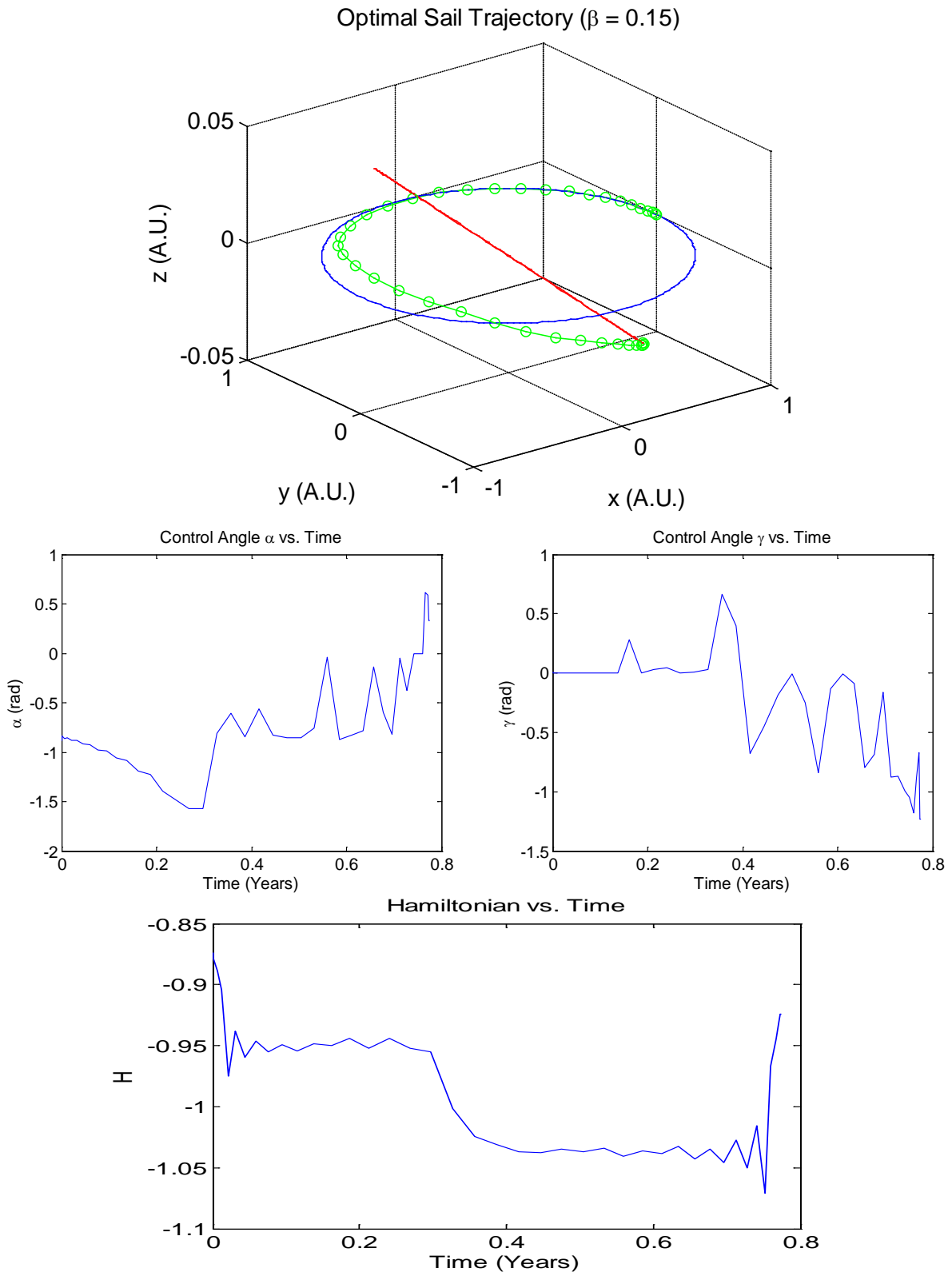


Figure 12: Optimal Solution when SLF = 0.15

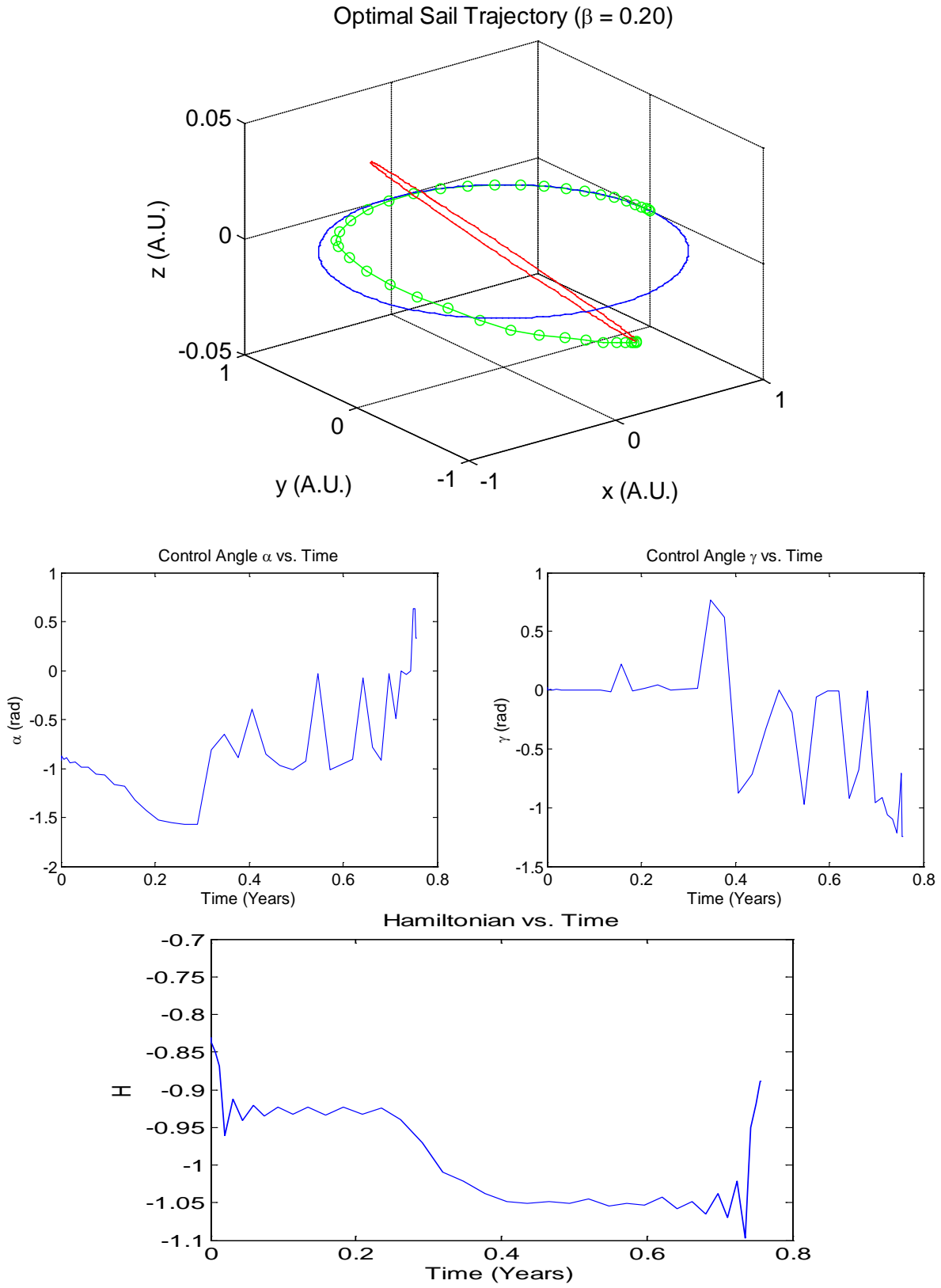


Figure 13: Optimal Solution when SLF = 0.20

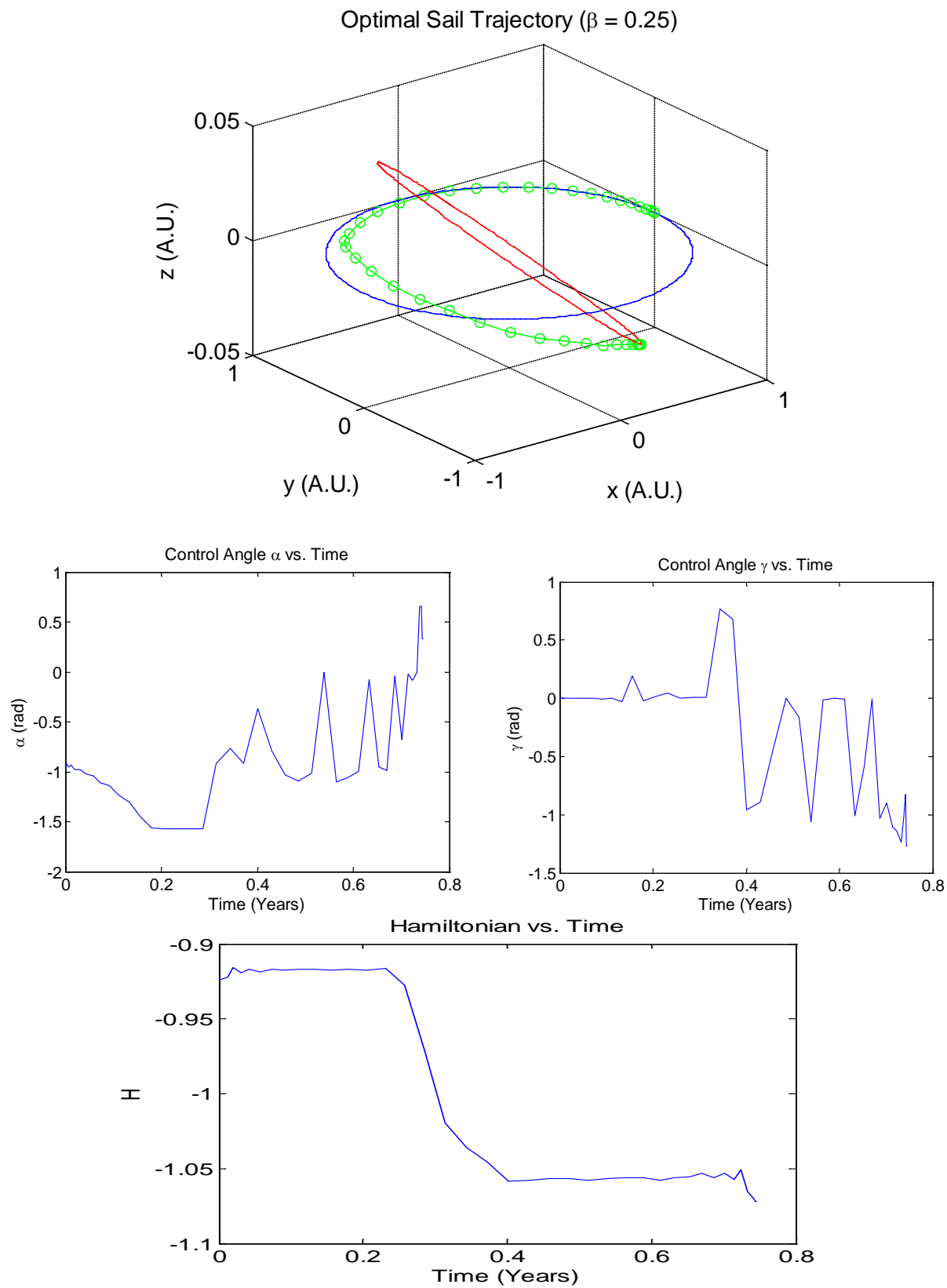


Figure 14: Optimal Solution when SLF = 0.25

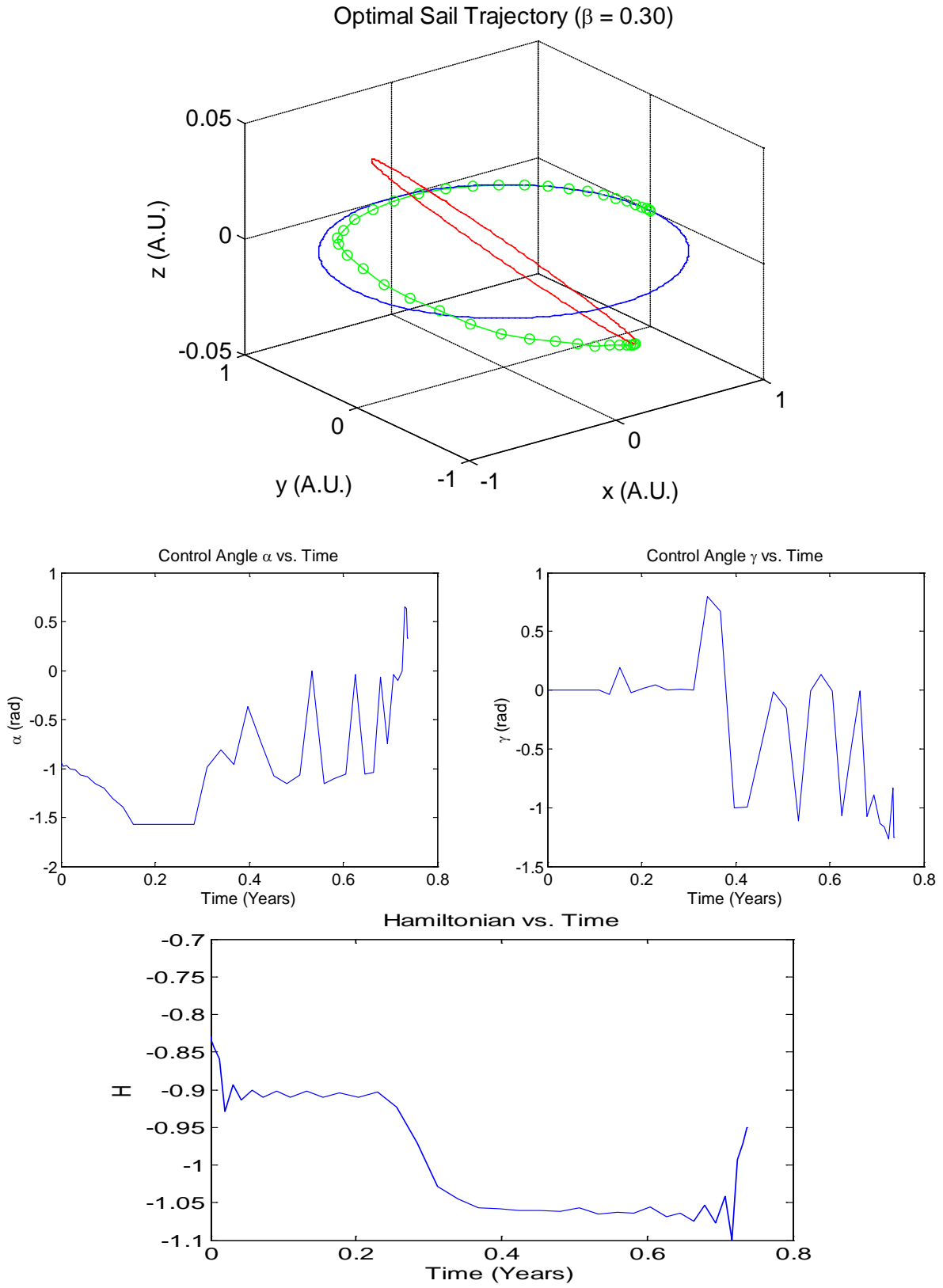


Figure 15: Optimal Solution when SLF = 0.30

Optimal Sail Trajectory ($\beta = 0.35$)

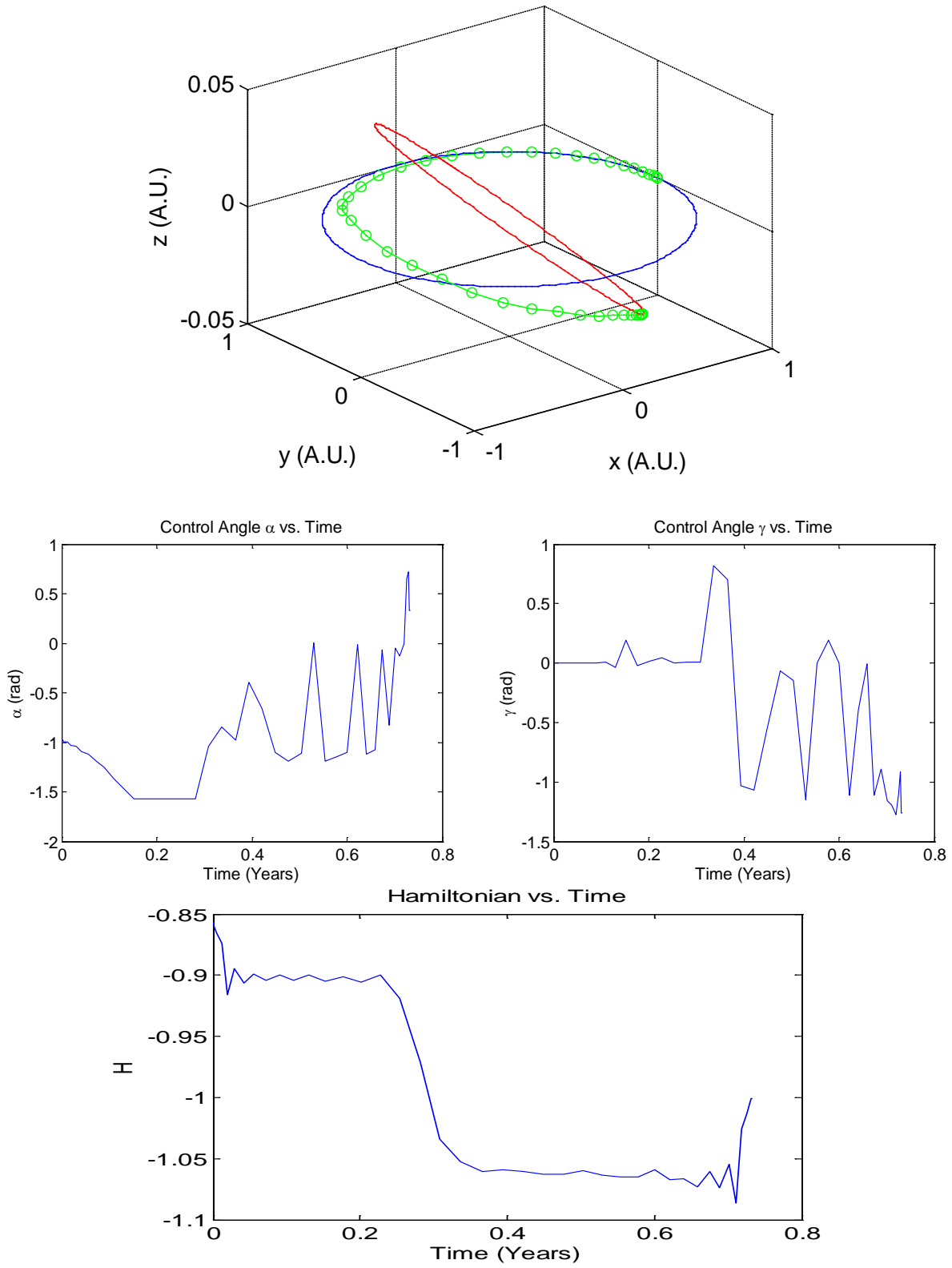


Figure 16: Optimal Solution when SLF = 0.35

Optimal Sail Trajectory ($\beta = 0.40$)

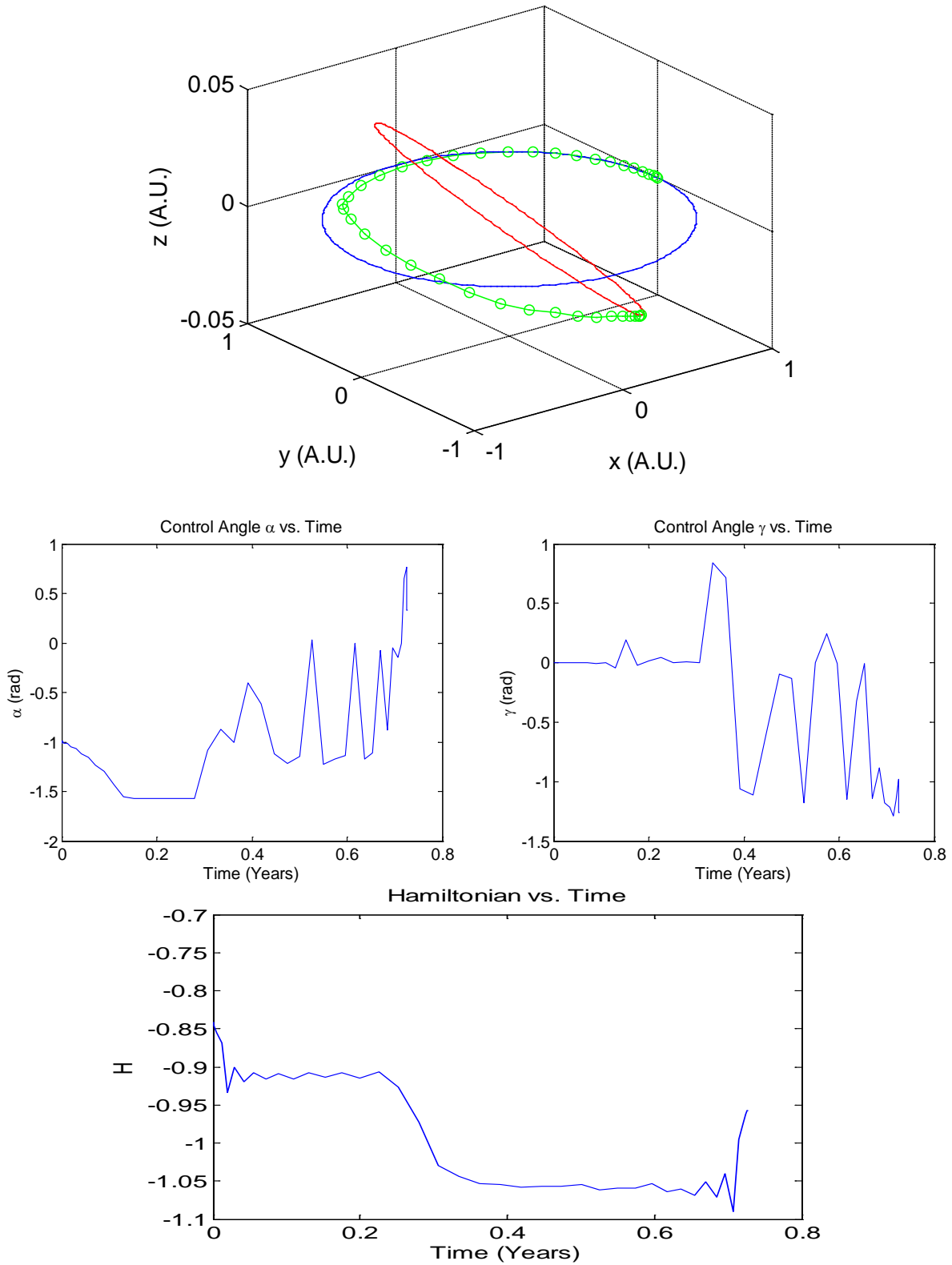


Figure 17: Optimal Solution when SLF = 0.40

Optimal Sail Trajectory ($\beta = 0.45$)

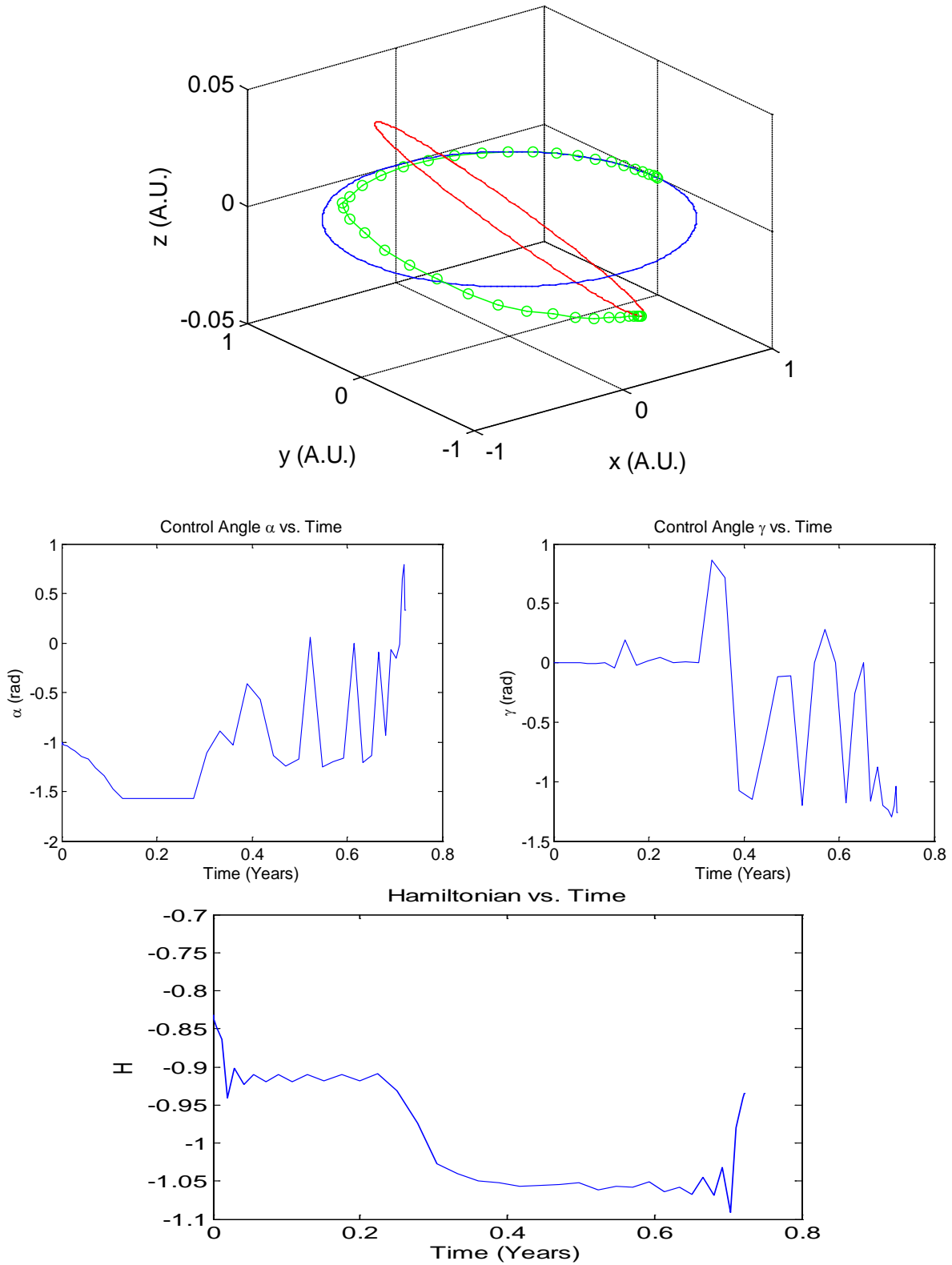


Figure 18: Optimal Solution when SLF = 0.45

Optimal Sail Trajectory ($\beta = 0.50$)

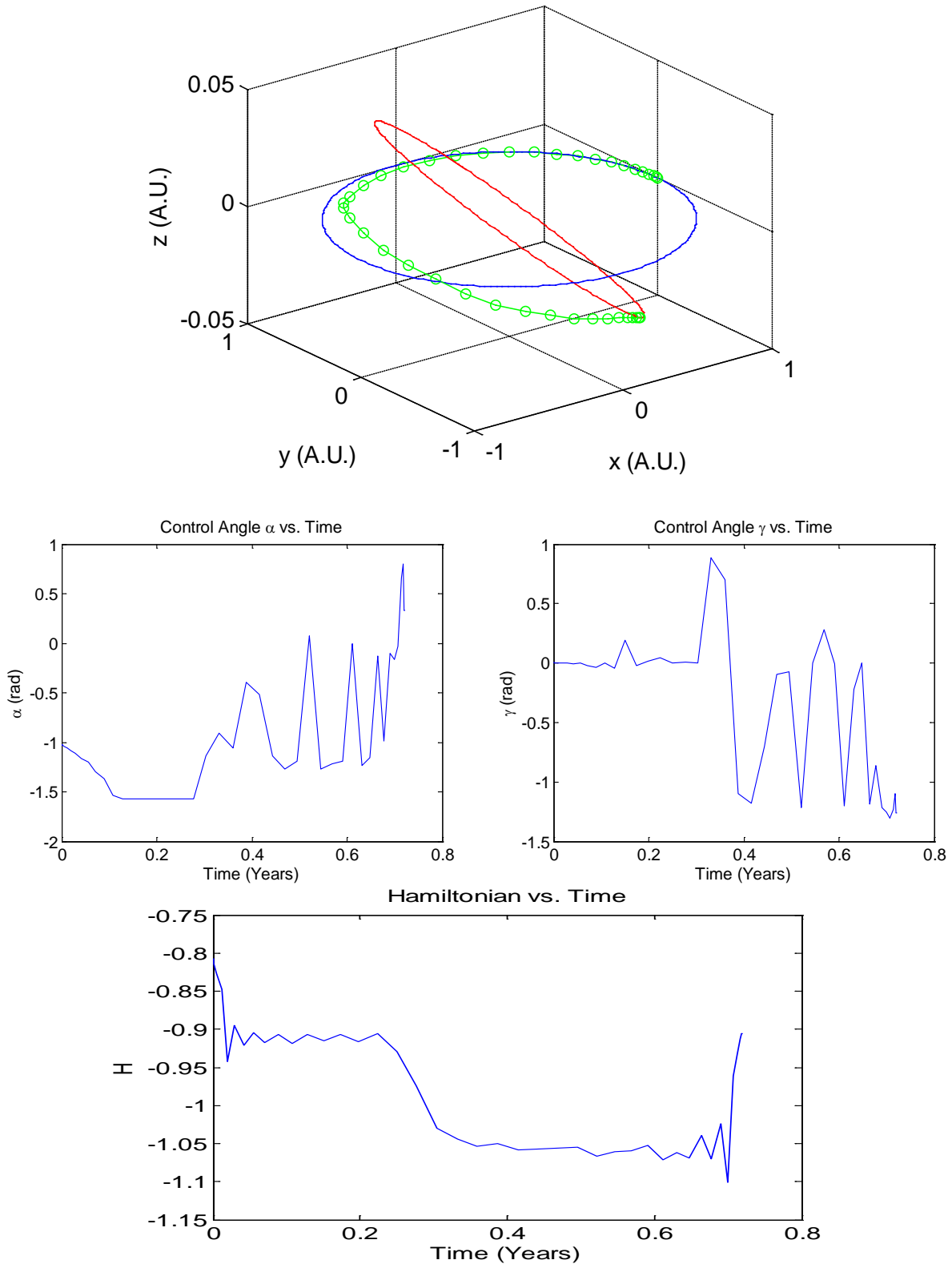


Figure 19: Optimal Solution when SLF = 0.50

Optimal Sail Trajectory ($\beta = 0.55$)

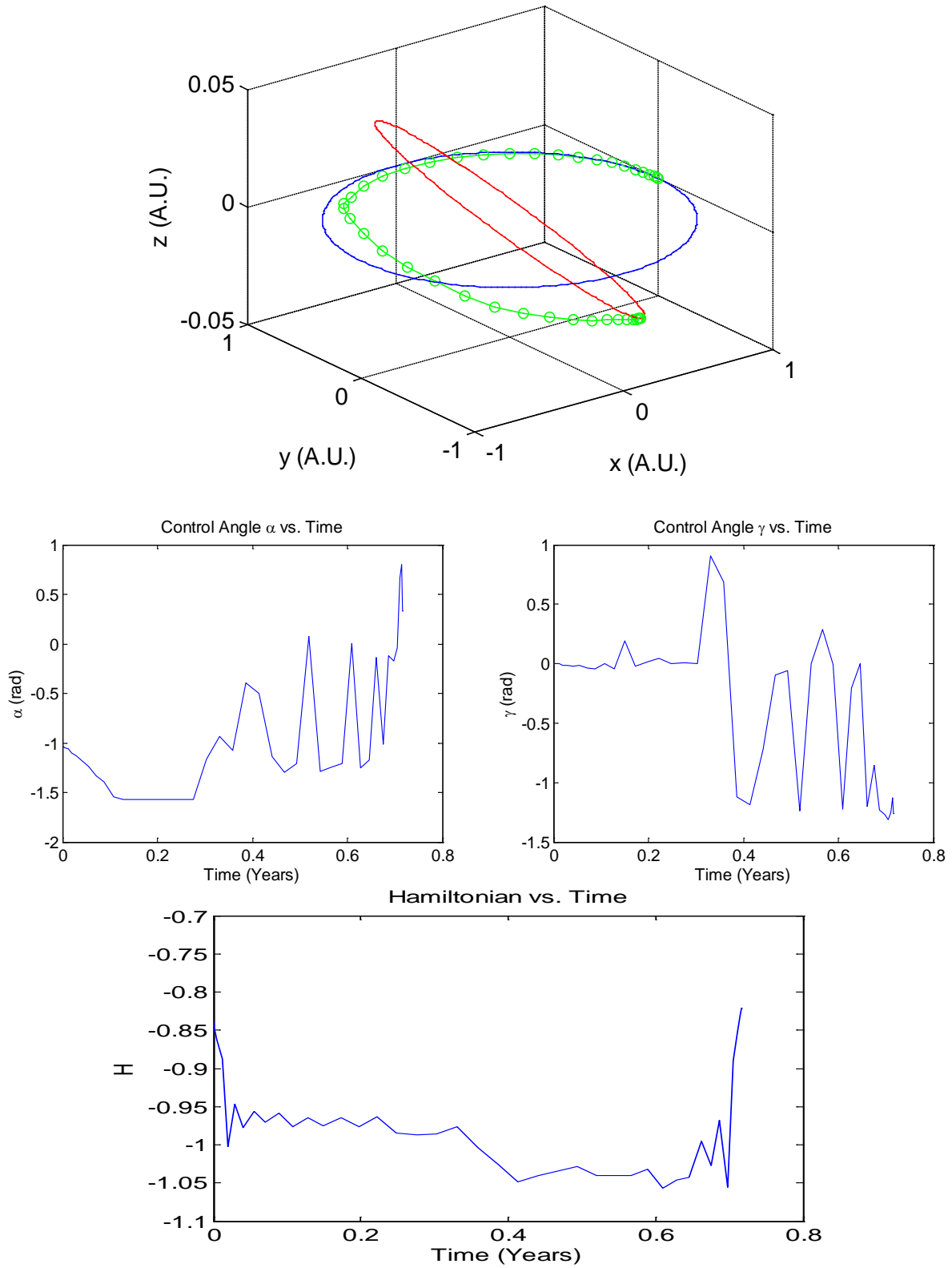


Figure 20: Optimal Solution when SLF = 0.55

Optimal Sail Trajectory ($\beta = 0.60$)

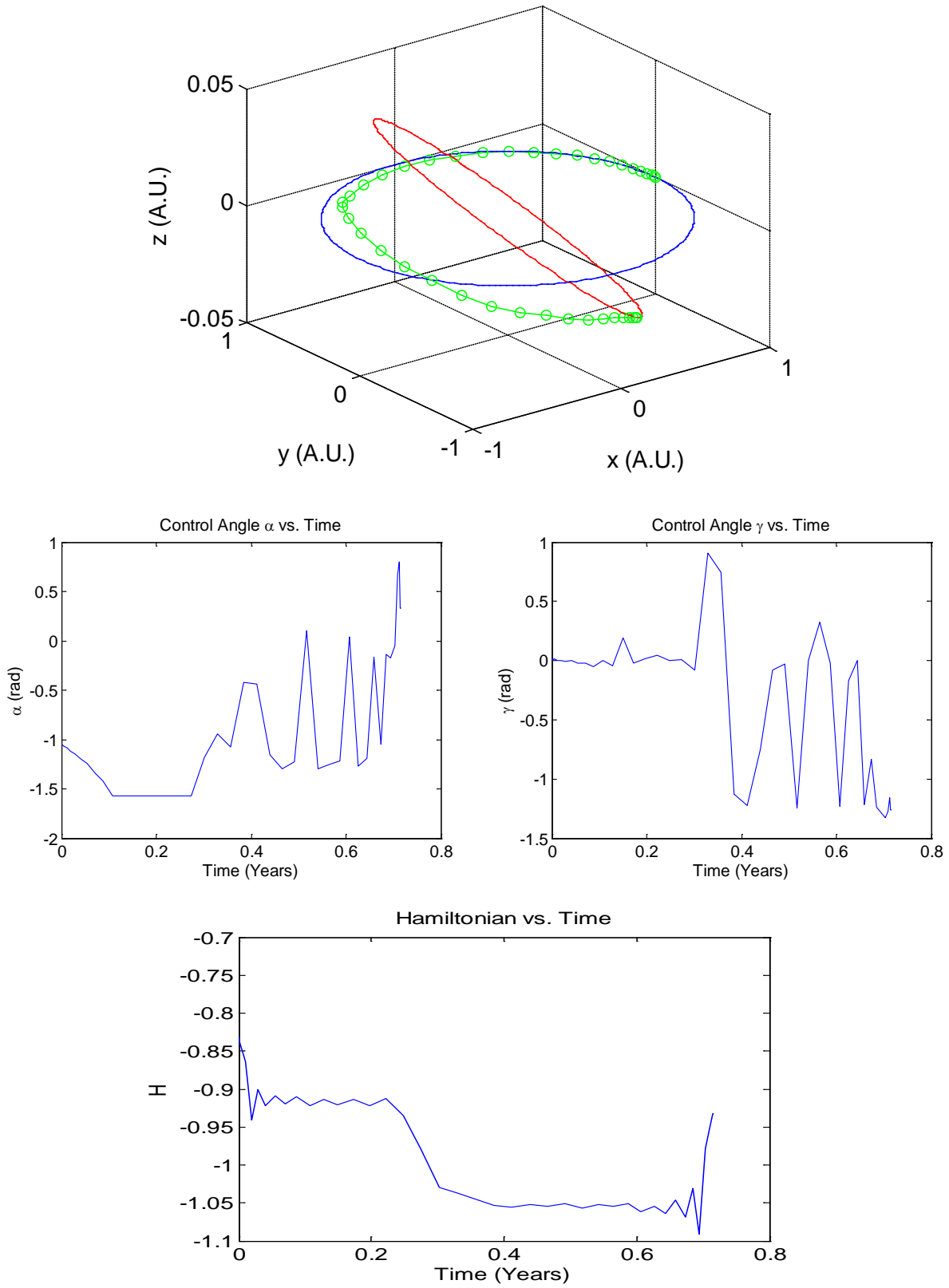


Figure 21: Optimal Solution when SLF = 0.60

Optimal Sail Trajectory ($\beta = 0.65$)

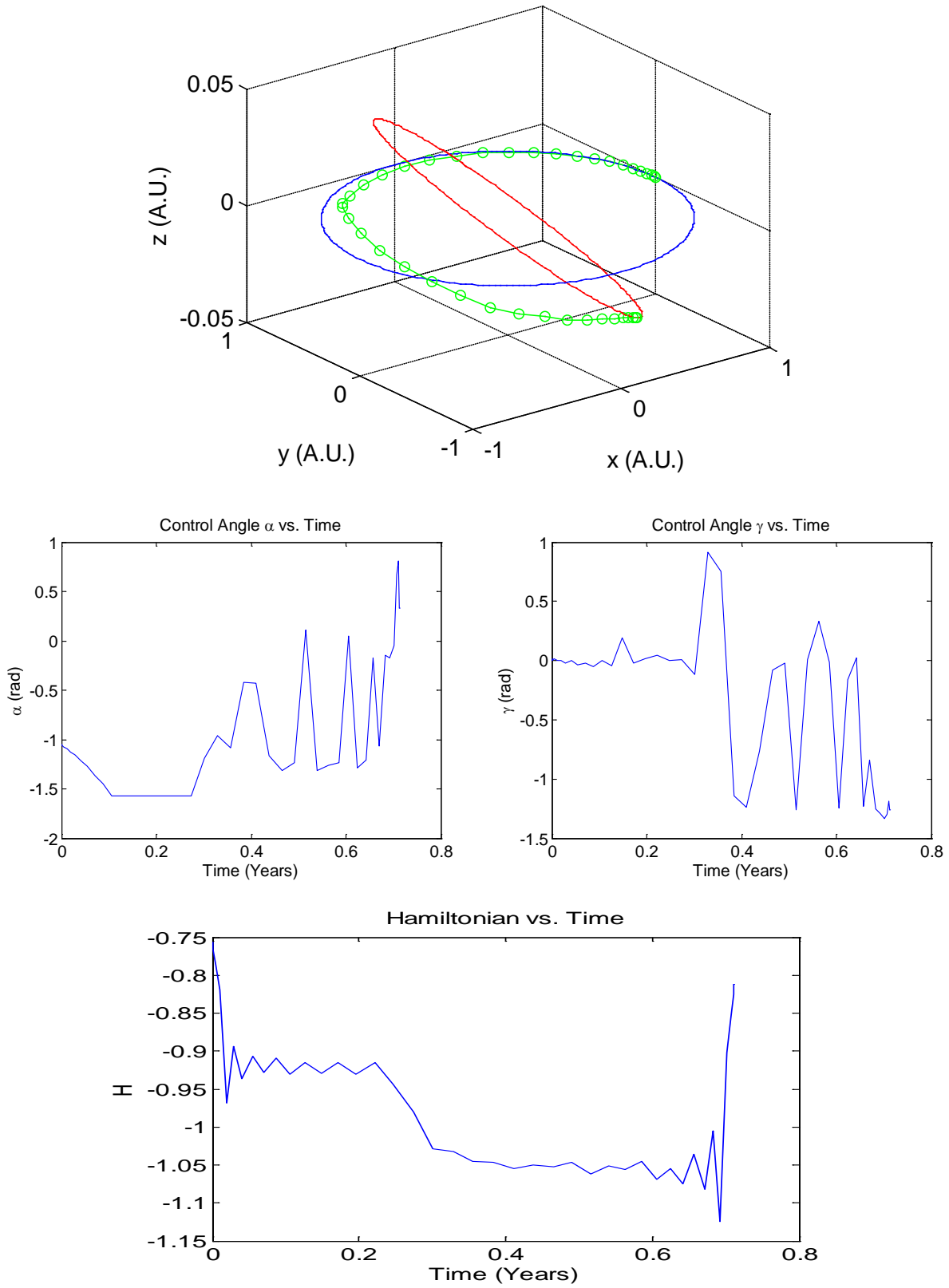


Figure 22: Optimal Solution when SLF = 0.65

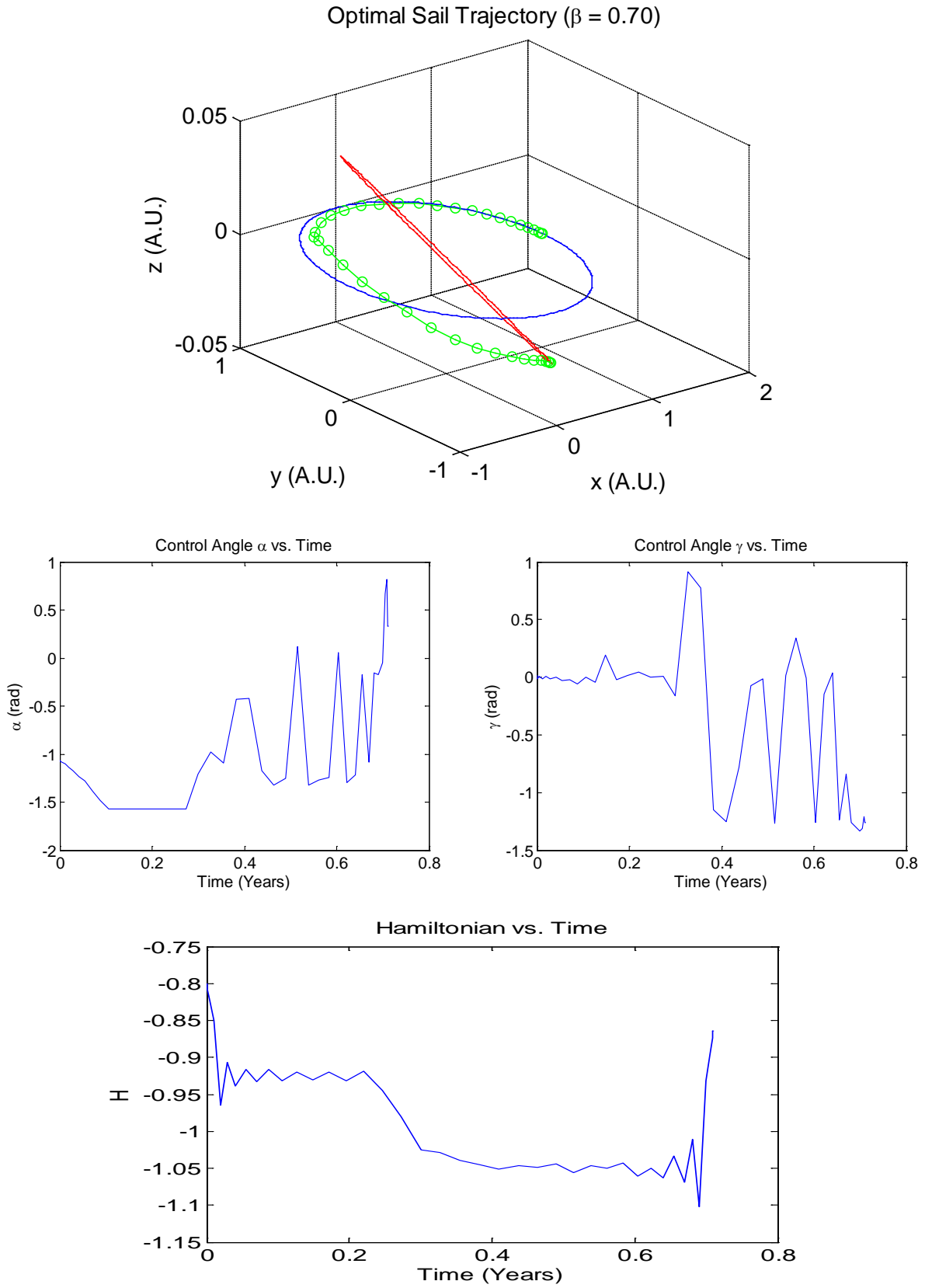


Figure 23: Optimal Solution when SLF = 0.70

Optimal Sail Trajectory ($\beta = 0.75$)

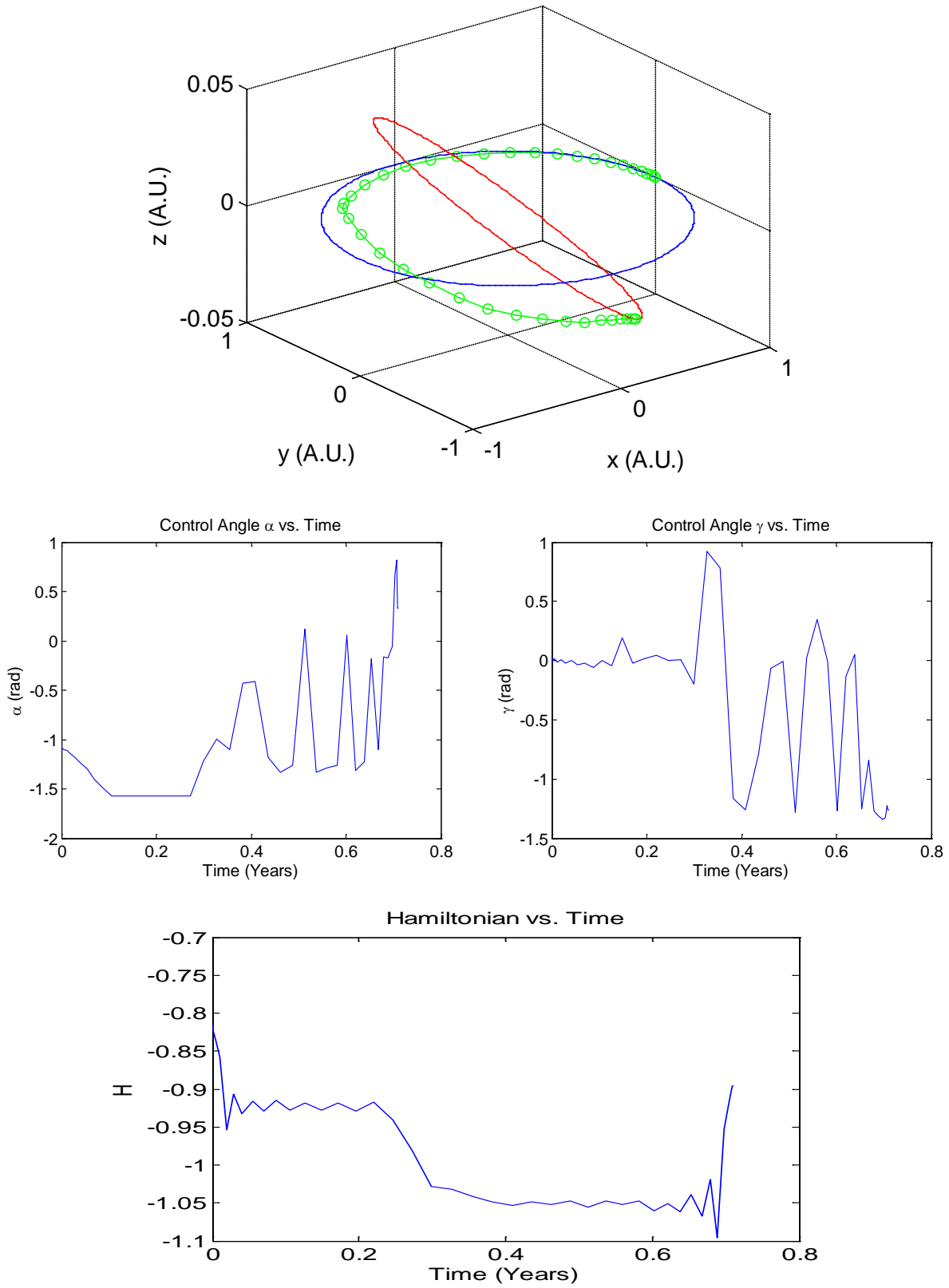


Figure 24: Optimal Solution when SLF = 0.75

Optimal Sail Trajectory ($\beta = 0.80$)

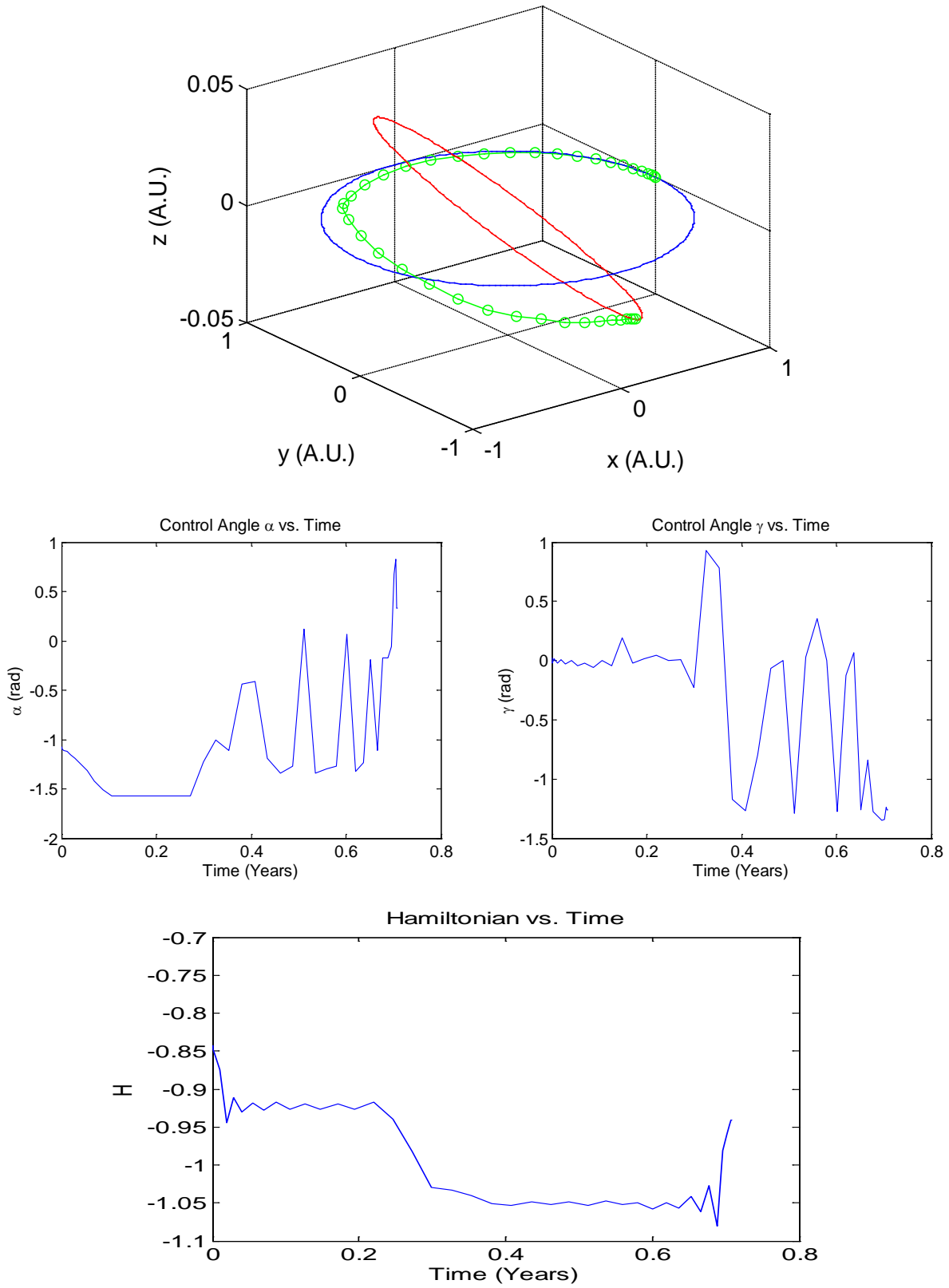


Figure 25: Optimal Solution when SLF = 0.80

Optimal Sail Trajectory ($\beta = 0.85$)

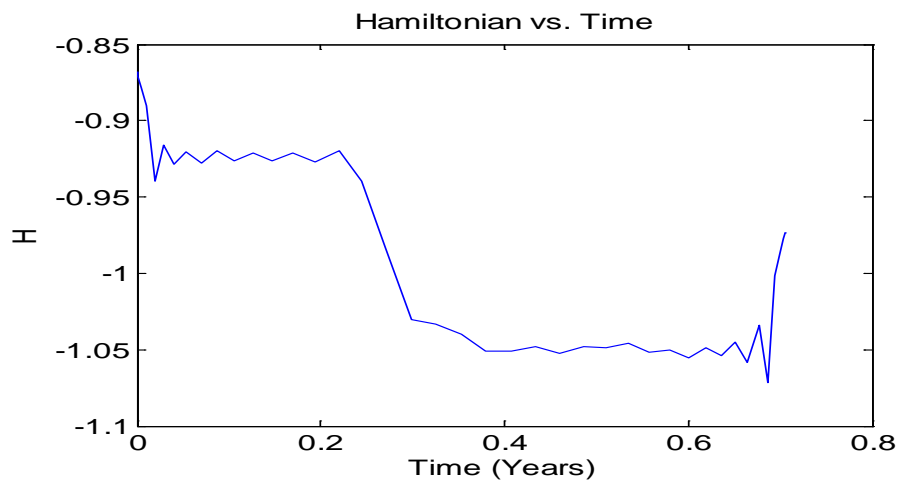
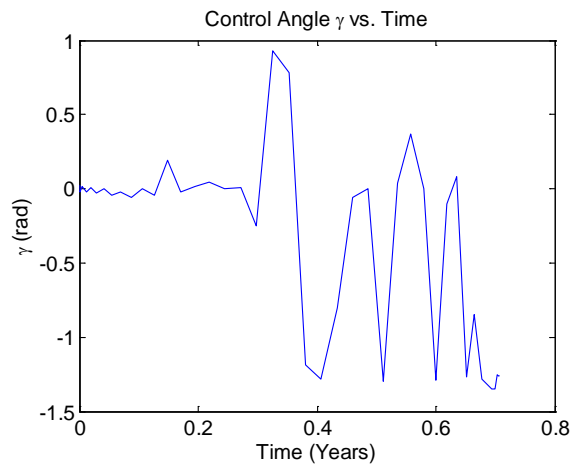
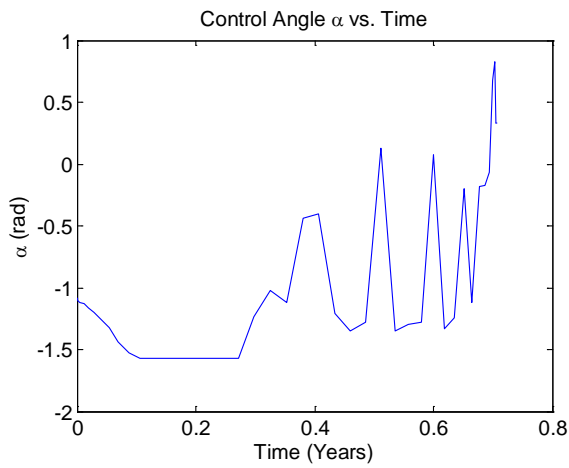
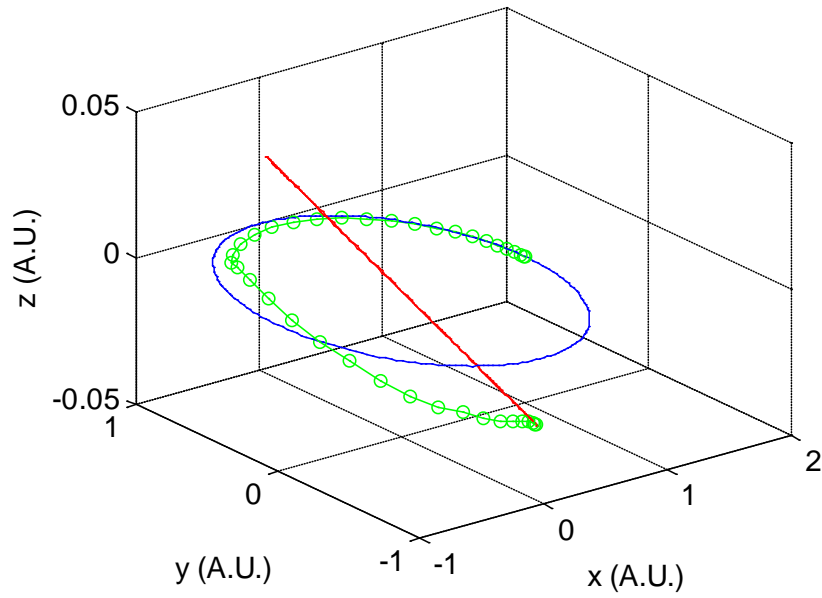


Figure 26: Optimal Solution when SLF = 0.85

Optimal Sail Trajectory ($\beta = 0.90$)

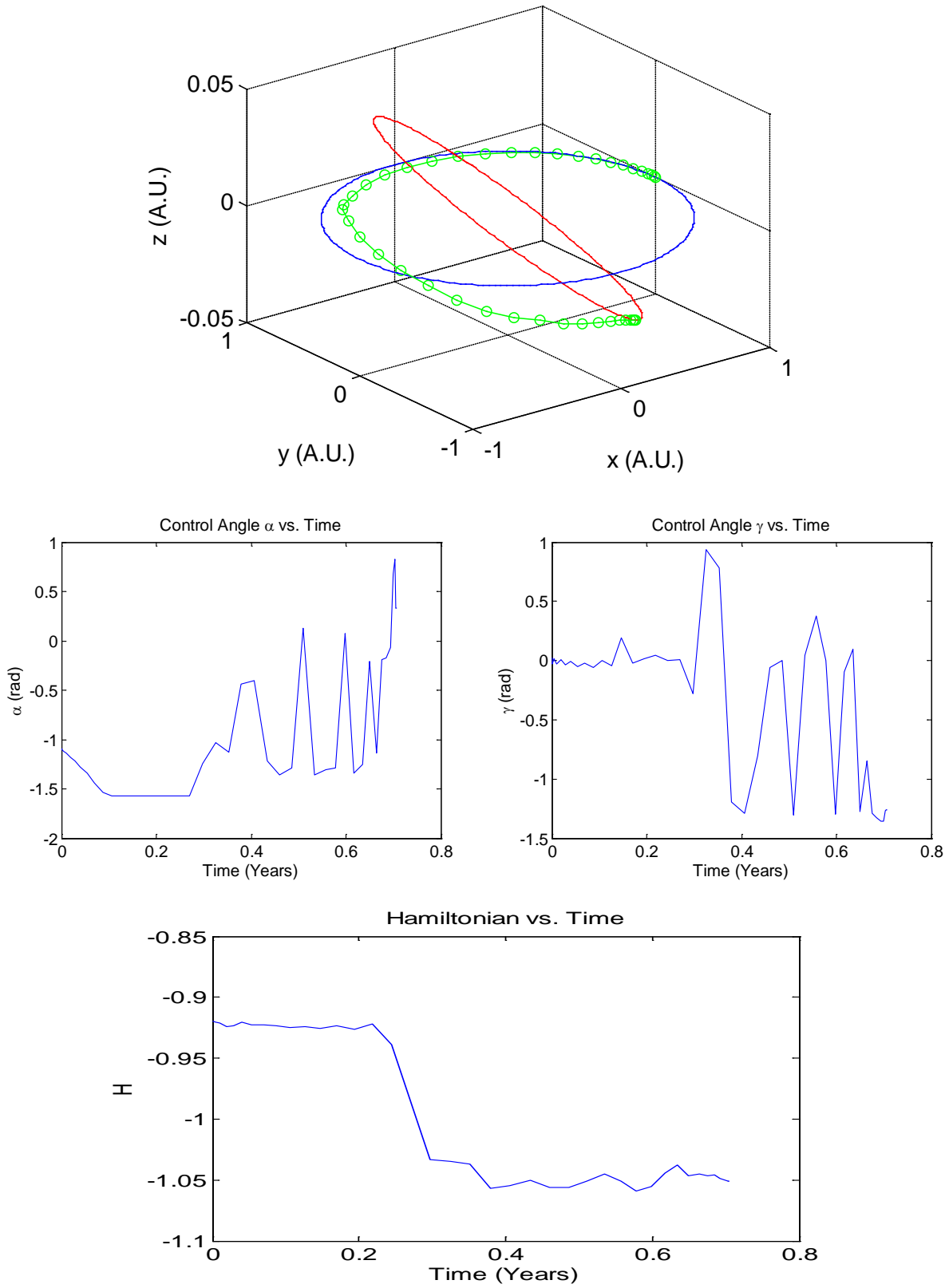


Figure 27: Optimal Solution when SLF = 0.90

Optimal Sail Trajectory ($\beta = 0.95$)

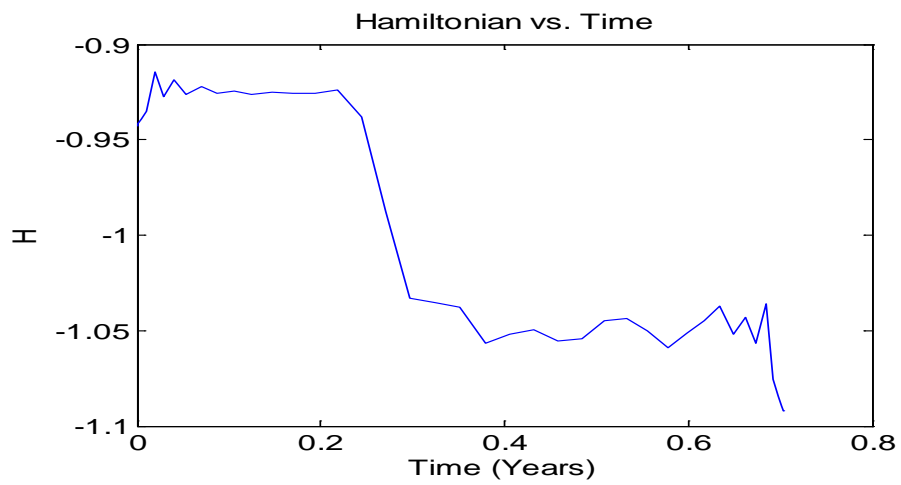
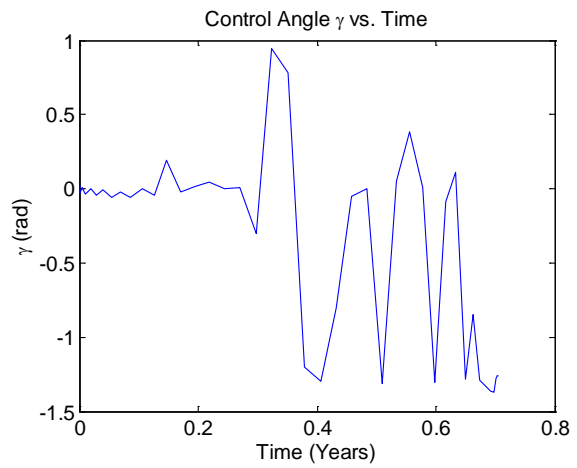
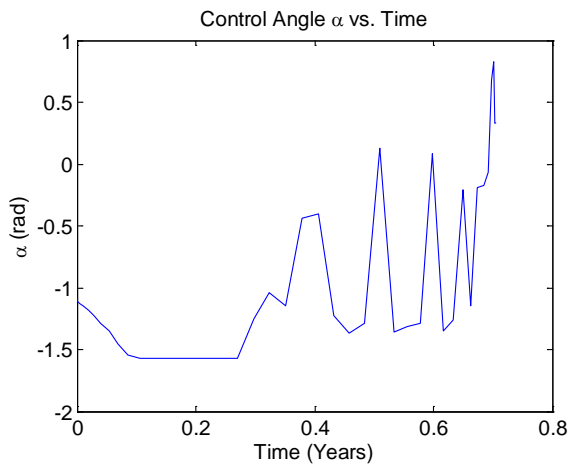
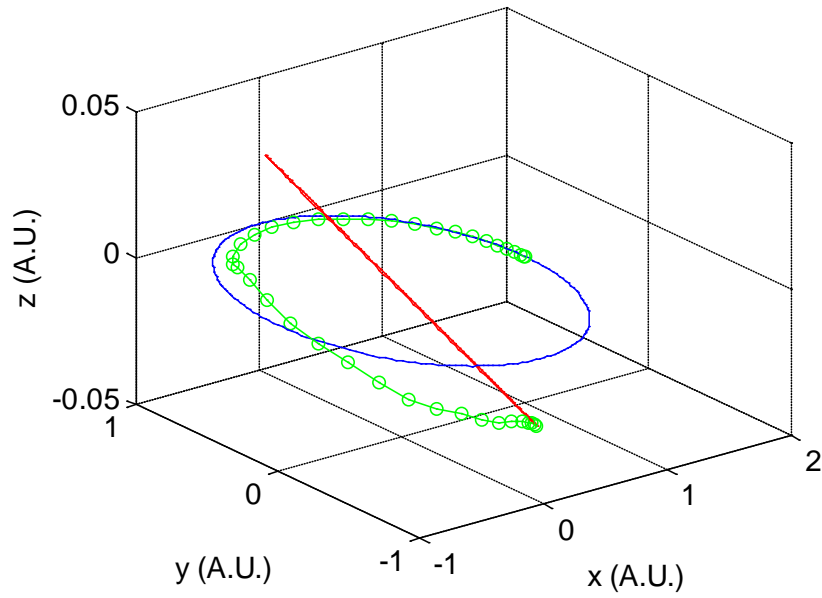


Figure 28: Optimal Solution when SLF = 0.95

Optimal Sail Trajectory ($\beta = 1.00$)

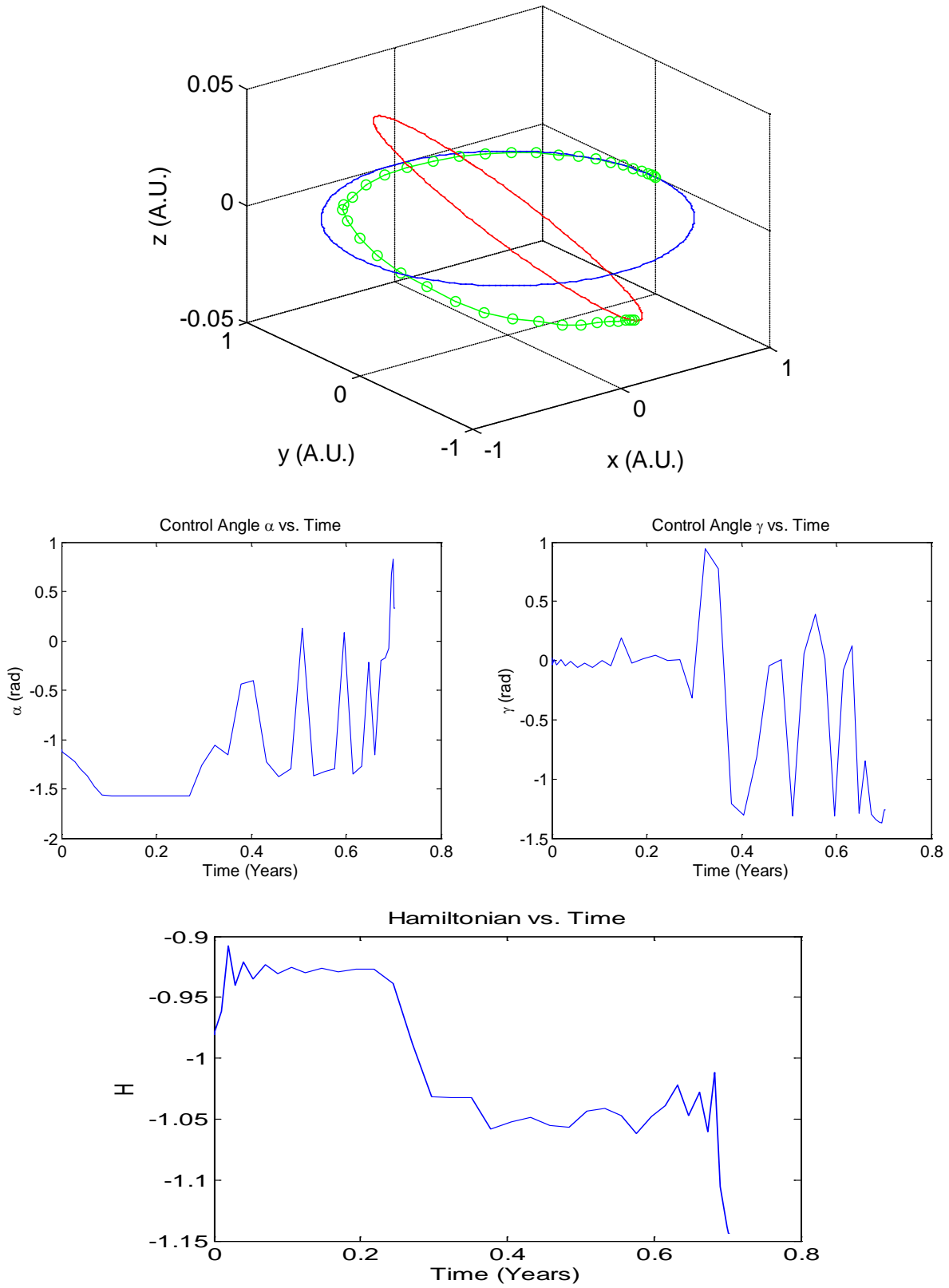


Figure 29: Optimal Solution when SLF = 1.00

References

- [1] McInnes, C. (1999) *Solar Sailing: Technology, Dynamics, and Mission Applications*. Praxis.
- [2] Curtis, H. (2005) *Orbital Mechanics for Engineering Students*. Elsevier.
- [3] Melton, R. (2002) "Comparison of Direct Optimization Methods Applied to Solar Sail Problems," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. August 2002. AIAA 2002-4728.
- [4] Ross, I. *A Beginner's Guide to DIDO: A MATLAB Application Package for Solving Optimal Control Problems*.
- [5] Bate, R.,D. Mueller, and J. White (1971) *Fundamentals of Astrodynamics*. Dover.
- [6] Enright, P. and B. Conway (1991) "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, **14**(5), pp.981-985.
- [7] Hargraves, C. and S. Paris (1987) "Direct Trajectory Optimization Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, **10**(4), pp. 338-342.
- [8] Kirk, D. (1970) *Optimal Control Theory: An Introduction*, Prentice-Hall.
- [9] Murphy, D. and B. Wie (2004) "Robust Thrust Control Authority for a Scalable Sailcraft," *14th AAS/AIAA Space Flight Mechanics Conference*. February 2004. AAS 04-285.
- [10] Garner, C., B. Diedrich, and M. Leipold (1999) "A Summary of Solar Sail Technology Developments and Proposed Demonstration Missions," *Jet Propulsion Laboratory Report*. JPC-99-2697.
- [11] McInnes, C. (1991) "Solar Sail Mission Applications for Non-Keplerian Orbits," *Acta Astronautica*, **45**(4-9), pp.567-575.

Campus Address:

121 W. Fairmount Ave. #10
State College, PA 16801
Mobile: (724) 518-3074

Academic Vita

ajp5095@psu.edu

Permanent Address:

103 Crabtree Ct.
Venetia, PA 15367
Home: (724) 941-2394

EDUCATION**Bachelor of Science, Aerospace Engineering****Spring 2010**

Schreyer Honors College at The Pennsylvania State University

- Honors: Aerospace Engineering
- Thesis: *Trajectory Optimization for an Ideal Solar Sail Model Using Pseudospectral Optimal Control*
- Thesis Supervisor: Dr. Robert G. Melton

EMPLOYMENT/EXPERIENCE**NASA Sounding Rocket Operations Contract**

Environmental Testing Lab Technician

- Flight qualified sounding rocket payloads by subjecting them to a battery of tests
- Supervisor: Glenn Maxfield
- January, 2008 – August, 2008
- Wallops Island, VA

HONORS/AWARDS

- Graduate Research Assistant position at Center for Space Research in Orbital Mechanics group, University of Texas-Austin
- Dean's List all semesters
- National Honors Society
- Tau Beta Pi Engineering Honor Society (invited) – Top 1/5 of senior engineering class
- Sigma Gamma Tau Aerospace Engineering Honor Society (invited) – Top 1/3 of senior aerospace class