Three-Dimensional Trajectory Optimization in Constrained Airspace

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Ran Dai

Certificate of Approval:

David Cicci
Professor
Aerospace Engineering

John E. Cochran Jr. Chair
Professor
Aerospace Engineering

Roy Hartfield
Professor
Aerospace Engineering

Anwar Ahmed
Associate Professor
Aerospace Engineering

George T. Flowers
Graduate School, Interim Dean

Three-Dimensional Trajectory Optimization in Constrained Airspace

Ran Dai

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
December 17, 2007

THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION IN CONSTRAINED AIRSPACE

Ran Dai

Permission is granted to Auburn University to make copies of this dissertation at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Ran Dai, the daughter of Degang Dai and Hongdi Qiao was born in Taizhou, Jiangsu Province, P.R.China. She graduated from Taizhou High School in June, 1998 and entered school of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics in September, 1998. She graduated in June, 2002 with a Bachelor of Automation Science. From January, 2004, she started her graduate study in Auburn University in the Department of Aerospace Engineering under the supervision of Dr. John, E. Cochran Jr. and received her master degree of Aerospace Engineering in August, 2005.

DISSERTATION ABSTRACT

THREE-DIMENSIONAL TRAJECTORY OPTIMIZATION IN CONSTRAINED AIRSPACE

Ran Dai

Doctor of Philosophy, December 17, 2007
(MAE, Auburn University, 2005)
(B.S., Beijing University of Aeronautics and Astronautics, 2002)

162 Typed Pages

Directed by John E. Cochran Jr.

This dissertation deals with the generation of three-dimensional optimized trajectory in constrained airspace. It expands the previously used two-dimensional aircraft model to a three-dimensional model and includes the consideration of complex airspace constraints not included in previous trajectory optimization studies. Two major branches of optimization methods, indirect and direct methods, are introduced and compared. Both of the methods are applied to solve a two-dimensional minimum-time-to-climb (MTTC) problem. The solution procedure is described in detail. Two traditional problems, the Brachistochrone problem and Zermelo's problem, are solved using the direct collocation and nonlinear programming method. Because analytical solutions to these problems are known. These solutions provide verification of the numerical methods. Three discretization methods, trapezoidal, Hermite-Simpson and

Chebyshev Pseudospectral (CP) are introduced and applied to solve the Brachistochrone problem. The solutions obtained using these discretization methods are compared with the analytical results.

An 3-D aircraft model with six state variables and two control variables are presented. Two primary trajectory optimization problems are considered using this model in the dissertation. One is to assume that the aircraft climbs up from sea level to a desired altitude in a square cross section cylinder of arbitrary height. Another is to intercept a constant velocity, constant altitude target in minimum time starting from sea level. Results of the optimal trajectories are compared with the results from the proportional navigation guidance law. Field of View constraint is finally considered in this interception problem.

The CP discretization and nonlinear programming method is shown to have advantages over indirect methods in solving three-dimensional (3-D) trajectory optimization problems with multiple controls and complex constraints.

Conclusions from both problems are presented and properties of each one are discussed. Finally, suggestions for future research are addressed.

ACKNOWLEDGMENTS

First I would like to express my great appreciation to my advisor, Dr. John E. Cochran Jr., for his instruction, guidance, encouragement and patience during my graduate study in Auburn. His invaluable advice and working attitude benefit me a lot in my research and will continue effect my future career.

Many thanks to Dr. David Cicci, Dr. Roy Hartfield, Dr. Anwar Ahmed, Dr. Andrew Sinclair and Dr. A. Scottedward Hodel for their academic guidance, time and assistance in my graduate courses. The knowledge and skills I studied from their courses made good basis for my dissertation work.

I feel lucky to take classes together with Oluseyi Onawola, Dakshesh Patel, Ravi Duggirala and Justin Martin and developed wonderful friendship with them. I also want to thank my friends Qi Hang, Yawei Han, Hongxia Zhang, Ran Zhou, Rui Chao, Wei Huang and Zhiyang Ding to accompany me through the time in Auburn and always offer me general help.

Finally I would like to express my gratitude to my family, their support and faith in me encouraged me throughout the completion of this degree .

Style manual or journal used Journal of Approximation Theory (together with the style known as "auphd"). Bibliography follows Journal of Aircraft

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `auphd.sty`.

TABLE OF CONTENTS

LIST OF FIGURES xi

LIST OF TABLES xiv

ACRONYM xv

1 INTRODUCTION 1
- 1.1 Background 1
- 1.2 Statement of Problem and Approach 3
- 1.3 Dissertation Outline 5

2 LITERATURE REVIEW 7
- 2.1 Methods in Solving MTTC Problem 7
  - 2.1.1 Gradient Method 7
  - 2.1.2 Energy State Method 8
  - 2.1.3 Singular Perturbation Method 9
  - 2.1.4 Modified Sweep Method 10
  - 2.1.5 Conclusion 11
- 2.2 Methods in Solving MTI Problem 12
- 2.3 DCNLP Application Examples 15
  - 2.3.1 Aircraft Optimal Trajectory Design 15
  - 2.3.2 Space Mission Planning 17
  - 2.3.3 Unmanned Aerial Vehicles (UAV) Optimal Path Planning . . 18
  - 2.3.4 Conclusion 19

3 INDIRECT OPTIMAL CONTROL METHOD 20
- 3.1 Optimal Control Problem 20
- 3.2 Hamiltonian Equation and Necessary Conditions 21
- 3.3 Path Constraints 24

4 THE METHOD OF DIRECT COLLOCATION AND NONLINEAR PROGRAMMING 28
- 4.1 Direct Collocation Method 28
  - 4.1.1 Trapezoidal Method 30
  - 4.1.2 Hermit-Simpson Method 32
  - 4.1.3 Chebyshev Pseudospectral Method 36

4.2 Nonlinear Programming Solver 40
4.3 Mistake Prevention 45

5 USING DCNLP TO SOLVE OPTIMAL CONTROL PROBLEMS 47
5.1 Brachistochrone Problem 47
5.2 Zermelo's Problem 53

6 CONSTRAINED AIRSPACE MINIMUM TIME AND FUEL FLIGHT PATH 56
6.1 2-D MTTC Problem 56
6.1.1 2-D Mathematical Model 56
6.1.2 Energy State Method 61
6.1.3 Gradient Method 63
6.1.4 DCNLP Method 65
6.2 3-D MTTC Problem 69
6.2.1 3-D Mathematical Model 69
6.2.2 Initial NLP Variable Inputs 71
6.2.3 Results 74
6.2.4 Conclusions 76
6.3 3-D Minimum-Fuel-To-Climb Problem 78

7 OTHER CONSTRAINED OPTIMAL TRAJECTORIES 99
7.1 Three-Dimensional Proportional Navigation Guidance Law 99
7.2 Three-Dimensional Minimum-Time Interception Trajectory Planning 103
7.3 Three-Dimensional Minimum-Time Rendezvous Trajectory Planning . 106
7.4 View Constraints 119
7.5 Conclusion 120

8 SUMMARY AND RECOMMENDATIONS 124

BIBLIOGRAPHY 126

A AIRCRAFT PROPULSION AND AERODYNAMIC DATA 130

B GRADIENT METHOD FOR 2-D MTTC PROBLEM 131

C PROPORTIONAL NAVIGATION GUIDANCE LAW ALGORITHM 134

D CHEBYSHEV PSEUDOSPECTRAL COLLOCATION AND NONLINEAR PROGRAMMING CODES IN SOLVING BRACHISTOCHRONE PROBLEM 140

List of Figures

4.1 Trajectory History Discretization 29

4.2 Trapezoidal Discretization 31

5.1 Fermat's Principle 48

5.2 Analytical and DCNLP Results for Brachistochrone Problem 52

5.3 Zermelo's Problem 54

6.1 2-D Aircraft Model 57

6.2 2-D MTTC Trajectory Using Energy State Model 62

6.3 2-D MTTC Trajectory Using Gradient Method 65

6.4 2-D MTTC State and Control Variables History Using Gradient Method 66

6.5 2-D MTTC Trajectory Using DCNLP Method 67

6.6 2-D MTTC State and Control Variables History Using DCNLP Method 68

6.7 2-D MTTC Altitude and Down Range Trajectory Using DCNLP Method 69

6.8 3-D Aircraft Model 71

6.9 Circular Helix Curve Wrapped on A Cylinder 72

6.10 3-D MTTC Trajectory for Case 1 81

6.11 Control and State Variables History for Case 1 82

6.12 3-D MTTC Trajectory for Case 2 83

6.13 Control and State Variables History for Case 2 84

6.14 3-D MTTC Trajectory for Case 3 85

6.15 Control and State Variables History for Case 3 86

6.16 3-D MTTC Trajectory for Case 4 87

6.17 Control and State Variables History for Case 4 88

6.18 3-D MTTC Trajectory for Case 5 89

6.19 Control and State Variables History for Case 5 90

6.20 3-D MFTC Trajectory for Case 6 91

6.21 Control and State Variables History for Case 6 92

6.22 3-D MFTC Trajectory for Case 7 93

6.23 Control and State Variables History for Case 7 94

6.24 3-D MFTC Trajectory for Case 8 95

6.25 Control and State Variables History for Case 8 96

6.26 3-D MFTC Trajectory for Case 9 97

6.27 Control and State Variables History for Case 9 98

7.1 Engagement Geometry 101

7.2 3-D MTI Trajectory using DCNLP for Case 1 108

7.3 3-D MTI Trajectory using PNG for Case 1 108

7.4 Control Factors and Velocity History for Case 1 109

7.5 3-D MTI Trajectory using DCNLP for Case 2 110

7.6 3-D MTI Trajectory using PNG for Case 2 110

7.7 Control Factors and Velocity History for Case 2 111

7.8 3-D MTI Trajectory using DCNLP for Case 3 112

7.9 3-D MTI Trajectory using PNG for Case 3 112

7.10 Control Factors and Velocity History for Case 3 113

7.11 3-D MTI Trajectory using DCNLP for Case 4 114

7.12 3-D MTI Trajectory using PNG for Case 4 114

7.13 Control Factors and Velocity History for Case 4 115

7.14 3-D MTR Trajectory using DCNLP for Case 5 116

7.15 3-D MTR Trajectory using PNG for Case 6 116

7.16 Control Factors and State Variables History for Case 5 117

7.17 Control Factors and State Variables History for Case 6 118

7.18 Control Factors and State Variables History for Case 7 120

7.19 3-D MTI Trajectory using DCNLP for Case 7 121

7.20 Control Factors and State Variables History for Case 7 122

LIST OF TABLES

4.1 Comparison of Three Discretization Methods 39

5.1 Analytical and DCNLP Results for Brachistochrone Problem 51

5.2 Comparison of Brachistochrone Results for Various Collocation Methods 53

5.3 Analytical and DCNLP Results for Zermelo's Problem 55

6.1 Polynomial Coefficients at Different March Number 60

6.2 Boundary Constraints and Performance Limitations for 3-D MTTC Problem 74

7.1 Boundary Constraints and Performance Limitations for 3-D MTI Problem 104

A.1 Thrust as a function of altitude and Mach number from Ref.[2] for aircraft 2. 130

A.2 Lift and drag coefficients as a function of angle of attack and Mach Number for aircraft 2. 130

Acronyms

Aeroassisted Orbital Transfer Vehicle	AOTV
Chebyshev-Gauss-Lobatto	CGL
Chebyshev Pseudospectral	CP
Direct Collocation and Nonlinear Programming	DCNLP
Direct Optimal Control	DOC
Field of View Constraint	FVC
Indirect Optimal Control	IOC
Karush-Kuhn-Tucker	KKT
Minimum-Time-To-Climb	MTTC
Minimum-Fuel-To-Climb	MFTC
Minimum-Time Interception	MTI
Minimum-Time Rendezvous	MTR
Nonlinear Programming Problems	NLPs
optimal control problem	OCP
Proportional Navigation	PN
Sequential Quadratic Programming	SQP
Two-Dimensional	2-D
Three-Dimensional	3-D
Two-Point-Boundary-Value-Problems	TPBVPs
Unmanned Aerial Vehicles	UAV

Chapter 1

Introduction

1.1 Background

Over the last 50 years, the trajectory optimization problem has attracted the interests of many researchers. Due to the development of high speed digital computers, the trajectory optimization technology has been widely applied in both military and civil areas, including space mission planning[24], missile guidance[11], aircraft transportation systems[17], towed-aerial-cable systems[23] and more. Some of the researchers have focused their attention on minimum time-to-climb (MTTC) trajectories. Many methods have been used to solve this kind of problem. As examples, Bryson and Denham[1] used the summarized steepest-ascent method, Calise[3] applied singular perturbation techniques and Ardema[4] used matched asymptotic expansions to get approximate analytical solutions to the MTTC problem. These methods, categorized as indirect optimal control (IOC) methods, are aimed at solving only two-dimensional (2-D) MTTC problems which satisfy the initial and final boundary conditions.

Meanwhile, other researchers paid more attention to minimum-time interception (MTI) problems. Most focused on the onboard generation of trajectories varying from short to long range. For example, Visser, Kelley and Cliff[11] used a singular perturbation method to find an approximate optimal trajectory and then apply neighboring optimal guidance to transfer an aircraft which has deviated from the reference trajectory to the desired trajectory. Kumar, Seywald and Cliff[12] proposed finding medium range optimal trajectories for air-to-air missiles using a three-stage guidance scheme. These methods and some similar ones[13]-[14] are primarily applicable to small deviations from the planned path. For large deviations and unplanned interception, other real-time guidance laws are applied. Proportional navigation guidance (PNG) is very generally applied. This type guidance firstly studied by Yuan[7] was in two dimensions and extended to three dimensions by Adler[8], Duflos[9] and Cochran[10]. PNG has the advantage of on-time realization, but its success depends on some initial conditions. Furthermore, PNG does not consider performance optimization and reduction of final miss distance.

With the increase in computing power, the use of direct collocation and nonlinear programming (DCNLP) to convert two-point-boundary-value-problems (TPBVPs) into nonlinear programming problems (NLPs) the so-called direct optimal control (DOC) methods have become feasible. Early in 1987, Hargraves and Paris[19] applied the collocation method to solve a 2-D MTTC problem. Then, in 1993, Betts and Huffman[20] explained the procedures of the DCNLP methods in more detail, especially the sparse sequential quadratic programming algorithm. In 1999 Horie and

Conway[25] published a paper in which they described how this method was used to solve minimum time and minimum fuel aeroassisted orbital interception problems. Quite recently, in 2006, Geiger, Horn and Delullo[27] applied DCNLP to find unmanned aerial vehicle trajectories that maximize viewing time.

Although the DCNLP has been widely used in aerospace trajectory optimization, it has had limited application in 3-D MTTC and short range MTI trajectory planning. These are the focus of this dissertation.

1.2 Statement of Problem and Approach

The operational airspace of aerospace vehicles, including airplanes and unmanned aerial vehicles, is often restricted. These restrictions may come from local terrain constraints, radar coverage constraints, or collision avoidance constraints. Considering these limitations, aerospace vehicles cannot be assumed to be free to fly anywhere in a given airspace. Forbidden zones can be defined in 2-D or 3-D spaces. So that climbs, descents and other maneuvers are all required to be performed in three dimensions.

One of the principal problems considered here is to find optimal trajectories of an aircraft that is climbing from sea level to a desired altitude in an airspace, consisting of cylindrical volume of unlimited height. The aircraft motion model used here is a 3-D, point-mass model consisting of six first-order differential system equations. It is intuitively expected that a MTTC trajectory in a constrained airspace would be similar to the horizontal projection of a 2-D MTTC trajectory with corresponding initial and final altitudes. Hence, considerable turning will be required if the horizonal range of the constrained airspace is shorter than the required distance in the 2-D MTTC results.

If the IOC method is used, the traditional procedure is to write the Hamiltonian and then derive the necessary conditions in terms of differential equations for Lagrangian multipliers and the maximum(minimum) of the Hamiltonian with respect to the control variables. Because some of the initial conditions are usually unknown, a method for determining them must be applied. For example, in the "shooting method", a guess is made of to all unknown initial state variables and the Lagrangian multipliers and the system equations are integrated forward to get final conditions, which generally deviate from the required boundary conditions. Some adjustment of the initial guess is then made according to the deviation in the final known boundary conditions. The above process is repeated until the final boundary conditions are satisfied within a specified tolerance.

When the DCNLP is applied, the trajectory is discretized into numerous segments, characterized by state and control variables as parameters. In this way, a TPBVP is transformed into a problem of determining the parameters that satisfy the constraints and at the same time maximize or minimize a performance index. The constraints mentioned here can be in the form of system equations constraints, state and control variables constraints, initial and final boundary constraints, and some other forms.

In the 3-D MTTC problem, the final objective of the DCNLP method is to find the parameterized vertical and horizontal load factors as controls and the state variables while minimizing the performance index: the final time. A simplified model representing a climbing helical curve similar to the final optimized MTTC trajectory is used as the initial input of the state and the control variables.

The 3-D MTI problem is to intercept a constant-velocity, constant-altitude target from sea level. This appears to be quite a different type of problem than the MTTC problem. But, if the DCNLP method is applied and all the control factors and state variables are parameterized, the interception point status will be transferred to one of the final boundary conditions. The interception trajectory also show similarities in terms of effect of different initial velocities on time consuming comparing with MTTC trajectory.

1.3 Dissertation Outline

This dissertation is organized as follows. The first part of Chapter 2 contains a review of the previous research on methods for trajectory optimization, especially those used on 2-D MTTC problems. These are the energy state method, gradient method, singular perturbation method and modified sweep method. Then methods used in solving MTI problems will be reviewed. Finally, the DCNLP method is considered and some previous applications discussed.

Chapter 3 deals with the traditional indirect optimal control method which considers both equality and inequality constraints. Chapter 4 introduces the DCNLP method in detail including three kinds of discretization methods, Trapezoidal, Hermit-Simpson method and CP Method and the nonlinear programming solver. Two traditional problems, the Brachistochrone Problem and Zermelo's Problem, are solved using the DCNLP method to illustrate the accuracy of this method in Chapter 5.

In Chapter 6 a description is given of the 3-D aircraft model. Then the indirect optimal method is applied to get a MTTC trajectory in constrained airspace. Following that, the DCNLP method is applied to this same problem. Results from the two methods are compared. Solutions to the minimum-fuel-to-climb (MFTC) problem are also presented.

Chapter 7 starts with a derivation of the traditional PNG law. It is then applied to 3-D intercept problem. Then, results using DCNLP method to solve this same MTI problem are compared to the results using PNG law. Finally, field-of-view constraints are added to the original problem and new trajectories that are consistent with the additional constraints are discussed.

Conclusions are presented in Chapter 8. Some suggestions for future research are also provided.

Chapter 2

Literature Review

The methods developed in solving trajectory optimization, especially MTTC problems, almost became the signs of the development in the fields of optimal control. Betts[18] summarized these methods in his survey and divided them into two major branches: Indirect Optimal Control (IOC) and Direct Optimal Control (DOC). In this chapter, methods in both branches are introduced and compared. The IOC methods as applied to the MTTC and MTI problems are introduced first, followed by the DOC methods, and then the DCNLP.

2.1 Methods in Solving MTTC Problem

2.1.1 Gradient Method

The objective of the gradient method[1], also called the method of steepest-ascent/descent, is to find control variable changes that will cause the maximum increase or decrease in the cost function. A steepest descent computation program will start from an initial guess of the control variables and then integrate the system differential equations to get a nominal path using this initial guess. An adjustment of the control is then is determined by the numerical integration of the adjoint differential equations when certain perturbations are added to the nominal path. This procedure is repeated until the convergence criteria are satisfied.

The gradient method has been applied to many TPBVPs. This systematic numerical procedure requires suitable initial guess and reasonable selection of the mean-square perturbation of the control variable program which is named as "step size". Under some cases, a good initial guess is not available, and this may lead to divergent results. Also, when system equations include many state variables, deriving the adjoint differential equations will be expensive and tedious. These properties limited the application of this method.

2.1.2 Energy State Method

Normally, the 2-D aircraft point-mass model is represented by its state variables of speed, flight path angle and altitude. At the same time, its total energy per unit mass can be treated as the summation of potential energy and kinetic energy expressed in terms of the variables of speed and altitude. When this energy is used as a state variable, the minimum-time approximate model is to find the maximum time derivative of total energy at different given energy levels. Then, the MTTC path is the connection of the continuous summit point of excess power at constant energy.

This approximation method can also be applied to MFTC and minimum range glide problems[2].

The results obtained from the energy-state approximation model are similar to those obtained using the gradient method. But, if higher accuracy is required for the solution, this approximate method is not suitable.

2.1.3 Singular Perturbation Method

In the system equations, some state variables have more effect on the solution than others. Some states are important, while others can be neglected without any obvious change to the original system. When these neglected terms are deleted from the original equations, the solution to the approximate model is called the outer or slow solution. Generally, the optimal control problems are in the form of TPBVPs. It is easy to see that the simplified model cannot satisfy all of the final boundary constraints. So in a fast time region, the state variables will vary rapidly from the outer stage to the boundary constraints, which is called inner or fast solution. This phenomena is similar to the solution of Navier-Stokes equations, here the names of outer and inner solution. In the fluid mechanics problem, fast and slow layers of the hydrodynamics solution exist. Finally, the theory of singular perturbation method is to find the combination of simplified model and asymptotic expansion series which represents the approximate and exact solution separately[3].

In the 2-D MTTC problem, the flight path angle is less important than the altitude and altitude is less important than the energy state. So the zeroth-order outer solution includes only the energy state variables. If higher accuracy is required, the first-order or higher order terms will be considered. For the MTTC problem, the first-order solution is very close to the solution obtained using the gradient method.

Compared to the gradient method, the singular perturbation method has advantage of greatly reduced computation burden and avoidance of initial guess for the adjoint variables. But its application does has some limitation. First of all, some of the terms must have smaller effect than others. Secondly, the boundary layer equations need to reach a stable solution which will makes the "matching" possible.

2.1.4 Modified Sweep Method

In the application of singular perturbation method in TPBVPs, it's well known that the two-time-scale solutions are composed of an outer boundary solution that slowly approaches an equilibrium point and an inner solution rapidly changes from the equilibrium point to the final boundary conditions. So, finding such a equilibrium point became the key point of the problem. Rao and Mease[5] proposed a modified sweep method to solve especially for the outer segment, also named as the infinite horizon regulator problem.

In the modified sweep method, a basis vector is first chosen to identify the stable and unstable variables in the system. Normally, the eigenvectors of the Jacobian matrix of the phase rate vector, which is the time derivative of the state and adjoint variables, is a good choice for this basis vector. The forward integration uses the unstable rate coordinate as input and the backward integration uses the stable rate coordinate as input so that the unstable behavior will be suppressed in each sweep. The new value for the final stable rate and initial unstable rate obtained from each loop are saved to be used in the next sweep to improve the convergency. The entire process is repeated until relative error reaches the required level. This modified sweep method provides a more accurate way to estimate the initial adjoint variables, but it is used only for the zeroth-order approximation of the system. A complete solution to the whole problem requires more work.

2.1.5 Conclusion

The above IOC methods reviewed are all designed for solving TPBVPs without considering path constraints. For the gradient method, finding a good initial guess for the adjoint variables is the major difficulty. Generally, the state variables and adjoint variables are combined together so that poor first time estimation may even cause these values to exceed the numerical range of the computer. Although some techniques are proposed to improve the efficiency of initial guess[28], they simultaneously add complexity to the problem while reducing the sensitivity of the solution. The energy state approximation method is applicable only to a 2-D airplane model. When another system model is used, another optimization method must be employed. Both singular perturbation and modified sweep methods are based on the Hamiltonian and the found necessary conditions. But these procedures do not produce unified solutions to every optimization problem which means that although the basic theory is same, the analytic expression must be derived according to different problems. Therefore, when a system model is complex and includes many state variables, this task is quite heavy.

When path constraints are considered, the Hamiltonian must be reformulated with introduction of new Lagrange multipliers. This makes the problem more difficult to solve, especially when the constraints are inequalities, the trajectory needs to be divided into piecewise parts of active and inactive arcs to meet the specified path inequality constraints.

2.2 Methods in Solving MTI Problem

Optimal methods for solving MTI problems have been used for many years. Practical realization of "optimality" requires the consideration of uncertainties and disturbances like wind gusts, temperature changes, weight loss, deviation of starting point and interception point, command error and other unexpected perturbations.

Most optimal methods focus on generation of onboard trajectories varying from short to long range.

For example, Visser, Kelley and Cliff[11] use a singular perturbation method to find the approximate optimal trajectory and then apply neighboring optimal guidance to transfer the aircraft, which has deviated from the reference trajectory, to the desired trajectory. This procedure is composed of two major parts. The first part is to get the nominal path that will find the minimum time transfer trajectory from the initial conditions to the dash point that will possess maximum velocity. The singular perturbation method applied there has the same theory basis as mentioned above and the zeroth order approximate model is considered here with application of state energy model. The second part is to generate the near-optimal guidance law to eliminate deviation effects and pull the interceptor back to the optimal path. Then two control functions, load factor and bank angle, are linearized with respect to the two state variables, altitude and flight path angle, respectively. The feedback coefficients derived from the linearized model are evaluated at the reference trajectory and multiplied by the perturbations of the two state variables. Results obtained from the above calculation are adjusted values about the nominal control commands.

Kumar, Seywald and Cliff[12] proposed finding medium range optimal trajectories for air-to-air missiles using a three-stage guidance scheme. The first stage is the boost phase which requires the missile reaches high altitude and desired horizontal distance before the boost motor shut down. Also the nominal time optimal path is derived with consideration of initial perturbation only. The resultant control load factor is limited which causes the missile remains at the maximum acceleration limit line for a finite time period. The second stage is mid-course guidance considering both state perturbations and a maneuvering target. Transversal comparison and performance augmentation methods has been applied to this longest duration phase. For the target is capable of maneuvers, the terminal guidance law requires autonomous tracking of the final object. Pure proportional navigation is applied in the final stage to intercept the target. These three guidance laws applied to boost, sustain and coast phases are more accurate than traditional single phase approximate model.

Even the three-phases guidance scheme is based on a simplified approximate model. But under some circumstances, such an approximate model cannot adequately represent the system and a more accurate optimization is required. Also at the final stage when classical Proportional Navigation (PN) guidance law is applied, the time efficiency is neglected in order to realize real-time control objective. So at least the final phase is not time optimized which leaves space to make the trajectory further improved.

Except for the MTI problem, the optimal guidance scheme is also applied to other areas. For example, Corban, Calise and Flandro[15] used the combination of singular perturbation and feedback linearization techniques to solve minimum fuel transatmospheric vehicle assent. Bollino, Ross and Doman[16] used pseudospectral methods to generate the optimal trajectory with minimal miss distance for the reentry vehicle, then Runge-Kutta integration scheme is applied to integrate forward with conjunction of disturbances factors to get the current state variables which is feedback to the on-time optimal trajectory generator to find new control variables at the current situation. Their simulation results showed that this optimal guidance scheme can guide the vehicle to an accurate landing even under hurricane wind effect Category 5.

2.3 DCNLP Application Examples

The following are examples of DCNLP applications to aircraft optimal trajectory design, space mission planning, and unmanned aerial vehicles optimal path planning.

2.3.1 Aircraft Optimal Trajectory Design

1. Hargraves and Paris[19] solved the 2-D MTTC problem using third-order descretization and nonlinear programming. This technology, applied in 1986, is not as mature as the method used today and some refinement is required to obtain more accurate results, such as variable scaling, nodes selection, partial

derivatives computation and data smoothing. Although not as mature as today's technology, the method was well developed to successfully solve a wide variety of problems at that time.

2. Betts and Huffman[20] described a nonlinear programming algorithm in detail in 1993, including a quadratic programming subproblem, merit function, parameter definition, algorithm strategy, finding a feasible point and minimization process. They also compared characteristics of different transcription methods, trapezoidal, hermite and Runge-Kutta, in terms of computation cost, error and robust estimation. Maximum crossrange and MTTC problem trajectories are simulated using the above different transcription methods.

3. Ringertz[21] solved the minimum fuel turn problem using this DCNLP method. He applied six-degree-of-freedom dynamics model of the aircraft with control of lift and bank angle. The aircraft was expected to make a heading angle turn of 180° and keep the same altitude, velocity and flight path angle as initial conditions while maximizing the final fuel mass.

4. Norsell[22] found a multistage aircraft long distance optimal trajectory considering radar coverage constraints. In this problem, the 3-D aircraft model is simplified under different additional conditions, such as constant altitude, constant indicated airspeed or constant march number. The performance to be

minimized is the radar cross section beam which will reduce the possibility of being detected when passing the radar station.

5. Williams, Sgarioto and Trivailo[23] applied a DCNLP method to aerial-towed cable system path planning. The cable tip was attached to a aircraft with wings and control surfaces and the interaction between the cable and aircraft was connected by the tension force of the cable tip. Aerodynamic drag and wind forces are also considered in the aircraft model. The task was designated to let the cable tip pass specified points in 3-D airspace while optimizing combined aircraft performance.

2.3.2 Space Mission Planning

1. Herman and Spencer[24] used higher-order collocation to solve a wide variety of orbital transfer problems including from low Earth orbit to geosynchronous Earth orbit, medium Earth orbit, and high Earth orbit while minimizing fuel consumption. The numerical results of the optimized trajectories were compared to the analytical solution which used burn-coast-burn structured assumption and the comparison showed the results using the two methods are very close.

2. Horie and Conway[25] developed the minimum-time and minimum-fuel optimal trajectories for an aeroassisted orbital transfer vehicle (AOTV) to intercept a target located on a lower circular orbit. Because the low density of the atmosphere at high altitude will cause the large reduction of the aerodynamic force and cause the disappearance of the control variables, the maximum altitude constraints are considered. Another constraint introduced is one on the aerodynamic heating that comes from the atmospheric flight. The results indicate that the AOTV has more advantages on time consumption than ballistic flight in interception problem.

3. Betts[26] published his results about space shuttle optimal orbit transfer in the presence of uncertainty. The transfer task was accomplished by two successive stages with different impulsive velocity increments and the performance, payload, was maximized. A solid rocket motor was ignited to perform the designed nominal task and a liquid propellant reaction control system was used to compensate for the off-nominal part. The major uncertainty considered here came from the specific impulse predicted by weight history.

2.3.3 Unmanned Aerial Vehicles (UAV) Optimal Path Planning

Although DCNLP has wide application in aircraft and space mission planning, it has limited use in UAVs. So in 2006 Geiger, Horn, DeLullo and Long[27] published their studies on UAVs focusing on maximizing the viewing time for a camera fixed on the UAV. A single UAV may perform tasks like rendezvousing a slow or fast target or surveil a stationary target while maximizing the sensor coverage of the object. When two UAVs are involved, the optimized trajectory may provides full time coverage of the stationary target. But this method is still not fast enough to be applied in realtime operation, faster algorithm is required for onboard calculation.

2.3.4 Conclusion

With the advantages of fast convergence, avoidance of adjoint variable estimation, and the capacity of including complex boundary conditions, the DCNLP has been successfully applied in wide areas of aircraft and spacecraft research. But this method has limited application in 3-D trajectory optimization in constrained airspace. In order to simplify the problem, the previous studies focused on 2-D model considering constraints about control variables only. The following discussion will make problems like MTTC and MFTC more complete and accurate with the same properties showed in 2-D model. The optimality verification will also be conducted to prove the accuracy of this method.

Chapter 3

Indirect Optimal Control Method

3.1 Optimal Control Problem

The objective of an optimal control problem (OCP) is to find the history of the control variable(s) that will maximize or minimize a given performance index while satisfying the system constraints. The system constraints considered herein include first-order, ordinary differential equations subject to initial and final boundary conditions and some additional constraints on the states and controls. The differential equations are written here as

$$\dot{\underline{x}} = \underline{f}(t, \underline{x}, \underline{u}) \tag{3.1}$$

where $\underline{x}$ is an $n \times 1$ vector of states, $\underline{f}$ is an $n \times 1$ vector of functions, t is the time and $\underline{u}$ is an $m \times 1$ vector of controls. With prescribed initial conditions

$$\underline{x}(0) = \underline{x}_0 \tag{3.2}$$

and prescribed final boundary conditions

$$\underline{\psi}(t_f, \underline{x}_f) = 0 \tag{3.3}$$

where $\underline{\psi}$ is a $p \times 1$ vector of functions, t_0 is the initial time and the terminal time t_f is free. The scalar performance index is expressed as

$$J = \phi(t_f, \underline{x}_f) + \int_{t_0}^{t_f} L(t, \underline{x}, \underline{u})dt \tag{3.4}$$

where ϕ and L are scalars.

The basic idea of IOC is to find the control u that will make the gradient of the objective function be zero. So it is necessary to compute the Jacobian gradient of J to decide which solution will drive it to zero. When the system constraints are considered, the Hamiltonian equation is introduced as well as the Euler-Lagrange equations.

3.2 Hamiltonian Equation and Necessary Conditions

The traditional, or indirect, optimal control method is to formulate the Hamiltonian function with Lagrange multipliers, also called adjoint variables vector $\lambda_j(t)$

$$H = L + \underline{\lambda}^T \underline{f} \tag{3.5}$$

where $\underline{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T$. Now, the optimal states $\underline{x}(t)$, controls $u(t)$ and Lagrange multipliers $\underline{\lambda}(t)$ must satisfy the Euler-Lagrange equations

$$\dot{\underline{x}} = \frac{\partial H}{\partial \underline{\lambda}^T} = f(t, \underline{x}, \underline{u}) \tag{3.6}$$

$$\dot{\underline{\lambda}}^T = -\frac{\partial H}{\partial \underline{x}} = -\frac{\partial L}{\partial \underline{x}} - \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{x}} \tag{3.7}$$

The optimized performance is obtained by choosing the control variables, such that

$$0 = \frac{\partial H}{\partial \underline{u}} = \frac{\partial L}{\partial \underline{u}} + \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{u}} \tag{3.8}$$

Then, the solution to the indirect OCP is found by solving Eqn.3.6-Eqn.3.8 subject to the constraints Eqn.3.2 and Eqn.3.3. If there are n state variables x_j and m control variables u, from Eqn.3.8, one can express control variables as functions of the n state variables x_j and n adjoint variables λ_j. By inserting these new forms of u into Eqn.3.6 and Eqn.3.7, the problem is solved by integrating the $2n$ first-order derivative functions of x and λ while satisfying the specified boundary conditions on x. If we also know the initial conditions of λ, the problem could be finished by solving the differential equations. Unfortunately, the initial conditions on the λ_j are not known. Moreover, it is often difficult to find their values.

The initial and final state and adjoint variables satisfy boundary constraints that have the following forms:

Initial Conditions

$$\delta x_j(t_0)\lambda_j(t_0) = 0 \tag{3.9}$$

Final Conditions

$$\lambda(t_f) = (\frac{\partial \phi}{\partial \underline{x}} + \underline{\nu}^T \frac{\partial \underline{\psi}}{\partial \underline{x}})^T_{t=t_f} \tag{3.10}$$

$$[\frac{\partial \phi}{\partial t} + \underline{\nu}^T \frac{\partial \underline{\psi}}{\partial t} + (\frac{\partial \phi}{\partial \underline{x}} + \underline{\nu}^T \frac{\partial \underline{\psi}}{\partial \underline{x}})\underline{f} + L]_{t=t_f} = 0 \tag{3.11}$$

$$\underline{\psi}(t_f, \underline{x}_f) = 0 \tag{3.12}$$

where $\underline{\nu}$ is a $p \times 1$ new introduced Lagrange multiplier vector.

From the initial conditions, it can be seen that either the initial condition $x_j(t_0)$ is given or $\lambda_j(t_0) = 0$. This means that the initial conditions of either the state variables or the adjoint variables are known. The final boundary conditions must satisfy all three constraints Eqn.3.10, Eqn.3.11 and Eqn.3.12. Eqn.3.11 is a general case with the state variables specified at an unspecified terminal time. When the final time is specified, this equation is not needed. In either case, the information provided at the final point is not sufficient to solve for the final state variables, Lagrange multipliers and final time. If all these variables were known at the final time , it would be easy to integrate them backward to get the initial conditions, but they are not. Finally, the backward integration method still turned out to be unsuccessful.

Considering the importance and difficultly of getting the initial values of the λ_j in solving optimal problems, many researchers have proposed methods to find them. Since the adjoint variables often have no easily discernable physical meaning, we cannot determine them from common sense. Furthermore, they sometimes tend to be very sensitive to changes in the adjoint variables. That is, a small change can cause great difference in the calculated answer. All this adds up to make obtaining initial values of adjoint variables very difficult. This difficulty provided the motivation for finding ways to avoid using the adjoint variables.

3.3 Path Constraints

In the above discussion, we did not consider the extra constraints, which are functions of state and control variables. Some are types of equality constraints including state and control variables

$$\underline{C}(\underline{x}, \underline{u}, t) = 0 \tag{3.13}$$

where $\underline{C}$ is a $q \times 1$ vector. When these constraints are considered in the OCP, the Hamiltonian function may be reformulated by introducing additional Lagrange multipliers μ

$$H = L + \underline{\lambda}^T \underline{f} + \underline{\mu}^T \underline{C} \tag{3.14}$$

as well as new adjoint equations

$$\dot{\underline{\lambda}}^T = -\frac{\partial L}{\partial \underline{x}} - \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{x}} - \underline{\mu}^T \frac{\partial \underline{C}}{\partial \underline{x}} \tag{3.15}$$

and optimality conditions

$$0 = \frac{\partial L}{\partial \underline{u}} + \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{u}} + \underline{\mu}^T \frac{\partial \underline{C}}{\partial \underline{u}} \tag{3.16}$$

The path constraints C here reduces the number of independent control variables to $m-q$. If L, f and C are not explicit functions of time, then the Hamiltonian equation will be a constant value over the whole time interval[29, 30]. The control variables can be decided from the Pontryagin's minimum principle which is formulated as

Theorem 3.3.1 *[30]The optimal control $u^*(t)$ must make the Hamiltonian an absolute minimum with respect to the control at every point of the minimal path $x^*(t)$ and stated as*

$$H(t, x^*, u^*, \lambda^*) \leq H(t, x^*, u, \lambda^*) \qquad t \in [t_0, t_f] \tag{3.17}$$

Other kinds of constraints are inequality constraints

$$\underline{C}(\underline{x}, \underline{u}, t) \leq 0 \tag{3.18}$$

where

$$\mu \begin{cases} = 0, & C < 0 \\ > 0, & C = 0 \end{cases} \tag{3.19}$$

In this case, the adjoint equations are

$$\dot{\underline{\lambda}}^T = -\frac{\partial H}{\partial \underline{x}} = \begin{cases} -\frac{\partial L}{\partial \underline{x}} - \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{x}}, & C < 0 \\ -\frac{\partial L}{\partial \underline{x}} - \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{x}} - \underline{\mu}^T \frac{\partial \underline{C}}{\partial \underline{x}}, & C = 0 \end{cases} \tag{3.20}$$

and the optimality conditions are

$$\frac{\partial H}{\partial \underline{u}} \equiv \frac{\partial L}{\partial \underline{u}} + \underline{\lambda}^T \frac{\partial \underline{f}}{\partial \underline{u}} + \underline{\mu}^T \frac{\partial \underline{C}}{\partial \underline{u}} = 0 \tag{3.21}$$

In solving such inequality constraints problems, it is common practice to divide the solution space into active($C = 0$) and inactive($C < 0$) arcs. In the inactive arcs, the solution is assumed to satisfy the additional constraints and necessary condition Eqn.3.21 is solved as discussed above. In the active pieces, Eqn.3.18 and Eqn.3.20 must be combined together to solve $u(t)$ and $\mu(t)$. The classification of active and inactive arcs may start by deciding the junction point between these two types. If the control $u(t)$ at the junction point is discontinuous, it is called a corner. But if λ, $\frac{\partial H}{\partial u}$ and H are continuous, it follows that $u(t)$ and $\mu(t)$ will be continuous across the junction point. Generally, making a priori estimate of the arc sequences is quite difficult. Without prior knowledge of the number of the constrained and unconstrained subarcs, it is very difficult to fix the accurate junction point and each arc boundary conditions.

Then, these constraints all add to the complexity of the problem and make finding the initial values of the adjoint variables even more difficult. Also when the system equations include many state variables, deriving the adjoint equations will seem to be a nightmare. These are some of the reasons why the DCNLP method introduced in Chapter 4 is so attractive as an efficient way to solve complex optimization problems.

CHAPTER 4

THE METHOD OF DIRECT COLLOCATION AND NONLINEAR PROGRAMMING

In this chapter we introduce the idea of direct collocation and describe procedures for the trapezoidal, Hermite-Simpson and CP discretization methods. The nonlinear programming solver is also introduced along with its detailed algorithm. The importance of the application of direct collocation method in setting up the NLP solver will also be illustrated.

4.1 Direct Collocation Method

A complete trajectory of an aerospace vehicle is composed of numerous points representing its continuous coordinates as they change from initial time to final time. Actually, if we pick up some discrete characteristic points on this trajectory, they can be used to reproduce an approximate trajectory that is almost as good, for some purpose, as the continuous trajectory. The basic idea of Direct Collocation (DC) is to discretize the continuous solution to a problem represented by state and control variables by using linear interpolation to satisfy the differential equations. In this way, an OCP is transformed into a nonlinear programming problem (NLPP). Since the solution to the OCP is in terms of infinitely many values of state and control

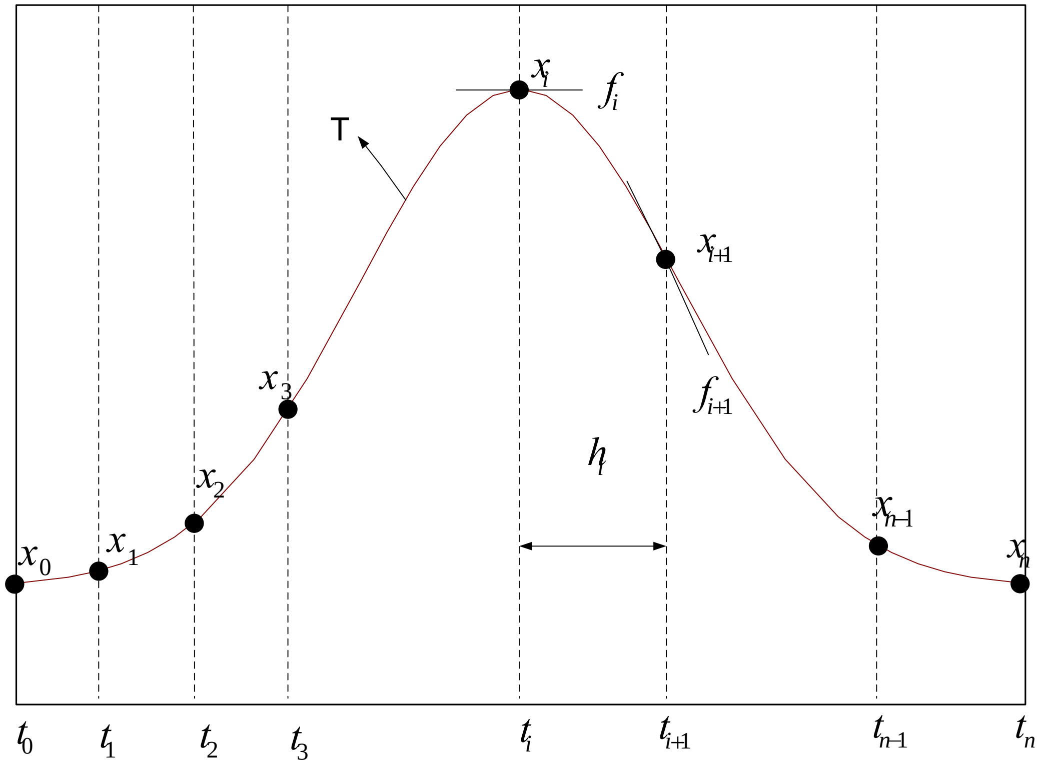

Figure 4.1: Trajectory History Discretization

variables, the DC is an approximation. However, this is also the case for any solution obtained via numerical integration.

To explain the DC method, we present a simple example illustrated in Fig. 4.2. The continuous trajectory, T, is divided into n segments at $n+1$ different time points such that $t_0 = t_1 < t_2 < t_i < \cdots < t_n = t_f$, where t_0 is the initial time, t_f is the final time, and the $n + 1$ individual time points are called nodes. The value of the state vector at the ith node is represented by $\underline{x}_i$, the control vector is $\underline{u}_i$, and the derivative of the state vector at node i is expressed as $\underline{f}_i$. The time interval between node i and

$i+1$ is

$$h_i = t_{i+1} - t_i \tag{4.1}$$

Our objective is to use these discrete nodes to reproduce system equations. Three discretization methods are introduced here: the Trapezoidal Method, Hermit-Simpson Method and CP method. A description of the procedure and properties of each of the methods is also provided. In both of the methods, the defect equations are enforced on each node equal to zero, so that the state vector and control vector at those nodes are constrained to satisfy the system equations.

4.1.1 Trapezoidal Method

In the trapezoidal method, the defect vector d_i of phase i is defined as

$$\underline{d}_i = \underline{x}_{i+1} - \underline{x}_i - \frac{h_i}{2}[f(\underline{x}_i, \underline{u}_i) + f(\underline{x}_{i+1}, \underline{u}_{i+1})] \tag{4.2}$$

These defect vectors are set according to the approximate trapezoidal integration algorithm as shown in Fig. 4.2. We know the system equation is expressed as $\dot{\underline{x}} = \underline{f}(t, \underline{x}, \underline{u})$, so in this figure the left area of the dash line under the curve is

$$\underline{x}_i = \int_{t_0}^{t_i} \underline{f}(t, \underline{x}, \underline{u}) dt \tag{4.3}$$

and the left area of the straight line under the curve is

$$\underline{x}_{i+1} = \int_{t_0}^{t_{i+1}} \underline{f}(t, \underline{x}, \underline{u})dt \tag{4.4}$$

The shaded area is the difference between the two areas listed above and if this area is treated as a trapezoidal area, it can be calculated as

$$\underline{x}_{i+1} - \underline{x}_i \cong \frac{h_i}{2}[\underline{f}_i + \underline{f}_{i+1}] \tag{4.5}$$

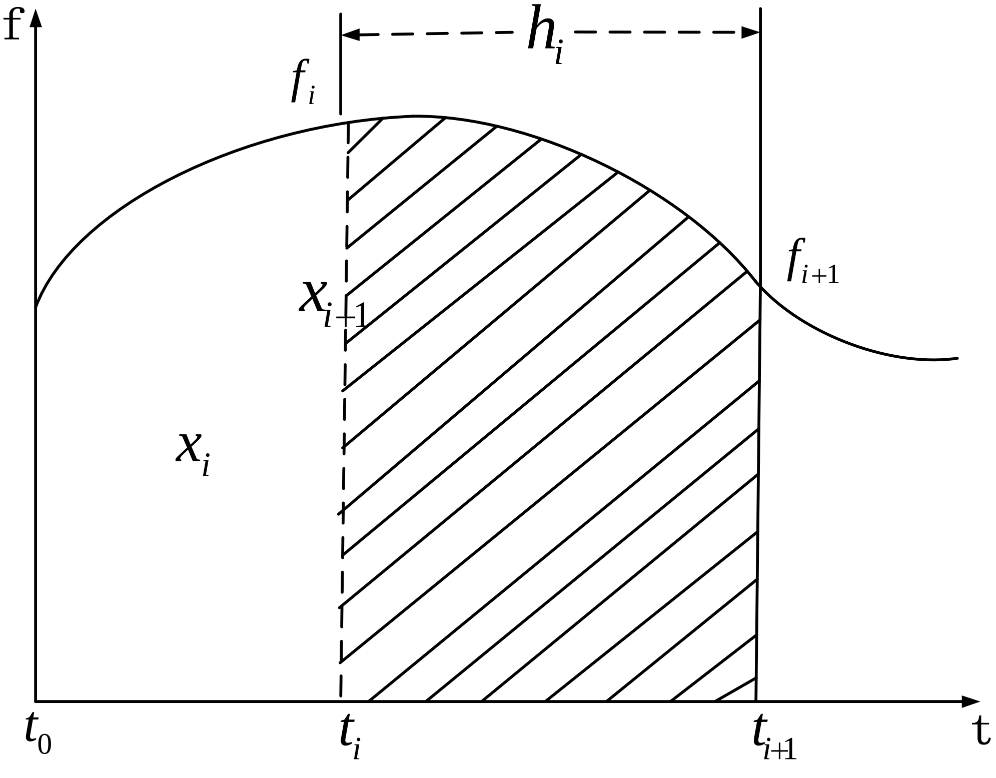

Figure 4.2: Trapezoidal Discretization

When more nodes are considered, and the difference in $\underline{x}_i$ and $\underline{x}_{i+1}$ becomes smaller, in this approximate area calculation, each approximate area can be treated as being very close to the actual area. Then, the right part of Eqn.4.5 may be moved to the left-hand side to get

$$\underline{x}_{i+1} - \underline{x}_i - \frac{h_i}{2}[\underline{f}_i + \underline{f}_{i+1}] = 0 \tag{4.6}$$

4.1.2 Hermit-Simpson Method

If Simpson's integration rule and the Hermite interpolation method are used instead of the trapezoidal rule, then the defect vector of phase i is defined to be

$$\underline{d}_i = \underline{x}_{i+1} - \underline{x}_i - \frac{h_i}{6}[f(\underline{x}_i, \underline{u}_i) + 4f(\underline{x}_{ci}, \underline{u}_{ci}) + f(\underline{x}_{i+1}, \underline{u}_{i+1})] \tag{4.7}$$

where $\underline{x}_{ci}$ and $\underline{u}_{ci}$ are the state and control vectors, respectively, at the center of nodes i and $i+1$ and are defined as

$$\underline{x}_{ci} = \frac{1}{2}(\underline{x}_i + \underline{x}_{i+1}) + \frac{h_i}{8}[f(\underline{x}_i, \underline{u}_i) - f(\underline{x}_{i+1}, \underline{u}_{i+1})] \tag{4.8}$$

and

$$\underline{u}_{ci} = \frac{1}{2}(\underline{u}_i + \underline{u}_{i+1}) \tag{4.9}$$

To derive this discretization result, first the Simpson's rules of integration is applied here instead of trapezoidal integration.

Theorem 4.1.1 *(Simpson's Rule) Consider* $y = f(x)$ *over* $[x_0, x_2]$*, where* $x_1 = x_0 + \frac{h}{2}$*, and* $x_2 = x_0 + h$*. Simpson's rule is*

$$SR(f, h) = \frac{h}{6}(f(x_0) + 4f(x_1) + f(x_2)) \tag{4.10}$$

This is an numerical approximation to the integral of $f(x)$ *over* $[x_0, x_2]$*, that is,*

$$SR(f, h) \cong \int_{t_0}^{t_2} f(x)dt \tag{4.11}$$

According to Simpson's rule, the area difference between x_i and x_{i+1} over the interval $[t_i, t_{i+1}]$ is expressed in a new form as

$$x_{i+1} - x_i = \frac{h_i}{6}[f(x_i, u_i) + 4f(x_{ci}, u_{ci}) + f(x_{i+1}, u_{i+1})] \tag{4.12}$$

In the above equation the center point x_{ci} is interpolated with application of Hermite interpolation. To explain Hermite interpolation, Newton's divided difference method of interpolation is introduced first.

Theorem 4.1.2 *(Newton Polynomial) Consider* $y = f(x)$*, given a set of* $k + 1$ *data points* $(x_0, y_0) \ldots (x_k, y_k)$*, where no two* x_j *are the same, the interpolation polynomial* *in the Newton form is a linear combination of Newton basis polynomials*

$$N(x) = \sum_{j=0}^{k} a_j [\prod_{i=0}^{j-1} (x - x_i)] \tag{4.13}$$

The coefficients a_j are defined as the divided differences

$$a_j = [y_0, ...y_j] \tag{4.14}$$

where the notation used for the divided differences is expressed as

$$[y_0, ...y_j] = \frac{[y_1, \ldots, y_j] - [y_0, \ldots, y_{j-1}]}{x_j - x_0} \tag{4.15}$$

Then the Newton polynomial can be written as

$$N(x) = [y_0] + [y_0, y_1](x - x_0) + \cdots + [y_0, \ldots, y_k](x - x_0) \ldots (x - x_k) \tag{4.16}$$

For better illustration, the divided differences can be written in a form of a table,

$$\begin{array}{cccccc}
x_0 & [y_0] & & & & \\
x_1 & [y_1] & \frac{[y_1]-[y_0]}{x_1-x_0} & & & \\
x_2 & [y_2] & \frac{[y_2]-[y_1]}{x_2-x_1} & \frac{\frac{[y_2]-[y_1]}{x_2-x_1}-\frac{[y_1]-[y_0]}{x_1-x_0}}{x_2-x_0} & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
x_n & \cdots & \cdots & \cdots & \cdots & [y_0,\ldots,y_n]
\end{array}$$

Hermite interpolation is very similar to the Newton divided difference method, but it uses the data points repeatedly and considers the derivatives at data points.

Theorem 4.1.3 *(Hermite Interpolation) The derivatives are treated as extra points, and in the divided difference table, the points are repeated. To avoid division by zero, the values where the division by zero would take place are replaced with the derivatives, multiplied by a constant, depending on the position in the table. Using the notation used in the Newton polynomial theorem, if point x_i is repeated n times, then $[x_i, x_i \cdots, x_i] = f^{(n-1)}(x_i)/(n-1)!$*

When this method is applied to the interpolation of $\dot{x} = f(x)$ with two data points x_i and x_{i+1}, its divided differences are listed as follows:

$$\begin{array}{ccccc}
x_i & x_i & & & \\
x_i & x_i & f_i & & \\
x_{i+1} & x_{i+1} & 1 & \frac{1-f_i}{x_{i+1}-x_i} & \\
x_{i+1} & x_{i+1} & f_{i+1} & \frac{f_{i+1}-1}{x_{i+1}-x_i} & \frac{f_{i+1}+f_i-2}{(x_{i+1}-x_i)^2}
\end{array}$$

So the Hermite polynomial can be written as

$$H(x) = x_i + f_i(x - x_i) + \frac{1 - f_i}{x_{i+1} - x_i}(x - x_i)^2 + \frac{f_{i+1} + f_i - 2}{(x_{i+1} - x_i)^2}(x - x_i)^2(x - x_{i+1}) \quad (4.17)$$

Then the center point $x_{ci} = H(\frac{x_i + x_{i+1}}{2})$, so we can insert $x = \frac{x_i + x_{i+1}}{2}$ in Eqn.4.17 to get the result for x_{ci} that is exactly the same as Eqn.4.8

4.1.3 Chebyshev Pseudospectral Method

The above two methods and some higher order-discretization techniques[24] put the constraints on the defect phase between two adjacent nodes and the distribution of the nodes is arbitrary. That is, the collocation points can be dense in one area and sparse in another. On the other hand the ideal distribution as we expected is to be evenly located for the nodes in the time interval considered. Thus, the CP method[32, 33, 34], which uses Chebyshev-Gauss-Lobatto (CGL) collocation to locate these points, may perform this nodes distribution task better. The standard interval considered here is unified as $\tau \in [-1, 1]$ with collocation points τ_k set as

$$\tau_k = -cos(\pi k/N), \quad k = 0, \ldots, N \quad (4.18)$$

By using a linear transformation, the actual time t can be expressed as a function of τ via

$$t = [(t_f - t_0)\tau + (t_f + t_0)]/2 \tag{4.19}$$

The state and control variables can then be approximated by Nth-order Lagrange interpolating polynomials

$$x^N(\tau) = \sum_{j=0}^{N} x_j \phi_j(\tau) \tag{4.20}$$

$$u^N(\tau) = \sum_{j=0}^{N} u_j \phi_j(\tau) \tag{4.21}$$

where

$$\phi_j(\tau) = \prod_{l=1, l\neq j}^{N} \frac{\tau - \tau_j}{\tau_j - \tau_l} \tag{4.22}$$

and

$$x^N(\tau_k) = x_k, \quad u^N(\tau_k) = u_k \tag{4.23}$$

From Eqn.4.20, it is clear that the derivative of $x^N(\tau_k)$ can be expressed as

$$s_k = \frac{d}{dt} x^N(\tau_k) = \sum_{j=0}^{N} x_j \frac{d}{dt} \phi_j(\tau_k) \quad k = 0, , \ldots, N \tag{4.24}$$

and expanded as

$$\begin{pmatrix} \frac{d}{dt}x^N(\tau_0) \\ \vdots \\ \frac{d}{dt}x^N(\tau_N) \end{pmatrix} = D \begin{pmatrix} x_0 \\ \vdots \\ x_N \end{pmatrix} \tag{4.25}$$

where D is the $(N+1) \times (N+1)$ differentiation matrix and is defined as

$$D_{kj} = \frac{d}{dt}\phi_j(\tau_k), \quad j, k = 1, \ldots, N+1 \tag{4.26}$$

Eqn.4.26 is the general form of the differentiation matrix. For the Chebyshev differentiation matrix, D_{kj} is defined in the following theorem.

Theorem 4.1.4 *For each $N \geq 1$, let the rows and columns of the $(N+1) \times (N+1)$ Chebyshev spectral differentiation matrix D_N be indexed from 0 to N. The entries of this matrix are*

$$D_{kj} = \begin{cases} -(c_k/c_j)[(-1)^{(k+j)}/(\tau_k - \tau_j)], & j \neq k \\ -(2N^2+1)/6, & j = k = 0 \\ (2N^2+1)/6, & j = k = N \\ \tau_k/2(1-\tau_k^2), & otherwise \end{cases} \tag{4.27}$$

where $c_0 = c_N = 2$ and $c_1 = \ldots = c_{N-1} = 1$.

Unlike the trapezoidal and Hermite-Simpson collocation methods which are based on enforcing the defect vector of mid points be zero to satisfy the system differential equations, the CP method enforces constraints directly at the CGL points selected by

$$d_k = s_k - \frac{t_f - t_0}{2} f(x_k, u_k, t_k) = 0, \quad k = 0, \ldots, N \tag{4.28}$$

Then, the derivative of the state variables can be calculated using these nodes themselves with the differentiation matrix. In this way, the CP method achieves a high degree of accuracy using orthogonal polynomials instead of the numerical integration polynomials.

All three collocation methods can approximate the system equations, but each of them has different properties, which are listed in Table.4.1. Examples using these methods are presented in Chapter 5.

Table 4.1: Comparison of Three Discretization Methods

	Trapezoidal	Hermite-Simpson	Chebyshev Pseudospectral
State Variables Error	$O(h^2)$	$O(h^4)$	$O(h^4)$
Control Variables Error	$O(h^1)$	$O(h^2)$	$O(h^3)$
Robust	good	less	less
Convergence	fast	slow	fast

4.2 Nonlinear Programming Solver

The NLP solver used to solve the NLPP considered in this work is based on a Sequential Quadratic Programming (SQP) algorithm and is called SNOPT[38]. SNOPT can be used to solve problems such as the following: Minimize a performance index $J(x)$, subject to constraints on individual state and/or control variables

$$\underline{x}_L < \underline{x} < \underline{x}_U \tag{4.29}$$

constraints defined by linear combinations of state and/or control variables

$$\underline{b}_L < A\underline{x} < \underline{b}_U \tag{4.30}$$

and/or constraints defined by nonlinear functions of state and/or control variables

$$\underline{c}_L < \underline{c}(\underline{x}) < \underline{c}_U \tag{4.31}$$

This SNOPT solver defines the slack variable S, which will transfer the inequality constraints into equality constraints, as follows:

$$l < \begin{pmatrix} \underline{c}(\underline{x}) \\ A\underline{x} \end{pmatrix} < u \Longrightarrow \begin{pmatrix} \underline{c}(\underline{x}) \\ A\underline{x} \end{pmatrix} - S = 0 \quad with \quad l < S < u \tag{4.32}$$

The algorithm of this solver performs two kinds of iterations: major and minor. In major iterations, the slack variables are classified as nonlinear slack variables, S_N, and linear slack variables, S_L, and the nonlinear constraint equations are expanded in a first-order Taylor series as

$$\underline{c}(\underline{x}_k) + \underline{c}'(\underline{x}_k)(\underline{x} - \underline{x}_k) - S_N = 0 \quad \mathit{or} \quad \underline{c}'(\underline{x}_k)\underline{x} - S_N = \underline{c}'(\underline{x}_k)\underline{x}_k - \underline{c}(\underline{x}_k) \tag{4.33}$$

The linear constraints are expressed as

$$A\underline{x} - S_L = 0 \tag{4.34}$$

The two kinds of constraints are combined in one equation as

$$A_t\underline{x} - S = \underline{b} \tag{4.35}$$

where $A_t = \begin{pmatrix} \underline{c}'(\underline{x}_k) \\ A \end{pmatrix}$, $\underline{b} = \begin{pmatrix} \underline{c}'(\underline{x}_k)\underline{x}_k - \underline{c}(\underline{x}_k) \\ 0 \end{pmatrix}$ and $l < S = \begin{pmatrix} S_N \\ S_L \end{pmatrix} < u$. The objective in this major iteration is to minimize

$$q(\underline{x}) = \underline{g}^T\underline{p} + \frac{1}{2}\underline{p}^T H\underline{p} \tag{4.36}$$

subject to the constraints in Eqn.4.35, where $\underline{g} = \nabla_x J(\underline{x})$, p is the search direction vector and H is the Hessian matrix of $J(\underline{x})$. This formulation comes from Newton's Method for Minimization. The approximation of the objective function can be extended in Taylor series at the current $\underline{x}$ point

$$J(\underline{x}^*) = J(\underline{x}) + [\nabla_x J(\underline{x})]^T(\underline{x}^* - \underline{x}) + \frac{1}{2}(\underline{x}^* - \underline{x})^T \nabla^2_{xx} J(\underline{x}^* - \underline{x}) \tag{4.37}$$

Where $\underline{x}^*$ is the optimized point and $\underline{x}^* - \underline{x}$ is the search direction $\underline{p}$. Then Eqn.4.37 is reformulated as

$$J(\underline{x}^*) = J(\underline{x}) + \underline{g}^T \underline{p} + \frac{1}{2}\underline{p}^T H \underline{p} \tag{4.38}$$

To make $\underline{x}^*$ be the minimum point of the quadratic approximate function of J, we must have

$$\frac{\partial J(\underline{x}^*)}{\partial \underline{x}^*} = \underline{g} + H(\underline{x}^* - \underline{x}) = 0 \tag{4.39}$$

which yields the Newton search direction as

$$\underline{p} = \underline{x}^* - \underline{x} = -H^{-1}\underline{g} \tag{4.40}$$

The above procedure doesn't include introducing the states only constraints such as $\underline{c}(\underline{x}) = 0$ with $p \times 1$ elements. In this case, Eqn.4.37 is adjoined to L using Lagrange multipliers $\underline{\lambda}$ $(p \times 1)$

$$L(\underline{x}, \underline{\lambda}) = J(\underline{x}) - \underline{\lambda}^T \underline{c}(\underline{x}) \tag{4.41}$$

The Hessian matrix of the above Lagrangian function L is

$$H_L = \nabla_{xx} L = \nabla_{xx} J - \underline{\lambda}^T \nabla_{xx} \underline{c}(\underline{x}) \tag{4.42}$$

The minimum points generate at

$$\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \underline{g} - G^T \underline{\lambda} \tag{4.43}$$

$$\nabla_\lambda L(\underline{x}^*, \underline{\lambda}^*) = -\underline{c}(\underline{x}) = 0 \tag{4.44}$$

where $G = \nabla_x \underline{c}$. The Taylor series expansions analogous to Eqn.4.37 are constructed as

$$0 = \underline{g} - G^T \underline{\lambda} + H_L(\underline{x}^* - \underline{x}) - G^T(\underline{\lambda}^* - \underline{\lambda}) \tag{4.45}$$

$$0 = -\underline{c}(\underline{x}) - G(\underline{x}^* - \underline{x}) \tag{4.46}$$

Finally, comparing to Eqn.4.39, the solution of the above linear equations, which is called the Karush-Kuhn-Tucker(KKT) system and write

$$\begin{pmatrix} H_L & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} -\underline{p} \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} g \\ c \end{pmatrix} \tag{4.47}$$

In the SQP algorithm, the search direction vector is found so that in the next iteration, the new state variables are updated using

$$x_{k+1} = x_k + \alpha p \tag{4.48}$$

where α is the step size and its value is restricted between $[0, 1]$.

In a minor iteration, Eqn.4.35 is partitioned into

$$Bx_B + Sx_S + Nx_N = b \tag{4.49}$$

where x_B is a basic variable that is to satisfy Eqn.4.35, x_S is a super basic variable chosen to improve the objective function and x_N is a nonbasic variable. The number of x_S indicates degrees of freedom. In each minor iteration, one of x_N is picked up and added to x_S to improve Eqn.4.36 and at the same time satisfying Eqn.4.49, this procedure is repeated until no further improvement can be made.

In summation, the NLP solver is to maximize the objective function and, at the same time, satisfy all constraints.The important point here is how to apply the DC method and the NLP solver settings so that the nonlinear system of equations can be satisfied in the form of constraints added to the solver. From the definition of defect vectors in both of the discretization methods, Eqn.4.2 and Eqn.4.7, it's obvious that those defect vectors are all nonlinear equations and are supposed to be zero when the system equations are satisfied. So, the nonlinear constraints of the SNOPT solver can be set as

$$c_L = c_i(X) = d_i(X) = c_U = 0 \qquad i = 1, \ldots, n \tag{4.50}$$

Then, the dynamic system equations will be defined as nonlinear constraints together with state variable constraints and linear constraints.

4.3 Mistake Prevention

Although the NLP solver have been used successfully to solve many problems, there are some specific things that require attention from user if unexpected mistakes are to be avoided.

First, the constraints added to the optimized problem must yield feasible solutions. This means the solution of each constraint must have a cross section with other constraints. Also, the Jacobian matrix of the constraint must be full rank. Otherwise it will cause the NLP solver to iterate without obvious progress.

Second, redundant constraints must be avoided because they will cause some Lagrange multipliers to converge to zero and make it more the difficult to find the active arcs.

Third, discontinuities encountered in the programming will cause slow convergence or divergence and the KTT necessary conditions are not applicative in such a case. So, in the optimal problem formulation, it is necessary to avoid using IF statement, absolute, min or max functions and linear interpolation of tabular data.

Forth, the independent variables should have similar magnitudes to expedite the convergency rate, improve termination tests and numerical conditioning. Normally, the state and control variables are scaled by using transformations like,

$$x_s(t) = [x(t) - x_N(t)]/S_x(t) \tag{4.51}$$

where S_x is an estimate of an upper bound for $x - x_N$ and x_N is the nominal trajectory. The time factor is scaled by

$$T_s = T/S_T \tag{4.52}$$

where S_T is the estimated upper bound for T

Chapter 5

Using DCNLP To Solve Optimal Control Problems

In this chapter, the DCNLP method is applied to solve two classical calculus of variations problems, the Brachistochrone Problem and Zermelo's Problem. Both these problems have known analytical solutions that can be used to verify the DCNLP method. The detailed problem formulation and SNOPT solver settings are presented. Results using DCNLP is compared with the analytical solutions to verify the accuracy of this method.

5.1 Brachistochrone Problem

The Brachistochrone[40] Problem is that of finding the shape of the curve down which a weight, classically "a bead", acted upon by the force of gravity, will descend from rest and accelerate to a desired point in the least time. If the starting point and ending point lie on the same vertical line, then clearly the shape of the curve is a straight line between the two points. But, if there is a horizontal distance between those two points, the weight will travel in a more complex way. Many researchers have contributed to the solution of this problem. Early in 1697, Newton published his solution, then Leibniz, L'Hospital and Bernoullis found analytical solutions in the form of cycloids.

The problem as formulated mathematically is to minimize t_{12}, the time required for the weight to travel between two points P_1 and P_2 with speed v. We can write

$$t_{12} = \int_{P_1}^{P_2} \frac{ds}{v} \tag{5.1}$$

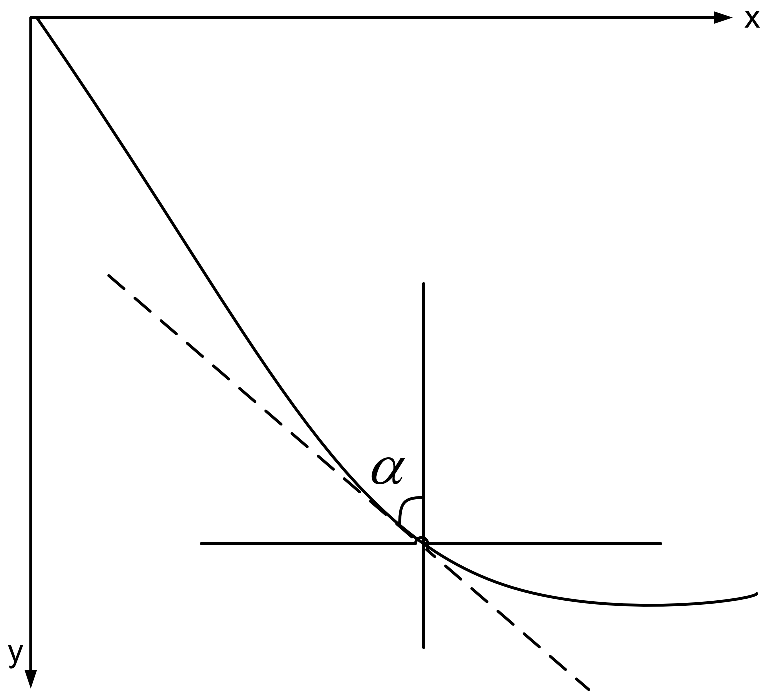

Figure 5.1: Fermat's Principle

According to Fermat's Principle as illustrated in Fig. 5.1, we have

$$v/\sin(\alpha) = k \tag{5.2}$$

where α is the angle between the vertical and the tangent to the curve and k is constant. Under the assumption that energy is conserved $\frac{1}{2}v^2 - gy = const.$, so that if $v = 0$ when $y = 0$,

$$v = \sqrt{2gy} \tag{5.3}$$

By substituting for v in Eqn.5.2, we obtain

$$y = k\mathrm{sin}^2(\alpha) \tag{5.4}$$

then, there is the relationship between y and y'

$$y(1 + {y'}^2) = h \tag{5.5}$$

where $h = 1/2k^2$.

A cycloid satisfies Eqn.5.4 and can be represented as its analytical solution in terms of parameter θ as

$$x(\theta) = h(\theta - \sin\theta) \tag{5.6}$$

$$y(\theta) = h(1 - \cos\theta) \tag{5.7}$$

From the energy equation Eqn.5.3, the minimum time t can be expressed by x, y and their derivatives

$$t_m = \int_{P_1}^{P_2} \sqrt{\frac{1 + y'^2}{2gy}} dx = \int_{P_1}^{P_2} \sqrt{\frac{h}{g}} \frac{dx}{|y|} \tag{5.8}$$

When y is substituted by the analytical solution, the above equation can be further simplified to get

$$t_m = \int_{\theta_0}^{\theta_f} \sqrt{\frac{h}{g}} d\theta = \sqrt{\frac{h}{g}} (\theta_f - \theta_0) \tag{5.9}$$

The DCNLP was applied to this problem. The first step was to set up the first-order derivatives of the system equations. From Eqn.5.3 the energy conservation equation can be reformulated as

$$v^2 = (\dot{x}^2 + \dot{y}^2) = 2g(y_0 - y) \tag{5.10}$$

where y_0 is the initial position of the y axis. The control variable $\tan u$ here is the gradient of the curve

$$\tan u = y' = \frac{\dot{y}}{\dot{x}} \tag{5.11}$$

We can insert $\dot{y} = \dot{x}\tan u$ in Eqn.5.10 and get the system equations as (assuming the weight can only slip in the positive direction of the x axis)

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \sqrt{2g(y_0 - y)}\cos u \\ \sqrt{2g(y_0 - y)}\sin u \end{pmatrix} = f(x, y, u) \tag{5.12}$$

If n nodes are used to represent the trajectory and trapezoidal discretization method is applied, the nonlinear constraints of the SNOPT are

$$c_L(i) = x_{i+1} - x_i - \frac{\Delta t_i}{2}[f(x_i, y_i, u_i) + f(x_{i+1}, y_{i+1}, u_{i+1})] = c_U(i) = 0 \tag{5.13}$$

where $i = 1, 2, \ldots, n$ and $\Delta t_i = \frac{t_f - t_0}{n-1}$. The objective function is

$$J = \sum_{i=1}^{n-1} \Delta t_i = t_f - t_0 \tag{5.14}$$

Table 5.1: Analytical and DCNLP Results for Brachistochrone Problem

initial position	final position	analytical solution	DCNLP solution
$x_0 = 0$, $y_0 = 0$	$x_f = 2$, $y_f = -1$	$t_m = 0.8052$ sec	$t_m = 0.8059$ sec
$x_0 = 0$, $y_0 = 0$	$x_f = 3$, $y_f = -1$	$t_m = 1.0178$ sec	$t_m = 1.0190$ sec
$x_0 = 0$, $y_0 = 0$	$x_f = 5$, $y_f = -1$	$t_m = 1.3870$ sec	$t_m = 1.3890$ sec

In table.5.1, the results of both the analytical and DCNLP method are provided. The minimum time difference is less than 0.13%, which serves to verify the accuracy of the DCNLP method. The major error here comes from the approximate discretization procedure. Also, the trajectories of both methods are illustrated in Fig. 5.2, where the circle line is the analytical solution and the plus sign (+) line stands the DCNLP method.

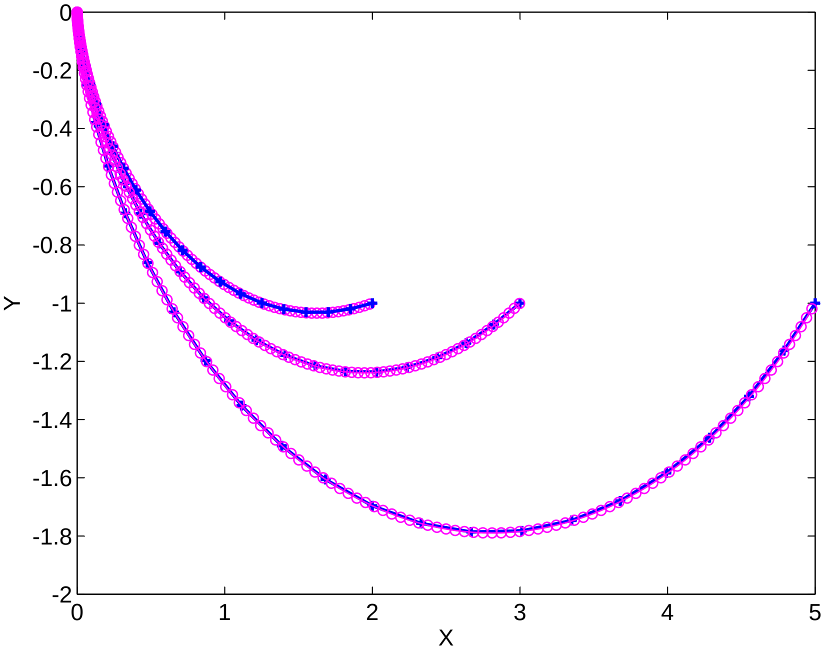

Figure 5.2: Analytical and DCNLP Results for Brachistochrone Problem

Hermite-Simpson method and CP method were also applied to this problem with initial position $x_0 = 0$, $y_0 = 0$ and final potion $x_f = 5$, $y_f = -1$. Together with trapezoidal method, the results for minimum time and computation time are given in Table.5.2 The results shown in Table.5.2 indicate that the minimum time obtained using the CP method is more accurate than that found using the Trapezoidal and Hermite-Simpson method with less computation time. Thus, it appears that the

Table 5.2: Comparison of Brachistochrone Results for Various Collocation Methods

Method	Minimal Time	Absolute Error	Computation Time
Analytical	$t_m = 1.386950$ sec	0	N/A
Trapezoidal	$t_m = 1.389010$ sec	0.002060 sec	5.2375 sec
Hermite-Simpson	$t_m = 1.386895$ sec	0.000055 sec	9.2633 sec
CP	$t_m = 1.386944$ sec	0.000006 sec	6.0587 sec

CP method is superior to the other two methods in terms of accuracy and rate of convergence for same problems. Considering the advantages of CP method, in the following cases only this technique is to calculate the final results.

5.2 Zermelo's Problem

In guidance and control problems, hitting a fixed point in minimum time in an appropriate coordinate is similar to Zermelo's problem. As Zermelo proposed it in 1931[41]: "A ship moves at constant velocity V, relative to water, through currents having constant velocity components p and q in the x and y directions, respectively, of a Cartesian coordinate system. Find the path of minimum time from the origin to a specified point x_f, y_f".

With reference to Fig. 5.3, the first-order differential kinematic equations for the boat's motion are

$$\begin{aligned} \dot{x} &= p + V\cos\theta \\ \dot{y} &= q + V\sin\theta \end{aligned} \tag{5.15}$$

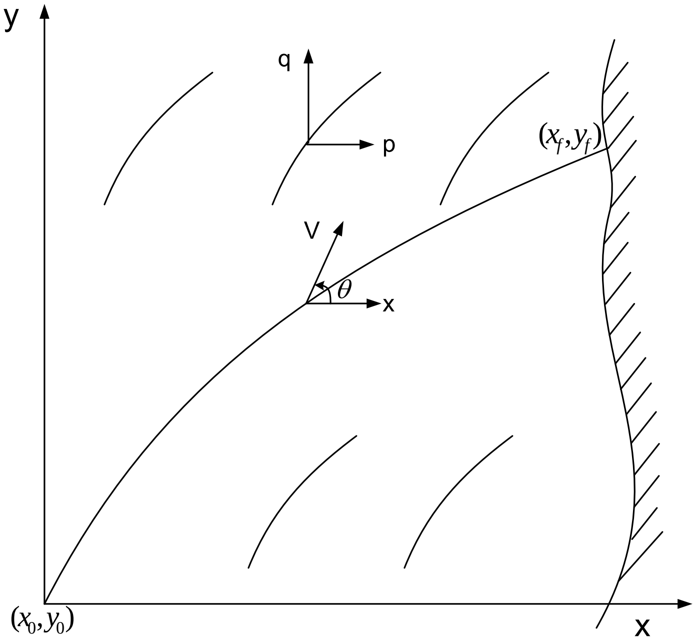

Figure 5.3: Zermelo's Problem

where θ is the angle between the velocity of the boat with respect to the water and the positive x direction and is treated as the only control variable. Powers[42] used the Hamilton-Jacobi method to obtain the following analytical solution for the minimum time

$$t_m = \frac{px_f + qy_f \pm \sqrt{V^2({x_f}^2 + {y_f}^2) - (py_f - qx_f)^2}}{(p^2 + q^2 - V^2)} \tag{5.16}$$

where the minus sign is chosen only when

$$px_f + qy_f > \sqrt{V^2({x_f}^2 + {y_f}^2) - (py_f - qx_f)^2} \tag{5.17}$$

Otherwise, the positive sign will be used here. To obtain this optimal solution, the control variable θ is

$$\theta = \arctan[\frac{y_f - qt_m}{x_f - pt_m}] \tag{5.18}$$

The results from the DCNLP are compared with this analytical results in table.5.3 under same initial condition $x_0 = 0$, $y_0 = 0$, but different final conditions and different velocity of the water.

Table 5.3: Analytical and DCNLP Results for Zermelo's Problem

final position (m, m)	water velocity (m/sec, m/sec)	analytical solution (sec, rad)	DCNLP solution (sec, rad)
$x_f = 100$, $y_f = 100$	$p = 1$, $q = 3$	$t_m = 11.111$, $\theta = 0.644$	$t_m = 11.111$, $\theta = 0.644$
$x_f = 50$, $y_f = 80$	$p = 1$, $q = 3$	$t_m = 7.231$, $\theta = 0.938$	$t_m = 7.231$, $\theta = 0.938$
$x_f = 50$, $y_f = 80$	$p = 3$, $q = 1$	$t_m = 7.712$, $\theta = 1.215$	$t_m = 7.712$, $\theta = 1.215$

The analytical solution and the DCNLP results were identical to three significant figures. These results again verify the accuracy of the DCNLP method in solving at least some optimal control problems.

Chapter 6

Constrained Airspace Minimum Time and Fuel Flight Path

This chapter starts with an introduction of a two-dimensional aircraft model and the solution of MTTC trajectories using energy state method, gradient method and DCNLP method. The horizontal projection of this 2-D MTTC trajectory is calculated and the necessities of considering space constraints are illustrated. Following this the model is expanded to three dimensions and its MTTC and MFTC trajectories under different constrained conditions are listed.

6.1 2-D MTTC Problem

6.1.1 2-D Mathematical Model

The state equations for a two-dimensional, point-mass aircraft model that is commonly used to formulate minimum time problems are[3]

$$\begin{aligned}
\dot{V} &= [(T(M,h) - D(M,h,L))/W - \sin\gamma]g \\
\dot{\gamma} &= (g/V)[L - \cos\gamma] \\
\dot{h} &= V\sin\gamma \\
\dot{x} &= V\cos\gamma
\end{aligned} \tag{6.1}$$

Here, V is the flight speed, γ is the flight path angle, h is the aircraft's altitude, and

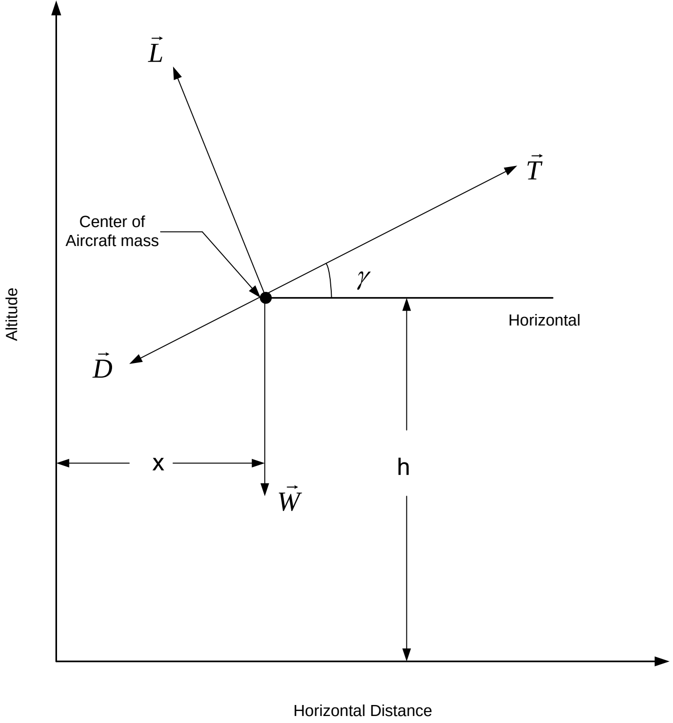

Figure 6.1: 2-D Aircraft Model

x is the horizontal coordinate of the aircraft. Also, T is the magnitude of the thrust, which is assumed to be aligned with the velocity, D is the drag; W is the weight of the aircraft, L is the lift; and M is the flight Mach number. The load factor n is the control variable and is defined as

$$n = L/W \tag{6.2}$$

The lift and drag are modeled using

$$L = \frac{1}{2}\rho V^2 S C_{L_\alpha} \tag{6.3}$$

and

$$D = \frac{1}{2}\rho V^2 S[C_{D_0} + \eta C_{L_\alpha}\alpha^2] \tag{6.4}$$

respectively. By setting $L = nW$, we may replace Eqn.6.4 by

$$D = \frac{1}{2}\rho V^2 S C_{D_0} + \eta \frac{2n^2W^2}{C_{L_\alpha}\rho V^2 S} \tag{6.5}$$

The lift curve slope C_{L_α} and zero lift drag coefficient C_{D_0}, and the drag due to lift factor η, together with the aircraft's weight and wing area S were used to obtain the numerical results given in this paper. The atmospheric density, ρ, is derived from the 1976 U.S. Standard Atmosphere[43]. The thrust magnitude is given by a table in Ref.[2] and reproduced in Appendix A.

One of the conditions identified for special attention in Chapter 4.3 is to avoid using linear interpolation of tabular data, thereby maintaining continuity and differentiability of the system equations and the objective function. Unfortunately, because of the physical constraints, analytical expressions for thrust, lift and drag coefficients are generally unavailable, because propulsion and aerodynamic data from experiments are usually stored in tabular forms. No matter what caused the format of these data, for computational efficiency, it is necessary to devise an alterative way to express these data. Therefore, an approximate model is developed and is used here to reproduce the thrust and aerodynamic properties. At a fixed March number, the thrust can be expressed as a polynomial function $T(h)$ of degree n that fits the data $T(h_i)$, h_i with least square error.

$$T(h) = p_n h^n + p_{n-1} h^{n-1} + \ldots + p_1 h + p_0 \tag{6.6}$$

Here, p_i, $i = 0, \ldots, n$ is the polynomial coefficients of the thrust function. If a series of such polynomial functions is given for different Mach numbers, then these p_i can be written as a function of Mach number $p_i(M)$. The polynomial function of degree m is used again to approximately fit the variation of these coefficients with Mach number

$$p_i(M) = q_m M^n + q_{m-1} M^{n-1} + \ldots + q_1 M + q_0 \tag{6.7}$$

In the thrust table.A.1, consider only the portion for which $0 \leq h \leq 80k$ ft and $0 \leq M \leq 2$. Using a second-order polynomial to fit the function $T(h)$, the coefficients at different March number are listed in table.6.1. From the table, the coefficients data is more irregular than the thrust function of altitude, so a higher degree of polynomial approximation function, degree four, is used here to approximate the coefficients as

Table 6.1: Polynomial Coefficients at Different March Number

M	0	0.4	0.8	1.2	1.6	2.0
p_2	0.0045	0.0041	0.0038	0.0040	0.0019	-0.0033
p_1	-0.6523	-0.6101	-0.6085	-0.6830	-0.5367	-0.3551
p_0	23.5739	22.6811	24.7042	30.1086	32.0151	32.0435

functions of Mach number. They are in the form of as

$$p_2(M) = 0.0014M^4 - 0.0070M^3 + 0.0091M^2 - 0.0039M + 0.0045 \tag{6.8}$$

$$p_1(M) = -0.1221M^4 + 0.6723M^3 - 1.0093M^2 + 0.4581M - 0.6556 \tag{6.9}$$

$$p_0(M) = 0.9393M^4 - 10.2110M^3 + 24.4391M^2 - 11.3770M + 23.6367 \tag{6.10}$$

Finally, the thrust functions variation with altitude and Mach number is expressed as

$$\begin{aligned} T(h, M) = {} & (0.0014M^4 - 0.0070M^3 + 0.0091M^2 - 0.0039M + 0.0045)h^2 + \\ & (-0.1221M^4 + 0.6723M^3 - 1.0093M^2 + 0.4581M - 0.6556)h + \\ & (0.9393M^4 - 10.2110M^3 + 24.4391M^2 - 11.3770M + 23.6367) \end{aligned} \tag{6.11}$$

Using this same method, the lift and drag coefficients as a function of Mach number can also be expressed as polynomial functions of degree 4.

$$C_{D_0}(M) = 0.0010M^4 - 0.0113M^3 + 0.0257M^2 - 0.0130M + 0.0066 \tag{6.12}$$

$$C_{L_\alpha}(M) = 0.0122M^4 - 0.0568M^3 - 0.1500M^2 + 0.2847M + 2.2397 \tag{6.13}$$

6.1.2 Energy State Method

In Eqn.6.1, the state variable V can be replaced mathematically by the total energy per unit mass, which is defined as

$$E = \frac{1}{2}V^2 + gh \tag{6.14}$$

and then

$$\dot{E} = V\dot{V} + g\dot{h} \tag{6.15}$$

Using $\dot{V}$ and $\dot{h}$ expressions from Eqn.6.1 to rewrite $\dot{E}$, we find

$$\dot{E} = V(T - D)/m \tag{6.16}$$

Assume $L \simeq W$, then the time integration can be expressed as

$$t_f - t_0 = \int \frac{m}{V(T - D)} dE \tag{6.17}$$

In order to minimize time interval $t_f - t_0$, it is obvious to maximize the denominator of the right part of Eqn.6.17 at a given energy value E. We already know that thrust T and drag D is a function about V and E, then the objective is to find the velocity V that will maximize excess power, $V(T(V, E) - D(V, E))$, at a fixed E.

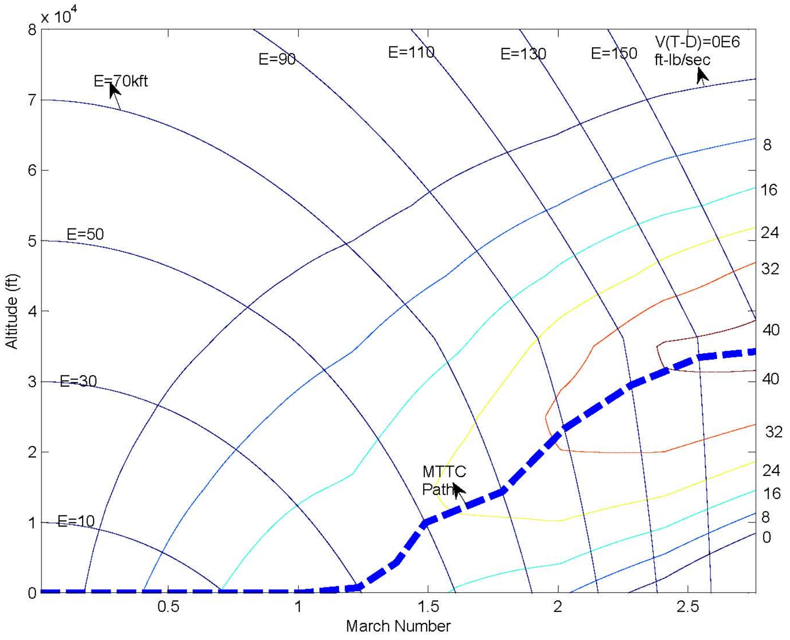

Figure 6.2: 2-D MTTC Trajectory Using Energy State Model

Assume the aircraft is going to takeoff from sea level to an altitude of $80k$ ft and the contours of constant E are plotted on a $h-V$ plane. According to the above theory, these numerous points of the maximum $V(T(V,E)-D(V,E))$ at different contours constitute the approximate MTTC trajectory which is shown in Fig. 6.2.

In Fig. 6.2, the contours of both the constant energy and constant excess power are illustrated. The dash line is the MTTC energy path which consists of an acceleration of flight at sea level and a climb after the aircraft gains the speed over march number. Then the gradient method is applied to this problem to see the difference from the trajectory obtained above.

6.1.3 Gradient Method

The gradient method for this 2-D MTTC problem requires to derive the adjoint equations for the system first. With introduction a 3×1 adjoint vector $[\lambda_V, \lambda_\gamma, \lambda_h]^T$, the Hamiltonian equation is written as

$$H = 1 + \lambda_V[(T - D)/W - \sin\gamma]g + \lambda_\gamma[(g/V)(n - \cos\gamma)] + \lambda_h V \sin\gamma \tag{6.18}$$

and the derivative of the adjoint equations are

$$\dot{\lambda_V} = -\frac{\partial H}{\partial V} = -\lambda_V\left(\frac{\partial T}{\partial V} - \frac{\partial D}{\partial V}\right)\frac{g}{W} + \lambda_\gamma\frac{g}{V^2}(n - \cos\gamma) - \lambda_h \sin\gamma \tag{6.19}$$

$$\dot{\lambda_\gamma} = -\frac{\partial H}{\partial \gamma} = \lambda_V g\cos\gamma - \lambda_\gamma\frac{g}{W}\sin\gamma - \lambda_h V\cos\gamma \tag{6.20}$$

$$\dot{\lambda_h} = -\frac{\partial H}{\partial h} = -\lambda_V\frac{g}{W}\frac{\partial T}{\partial h} \tag{6.21}$$

with the initial and final boundary conditions

$$\begin{aligned} V(0) &= V_0, & h(t_f) &= h_f \\ \gamma(0) &= \gamma_0, & \lambda_\Phi(t_f) &= [0 \quad 0 \quad 0]' \\ h(0) &= h_0, & \lambda_\Omega(t_f) &= [0 \quad 0 \quad 1]' \end{aligned} \tag{6.22}$$

Here λ_Φ and λ_Ω are Lagrange multipliers, their definition and the detailed procedure of the gradient method are given in Appendix B. In using the gradient method, it starts from an initial guess of control variables and final time to get the nominal path and then gets the corrections of the control variables according to the procedure in Appendix B. This process is repeated until the convergence condition is reached. As an example, in the 2-D MTTC problem, assume that the aircraft starts at at sea level with initial and final conditions,

$$\begin{aligned} V(0) &= 558.2 ft/sec, \qquad h(0) = 0 \\ \gamma(0) &= 0, \qquad\qquad h(t_f) = 80kft \end{aligned} \tag{6.23}$$

The vehicle's MTTC trajectory is shown in Fig. 6.3 and the time histories of the load factor, velocity, flight path angle and altitude are shown in Fig. 6.4 separately. The minimum time for the aircraft fly from sea level to an altitude of $80k$ ft is approximately 180 sec. Compared to the energy state method, the results found from the gradient method is more accurate and the state and control variable histories reflect the real values instead the approximate values used in previous method. But, this method's successful implement depends on an adequate guess of the initial control variables. This means that a large deviation of the initial guess will cause divergence. Furthermore, the whole gradient calculation procedure is a burdensome task. However, these difficulties can be avoided in the DCNLP method.

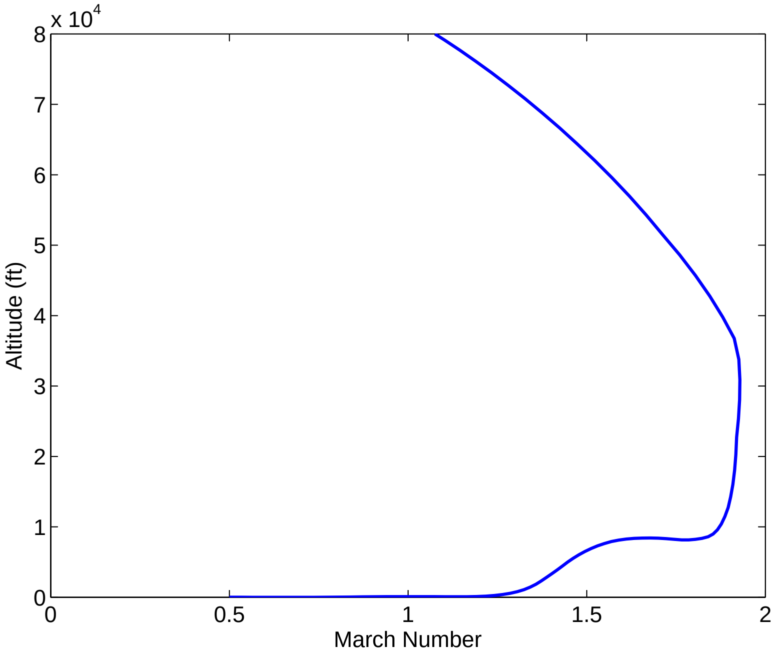

Figure 6.3: 2-D MTTC Trajectory Using Gradient Method

6.1.4 DCNLP Method

The DCNLP method that is applied in this sub-section to solve a 2-D MTTC problem uses CP discretization method with 18 collocation points. The objective is to minimize t_f subject to the equality constraints

$$\begin{aligned} \tfrac{2}{t_f} D\underline{V} &= [(\underline{T} - \underline{D})/W - \sin\underline{\gamma}]g \\ \tfrac{2}{t_f} D\underline{\gamma} &= (g/\underline{V})[\underline{n} - \cos\underline{\gamma}] \\ \tfrac{2}{t_f} D\underline{h} &= \underline{V}\sin\underline{\gamma} \end{aligned} \tag{6.24}$$

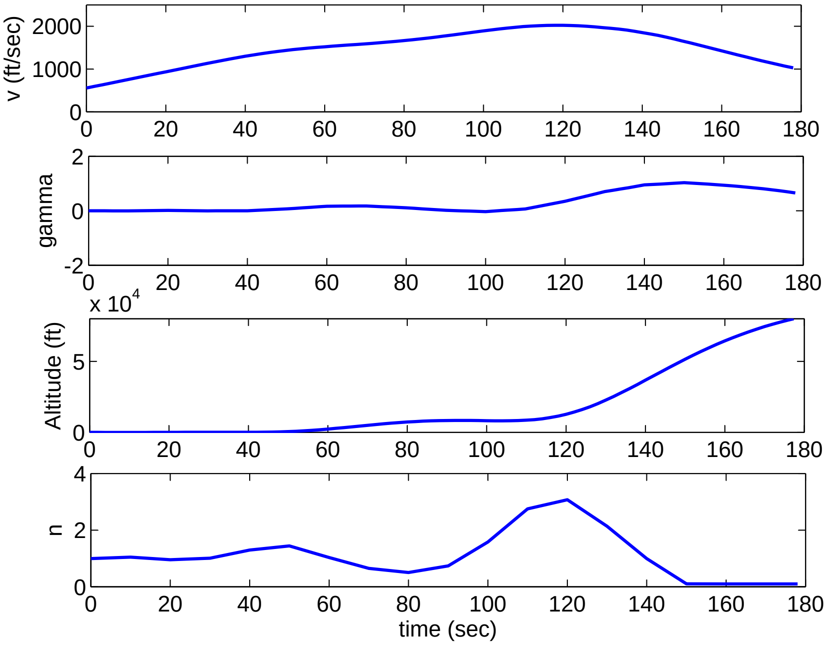

Figure 6.4: 2-D MTTC State and Control Variables History Using Gradient Method

Here $\underline{V}$, $\underline{\gamma}$, $\underline{h}$, $\underline{T}$, $\underline{D}$ and $\underline{n}$ represent the solution vectors to this problem at the collocation points. The initial and final conditions are the same as in the gradient method examples. The minimum time obtained here is 178.25 sec with boundary constraints and system equality constraints well satisfied. The MTTC trajectory using this method is shown in Fig. 6.5 and the time history of load factor, velocity, flight path angle and altitude are shown in Fig. 6.6 separately.

The results using DCNLP method are very close to the results obtained using the gradient method, but the initial guesses for all the collocation points is more arbitrary. Also, the convergence of the solution is much more rapid.

Considering the complexity of 3-D aircraft model which will includes more state variables and control variables, it will be more difficult to obtain the adjoint derivative equations and guess adequate nominal control variables. So the DCNLP method which will avoid deriving adjoint equations and guessing initial values is superior than other two methods in solving the following 3-D problems.

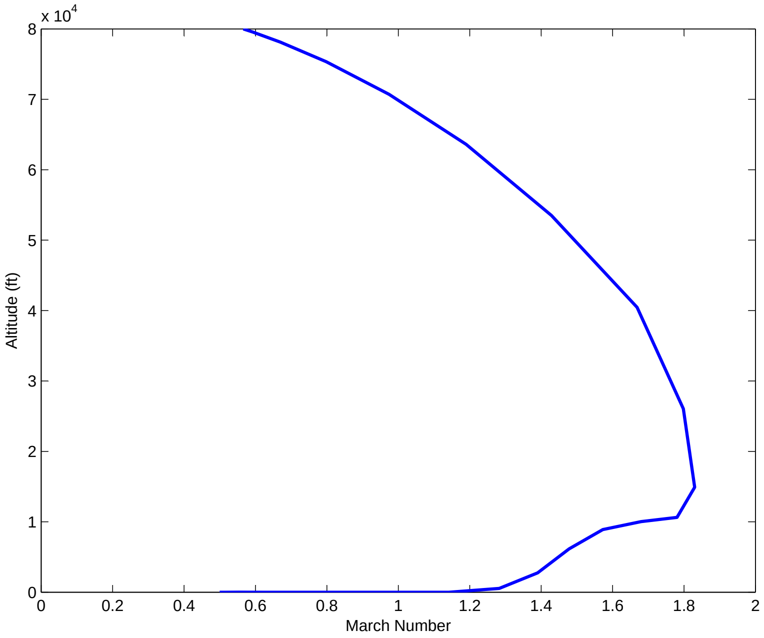

Figure 6.5: 2-D MTTC Trajectory Using DCNLP Method

In Fig. 6.7, the trajectory of an ”Altitude-Down Range” plot is illustrated. From this figure, we see that the footprint required for an aircraft with modest performance to climb from sea level to the desired altitude of $80k$ ft, the down range distance will be more than half of its altitude. This is because most of the time the aircraft’s flight path

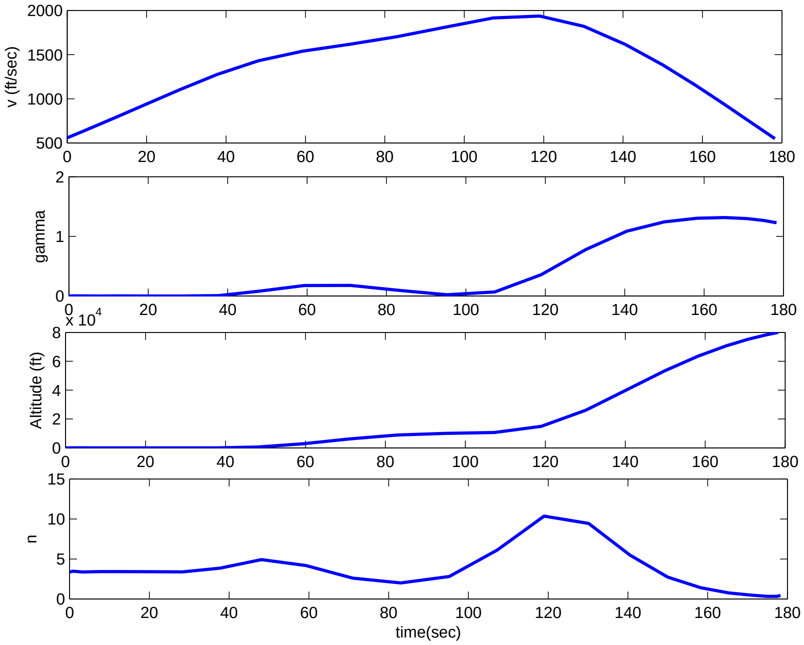

Figure 6.6: 2-D MTTC State and Control Variables History Using DCNLP Method

angle is less than $\pi/3$ which makes the horizontal projection of the trajectory longer than half of its magnitude of the altitude. While for some cases, such long horizontal distance is not available because of local terrains constraints, so it's necessary for the aircraft to make some turns to avoid the collision. Thus, the 3-D aircraft model is considered and some maneuvers are conducted based on this new model.

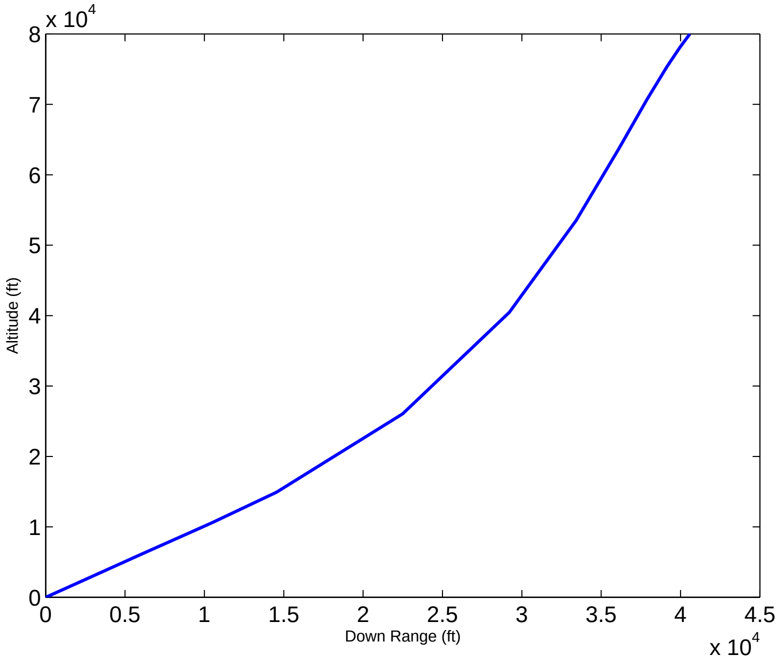

Figure 6.7: 2-D MTTC Altitude and Down Range Trajectory Using DCNLP Method

6.2 3-D MTTC Problem

6.2.1 3-D Mathematical Model

Unlike the 2-D aircraft model, which focuses on the control in the longitudinal plane only, the 3-D aircraft model also considers the control in lateral plane. So the previously used control variable is replaced with two controls, a vertical load factor, n_V, and a horizontal load factor, n_h. The heading angle, χ, and cross range, y, as additional state variables are included in the system functions. Then, the system equations are also expanded to six first-order differential equations,

$$
\begin{aligned}
\dot{V} &= [(T(M,h) - D(M,h,n))/W - \sin\gamma]g \\
\dot{\gamma} &= (g/V)[n_V - \cos\gamma] \\
\dot{\chi} &= (g/V)[n_h/cos\gamma] \\
\dot{h} &= V\sin\gamma \\
\dot{x} &= V\cos\gamma\cos\chi \\
\dot{y} &= V\cos\gamma\sin\chi
\end{aligned}
\tag{6.25}
$$

The resultant load factor n is given by

$$n = \sqrt{n_V^2 + n_h^2} \tag{6.26}$$

and the aircraft bank angle, μ, is given by

$$\mu = \tan^{-1}(n_h/n_V) \tag{6.27}$$

Together with other variables, μ is illustrated in Fig. 6.8. The other properties, such as thrust, aerodynamic data and air density are treated the same as in the 2-D model.

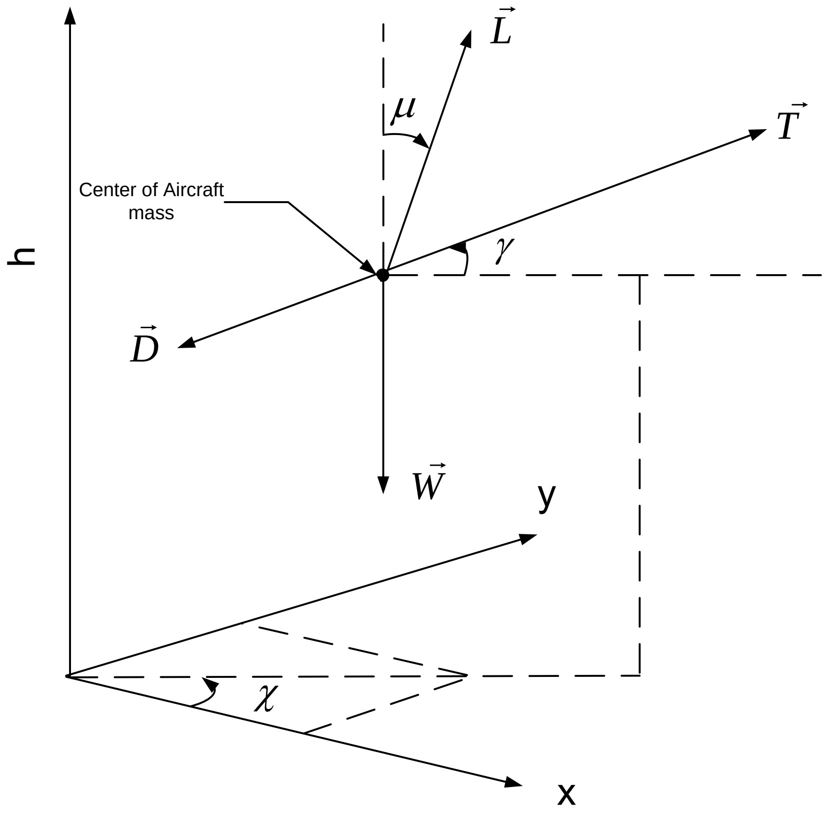

Figure 6.8: 3-D Aircraft Model

6.2.2 Initial NLP Variable Inputs

Although in most cases the choice of initial input of the NLP variables are arbitrary, a good guess generally improves the rate of convergence and avoids singularities of the Jacobian matrix. Assuming that the aircraft is climbing inside a cylinder with square projection in x-y plane, in order to allow for a large enough horizontal flight distance, the final trajectory will make good use of the constrained airspace as much as possible. An initial guess of the path will probably be similar to a helix curve wrapped on the a right-circular cylinder of radius R inscribed in the square cylinder, as shown in Fig. 6.9.

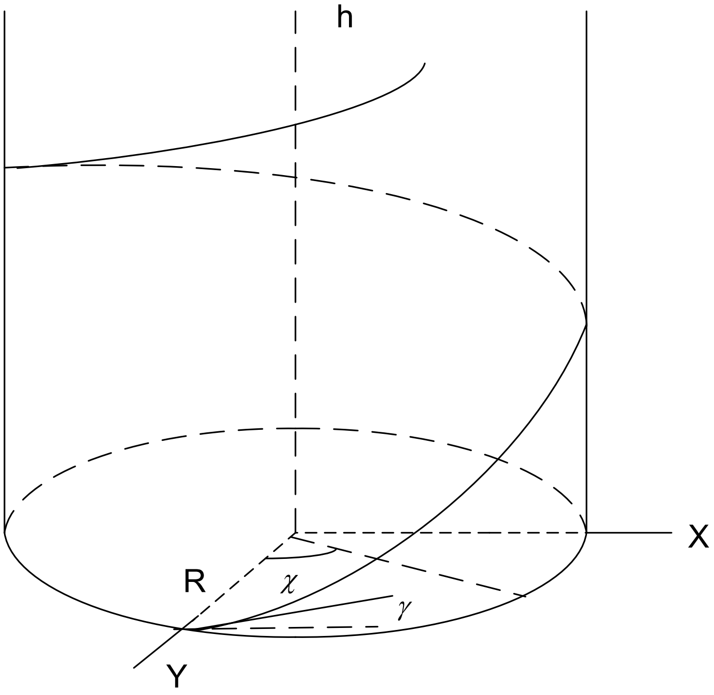

Figure 6.9: Circular Helix Curve Wrapped on A Cylinder

If the aircraft flies the helix curve with a constant speed and inclination angle. Then, the system equations for the initial trajectory can be simplified to

$$\begin{gathered}
\dot{V} = 0 \Longrightarrow V = V_I \\
\dot{\gamma} = 0 \Longrightarrow \gamma = \gamma_I \\
\dot{\chi} = V_I \cos\gamma / R \\
\dot{h} = V \sin\gamma \Longrightarrow h = V_I t \sin\gamma \\
x = R \sin\gamma \\
y = R \cos\gamma
\end{gathered} \tag{6.28}$$

where V_I and γ_I are the pre-assumed constant velocity and constant flight path angle of the aircraft. Normally, they are assumed to be the take-off velocity and flight path angle separately. If the time intervals are chosen as Chebyshev points, then, all the initial NLP variables can be fixed according to Eqn.6.28.

6.2.3 Results

The system equation constraints using the CP discretization method in this 3-D MTTC problem are formulated as

$$\begin{aligned}\tfrac{2}{t_f}D\underline{V} &= [(\underline{T}-\underline{D})/W-\sin\underline{\gamma}]g\\ \tfrac{2}{t_f}D\underline{\gamma} &= (g/\underline{V})[\underline{n}_V-\cos\underline{\gamma}]\\ \tfrac{2}{t_f}D\underline{\chi} &= (g/\underline{V})[\underline{n}_h/\cos\underline{\gamma}]\\ \tfrac{2}{t_f}D\underline{h} &= \underline{V}\sin\underline{\gamma}\\ \tfrac{2}{t_f}D\underline{x} &= \underline{V}\cos\underline{\gamma}\cos\underline{\chi}\\ \tfrac{2}{t_f}D\underline{y} &= \underline{V}\cos\underline{\gamma}\sin\underline{\chi}\end{aligned} \tag{6.29}$$

with the boundary constraints and performance limitations listed in table.6.2

Table 6.2: Boundary Constraints and Performance Limitations for 3-D MTTC Problem

Constraints	Values
Initial Coordinate Constraints	$x = x_0$, $y = y_0$, $h = 0$
Initial V, γ and χ Constraints	$V = V_0$, $\gamma = \gamma_0$, $\chi = \chi_0$
Final Altitude	$h = h_f$
Airspace Constraints	$x_{min} \le x \le x_{max}$, $y_{min} \le y \le y_{max}$
Maximum and Minimum Vertical Load Factor	$n_{V max} = 10$, $n_{V min} = -10$
Maximum and Minimum Horizontal Load Factor	$n_{hmax} = 10$, $n_{hmin} = -10$

The system equation constraints are treated as nonlinear constraints in the NLP solver and the boundary constraints and performance limitations are set as NLP variable constraints in the solver. The objective function is the time interval between initial time t_0 and final time t_f, which is free.

Case 1: Set the airspace constraints on x and y as $\pm 10,000$ ft. The airplane is required to fly from sea level to an altitude of $h_f = 30,000ft$ with an initial flight speed of $Mach = 0.5$, an initial flight path angle of $\gamma_0 = 12.6°$, and an initial heading angle of $\chi_0 = 0$. The minimum time obtained for this case is 89.76 sec. The three-dimensional trajectory is shown in Fig. 6.10 and the time histories of vertical and horizontal load factors n_V and n_h, velocity V, flight path angle γ and heading angle χ are shown in Fig. 6.11.

Case 2: Using the same initial conditions as in Case 1, but with the constraints on both x and y of $\pm 7,500$ ft, the aircraft cannot reach 30,000 ft within a time similar to that in case 1. However, when the final altitude was reduced to $25,000$ ft, the three-dimensional MTTC trajectory shown in Fig. 6.12 with a similar climb time of 91.49 sec was obtained. The corresponding state and control variable time histories are shown in Fig. 6.13.

Case 3: When the final 30,000 ft altitude was reached under the same initial condition of Case 1 and with the constraints on both x and y of $\pm 7,500$ ft, the minimum time calculated here is 125.24 sec. The trajectory, corresponding state and control variable time histories are shown in Fig. 6.14 and Fig. 6.15 respectively.

Case 4: When the final flight speed constraint of $M = 0.8$ is added to Case 1, the climb time is increased to 156.01 sec with other initial and final conditions unchanged. The trajectory, corresponding state and control variable time histories are shown in Fig. 6.16 and Fig. 6.17, respectively.

Case 5: When the same final flight speed constraint $M = 0.8$ is added to Case 3 which has the smaller airspace horizontal projection, the climb time under this condition is 175.28 sec. The trajectory, corresponding state and control variable time histories are shown in Fig. 6.18 and Fig. 6.19 respectively.

6.2.4 Conclusions

Using the CP discretization method and the NLP solver, the 3-D MTTC problem has been solved under different assumption conditions. The results show that the performance index, the time to climb, has been determined accurately and system equality constraints, boundary constraints and control constraints were satisfied.

In all the five cases, the final trajectory is similar to a helical curve wrapped on a cylinder, which indicates that the initial guess of a helical trajectory is reasonable. Cases 1 and 2 have similar climb times, but in Case 1, a higher altitude was reached. The only difference between the cases is that Case 2 had a smaller constrained space in terms of the $x - y$ base. As expected, smaller airspace volumes result in more time for the aircraft to climb to a desired altitude.

Case 3 had the same airspace constraints as Case 2, but 20% higher final altitude. The extra altitude in Case 2 cost 36.9% more in terms of climb time. These data indicate that the climb time is not proportional to the altitude. It can also be said that the time distribution is different at different altitude.

The optimal trajectory in each case can be treated as a process of "climb-dive-climb". The aircraft stays at sea level initially to gain speed and then makes a fast climb. In some cases, the aircraft's speed decreases to close to zero before the aircraft reaches its final altitude. To complete the climb task, it's necessary for the aircraft to make a dive to gain speed and then climb again. Cases 3 to 5 illustrate this characteristic. When higher altitudes are required, more dives and climbs may be performed. That's why the trajectory is a repeat process of "climb-dive-climb". This property can also be shown in the 2-D MTTC problem, when higher final altitudes are required.

Another characteristic property of the trajectory is the final speed. If there is no constraint on the final speed, then the aircraft speed will normally decrease close to stall speed. This is not realistic. When a constraint on the final flight speed is added, the stall point can be avoided subject to the penalty of more climb time. This can be seen by comparing climb times for Cases 1 and 4 or Cases 3 and 5.

6.3 3-D Minimum-Fuel-To-Climb Problem

In the 3-D MTTC problem, the aircraft weight is treated as a constant. Actually, the consumption of fuel during this time interval makes the weight of the aircraft a function of time. The change of the weight is omitted in above discussion because the climb task is completed in a few minutes, which makes this change of weight negligible. If this weight change is modeled, then the additional equation,

$$\dot{m} = -f = T/cg \tag{6.30}$$

where $c = 2800$ sec and f is the thrust specific fuel consumption, is included and the mass is an additional state variable to the pervious 6×1 state variable vector. And the previous system equations keep the same except the weight is substituted as a function of time

$$W(t) = m(t)g \tag{6.31}$$

The objective function now is to maximize the final weight

$$J = -W_f \tag{6.32}$$

Some cases are tested with same boundary constraints and performance limitations set as in previous MTTC problem.

Case 6: Use the same initial and final conditions and airspace constraints in Case 1 with unspecified climb time. The MFTC trajectory spent similar time, 93.13 sec, comparing to Case 1. The trajectory, corresponding state and control variable time histories are shown in Fig. 6.20 and Fig. 6.21 respectively. The trajectory in Case 6 is similar to the trajectory in Case 1 and the mass of the aircraft changes in a small scale which can be negligible. In order to obtain the objective of minimum fuel consumption in the climb procedure, it is expected that the aircraft can reach the final altitude in the least time to consume as little as possible fuel. From this point of view, it is reasonable that the MTTC trajectory and MFTC trajectory is close to each other.

Case 7: Use the same initial and final conditions, airspace constraints as in Case 3 with unspecified climb time. The MFTC trajectory also spent similar time 140.54 sec comparing to Case 3. The trajectory, corresponding state and control variable time histories are shown in Fig. 6.22 and Fig. 6.23 respectively. When the airspace constraints became smaller, it will cost more fuel to climb to the same altitude, the same conclusion as the MTTC trajectories in different airspace constraints.

Case 8: When the same final flight speed constraint $M = 0.8$ is added to case 6 with final time still unspecified, the climb time under this condition is 175.28 sec. The trajectory, corresponding state and control variable time histories are shown in Fig. 6.24 and Fig. 6.25 respectively. Also as in corresponding MTTC conclusion, final speed constraints cost more time and fuel consumption comparing to none final speed constraints.

Case 9: All the above MFTC cases consider only unspecified final time, the NLP solver will choose the optimized time to get the minimum fuel results. When the final time is specified as 100 sec in Case 6, the consumption of fuel is more than previous result. The MFTC trajectory, corresponding state and control variable time histories are shown in Fig. 6.26 and Fig. 6.27 respectively. Any specified time larger than the time got from the unspecified final time result will cost more fuel consumption.

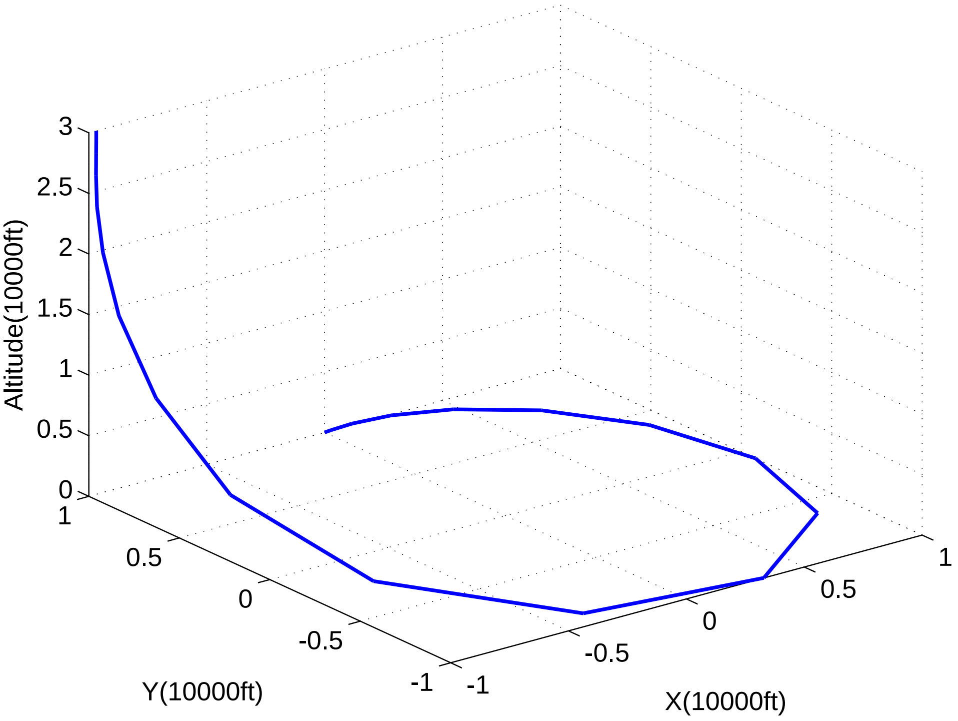

Figure 6.10: 3-D MTTC Trajectory for Case 1

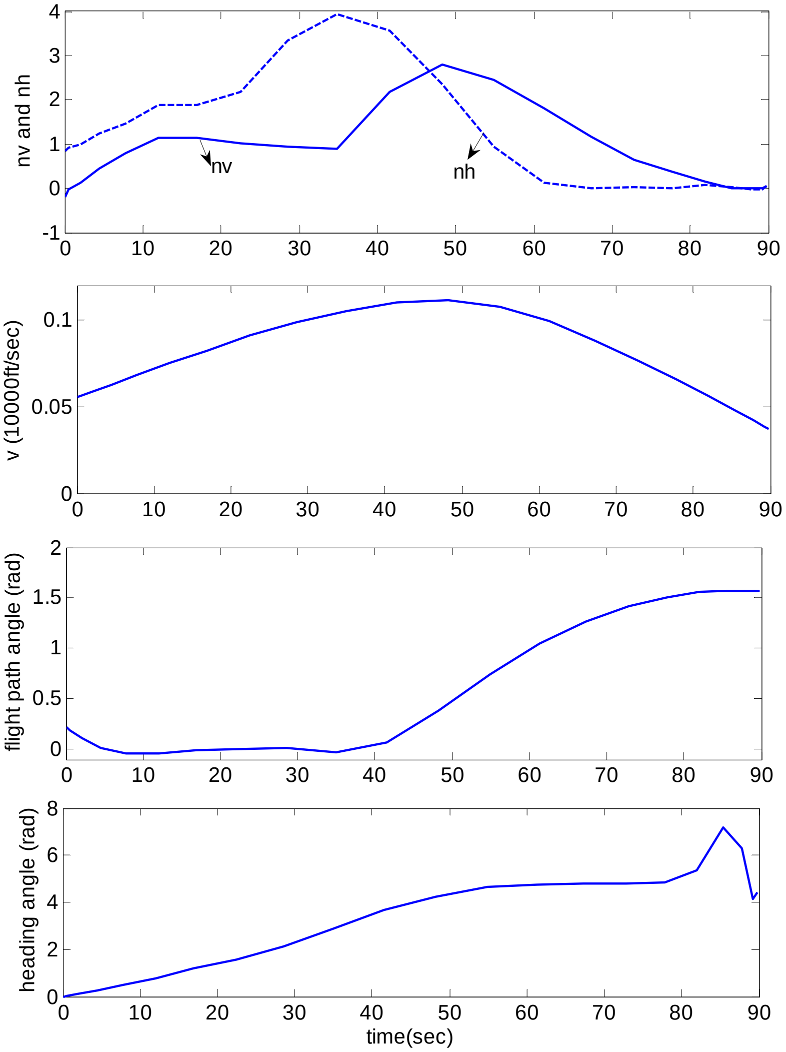

Figure 6.11: Control and State Variables History for Case 1

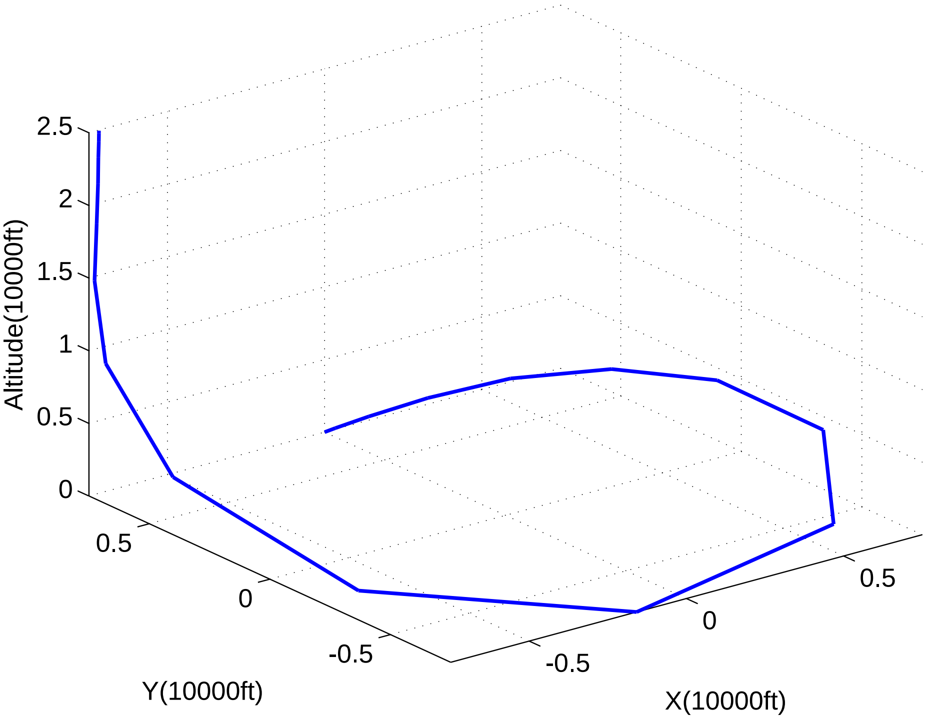

Figure 6.12: 3-D MTTC Trajectory for Case 2

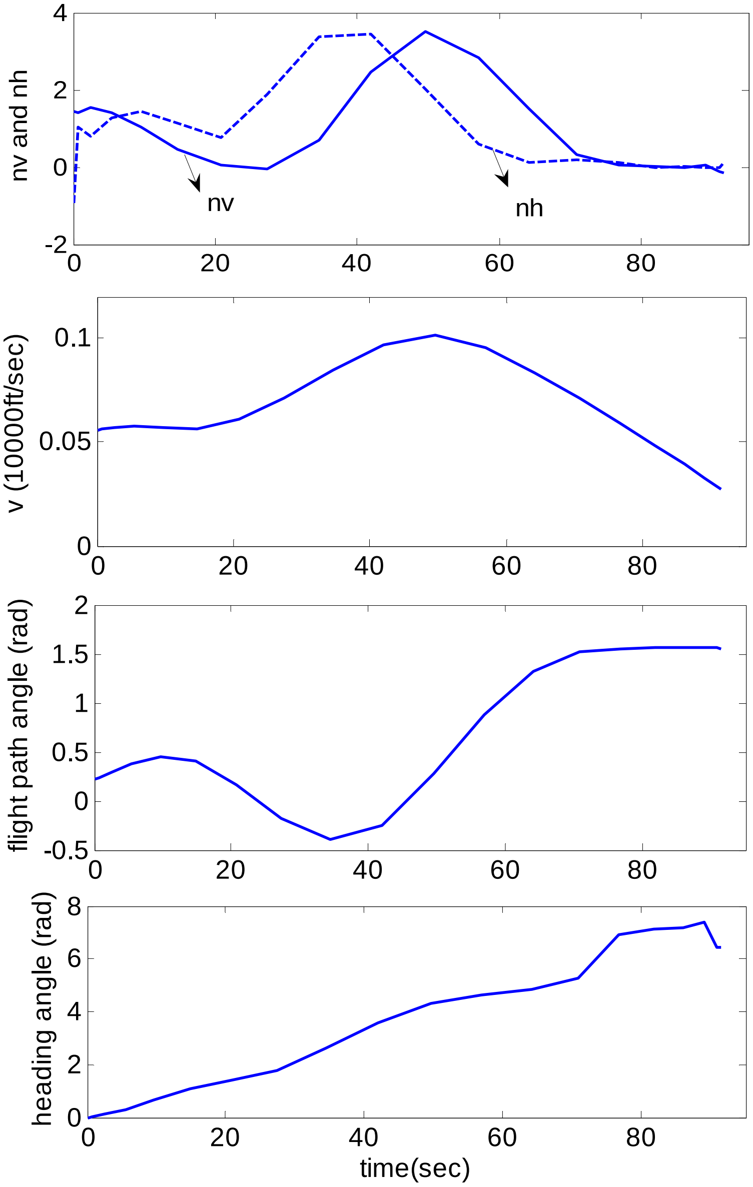

Figure 6.13: Control and State Variables History for Case 2

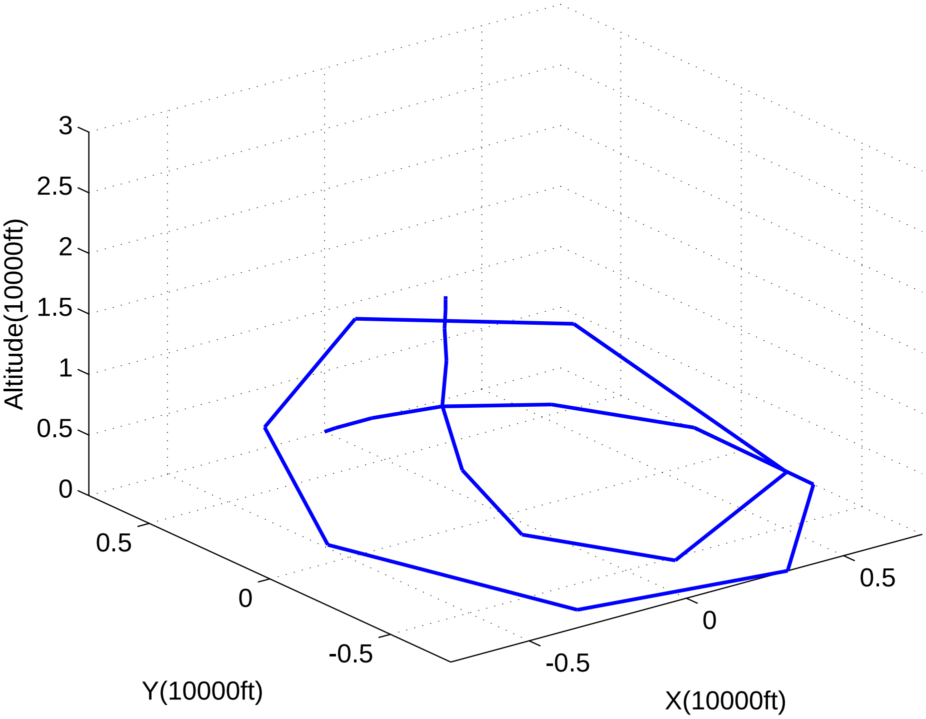

Figure 6.14: 3-D MTTC Trajectory for Case 3

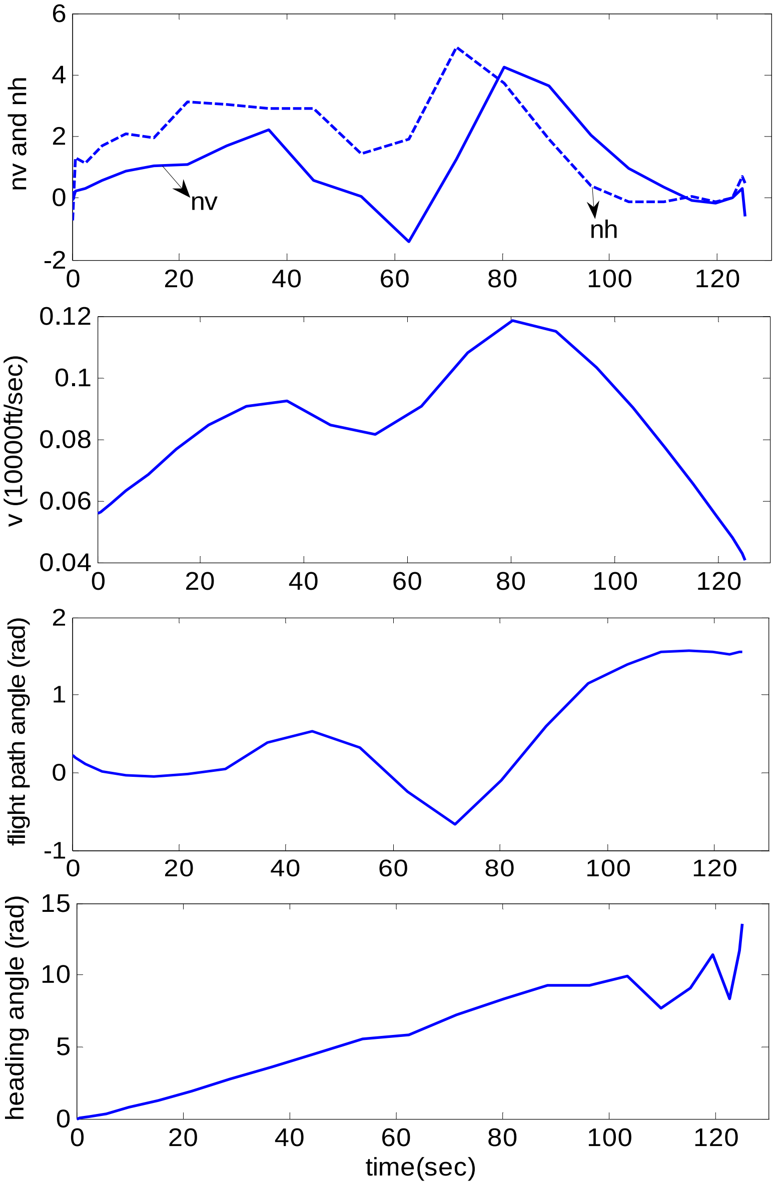

Figure 6.15: Control and State Variables History for Case 3

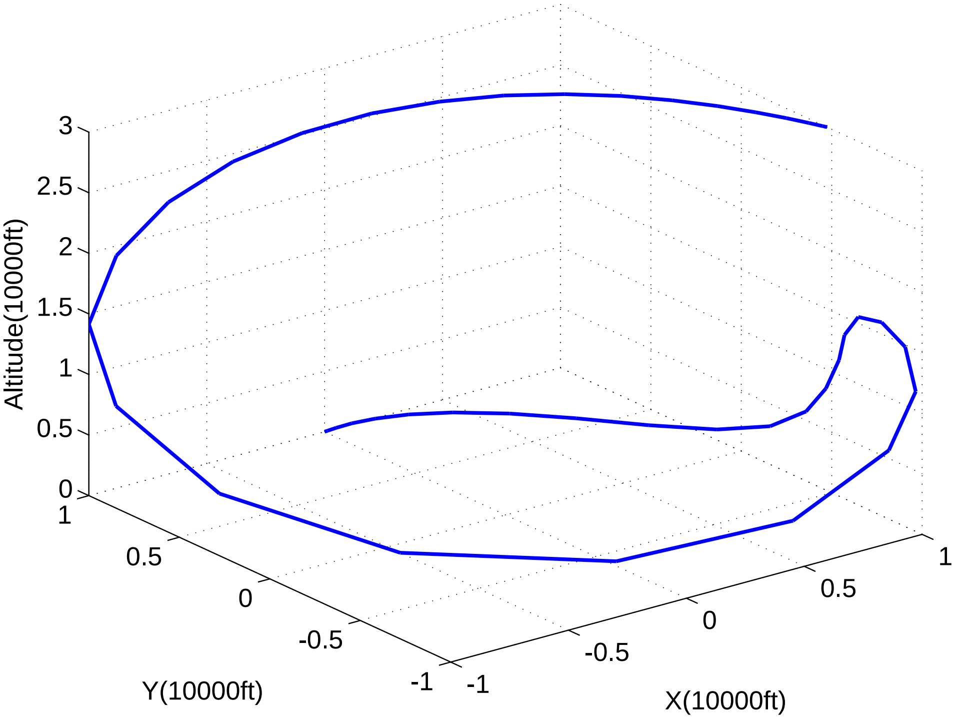

Figure 6.16: 3-D MTTC Trajectory for Case 4

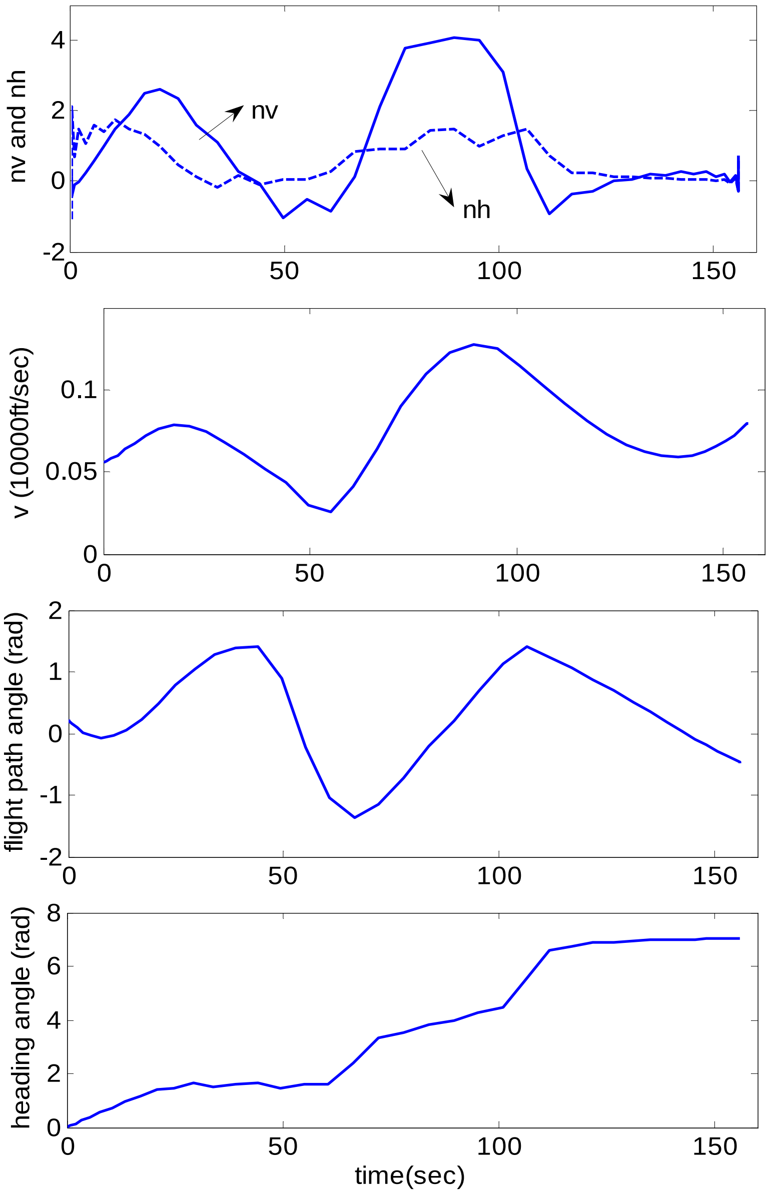

Figure 6.17: Control and State Variables History for Case 4

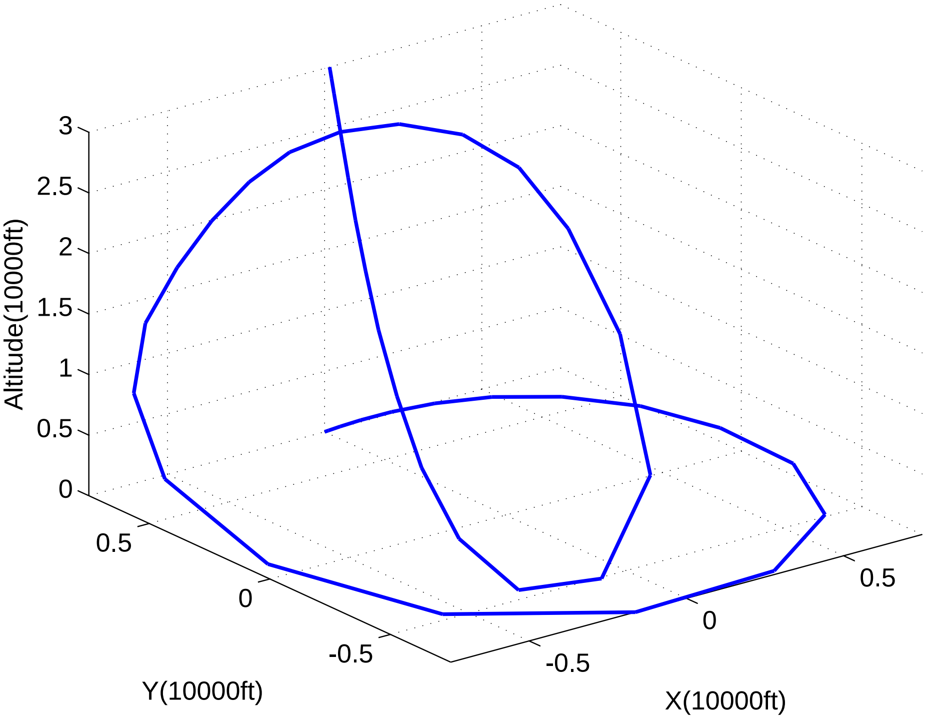

Figure 6.18: 3-D MTTC Trajectory for Case 5

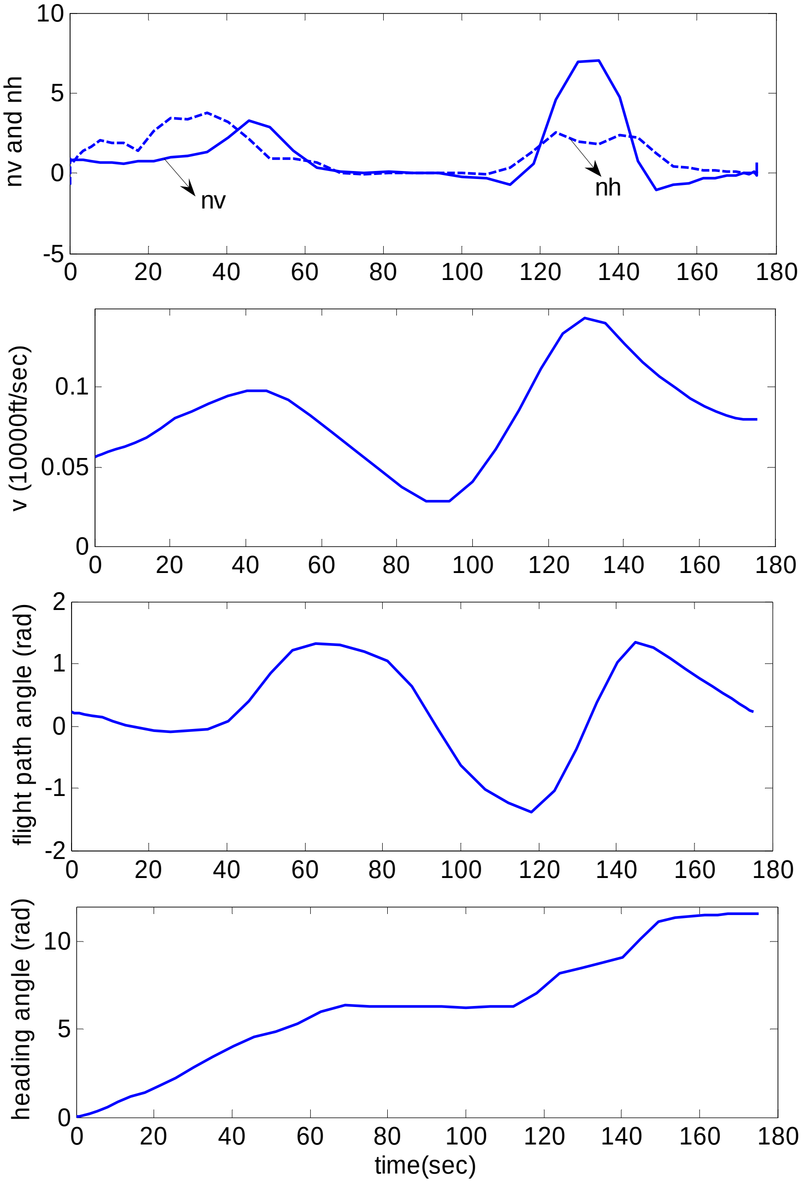

Figure 6.19: Control and State Variables History for Case 5

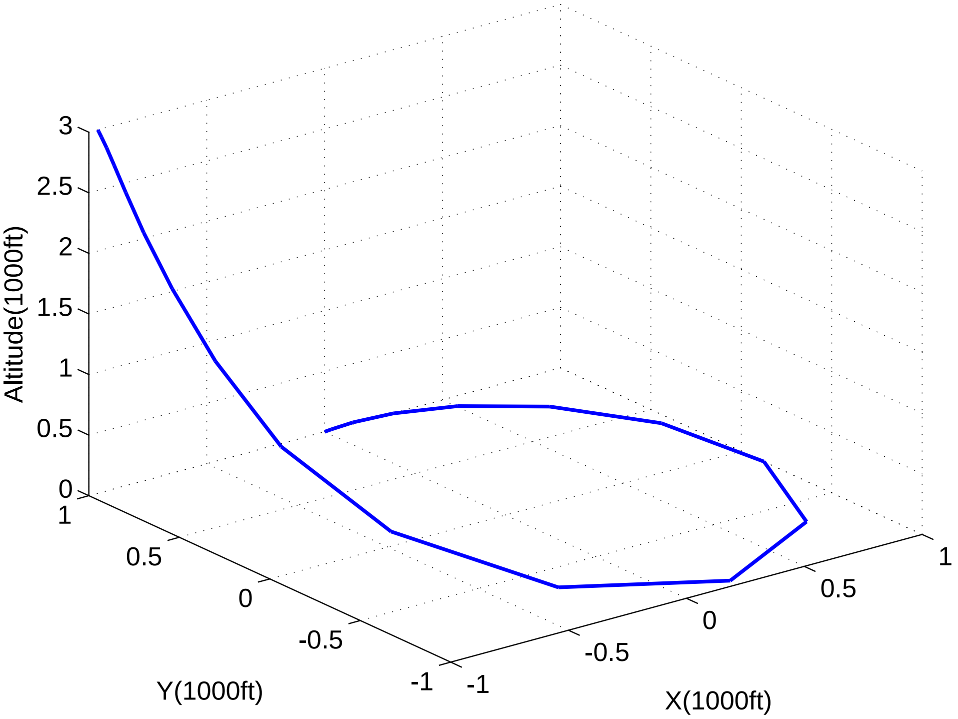

Figure 6.20: 3-D MFTC Trajectory for Case 6

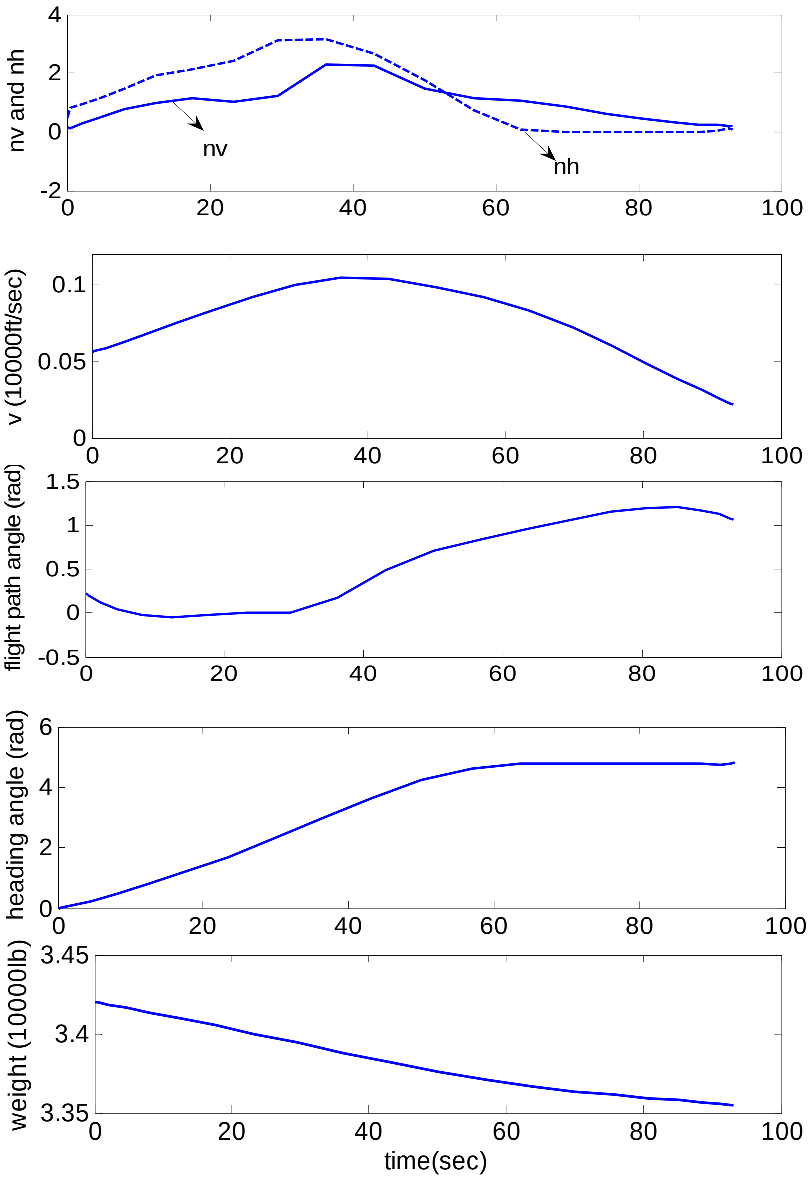

Figure 6.21: Control and State Variables History for Case 6

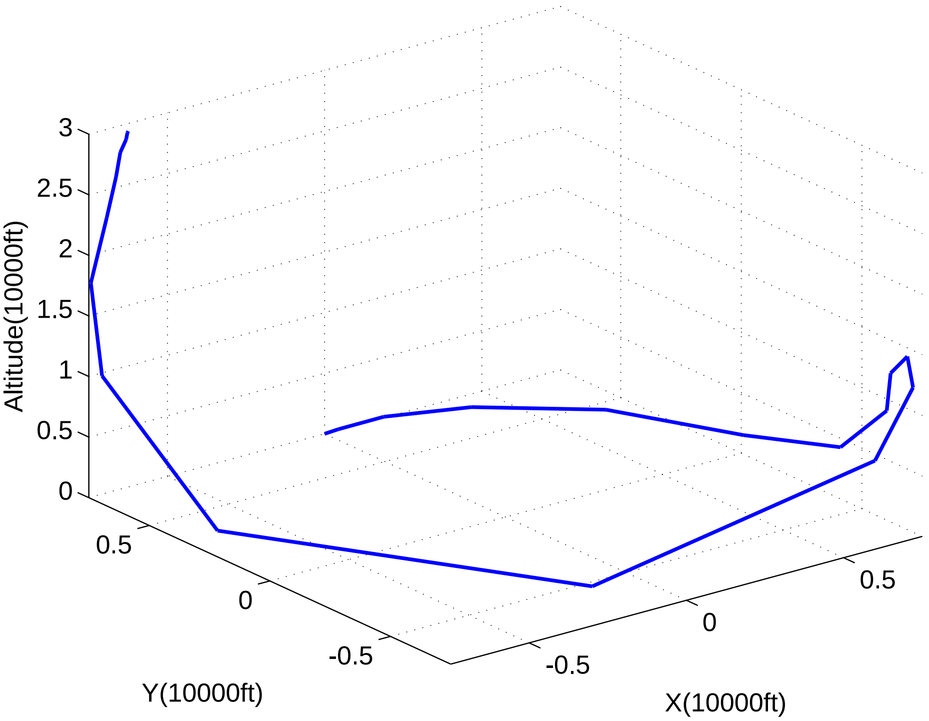

Figure 6.22: 3-D MFTC Trajectory for Case 7

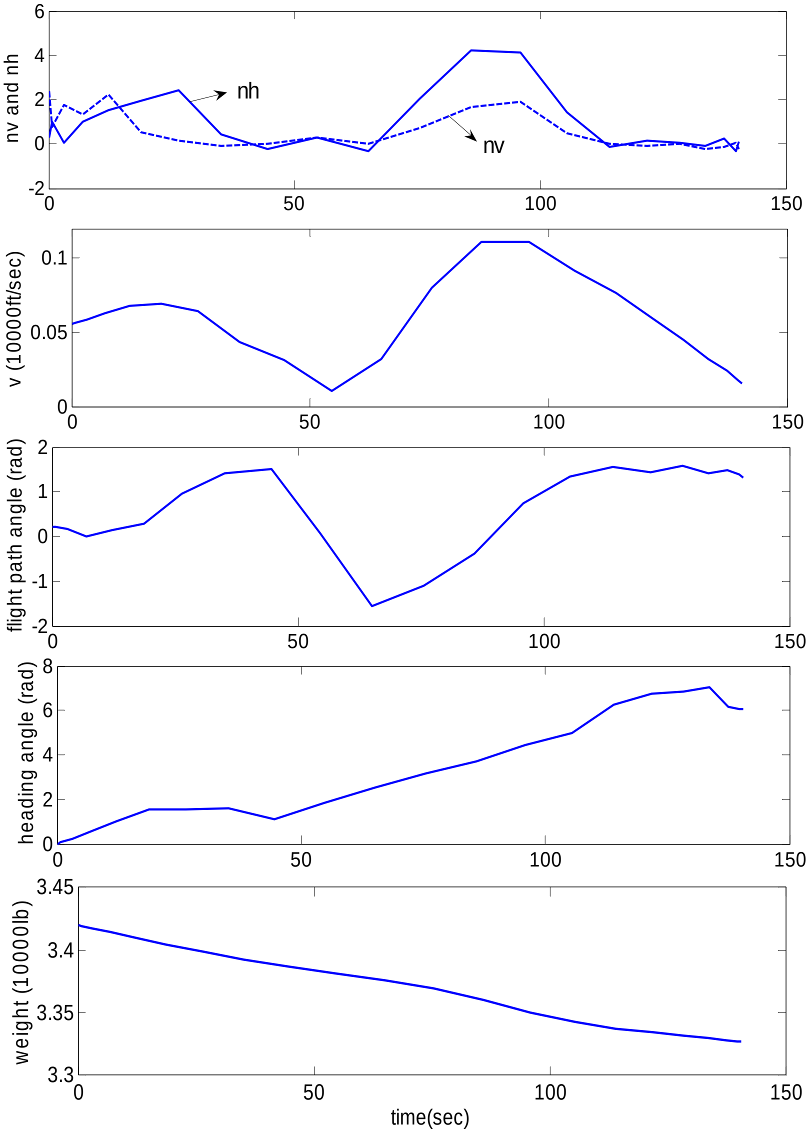

Figure 6.23: Control and State Variables History for Case 7

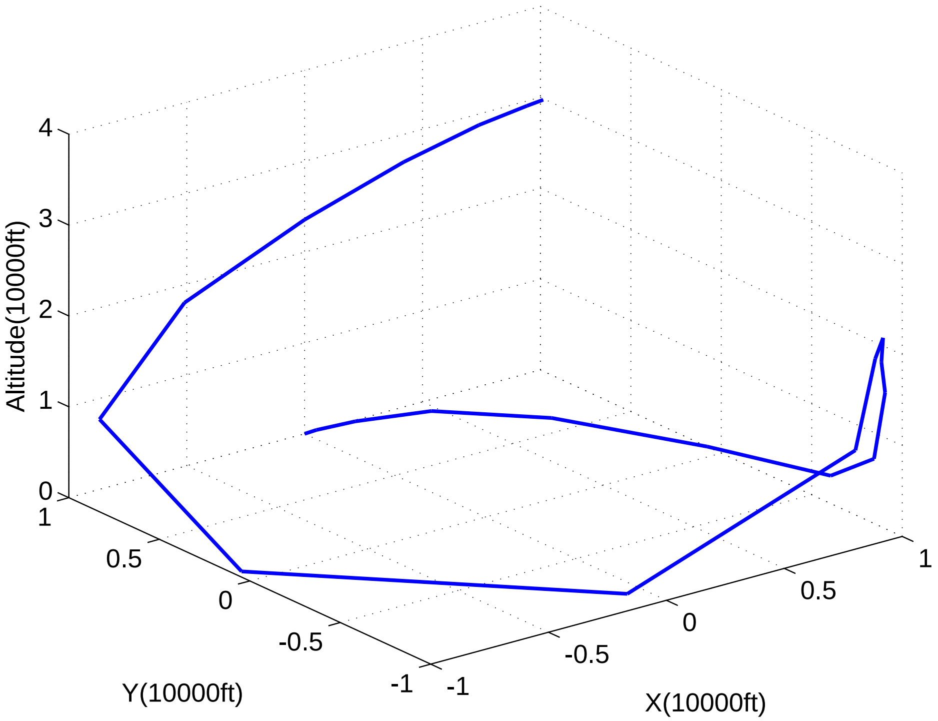

Figure 6.24: 3-D MFTC Trajectory for Case 8

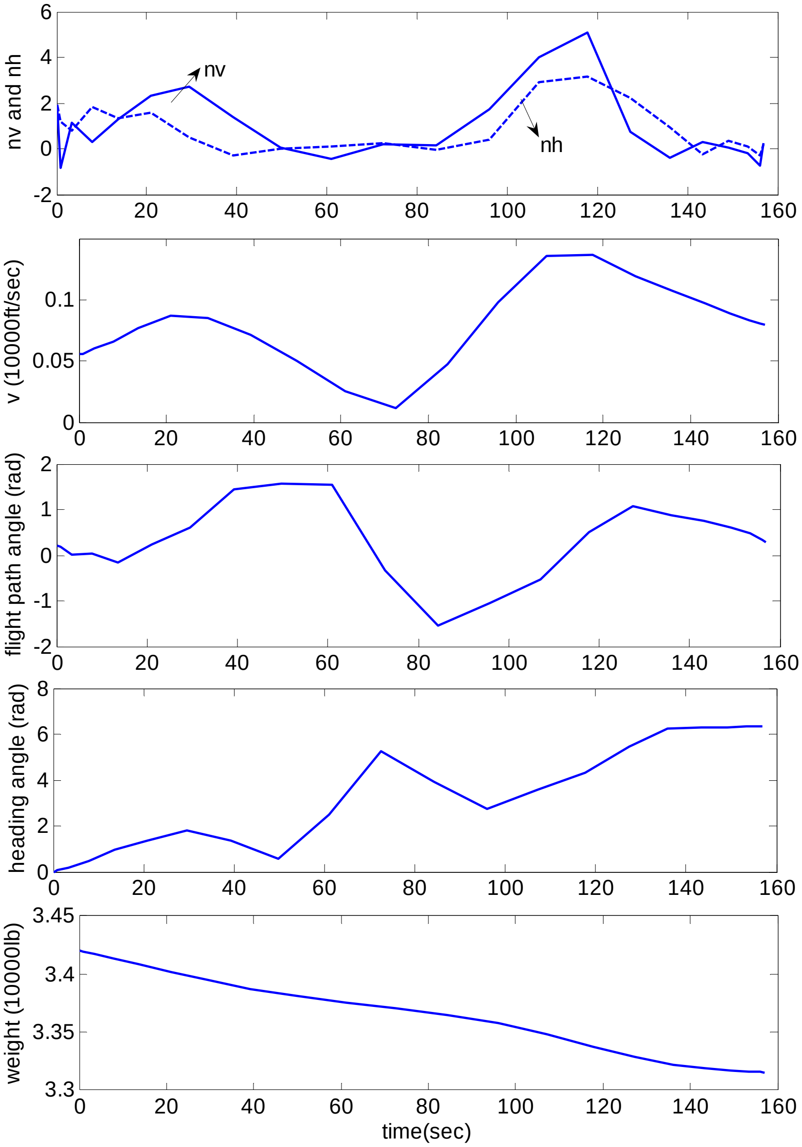

Figure 6.25: Control and State Variables History for Case 8

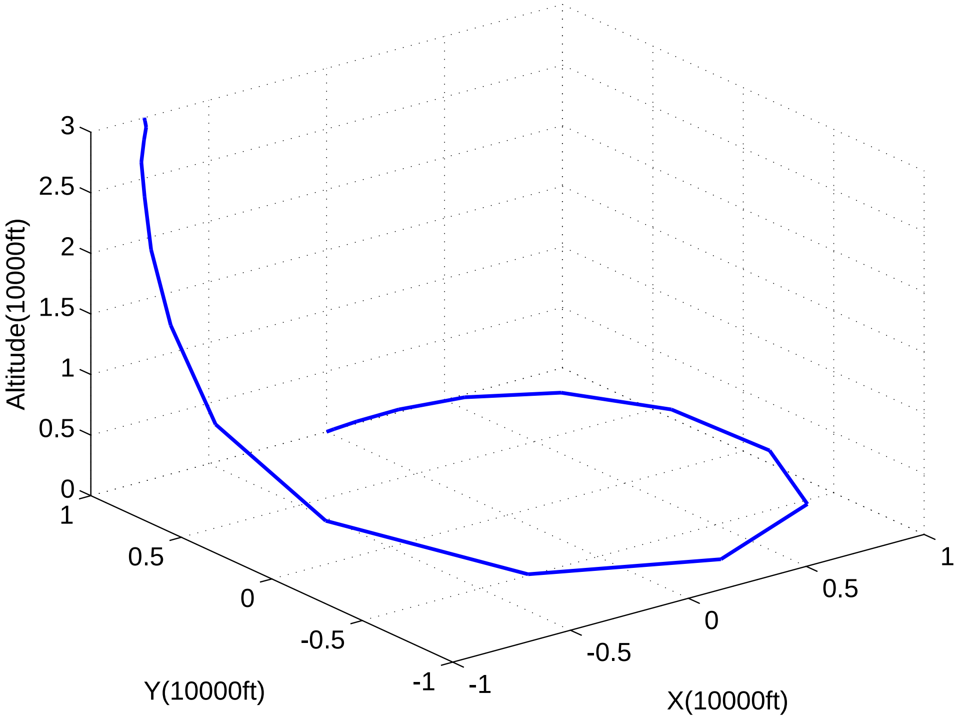

Figure 6.26: 3-D MFTC Trajectory for Case 9

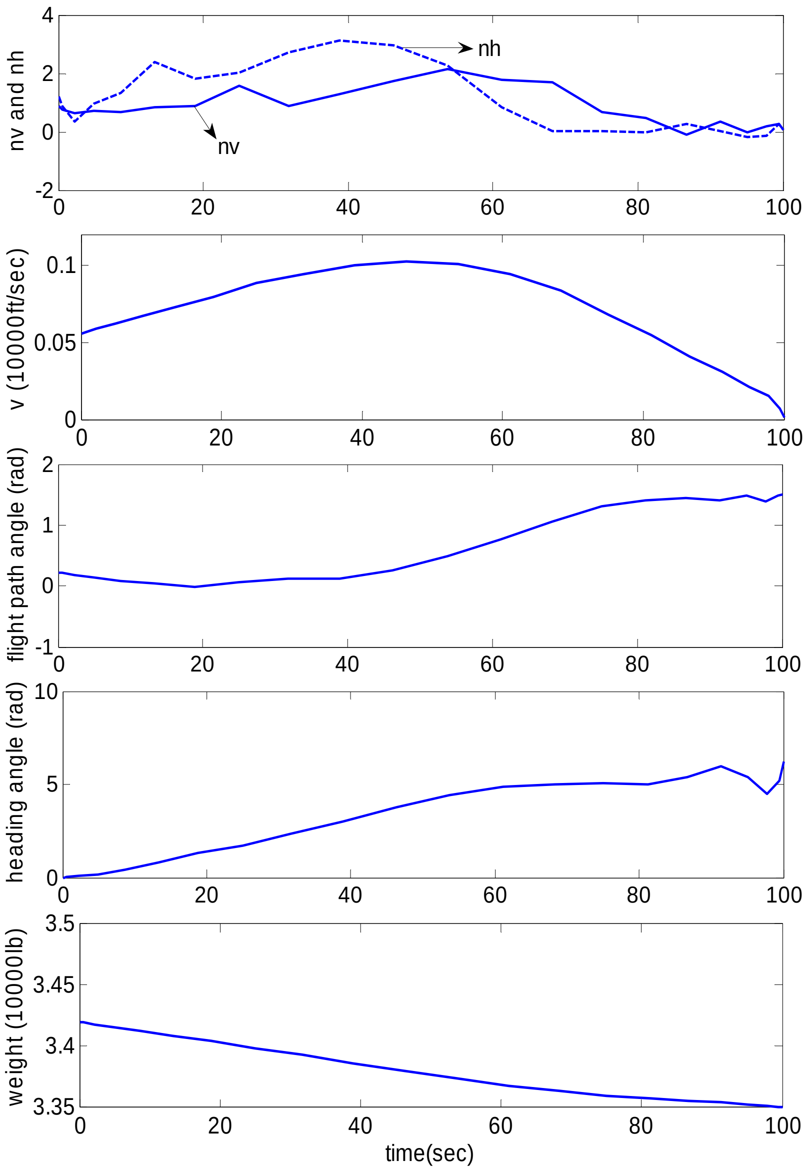

Figure 6.27: Control and State Variables History for Case 9

CHAPTER 7

OTHER CONSTRAINED OPTIMAL TRAJECTORIES

This chapter will focus on solving problems about optimal trajectories for an aircraft intercepting or rendezvousing a constant velocity target. Results for several different initial interceptor and target velocities are presented. A proportional navigation type guidance law has been applied to the same interception problem to obtain results that are compared to the time optimal trajectories. Then Field-of-view[35, 36] limitations are introduced and added to the DCNLP problem and results for this more constrained case are compared with original problem results.

7.1 Three-Dimensional Proportional Navigation Guidance Law

The most commonly used guidance laws are forms of Proportional Navigation Guidance (PNG). It is therefore of interest to compare the trajectories generated using DCNLP with corresponding PNG trajectories. Here, we use the idea of a three-dimensional PNG law formulated in terms of the relative geometry. The commanded acceleration from this law is perpendicular to the instantaneous interceptor to target line-of-sight (LOS) and proportional to the LOS angular velocity magnitude ("LOS rate") times the magnitude of the closing velocity[6]. Mathematically, this can be expressed as[9, 10]:

$$\overrightarrow{A}_I = -K\dot{R}_{TI}\frac{\overrightarrow{R}_{TI}}{|R_{TI}|} \times \overrightarrow{\Omega} \tag{7.1}$$

In Eqn.7.7, K is the constant effective navigation ratio (to be chosen greater than 2), $\overrightarrow{\Omega}$ is the LOS angular velocity which is provided by the interceptor radar, and $\overrightarrow{R}_{TI} = \overrightarrow{R}_T - \overrightarrow{R}_I$ is the relative position vector of the target with respect to the interceptor and can be easily measured by the Doppler radar. The desired acceleration command $\overrightarrow{A}_I$ is perpendicular to both the LOS and the LOS angular velocity. This relationship is represented geometrically in Fig. 7.1.

In Fig. 7.1, the line connecting the interceptor and target is known as LOS, the LOS composed an angle velocity Ω with respect to the fixed reference frame. For simulation purposes, the LOS angular velocity may be computed using:

$$\overrightarrow{\Omega} = \frac{\overrightarrow{R}_{TI} \times \overrightarrow{V}_{TI}}{|R_{TI}|^2} \tag{7.2}$$

where $\overrightarrow{V}_{TI} = \overrightarrow{V}_T - \overrightarrow{V}_I$ is the relative velocity of the target with respect to the interceptor. The other component of acceleration $\dot{R}_{TI}$ is found from

$$\dot{R}_{TI} = \frac{R_{TI_x}V_{TI_{xx}} + R_{TI_y}V_{TI_y} + R_{TI_z}V_{TI_z}}{|R_{TI}|} \tag{7.3}$$

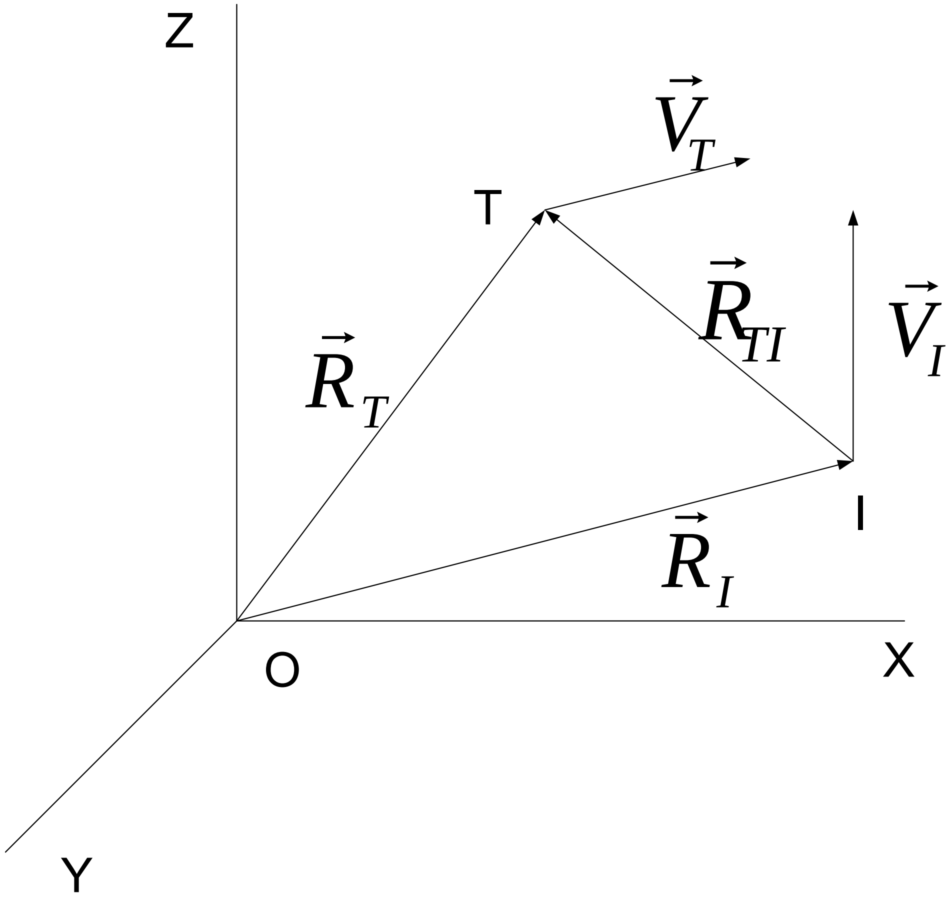

Figure 7.1: Engagement Geometry

where R_{TI_x}, R_{TI_y} and R_{TI_z} and the X, Y and Z components respectively of R_{TI} and V_{TI_x}, V_{TI_y} and V_{TI_z} are X, Y and Z component of V_{TI}.

For the purpose of interception, is was expected that the distance between the interceptor and target R_{TI} is as small as possible and the identical value is zero. The final closest distance between them is known as miss distance. From the theory of maximum and minimum, when a maximum or minimum point of a function is obtained, the derivative of this function at this point will be zero and sign of this derivative function will change before and after this summit point. So that at the closest point, the derivative of the distance between interceptor and target, V_{TI}, is zero and the sign of this value will change after this point is reached. So the ending point of this maneuver is when the V_{TI} changes its sign. This is also called necessary conditions for capture. If this condition can not be satisfied in expected intercept time, it indicates that the interceptor can not perform this task under the current initial condition and desired K value. Then it is necessary to adjust the initial condition of the interceptor or change the K value to meet the capture condition.

This 3-D PNG law is applied in the problem of intercepting a constant-velocity, constant-altitude target under different flight conditions. So the target velocity is treated as a pre-defined value during the whole process as well as its initial coordinates. The target's corresponding position change can also be easily derived according to its initial condition. The interceptor's initial velocity and coordinates are also known. Thereafter, the components of its acceleration Ω, $\overrightarrow{R}_{TI}$ and $\dot{R}_{TI}$ are all known at this launch point. So that we can integrate the interceptor velocity and 3-D coordinates values forward using trapezoidal integration method to get its corresponding velocity and coordinates in the next step time point. And this procedure is repeated until the closing velocity V_{TI} changes its sign.

Some Cases are simulated using this PNG law in the following section with different launch velocity and target velocity. Trajectory results, state variables and control variables history are compared with the corresponding simulation results using DCNLP method with the objective of minimizing interception time. Some cases also include the situation when the capture condition can not be satisfied in the desired interception time to show the advantage of DCNLP method in solving such kind of problems. The PNG algorithm code is included in Appendix C.

7.2 Three-Dimensional Minimum-Time Interception Trajectory Planning

The ideal interception condition is that the interceptor and target has the same coordinate, so we have the following relationship

$$h_I = h_T, \quad x_I = x_T, \quad y_I = y_T \tag{7.4}$$

here, the subscripts T and I are used to denote the interceptor and the target, respectively. Eqn.7.4 can be treated as final boundary conditions of the MTI problem. Because the target is flying under constant velocity. Assuming the y axis is parallel to the velocity direction of the target, then the final position of the target can be expressed as

$$\begin{aligned} h_{Tf} &= h_T \\ x_{Tf} &= x_T \\ y_{Tf} &= y_{T0} + V_T(t_f - t_0) \end{aligned} \tag{7.5}$$

where h_T, x_T and V_T are the constant altitude, x-position and speed of the target, respectively. Also, y_{T0} and y_{Tf} are the initial and final y-position of target, respectively. t_f is the final time and t_0 is the initial time. Its performance limitations for Aircraft 2 as well as the initial conditions and final interception conditions are included in table.7.1.

Table 7.1: Boundary Constraints and Performance Limitations for 3-D MTI Problem

Constraints	Values
Initial Coordinate Constraints	$x_{I0} = x_0$, $y_{I0} = y_0$, $h_{I0} = 0$
Initial V, γ and χ Constraints	$V = V_0$, $\gamma = \gamma_0$, $\chi = \chi_0$
Final Altitude and x-position of interception point	$h_{If} = h_T$, $x_{If} = x_T$
Final y position of interception point	$y_{If} = y_{T0} + V_T(t_f - t_0)$
Maximum and Minimum Vertical Load Factor	$n_{Vmax} = 2$, $n_{Vmin} = -2$
Maximum and Minimum Horizontal Load Factor	$n_{hmax} = 2$, $n_{hmin} = -2$

In table.7.1, h_{If}, x_{If} and y_{If} are the final altitude, x-position and y-position of the interceptor, respectively. The state and control variables constraints in table.7.1 together with the nonlinear constraints listed in Eqn.6.29 and objective function $J = t_f - t_0$ formulated the MTI problem. The DCNLP solution will optimize the interception time and at the same time satisfy both the maneuvered system equations and the interception conditions.

Case 1: set the initial conditions on the interceptor state as: $x_{I0} = 0$, $y_{I0} = -10000$ ft, $h_{I0} = 0$, $\gamma_0 = 12.6°$, $M_{I0} = 0.4$; set the initial and constant conditions on the target as: $x_T = 10000$ ft, $y_{T0} = -10000$ ft, $h_T = 10000$ ft and $M_T = 0.5$. In Case 1, the minimum time for the interceptor to reach the target is 47.82 sec and the miss distance at the terminal time is very close to zero. The 3-D trajectory and relative properties for this case are shown in Fig. 7.2 and Fig. 7.4. The trajectory and velocity history of Case 1 using PNG laws with $K = 5$ is shown in Fig. 7.3 and Fig. 7.4. Interception time using this method is 418.473 sec and the miss distance is 0.0257 ft. It's obvious to see that the DCNLP planning trajectory will intercept the target with much less time and high accuracy.

Case 2: For the interceptor, the same initial conditions as Case 1 are used, except we set $M_{I0} = 0.5$ and $M_T = 0.4$. The minimum time for the interceptor to reach the target is 36.52 sec and the miss distance at the terminal time is also close to zero. The 3-D trajectory and relative properties of this case is shown in Fig. 7.5 and Fig. 7.7. The trajectory and flight speed history for Case 2 using the PNG laws with $K = 5$ are shown in Fig. 7.6 and Fig. 7.7. The interception time using this method is 272.025 sec and the miss distance is 0.0026 ft. The time is so large because the interception speed does not increase as much as in the MTI solution.

Case 3: For the interceptor, the same initial conditions as Case 1 are used, except we set $M_T = 0.6$. The minimum time for the interceptor to reach the target is 51.45

sec and the miss distance at the terminal time is also close to zero. The 3-D trajectory and relative properties of this case is shown in Fig. 7.8 and Fig. 7.10. While the PNG law does not work on this case. The closing velocity will change its sign in the distance of 12492.1 ft which is far away from the desired miss distance. The trajectory and flight speed history for Case 3 using the PNG laws with $K = 5$ are shown in Fig. 7.9 and Fig. 7.10.

Case 4: For the interceptor, the same initial conditions as Case 2 are used, except we set $M_I = 0.6$. The minimum time for the interceptor to reach the target is 30.43 sec and the miss distance at the terminal time is also close to zero. The 3-D trajectory and relative properties of this case is shown in Fig. 7.11 and Fig. 7.13. While the PNG law does not work on this case. The closing velocity will change its sign in the distance of 13368.7 ft which is far away from the desired miss distance. The trajectory and flight speed history for Case 4 using the PNG laws with $K = 5$ are shown in Fig. 7.12 and Fig. 7.13.

7.3 Three-Dimensional Minimum-Time Rendezvous Trajectory Planning

The 3-D minimum-time rendezvous (MTR) problem is to catch up the target, also in constant-altitude and constant-velocity, with same velocity and desired distance. So this kind of problem is similar to the 3-D MTI problem except that MTR problem requires the interceptor has same final velocity as the target's velocity in addition to the final coordinates constraints. They are formulated as

$$\begin{aligned} h_I &= h_T \quad & x_I - x_T &= d \\ y_I &= y_T \quad & v_{If} &= v_T \\ \gamma_{If} &= 0 \quad & \chi_{If} &= -\pi/2 \end{aligned} \tag{7.6}$$

where d is the final x-position distance between the interceptor and the target, v_{If}, γ_{If} and χ_{If} are the final velocity, flight path angle and heading angle of the interceptor, respectively. Two cases are simulated using different launch velocity and target velocity for the MTR problem with same performance limitations illustrated in table.7.1.

Case 5: Use the same initial conditions of interceptor and target as Case 1. The final distance in x-position is $d = 50$ ft. The minimum time for the interceptor to catch up the target is 38.84 sec using DCNLP method. The 3-D trajectory and relative properties of this case is shown in Fig. 7.14 and Fig. 7.16.

Case 6: Use the same initial conditions of interceptor and target as Case 2. The final distance in x-position is also 50 ft. The minimum time for the interceptor to catch up the target is 57.90 sec using DCNLP method. The 3-D trajectory and relative properties of this case is shown in Fig. 7.15 and Fig. 7.17.

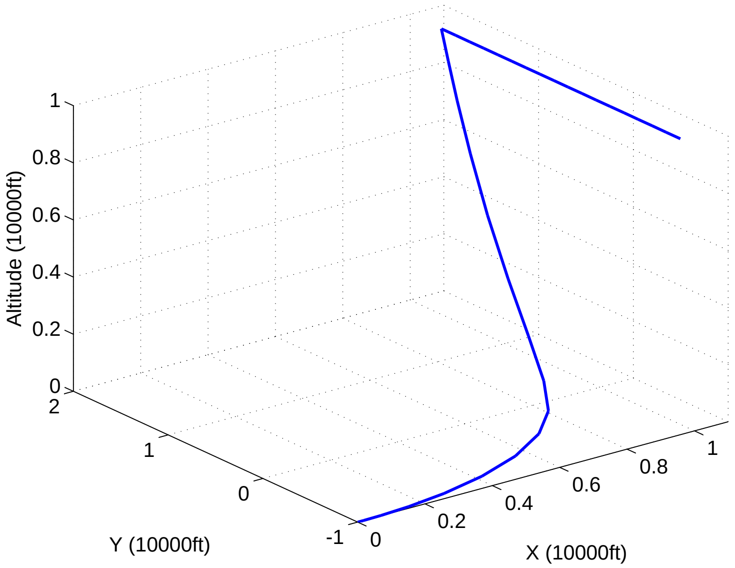

Figure 7.2: 3-D MTI Trajectory using DCNLP for Case 1

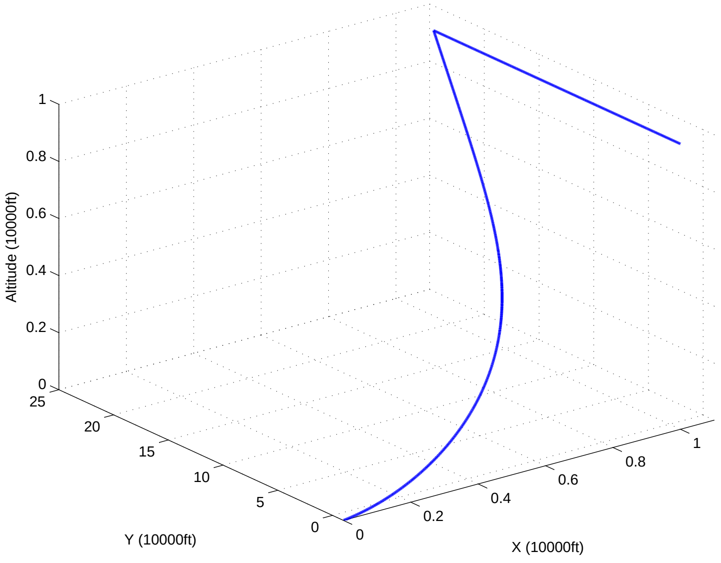

Figure 7.3: 3-D MTI Trajectory using PNG for Case 1

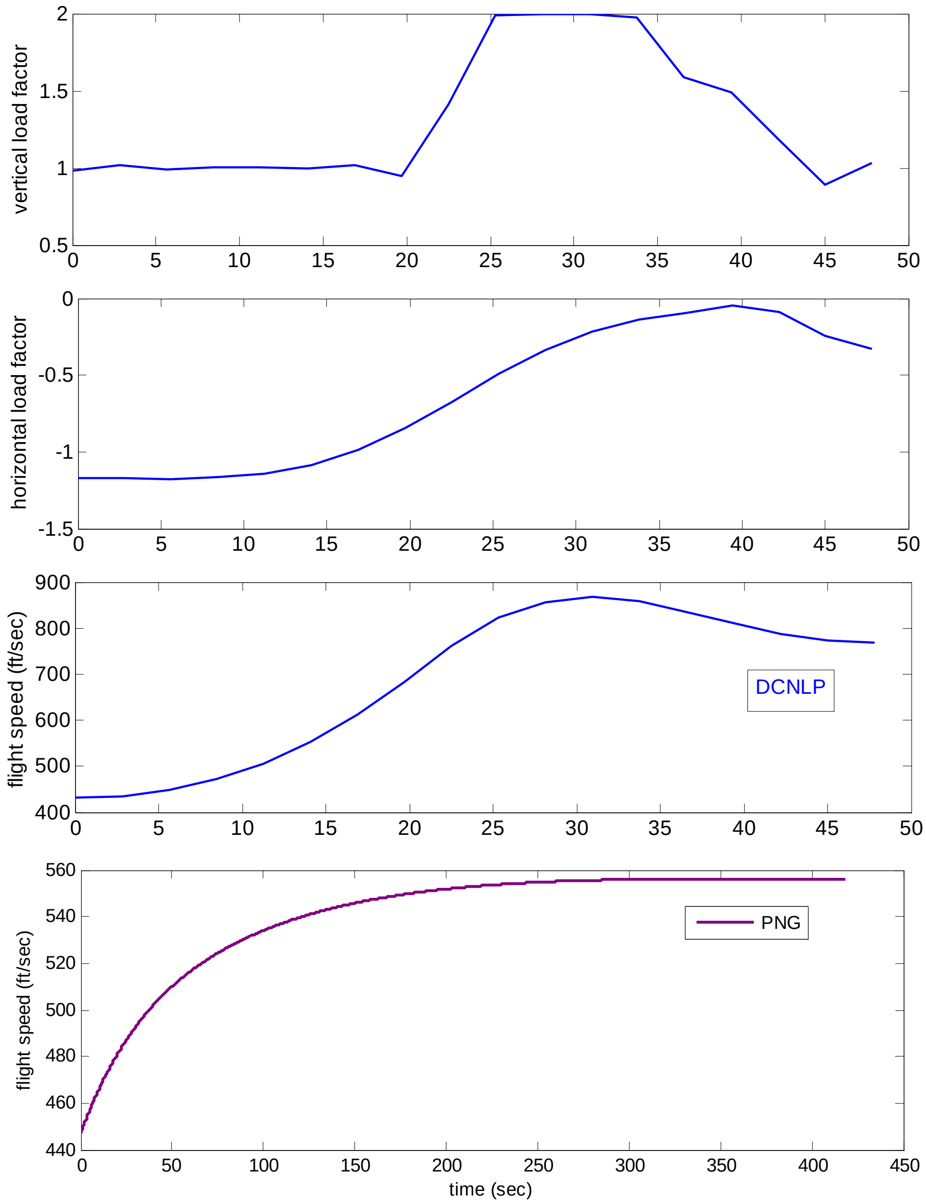

Figure 7.4: Control Factors and Velocity History for Case 1

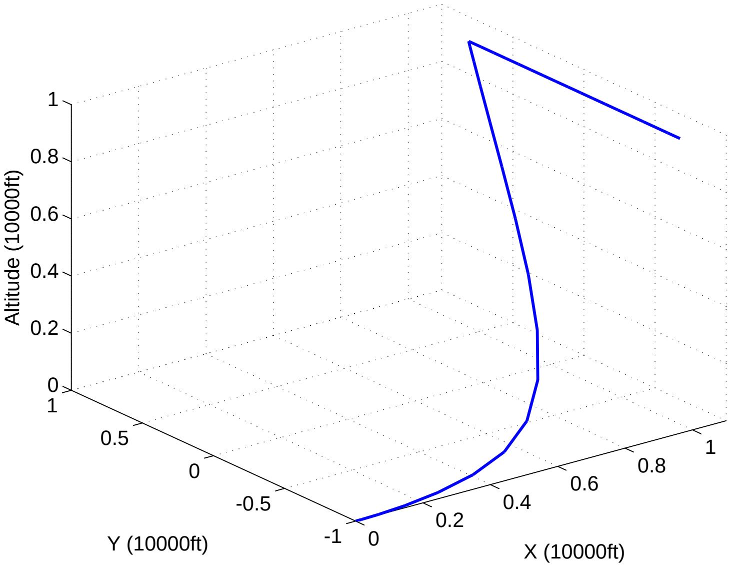

Figure 7.5: 3-D MTI Trajectory using DCNLP for Case 2

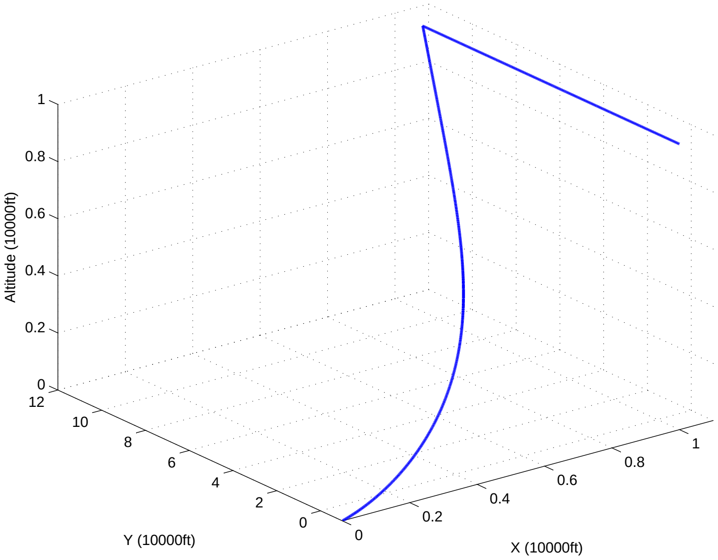

Figure 7.6: 3-D MTI Trajectory using PNG for Case 2

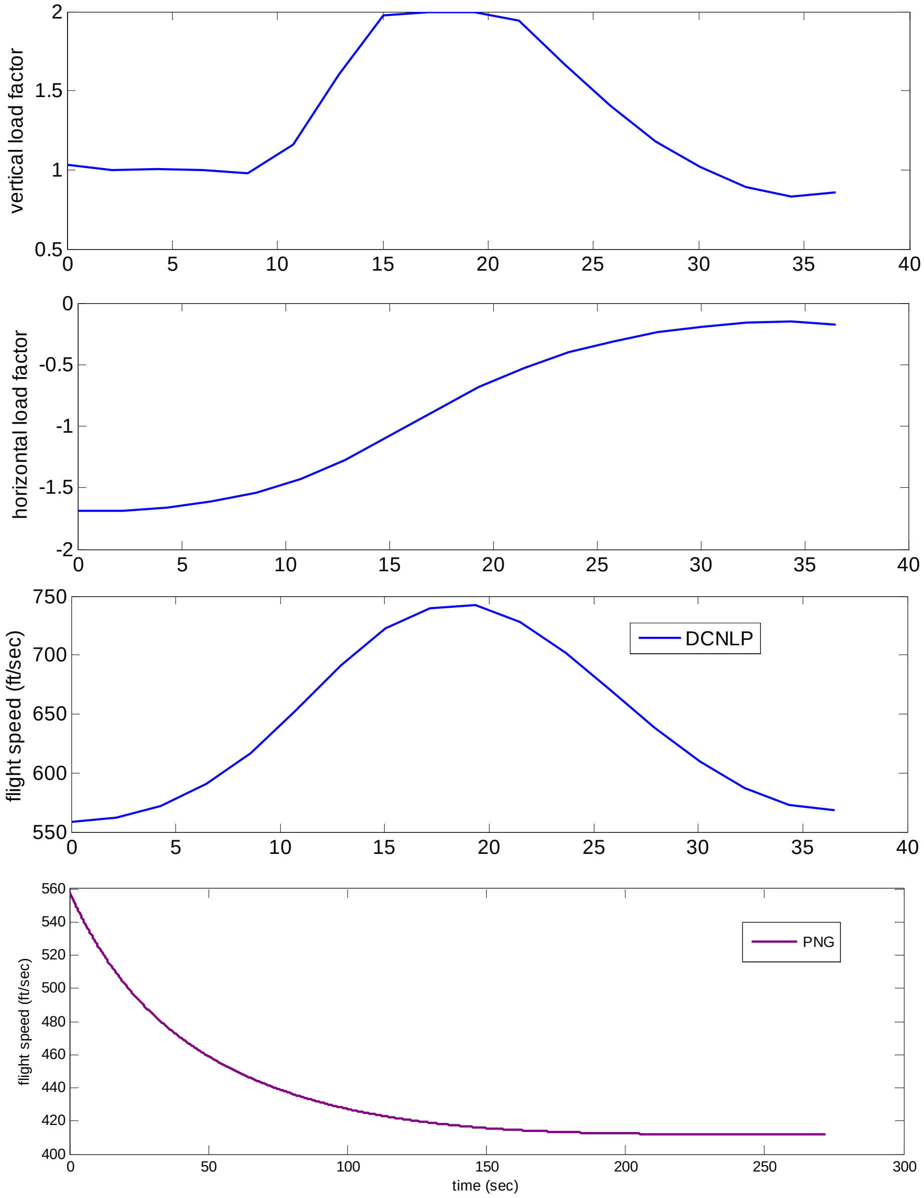

Figure 7.7: Control Factors and Velocity History for Case 2

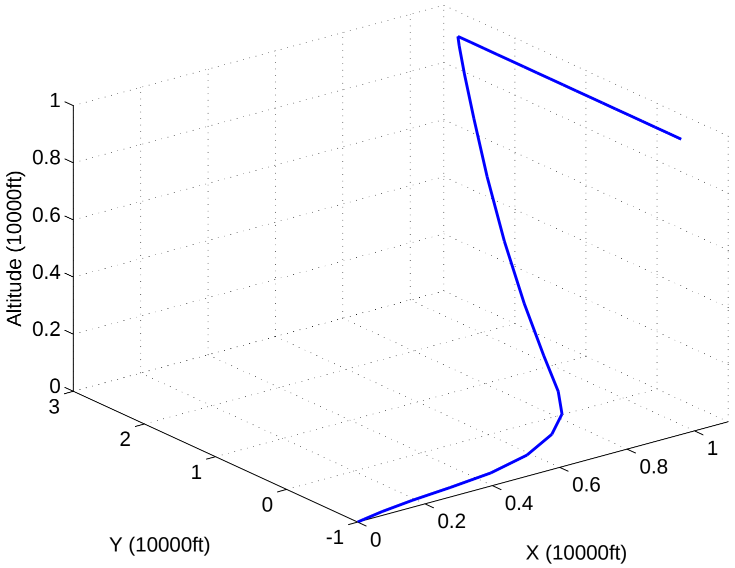

Figure 7.8: 3-D MTI Trajectory using DCNLP for Case 3

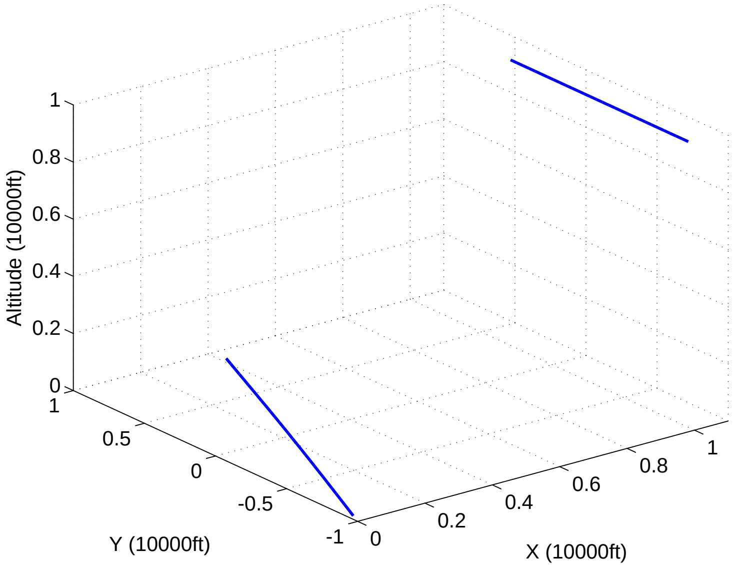

Figure 7.9: 3-D MTI Trajectory using PNG for Case 3

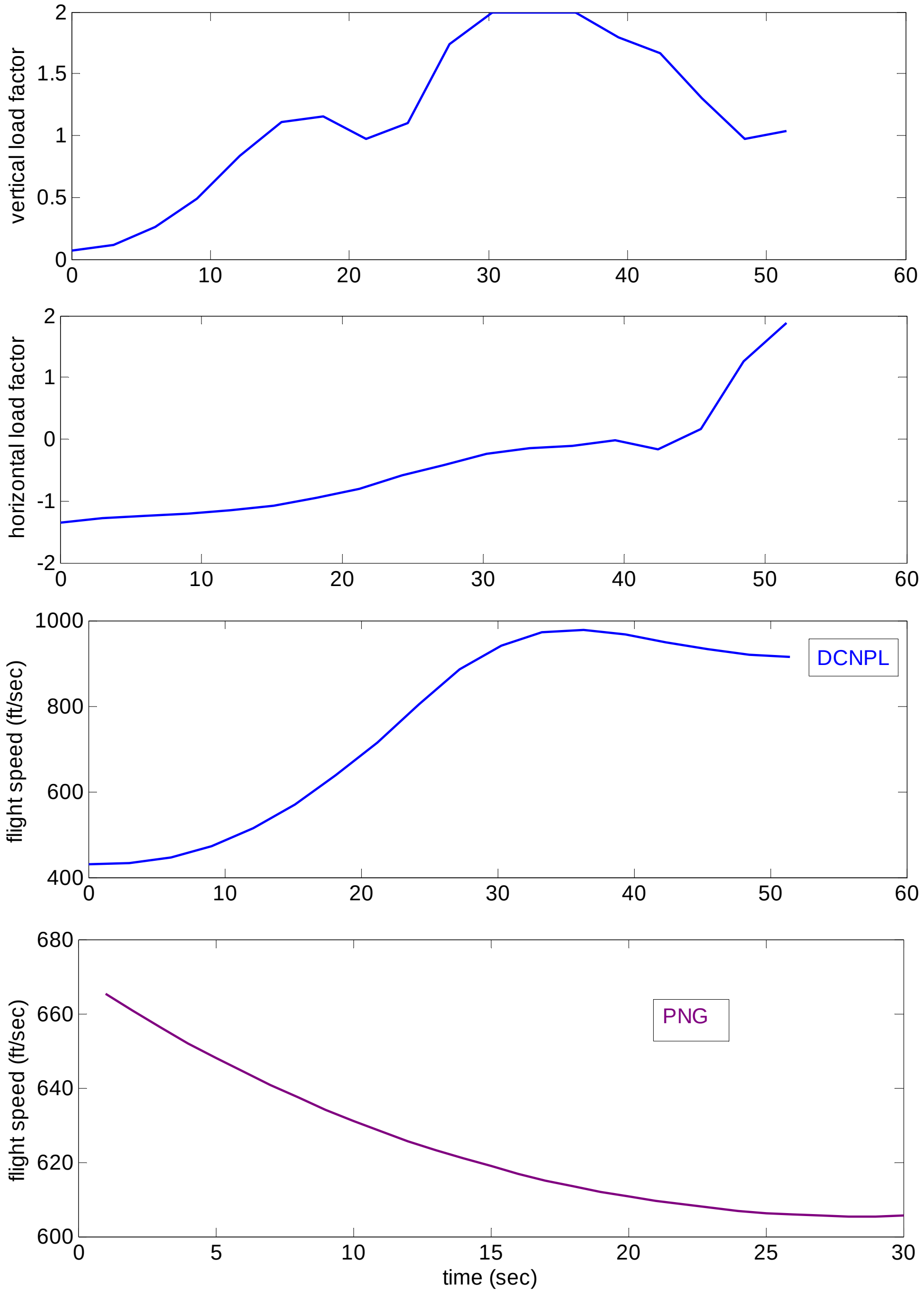

Figure 7.10: Control Factors and Velocity History for Case 3

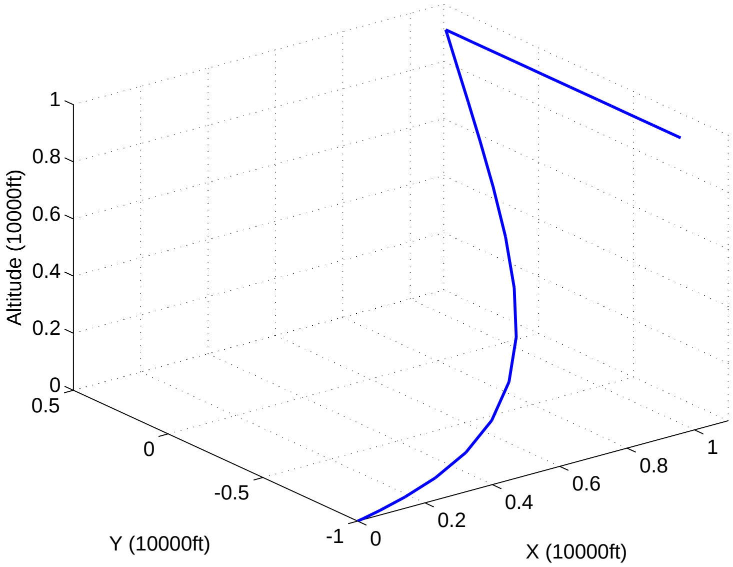

Figure 7.11: 3-D MTI Trajectory using DCNLP for Case 4

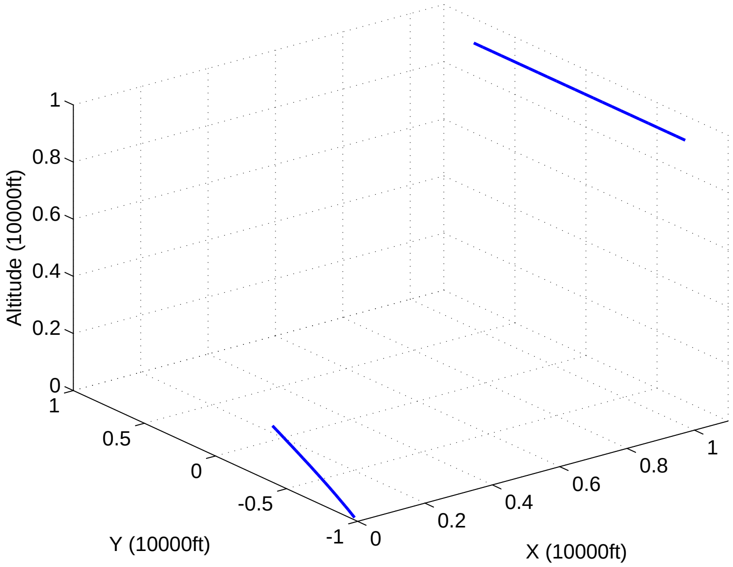

Figure 7.12: 3-D MTI Trajectory using PNG for Case 4

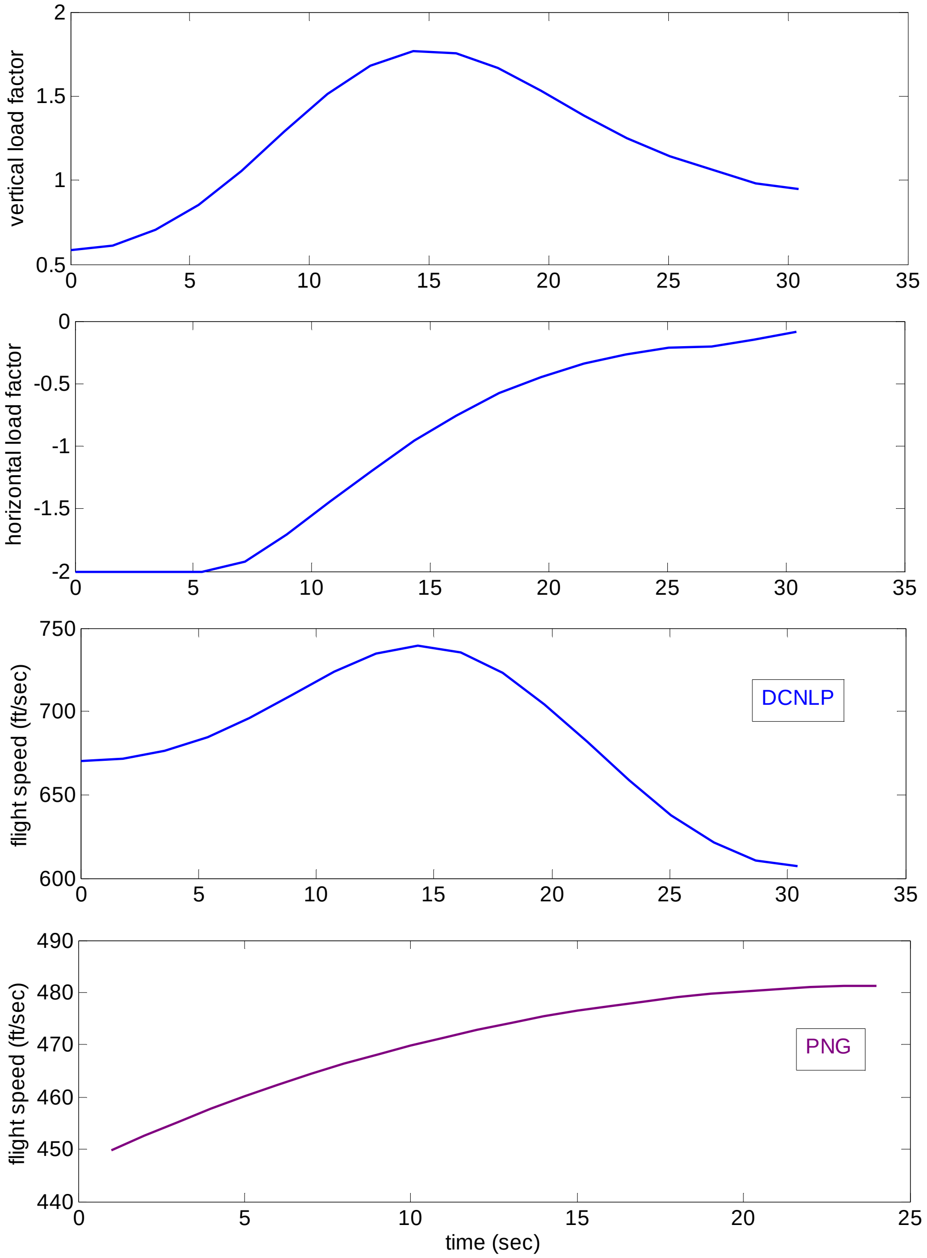

Figure 7.13: Control Factors and Velocity History for Case 4

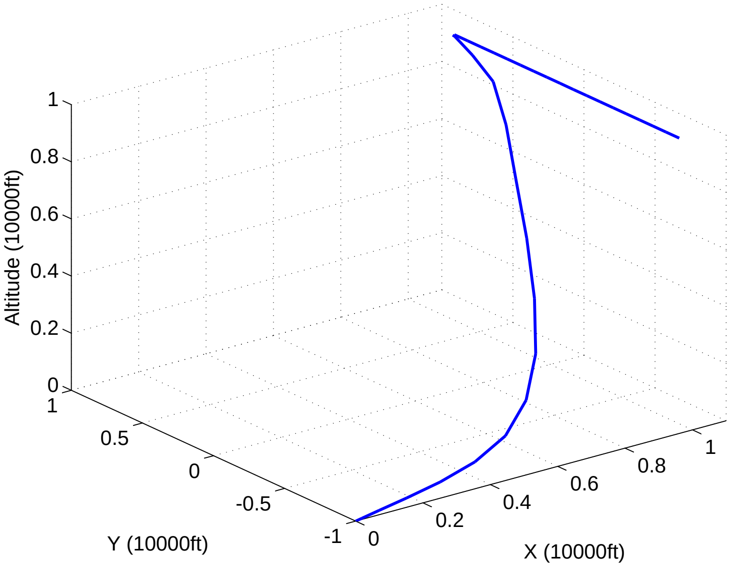

Figure 7.14: 3-D MTR Trajectory using DCNLP for Case 5

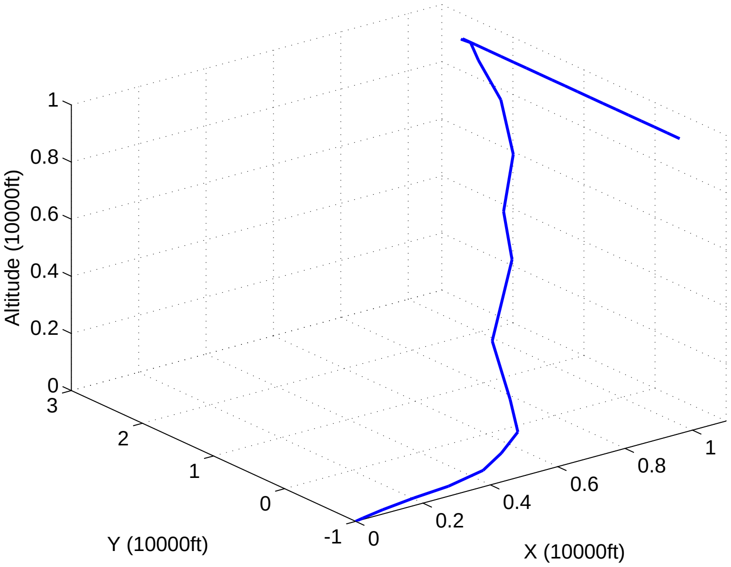

Figure 7.15: 3-D MTR Trajectory using PNG for Case 6

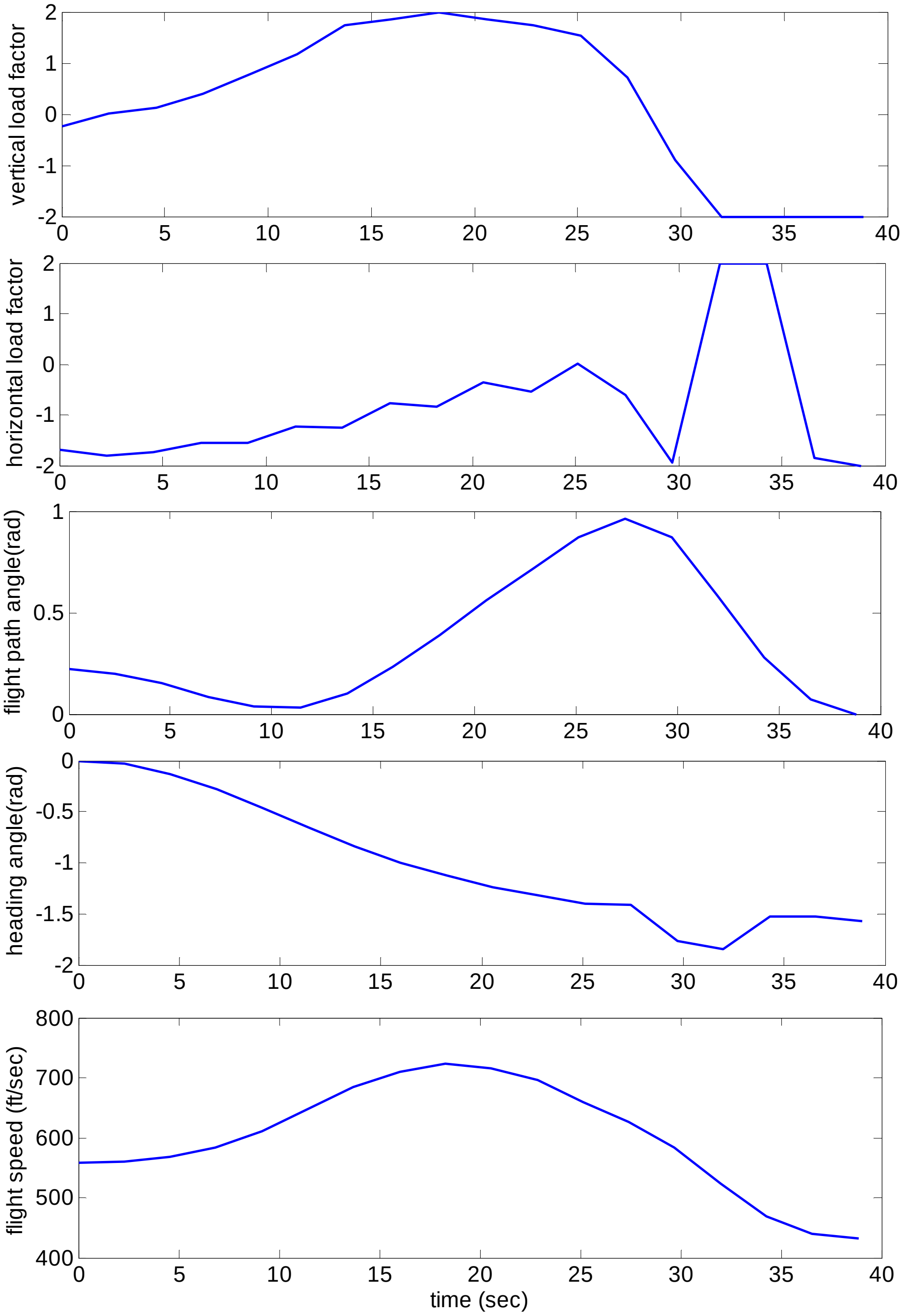

Figure 7.16: Control Factors and State Variables History for Case 5

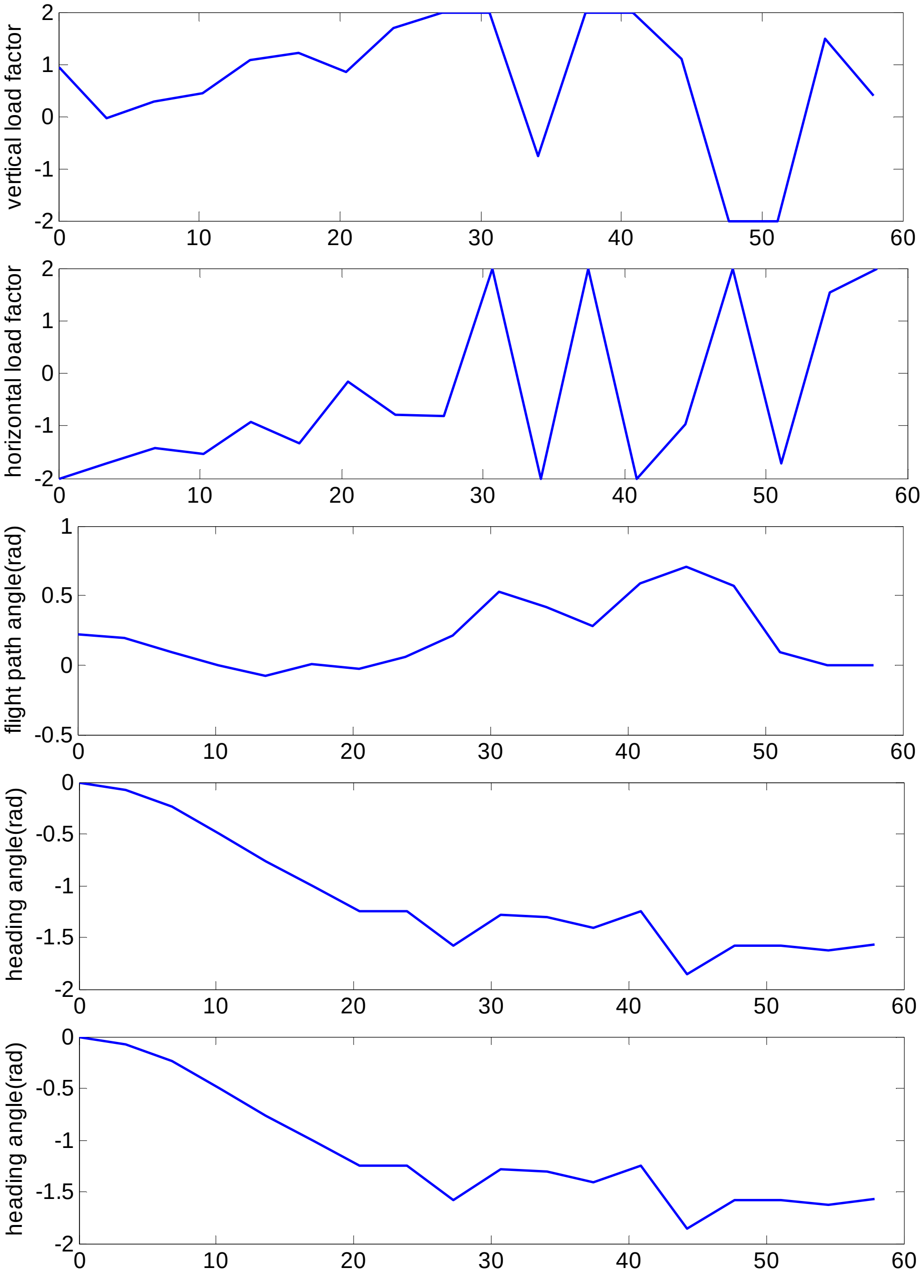

Figure 7.17: Control Factors and State Variables History for Case 6

7.4 View Constraints

In tactical flight, the interceptor seeks the target using the homing eye to provide the necessary measurement required for the implemented guidance law. So the success of the interception task depends largely on accuracy of the measurement. Some typical missile uses gyroscopes and a antenna mounted on the gimbals of the interceptor and the rotation of these gimbals is in a limited angle. So that the interceptor's look angle is also limited. These missiles' velocity direction is aligned with the imaginary LOS and their looking angle is measured from this imaginary LOS.

When a field of view constraint (FVC) is considered for the interceptor the target is expected to remain within the volume specified during the engagement. In this paper, we address only the interceptor velocity view limitation. In Fig. 7.18, the dotted line is the interceptors trajectory. At any point the angle between the interceptors velocity vector and LOS, θ_{LOS}, is expected to be smaller than the constraint cone angle, θ_c. In other words, the target is required to stay between the two points intercepted with the target trajectory and cone bottom. This additional condition can be treated as a nonlinear constraint and stated as:

$$\theta_{LOS} = \arctan(\frac{\overrightarrow{V} \cdot \overrightarrow{R}_{TI}}{|\overrightarrow{V}| \cdot |\overrightarrow{R}_{TI}|} \leq \theta_c) \tag{7.7}$$

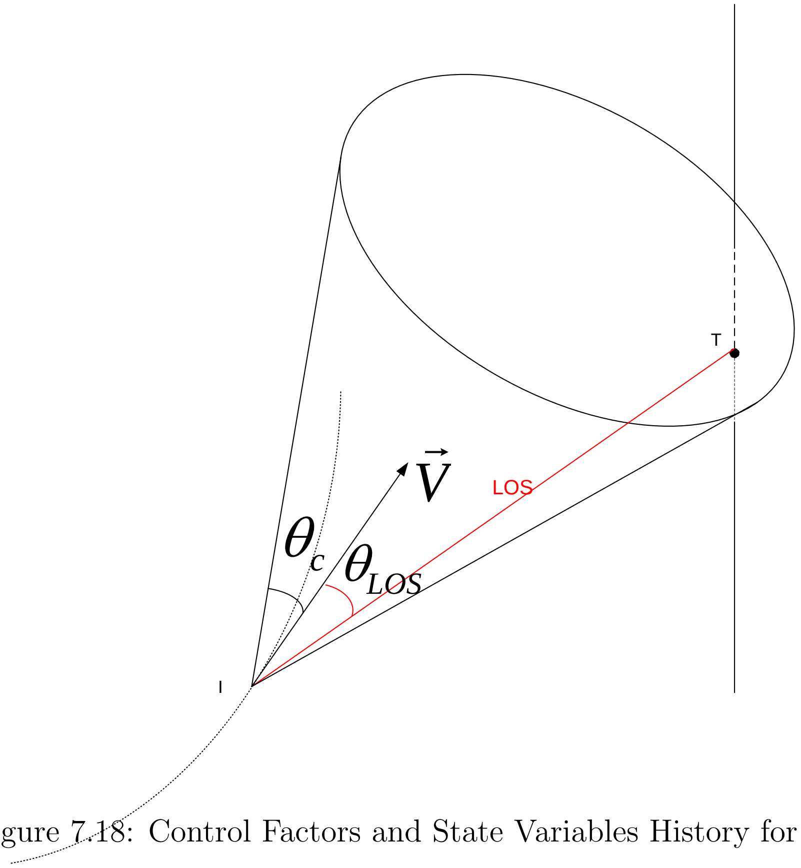

Figure 7.18: Control Factors and State Variables History for Case 7

One case is simulated considering this FVC. Case 7: Set the initial flight path angle of the interceptor as $\gamma_{I0} = 0.5$ with other conditions the same as Case 1. When we added the FVC condition to the same problem and let $\theta_c = 30^\circ$, the minimum time increased to 42.43 sec. The new trajectory is shown in Fig. 7.19 with relative properties shown in Fig. 7.20.

7.5 Conclusion

Three-dimensional minimum time intercept trajectory planning has been considered using direct collocation and nonlinear programming based on CP discretization. The results show that the performance indices, the interception times, are much less

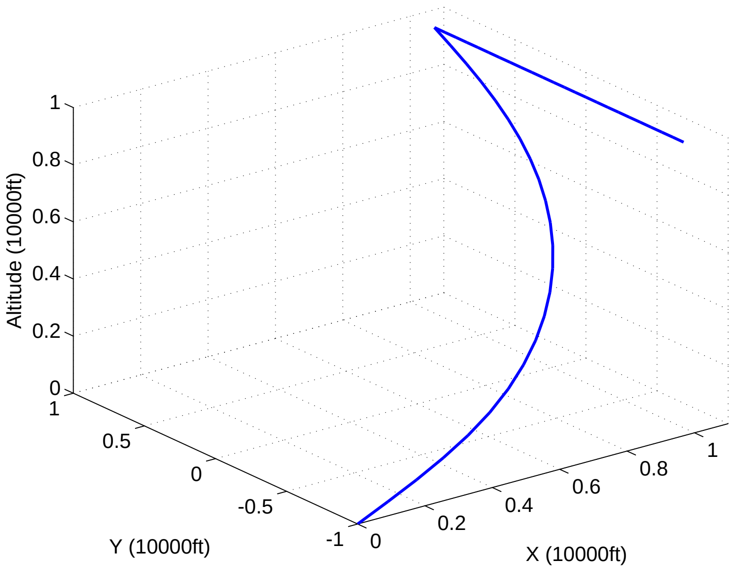

Figure 7.19: 3-D MTI Trajectory using DCNLP for Case 7

than those for trajectories generated by an idealized Proportional Navigation Guidance law, and the miss distances are smaller.

The interception time will of course depend on the engagement conditions, such as the interceptor's starting velocity and the target's speed, since to reach the desired target altitude, the interceptor needs to make a climb after it gets enough speed. Hence, the larger the magnitude of the starting velocity, the shorter the interception time. This property is similar to the 3-D MTTC problem. On the other side, a larger magnitude constant velocity of target directed away from the interceptor will require a larger interception time. The DCNLP method showed advantage in short interception time, low miss distance as well as wide range of initial conditions constraints for

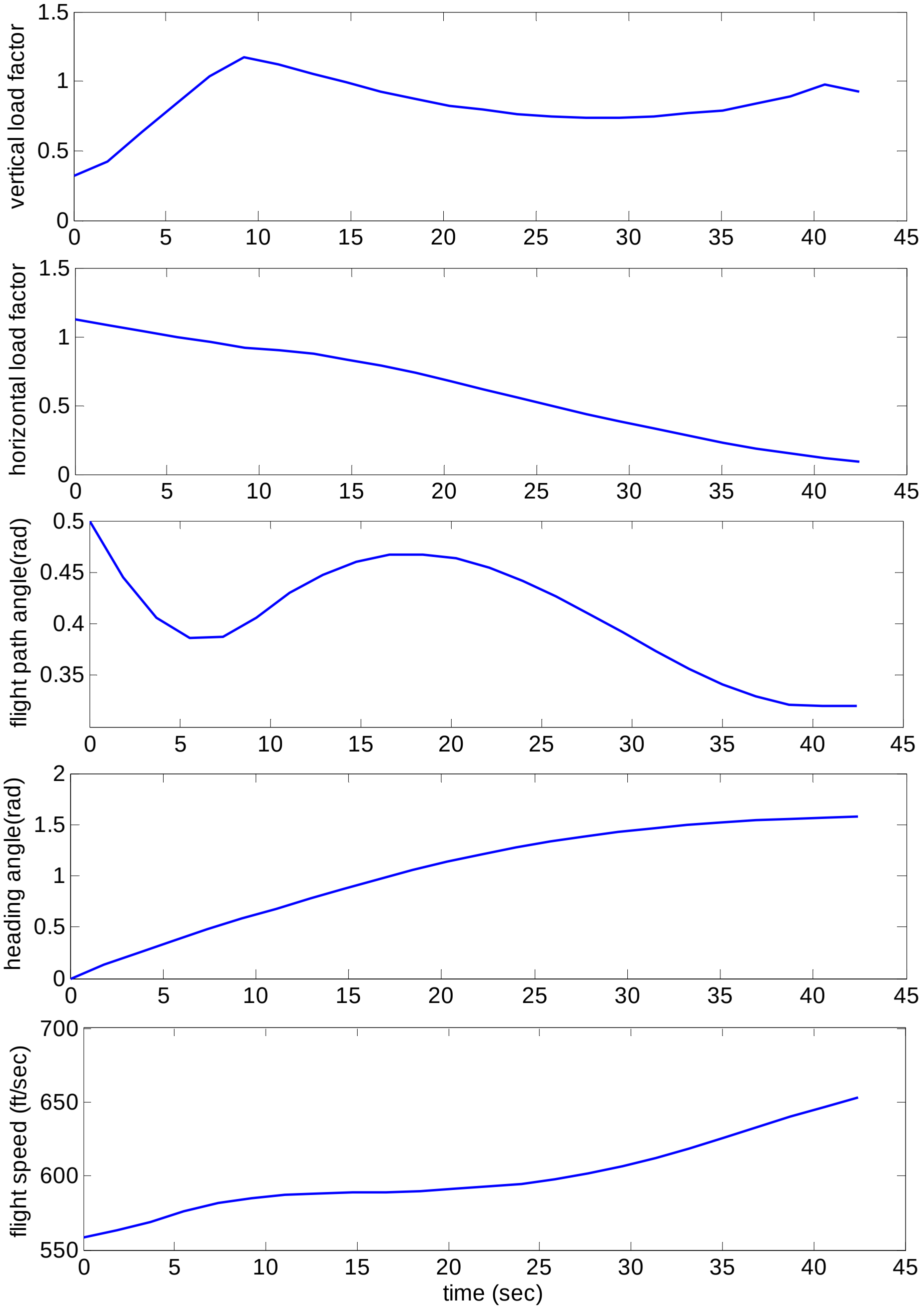

Figure 7.20: Control Factors and State Variables History for Case 7

interception. While PNG law will not work out for any condition of interception which is illustrated in Case 3 and 4.

The rendezvous results cost more time than the corresponding MTI trajectory to compensate the time used to satisfy the additional final constraints, those are the speed, flight path angle and heading angle constraints.

When field-of-view constraints are added to the original problem, more time is required to intercept because the interceptor cannot turn as rapidly. Although the direct collocation and nonlinear programming method has many advantages in solving this kind of optimal control problem, the computation time is large enough that even with fast computers it is not suitable for real-time applications. When the target's velocity is variable and trajectories need to be updated frequently according to radar tracking data, even more computation time will be needed. Then PNG law is superior considering on-board application.

Chapter 8

Summary and Recommendations

In this dissertation, the problems of finding solutions to three-dimensional minimum-time-to-climb (MTTC), minimum-fuel-to-climb (MFTC), minimum-time-interception (MTI) and minimum-time-rendezvous (MTR) problems in constrained airspace have been considered. The direct optimization method, direct collocation and nonlinear programming (DCNLP), has advantages in terms of (1) the avoidance of guessing initial adjoint variables, (2) fast convergence and (3) the capacity of including complex boundary conditions. Hence, it is considered to be the best method for solving these minimum time problems. The Chebyshev Pseudospectral discretization method was chosen as the primary collocation method due to its high accuracy and fast convergence in interpolating between the collocation points.

The MTTC and MFTC trajectories are similar to each other under same boundary conditions if the climb task is performed in a relatively short period time. Similarities also exist between the effect of engagement conditions on the performance of MTI and MTR trajectories. There are also common points between the MTTC and MTI trajectories in the effect of start up velocity on the final time. Accurate results and properties of the optimal trajectories are useful for the studies of the approximate or analytical solutions for these problems. They also provide valuable guidance to pilots regarding maneuver an aircraft how to achieve high performance.

Although the DCNLP method has many advantages in solving this kind of optimal control problems, the computation time is large enough that even with fast computers it is probably not suitable for real-time applications. It follows that the trajectories calculated here are flight planning trajectories which can be used as nominal for on-board guidance methods. This is specially true regarding the MTI problem, when the target's velocity is variable and trajectories need to be updated frequently according to radar tracking data, more computation time will be needed. Future research should focus on the development of simplified computation algorithms that can be applied on-board.

Bibliography

[1] Bryson, A. E. and Denham, W. F., “A Steepest-Ascent Method for Solving Optimum Programming Programs,” Journal of Applied Mechanics, Vol. 29, pp. 247-257, June 1962.

[2] Bryson, A. E. and Desai , M. N. and Hoffman, W. C., ”Energy-State Approximation in Performance Optimization of Supersonic Aircraft,” Journal of Aircraft, Vol. 6, No. 6, pp. 481-488, 1969.

[3] Calise, A. J., “Extended Energy Management for Flight Performance Optimization,” AIAA Journal, Vol. 15, No. 3, pp. 314-321, March 1977.

[4] Ardema, M. D., “Solution of the Minimum-Time-to-Climb Problem by Matched Asymptotic Expansions,” AIAA Journal, Vol. 14, No. 7, pp.843-850, 1976.

[5] Rao, A. V. and Mease, K. D., ”Minimum Time-to-Climb Trajectories Using a Modified Sweep Method,” AIAA-95-3263-CP, 1995.

[6] Zarchan, P., ”Tactical and Strategic Missile Guidance,” AIAA Tactical Missile Series, AIAA, Chap 2, pp. 11-29, 1997.

[7] Yuan, C.L., ”Homing and Navigation Courses of Automatic Target-Seeking Devices,” , Journal of Applied Physics, Vol 19, pp. 1122-1128, 1948.

[8] Adler, F. P., ”Missile Guidance by Three-Dimensional Proportional Navigation,” Journal of Applied Physics, Vol. 27, No. 5, pp. 500-507, 1956.

[9] Duflos, E., Penel, P. and Vanheeghe, P., ”3D Guidance Law Modeling,” IEEE Transactions on Aerospace and Electronic Systems, Vol. 35, No.1, pp. 72-83, 1999.

[10] Cochran, J. E. and No, T. S. and Thaxton, D. G., ”Analytical Solutions to a Guidance Problem,” Journal of Guidance, Control, and Dynamics, Vol. 14, No. 1, pp. 117-122, 1991.

[11] Visser, H. G., Kelley, H. J. and Cliff, E. M., ”Energy Management of Three-Dimensional Minimum-Time Intercept,” Journal of Guidance, Control, and Dynamics, Vol. 10, No. 6, pp. 574-580, 1987.

[12] Kumar, R. K., Seywald, H., and Cliff, E. M., "Near-Optimal Three-Dimensional Air-to-Air Missile Guidance Against Maneuvering Target," Journal of Guidance, Control, and Dynamics, Vol. 18, No. 3, pp. 457-464, 1995.

[13] Weston, A., Cliff, G. and Kelley, H., "Onboard Near-Optimal Climb-Dash Energy Management," Journal of Guidance, Control, and Dynamics, Vol. 8, No. 3, pp. 320-324, 1985.

[14] Bilimoria, K. and Cliff, E. M, "Singular Trajectories in Airplane Cruise-Dash Optimizaiton," Journal of Guidance, Control, and Dynamics, Vol. 12, No. 3, pp. 304-310, 1989.

[15] Corban, J. E., Calise, A. J. and Flandro, G. A, "Trajectory Optimization and Guidance Law Development for Transatmospheric Vehicles," Control and Applications, Proceedings. ICCON '89. IEEE International Conference, pp. 460-465, 1989.

[16] Bollino, K. P., Ross, M. I. and Doman, D. D., "Optimal Nonlinear Feedback Guidance for Reentry Vehicles," AIAA Guidance, Navigation and Control Conference and Exhibit, AIAA 2006-6074, 2006.

[17] Newman, B., Britcher, C., Kassaye, Y., Krizansky, M. and Acheson, M., "Trajectory Management Concepts for Future Small Aircraft Transportation Systems," Journal of Aircraft, Vol. 43, No. 6, pp. 1643-1654, 2006.

[18] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," Journal of Guidance, Control, and Dynamics, Vol. 21, No. 2, pp. 193-207, 1998.

[19] Hargraves, C. R. and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," Journal of Guidance, Control and Dynamics, Vol. 10, No. 4, pp 338-342, 1987.

[20] Betts, J. T. and Huffman, W. P., "Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming," Journal of Guidance, Control, and Dynamics, Vol. 16, No. 1, pp. 59-68, 1993.

[21] Ringertz, U., "Optimal Trajectory for a Minimum Fuel Turn," Journal of Aircraft, Vol. 37, No. 5, pp. 932-934, 2000.

[22] Norsell M., "Multistage Trajectory Optimization with Radar-Range Constraints," Journal of Aircraft, Vol. 42, No. 4, pp. 849-857, 2005.

[23] Williams, P., Sgarioto, D. and Trivailo, P. M., "Constrained Path-Planning for an Aerial-Towed Cable System," EUCASS 2005, 3.01.03, 2005.

[24] Herman, A. L. and Spencer, D. B., "Optimal, Low-Thrust Earth-Orbit Transfers Using Higher-Order Collocation Methods," Journal of Guidance, Control, and Dynamics, Vol. 25, No. 1, pp. 40-47, 2002.

[25] Horie, K. and Conway, B. A., "Optimal, Aeroassisted Orbital Interception," Journal of Guidance, Control, and Dynamics, Vol. 22, No. 5, pp. 625-631, 1999.

[26] Betts, J. T., "Trajectory Optimization in the Process of Uncertainty," The Journal of the Astronautical Sciences , Vol. 54, No. 2, pp. 227-243, 2007.

[27] Geiger, B. R., Horn, J. F. DeLullo, A. M. AND Long, L. N., "Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming," AIAA Guidance, Naviagtion, and Control Conference and Exhibit, AIAA 2006-6199, 2006.

[28] Yeo, B. P., "An Extension in Optimal Initial Choice of Multipliers in the Quasilinearization Method," International Journal of Control, Vol. 24, No. 5, pp. 593-608, 1976.

[29] Bryson, A. E. and Ho, Y. C., "Applied Optimal Control; Optimization, Estimation, and Control," Waltham, Mass., Ginn and Co, 1969.

[30] Hull, D. G., "Optimal Control Theroy for Applications," Mechanical Engineering Series, Springer.

[31] Herman, A. L. and Spencer, D. B., "Optimal, Low-Thrust Earth-Orbit Transfers Using Higer-Order Collocation Methods," Journal of Guidance, Control, and Dynamics, Vol. 25, No. 1, pp. 40-47, 2002.

[32] Fahroo, M. and Ross, I. M., "Direct Trajectory Optimization by a Chebychev Pseudospectral Method," Journal of Guidance, Control, and Dynamics, Vol. 25, No. 1, pp. 160-166, 2002.

[33] Trefethen, L. N., Spectral Methods in MATLAB, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[34] Pietz, J. A., "Pseudospectral Collocation Methods for the Direct Transcription of Optimal Control Problems," M.A. Thesis, Dept. of Computational and Applied Mathematics, Rice University, Houston, TX, 2003.

[35] Xin, M., Balakrishnan, S.N. and Ohlmeyer, E.J.,"Guidance Law Design for Missiles with Reduced Seeker Field-of-View," AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, 2006.

[36] Manchester, I.R., Savkii, A.V. and Faruqi, F.A.x, "Optical-flow based Precision Missile Guidance inspired by Honeybee Navigation," Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, 2006.

[37] Betts, J. T., "Practical Methods for Optimal Control Using Nonlinear Programming," Industrial and Applied Mathematics.

[38] Holmstrom, H., Goran, A.O. and Edvall, M. M, "User's Guide For TOMLAB /SNOPT," Tomlab Optimization Inc., 2005.

[39] Gano, S. E., Perez, V. M. and Renaud, J. E., "Development and Verification of A MATLAB Driver For The SNOPT Optimization Software," AIAA Paper 2001-1620, 2001.

[40] Phillips, J. P., "Brachistochrone, Tautochrone, Cycloid–Apple of Discord," Math. Teacher 60, pp.506-508, 1967.

[41] Zernelo, E., "Uber das Navigationsproblen bei ruhender oder veranderlicher Windverteilung," Z. Angrew. Math. und. Mech., 1931.

[42] Powers, W. F., "Hamiltonian Perturbation Theory for Optimal Trajectory Analysis," Master Thesis, Dept. of Aerospace Engineering, The University of Texas, Austin, TX, 1966.

[43] Etkin, B. and Reid, L. D., "Dynamics of Flight: Stability and Control (3rd edition)," John Wiley and Sons Inc, 1995.

[44] Dai, R. and Cochran, J. E., "Three-Dimensional Minimum-Time-To-Climb in Constrained Aerospace," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2007.

[45] Dai, R. and Cochran, J. E., Three-Dimensional Minimum-Time Interception Trajectory Planning Using Nonlinear Programming and Collocation, AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, 2007.

Appendix A

Aircraft Propulsion and Aerodynamic Data

This Appendix presents the aircraft propulsion data effected by Altitude and March number and aerodynamic data effected by March number only in tables.

Table A.1: Thrust as a function of altitude and Mach number from Ref.[2] for aircraft 2.

	Thrust T (thousands of lb)											
Mach	Altitude h (thousands of ft)											
No.M	0	5	15	25	35	45	55	65	75	85	95	105
0	23.3	20.6	15.4	9.9	5.8	2.9	1.3	0.7	0.3	0.1	0.1	0.0
0.4	22.8	19.8	14.4	9.9	6.2	3.4	1.7	1.0	0.5	0.3	0.1	0.1
0.8	24.5	22.0	16.5	12.0	7.9	4.9	2.8	1.6	0.9	0.5	0.3	0.2
1.2	29.4	27.3	21.0	15.8	11.4	7.2	3.8	2.7	1.6	0.9	0.6	0.4
1.6	29.7	29.0	27.5	21.8	15.7	10.5	6.5	3.8	2.3	1.4	0.8	0.5
2.0	29.9	29.4	28.4	26.6	21.2	14.0	28.7	5.1	3.3	1.9	1.0	0.5
2.4	29.9	29.2	28.4	27.1	25.6	17.2	10.7	6.5	4.1	2.3	1.2	0.5
2.8	29.8	29.1	28.2	26.8	25.8	3.4	1.7	1.0	0.5	0.3	0.1	0.5
3.2	29.7	28.9	27.5	26.1	24.9	20.3	13.0	8.0	4.9	2.8	1.4	0.5

Table A.2: Lift and drag coefficients as a function of angle of attack and Mach Number for aircraft 2.

M	0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2
C_{L_α}	2.240	2.325	2.350	2.290	2.160	1.950	1.700	1.435	1.250
C_{D_0}	0.0065	0.0055	0.0060	0.0118	0.0110	0.0086	0.0074	0.0069	0.0068

$S = 500ft^2$, $W = 34200lb$, $\eta = 1$

Appendix B

Gradient Method for 2-D MTTC problem

This Appendix listed the calculation procedure for the 2-D MTTC problem.

1. Assume a nominal control load factor history $n^*(t)$, integrate the system differential equations forward from the starting point, store state variables history $V(t)$, $\gamma(t)$ and $h(t)$.

2. Derive the adjoint variables derivative equations as

$$\begin{gathered}
\dot{\lambda_V} = -\lambda_V\left(\tfrac{\partial T}{\partial V} - \rho V S C_{D_0} + \tfrac{1}{2}\rho V^2 S \tfrac{\partial C_{D_0}}{\partial V} + \tfrac{4n^2W^2}{C_{L\alpha}\rho V^3 S} + \tfrac{2n^2W^2}{C_{L\alpha}^2\rho V^2 S}\tfrac{\partial C_{L\alpha}}{\partial V}\right)g/W \\
+\lambda_\gamma \tfrac{g}{V^2}(n - \cos\gamma) - \lambda_h \sin\gamma \\
\dot{\lambda_\gamma} = \lambda_V g\cos\gamma - \lambda_\gamma \tfrac{g}{V}\sin\gamma - \lambda_h V\cos\gamma \\
\dot{\lambda_h} = -\lambda_V \tfrac{g}{W}\tfrac{\partial T}{\partial h}
\end{gathered} \tag{B.1}$$

define the objective function

$$\Phi = t_f \tag{B.2}$$

and the final conditions function as

$$\Omega = h - h_f = 0 \tag{B.3}$$

There is no path constraints, so

$$\Psi = 0 \tag{B.4}$$

Then the lagrange multipliers with boundary conditions are defined as

$$\begin{aligned} \lambda'_\Phi(t_f) &= (\tfrac{\partial \Phi}{\partial x})^*_{t=t_f} = (\tfrac{\partial \Phi}{\partial V} \quad \tfrac{\partial \Phi}{\partial \gamma} \quad \tfrac{\partial \Phi}{\partial h})^*_{t=t_f} = [0 \quad 0 \quad 0] \\ \lambda'_\Omega(t_f) &= (\tfrac{\partial \Omega}{\partial x})^*_{t=t_f} = (\tfrac{\partial \Omega}{\partial V} \quad \tfrac{\partial \Omega}{\partial \gamma} \quad \tfrac{\partial \Omega}{\partial h})^*_{t=t_f} = [0 \quad 0 \quad 1] \end{aligned} \tag{B.5}$$

Integrate λ_Φ and λ_Ω backward with stored state variables in step (1).

3. At the same time, calculate the quantities $\lambda_{\Phi\Omega}$ and $\lambda_{\Phi\Omega}$ defined as

$$\begin{aligned} \dot{\Phi} &= (\tfrac{\partial \Phi}{\partial t} + \tfrac{\partial \Phi}{\partial x} f)^*_{t=t_f} = 1 \\ \dot{\Omega} &= (\tfrac{\partial \Omega}{\partial t} + \tfrac{\partial \Omega}{\partial x} f)^*_{t=t_f} = \dot{h_f} \\ \dot{\Psi} &= 0 \\ \lambda_{\Phi\Omega} &= \lambda_\Phi - \tfrac{\dot{\Phi}}{\dot{\Omega}} \lambda_\Omega = -\tfrac{\lambda_\Omega}{\dot{h_f}} \\ \lambda_{\Psi\Omega} &= \lambda_\Psi - \tfrac{\dot{\Psi}}{\dot{\Omega}} \lambda_\Omega = [0 \quad 0 \quad 0]' \end{aligned} \tag{B.6}$$

4. Also, integrate the following numbers simultaneously with step(2)

$$\begin{aligned} I_{\Psi\Psi} &= \textstyle\int_{t_0}^{T} \lambda'_{\Psi\Omega} G W^{-1} G \lambda_{\Psi\Omega} dt \\ I_{\Psi\Phi} &= \textstyle\int_{t_0}^{T} \lambda'_{\Psi\Omega} G W^{-1} G \lambda_{\Phi\Omega} dt \\ I_{\Phi\Phi} &= \textstyle\int_{t_0}^{T} \lambda'_{\Phi\Omega} G W^{-1} G \lambda_{\Phi\Omega} dt \end{aligned} \tag{B.7}$$

where G is

$$G = \frac{\partial f}{\partial n} = \begin{bmatrix} \frac{\partial f_1}{\partial n} \\ \\ \frac{\partial f_2}{\partial n} \\ \\ \frac{\partial f_3}{\partial n} \end{bmatrix} = \begin{bmatrix} -\frac{4ngW}{\rho C_{L\alpha} S V^2} \\ \\ \frac{\partial g}{\partial V} \\ \\ 0 \end{bmatrix} \tag{B.8}$$

and W is the pre-selected weighting matrix.

5. Select reasonable valuable dP and the correction of control variable value δn is calculated as

$$\delta n(t) = \pm W^{-1} G' \lambda_{\Phi\Omega} [\frac{(dP)^2}{I_{\Phi\Phi}}]^{\frac{1}{2}} \tag{B.9}$$

6. The new control variable history is obtained as

$$n(t) = n^*(t) + \delta n(t) \tag{B.10}$$

and repeate process (1)-(5) until convergent.

APPENDIX C

PROPORTIONAL NAVIGATION GUIDANCE LAW ALGORITHM

```
% This program is to use the Proportional Navigation Proportional

% law to simulate the three-dimensional intercepting trajectory

clear all close all

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

global K Vt

% set initial condition of target and interceptor

V_air_sea = 1116.9;

V_air_h = 1077.9;

Rt0 = [10000;-10000;10000];

Mt= 0.6;

Vty = V_air_h*Mt;

Vt = [0;Vty;0];

Rm0 = [0;-10000;0];

Mm = 0.4;

Vm0_abs = V_air_sea*Mm;

gama0 = 0.22;

kai0 = 0;
```

```
Vm0 = [Vm0_abs*sin(gama0);

Vm0_abs*cos(gama0)*cos(kai0);...

       Vm0_abs*cos(gama0)*sin(kai0);];

% set the constant effective navigation ratio

K = 5;

% set integration step size

dt = 1;

% set integration initial condition

Rt = Rt0;

Rm = Rm0;

Vm = Vm0;

Rtm = Rt-Rm;

Vtm = Vt-Vm;

Rtm_abs =norm(Rtm);

Vc =-(Rtm(1)*Vtm(1)+Rtm(2)*Vtm(2)+Rtm(3)*Vtm(3))/Rtm_abs;

% integrate the velocity and coordinates forward using trapezoidal method

for i = 1:1000000

    % set loop stop condition, that is when the close velocity changes its

    % sign

    if Vc<=0
    break
end
if Rtm_abs <1000
    dt=0.001;
else
    dt=0.1;
end
Rt_old = Rt;
Rm_old = Rm;
Vm_old = Vm;
Rtm = Rt-Rm;
Vtm = Vt-Vm;
Rtm_abs = norm(Rtm);
Vc = -(Rtm(1)*Vtm(1)+Rtm(2)*Vtm(2)+Rtm(3)*Vtm(3))/Rtm_abs;
A = acc(Rm,Vm,Rt);
Rt = Rt+dt*Vt;
Rm = Rm+dt*Vm;
Vm = Vm+dt*A;
Rtm = Rt-Rm;
Vtm = Vt-Vm;
```

```
    Rtm_abs = norm(Rtm);

    Vc = -(Rtm(1)*Vtm(1)+Rtm(2)*Vtm(2)+Rtm(3)*Vtm(3))/Rtm_abs;

    A = acc(Rm,Vm,Rt);

    Rt = 0.5*(Rt_old+Rt+dt*Vt);

    Rm = 0.5*(Rm_old+Rm+dt*Vm);

    Vm = 0.5*(Vm_old+Vm+dt*A);

    Rmx(i) = Rm(1)/10000;

    Rmy(i) = Rm(2)/10000;

    Rmz(i) = Rm(3)/10000;

    Rtx(i) = Rt(1)/10000;

    Rty(i) = Rt(2)/10000;

    Rtz(i) = Rt(3)/10000;

    if i==1

        t(i) = dt;

    else

        t(i) = dt+t(i-1);

    end

    vi(i) = norm(Vm);

    acel(i) = norm(A);

end
```

```
time = t(i-1);

fprintf('The interception time usd is:\n[%3.6f] seconds\n ', time);

fprintf('The final miss distance is:\n[%3.6f] feet\n ', Rtm_abs);

figure (1)

plot3(Rmx,Rmy,Rmz)

hold on

plot3(Rtx,Rty,Rtz)

xlabel('X(10000ft)')

ylabel('Y (10000ft)')

zlabel('Altitude (10000ft)')

figure(2)

plot(t,vi)

ylabel('flight speed (ft/sec)')

% This function calculate the acceleration command of the interceptor at

% each step time

function A=acc(Rm,Vm,Rt)

global K Vt

% calculate the relative position and relative velocity

Rtm=Rt-Rm;

Vtm=Vt-Vm;
```

```
% calculate the closing velocity and line-of-sight angular velocity

Rtm_abs=norm(Rtm);

dRtm=-(Rtm(1)*Vtm(1)+Rtm(2)*Vtm(2)+Rtm(3)*Vtm(3))/Rtm_abs;

omega=cross(Rtm,Vtm)/Rtm_abs^2;

% calculate the acceleration

A=-K*dRtm/Rtm_abs*cross(Rtm,omega);
```

Appendix D

Chebyshev Pseudospectral Collocation and Nonlinear Programming Codes in Solving Brachistochrone Problem

```
% This is the major function which will use SNOPT to run the nonlinear
% programming solver
clear all Name = 'Brachistochrone Problem';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global nc_defect ndiffeq nnodes nlp_state ncv nlpv x_0 xf2 xf1 x01
x02
% number of differential equations
ndiffeq = 2;
% number of control variables
ncv = 1;
% number of discretization nodes
nnodes = 25;
% number of state nlp variables
nlp_state = ndiffeq * nnodes;
% number of control nlp variables
nlp_control = ncv * nnodes;
```

```
% total number of nlp variables

nlpv = nlp_state + nlp_control

% number of state vector defect equality constraints

nc_defect = nlp_state;

% number of auxiliary equality constraints (boundary conditions)

nc_aux = 4;

% total number of equality constraints

nc_total = nc_defect + nc_aux;

% set the initial input of NLP variables

x01 = 0;  % starting point

x02 = 0;

xf1 = 5;  % ending point

xf2 = -1;

% guess the initial input of the control variables

u_guess=atan((xf2-x02)/(xf1-x01));

% guess the initial input of the state variables

for i=1:1:nnodes

    x_0(2*i-1)=x01+(xf1-x01)/(nnodes-1)*(i-1);

    x_0(2*i)=x02+(xf2-x02)/(nnodes-1)*(i-1);

    x_0(nlp_state+i)=u_guess;
```

```
end

x_0(nlpv+1)=1.3;    % guess the final time

% set the NLP variables bounds

for i=1:1:nnodes

    x_L(2*i-1)=-20;

    x_U(2*i-1)=20;

    x_L(2*i)=-20;

    x_U(2*i)=20;

    x_L(nlp_state+i)=-2*pi;

    x_U(nlp_state+i)=2*pi;

end

x_L(nlpv+1)=0;

x_U(nlpv+1)=10;

% set the nonlinear function constraints

for i=1:1:nc_total

    c_L(i)=0;

    c_U(i)=0;

end fLowBnd = 0;

Prob = conAssign('trapm3_f', [], [], [], x_L, x_U,Name, x_0,[],

fLowBnd, [], [], [], 'trapm3_cbr', [], [], [], c_L, c_U);
```

```
Prob.Warning = 0;    % Turning off warnings.

Result = tomRun('snopt', Prob, 1);

X_OUT= Result.x_k;

for i=1:1:nnodes

    yff(i)=X_OUT(2*i);

    xff(i)=X_OUT(2*i-1);

    uff(i)=X_OUT(nlp_state+i);

end

figure (1)  % plot the trajectory

plot(xff,yff)

function c = trapm3_cbr (x,prob)

% equality constraints

% Chebyshev pseudospectral collocation method

% inputs: x = current nlp variable values

% outputs: c = vector of nonlinear equality constraints evaluated at x

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

global nc_defect ndiffeq nnodes nlp_state ncv nlpv xf2 xf1 x01 x02

% compute state vector defect equality constraints

t_total = x(nlpv+1);

if t_total==0
```

```
   t_total=0.001;

end

for k = 1:1:nnodes

    % state vector elements

    % reset to previous node

    nks = (k - 1) * ndiffeq;

    for i = 1:1:ndiffeq

        xk(i) = x(nks + i);

    end

    % control variable elements

    % reset to previous node

    nkc = nlp_state + (k - 1) * ncv;

    for i = 1:1:ncv

        uk(i) = x(nkc + i);

    end

    % save one node states in one column of a matrix

    sx(:,(nnodes-k+1)) = xk';

    % compute state vector defects for current node

    f = deriv2 (xk, uk);

    % save one node f value in one column of a matrix
    sf(:,(nnodes-k+1)) = f';

end

% calculte the Chebyshev-Gauss-Lobatto differentiation matrix D

  N = nnodes-1;

  t = cos(pi*(0:N)/N)';

  p = [2; ones(N-1,1); 2].*(-1).^(0:N)';

  T = repmat(t,1,N+1);

  dT = T-T';

  D  = (p*(1./p)')./(dT+(eye(N+1)));      % off-diagonal entries

  D  = D - diag(sum(D'));                 % diagonal entries

% compute the defects array for current node

for k=1:1:nnodes

    nks = (k - 1) * ndiffeq;

    for i=1:1:ndiffeq

        d(i,k) = D(k,:)*sx(i,:)'; % calculate the derivative of f at node k

        resid(nks + i) = 2/t_total*d(i,k)-sf(i,k);

    end

end

% set active defect constraints

% (offset by 1)
```

```
for i = 1:1:nc_defect

    c(i) = resid(i);

end

% current final state vector

xfinal(1) = x(nlp_state - 1);

xfinal(2) = x(nlp_state);

% current initial state vector

xinitial(1) = x(1);

xinitial(2) = x(2);

% initial boundary conditions

c(nc_defect + 1) = xinitial(1)-x01;

c(nc_defect + 2)=xinitial(2)-x02;

% final boundary conditions

c(nc_defect + 3) = xfinal(1)-xf1;

c(nc_defect + 4) = xfinal(2)-xf2;

c = c’; % transpose

function f = trapm3_f (x,prob)

% objective function

% inputs: x = current nlp variable values

% outputs: f = objective function evaluated at x
```

```
global nlpv

tfinal = x(nlpv+1);

f = tfinal; % objective function (maximize final time)


function xdot=deriv2(x,u)

% equations of motion

% input: x = current state vector, u = current control vector

% output: xdot = derivative of x and y

global x02

% constant parameter

g = 9.82;

% evaluate equations of motion at current conditions

xdot(1) = sqrt(2*g*(x02-x(2)))*cos(u);

xdot(2) = sin(u)*sqrt(2*g*(x02-x(2)));
```