

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600



**COLLOCATION WITH NONLINEAR PROGRAMMING
FOR TWO-SIDED FLIGHT PATH OPTIMIZATION**

BY

KAZUHIRO HORIE

**B.Engr., Nagoya University, 1987
M.S., University of Illinois at Urbana-Champaign, 1998**

THESIS

**Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aeronautical and Astronautical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2002**

Urbana, Illinois

UMI Number: 3044118



UMI Microform 3044118

**Copyright 2002 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.**

**ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346**

© Copyright by Kazuhiko Horie, 2002

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

GRADUATE COLLEGE

FEBRUARY 2002

date

WE HEREBY RECOMMEND THAT THE THESIS BY

KAZUHIRO HORIE

ENTITLED COLLOCATION WITH NONLINEAR PROGRAMMING
FOR TWO-SIDED FLIGHT PATH OPTIMIZATION

**BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY**

Bruce A. Conway

Director of Thesis Research

M.F. Pruzzo

Head of Department

Committee on Final Examination†

Bruce A. Conway

Chairperson

[Signature]

*(Victor Gorovitz)
(David E. Bellamy)*

†Required for doctor's degree but not for master's.

Abstract

This research successfully develops a new numerical method for the problem of two-sided flight path optimization, that is, a method capable of finding trajectories satisfying the necessary condition of an open-loop representation of a saddle-point trajectory. The method of direct collocation with nonlinear programming is extended to find the solution of a zero-sum two-person differential game by incorporating the analytical optimality condition for one player into the system equations. The new method is named semi-direct collocation with nonlinear programming (semi-DCNLP). We apply the new method to a variety of problems of increasing complexity; the dolichobrachistochrone, a problem of ballistic interception, the homicidal chauffeur problem and minimum-time spacecraft interception for optimally evasive target, and thus verify that the method is capable of identifying saddle-point trajectories. While the method is quite robust, ambitious problems require a reasonable initial guess of the discretized solution from which the optimizer may converge. A method for generating a good initial guess, requiring no *a priori* information about the solution, is developed using genetic algorithms. The semi-DCNLP, in combination with the genetic algorithm-based pre-processor, is then used to solve a very complicated pursuit-evasion problem; optimal air combat for realistic fighter aircraft models in three dimensions. Characteristics of the optimal air combat maneuvers for both aircraft are identified for many different initial conditions.

to Setsuko, Minako and Yusuke Alexander

Acknowledgments

I would like to express my appreciation for my adviser, Prof. Bruce Conway. He leads my research with much valuable advice, which is sometimes communicated beyond Pacific Ocean. Next, I would like to express my thankfulness for my committee members, Prof. David Goldberg, Prof. Sri Namachchivaya and Prof. Victoria Coverstone. A debt of gratitude is owed to Dr. David Carroll for providing a Fortran code of a genetic algorithm on the web. Special thanks to my friends in the AAE: Bruce Powers, Bill Hartmann, Bill Mason, Chet Hamill, Jennifer Hergens, Holly Gubacki, Dr. Jyun-Jye Chen, Dr. Jang-won Jo and Kathy Larson. They always support me when I have any difficulties.

I also would like to express my gratitude to Dr. Yasue, Mr. Beppu and many people who encourage me to complete my research in Technical Research and Development Institute, Japan Defense Agency.

I am grateful for my father, Hiroshi Horie, my mother Takako Horie. Finally, I would like to thank to my family, Setsuko, Minako and Yusuke Alex, for their understanding and patience for my research.

Table of Contents

List of Figures	viii
List of Tables	xvi
List of Symbols	xvii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Outline of Thesis	6
Chapter 2: One-Sided Flight Path Optimization	8
2.1 Optimal Control Problem	8
2.2 Direct Collocation with Nonlinear Programming	11
2.3 Application to Air Combat Problem	19
Chapter 3: Two-Sided Flight Path Optimization	36
3.1 A Zero-Sum Two-Person Differential Game	36
3.2 Necessary Condition for Saddle-Point Trajectory	39
3.3 Semi-Direct Collocation with Nonlinear Programming	40
3.4 Application and Verification of the Method	45
3.4.1 Dolichobrachistochrone	45
3.4.2 Ballistic Interception	52
3.4.3 Homicidal Chauffeur Problem	58
3.4.4 Minimum Time Spacecraft Interception for Optimally Evasive Target	71
Chapter 4: Pre-Processing for Collocation with Nonlinear Programming	81
4.1 Desired Characteristics for Pre-Processing Algorithm	81
4.2 Simple Genetic Algorithm as a Pre-Processing Algorithm	82
4.2.1 Simple Genetic Algorithm Overview	82
4.2.2 Pre-Processing Algorithm	85

4.3 Efficiency of the Pre-Processing Algorithm	96
Chapter 5: Air Combat Problem	103
 5.1 Problem Definition	103
 5.2 Simplified Air Combat	108
 5.2.1 Two-Dimensional Problem	108
 5.2.2 Three-Dimensional Problem	117
 5.3 Realistic Air Combat	128
 5.3.1 Two-Dimensional Problem	128
 5.3.2 Three-Dimensional Problem	137
Chapter 6: Conclusions	162
 6.1 Summary of Research	162
 6.2 Recommendations for Future Research	165
References	168
Vita	172

List of Figures

Figure 2.1	Node and Segment on the Time History of a Variable	12
Figure 2.2	Collocation Points on the High Resolution DCNLP	16
Figure 2.3	Time Sequence of the Evasive-Offensive Maneuver	20
Figure 2.4	Velocity History for Case with "Pointing" Constraint Alone	26
Figure 2.5	Control Variable History for Case with "Pointing" Constraint Alone .	27
Figure 2.6	Trajectories of the Evader and Pursuer for Case with "Pointing" Constraint Alone (Max. Thrust Vectoring Angle = 20°)	28
Figure 2.7	Velocity History for Case with "Pointing" and "Kinematic" Constraints	29
Figure 2.8	Control Variable History for Case with "Pointing" and "Kinematic" Constraints	30
Figure 2.9	Control Variable History for Case with "Pointing", "Kinematic" and "Following" Constraints	31
Figure 2.10	Trajectories of the Evader and Pursuer for Case with "Pointing", "Kinematic" and "Following" Constraints (Max. Thrust Vectoring Angle = 20°)	31
Figure 2.11	Sensitivity of Maneuver Time to Thrust Vectoring Capability	32
Figure 2.12	Control Variable History in Case I	33
Figure 2.13	Velocity History in Case IV	34
Figure 3.1	Dolichobrachistochrone Problem	46
Figure 3.2	Saddle-Point Trajectories of the Dolichobrachistochrone	50

Figure 3.3 Time History of Control Variable α of Dolichobrachistochrone (($x_{10}, x_{20})=(3.5, 1.5)$)	50
Figure 3.4 Time History of Control Variable β of Dolichobrachistochrone (($x_{10}, x_{20})=(3.5, 1.5)$)	51
Figure 3.5 Time History of Ratio of Adjoint Variables of Dolichobrachistochrone (($x_{10}, x_{20})=(3.5, 1.5)$)	51
Figure 3.6 Ballistic Interception	52
Figure 3.7 Saddle-Point Trajectories for Ballistic Interception	56
Figure 3.8 Time History of Flight Path Angle for Ballistic Interception	57
Figure 3.9 Time History of Adjoint Variables for Ballistic Interception	57
Figure 3.10 Homicidal Chauffeur Problem in Relative Coordinates	59
Figure 3.11 Homicidal Chauffeur Problem in Fixed Coordinates	60
Figure 3.12 A Group of Saddle-Point Trajectories of the Homicidal Chauffeur Problem in Relative Coordinates	62
Figure 3.13 Saddle-Point Trajectories of the Homicidal Chauffeur Problem in Fixed Coordinates (Non Singular Surface)	65
Figure 3.14 Time History of Control Variable for Pursuer of Homicidal Chauffeur Problem in Fixed Coordinates (Non Singular Surface)	66
Figure 3.15 Time History of Control Variable for Evader of Homicidal Chauffeur Problem in Fixed Coordinates (Non Singular Surface)	66
Figure 3.16 Saddle-Point Trajectories of Homicidal Chauffeur Problem in Fixed Coordinates (Universal Surface).....	67
Figure 3.17 Time History of Control Variable for Pursuer of Homicidal Chauffeur Problem in Fixed Coordinates (Universal Surface)	68

Figure 3.18 Time History of Control Variable for Evader of Homicidal Chauffeur Problem in Fixed Coordinates (Universal Surface)	68
Figure 3.19 Saddle-Point Trajectories of Homicidal Chauffeur Problem in Fixed Coordinates (Transition Surface)	69
Figure 3.20 Time History of Control Variable for Pursuer of Homicidal Chauffeur Problem in Fixed Coordinates (Transition Surface)	70
Figure 3.21 Time History of Control Variable for Evader of Homicidal Chauffeur Problem in Fixed Coordinates (Transition Surface)	70
Figure 3.22 Spacecraft Interception for Optimally Evasive Target	72
Figure 3.23 Game of Values for Spacecraft Interception/Evasion Problem.....	75
Figure 3.24 A Group of Saddle-Point Trajectories for the Spacecraft Interception/Evasion Problem	76
Figure 3.25 Thrust Angle Histories for Spacecraft Interception/Evasion Problem ($\theta_0=0.4\text{rad}$)	77
Figure 3.26 Line-of-Sight from Spacecraft to Target for Spacecraft Interception/Evasion Problem	77
Figure 3.27 Saddle-Point Trajectories for Spacecraft Interception/Evasion Problem (for Bounded Target Thrust Angle)	79
Figure 3.28 Thrust Angle Histories for Spacecraft Interception/Evasion Problem (for Bounded Target Thrust Angle)	79
Figure 3.29 Saddle-Point Trajectories for Spacecraft Interception/Evasion Problem (for Bounded Spacecraft and Target Thrust Angles)	80
Figure 3.30 Thrust Angle Histories for Spacecraft Interception/Evasion Problem (for Bounded Spacecraft and Target Thrust Angles)	80
Figure 4.1 Relationship between Optimized Parameters and String	84

Figure 4.2	Convergence History of the Simple TPBVP.....	88
Figure 4.3	Relationship between String and Parameters in Example Problem 2 ...	89
Figure 4.4	Convergence History for Maximum Velocity Transfer to a Rectilinear Path	91
Figure 4.5	Relationship between String and Parameters in Homicidal Chauffeur Problem	93
Figure 4.6	Convergence History of Homicidal Chauffeur Problem	94
Figure 4.7	Trajectories at Initial Guess and Saddle-Point	95
Figure 4.8	Structure of Hybrid Algorithm of GA and Local Optimizer.....	96
Figure 4.9	Convergence Histories for Standard and Hybrid GA Pre-Processors ..	100
Figure 4.10	Convergence Histories for Standard and Hybrid GA Pre-Processors by Generation	101
Figure 5.1	Approximate Drag Polar of F-16A Aircraft.....	110
Figure 5.2	Convergence History of GA Pre-Processing for Simplified Two-Dimensional Air Combat	114
Figure 5.3	Histories of Adjoint Variables of Saddle-Point Trajectories for Two-Dimensional Simplified Air Combat (Nominal Case)	115
Figure 5.4	Saddle-Point Trajectories for Simplified Two-Dimensional Air Combat.....	116
Figure 5.5	Histories of Flight Path Angle for Simplified Two-Dimensional Air Combat.....	116
Figure 5.6	Horizontal Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Relative Positions	121

Figure 5.7	Vertical (X-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Relative Positions	121
Figure 5.8	Vertical (Y-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Relative Positions	122
Figure 5.9	Heading Angle Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Relative Positions	122
Figure 5.10	Flight Path Angle Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Relative Positions	123
Figure 5.11	Adjoint Variable Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Relative Positions	123
Figure 5.12	Horizontal Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Altitudes	124
Figure 5.13	Vertical (X-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Altitudes	125
Figure 5.14	Vertical (Y-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Altitudes	125
Figure 5.15	Flight Path Angle Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Altitudes	126
Figure 5.16	Velocity Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Altitudes.....	126
Figure 5.17	Adjoint Variable Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Altitudes	127
Figure 5.18	Adjoint Variable Histories for Saddle-Point Trajectories of Realistic Two-Dimensional Air Combat	133

Figure 5.19 Saddle-Point Trajectories for Realistic Two-Dimensional Air Combat.....	135
Figure 5.20 Velocity Histories for Saddle-Point Trajectories of Realistic Two-Dimensional Air Combat	135
Figure 5.21 Flight Path Angle Histories for Saddle-Point Trajectories of Realistic Two-Dimensional Air Combat	136
Figure 5.22 Angle of Attack Histories for Saddle-Point Trajectories of Realistic Two-Dimensional Air Combat	136
Figure 5.23 Saddle-Point Trajectories on a Horizontal Plane for Three-Dimensional Realistic Air Combat (Nominal Case)	144
Figure 5.24 Saddle-Point Trajectories on a Vertical (X-Z) Plane for Three-Dimensional Realistic Air Combat (Nominal Case).....	145
Figure 5.25 Saddle-Point Trajectories on a Vertical (Y-Z) Plane for Three-Dimensional Realistic Air Combat (Nominal Case)	145
Figure 5.26 Histories of Bank Angle of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case).....	146
Figure 5.27 Histories of Angle of Attack of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)	146
Figure 5.28 Histories of Normal Acceleration of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)	147
Figure 5.29 Histories of Velocity of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case).....	147
Figure 5.30 Histories of Flight Path Angle of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)	148
Figure 5.31 Histories of Heading Angle of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case).....	148

Figure 5.32 Histories of Adjoint Variables of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)	149
Figure 5.33 Saddle-Point Trajectories on a Horizontal Plane (Parameter: Initial Heading Angle of Evader)	150
Figure 5.34 Saddle-Point Trajectories on a Horizontal Plane - Detail (Parameter: Initial Heading Angle of Evader)	151
Figure 5.35 Saddle-Point Trajectories on a Vertical (X-Z) Plane (Parameter: Initial Heading Angle of Evader)	151
Figure 5.36 Saddle-Point Trajectories on a Vertical (Y-Z) Plane (Parameter: Initial Heading Angle of Evader)	152
Figure 5.37 Time to Interception (Parameter: Initial Heading Angle of Evader)...	152
Figure 5.38 Initial Heading Angle vs. Final Heading Angle (Parameter: Initial Heading Angle of Evader)	153
Figure 5.39 Saddle-Point Trajectories on a Horizontal Plane (Parameter: Initial LOS).....	154
Figure 5.40 Saddle-Point Trajectories on a Horizontal Plane - Detail (Parameter: Initial LOS).....	155
Figure 5.41 Saddle-Point Trajectories on a Vertical (X-Z) Plane (Parameter: Initial LOS)	155
Figure 5.42 Saddle-Point Trajectories on a Vertical (Y-Z) Plane (Parameter: Initial LOS).....	156
Figure 5.43 Initial LOS vs. Final Heading Angle (Parameter: Initial LOS).....	156
Figure 5.44 Time to Interception (Parameter: Initial LOS)	157
Figure 5.45 Saddle-Point Trajectories on a Horizontal Plane (Parameter: Initial Altitude of Evader)	158

Figure 5.46 Saddle-Point Trajectories on a Vertical (X-Z) Plane
(Parameter: Initial Altitude of Evader) 159

Figure 5.47 Saddle-Point Trajectories on a Vertical (Y-Z) Plane
(Parameter: Initial Altitude of Evader) 159

Figure 5.48 Altitude Difference vs. Time to Interception 160

List of Tables

Table 2.1 Data and Initial Condition for Numerical Analysis	24
Table 5.1 Reference Values for Normalization	105
Table 5.2 Typical Parameters in the Air Combat Problem.....	105

List of Symbols

c_0, \dots, c_3	coefficients of cubic polynomials in a DCNLP segment
C_D	drag coefficient
C_L	lift coefficient
D	drag
f	system equations of motion
g	algebraic constraints or gravitational acceleration
H	Hamiltonian
h	altitude
J	cost function
L	integrand in the cost function or lift
l	length of string
m	mass
Nz	normal acceleration
n	the number of the segments or population size in GA
n_g	the number of the generations
p_c	crossover probability
p_m	mutation probability
S	wing area
T	total time or thrust
t	time
u	control variables
V	value of game
v	velocity
x	state variables or downrange
x_c	state variables at the center of segment approximated by polynomials
y	crossrange
α	angle of attack
γ	flight path angle or strategy
Δt	width of time segment
δ	thrust angle
λ	adjoint variable
Φ	scalar function of terminal conditions
ϕ	function associated with state and time variables at the final time or bank angle
ρ	atmospheric density
ν	Lagrange multipliers for terminal conditions

ψ	terminal conditions or heading angle
τ	time from node or thrust vectoring angle
$(\bullet)_0$	value at the initial time
$(\bullet)_{1,i}$	value at the first collocation point in i th segment
$(\bullet)_{2,i}$	value at the second collocation point in i th segment
$(\bullet)_{c,i}$	value at the center in i th segment
$(\bullet)_{\text{EXT}}$	variables for extended system for deriving the semi-DCNLP
$(\bullet)_e$	value for evader
$(\bullet)_f$	value at the final time
$(\bullet)_i$	value at the i th node
$(\bullet)_p$	value for pursuer
$(\bullet)_t$	value for target

Chapter 1: Introduction

1.1 Background

Optimization of air combat maneuvering, a variation of flight path optimization, involves finding the best flight path so that a fighter aircraft may overcome an opposing fighter aircraft. The obtained optimal path is useful for the evaluation of aircraft performance because performance of a modern fighter aircraft in combat is determined dynamically. In addition, the requirements for the flight control system of a modern fighter aircraft are driven by the optimized flight path and control. Thus, optimization of air combat maneuvering is a potentially powerful tool or aid for fighter aircraft development. While flight path optimization is presently done *a priori*, the development of a high-speed airborne computer in the future may enable real-time calculation of optimal air combat maneuvering.

In general, a flight path optimization problem can be treated as a one-sided optimization problem (optimal control problem) or a two-sided optimization problem. The one-sided optimization problem considers only one player and has been successfully applied to a variety of applications for several decades. It minimizes or maximizes a cost function for a single aircraft's flight path, for example a minimum-time climb. On the other hand, a problem such as air combat is most accurately modeled using two competitive players. In that case a path optimization problem often becomes a two-sided optimization called a zero-sum two-person differential game. A zero-sum two-person

differential game "mini-maximizes" a cost function, which means that one of two competitive players minimizes a given cost function while another maximizes the cost function. It was originally formulated by Isaacs [1]; Bryson and Ho [2] researched the same problem as an extension of an optimal control problem.

For the past several decades, a variety of one-sided flight path optimization problems for fighter aircraft have been successfully solved. Ryan and Downing [3] evaluated the dynamic performance of fighter aircraft using flight path optimization. Well, Faber and Berger [4] showed that use of a post-stall maneuver minimizes time for a turning maneuver, a pointing maneuver, a slicing maneuver and an evasive maneuver. Murayama and Hull [5] showed that the "Cobra" maneuver, which is 'a maneuver which enables an evader to become a pursuer', can be found as the solution to a one-sided optimization problem.

One important disadvantage of a one-sided optimization is that it cannot consider the optimal maneuvering of an opposing aircraft. Therefore, it is necessary to introduce a two-sided path optimization to solve the problem of air combat with an optimally maneuvering opponent.

A variety of air combat problems modeled by a zero-sum two-person differential game have been solved analytically using simplified dynamics. One of the well-developed problems is the "homicidal chauffeur" problem, which is an analog to air combat between a low-speed, highly-maneuverable evader and high-speed less-

maneuverable pursuer. The problem was originally formulated by Isaacs [6] using aircraft models controlling only turn rate and without considering aerodynamic characteristics.

Breakwell and Merz solved the problem in two-dimensional space with a limited turn rate for both aircraft [7]. Segal and Miloh succeeded in extending the study of Breakwell and Merz to three-dimensional space [8]. One disadvantage of these analytical approaches is the quite simplified dynamics which must be assumed. A problem solved by Guelman, Shinar and Green [9] is described using a model in which aerodynamic characteristics of the pursuer are considered but not those of the evader. This problem may represent the limit of the type of problems that may be solved analytically.

An answer to this disadvantage is to formulate a problem using realistic dynamics directly and solve it numerically. However, only a limited number of studies have been done using this approach because the optimization of a realistic air combat in which both fighter aircraft maneuver optimally (using accurate aerodynamic characteristics) is very challenging to solve even numerically.

In general, numerical methods for the flight path optimization problem fall into two categories, indirect methods and direct methods. Necessary conditions for an optimal control are expressed using the Pontryagin principle with state variables and associated adjoint variables, and then form a two-point boundary value problem (TPBVP). In the indirect method, the TPBVP is solved by a boundary value problem solver such as a shooting method; the adjoint variables of the problem appear explicitly.

In the direct method the control variables or state variables (or both) are known at discrete points rather than continuously. The optimal control is found directly, i.e. without the use of the Pontryagin principle and hence without knowledge of the adjoint variables using, for example, nonlinear programming. The indirect method is less robust than the direct method because boundary value problem solvers such as shooting methods are often too sensitive to errors in initial prediction of boundary values to get convergent solutions. The principal limitation of the direct method is that for complex problems very large numbers of parameters, which are the discretized state and control variables, need to be used for reasonably accurate solutions. This makes the problem a very large nonlinear programming problem. However, recent advances in discretization algorithms and the availability of sparse NLP problem solvers are allowing more ambitious problems to be solved [10].

With regard to methods applied in recent two-sided problem solutions, Hillberg and Järmark [11] solved an air combat maneuvering problem in the horizontal plane with steady turn and realistic drag and thrust data. They used a method referred to as differential dynamic programming. In this method, the calculus of variation-based necessary conditions are not explicitly satisfied. Rather, a solution to the Hamilton-Jacobi-Bellman equation, which is a partial differential equation in terms of the cost function and its partial derivatives, is found. A pursuit-evasion problem between missile and aircraft has been solved using an indirect, multiple shooting method. The two-

dimensional case is solved by Breitner, Grimm and Pesch [12,13,14] and the three-dimensional case by Lachner, Breitner and Pesch [15]. Raivio and Ethamo [16] solved a pursuit-evasion problem for a visual identification of the target by iterating a direct method.

The method of direct collocation with nonlinear programming (DCNLP) is a direct optimization method to solve a one-sided path optimization problem numerically. It originated as a collocation method for a one-sided optimization problem in work by Dickmanns and Well [17]. However, the method was applied to the TPBVP and so would be categorized as an indirect method. Hargraves and Paris [18] converted this work into a true direct method by parameterizing the control variables and finding them using nonlinear programming, dispensing with the adjoint variables entirely. Enright and Conway [19] showed that this method has strong robustness and adaptability. Herman and Conway [20] succeeded in modifying the original DCNLP to a high resolution DCNLP using high-order polynomial approximation. Horie and Conway [21] applied the high resolution DCNLP to a complicated problem of optimal aeroassisted orbital interception, which included complications such as a wide change of atmospheric density and an aerodynamic heating constraint, and showed that the DCNLP method overcomes these complications. Through these studies, DCNLP has been demonstrated to be a powerful one-sided path optimization method.

Unfortunately, DCNLP cannot solve the differential game problem directly because the most suitable nonlinear programming solvers for DCNLP, e.g. NPSOL [22] or similar programs, can only minimize a single cost function. To overcome this problem Raivio and Ethamo [23] decomposed the pursuit-evasion problem into two sub-problems for optimal control and solved a sequence of sub-problems with a pre-specified capture point and pre-specified trajectory of one player.

In this Ph.D. Thesis research, a new, robust numerical method for zero-sum two-person differential games is developed. The new method is developed employing DCNLP, without decomposing the problem, with the hope that it will share the robustness of this method. To solve the problem, some necessary conditions are determined analytically and incorporated into the DCNLP problem. This almost invariably requires the explicit appearance of some of the system adjoint variables, something which is ordinarily avoided in the use of a “direct” solution.

1.2 Outline of Thesis

In Chapter 2, the DCNLP formulation is presented and applied to a one-sided air combat problem using a realistic aerodynamic model. This will verify the suitability of DCNLP to the flight path optimization of a realistic air combat problem. A new, two-sided optimization solver is developed on the basis of DCNLP in Chapter 3. The efficiency of this method is evaluated by solving a variety of two-person zero-sum

differential games including the well-known “homicidal chauffeur” problem. Chapter 4 presents a genetic algorithm-based pre-processing algorithm, i.e., an algorithm for finding an initial guess, for the collocation with nonlinear programming method. The major subject of this research, a realistic air combat problem modeled as a pursuit-evasion game, is discussed in Chapter 5. Finally, Chapter 6 concludes the research and provides recommendations for future research.

Chapter 2: One-Sided Flight Path Optimization

2.1 Optimal Control Problem

An optimal control problem can be defined as the problem of finding the control variable history to minimize a cost function for a system. For example, a problem of minimum-time turning of an aircraft could be an optimal control problem which finds the airplane bank angle history to minimize the time of turning. We consider a system described by equations of motion (2.2), initial condition (2.3), terminal condition (2.4), and possibly other algebraic constraints (2.5) such as control constraints and path constraints. The optimal control problem is formulated as:

$$\min_{\bar{u}} J(\bar{x}, \bar{u}, t) = \min_{\bar{u}} [\phi(\bar{x}(t_f), t_f) + \int_{t_0}^{t_f} L(\bar{x}, \bar{u}, t) dt] \quad (2.1)$$

subject to

$$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}, t) \quad (2.2)$$

$$\bar{x}(t_0) = \bar{x}_0 \quad (2.3)$$

$$\bar{\psi}(\bar{x}(t_f), t_f) = 0 \quad (2.4)$$

$$\bar{g}(\bar{x}, \bar{u}, t) \leq 0 \quad (2.5)$$

where $\bar{x}(t)$ is the vector of state variables, $\bar{u}(t)$ is the vector of control variables, t_0 is the initial time and t_f is the terminal time. A cost function J consists of ϕ , a term related to the terminal state variables, and L , a term related to interim variables. The function

$\bar{\psi}$ expresses the terminal condition and \bar{g} expresses the algebraic constraints. The initial value \bar{x}_0 is specified.

Necessary conditions are established by applying the calculus of variation and the Pontryagin principle to the problem. The following discussion derives the necessary conditions assuming no path constraints in (2.5). It is noted that the problem might become a multi-point (internal-point) boundary value problem if path constraints are included in (2.5).

The system Hamiltonian and a parameter at terminal conditions are defined as:

$$H = L + \bar{\lambda}^T \bar{f} \quad (2.6)$$

$$\Phi = \phi + \bar{v}^T \bar{\psi} \quad (2.7)$$

where $\bar{\lambda}$ is the vector of adjoint variables and \bar{v} is a set of Lagrange multipliers conjugate to the terminal conditions.

Applying the calculus of variation and the Pontryagin principle to the problem yields the following equations:

$$\dot{\bar{\lambda}} = -\left(\frac{\partial H}{\partial \bar{x}}\right)^T \quad (2.8)$$

$$\bar{u} = \arg \min_{\bar{u}} H \quad (2.9)$$

$$\bar{\lambda}(t_f) = \left(\frac{\partial \Phi}{\partial \bar{x}}\right) \quad (2.10)$$

$$\left[H + \frac{\partial \Phi}{\partial t} \right]_{t=t_i} = 0 \quad (2.11)$$

It is observed that the optimal control problem, (2.1) – (2.5), and its necessary conditions, (2.8) – (2.11), form a two-point-boundary-value problem (TPBVP). The indirect method solves the TPBVP, including all necessary conditions, numerically. Another method, the direct method, minimizes (2.1) directly for a set of parameterized control variables under constraints (2.2) – (2.5), without using the necessary conditions (2.8) – (2.11).

There are three problems associated with using the indirect method. The first is that no information is available regarding the initial values of the adjoint variables. Secondly, the indirect method requires twice the number of differential equations for the system because of the adjoint equations (2.8). It is reasonable to say that in numerical analysis a larger number of differential equations makes the problem more difficult to solve. In addition, the indirect method has to solve (2.9) either analytically or numerically. In particular the method might have to predict the switching structure of the control and/or the adjoint variables to solve the problem if path constraints are applied in (2.5). On the other hand, the principal drawback of the direct method is that large numbers of discrete state and control parameters as well as large numbers of nonlinear constraint equations are required for an accurate solution. However, recent advances in computer hardware, in addition to the development of efficient nonlinear programming solvers for sparse problems [10], diminish the difficulty of using the direct method.

In this research a direct numerical method is introduced to solve an optimal control problem. The direct collocation with nonlinear programming (DCNLP) method introduced in this chapter, a type of direct method, converts the optimal control problem into a nonlinear programming problem and thus the solution has the robustness and adaptability typical of NLP solution methods. In the following sections, the form of the DCNLP method is described and the method is applied to an air combat problem. The application is expected to provide experience in applying the DCNLP method to the problem of fighter aircraft maneuvering, including realistic lift and drag models and thus verify the DCNLP as an appropriate method to solve an optimal air combat maneuvering problem.

2.2 Direct Collocation with Nonlinear Programming

DCNLP is a direct numerical method for solving an optimal control problem. It uses collocation and nonlinear programming to solve the optimal control problem described by (2.1) - (2.5). The DCNLP method parameterizes state and control variables by discretizing them in time, as shown in Fig. 2.1, and finds optimal state and control variables using nonlinear programming. A parameter space consists of the discretized state and control variables, additional parameter such as total time, T, and is constrained by (2.2) - (2.5). However, (2.2) cannot constrain the parameter space directly because of

the system differential equations. Thus, it is required that (2.2) be approximated using an algebraic equation.

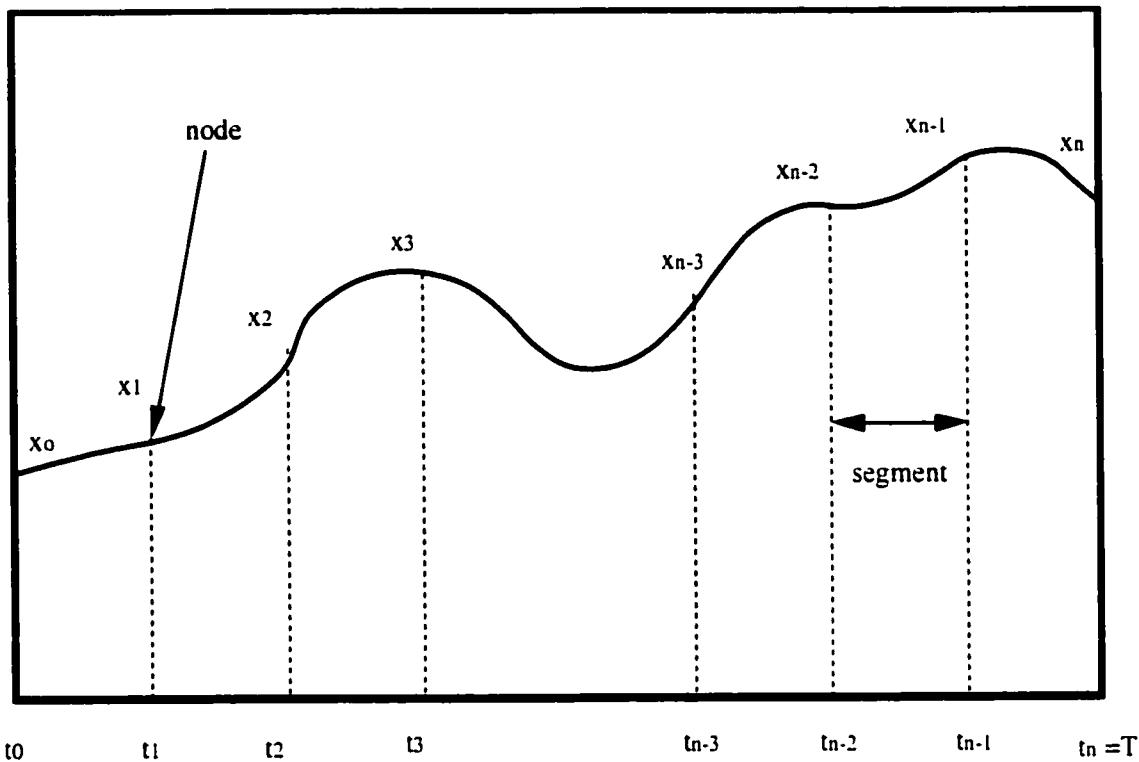


Figure 2.1 Node and Segment on the Time History of a Variable

The collocation method solves the boundary value problem by approximating the differential equation as some function. In the collocation method in the DCNLP, it is common that the differential equation is approximated by a piecewise polynomial within each segment the total time is divided into. We note here that the independent variable has been referred to as "time". This is not required for the application of the DCNLP method; it is simply the case that for the dynamical systems of interest here, i.e. real moving aircraft or spacecraft, time is the independent variable.

Hargraves and Paris [18] applied piecewise cubic polynomials for the state variables to the segments in their original DCNLP. At first, they considered a segment between the node $i-1$ and the node i , where i ranges from 1 to n . A piecewise polynomial is expressed as:

$$\tilde{x}(\tau) = \bar{c}_0 + \bar{c}_1\tau + \bar{c}_2\tau^2 + \bar{c}_3\tau^3 \quad (2.12)$$

$$\tau = t - t_{i-1} \quad (2.13)$$

where \tilde{x} represents the state variables approximated by the polynomials and τ is time measured from the node $i-1$.

They required the value of the polynomial (2.12) and its first derivative at the nodes be consistent with the discretized state variable and its derivative. Therefore,

$$\tilde{x}(0) = \bar{c}_0 = \bar{x}(t_{i-1}) \quad (2.14)$$

$$\dot{\tilde{x}}(0) = \bar{c}_1 = \bar{f}(\bar{x}(t_{i-1}), \bar{u}(t_{i-1}), t_{i-1}) \quad (2.15)$$

$$\tilde{x}(\Delta t) = \bar{c}_0 + \bar{c}_1\Delta t + \bar{c}_2\Delta t^2 + \bar{c}_3\Delta t^3 = \bar{x}(t_i) \quad (2.16)$$

$$\dot{\tilde{x}}(\Delta t) = \bar{c}_1 + 2\bar{c}_2\Delta t + 3\bar{c}_3\Delta t^2 = \bar{f}(\bar{x}(t_i), \bar{u}(t_i), t_i) \quad (2.17)$$

where $\Delta t = t_i - t_{i-1}$. Also, the "collocation point" for the piecewise polynomial is selected at $\tau = 0.5\Delta t$, i.e. the first derivative of the polynomial is required to match the value of the right hand side of the system differential equation (2.2), evaluated at the center. Then,

$$\dot{\tilde{x}}\left(\frac{1}{2}\Delta t\right) = \bar{f}\left(\tilde{x}\left(\frac{1}{2}\Delta t\right), \bar{u}(t_{i-1} + \frac{1}{2}\Delta t), t_{i-1} + \frac{1}{2}\Delta t\right) \quad (2.18)$$

Solving (2.14) - (2.17) for \bar{c}_0 , \bar{c}_1 , \bar{c}_2 and \bar{c}_3 , yields

$$\bar{c}_0 = \bar{x}(t_{i-1}) \quad (2.19)$$

$$\bar{c}_1 = \bar{f}(\bar{x}(t_{i-1}), \bar{u}(t_{i-1}), t_{i-1}) \quad (2.20)$$

$$\bar{c}_2 = -\frac{3}{\Delta t^2}(\bar{x}(t_{i-1}) - \bar{x}(t_i)) - \frac{1}{\Delta t}(2\bar{f}(\bar{x}(t_{i-1}), \bar{u}(t_{i-1}), t_{i-1}) + \bar{f}(\bar{x}(t_i), \bar{u}(t_i), t_i)) \quad (2.21)$$

$$\bar{c}_3 = \frac{2}{\Delta t^3}(\bar{x}(t_{i-1}) - \bar{x}(t_i)) + \frac{1}{\Delta t^2}(\bar{f}(\bar{x}(t_{i-1}), \bar{u}(t_{i-1}), t_{i-1}) + \bar{f}(\bar{x}(t_i), \bar{u}(t_i), t_i)) \quad (2.22)$$

Using (2.12) and (2.19) - (2.22), the state variables at $\tau=0.5\Delta t$ approximated by the polynomial are expressed as:

$$\begin{aligned} \bar{\bar{x}}_c &= \bar{\bar{x}}\left(\frac{1}{2}\Delta t\right) = \frac{1}{2}(\bar{x}(t_{i-1}) + \bar{x}(t_i)) + \frac{\Delta t}{8}(\bar{f}(\bar{x}(t_{i-1}), \bar{u}(t_{i-1}), t_{i-1}) - \bar{f}(\bar{x}(t_i), \bar{u}(t_i), t_i)) \\ &\quad (2.23) \end{aligned}$$

Finally, the collocation condition (2.18) is expressed using (2.12) and (2.19) - (2.23) as:

$$\begin{aligned} -\frac{3}{2\Delta t}(\bar{x}(t_{i-1}) - \bar{x}(t_i)) - \frac{i}{4}(\bar{f}(\bar{x}(t_{i-1}), \bar{u}(t_{i-1}), t_{i-1}) - \bar{f}(\bar{x}(t_i), \bar{u}(t_i), t_i)) \\ = \bar{f}(\bar{\bar{x}}_c, \bar{u}(t_{i-1} + \frac{1}{2}\Delta t), t_{i-1} + \frac{1}{2}\Delta t) \quad (2.24) \end{aligned}$$

The differential equation (2.2) is approximated by piecewise algebraic equations (2.24) using discretized state variables at the nodes, discretized control variables and time. It is noted that the discretized control variable at $\tau=0.5\Delta t$ is found (as the average) from the discrete controls at the left and right nodes in the work of Hargraves and Paris [18] but it is also possible to consider them to be independent optimized parameters [20]. Equations (2.1), (2.3) - (2.5) and (2.24) constitute a nonlinear programming problem.

The DCNLP method proved to be very successful at solving flight path optimization problems and this motivated work to further improve the method. Accuracy of the method was improved by using polynomials of higher degree to approximate the system state variables. A discretization using fifth-degree polynomials for each segment is used in this research, based on work by Herman and Conway [20].

The relationship between a fifth-order polynomial and the discretized state variables is shown in Fig. 2.2 for the case of a single state variable. In this case, discretized variables have to be defined not only at the nodes but also at the center of the segment because the fifth-order polynomial requires six conditions, i.e., three state variables and their derivatives, to determine its coefficients. The polynomials are defined such that the polynomial's values and derivatives at the nodes and at the center of the segment are consistent with the discretized state variables and their derivatives. In this high resolution DCNLP, two collocation points are also required. Herman and Conway used Gauss-Lobatto quadrature rules to select optimal collocation points within each segment [20]. Two collocation points, $t_{1,i}$ and $t_{2,i}$, are determined as:

$$t_{1,i} = t_{c,i} - \frac{1}{2} \sqrt{\frac{3}{7}} \Delta t \quad (2.25)$$

$$t_{2,i} = t_{c,i} + \frac{1}{2} \sqrt{\frac{3}{7}} \Delta t \quad (2.26)$$

By a derivation in the same manner as for the cubic polynomial case, the state variables at the collocation points, $t_{1,i}$ and $t_{2,i}$, are expressed by the approximating polynomial, yielding:

$$\begin{aligned}\bar{x}(t_{1,i}) &= \frac{1}{686} [(39\sqrt{21} + 231)\bar{x}(t_i) + 224\bar{x}(t_{c,i}) + (-39\sqrt{21} + 231)\bar{x}(t_{i+1}) \\ &\quad + \Delta t \{(3\sqrt{21} + 21)\tilde{f}(\bar{x}(t_i), \bar{u}(t_i)) - 16\sqrt{21} \tilde{f}(\bar{x}(t_{c,i}), \bar{u}(t_{c,i})) \\ &\quad + (3\sqrt{21} - 21)\tilde{f}(\bar{x}(t_{i+1}), \bar{u}(t_{i+1}))\}] \quad (2.27)\end{aligned}$$

$$\begin{aligned}\bar{x}(t_{2,i}) &= \frac{1}{686} [(-39\sqrt{21} + 231)\bar{x}(t_i) + 224\bar{x}(t_{c,i}) + (39\sqrt{21} + 231)\bar{x}(t_{i+1}) \\ &\quad + \Delta t \{(-3\sqrt{21} + 21)\tilde{f}(\bar{x}(t_i), \bar{u}(t_i)) - 16\sqrt{21} \tilde{f}(\bar{x}(t_{c,i}), \bar{u}(t_{c,i})) \\ &\quad + (-3\sqrt{21} - 21)\tilde{f}(\bar{x}(t_{i+1}), \bar{u}(t_{i+1}))\}] \quad (2.28)\end{aligned}$$

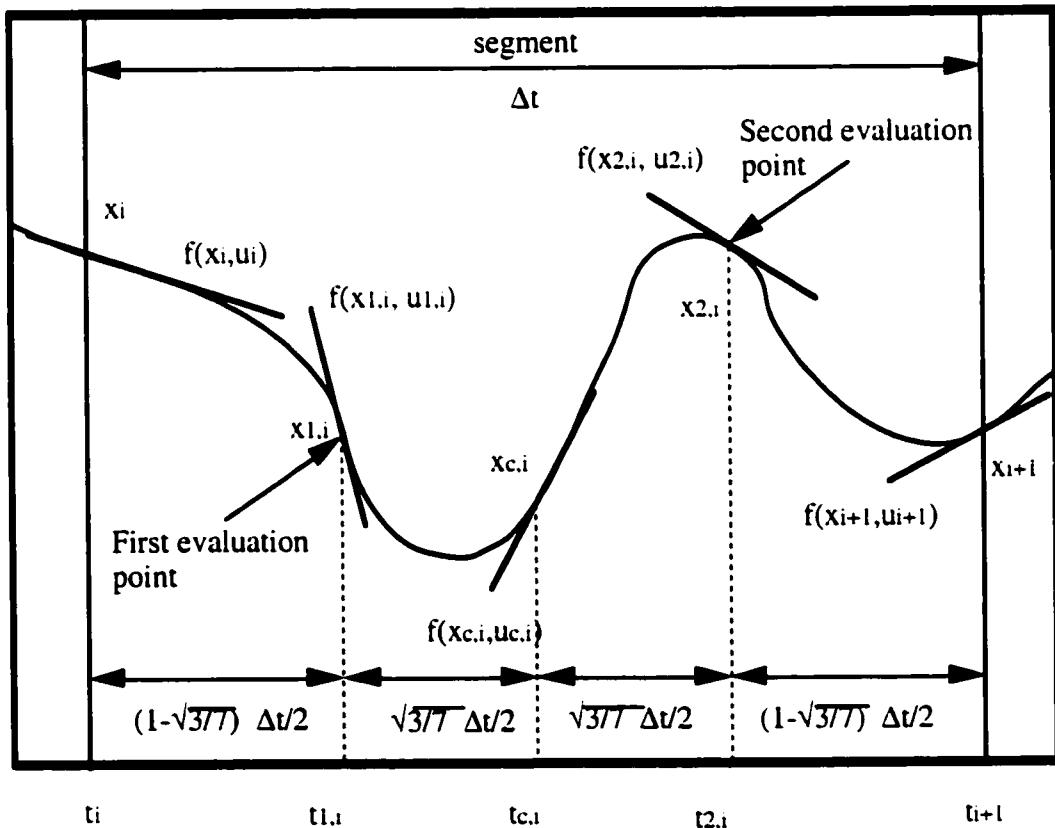


Figure 2.2 Collocation Points on the High Resolution DCNLP

The collocation condition requires satisfaction of the following conditions:

$$\dot{\tilde{x}}(t_{1,i}) - \tilde{f}(\tilde{x}(t_{1,i}), \bar{u}(t_{1,i})) = 0 \quad (2.29)$$

$$\dot{\tilde{x}}(t_{2,i}) - \tilde{f}(\tilde{x}(t_{2,i}), \bar{u}(t_{2,i})) = 0 \quad (2.30)$$

The derivatives of the fifth-order polynomial are substituted into (2.29) and (2.30) yielding:

$$\begin{aligned} & [(32\sqrt{21} + 180)\tilde{x}(t_i) - 64\sqrt{21}\tilde{x}(t_{c,i}) + (32\sqrt{21} - 180)\tilde{x}(t_{i+1})]/360 \\ & + \Delta t[(9 + \sqrt{21})\tilde{f}(\tilde{x}(t_i), \bar{u}(t_i)) + 98\tilde{f}(\tilde{x}(t_{1,i}), \bar{u}(t_{1,i})) + 64\tilde{f}(\tilde{x}(t_{c,i}), \bar{u}(t_{c,i}))] \\ & + (9 - \sqrt{21})\tilde{f}(\tilde{x}(t_{i+1}), \bar{u}(t_{i+1})) = 0 \end{aligned} \quad (2.31)$$

$$\begin{aligned} & [(-32\sqrt{21} + 180)\tilde{x}(t_i) + 64\sqrt{21}\tilde{x}(t_{c,i}) + (-32\sqrt{21} - 180)\tilde{x}(t_{i+1})]/360 \\ & + \Delta t[(9 - \sqrt{21})\tilde{f}(\tilde{x}(t_i), \bar{u}(t_i)) + 98\tilde{f}(\tilde{x}(t_{2,i}), \bar{u}(t_{2,i})) + 64\tilde{f}(\tilde{x}(t_{c,i}), \bar{u}(t_{c,i}))] \\ & + (9 + \sqrt{21})\tilde{f}(\tilde{x}(t_{i+1}), \bar{u}(t_{i+1})) = 0 \end{aligned} \quad (2.32)$$

It is noted that the discretized control variables at the nodes, at the center of the segment and at the collocation points become the optimized parameters in the nonlinear programming problem in this method.

Thus, nonlinear programming solves (2.1) in given parameter space under constraints of (2.5) for each segment, (2.31), (2.32) at each collocation point and initial and terminal conditions, (2.3) and (2.4).

NZSOL [24] is used to solve the nonlinear programming problem in this research. NZSOL locally minimizes a cost function under lower and upper limitation of parameters, linear constraints, and smooth nonlinear constraints, by using sequential

quadratic programming. An initial guess of the solution parameters is required as input for NZSOL. A cost function and nonlinear constraints are given to NZSOL as user defined subroutines. After finishing the optimization, NZSOL provides a set of optimal parameters and the value of the cost function.

NZSOL requires the calculation of the Jacobian of nonlinear constraints. The Jacobian can be numerically calculated from nonlinear constraints by using a difference method. Although this is simple, it is not optimum when the number of parameters is large because of the long calculation time required. Therefore, an analytical Jacobian facilitates effectiveness and accuracy of the solution obtained by DCNLP and thus is used in this research.

2.3 Application to Air Combat Problem

A minimum-time, vertical-plane, evasive-offensive maneuver; a maneuver in which the initial evader becomes the pursuer, is solved as an example of the application of DCNLP to an air combat problem.

The air combat scenario is established by modifying a scenario used by Murayama and Hull [5]. The initial pursuer is in flight at constant velocity and constant altitude. The initial evader, an aircraft similar to an F-16 but which has the ability to fly in the post-stall region, can maneuver in the vertical plane. At the final time, the evader transfers position and becomes able to attack the pursuer. An optimal trajectory for this maneuver is determined by how fast the evader can transfer position: i.e. the cost function for this optimization problem is the time to completion of the evasive-offensive maneuver. The maneuver is illustrated in Figure 2.3.

A set of equations of motion for the evader, a point mass model given by Vinh [25], is modified in order to allow only motion in a vertical plane and to include a thrust vectoring angle, τ :

$$\frac{dv}{dt} = \frac{1}{m}(T \cos(\alpha + \tau) - D) - g \sin \gamma \quad (2.33)$$

$$v \frac{dy}{dt} = \frac{1}{m}(T \sin(\alpha + \tau) + L) - g \cos \gamma \quad (2.34)$$

$$\frac{dx}{dt} = v \cos \gamma \quad (2.35)$$

$$\frac{dh}{dt} = v \sin \gamma \quad (2.36)$$

where angle of attack α can take a value between 0° and 90° , and τ ranges from $-\tau_{LMT}$ to τ_{LMT} .

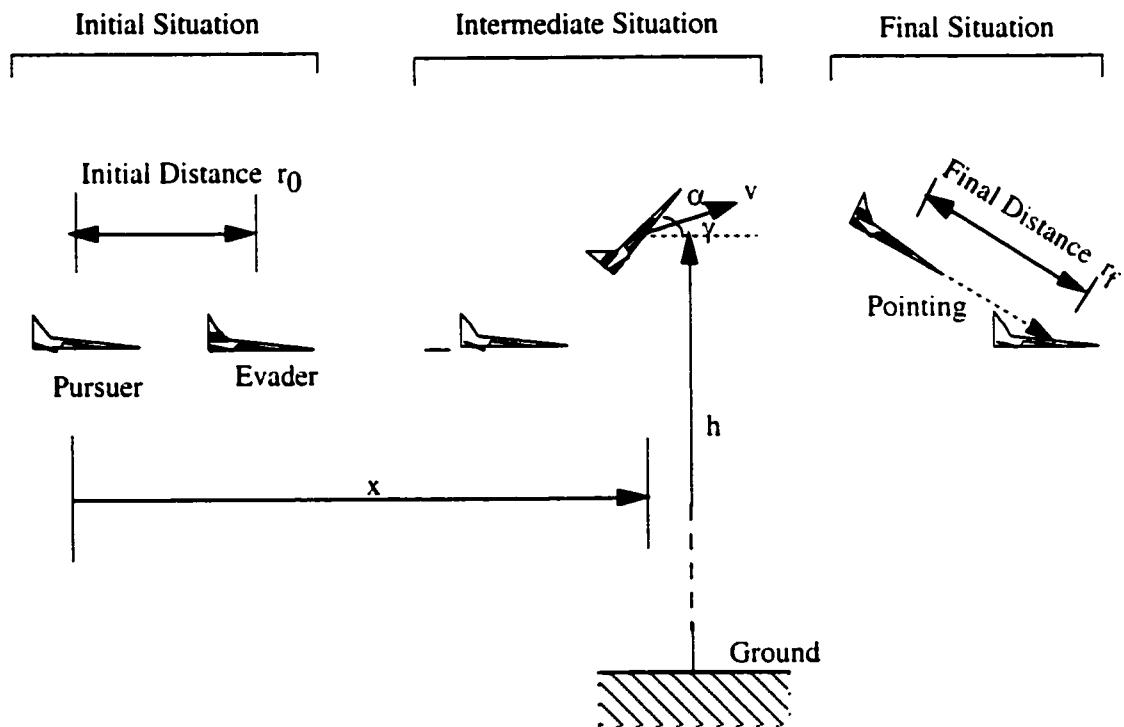


Figure 2.3 Time Sequence of the Evasive-Offensive Maneuver

Since the maneuver time is very brief the mass of the aircraft is assumed constant. The thrust is set to be equal to the weight of the aircraft. Aerodynamic data for the F-16-like maneuvering aircraft is essentially the same as used by Murayama and Hull [5]. A continuous approximation to their discrete tables of lift and drag coefficients is obtained using fourth degree, piecewise polynomials found using least square fitting

under the constraint that they be continuous and at least twice differentiable everywhere.

The lift coefficient may be represented by;

$$C_L = \begin{cases} 0.0174 + 4.3329\alpha - 1.3048\alpha^2 \\ \quad + 2.2442\alpha^3 - 5.8517\alpha^4 (0 \leq \alpha \leq \pi/6) \\ -1.3106 + 10.7892\alpha - 9.2317\alpha^2 \\ \quad - 1.1194\alpha^3 + 2.1793\alpha^4 (\pi/6 \leq \alpha \leq \pi/3) \\ 24.6577 - 71.0446\alpha + 83.1234\alpha^2 \\ \quad - 44.0862\alpha^3 + 8.6582\alpha^4 (\pi/3 \leq \alpha \leq \pi/2) \end{cases} \quad (2.37)$$

The drag coefficient may be represented by;

$$C_D = \begin{cases} 0.0476 - 0.1462\alpha + 0.0491\alpha^2 \\ \quad + 12.8046\alpha^3 - 12.6985\alpha^4 (0 \leq \alpha \leq \pi/6) \\ 0.5395 - 5.7972\alpha + 21.6625\alpha^2 \\ \quad - 21.6213\alpha^3 + 7.0364\alpha^4 (\pi/6 \leq \alpha \leq \pi/3) \\ 16.6957 - 52.5918\alpha + 67.3227\alpha^2 \\ \quad - 37.086\alpha^3 + 7.4807\alpha^4 (\pi/3 \leq \alpha \leq \pi/2) \end{cases} \quad (2.38)$$

$$\begin{aligned} L &= \frac{1}{2} \rho v^2 S C_L \\ D &= \frac{1}{2} \rho v^2 S C_D \end{aligned} \quad (2.39)$$

Atmospheric density, in the neighborhood of the nominal 10,000 ft altitude at which both airplanes are assumed to fly, is given as a function of altitude by [26]

$$\rho = \rho_s \left(1 - 0.00688 \left(\frac{h}{1000} \right) \right)^{4.256} \quad (2.40)$$

where h is in feet and ρ_s is the sea level atmospheric density.

The normal acceleration of the aircraft is restricted because of limits on structural strength and on the load a pilot can tolerate. This restriction is introduced as a path constraint:

$$N_{z_{\max}} \geq N_z = \frac{(L \cos \alpha + D \sin \alpha + T \sin \tau)}{mg} \quad (2.41)$$

This research introduces a “pointing constraint”, as a terminal constraint for the evasive-offensive maneuver. The pointing constraint requires that at the end of the maneuver the nose of the (original) evader aircraft points toward the (original) pursuer, with at least a given separation of the aircraft. This guarantees that the pilot of the evader aircraft would be able to see his opponent, i.e. that the pitch attitude of the airplane would not cause the opposing airplane to be obscured by the nose or forward fuselage of the airplane, and of course also point a weapon at the opponent. Letting \vec{r}_f represent the position vector of the pursuer relative to the evader at the final time, t_f , and with \vec{e} being a unit vector directed along the longitudinal axis of the evader airplane, then,

$$\vec{r}_f \cdot \vec{e} = \|\vec{r}_f\| \|\vec{e}\| = \|\vec{r}_f\| \quad (2.42)$$

Thus,

$$(v_e t_f - x_f) \cos(\gamma + \alpha) + (h_e - h_f) \sin(\gamma + \alpha) = \sqrt{(v_e t_f - x_f)^2 + (h_e - h_f)^2} \quad (2.43)$$

The separation constraint is

$$\|\vec{r}_f\| \geq r_{f,\min} \quad (2.44)$$

In addition to the pointing constraint, solutions are found with various additional terminal constraints. One is a constraint that the final velocity of the (original) evader, $v_e(t_f)$, be the same as that of the (original) pursuer. This prevents the evader aircraft, after it has “exchanged position” with the pursuing aircraft, from either rapidly overtaking or rapidly falling behind its opponent. Also, velocity loss should be avoided during air combat in order to maintain maneuverability for another engagement. Therefore, a constraint is introduced as a “kinematic constraint”:

$$v_e(t_f) = v_f \quad (2.45)$$

Another constraint is that the final flight path angle of the evader be zero. As we assume that the (original) pursuer flies at a constant altitude, the evader cannot continue following the pursuer if the flight path angle is not zero. Therefore, a “following constraint” is introduced as:

$$\gamma_f = 0 \quad (2.46)$$

Murayama and Hull [5] specified a final horizontal distance between the evader and pursuer as a terminal constraint. In this paper, their primary constraint is called a “horizontal distance constraint”. It is expressed as:

$$r_{hc} = v_e t_f - x_f \quad (2.47)$$

where r_{hc} is the desired horizontal separation at the final time. They also in some cases applied a constraint we term the “altitude constraint”,

$$h_f = h_e \quad (2.48)$$

requiring the evader to return to its original altitude (which is the same as the altitude of the original pursuer). In this work the horizontal distance constraint and/or the altitude constraint are applied so that current results may be compared with the corresponding Murayama and Hull [5] results.

Aircraft dimensions, physical constants and initial conditions necessary for the numerical solutions are shown in Table 2.1. The numerical solution is facilitated by converting the customary length, time and mass units into non-dimensional units also shown in Table 2.1.

Quantities	Dimensional Value	Non-dimensional Value
m Mass of aircraft	637.16 (slug)	1.0
T Thrust	20500 (slug·ft/s ²)	0.80435
ρ_s Air density at Sea Level	1.7556×10^{-3} (slug/ft ³)	1.76340×10^5
g Gravitational acceleration	32.174 (ft/s ²)	0.80435
S Wing area	300.0 (ft ²)	1.875×10^{-5}
Nz Max. normal acceleration	9.0 (G)	9.0
v_e Velocity of initial pursuer	400.0 (ft/s)	1.0
h_e Altitude of initial pursuer	10000 (ft)	2.5
r_0 Initial distance bet. evader and pursuer	1000 (ft)	0.25
$r_{f,min}$ Min. final distance bet. evader and pursuer	1000 (ft)	0.25
v_0 Initial velocity of evader	400.0 (ft/s)	1.0
γ_0 Initial flight path angle	0.0 (rad)	0.0
x_0 Initial horizontal position	1000 (ft)	0.25
h_0 Initial altitude of evader	10000 (ft)	2.5

Table 2.1 Data and Initial Condition for Numerical Analysis

Optimal flight paths are found for three different combinations of terminal constraints; all the solutions include the constraint on vertical acceleration (2.41) and all assume that the aircraft which is initially the pursuer maintains constant level flight. Both a feasibility parameter and an optimality tolerance parameter in NZSOL are set at 10^{-8} .

The use of DCNLP gives convergent solutions without special treatments for all cases.

The result of applying the “pointing constraint”, given by eqns. (2.43) and (2.44) is shown in Figures 2.4 - 2.6. These figures show that the maneuver may be divided into three phases. In the first phase, the evader has an angle of attack between 40° to 80° and establishes a climb. In the second phase, the evader aircraft has a 90° angle of attack and experiences dramatic aerodynamic deceleration. In the third phase the aircraft establishes an angle of attack of zero degrees, begins to descend, and finally points its longitudinal axis toward its target, the (original) pursuer aircraft. This result includes two of the characteristics of the cobra maneuver, the high angle of attack and the large velocity loss. The maneuver time, of about 2 seconds, is essentially the same as the cobra maneuver time described by Zagainov [27].

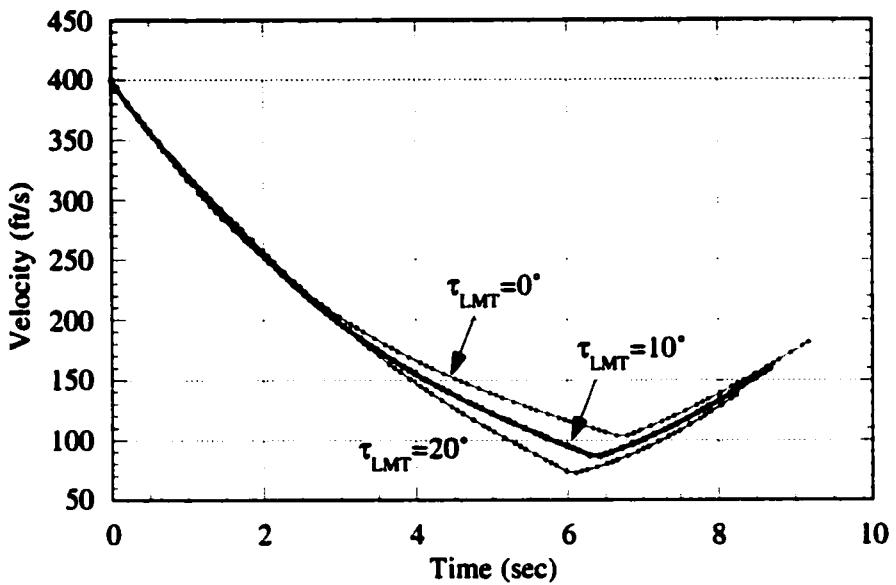


Figure 2.4 Velocity History for Case with "Pointing" Constraint Alone

Physically, the first two phases are explained as phases for rapid deceleration so that the evader will be passed by the pursuer. By establishing a positive flight path angle by the nose-up motion, the evader decelerates using drag but also gravity. After establishing sufficient flight path angle, the evader maximizes drag force with a vertical attitude. After enough deceleration, the evader takes a zero angle of attack so that it can satisfy the "pointing" terminal constraint. The deceleration, from an initial speed of 400ft/sec, is very rapid and while some speed is recovered in the third phase the airplane ends with a speed less than 200 ft/sec, as seen in Fig. 2.4. When the use of thrust

vectoring is permitted, the deceleration during the first two phases is supplemented by choosing full positive thrust vectoring angle, as seen in Fig. 2.5.

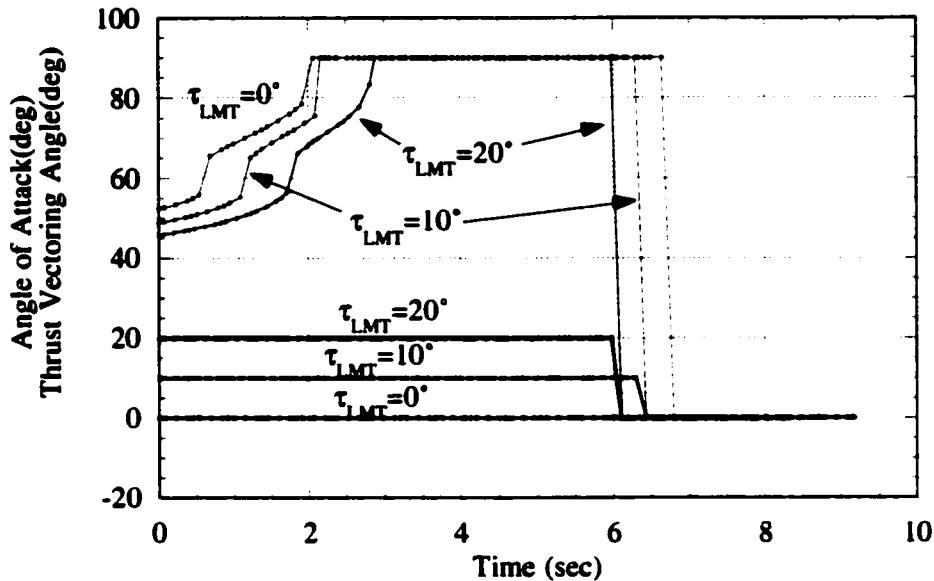
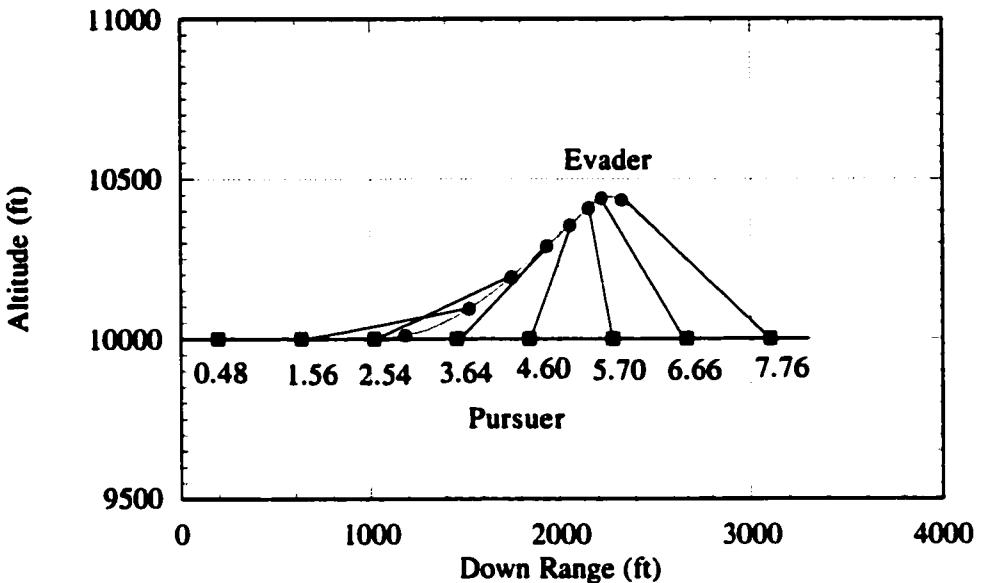


Figure 2.5 Control Variable History for Case with "Pointing" Constraint Alone

The final time for the maneuver using thrust vectoring is 8.24 sec, in the case of a thrust vectoring authority of 20°, whereas that without thrust vectoring is 9.16 sec. The maximum normal acceleration is around 5.1 G and doesn't violate the path constraint. Figure 2.6 shows the trajectories of the two aircraft, for a case using thrust vectoring capability, and how the evader becomes the pursuer.



**Figure 2.6 Trajectories of the Evader and Pursuer for Case with "Pointing" Constraint Alone
(Max. Thrust Vectoring Angle = 20°)**

Adding the kinematic constraint (2.45) to the pointing constraint eqns. (2.43)

and (2.44) yields results shown in Figures 2.7 and 2.8. The requirement of (2.45) that the final speed of the (original) evader match the constant speed of the (original) pursuer, 400 ft/sec, yields a substantially different flight path from that of the previous case, i.e. the evader descends instead of climbing. Figure 2.8 shows that there are still three different phases to the maneuver, but they are different from those of the previous case having only the pointing constraint, which was shown in Fig. 2.5. The first phase is the vertical

phase for taking a maximum drag force. The airplane “stands on its tail” immediately, rather than gradually as in Fig. 2.5. Next, the evader takes a zero angle of attack in order to descend and accelerate to satisfy the kinematic condition. During the final phase of the maneuver, the evader increases the angle of attack, aligns its velocity to the constraint value and points toward the pursuer, which requires a very high angle of attack since the pursuer is close and above.

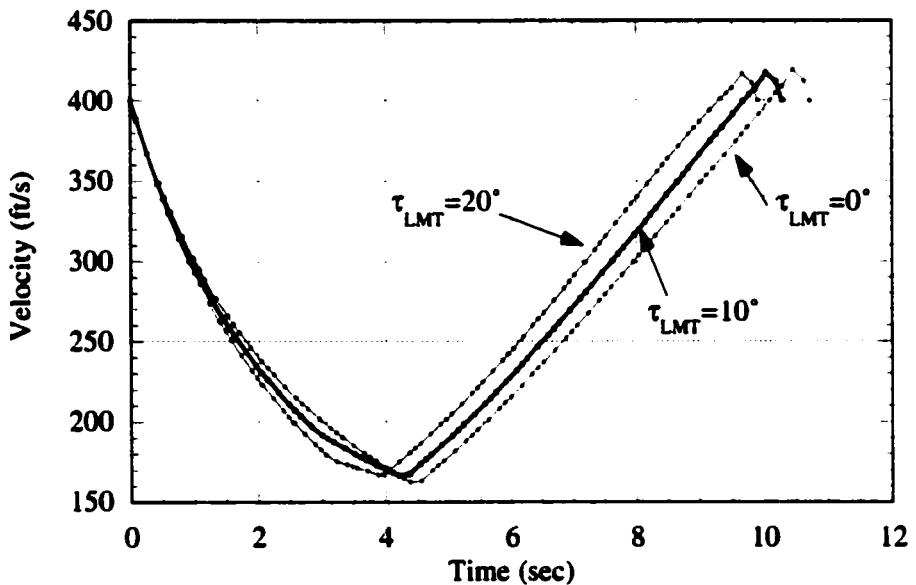


Figure 2.7 Velocity History for Case with “Pointing” and “Kinematic” Constraints

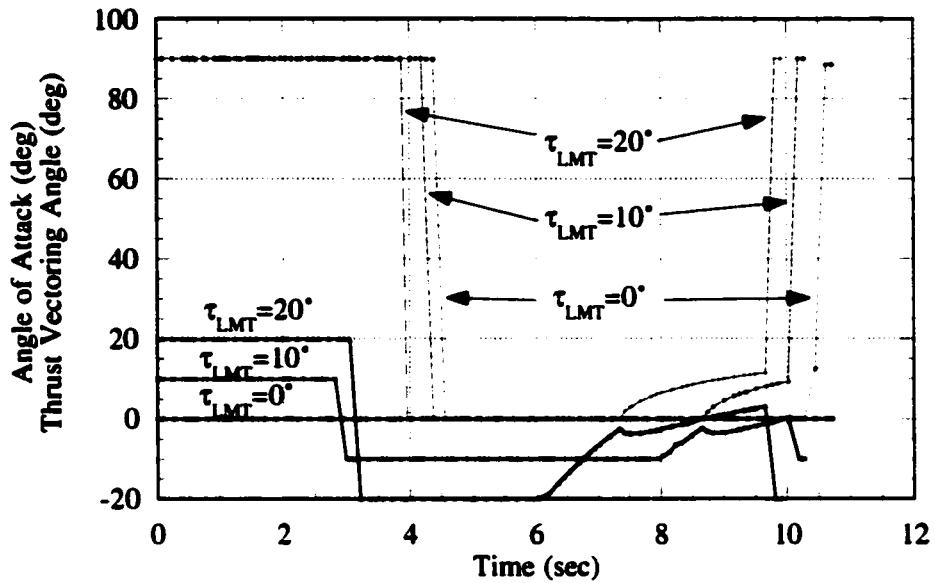


Figure 2.8 Control Variable History for Case with "Pointing" and "Kinematic" Constraints

Adding the “following constraint” (2.46) to the pointing and kinematic constraints does not dramatically change the angle of attack history or the trajectory, though it does significantly increase the maneuver time. Figure 2.9, showing angle of attack for this case, is very similar to the corresponding result shown in Fig. 2.8, except that at the end the requirement of horizontal flight reduces the final angle of attack. The trajectories of the evader and pursuer are shown in Fig. 2.10.

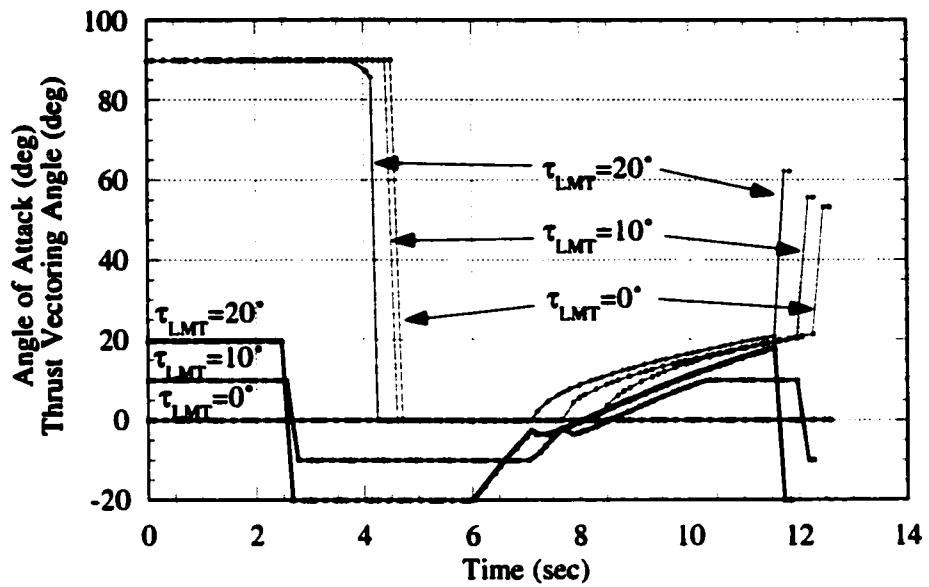


Figure 2.9 Control Variable History for Case with "Pointing", "Kinematic" and "Following" Constraints

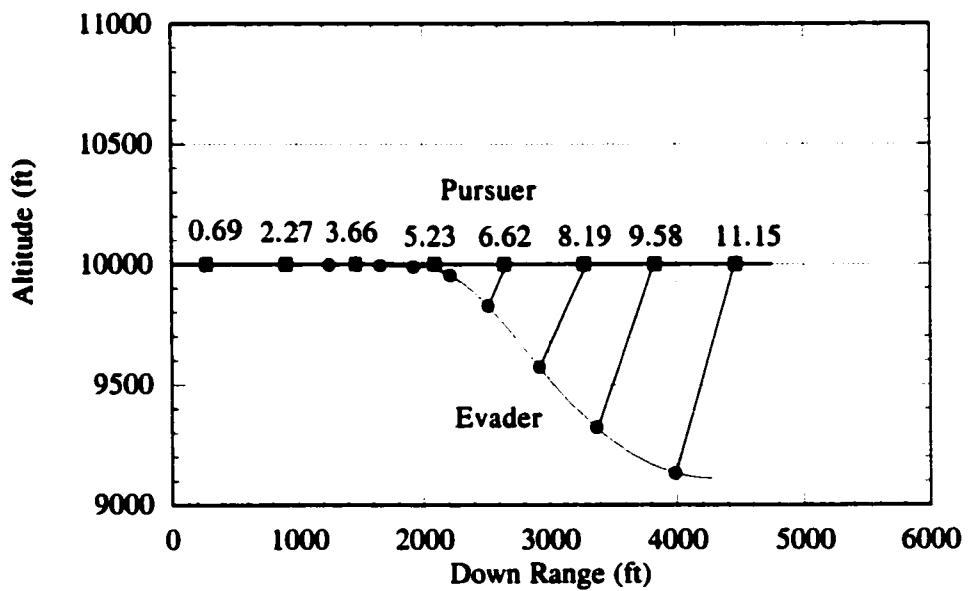


Figure 2.10 Trajectories of the Evader and Pursuer for Case with "Pointing", "Kinematic" and "Following" Constraints (Max. Thrust Vectoring Angle = 20°)

Thrust vectoring ability is used to enhance not only the combat maneuverability but also aircraft stability in the post-stall region. In addition, an increase in the thrust vectoring authority always reduces the time required for the maneuver, as shown in Figure 2.11, i.e. using the thrust vectoring ability decreases the duration that the evader is followed by the pursuer. In combat that would of course be beneficial to the evader.

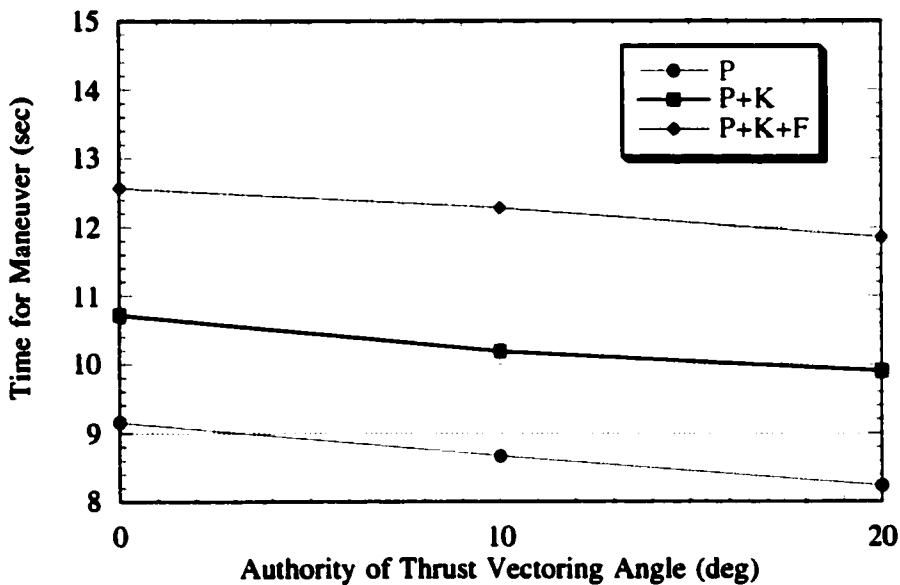


Figure 2.11 Sensitivity of Maneuver Time to Thrust Vectoring Capability

Since the aircraft, initial conditions, and terminal conditions were in some instances made the same as those Murayama and Hull [5] used in their research a direct comparison of results is possible (note that the names Case I, Case II, etc. are those used in [5]).

For Case I, employing the “horizontal distance” constraint, (2.47), the control time history in this research (shown in Figure 2.12) is qualitatively different from that of Murayama and Hull (cf. Figure 4 in [5]). The angle of attack in Fig. 2.12 increases at first and decreases toward the end whereas in the corresponding figure in the paper by Murayama and Hull the angle of attack is almost constant. Despite this, the final time for the maneuver found in this research, 8.33 sec, is essentially the same as their result.

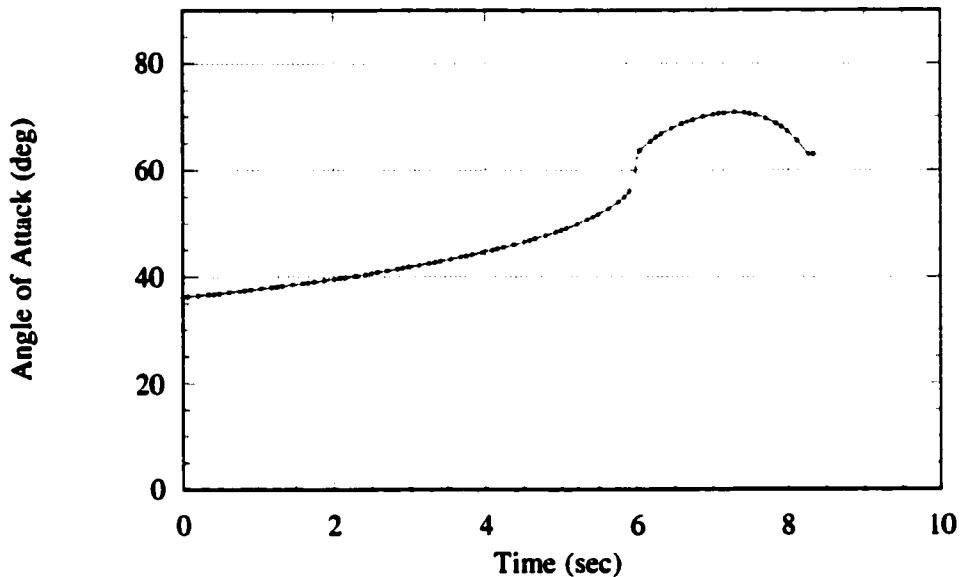


Figure 2.12 Control Variable History in Case I

For Case II, employing the “horizontal distance and following” constraints, (2.47) and (2.48), the results for the control and state variable histories are qualitatively the same. However, there is a small difference in the final time of the maneuver; 8.97 sec in this research vs. 9.3 sec in their research [5].

For Case IV, employing the “horizontal distance, kinematic, following and altitude” constraints, (2.47), (2.46), (2.45) and (2.48), the velocity history (Figure 2.13) is different from that found by Murayama and Hull (cf. Figure 7 in [5]) as is the control time history. The velocity in Fig. 2.13 decreases in the early phase and increases in the later phase whereas in the corresponding figure in the paper by Murayama and Hull the velocity increases at first, decreases then and again increases to the end. The final time for the maneuver in this research, 13.60 sec, is substantially different from (and improved from) that in their research, 16.3 sec [5].

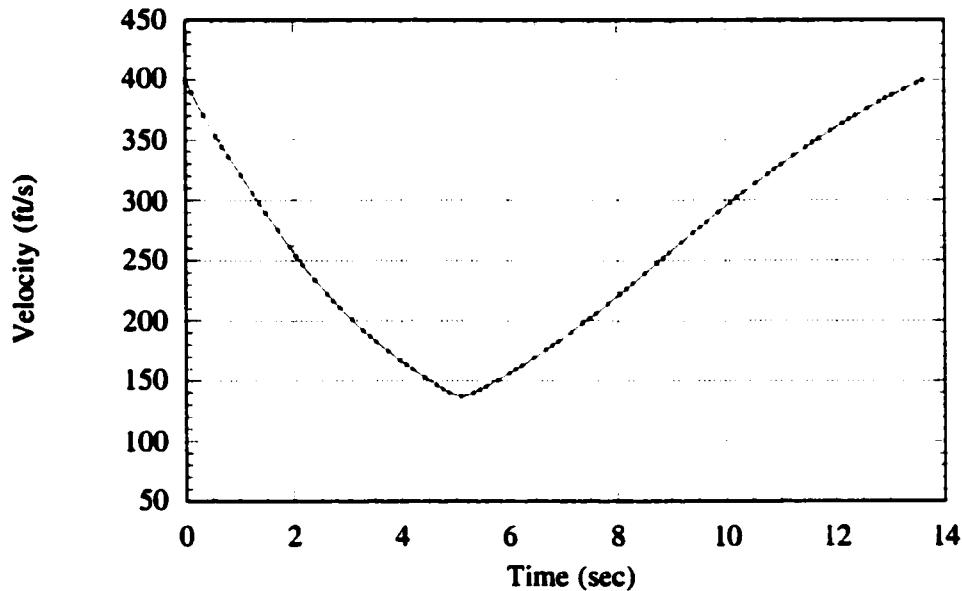


Figure 2.13 Velocity History in Case IV

These differences may be explained by the fact that the solution method in this research uses very accurate, high-order, implicit integration. Their convergence tolerance [5] is 10^{-4} , while in this research 10^{-8} is used. In addition, this research uses many more discrete control variables, 81, vs. 11 for Murayama and Hull, over the maneuver time span, thus capturing better the control time history.

Using a model aircraft similar to an F-16, but with the capability of post-stall flight, the minimum-time vertical-plane evasive-offensive maneuver has been found. The evader aircraft quickly exchanges position so that it becomes the pursuer. The optimal trajectories and the angle of attack history of the airplane qualitatively resemble those described as the “Cobra” maneuver [27]. Thrust vectoring authority reduces the flight time for the evasive-offensive maneuver decreasing the exposure of the evader to attack by the pursuer.

Perhaps the most significant result is that the DCNLP method has solved this problem successfully and robustly for a variety of terminal constraints. We thus recommend it as an appropriate method for trajectory optimization for maneuverable aircraft.

Chapter 3: Two-Sided Flight Path Optimization

3.1 A Zero-Sum Two-Person Differential Game

A conflict frequently occurs when multiple players focus on their own purposes.

Isaacs [6] formulated the optimal behaviors for two competitive players. The problem formulated is called a zero-sum two-person differential game and solved in such a way that one of two competitive players minimizes a given cost function while another maximizes the cost function. A two-sided flight path optimization problem such as air combat is often described as a zero-sum two-person differential game.

In this research, we develop a robust numerical method for zero-sum two-person differential games. The method employs DCNLP with the anticipation that it will share the robustness of the DCNLP method demonstrated in the solution of many problems [19,21]. While the work here is new, Raivio and Ethamo [23] developed a two-sided optimization algorithm using DCNLP by decomposing the problem into two one-sided optimization problems. However they do this by solving two one-sided optimization problems with a pre-specified trajectory for one player, which trajectory is then iteratively improved. The new method being developed in this research is not an iterative method; it solves for the optimal trajectories of the two players simultaneously. Some optimality conditions are determined analytically and incorporated into the DCNLP problem. Thus the method can avoid the complication of decomposition and iteration.

This almost invariably requires the explicit appearance of some of the system adjoint variables, something that is ordinarily avoided in the use of a direct solution.

In a zero-sum two-person differential game, two competitive sets of control variables, \bar{u}_p and \bar{u}_e , drive a dynamic system. In this study, the following equations of motion, (3.1) and (3.2), are considered:

$$\dot{\bar{x}}_p = \bar{f}_p(\bar{x}_p, \bar{u}_p, t) \quad (3.1)$$

$$\dot{\bar{x}}_e = \bar{f}_e(\bar{x}_e, \bar{u}_e, t) \quad (3.2)$$

with initial conditions:

$$\bar{x}_p(t_0) = \bar{x}_{p0} \quad (3.3)$$

$$\bar{x}_e(t_0) = \bar{x}_{e0} \quad (3.4)$$

It is noted that these equations of motion are often found in a pursuit-evasion game in fixed coordinates.

Terminal constraints which are functions of the states at the final time and possibly of the final time are:

$$\bar{\psi}(\bar{x}_p(t_f), \bar{x}_e(t_f), t_f) = 0 \quad (3.5)$$

where t is time, t_0 is initial time and t_f is the terminal time of a problem.

It is assumed that some control variables are bounded as:

$$\bar{u}_{p,l} \leq \bar{u}_p \leq \bar{u}_{p,u} \quad (3.6)$$

$$\bar{u}_{e,l} \leq \bar{u}_e \leq \bar{u}_{e,u} \quad (3.7)$$

Path constraints are not considered in this research.

A problem of Mayer type is considered in this research. Then, the cost function for the problem is given as a function of the state variables and terminal time in the following form:

$$J(\bar{x}_p, \bar{x}_e, \bar{u}_p, \bar{u}_e, t) = \phi(\bar{x}_p(t_f), \bar{x}_e(t_f), t_f) \quad (3.8)$$

The feedback strategies, $\bar{\gamma}_p$ and $\bar{\gamma}_e$, are introduced to determine the control variables, \bar{u}_p and \bar{u}_e , as a function of state variables, i.e., $\bar{u}_p(t) = \bar{\gamma}_p(t, \bar{x}_p, \bar{u}_p)$, $\bar{u}_e(t) = \bar{\gamma}_e(t, \bar{x}_e, \bar{u}_e)$. Using $\bar{\gamma}_p$ and $\bar{\gamma}_e$, the value of the game, if it exists, is defined as:

$$V = \min_{\gamma_p} \max_{\gamma_e} J = \max_{\gamma_e} \min_{\gamma_p} J \quad (3.9)$$

The existence of the value of the game is assumed in the following discussion.

The open-loop representation of optimal feedback strategy, i.e., the strategy along the optimal path as a function only of initial states, is defined as $\bar{u}_p^*(t) = \bar{\gamma}_p(t, \bar{x}_{p0}, \bar{x}_{e0})$, $\bar{u}_e^*(t) = \bar{\gamma}_e(t, \bar{x}_{p0}, \bar{x}_{e0})$, which are considered to be satisfied with (3.9). Then, using initial states from equation (3.3) and (3.4), the value of the game is expressed as:

$$V = J(\bar{x}_p, \bar{x}_e, \bar{u}_p^*, \bar{u}_e^*, t) \quad (3.10)$$

A feedback saddle-point trajectory is obtained using an open-loop representation of optimal feedback strategy, \bar{u}_p^* and \bar{u}_e^* , under constraints (3.1)-(3.7).

3.2 Necessary Conditions for Saddle-Point Trajectory

Basar and Olsder [28] provide a set of necessary conditions for an open-loop representation of a feedback saddle-point trajectory. Here, the conditions are modified for (3.1) – (3.7), a pursuit-evasion game in fixed coordinates. At first, a Hamiltonian and a parameter at terminal conditions are introduced as:

$$H = \bar{\lambda}_p^T \bar{f}_p + \bar{\lambda}_e^T \bar{f}_e \quad (3.11)$$

$$\Phi = \phi + \bar{v}^T \bar{\psi} \quad (3.12)$$

where $\bar{\lambda}_p$ and $\bar{\lambda}_e$ are adjoint variables and \bar{v} is a set of Lagrange multipliers conjugate to the terminal constraints. As the existence of the value of the game is assumed, the Hamiltonian is separable.

Using the Hamiltonian, adjoint equations are determined as:

$$\dot{\bar{\lambda}}_p = -\left(\frac{\partial H}{\partial \bar{x}_p} \right)^T = -\left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p \quad (3.13)$$

$$\dot{\bar{\lambda}}_e = -\left(\frac{\partial H}{\partial \bar{x}_e} \right)^T = -\left(\frac{\partial \bar{f}_e}{\partial \bar{x}_e} \right)^T \bar{\lambda}_e \quad (3.14)$$

$$\bar{u}_p = \arg \min_{\bar{u}_p} H = \arg \min_{\bar{u}_p} (\bar{\lambda}_p^T \bar{f}_p) \quad (3.15)$$

$$\bar{u}_e = \arg \max_{\bar{u}_e} H = \arg \max_{\bar{u}_e} (\bar{\lambda}_e^T \bar{f}_e) \quad (3.16)$$

$$\bar{\lambda}_p(t_f) = \left(\frac{\partial \Phi}{\partial \bar{x}_p} \right)^T = \left(\frac{\partial \phi}{\partial \bar{x}_p} + \bar{v}^T \frac{\partial \bar{\psi}}{\partial \bar{x}_p} \right)^T \quad (3.17)$$

$$\bar{\lambda}_e(t_f) = \left(\frac{\partial \Phi}{\partial \bar{x}_e} \right)^T = \left(\frac{\partial \phi}{\partial \bar{x}_e} + \bar{v}^T \frac{\partial \bar{\psi}}{\partial \bar{x}_e} \right)^T \quad (3.18)$$

$$\left[H + \frac{\partial \Phi}{\partial t} \right]_{t=t_f} = 0 \quad (3.19)$$

Then, (3.1), (3.2), (3.13) and (3.14) constitute a two-point boundary value problem (TPBVP) with the initial and terminal conditions, (3.3) - (3.5) and (3.17) - (3.19) and controls satisfying (3.6), (3.7), (3.15) and (3.16). However, it is normally very difficult to solve this TPBVP if the problem is large, i.e. has many states and/or controls, or has strong nonlinearity, which often obtains for problems including realistic dynamics.

3.3 Semi-Direct Collocation with Nonlinear Programming

In Section 3.2, it was noted that the TPBVP is often difficult to solve. Thus, a new method based on DCNLP is proposed instead of solving the TPBVP. The new method is constructed on the basis of the following concepts:

(1) In DCNLP, control parameters are usually chosen by an optimizer such as

NPSOL, i.e. the analytical optimality condition (or Pontryagin's principle)

are not required and hence the system adjoint variables are not required.

This is always applicable to the one-sided optimization problem.

(2) If the adjoint variables for one player, for example the pursuer, in a two-sided optimization are included in the DCNLP, then the control variables for the pursuer can be found from optimality condition (3.15).

(3) The optimal control variables for another player, for example the evader, can be found numerically by the optimizer, to minimize the cost function just for the evader.

The method is expected to maintain the robust characteristics of DCNLP. In this method, the optimality condition (3.15), associated adjoint equations (3.13), and terminal boundary conditions are incorporated into the DCNLP formulation. The terminal boundary conditions become:

$$\bar{\Psi}_{\text{EXT}}(\bar{x}_p, \bar{x}_e, \bar{\lambda}_p, t_f) = 0 \quad (3.20)$$

that is boundary conditions derived from (3.17) which are not a function of $\bar{\lambda}_e$ or \bar{v} .

The control \bar{u}_p for one of the players, obtained from (3.15), minimizes the cost function. Then, the original problem can be converted to:

$$V = \max_{u_p} J \quad \text{subject to (3.1) - (3.7), (3.13), (3.15) and (3.20)} \quad (3.21)$$

The problem represented by (3.21) can be used to construct a nonlinear programming problem, as described in Chapter 2, which can be solved using DCNLP, because now a (single) cost function is maximized and the constraints consist of differential equations and algebraic equations.

It is necessary to evaluate the characteristics of the solution of (3.21). The method of calculus of variation and the Pontryagin principle are applied to the system (3.21). A Hamiltonian and a parameter at terminal conditions for (3.21) are introduced as:

$$\begin{aligned} H_{\text{EXT}} &= \bar{\lambda}_{E_p}^T \bar{f}_p + \bar{\lambda}_{E_e}^T \bar{f}_e + \bar{\lambda}_{E\bar{\lambda}_p}^T \left(- \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p \right) \\ &= \bar{\lambda}_{E_p}^T \bar{f}_p + \bar{\lambda}_{E_e}^T \bar{f}_e - \bar{\lambda}_{E\bar{\lambda}_p}^T \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p \end{aligned} \quad (3.22)$$

$$\Phi_{\text{EXT}} = \phi(\bar{x}_p, \bar{x}_e, t_f) + \bar{v}_{E_1}^T \bar{\psi} + \bar{v}_{E_2}^T \bar{\psi}_{\text{EXT}} \quad (3.23)$$

The control variables, \bar{u}_e , and associated adjoint variables are satisfied with the following relationship, where \bar{u}_p is a function of \bar{x}_p and $\bar{\lambda}_p$:

$$\begin{aligned} \dot{\bar{\lambda}}_{E_p} &= - \left(\frac{\partial H_{\text{EXT}}}{\partial \bar{x}_p} \right)^T = - \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_{E_p} + \left\{ \frac{\partial}{\partial \bar{x}_p} \left[\bar{\lambda}_{E\bar{\lambda}_p}^T \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p \right] \right\}^T \\ &= - \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_{E_p} + \left\{ \frac{\partial}{\partial \bar{x}_p} \left(\left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p \right) \right\}^T \bar{\lambda}_{E\bar{\lambda}_p} \end{aligned} \quad (3.24)$$

$$\dot{\bar{\lambda}}_{E_e} = - \left(\frac{\partial H_{\text{EXT}}}{\partial \bar{x}_e} \right)^T = - \left(\frac{\partial \bar{f}_e}{\partial \bar{x}_e} \right)^T \bar{\lambda}_{E_e} \quad (3.25)$$

$$\dot{\bar{\lambda}}_{E\bar{\lambda}_p} = - \left(\frac{\partial H_{\text{EXT}}}{\partial \bar{\lambda}_p} \right)^T = \left\{ \frac{\partial}{\partial \bar{\lambda}_p} \left(\bar{\lambda}_{E\bar{\lambda}_p}^T \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p \right) \right\}^T \quad (3.26)$$

$$\bar{u}_e = \arg \max_{\bar{u}_e} (\bar{\lambda}_{E_e}^T \bar{f}_e) \quad (3.27)$$

$$\bar{\lambda}_{E_p}(t_f) = \frac{\partial \phi}{\partial \bar{x}_p} + \bar{v}_{E_1}^T \frac{\partial \bar{\psi}}{\partial \bar{x}_p} + \bar{v}_{E_2}^T \frac{\partial \bar{\psi}_{EXT}}{\partial \bar{x}_p} \quad (3.28)$$

$$\bar{\lambda}_{E_e}(t_f) = \frac{\partial \phi}{\partial \bar{x}_e} + \bar{v}_{E_1}^T \frac{\partial \bar{\psi}}{\partial \bar{x}_e} + \bar{v}_{E_2}^T \frac{\partial \bar{\psi}_{EXT}}{\partial \bar{x}_e} \quad (3.29)$$

$$\bar{\lambda}_{E_{\lambda_p}}(t_f) = \bar{v}_{E_2}^T \frac{\partial \bar{\psi}_{EXT}}{\partial \bar{\lambda}_p} \quad (3.30)$$

$$\bar{\lambda}_{E_{\lambda_p}}(t_0) = 0 \quad (3.31)$$

$$\left[H_{EXT} + \frac{\partial \Phi_{EXT}}{\partial t} \right]_{t=t_f} = \left[H + \frac{\partial \Phi}{\partial t} - \bar{\lambda}_{E_{\lambda_p}}^T \left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p} \right)^T \bar{\lambda}_p + \bar{v}_{E_2}^T \frac{\partial \bar{\psi}_{EXT}}{\partial t_f} \right]_{t=t_f} = 0 \quad (3.32)$$

The system (3.1) - (3.7), (3.13), (3.15), (3.20) and (3.24) - (3.32) constitutes a TPBVP.

The characteristics of the TPBVP are discussed when the following conditions hold:

$$\bar{\lambda}_{E_{\lambda_p}} = \bar{0} \quad (3.33)$$

$$\bar{v}_{E_2} = \bar{0} \quad (3.34)$$

Then, the substitution of (3.34) into (3.28) and (3.29) provides:

$$\bar{\lambda}_{E_p}(t_f) = \frac{\partial \phi}{\partial \bar{x}_p} + \bar{v}_{E_1}^T \frac{\partial \bar{\psi}}{\partial \bar{x}_p} \quad (3.35)$$

$$\bar{\lambda}_{E_e}(t_f) = \frac{\partial \phi}{\partial \bar{x}_e} + \bar{v}_{E_1}^T \frac{\partial \bar{\psi}}{\partial \bar{x}_e} \quad (3.36)$$

Also, (3.24) can be re-written as:

$$\begin{aligned}\dot{\bar{\lambda}}_{E_p} &= -\left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p}\right)^T \bar{\lambda}_{E_p} + \left\{ \frac{\partial}{\partial \bar{x}_p} \left(\left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p}\right)^T \bar{\lambda}_p \right) \right\}^T \bar{\lambda}_{E_p}, \\ &= -\left(\frac{\partial \bar{f}_p}{\partial \bar{x}_p}\right)^T \bar{\lambda}_{E_p}\end{aligned}\tag{3.37}$$

where the last term vanishes because of condition (3.33). Then, (3.35) and (3.37)

provide:

$$\bar{\lambda}_{E_p} = \bar{\lambda}_p \tag{3.38}$$

The transversality condition (3.32) is transformed by substitution of (3.34) and (3.38):

$$\left[H_{EXT} + \frac{\partial \Phi_{EXT}}{\partial t} \right]_{t=t_f} = \left[H + \frac{\partial \Phi}{\partial t} \right]_{t=t_f} = 0 \tag{3.39}$$

Therefore, the TPBVP of (3.1)-(3.7), (3.13), (3.15), (3.20) and (3.24)-(3.32) becomes a TPBVP of (3.1)-(3.7), (3.13), (3.15), (3.25), (3.27), (3.35), (3.36), (3.38) and (3.39). By replacing $\bar{\lambda}_{Ee}$ and \bar{v}_{EI} into $\bar{\lambda}_e$ and \bar{v} and using (3.38), the new TPBVP is consistent with the necessary conditions for an open-loop representation of a feedback saddle-point trajectory, (3.1)-(3.7) and (3.13)-(3.19). We conclude that the solution of the TPBVP is satisfied with the necessary conditions for an open-loop representation of the feedback saddle-point trajectory when the conditions (3.33) and (3.34) hold.

$\bar{\lambda}_{Ep}$, $\bar{\lambda}_{E\lambda_p}$ and \bar{v}_{E2} can be obtained from the output of NPSOL or whichever NLP problem solver is used for the DCNLP-based method. Then, it is evaluated whether a DCNLP-based method provides the solution satisfying the necessary conditions by checking $\bar{\lambda}_{Ep}$, $\bar{\lambda}_{E\lambda_p}$ and \bar{v}_{E2} .

Before discussing example problems, the characteristics of the DCNLP-based method are discussed. Eqn. (3.21) includes optimality conditions with respect to the control variables of one player, \bar{u}_p . On the other hand, (3.21) does not include optimality conditions with respect to the control variables, \bar{u}_e , of the second player. Thus, the method is regarded as indirect with respect to control variables \bar{u}_p , and direct with respect to control variables \bar{u}_e . Therefore, the method is referred to as semi-direct collocation with nonlinear programming (semi-DCNLP) in this research. In addition to having the robustness of DCNLP, the problem size will be reduced in comparison to the original TPBVP because only adjoint equations (3.13), i.e. only those required for determining control \bar{u}_p algebraically, are required. These are potential advantages of the semi-DCNLP method as a numerical solver.

The discussions for the necessary condition have been based on the case maximizing the cost function subject to certain constraints. It is also obvious that cases of minimizing the cost function are also suitable for the semi-DCNLP method.

3.4 Application and Verification of the Method

3.4.1 Dolichobrachistochrone

Performance of the semi-DCNLP method as a zero-sum two-person differential game solver is demonstrated by solving a simple problem, the dolichobrachistochrone.

The famous brachistochrone problem proposed by John Bernoulli is the problem of finding a minimum-time descent trajectory of a mass from an initial point to a terminal point under gravity by controlling the shape of the frictionless path. The dolichobrachistochrone problem, which was proposed by Isaacs [6], adds a player who acts to delay the progress of the mass toward the terminal surface, i.e., the additional control tries to maximize the descent time. The problem is illustrated in Fig. 3.1, though with the “descent” direction and the gravitational acceleration assumed upward.

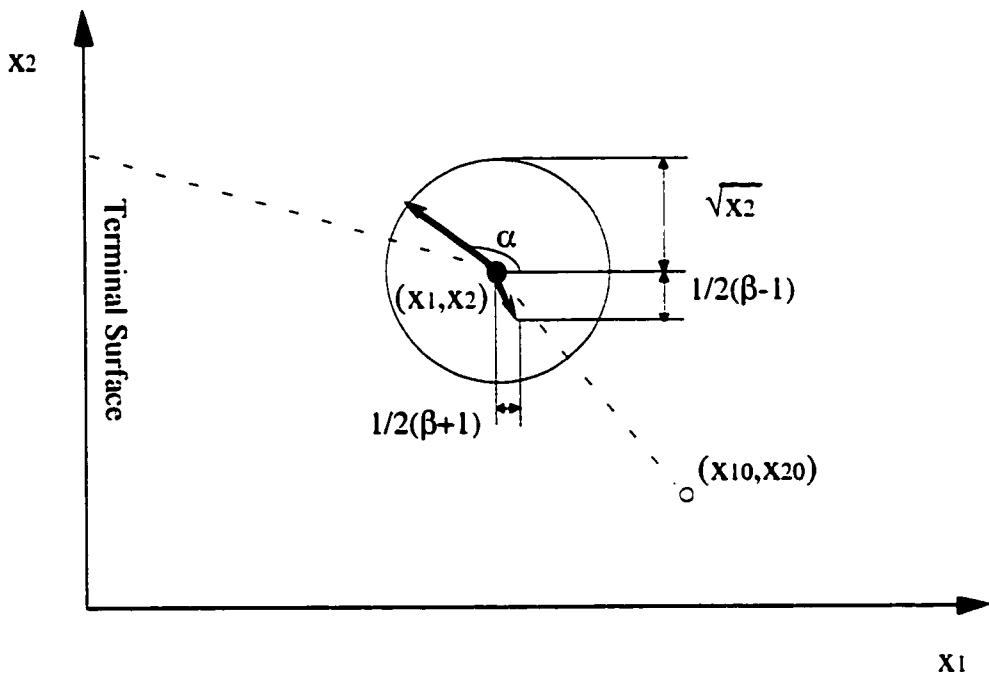


Figure 3.1 Dolichobrachistochrone Problem

A set of equations of motion and terminal conditions are given as;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \sqrt{x_2} \cos \alpha + (\beta + 1)/2 \\ \sqrt{x_2} \sin \alpha + (\beta - 1)/2 \end{bmatrix} \quad (3.40)$$

$$x_1(t_f) = 0 \quad (3.41)$$

where (x_1, x_2) is the position of the mass, α (a control) is the instantaneous direction of the motion, chosen to minimize descent time, and β (a control) determines the magnitude of the horizontal and vertical components of the acceleration opposing gravity, and is chosen (by the “second player”) to maximize the travel time from a given initial position of the mass, (x_{10}, x_{20}) .

The control variable β is bounded as:

$$-1 \leq \beta \leq 1 \quad (3.42)$$

The cost function is the terminal time of the problem:

$$J = t_f \quad (3.43)$$

Then, the Hamiltonian corresponding to (3.11) becomes:

$$H = -\lambda_{x_1} [\sqrt{x_2} \cos \alpha + (\beta + 1)/2] + \lambda_{x_2} [\sqrt{x_2} \sin \alpha + (\beta - 1)/2] \quad (3.44)$$

The optimality conditions with respect to α are:

$$H_\alpha = -\lambda_{x_1} \sqrt{x_2} \sin \alpha + \lambda_{x_2} \sqrt{x_2} \cos \alpha = 0 \quad (3.45)$$

$$H_{\alpha\alpha} = -\lambda_{x_1} \sqrt{x_2} \cos \alpha - \lambda_{x_2} \sqrt{x_2} \sin \alpha \geq 0 \quad (3.46)$$

The equations for the adjoint variables are:

$$\dot{\lambda}_{x_1} = 0 \quad \text{with} \quad \lambda_{x_1}(t_f) = v_{x_1} \quad (3.47)$$

$$\dot{\lambda}_{x_2} = -(\lambda_{x_1} \cos \alpha + \lambda_{x_2} \sin \alpha)/2\sqrt{x_2} \quad \text{with} \quad \lambda_{x_2}(t_f) = 0 \quad (3.48)$$

From (3.47), the adjoint variable, λ_{x_1} is constant. To simplify the analysis, (3.48) is divided by λ_{x_1} and becomes:

$$\dot{\lambda}_{x_2}/\lambda_{x_1} = -(\cos \alpha + (\lambda_{x_2}/\lambda_{x_1}) \sin \alpha)/2\sqrt{x_2} \quad \text{with} \quad \lambda_{x_2}(t_f)/\lambda_{x_1}(t_f) = 0 \quad (3.49)$$

Finally, the problem is considered as being of the form of system (3.21); i.e. the objective function (3.43) is maximized under constraints, (3.40) - (3.42), (3.45), (3.46) and (3.49). The control α is determined using the optimality condition (3.45) with the required adjoint variable satisfying (3.49); the control β is determined, at the collocation points, by the NLP problem solver.

The dolichobrachistochrone problem is solved for three different initial conditions; $(x_{10}, x_{20}) = (0.5, 1.5), (1.5, 1.5)$ and $(3.5, 1.5)$. The discretization is constructed using 20 segments. A solution has also been found for this relatively simple problem using a shooting method, for the purpose of comparing the results. Saddle-point trajectories obtained using both the collocation with nonlinear programming and shooting methods are shown in Fig. 3.2. For $(x_{10}, x_{20}) = (0.5, 1.5)$ and $(1.5, 1.5)$ the trajectory found using the collocation with nonlinear programming corresponds well to the trajectories found using the shooting method. On the other hand, a small difference between the trajectory using the semi-DCNLP method and that using the shooting method are observed for $(x_{10}, x_{20}) = (3.5, 1.5)$. Figures 3.3 through 3.5 show the time histories of the control variables α and β , and the ratio of adjoint variables, $\lambda_{x_2}/\lambda_{x_1}$,

respectively, for the initial condition $(x_{10}, x_{20}) = (3.5, 1.5)$. These time histories are virtually indistinguishable with the exception of the switching time for the discontinuous control, β , as shown in Fig. 3.4.

This difference is quite small and is an artifact of the discretization; the optimal switching time does not coincide exactly with a collocation point. This difference could be reduced by using a finer discretization, i.e. more than 20 segments, or, now that the character of the optimal control is known, by adding another variable, an optimal switching time for β , (which time would then be made a collocation point) to be determined by the NLP problem solver.

From this example problem, it is seen that the collocation with nonlinear programming method can solve a simple zero-sum two-person differential game problem. with bounded control variables for one-player only.

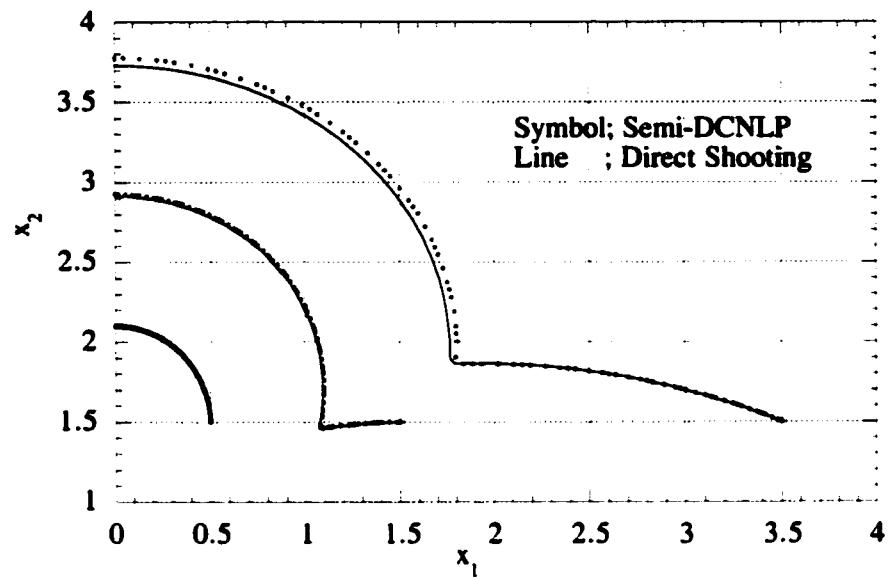


Figure 3.2 Saddle-Point Trajectories of the Dolichobrachistochrone

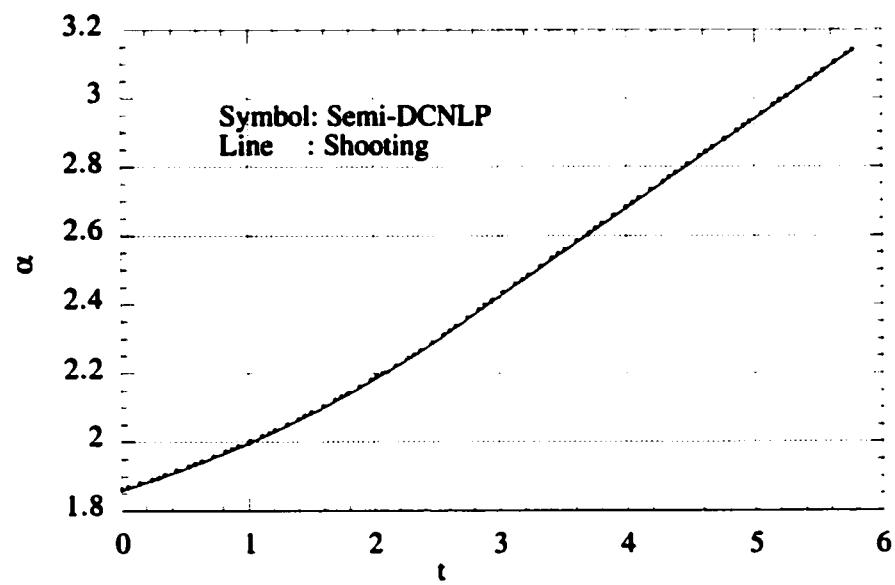
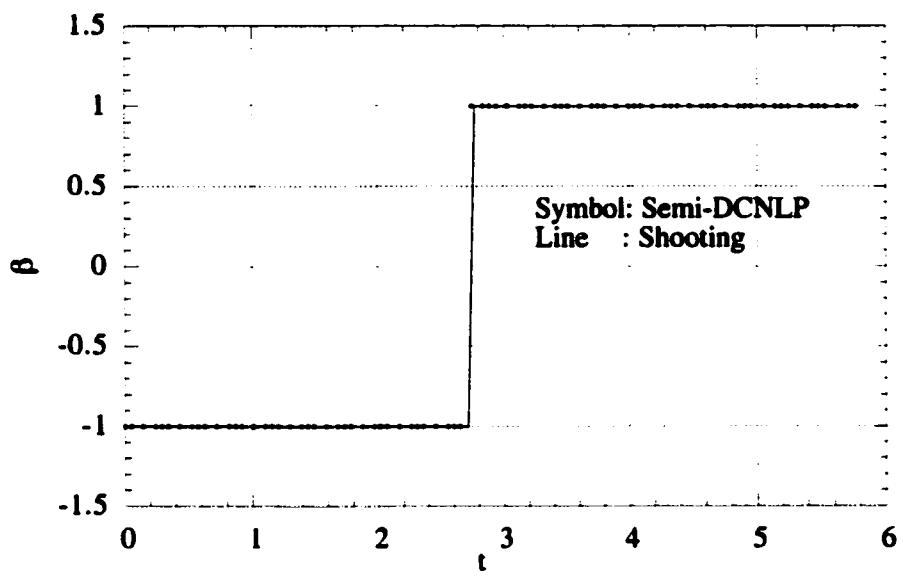
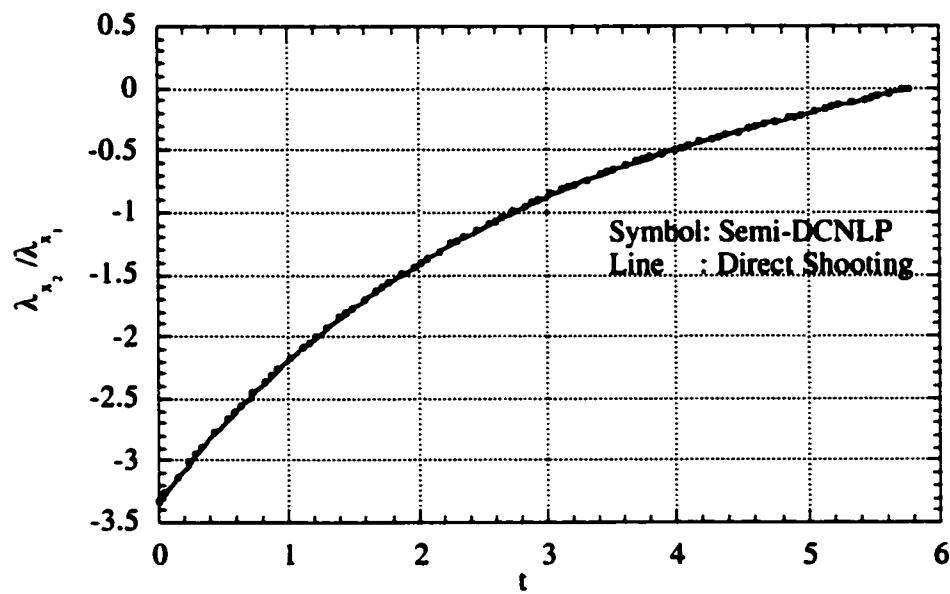


Figure 3.3 Time History of Control Variable α of Dolichobrachistochrone
 $((x10,x20)=(3.5,1.5))$



**Figure 3.4 Time History of Control Variable β of Dolichobrachistochrone
 $((x_{10}, x_{20})=(3.5, 1.5))$**



**Figure 3.5 Time History of Ratio of Adjoint Variables of Dolichobrachistochrone
 $((x_{10}, x_{20})=(3.5, 1.5))$**

3.4.2 Ballistic Interception

A ballistic interception problem is solved to verify the feasibility of the semi-DCNLP method as a pursuit-evasion game solver. In the ballistic interception, two objects, a chaser and a target, construct a pursuit-evasion game. The pursuer with initially high velocity, but no thrust, and controlling flight path angle, intercepts a target with initial low velocity, but no thrust, also controlling flight path angle, in a uniform gravitational field. (Fig. 3.6) The problem is simple enough that it may also be solved using a shooting method. Then, the solution obtained using semi-DCNLP can be compared to the solution obtained using a shooting method.

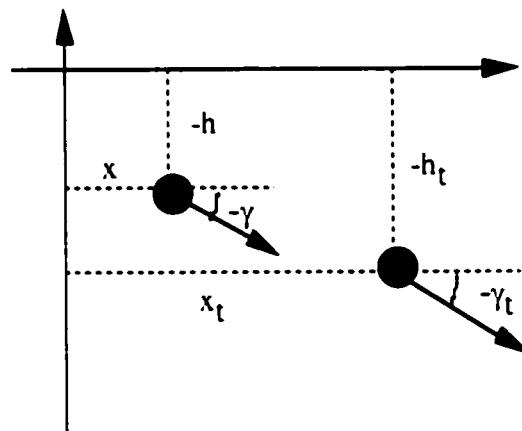


Figure 3.6 Ballistic Interception

The velocities of the objects under uniform gravitational force, \mathbf{g} , are provided as follows:

$$v = \sqrt{v_{\text{init}}^2 - 2gh} \quad (3.50)$$

where the initial velocity is v_{init} , the height is h and the initial height is assumed as zero.

Using (3.50), the equations of motion for the pursuer and the target are:

$$\frac{dx}{dt} = \sqrt{v_0^2 - 2gh} \cos \gamma \quad (3.51)$$

$$\frac{dh}{dt} = \sqrt{v_0^2 - 2gh} \sin \gamma \quad (3.52)$$

$$\frac{dx_t}{dt} = \sqrt{v_{0t}^2 - 2gh_t} \cos \gamma_t \quad (3.53)$$

$$\frac{dh_t}{dt} = \sqrt{v_{0t}^2 - 2gh_t} \sin \gamma_t \quad (3.54)$$

where x is downrange, h is height and γ is flight path angle. The pursuit-evasion game is on a vertical plane, $x-h$.

The cost of this differential game is terminal time, t_f :

$$J = t_f \quad (3.55)$$

In the game, the pursuer minimizes t_f and the evader maximizes t_f . Terminal constraints are;

$$\begin{bmatrix} x_t - x \\ h_t - h \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.56)$$

A Hamiltonian for this problem is:

$$\begin{aligned} H = & 1 + \lambda_x \sqrt{v_0^2 - 2gh} \cos \gamma + \lambda_h \sqrt{v_0^2 - 2gh} \sin \gamma \\ & + \lambda_{x_t} \sqrt{v_{0t}^2 - 2gh_t} \cos \gamma_t + \lambda_{h_t} \sqrt{v_{0t}^2 - 2gh_t} \sin \gamma_t \end{aligned} \quad (3.57)$$

Using the Pontryagin principle and adjoint equations, the following equations are derived.

$$\frac{\partial H}{\partial \gamma} = -\lambda_x \sqrt{v_0^2 - 2gh} \sin \gamma + \lambda_h \sqrt{v_0^2 - 2gh} \cos \gamma = 0 \quad (3.58)$$

$$\frac{\partial^2 H}{\partial \gamma^2} = -\lambda_x \sqrt{v_0^2 - 2gh} \cos \gamma - \lambda_h \sqrt{v_0^2 - 2gh} \sin \gamma \geq 0 \quad (3.59)$$

$$\frac{\partial H}{\partial \gamma_t} = -\lambda_{x_t} \sqrt{v_{0t}^2 - 2gh_t} \sin \gamma_t + \lambda_{h_t} \sqrt{v_{0t}^2 - 2gh_t} \cos \gamma_t = 0 \quad (3.60)$$

$$\frac{\partial^2 H}{\partial \gamma_t^2} = -\lambda_{x_t} \sqrt{v_{0t}^2 - 2gh_t} \cos \gamma_t - \lambda_{h_t} \sqrt{v_{0t}^2 - 2gh_t} \sin \gamma_t \leq 0 \quad (3.61)$$

$$\frac{d\lambda_x}{dt} = 0 \quad (3.62)$$

$$\frac{d\lambda_h}{dt} = \frac{g}{\sqrt{v_0^2 - 2gh}} (\lambda_x \cos \gamma + \lambda_h \sin \gamma) \quad (3.63)$$

$$\frac{d\lambda_{x_t}}{dt} = 0 \quad (3.64)$$

$$\frac{d\lambda_{h_t}}{dt} = \frac{g}{\sqrt{v_{0t}^2 - 2gh_t}} (\lambda_{x_t} \cos \gamma_t + \lambda_{h_t} \sin \gamma_t) \quad (3.65)$$

$$\lambda_x(t_f) + \lambda_{x_t}(t_f) = 0 \quad (3.66)$$

$$\lambda_h(t_f) + \lambda_{h_t}(t_f) = 0 \quad (3.67)$$

$$H(t_f) = 0 \quad (3.68)$$

From (3.62) and (3.64), λ_x and λ_{x_t} are constant. Then, λ_h and λ_{h_t} can be normalized by λ_x and λ_{x_t} . Identifying normalized adjoint variables by (), (3.58), (3.60), (3.63), (3.65) and (3.67) can be expressed as follows:

$$-\sqrt{v_0^2 - 2gh} \sin \gamma + \lambda_h' \sqrt{v_0^2 - 2gh} \cos \gamma = 0 \quad (3.69)$$

$$-\sqrt{v_{0t}^2 - 2gh_t} \sin \gamma_t + \lambda_{h_t}' \sqrt{v_{0t}^2 - 2gh_t} \cos \gamma_t = 0 \quad (3.70)$$

$$\frac{d\lambda_h'}{dt} = \frac{g}{\sqrt{v_0^2 - 2gh}} (\cos \gamma + \lambda_h' \sin \gamma) \quad (3.71)$$

$$\frac{d\lambda_{h_t}'}{dt} = \frac{g}{\sqrt{v_{0t}^2 - 2gh_t}} (\cos \gamma_t + \lambda_{h_t}' \sin \gamma_t) \quad (3.72)$$

$$\lambda_h(t_f)' - \lambda_{h_t}(t_f)' = 0 \quad (3.73)$$

In this formulation, (3.66) and the transversality condition (3.68) are only used to solve for λ_x and λ_{x_t} .

Eqns. (3.51)-(3.54), (3.56), (3.69)-(3.73) and the initial conditions for the state variables constitute a TPBVP. This TPBVP is relatively simple, so the problem can be solved using a shooting method.

The problem is also solved using semi-DCNLP. A formulation for the semi-DCNLP is defined as maximizing interception time, (3.55), subject to Eqns. (3.51)-(3.54), (3.56), (3.69) and (3.71). The trajectories, control variables and adjoint variables obtained by the semi-DCNLP method are shown and compared to those obtained with a shooting method in Figs. 3.7-3.9. The result of the semi-DCNLP is consistent with the result of the shooting method in all figures. The condition (3.33) is also satisfied in this problem. (in this case, (3.34) is not required because of lack of a boundary condition

defined as (3.20)) The result verifies that the semi-DCNLP solution algorithm constructed in Sec. 3.3 is capable of solving a pursuit-evasion game.

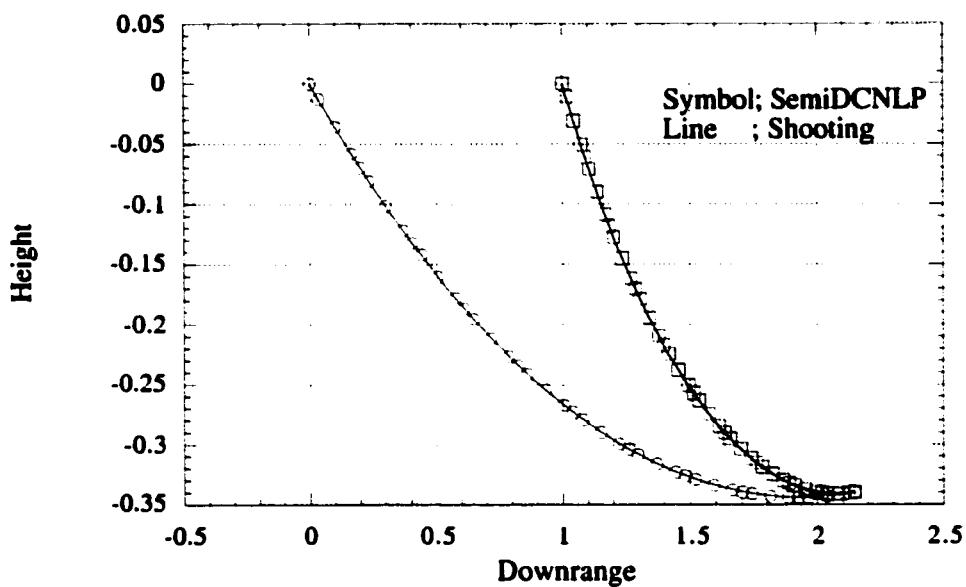


Figure 3.7 Saddle-Point Trajectories for Ballistic Interception

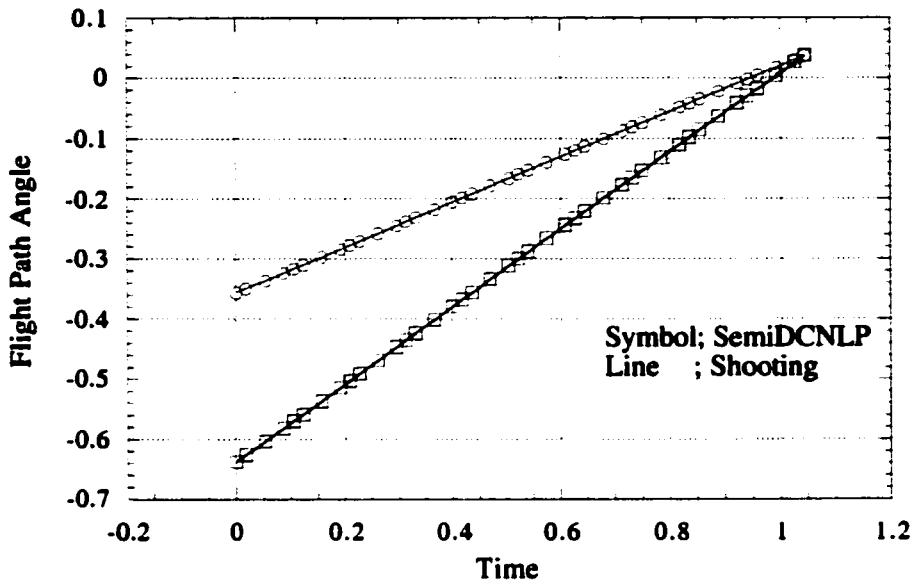


Figure 3.8 Time History of Flight Path Angle for Ballistic Interception

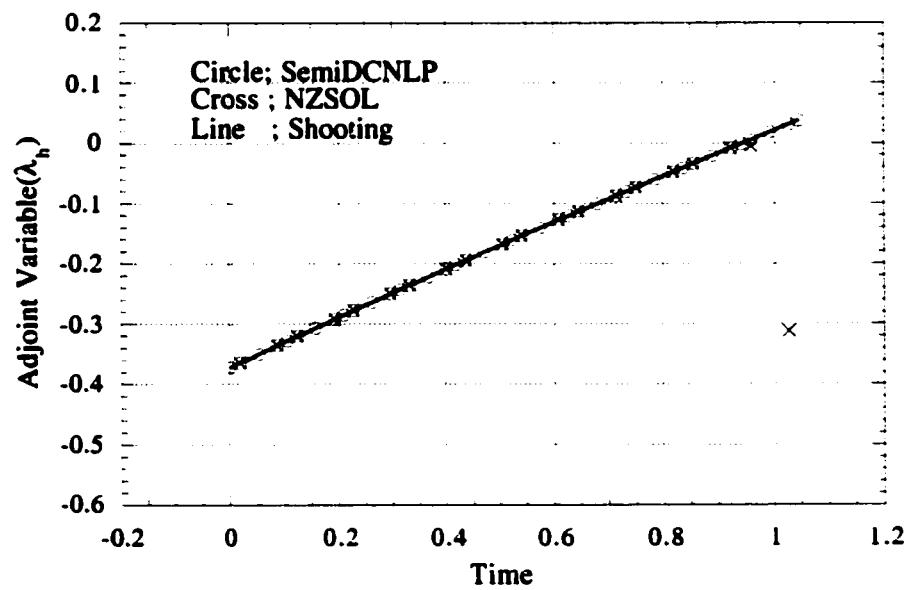


Figure 3.9 Time History of Adjoint Variables for Ballistic Interception

3.4.3 Homicidal Chauffeur Problem

The problem of the "homicidal chauffeur", after Isaacs [6], is one of the most studied problems in the theory of differential games. The characteristics of the optimal trajectories have already been clarified, i.e. three singular surfaces, a universal surface, a dispersal surface and a transition surface, are known for the solution of the homicidal chauffeur problem. Qualitatively, the three singular surfaces in the homicidal chauffeur problem show the following characteristics: (1) the universal surface expresses that the pursuer goes straight until the game terminates, (2) the dispersal surface emerges when the pursuer is located precisely in front of the evader and (3) the transition surface is recognized when the pursuer changes its direction of turning, i.e., "swerving". In this research the homicidal chauffeur problem is solved using the semi-DCNLP method and then evaluated for its ability to identify the characteristics of the trajectories of the homicidal chauffeur's problem.

In the homicidal chauffeur problem, a high-speed car (or pursuer aircraft) with limited turn radius captures a low-speed pedestrian (or evader aircraft) capable of instant change of direction. The problem is solved for relative coordinates, in the same manner as the system in Sec. 3.4.1, at first. (Fig. 3.10) This provides the group of the trajectories, which is expected to identify the singular surfaces easily. Thereafter, the semi-DCNLP is applied to the problem in fixed coordinates (Fig. 3.11) by selecting typical cases of trajectories from the solution in the relative coordinates.

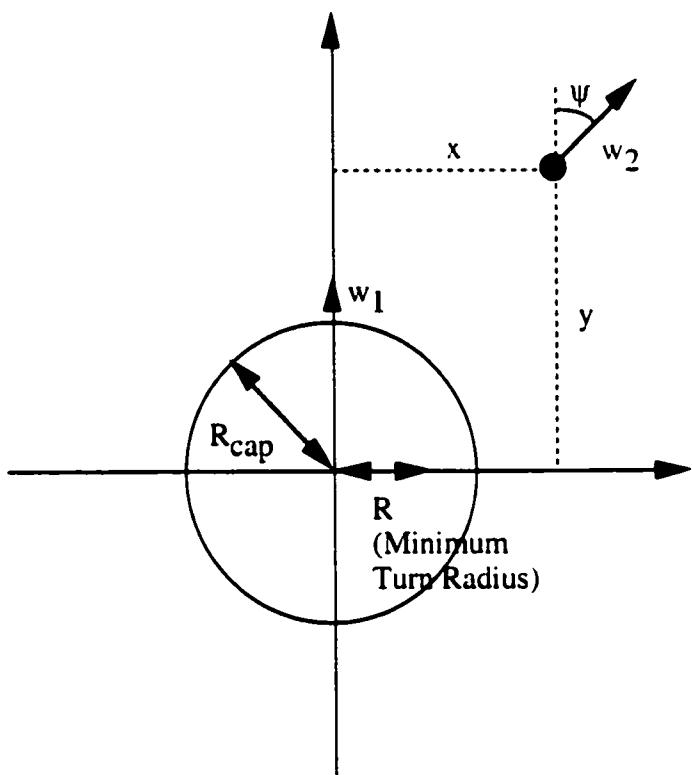


Figure 3.10 Homicidal Chauffeur Problem in Relative Coordinates

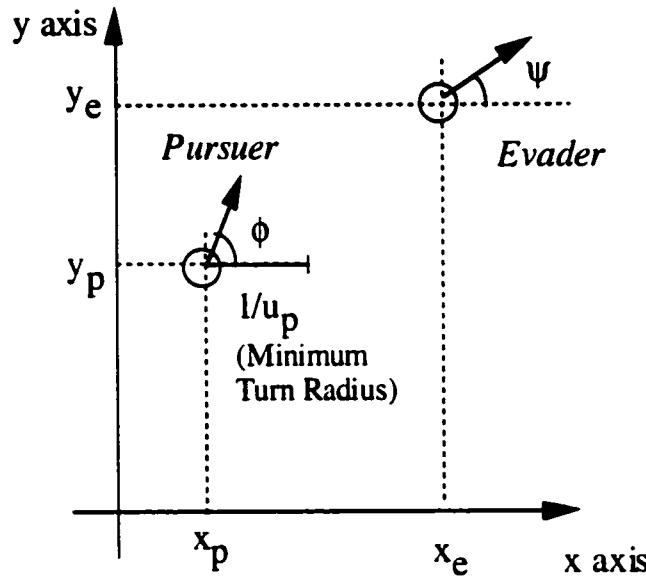


Figure 3.11 Homicidal Chauffeur Problem in Fixed Coordinates

In the homicidal chauffeur problem in relative coordinates, the pursuer always moves along the y-axis as shown in Fig. 3.10, i.e., the coordinates rotate as the y-axis is aligned with the pursuer path. A system of equations of motion for the homicidal chauffeur problem in relative coordinates and the control constraints are expressed as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{w_1}{R} y\phi + w_2 \sin \psi \\ \frac{w_1}{R} x\phi - w_1 + w_2 \cos \psi \end{pmatrix} \quad (3.74)$$

$$-1 \leq \phi \leq 1 \quad (3.75)$$

where (x, y) are the Cartesian coordinate of the evader with respect to the pursuer, ϕ and ψ are control variables of pursuer and evader respectively, w_1 is velocity of pursuer,

w_2 is velocity of evader and R is the minimum turn radius of the pursuer. The termination of the game, which defines terminal condition of the problem, occurs when the range between the pursuer and evader becomes the capture range, R_{cap} , i.e.:

$$x^2 + y^2 = R_{cap}^2 \quad (3.76)$$

The cost function of the problem is the terminal time:

$$J = t_f \quad (3.77)$$

which means that the pursuer tries to minimize time to capture and the evader tries to maximize it.

For the method of collocation with nonlinear programming, it is required to solve one player's control using the Pontryagin principle. Then, ψ is determined using the Pontryagin principle as:

$$\begin{aligned} \lambda_x \cos \psi - \lambda_y \sin \psi &= 0 \\ \lambda_x \sin \psi + \lambda_y \cos \psi &\leq 0 \end{aligned} \quad (3.78)$$

where adjoint variables are expressed as:

$$\begin{pmatrix} \dot{\lambda}_x \\ \dot{\lambda}_y \end{pmatrix} = \begin{pmatrix} -\lambda_y \frac{w_1}{R} \phi \\ \lambda_x \frac{w_1}{R} \phi \end{pmatrix} \quad (3.79)$$

$$x(t_f) \lambda_y(t_f) - y(t_f) \lambda_x(t_f) = 0 \quad (3.80)$$

Finally, (3.77) is minimized subject to (3.74)-(3.76) and (3.78)-(3.80) using the DCNLP method previously described.

The problem is solved for a variety of initial conditions with parameters $w_1=1$, $w_2=0.1$, $R_{cap}=0.8$ and $R=1$ using collocation with nonlinear programming as shown in Fig. 3.12. This figure shows that the method identifies the singular surfaces in the homicidal chauffeur problem.

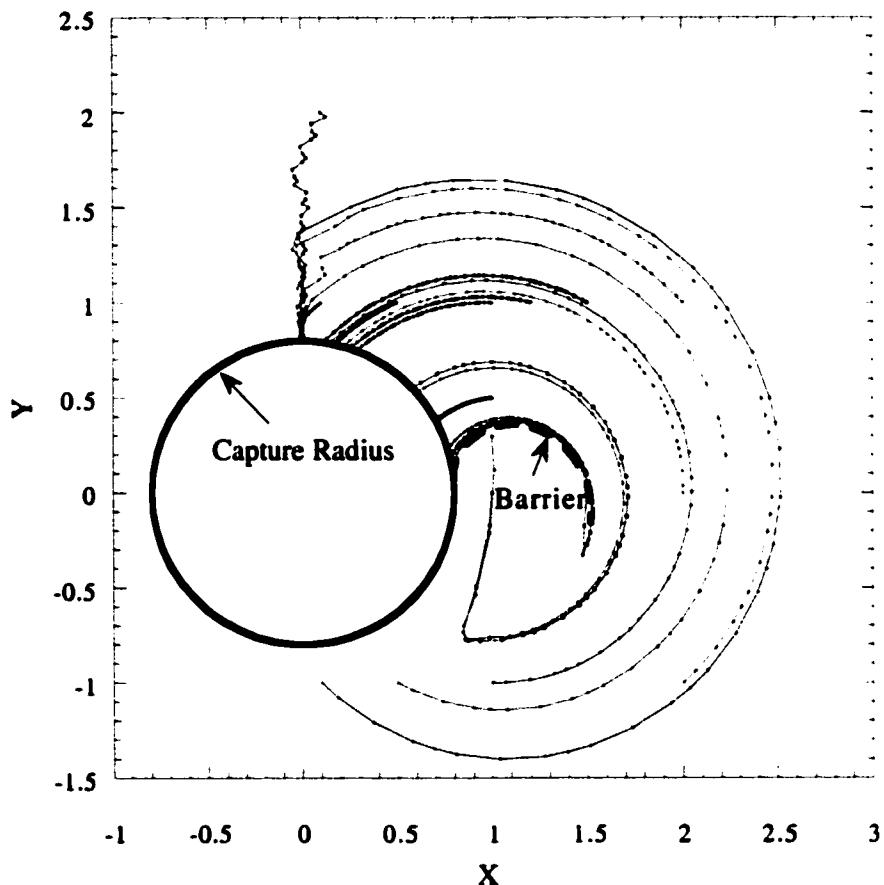


Figure 3.12 A Group of Saddle-Point Trajectories of the Homicidal Chauffeur Problem in Relative Coordinates

A set of equations of motion and terminal conditions in fixed coordinates, which is shown in Fig. 3.11, are given as;

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_p \cos \phi \\ v_p \sin \phi \\ u_p \end{bmatrix} \quad (3.81)$$

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} v_e \cos \psi \\ v_e \sin \psi \end{bmatrix} \quad (3.82)$$

$$(x_p(t_f) - x_e(t_f))^2 + (y_p(t_f) - y_e(t_f))^2 = R_{cap}^2 \quad (3.83)$$

where (x_p, y_p) and (x_e, y_e) are the position of the pursuer and the evader, ϕ and ψ are the heading angle of the pursuer and the evader respectively, and u_p is the rate of change of the heading angle of the pursuer. The game terminates when the separation between the pursuer and the evader is equal to R_{cap} , defined as a minimum length.

The control variable for the pursuer, u_p , is bounded as:

$$-l \leq u_p \leq l \quad (3.84)$$

The cost function is the terminal time of the problem:

$$J = t_f \quad (3.85)$$

Then, the Hamiltonian corresponding to (3.11) becomes:

$$H = \lambda_{x_p} v_p \cos \phi + \lambda_{y_p} v_p \sin \phi + \lambda_u u_p + \lambda_{x_e} v_e \cos \psi + \lambda_{y_e} v_e \sin \psi \quad (3.86)$$

The optimality conditions with respect to ψ are:

$$H_{u_e} = -\lambda_{x_e} v_e \sin \psi + \lambda_{y_e} v_e \cos \psi = 0 \quad (3.87)$$

$$H_{u_e u_e} = -\lambda_{x_e} v_e \cos \psi - \lambda_{y_e} v_e \sin \psi \leq 0 \quad (3.88)$$

The equations for the adjoint variables are:

$$\dot{\lambda}_{x_e} = 0 \quad \text{with} \quad \lambda_{x_e}(t_f) = -2v(x_p(t_f) - x_e(t_f)) \quad (3.89)$$

$$\dot{\lambda}_{y_e} = 0 \quad \text{with} \quad \lambda_{y_e}(t_f) = -2v(y_p(t_f) - y_e(t_f)) \quad (3.90)$$

From (3.89) and (3.90), the adjoint variables, λ_{x_e} and λ_{y_e} , are constant. To simplify the analysis, (3.90) is divided by the constant λ_{x_e} and becomes:

$$\dot{\lambda}_{y_e}/\lambda_{x_e} = 0 \quad \text{with} \quad \lambda_{y_e}/\lambda_{x_e} = (x_p(t_f) - x_e(t_f))/(y_p(t_f) - y_e(t_f)) \quad (3.91)$$

Finally, the problem is solved using the semi-DCNLP method; i.e. the objective function (3.85) is minimized under constraints (3.81) - (3.84), (3.87), (3.88) and (3.91).

Three different initial conditions are selected from the results in Fig. 3.12 to identify two trajectories across the transition surface or partially along the universal surface in addition to a trajectory not affected by the singular surfaces. The semi-DCNLP method solves for the trajectories and control variables of the pursuer and the evader as shown in Figs. 3.13-3.21.

The trajectory in Fig. 3.13 corresponds to the trajectory not affected by the singular surface. The control variable of the pursuer always takes an extreme value, -1. in Fig. 3.14, which means that the heading angle change continues in the same direction. The trajectory in Fig. 3.16 identifies the universal surface, i.e., the pursuer goes straight until the game terminates. However, the control history in Fig. 3.17 shows a chattering

about zero. This is a potential limitation of this method, i.e., some of the singular surfaces can be identified not on the history of the control variables but on the history of the state. The trajectory in Fig. 3.19 includes the swerving maneuvers across the transition surface. This is also clearly found on the history of the control variables in Fig. 3.20 which change from 1 to -1.

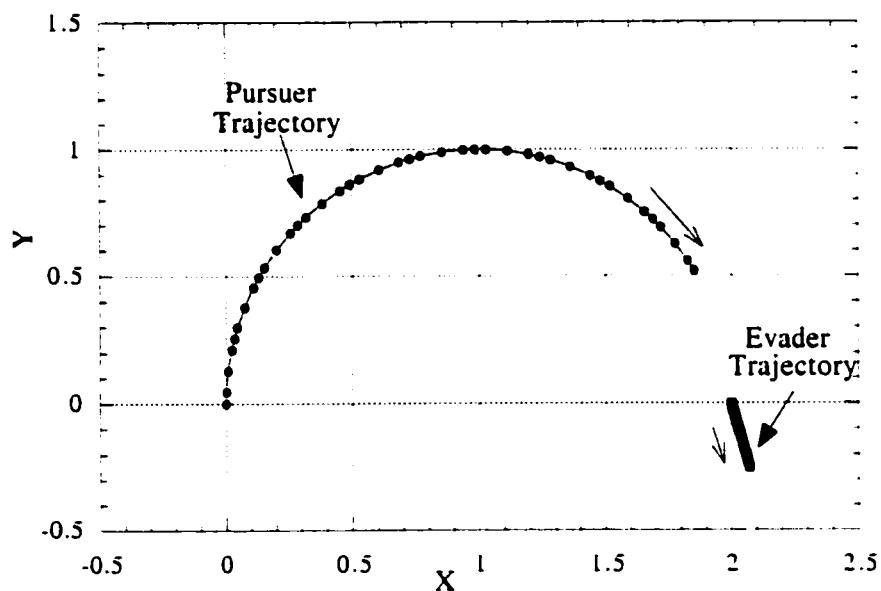


Figure 3.13 Saddle-Point Trajectories of the Homicidal Chauffeur Problem in Fixed Coordinates (Non Singular Surface)

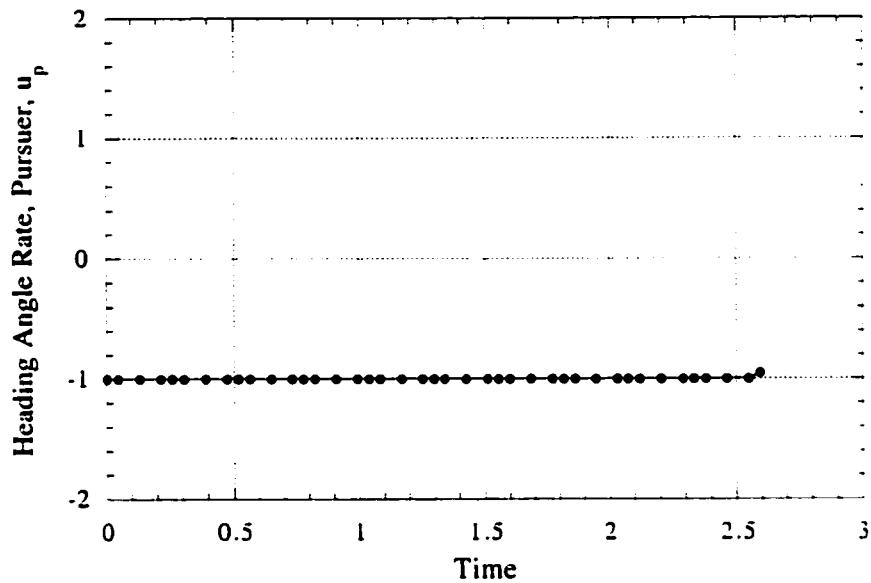


Figure 3.14 Time History of Control Variable for Pursuer of Homicidal Chauffeur Problem in Fixed Coordinates (Non Singular Surface)

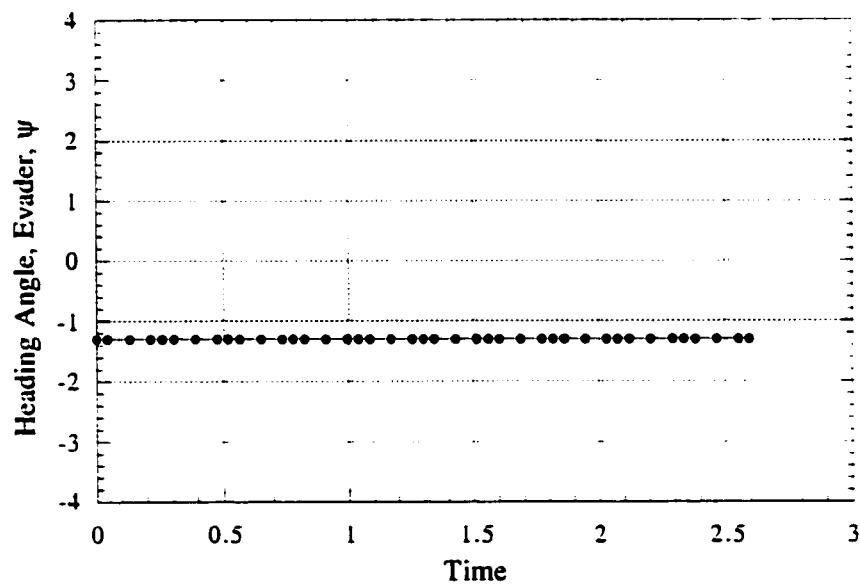


Figure 3.15 Time History of Control Variable for Evader of Homicidal Chauffeur Problem in Fixed Coordinates (Non Singular Surface)

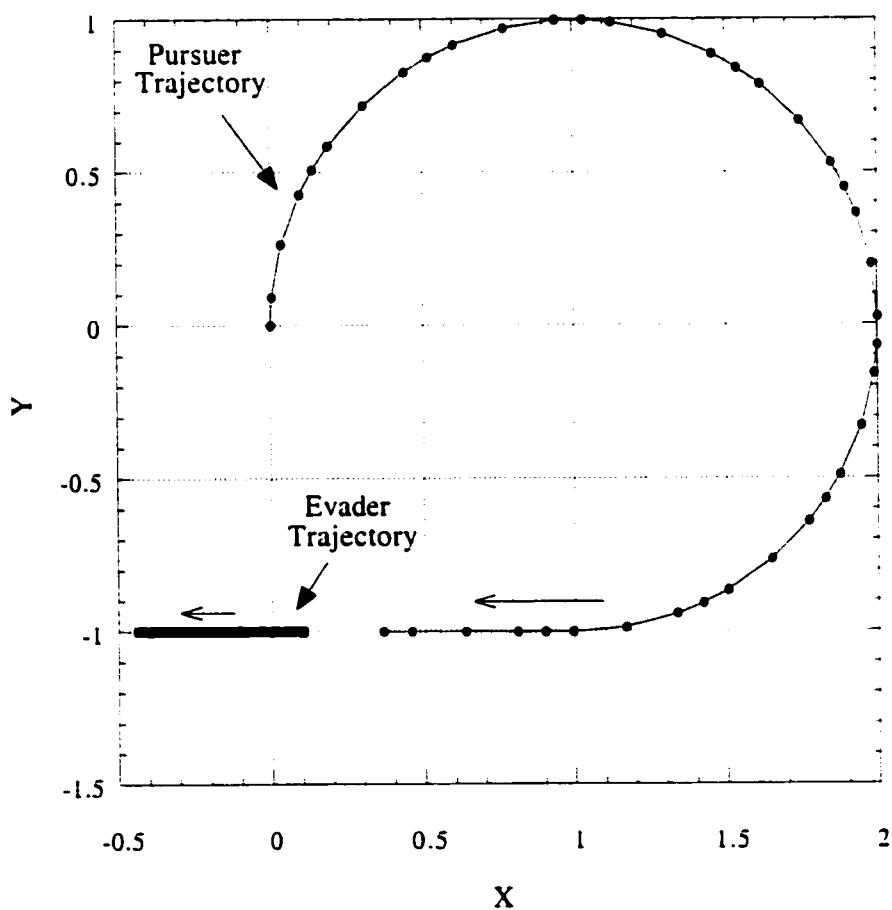


Fig.3.16 Saddle-Point Trajectories of Homicidal Chauffeur Problem
in Fixed Coordinates (Universal Surface)

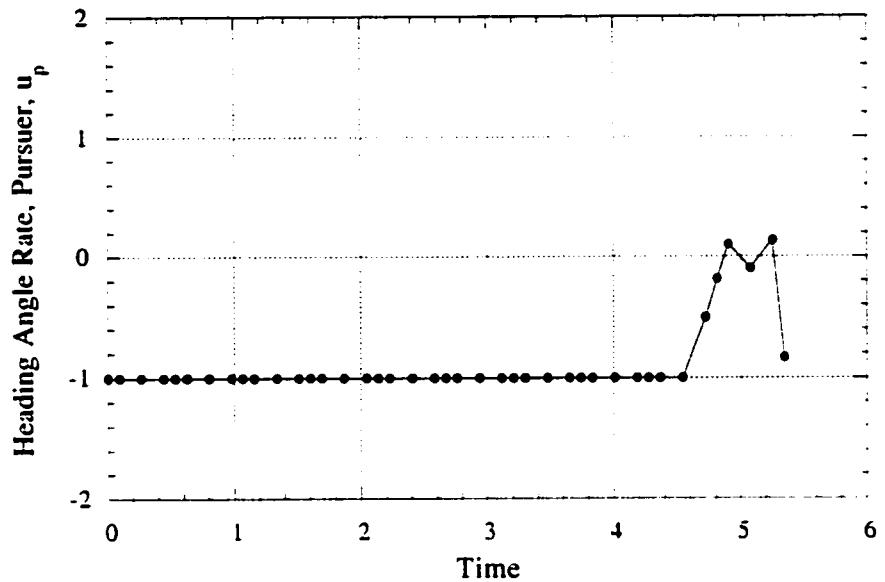


Fig.3.17 Time History of Control Variable for Pursuer of Homicidal Chauffeur Problem in Fixed Coordinates (Universal Surface)

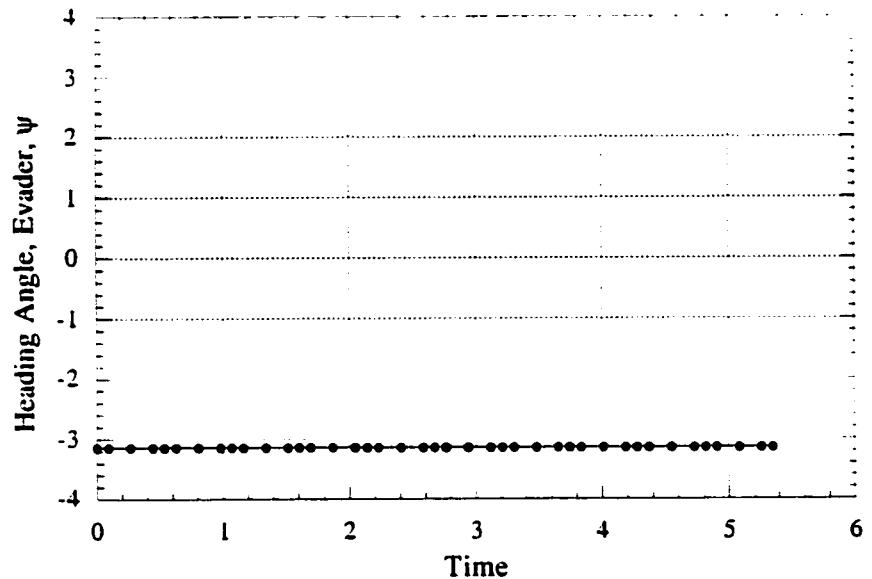


Fig.3.18 Time History of Control Variable for Evader of Homicidal Chauffeur Problem in Fixed Coordinates (Universal Surface)

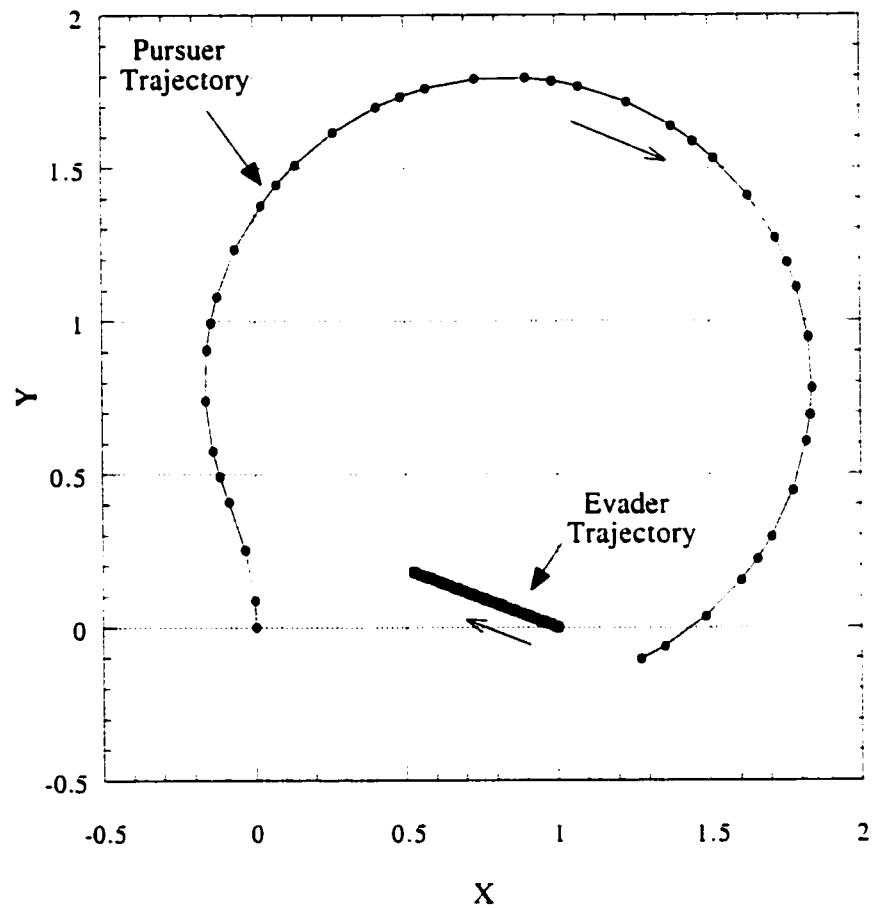


Fig.3.19 Saddle-Point Trajectories of Homicidal Chauffeur Problem
in Fixed Coordinates (Transition Surface)

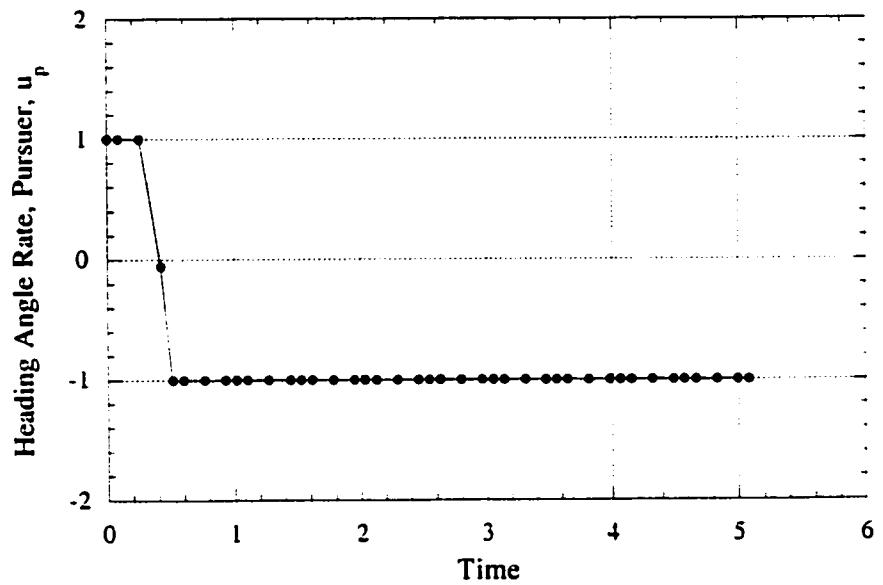


Fig.3.20 Time History of Control Variable for Pursuer of Homicidal Chauffeur Problem in Fixed Coordinates (Transition Surface)

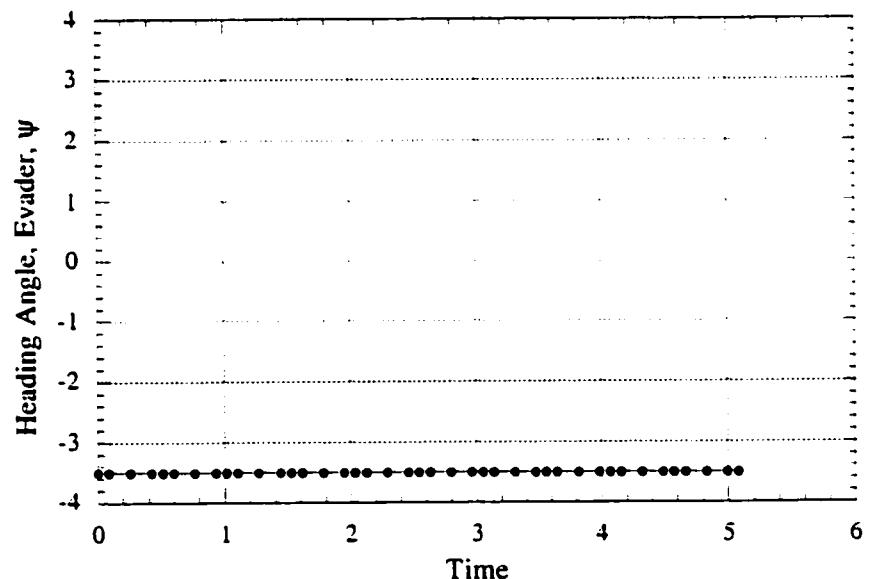


Fig.3.21 Time History of Control Variable for Evader of Homicidal Chauffeur Problem in Fixed Coordinates (Transition Surface)

The solution to the problem of the homicidal chauffeur, a simple pursuit-evasion problem with a complicated solution structure, shows that the semi-DCNLP method is capable of solving for the saddle-point trajectory and identifying some of the singular surfaces.

3.4.4 Minimum Time Spacecraft Interception for Optimally Evasive Target

The semi-DCNLP method is now applied to a problem in which a spacecraft with constant thrust acceleration, in an inverse-square gravitational field, desires to intercept an optimally-evasive target in minimum time. This problem is chosen to prove that the semi-DCNLP method can solve a realistic dynamic problem including actual forces. The spacecraft and target vehicle are assumed to be initially in different but coplanar orbits about a planet, and subject to Newtonian gravity. Both vehicles are capable of constant thrust acceleration but the thrust acceleration of the target, T_{t} , is set smaller than that of the spacecraft, T_s , so that the spacecraft can intercept the target in a finite time. The control variables are the thrust pointing angles of the respective vehicles.

Fig. 3.22 illustrates the problem.

In polar coordinates, the equations of motion may be expressed as:

$$\begin{bmatrix} \dot{v}_r \\ \dot{v}_\theta \\ \dot{r} \\ \dot{\theta} \\ \dot{v}_\pi \\ \dot{v}_{\theta_t} \\ \dot{r}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} T \sin \delta - (\mu - v_\theta^2 r) / r^2 \\ T \cos \delta - v_r v_\theta / r \\ v_r \\ v_\theta / r \\ T_t \sin \delta_t - (\mu - v_{\theta_t}^2 r_t) / r_t^2 \\ T_t \cos \delta_t - v_\pi v_{\theta_t} / r_t \\ v_\pi \\ v_{\theta_t} / r_t \end{bmatrix} \quad (3.92)$$

where (r, θ) is the position, (v_r, v_θ) the velocity and δ the thrust pointing angle of the spacecraft; (r_t, θ_t) is the position, (v_π, v_{θ_t}) the velocity and δ_t the thrust pointing angle of the target.

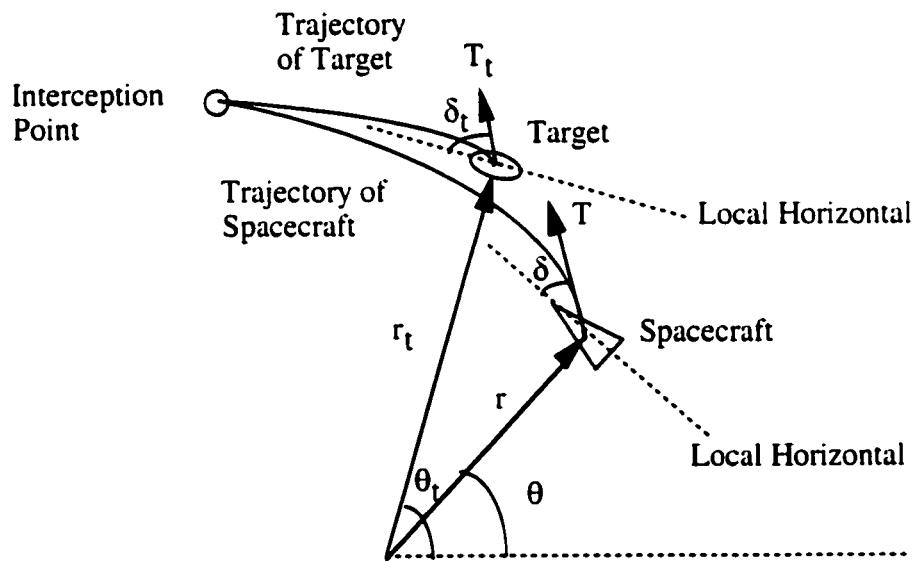


Fig.3.22 Spacecraft Interception for Optimally Evasive Target

The terminal condition is interception:

$$\begin{bmatrix} r(t_f) - r_t(t_f) \\ \theta(t_f) - \theta_t(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.93)$$

The cost function is the time to interception:

$$J = t_f \quad (3.94)$$

Then, the Hamiltonian becomes:

$$\begin{aligned} H = & 1 + \lambda_{v_r} [T \sin \delta - (\mu - v_\theta^2 r) / r^2] + \lambda_{v_\theta} [T \cos \delta - v_r v_\theta / r] \\ & + \lambda_r v_r + \lambda_\theta v_\theta / r + \lambda_{v_n} [T_t \sin \delta_t - (\mu - v_{\theta t}^2 r_t) / r_t^2] \\ & + \lambda_{v_n} [T_t \cos \delta_t - v_n v_{\theta t} / r_t] + \lambda_r v_n + \lambda_\theta v_{\theta t} / r_t \end{aligned} \quad (3.95)$$

The problem is formulated as a zero-sum two-person differential game. The optimality conditions with respect to δ become:

$$H_\delta = T(\lambda_{v_r} \cos \delta - \lambda_{v_n} \sin \delta) = 0 \quad (3.96)$$

$$H_{\delta\delta} = -T(\lambda_{v_r} \sin \delta + \lambda_{v_n} \cos \delta) \geq 0 \quad (3.97)$$

The equations for the adjoint variables associated with the optimality conditions, (3.96) and (3.97), are:

$$\dot{\lambda}_{v_r} = (\lambda_{v_n} v_\theta - \lambda_r r) / r \quad \text{with } \lambda_{v_r}(t_f) = 0 \quad (3.98)$$

$$\dot{\lambda}_{v_n} = (-2\lambda_{v_r} v_\theta + \lambda_{v_n} v_r - \lambda_\theta) / r \quad \text{with } \lambda_{v_n}(t_f) = 0 \quad (3.99)$$

$$\dot{\lambda}_r = [\lambda_{v_r}(-2\mu + v_\theta^2 r) - \lambda_{v_n} v_r v_\theta r + \lambda_\theta v_\theta r] / r^3 \quad \text{with } \lambda_r(t_f) = v_r \quad (3.100)$$

$$\dot{\lambda}_\theta = 0 \quad \text{with } \lambda_\theta(t_f) = v_\theta \quad (3.101)$$

Hence, the problem is to maximize (3.94) under constraints (3.92), (3.93) and (3.96)-(3.101) and thus can in principle be solved using the semi-DCNLP method. Note that it is not necessary, when using this method, to solve for the four adjoint variables conjugate to the state variables of the target since the Hamiltonian is not required

explicitly and the control for the target, δ_t , will be found at the collocation points by the NLP problem solver.

The problem in which the intercepting spacecraft has $r(0)=1$ with thrust acceleration of $T=0.05$ and the target spacecraft has $r_t(0)=1.05$ with $T_t=0.0025$, with both vehicles initially in circular orbits, is solved for several initial conditions. Normalized variables are used for convenience i.e. the planet's gravitational constant is set to $\mu=1$ so that the gravitational acceleration at a radial distance $r=1$, would be 1 distance unit per time unit squared. The time to interception, which is the value of the game in this problem, and the corresponding saddle-point trajectories are shown in Fig. 3.23 and Fig. 3.24. The semi-DCNLP solution uses 10 segments.

As an example, histories of thrust pointing angles for the case $\theta(0)=0$, $\theta_t(0)=0.4$ are shown in Fig. 3.25 respectively. The thrust of both the spacecraft and target have a radially inward component initially which changes to radially outward by the time of interception. Interception occurs after approximately half of a revolution of the planet. Fig. 3.26 shows the absolute direction of the line-of-sight from the spacecraft to the target for this same example. Guelman, Shinar and Green suggested "both players turn toward the final line-of-sight direction with an asymptotically decaying rate" in the case of an optimal pursuit-evasion between two variable speed players [9]. In Fig. 3.26, it is seen that the line-of-sight time history for this case has this same characteristic. This

observation qualitatively supports the conclusion that the result is a saddle-point trajectory in the pursuit-evasion game.

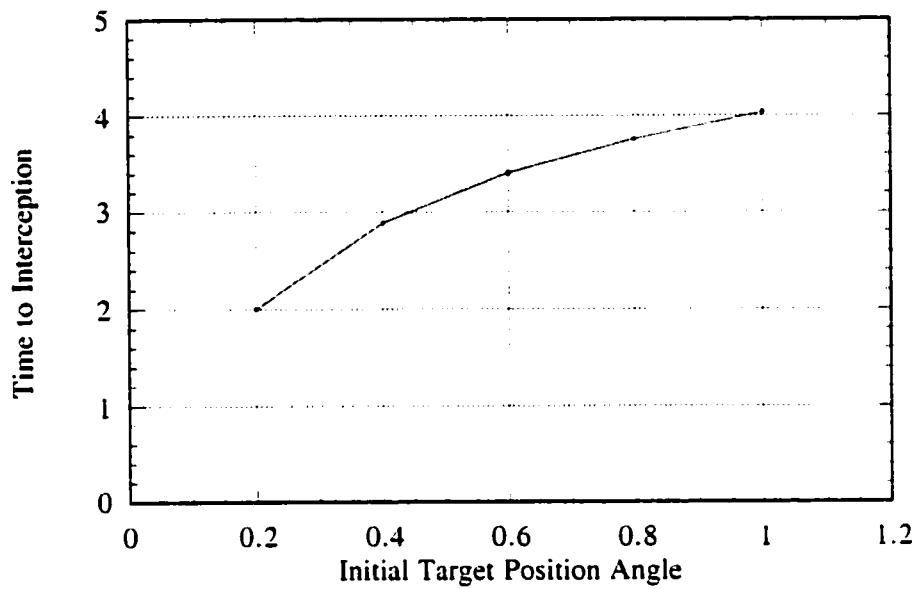


Fig.3.23 Game of Values for Spacecraft Interception/Evasion Problem

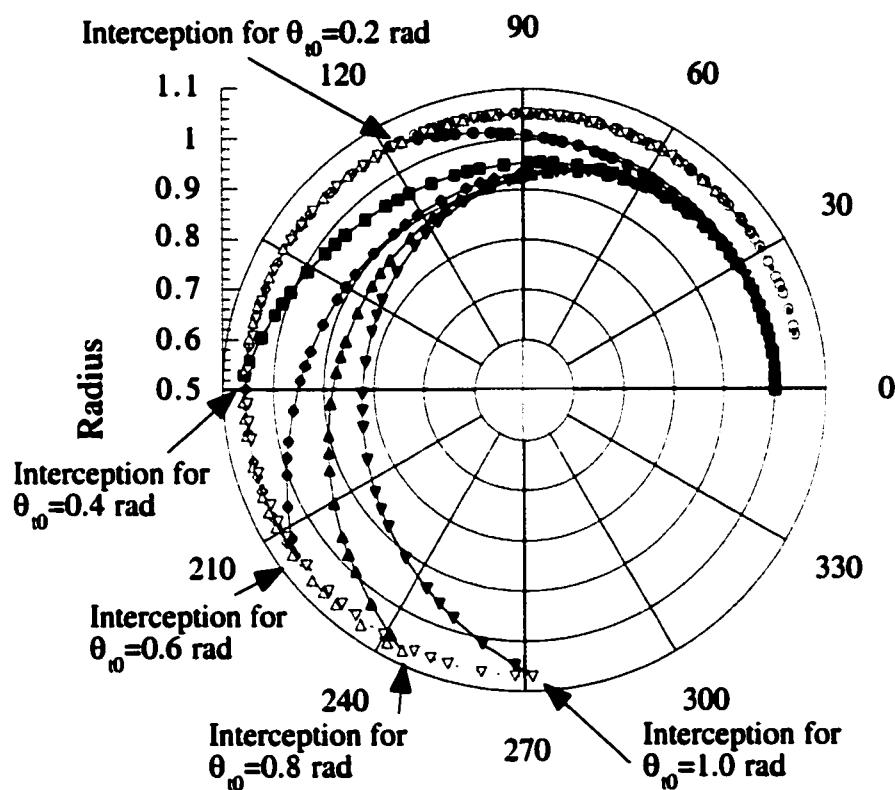


Fig.3.24 A Group of Saddle-Point Trajectories
for the Spacecraft Interception/Evasion Problem

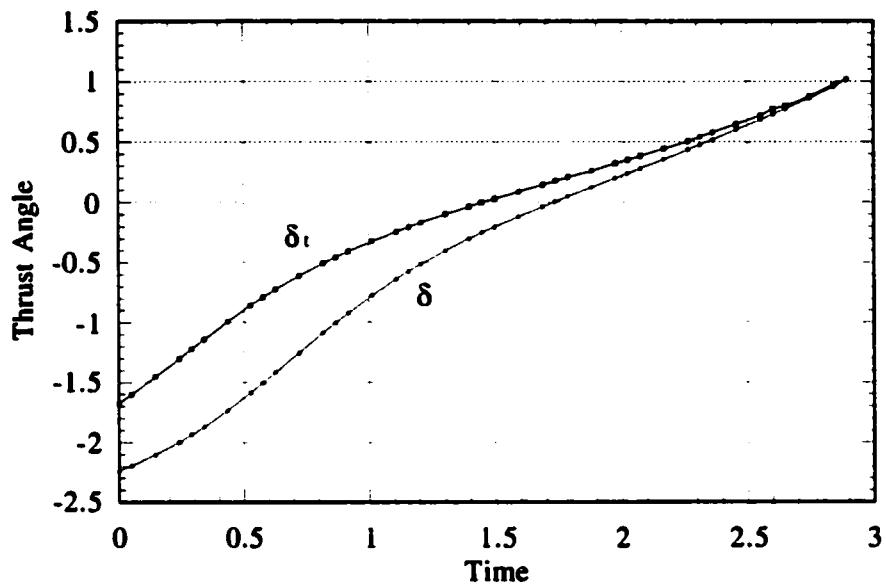


Fig.3.25 Thrust Angle Histories for Spacecraft Interception/Evasion Problem
 $(\theta_{t0}=0.4\text{rad})$

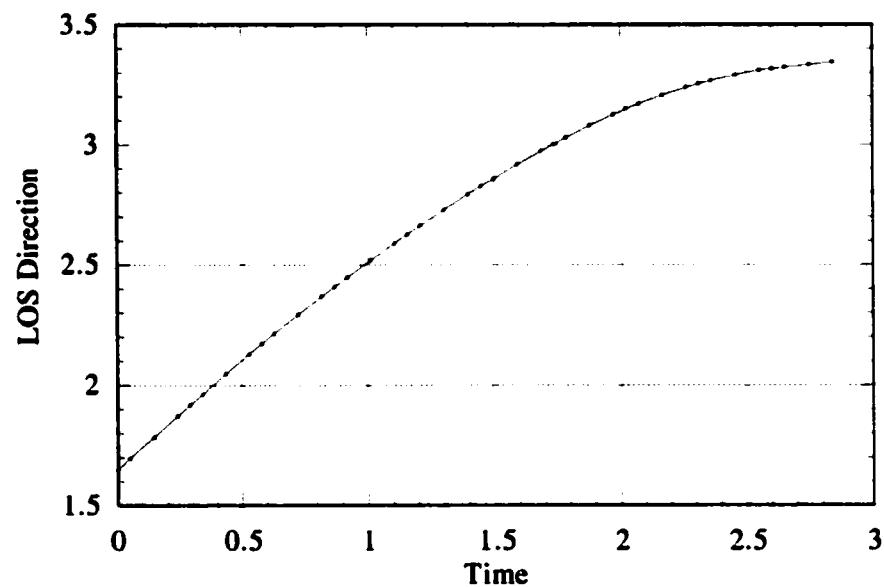


Fig.3.26 Line-of-Sight from Spacecraft to Target
for Spacecraft Interception/Evasion Problem

In addition to the control-unconstrained case shown above, two control-constrained cases are evaluated using the semi-DCNLP method. The first case constrains the target thrust pointing angle to the range $\pi/6 < \delta_t < \pi/3$. The second case constrains the thrust pointing angle for both the spacecraft and the target. In the second case, instead of (3.96) and (3.97), the optimality condition for the spacecraft is:

$$\delta = \arg \min_{\delta} H \quad (3.102)$$

Trajectories and time histories for the control variables for the same initial and final conditions as the unconstrained case are shown in Fig. 3.27 and Fig. 3.28 for the first case and Fig. 3.29 and Fig. 3.30 for the second case. In these figures, the solutions using a shooting method are also provided. The figures show that the semi-DCNLP method provides the same trajectories and control time-histories as the shooting method.

The spacecraft interception problem thus verifies that the semi-DCNLP method can solve a practical pursuit-evasion game when the controls are constrained or unconstrained.

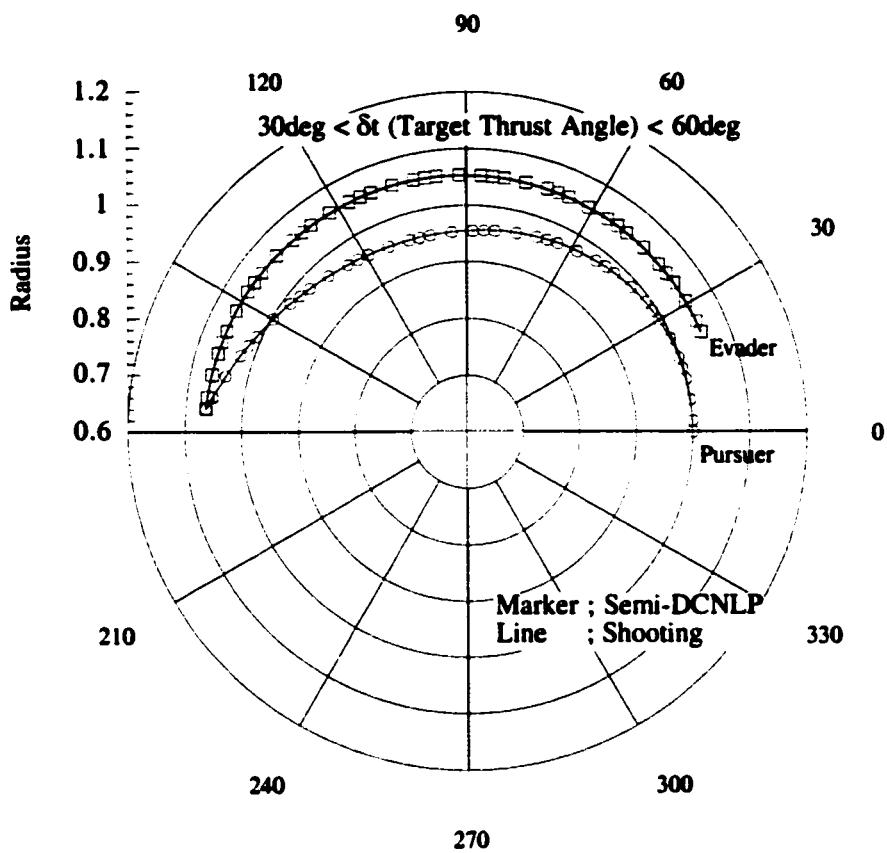


Fig.3.27 Saddle-Point Trajectories for Spacecraft Interception/Evasion Problem
(for Bounded Target Thrust Angle)

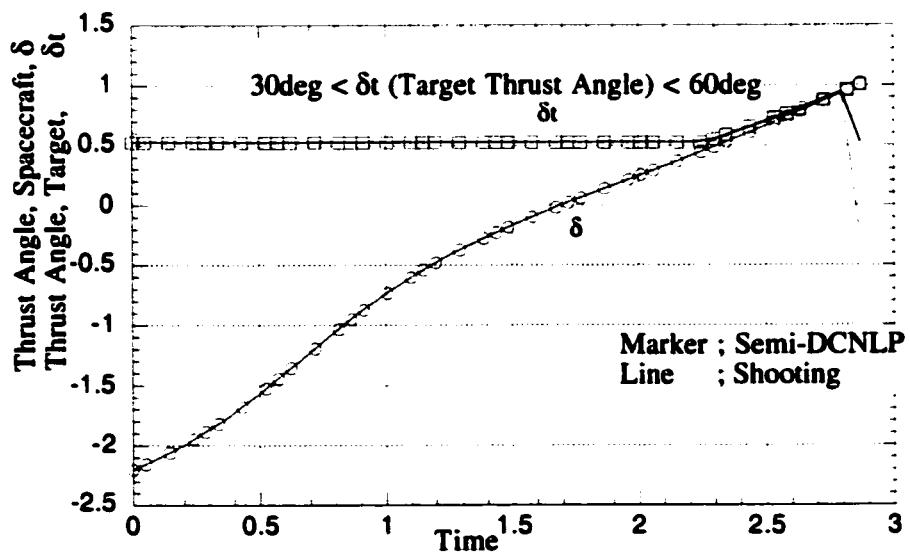


Fig.3.28 Thrust Angle Histories for Spacecraft Interception/Evasion Problem
(for Bounded Target Thrust Angle)

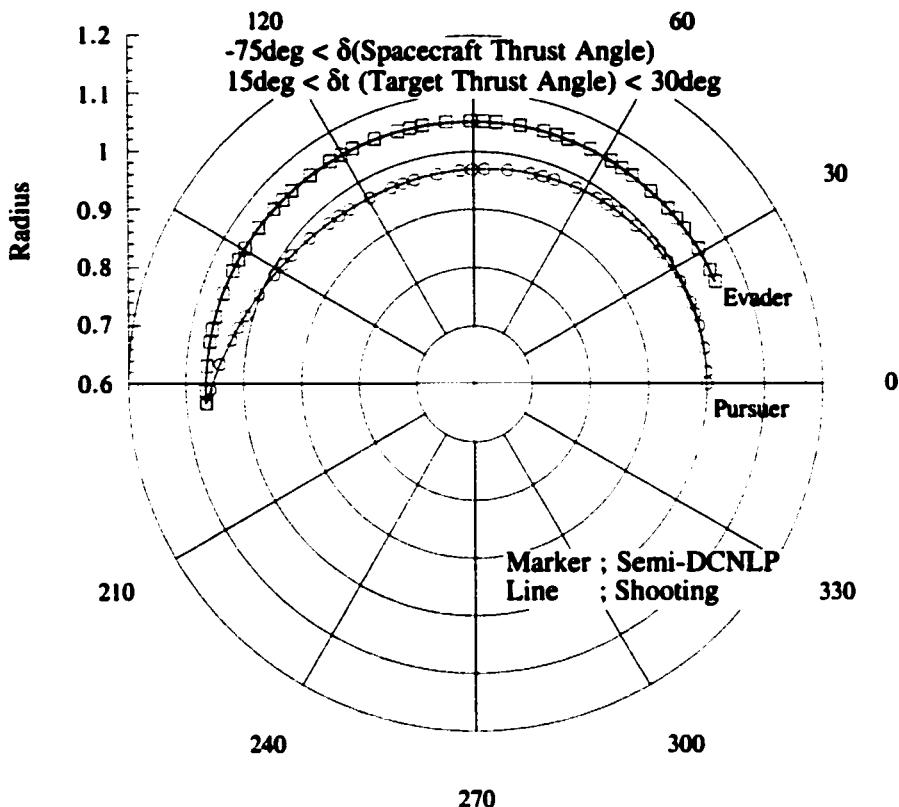


Fig.3.29 Saddle-Point Trajectories for Spacecraft Interception/Evasion Problem
(for Bounded Spacecraft and Target Thrust Angles)

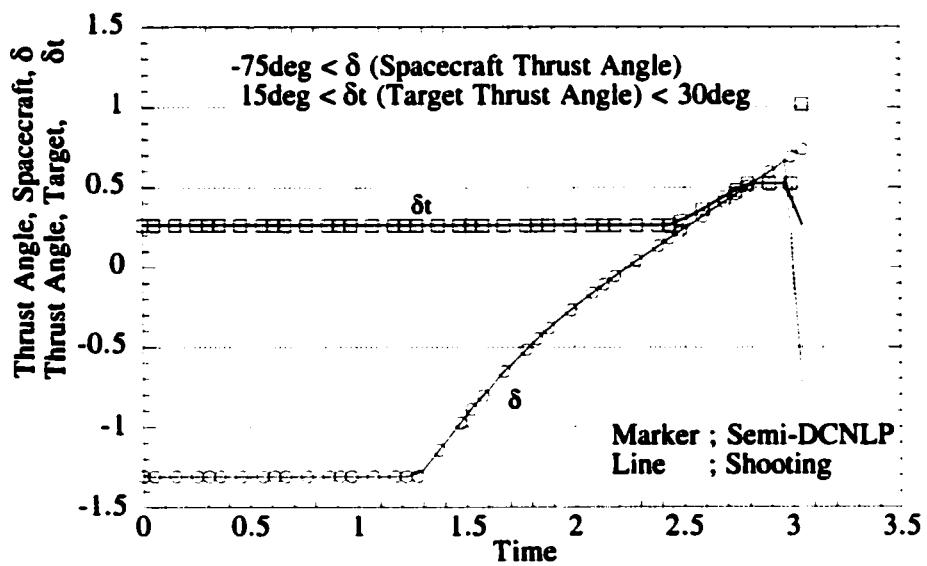


Fig.3.30 Thrust Angle Histories for Spacecraft Interception/Evasion Problem
(for Bounded Spacecraft and Target Thrust Angles)

Chapter 4: Pre-Processing for Collocation with Nonlinear Programming

4.1 Desired Characteristics for the Pre-Processing Algorithm

A nonlinear programming problem solver, for example NZSOL, used when one uses the DCNLP method, requires an initial guess of the parameter vector at the solution. One of the great advantages of the DCNLP method in flight path optimization is robustness, which means tolerance for a very poor initial guess of the solution. In some cases, even very unreasonable initial guesses yield convergent solutions. [13]

That is to say, the DCNLP method provides a convergent solution if an initial guess is in or near the feasible region of the nonlinear programming problem. On the other hand, even the DCNLP method cannot find a convergent solution when the initial guess is “far” from the feasible region. In the case of the semi-DCNLP algorithm, it is even more difficult to prepare an initial guess near the feasible region because the system used by the semi-DCNLP is larger than that of the DCNLP because of the adjoint variables that are required. Thus, it is desirable to develop a pre-processing algorithm to find a good initial guess for the semi-DCNLP algorithm.

The desired characteristics for a pre-processing algorithm are that it should:

- (i) yield an initial guess in or near a feasible region for the nonlinear programming problem and
- (ii) require no initial information regarding the solution.

It is noted that a pre-processing algorithm is not required to find the set of optimal variables for the problem because the DCNLP and semi-DCNLP algorithms can do this very successfully if provided with a satisfactory initial guess.

In this research, a pre-processing algorithm for the method of collocation with nonlinear programming is developed on the basis of a simple genetic algorithm to satisfy these requirements. Then, the problem of the homicidal chauffeur is solved to verify the algorithm developed and evaluate its efficiency.

4.2 Simple Genetic Algorithm as a Pre-Processing Algorithm

4.2.1 Simple Genetic Algorithm Overview

A genetic algorithm (GA) is a search algorithm using the mechanisms of natural genetics to find a set of parameters providing the best fitness. ("fitness" is GA terminology and equivalent to a value of cost.) Simply speaking, a GA provides a large number of individuals representing parameter sets, improves them, and then produces an individual with optimal set of parameters improved. The overview of the mechanism of a simple GA is described in accordance with a text by Goldberg [29]. A numerical analysis code used in this research, which is provided by Carroll [30], is also consistent with the overview.

A simple GA searches for the best individual from a population. Each individual in a population is randomly provided in the search area at the beginning of the

simple GA operation. It is noted that a simple GA has better convergence characteristics as the population size, n , is larger and length of string, l , is generally smaller. A simple GA improves the individuals in a population via three genetic operators in a generation. The improvement progresses when the generation proceeds. Finally, a simple GA operation is convergent when the best individual through a population is maintained even if the generation proceeds.

In this research, the individual strings are expressed in binary data. The string is a combination of binary data coded from the parameters to be optimized. Fig. 4.1 shows the relationship between the optimized parameter and the string. When a string is evaluated, it is decoded to a set of parameters. Using the decoded parameters, a value of the cost function is calculated.

Three operators are employed, reproduction, crossover and mutation. A reproduction is a process to select the individuals surviving to the next generation. A tournament selection is the most popular reproduction operator. The tournament selection compares two individuals and selects one, which has a better value of the cost, for the next generation. The tournament selection repeats until all individuals belonging to the next generation are selected.

A crossover operator is the most characteristic of a "genetic" algorithm. Surviving individuals for the next generation are crossed over as in natural genetics. The simplest crossover operator, single-point crossover, selects two individuals, cuts each

individual string at the same point and exchanges a part of each string under crossover probability p_c .

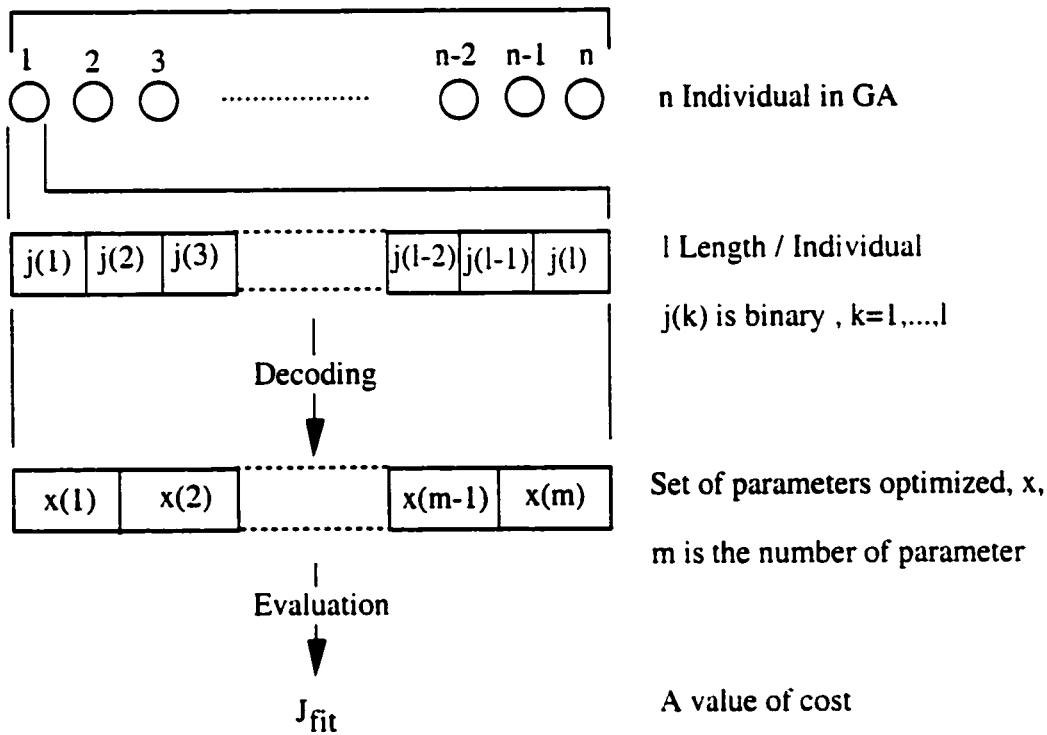


Figure 4.1 Relationship between Optimized Parameters and String

A mutation operator flips a digit in a string under mutation probability p_m . The mutation applies to all digits in all individual strings after operations of reproduction and crossover. In addition, an additional genetic operator, elitism, is usually applied for this research. Elitism keeps the best individual in the generation.

Through this overview of a simple GA mechanism, it is suggested that a simple GA may be a satisfactory pre-processing algorithm for the method of collocation with

nonlinear programming. The advantage of using GA in pre-processing is that, unlike a nonlinear programming problem solver, the GA method does not require an initial guess of the discretized optimal trajectory. It needs only ranges of the parameters as a search area. On the other hand, a simple GA sometimes prematurely converges and the solution is only a "near" optimal solution. However, this will not be critical to the solution process because the semi-DCNLP method will use the GA-obtained solution only as an initial guess for finding the optimal flight path. Indeed, a simple GA is successfully applied to some flight path optimizations [31,32] by hybriding with local optimizers.

4.2.2 Pre-Processing Algorithm

A flight path optimization problem is expressed by the following equations:

$$V = \max_{\bar{u}} J(\bar{x}(t_f), t_f) \quad (4.1)$$

$$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}, t) \quad (4.2)$$

$$\bar{\chi}(\bar{x}(t_0), t_0) = \bar{0} \quad (4.3)$$

$$\bar{\psi}(\bar{x}(t_f), t_f) = \bar{0} \quad (4.4)$$

Eqns. (4.1) – (4.4) include the two-sided optimization problem discussed in Chapter 3, i.e., control variables of one-side are obtained via the Pontryagin principle, by including some of the system adjoint equations as a part of (4.2) and some terminal conditions as a part of (4.4).

A simple GA is applied to the optimization problem (4.1) – (4.4). In a simple GA operation, a cost function is defined as:

$$J_{\text{fit}} = J(\bar{x}(t_f), t_f) - k_w \bar{\psi}(\bar{x}(t_f), t_f)^T \bar{\psi}(\bar{x}(t_f), t_f) \quad (4.5)$$

The terminal constraint (4.4) is dealt with as penalty terms in (4.5) with weighting coefficient, k_w . The dynamic system (4.2) is integrated using a Runge-Kutta method to obtain the state variables at the terminal time. Some of the initial values required for the numerical integration are satisfied with (4.3), and other initial values are optimization parameters. The parameters optimized in a simple GA operation are the final time (if time is open) and discretized control variables in addition to unknown initial values.

Once a simple GA has converged to a solution, unknown initial values, final time (if a problem is time-open) and control variables are obtained from the best individual. Then, (4.2) is again integrated, as an initial value problem, using a Runge-Kutta method. By extracting state and control variables from the continuous forward integration at selected times, corresponding to the nodal times of the discrete DCNLP solution, the initial guess for the semi-DCNLP algorithm is created.

The problem described in (4.1) – (4.4) can be expressed as a TPBVP by fixing the control variables u . Therefore, we first consider a simple GA applied to a TPBVP as an intermediate step to developing a pre-processing algorithm. In these cases, the first term, the terminal cost term, in (4.5) is eliminated.

Example Problem 1: Simple TPBVP

To evaluate the use of a simple GA as a TPBVP solver we consider the problem [33]:

$$\begin{aligned}\frac{d^2y}{dt^2} &= 6t, \quad 0 \leq t \leq 1 \\ y(0) &= 0, \quad y(1) = 1\end{aligned}\tag{4.6}$$

The system of eqn. (4.6) is a time-fixed problem. The only value necessary to solve (4.6) is $dy(0)/dt$. Thus, a string in a simple GA consists of $dy(0)/dt$ only. A cost function for the simple GA is defined as:

$$J_{fit} = -100\{y(1) - 1\}^2\tag{4.7}$$

In this case, GA parameters are chosen as $n=50$, $l=15$, $p_c=0.7$ and $p_m=0.02$. That means that the population consists of 50 individuals, each consisting of 15 binary digits, where this 15 digit binary number decoded, when necessary, to yield the one unknown, $dy(0)/dt$. The convergence history is shown in Fig. 4.2. The correct solution, $dy(0)/dt=0$, appears at the 15th generation. This result shows that a simple GA has the ability to solve a TPBVP.

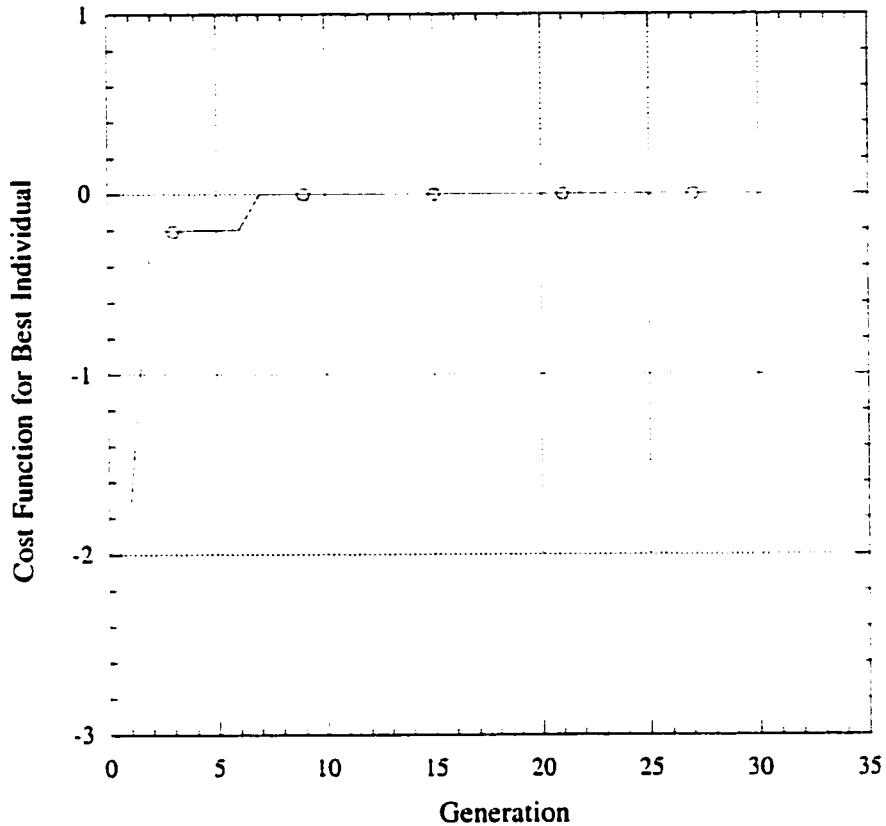


Figure 4.2 Convergence History of the Simple TPBVP

Example Problem 2: Maximum Velocity Transfer to a Rectilinear Path

An optimal control problem, maximum velocity transfer to a rectilinear path in fixed time [2], is solved using a simple GA to verify its ability to solve a complicated TPBVP. The problem is interpreted as a TPBVP by using the calculus of variation and the Pontryagin principle. The system is expressed as [2]:

$$\frac{d}{dt} \begin{bmatrix} u \\ v \\ x \\ y \\ \lambda_u \\ \lambda_v \\ \lambda_x \\ \lambda_y \end{bmatrix} = \begin{bmatrix} a \cos \beta \\ a \sin \beta \\ u \\ v \\ -\lambda_x \\ -\lambda_y \\ 0 \\ 0 \end{bmatrix} \quad (4.8)$$

$$\beta = \tan^{-1} \frac{\lambda_v}{\lambda_u} \quad (4.9)$$

$$(u(0), v(0), x(0), y(0))^T = (0, 0, 0, 0)^T \quad (4.10)$$

$$(v(t_f), y(t_f), \lambda_u(t_f), \lambda_x(t_f))^T = (0, h, 1, 0)^T \quad (4.11)$$

where t_f is final time, a is acceleration and h is final altitude.

Four adjoint variables at the initial time are unknown; the problem is time-fixed. A string in a simple GA consists of four unknown initial adjoint variables as shown in Fig. 4.3.

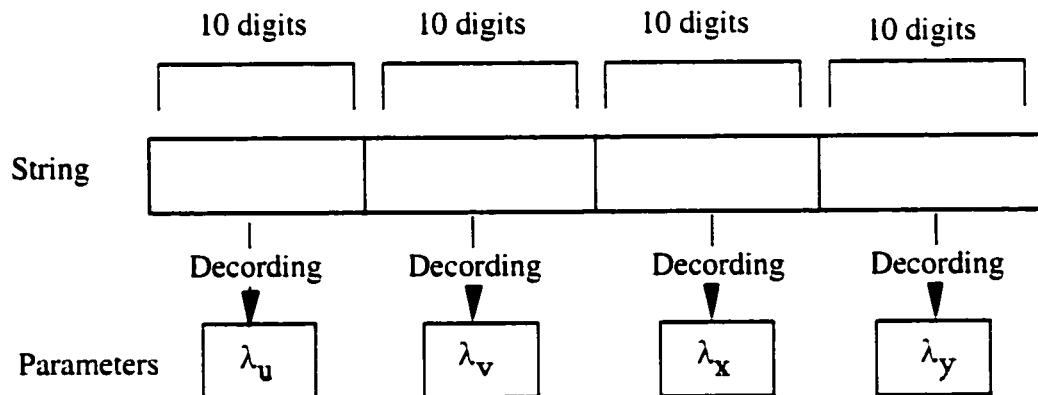


Figure 4.3 Relationship between String and Parameters in Example Problem 2

A cost function for this simple GA is defined as:

$$J_{\text{fit}} = -100 \left\{ v(t_f)^2 + (y(t_f) - h)^2 + (\lambda_u(t_f) - 1)^2 + \lambda_v(t_f)^2 \right\} \quad (4.12)$$

The GA parameters are chosen as $n=100$, $l=40$, $p_c=0.7$ and $p_m=0.01$. The problem is solved for a case with $a=4$, $h=0.6$ and $t_f=1.0$. The convergence history is shown in Fig. 4.4. A convergent solution

$$(\lambda_u, \lambda_v, \lambda_x, \lambda_y)_{t=t_0}^T = (1.007, 1.261, 0.010, 2.493)^T,$$

appears at the 84th generation. The solution is satisfied with the terminal conditions with accuracy of 3.3×10^{-2} at worst. The converged solution yields an initial thrust angle, β_0 , of 51.4° by eqn. (4.9). Bryson and Ho [2] show the relationship between the initial thrust angle and the parameter $4h/at_f^2$ as:

$$\frac{4h}{at_f^2} = \frac{1}{\sin \beta_0} - \left\{ \log \frac{\sec \beta_0 + \tan \beta_0}{\sec \beta_0 - \tan \beta_0} \right\} / 2 \tan^2 \beta_0 \quad (4.13)$$

By using $\beta_0=51.4^\circ$ obtained from the simple GA, eqn. (4.13) yields $4h/at_f^2=0.61$, whereas $4h/at_f^2=0.60$ from initial conditions. Thus GA can solve a complicated TPBVP, which is derived from an optimal control problem.

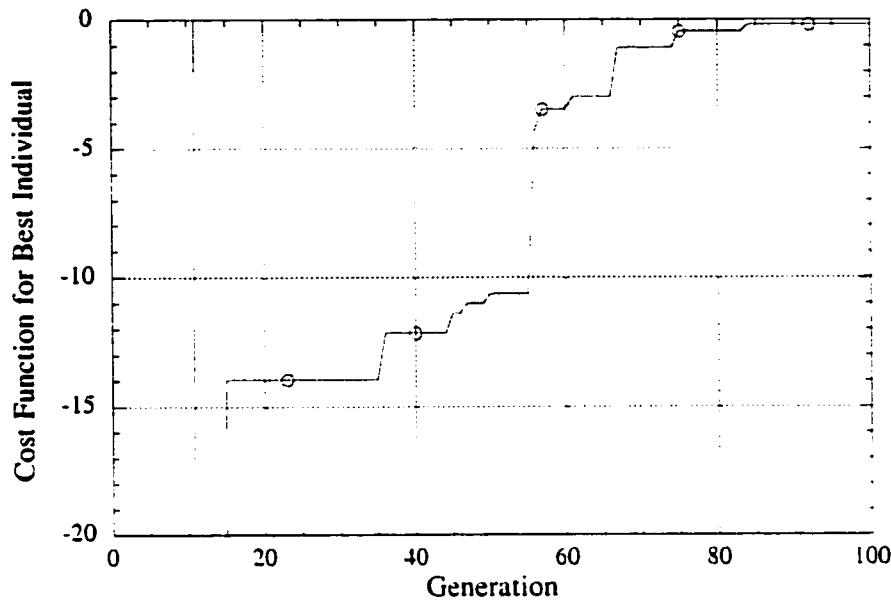


Figure 4.4 Convergence History for Maximum Velocity Transfer to a Rectilinear Path

Example Problem 3: The Homicidal Chauffeur

The homicidal chauffeur problem in relative coordinates is one of most well studied differential game problems and may be regarded as the analogue of a simple air combat game. This differential game problem is selected as an application of the pre-processing algorithm, i.e., the final step for development of GA-based pre-processing.

The system equations necessary for solving the homicidal chauffeur problem, introduced in Sec. 3.4, are re-stated as follows. First, the equation of motion and the control constraints are expressed as:

$$\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -\frac{w_1}{R} y \phi + w_1 \sin \psi \\ \frac{w_1}{R} x \phi - w_1 + w_2 \cos \psi \end{pmatrix} \quad (4.14)$$

$$-1 \leq \phi \leq 1 \quad (4.15)$$

where (x, y) are the Cartesian coordinates of the evader with respect to the pursuing car,

ϕ and ψ are control variables of the pursuer and evader, w_1 is the velocity of the pursuer,

w_2 is velocity of the evader and R is the minimum turning radius of the pursuer.

Termination of the game occurs when the range between the pursuer and evader becomes

the capture range, R_{cap} , i.e.:

$$x^2 + y^2 = R_{cap}^2 \quad (4.16)$$

The cost function of the problem is terminal time:

$$J = t_f \quad (4.17)$$

which means that the pursuer tries to minimize time for capture and the evader to

maximize it.

To use the semi-DCNLP solver, it is required to solve for the control of one player, in this case for the evader heading angle ψ , using the Pontryagin principle.

Then, ψ is found using the Pontryagin principle as:

$$\begin{aligned} \lambda_x \cos \psi - \lambda_y \sin \psi &= 0 \\ \lambda_x \sin \psi + \lambda_y \cos \psi &\leq 0 \end{aligned} \quad (4.18)$$

where adjoint variables are expressed as:

$$\frac{d}{dt} \begin{pmatrix} \lambda_x \\ \lambda_y \end{pmatrix} = \begin{pmatrix} -\lambda_y \frac{w_1}{R} \phi \\ \lambda_x \frac{w_1}{R} \phi \end{pmatrix} \quad (4.19)$$

$$x(t_f) \lambda_y(t_f) - y(t_f) \lambda_x(t_f) = 0 \quad (4.20)$$

Since the control for the evader, ψ , is found using the Pontryagin principle (4.18), final time is already maximized for the evader. Then, using (4.16), (4.17) and (4.20), a cost function for a simple GA is defined as:

$$J_{\text{fit}} = -t_f + 10000 \{(x(t_f)^2 + y(t_f)^2 - R_{\text{cap}}^2)^2 + (x(t_f)\lambda_y(t_f) - y(t_f)\lambda_x(t_f))^2\} \quad (4.21)$$

The control variable ϕ is discretized at three points; initial time, middle time and final time, and these values become parameters to be optimized. Then, a string for this problem consists of six parameters; the terminal time, initial adjoint variables λ_x and λ_y , and three discretized control variables. The control variable ϕ is found outside of the discrete times by linear interpolation. The GA parameters are selected as $n=1000$, $l=60$, $p_c=0.7$ and $p_m=0.001$. Fig. 4.5 shows the relation between the string and the optimized parameters.

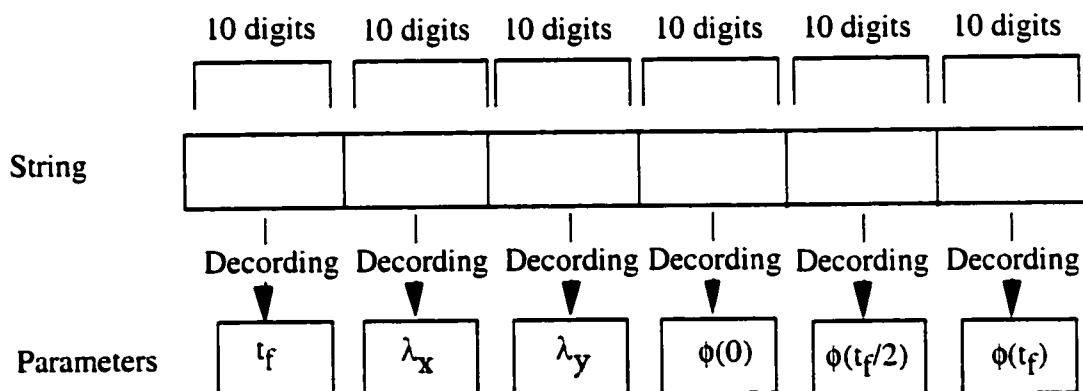


Figure 4.5 Relationship between String and Parameters in Homicidal Chauffeur Problem

The problem is solved for the condition that $(x_0, y_0) = (1.2, 1.0)$, $R_{cap} = 0.8$, $R = 1.0$, $w_1 = 1.0$ and $w_2 = 0.1$. The convergence history is shown in Fig. 4.6. A convergent solution appears at the 48th generation. The solution satisfies the terminal conditions with error of less than 1×10^{-3} . However, the discretized control variables ϕ are [0.865 0.977 0.658] whereas the optimal ϕ should be 1 throughout the trajectories in accordance with the results of Sec. 3.4. Thus, the simple GA is convergent to a “near” optimal set of parameters, in addition to finding a feasible set of parameters. Since this is all that is required for a pre-processing algorithm, the simple GA has accomplished the pre-processing.

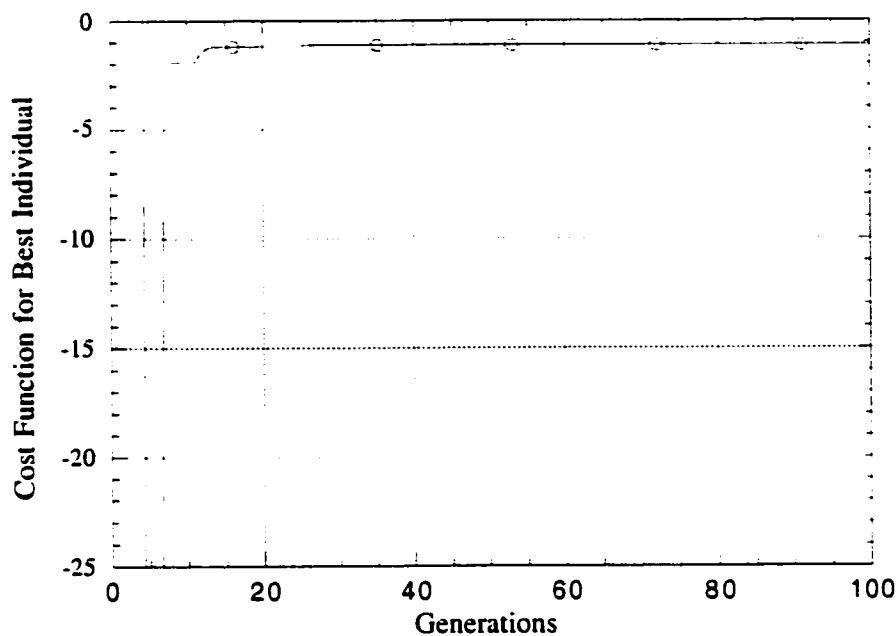


Figure 4.6 Convergence History of Homicidal Chauffeur Problem

To verify that the simple GA solution is satisfactory as an initial guess, the semi-DCNLP method is now used to solve the homicidal chauffeur problem starting from the initial guess obtained from the simple GA. After seven iterations of the semi-DCNLP algorithm, the resulting trajectory is shown in Fig. 4.7. It is consistent with the result of Sec. 3.4. Therefore, it is concluded that a simple GA is an appropriate pre-processing algorithm for the semi-DCNLP method.

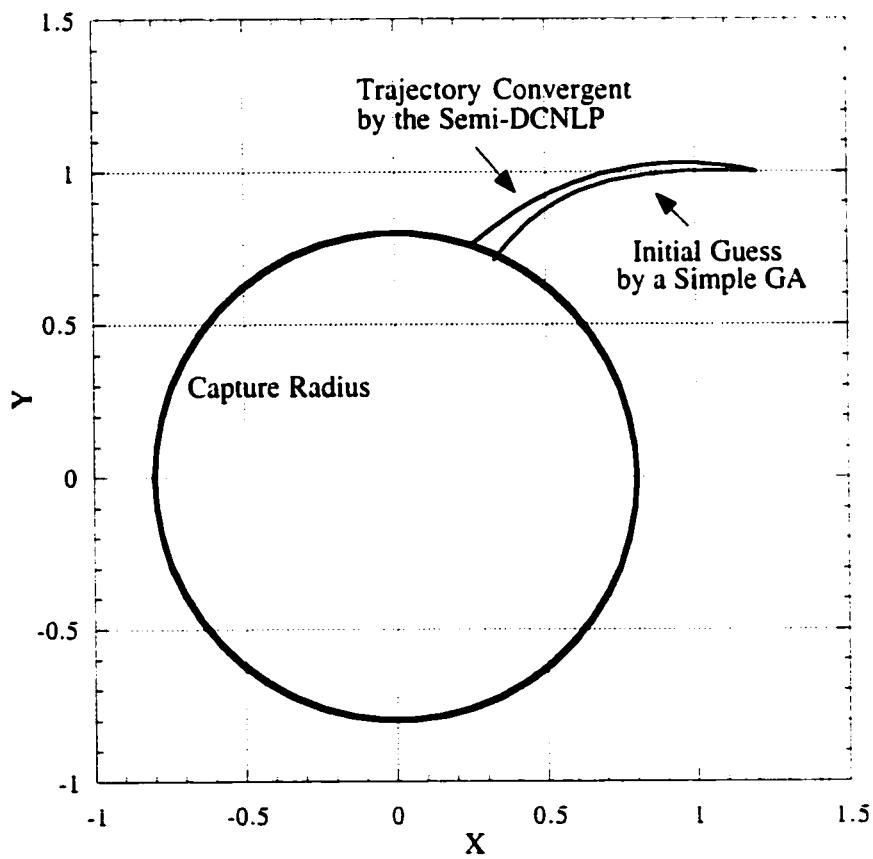


Figure 4.7 Trajectories at Initial Guess and Saddle-Point

4.3 Efficiency of the Pre-Processing Algorithm

A hybridization between a GA and a local optimization method is often used to improve efficiency and reliability of a numerical optimization. The structure of the hybridization is shown in Fig. 4.8. In the hybridization, a GA uses a local optimizer to find the value of cost of individuals. A local optimization method such as the semi-DCNLP solves an optimization problem (4.1) - (4.4) for each individual starting from parameters provided by a GA. This means that the hybrid GA/local optimizer can find the optimal point not only via direct hitting by a GA operation but by hitting on a point in or near a feasible region.

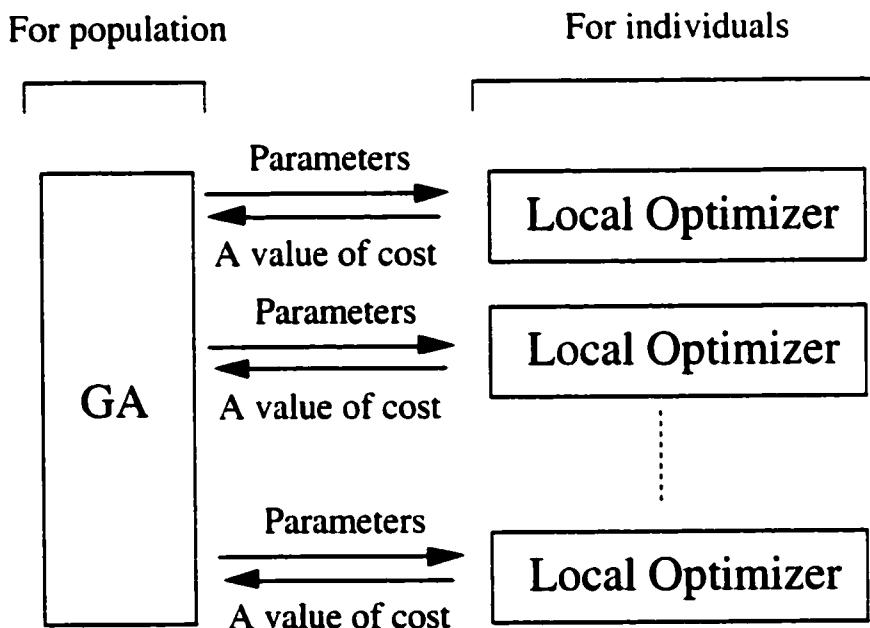


Figure 4.8 Structure of Hybrid Algorithm of GA and Local Optimizer

Goldberg and Voessner [34] construct a system-level theoretical framework of optimizing the hybridization between a GA and a local optimization method. They analyze the efficiency and reliability of a hybrid algorithm. The following discussion related to efficiency of the pre-processing algorithm is developed on the basis of their work.

The operation time of a hybrid algorithm, T, is given as:

$$T = (1 + \lambda)n_g \quad (4.22)$$

where n_g is the number of the generation and λ is the average time consumed by a local optimizer in a generation. It is noted that time is normalized by average time for a GA in a generation.

Hitting a point in or near the feasible region, which Goldberg calls the "basin of attraction", does not always bring an optimal point within the allocated time to a local optimizer, λ_a . The probability of finding an optimal point within allowable time after hitting basins or optimal point in a generation is introduced as P_{λ_a} . Using P_{λ_a} , a probabilistic error, α , is expressed as:

$$\alpha = (1 - P_{\lambda_a})^{n_g} \quad (4.23)$$

Allowable probabilistic error, α_a , is introduced to derive minimum time subject to given reliability. Then, the number of generations is derived for a given allowable probabilistic error as:

$$n_g = \frac{\ln \alpha_a}{\ln (1 - P_{\lambda a})} \quad (4.24)$$

Finally, the operation time (4.22) becomes:

$$T = (1 + \lambda) \frac{\ln \alpha_a}{\ln (1 - P_{\lambda a})} \quad (4.25)$$

This research introduces a ratio of times of a local optimizer operation to total times of evaluation, k_L , and average time consumed by a local optimizer for an individual, λ_L . Using these parameters, λ is defined as:

$$\lambda = k_L \lambda_L n \quad (4.26)$$

By substituting (4.26) into (4.25),

$$T = (1 + k_L \lambda_L n) \frac{\ln \alpha_a}{\ln (1 - P_{\lambda a})} \quad (4.27)$$

Maximum k_L is 1. At maximum k_L , the local optimizer works for evaluating every individual in every generation. On the other hand, minimum k_L makes the local optimizer work only for one individual through all generations and is expressed as:

$$k_L = \frac{1}{n \cdot n_g} = \frac{\ln (1 - P_{\lambda a})}{n \ln \alpha_a} \quad (4.28)$$

Thus, the combination of the GA pre-processing and the semi-DCNLP algorithm developed in Sec. 4.2 is regarded as an extreme case, i.e., a case of minimum k_L , of the hybridization. Then, the operation time is expressed by (4.29) for the standard hybrid GA/local optimizer ($k_L=1$) and by (4.30) for the GA pre-processing and the semi-DCNLP algorithm:

$$T_{STD} = (1 + \lambda_L n) n_{g,STD} = (1 + \lambda_L n) \frac{\ln \alpha_a}{\ln(1 - P_{\lambda_a, \max k_L})} \quad (4.29)$$

$$T_{PreGA} = n_{g,PreGA} + \lambda_L = \frac{\ln \alpha_a}{\ln(1 - P_{\lambda_a, \min k_L})} + \lambda_L \quad (4.30)$$

In general, the calculation time of the semi-DCNLP, λ_L , is much larger than that of a GA. Then

$$\begin{aligned} T_{STD} - T_{PreGA} &= (1 + \lambda_L n) n_{g,STD} - (n_{g,PreGA} + \lambda_L) \\ &\approx \lambda_L (n \cdot n_{g,STD} - 1) - n_{g,PreGA} \\ &\approx \lambda_L n \cdot n_{g,STD} - n_{g,PreGA} \end{aligned} \quad (4.31)$$

Qualitatively speaking, (4.31) is positive because usually both λ_L and n are large.

Therefore, a standard hybrid GA/semi-DCNLP is predicted to be a less time efficient method than the GA pre-processing followed by the semi-DCNLP algorithm.

To support the qualitative discussion, the comparison between the standard hybrid method and the pre-processing method is done for the calculation of the homicidal chauffeur trajectory.

The standard hybrid GA/semi-DCNLP is constructed as shown in Fig. 4.8. A local optimizer in Fig. 4.8 is the semi-DCNLP. A GA is also constructed using the method described in Sec. 4.2, except for the cost function. The cost function is the terminal time if the semi-DCNLP is converged. If not, the terminal time as an output of the semi-DCNLP is multiplied by a weighting factor, 1×10^2 in this case. The result of the

semi-DCNLP does not change the parameters in a simple GA, i.e., using a Baldwinian concept [35], because of avoiding a loss of diversity of the individual.

The problem of the homicidal chauffeur, for the same conditions as Sec. 4.2, is solved using the standard hybrid GA/semi-DCNLP ($n=10$ and $n=100$) and the GA pre-processing and subsequent semi-DCNLP algorithm ($n=100$ and $n=1000$). Convergence histories are shown in Figs. 4.9 and 4.10. Fig. 4.9 shows the convergence histories for the operation time which is normalized by average GA operation in a generation (from the result of a simple GA for 100 generation at $n=1000$) whereas Fig. 4.10 shows the convergence histories for the generation.

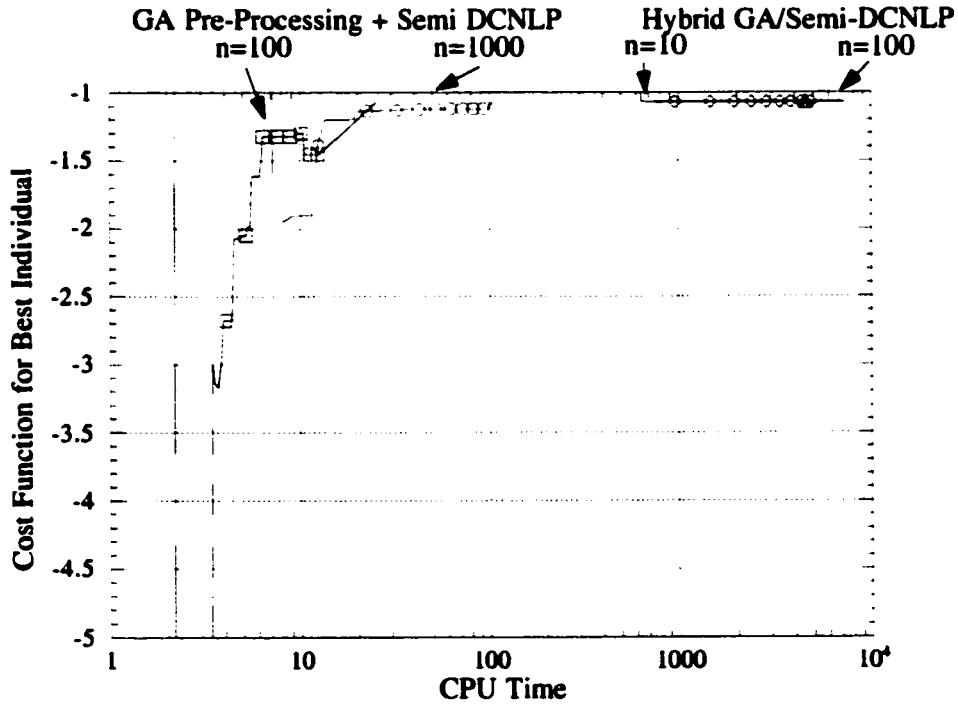


Figure 4.9 Convergence Histories for Standard and Hybrid GA Pre-Processors

Fig. 4.9 suggests that the GA pre-processing and subsequent semi-DCNLP is more efficient than the standard hybrid GA/semi-DCNLP. The GA pre-processing solves the problem in around one-tenth of the time of the standard hybrid GA/semi-DCNLP. Because the semi-DCNLP needs about seven units of time for an individual, the GA pre-processing becomes more efficient.

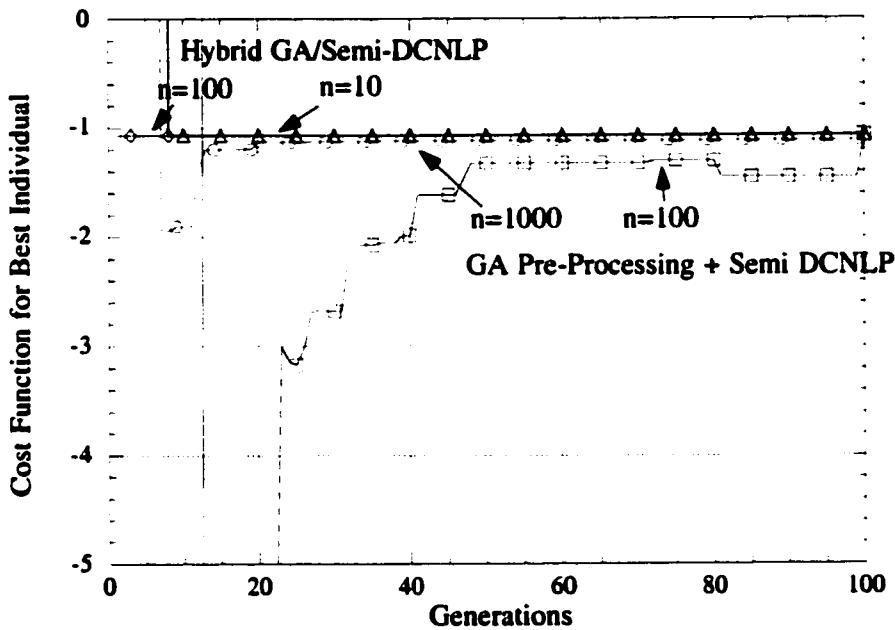


Figure 4.10 Convergence Histories for Standard and Hybrid GA Pre-Processors by Generation

One the other hand, Fig. 4.10 shows the standard hybrid GA/semi-DCNLP converges within ten generations. In Fig. 4.10, the pre-processing GA does not converge (converged pre-maturely) even at hundreds of generations without the subsequent semi-DCNLP operation. This supports the conclusion that the standard hybrid method

drastically improves P_{λ_a} and then expects to find an optimal point with a small number of generations.

We conclude that while the semi-DCNLP method needs a lot of time to get the converged solution it is suggested that GA pre-processing and subsequent use of semi-DCNLP is the efficient solution method. However, the hybrid GA/semi-DCNLP will be a more useful method when computer capability is improved in the future, because of the smaller number of generations required for convergence.

Chapter 5: The Air Combat Problem

5.1 Problem Definition

In this chapter we will consider realistic air combat using the new numerical solver developed in Chapter 3. This research focuses on the case of air combat between a superior fighter aircraft and an inferior fighter aircraft. A superior fighter, which has high angle-of-attack (AOA) flight capability and thrust power with an afterburner, will attack an inferior fighter, which has conventional AOA flight capability and thrust power without afterburner.

In general, a superior fighter aircraft will succeed in combat with an inferior fighter aircraft. Therefore, the completion of the maneuver occurs when a superior fighter takes a shooting position on the inferior fighter. The superior fighter intends to minimize time of the completion of the maneuver, t_f , whereas the inferior fighter tries to maximize it. The air combat problem is thus a pursuit-evasion problem, a type of differential game, in which the cost function of the game is the time of completion.

A three degree-of-freedom, point-mass model is used to describe the aircraft trajectories. The trajectories of the aircraft are represented using six variables; velocity, v , flight path angle, γ , heading angle, ψ , down range, x , cross range, y , and altitude, h . Angle of attack, α , and bank angle, ϕ , are used for the control of the fighter aircraft. The set of equations of motion is provided as following on the basis of the work of Miele [36]:

$$\frac{dv_i}{dt} = \frac{1}{m_i}(T_i \cos \alpha_i - D_i) - g \sin \gamma_i \quad (5.1)$$

$$\frac{dy_i}{dt} = \frac{1}{m_i v_i} (T_i \sin \alpha_i + L_i) \cos \phi_i - \frac{g}{v_i} \cos \gamma_i \quad (5.2)$$

$$\frac{d\psi_i}{dt} = \frac{1}{m_i v_i \cos \gamma_i} (T_i \sin \alpha_i + L_i) \sin \phi_i \quad (5.3)$$

$$\frac{dx_i}{dt} = v_i \cos \gamma_i \cos \psi_i \quad (5.4)$$

$$\frac{dy_i}{dt} = v_i \cos \gamma_i \sin \psi_i \quad (5.5)$$

$$\frac{dh_i}{dt} = v_i \sin \gamma_i \quad (5.6)$$

The suffix i indicates a superior fighter aircraft (the pursuer) when $i=p$ and an inferior fighter aircraft (the evader) when $i=e$. The lift force, L , and the drag force, D , are defined as:

$$L_i = \frac{1}{2} \rho(h_i) v_i^2 S_i C_L(\alpha_i) \quad (5.7)$$

$$D_i = \frac{1}{2} \rho(h_i) v_i^2 S_i C_D(\alpha_i) \quad (5.8)$$

A standard technique in numerical analysis, normalization, is applied to solve the numerical problem. Normalization avoids reducing the accuracy of the variables by keeping them to similar order and then stabilizes the process of numerical analysis. All the quantities are normalized by the reference time, length and mass in Table 5.1.

Reference	Normalizing Quantities
Mass	637.16(slug)
Time	10.000 (sec)
Length	4000.0 (ft)

Table 5.1 Reference Values for Normalization

The gravitational acceleration, g , and the wing area, S , are constant. The mass of the aircraft, m , is set as constant because the air combat time is generally brief. Thrust, T , is also assumed to be constant because a fighter aircraft uses its maximum power in air combat. The models for atmospheric density, ρ , and the thrust, T , will be described in the next section. Some of the constant parameters of the problem are shown in Table 5.2.

Parameters	Dimensional Quantities	Normalized Quantities
Mass, m	637.16(slug)	1.00000×10^0
Wing Area, S	300.0(ft^2)	1.87500×10^{-5}
Gravitational Acceleration, g	32.174(ft/s^2)	8.04350×10^{-1}

Table 5.2 Typical Parameters in the Air Combat Problem

In this research, both fighter aircraft are modeled on the F-16A [5]. F-16A wind tunnel testing [37] provides the following aerodynamic characteristics (see Sec. 2.3 in detail):

$$C_L = \begin{cases} 0.0174 + 4.3329\alpha - 1.3048\alpha^2 \\ \quad + 2.2442\alpha^3 - 5.8517\alpha^4 (0 \leq \alpha \leq \pi/6) \\ -1.3106 + 10.7892\alpha - 9.2317\alpha^2 \\ \quad - 1.1194\alpha^3 + 2.1793\alpha^4 (\pi/6 \leq \alpha \leq \pi/3) \\ 24.6577 - 71.0446\alpha + 83.1234\alpha^2 \\ \quad - 44.0862\alpha^3 + 8.6582\alpha^4 (\pi/3 \leq \alpha \leq \pi/2) \end{cases} \quad (5.9)$$

$$C_D = \begin{cases} 0.0476 - 0.1462\alpha + 0.0491\alpha^2 \\ \quad + 12.8046\alpha^3 - 12.6985\alpha^4 (0 \leq \alpha \leq \pi/6) \\ 0.5395 - 5.7972\alpha + 21.6625\alpha^2 \\ \quad - 21.6213\alpha^3 + 7.0364\alpha^4 (\pi/6 \leq \alpha \leq \pi/3) \\ 16.6957 - 52.5918\alpha + 67.3227\alpha^2 \\ \quad - 37.0862\alpha^3 + 7.4807\alpha^4 (\pi/3 \leq \alpha \leq \pi/2) \end{cases} \quad (5.10)$$

A terminal condition is required to solve the air combat problem as a pursuit-evasion problem. In the real world, many fighter aircraft have a rear-aspect short-range missile and/or gun for attack. Thus tail-chasing is one of the traditional tactics of air combat. Tail-chasing is used as a proper terminal condition in the following sections.

A progressive approach is applied to reach the final objective, to know the characteristics of three-dimensional realistic air combat. In the first step, the problem is simplified in order to obtain insight regarding the complicated, realistic air combat problem. The problem is simplified by transforming (5.1) - (5.10) on the basis of the

work of Breitner et.al [12]. It is expected that the simplified air combat problem provides an intermediate case between the homicidal chauffeur problem and realistic three-dimensional air combat. A two-dimensional case is solved first. Then, the solution of the three-dimensional air combat maneuvering as a pursuit-evasion game is obtained using the semi-DCNLP method. The problem is solved for various initial conditions and parameters. The obtained solutions show quantitative and qualitative characteristics of three dimensional optimal air combat maneuvering.

5.2 Simplified Air Combat

5.2.1 Two-Dimensional Problem

This sub-section focuses on the simplified two-dimensional air combat problem. The set of equations of motion, (5.1) - (5.8), is simplified by the method of Breitner et.al [12], which is applied to the problem of vertical-plane maneuvering between an aircraft and missile as a pursuit-evasion game. To transform the three-dimensional problem to two dimensions the aircraft heading angle and bank angle are forced to zero, putting all maneuvers in a vertical plane.

Breitner assumes that angle of attack (AOA) is small and that the lift force on the aircraft cancels the component of the gravitational force normal to the aircraft, i.e.,

$$L_i = m_i g \cos \gamma_i \quad (5.11)$$

$$C_{Li} = \frac{m_i g \cos \gamma_i}{\frac{1}{2} \rho v_i^2 S_i} \quad (5.12)$$

Eqn. (5.12) implies that the lift can be determined by flight path angle, rather than angle of attack. This implies that (5.2) is eliminated from the set of equations. Then flight path angle becomes a control variable instead of angle of attack. Also, the drag force is expressed as:

$$D_i = \frac{1}{2} \rho v_i^2 S_i (C_{D_0i} + k_i C_{Li}^2) \quad (5.13)$$

The drag co-efficient is a second order function of the lift coefficient in (5.13).

Substituting (5.13) into the system equations and making heading angle zero yields:

$$\frac{dv_i}{dt} = \frac{1}{m_i} \left(T_i - \frac{1}{2} \rho v_i^2 S_i C_{D_0i} - \frac{2k_i m_i^2 g^2}{\rho_i v_i^2 S_i} \cos^2 \gamma_i \right) - g \sin \gamma_i \quad (5.14)$$

$$\frac{dx_i}{dt} = v_i \cos \gamma_i \quad (5.15)$$

$$\frac{dh_i}{dt} = v_i \sin \gamma_i \quad (5.16)$$

Breiter's assumption is reasonable when flight path angle and AOA are small. If not, the motion of the aircraft is not well represented. For example, in spite of requiring

more normal force at large flight path angle, this model needs less force for large flight path angle than for 1G level flight. Breitner agrees in his paper that his model includes some deficiencies [12].

The relationship between the drag coefficient and the lift coefficient used in (5.13) is shown in Fig. 5.1. A drag polar in Fig. 5.1 is obtained by least square fitting on the basis of (5.9) and (5.10). As a result, the quantities in (5.13) are provided as $C_{D0}=0.0165$ and $k_i=k=0.1676$.

The thrust of the pursuer, T_p , is set equal to the weight of the aircraft, so that the thrust-weight ratio becomes one. The thrust of the evader, T_e , is set to half of that of the pursuer. It is noted that atmospheric density, ρ , is assumed to be constant, 1.76340×10^5 (as a non-dimensional quantity), at 10,000 ft altitude. This also simplifies the problem because (5.14) does not become a function of the altitude.

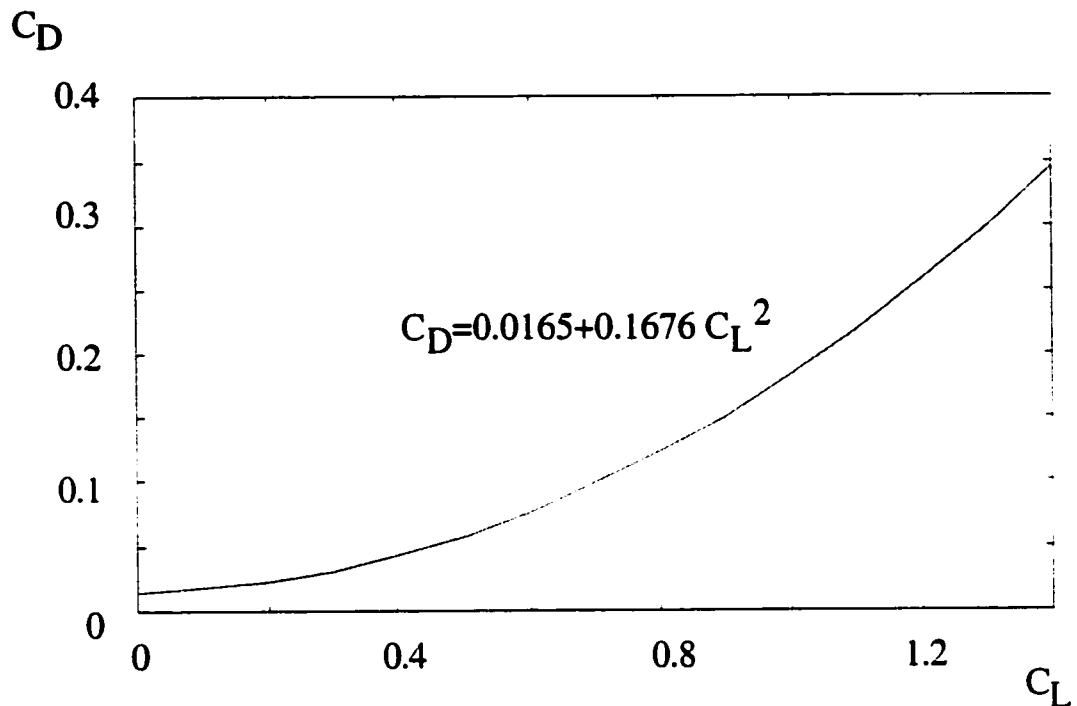


Figure 5.1 Approximate Drag Polar of F-16A Aircraft

The terminal condition in the problem is that the pursuer reaches a point behind an evader having a specified horizontal separation but the same altitude. Then, the set of terminal conditions becomes:

$$x_e - x_p = r_{\text{shoot}} \quad (5.17)$$

$$h_e - h_p = 0 \quad (5.18)$$

Here, a two-sided flight path optimization problem as a class of a pursuit-evasion game is constructed as:

$$V = \min_{\gamma_p} \max_{\gamma_e} t_f \quad (5.19)$$

subject to (5.14) - (5.16) for both a pursuer and an evader, (5.17), (5.18) and a given initial condition.

The semi-DCNLP method is applied to solve this problem of a pursuit-evasion game. First we must derive adjoint equations for the pursuer and their boundary conditions. The Hamiltonian of the problem is:

$$H = \lambda_{vp} \left\{ \frac{1}{m} (T_p - \frac{1}{2} \rho v_p^2 S C_{D_0} - \frac{2km^2 g^2}{\rho v_p^2 S} \cos^2 \gamma_p) - g \sin \gamma_p \right\} + \lambda_{xp} v_p \cos \gamma_p + \lambda_{hp} v_p \sin \gamma_p + H_e \quad (5.20)$$

where H_e is terms of Hamiltonian related to the state of an evader. Then, the adjoint equations for the pursuer are:

$$\frac{d\lambda_{vp}}{dt} = -\frac{\partial H}{\partial v_p} = \frac{\lambda_{vp}}{m} (\rho v_p S C_{D_0} - \frac{4km^2 g^2}{\rho v_p^3 S} \cos^2 \gamma_p) - \lambda_{xp} \cos \gamma_p - \lambda_{hp} \sin \gamma_p \quad (5.21)$$

$$\frac{d\lambda_{xp}}{dt} = -\frac{\partial H}{\partial x_p} = 0 \quad (5.22)$$

$$\frac{d\lambda_{hp}}{dt} = -\frac{\partial H}{\partial h_p} = 0 \quad (5.23)$$

The Pontryagin minimum principle provides optimality conditions for the pursuer as:

$$\frac{\partial H}{\partial \gamma_p} = \lambda_{vp} \left\{ \frac{2km g^2}{\rho v_p^2 S} \sin(2\gamma_p) - g \cos \gamma_p \right\} - \lambda_{xp} v_p \sin \gamma_p + \lambda_{hp} v_p \cos \gamma_p = 0 \quad (5.24)$$

The terminal conditions of the adjoint variables are derived as:

$$\Phi = t_f + v_1(x_e - x_p - r_{shoot}) + v_2(h_e - h_p) \quad (5.25)$$

$$\lambda_{vp}(t_f) = \frac{\partial \Phi}{\partial v_p(t_f)} = 0 \quad (5.26)$$

$$\lambda_{xp}(t_f) = \frac{\partial \Phi}{\partial x_p(t_f)} = -v_1 \quad (5.27)$$

$$\lambda_{hp}(t_f) = \frac{\partial \Phi}{\partial h_p(t_f)} = -v_2 \quad (5.28)$$

Eqn. (5.27) and (5.28) include unknown variables v_1 and v_2 . Thus only (5.26) can be used as a boundary condition to solve the problem, i.e. the set of equations for the optimal flight path angle of the pursuer is constructed from (5.21) - (5.24) and (5.26).

Eqn. (5.27) shows that λ_{xp} is constant. Dividing the other adjoint variables by λ_{xp} introduces new variables as:

$$\lambda_{vp}^* = \frac{\lambda_{vp}}{\lambda_{xp}} \quad (5.29)$$

$$\lambda_{hp}^* = \frac{\lambda_{hp}}{\lambda_{xp}} \quad (5.30)$$

This eliminates (5.22) and then λ_{xp} from the set of the adjoint equations. Hence, a new set of equations for the optimal flight path angle for a pursuer are described as:

$$\frac{d\lambda_{vp}^*}{dt} = \frac{\lambda_{vp}^*}{m} (\rho v_p S C_{D_0} - \frac{4km^2 g^2}{\rho v_p^3 S} \cos^2 \gamma_p) - \cos \gamma_p - \lambda_{hp}^* \sin \gamma_p \quad (5.31)$$

$$\frac{d\lambda_{hp}^*}{dt} = 0 \quad (5.32)$$

$$\lambda_{vp}'' \left\{ \frac{2kmg^2}{\rho v_p^2 S} \sin(2\gamma_p) - g \cos \gamma_p \right\} - v_p \sin \gamma_p + \lambda_{hp}'' v_p \cos \gamma_p = 0 \quad (5.33)$$

$$\lambda_{vp}''(t_f) = 0 \quad (5.34)$$

Finally, the semi-DCNLP formulation for the problem becomes:

$$V = \max_{\gamma_e} t_f \quad (5.35)$$

subject to (5.14) - (5.18) and (5.31) - (5.34) for a given initial condition.

To find an initial guess of the solution for the semi-DCNLP problem, the genetic algorithm is applied on the basis of the concept introduced in Chapter 4. A fitness function for this problem is:

$$J_{fit} = -100 \{(x_e - x_p - r_{shoot})^2 + (h_e - h_p)^2 + \lambda_{vp}''^2\} |_{t=t_f} \quad (5.36)$$

The characteristics of the genetic algorithm are provided as $n=1000$ (population size), $l=42$ (length of string), $p_c=0.6$ (crossover probability) and $p_m=0.001$ (mutation probability). Unknown parameters are t_f , $\lambda_{vp}''(0)$, $\lambda_{vp}''(0)$, $\gamma_e(0)$, $\gamma_e(t_f/2)$ and $\gamma_e(t_f)$. The first three parameters are decoded from 10 digits each and the last three are from 4 digits each.

A simple GA program provides a convergent solution for the initial condition. $v_{p,0}=v_{e,0}=1.0$, $h_{p,0}=h_{e,0}=2.5$, $x_{p,0}=0$ and $x_{e,0}=1.0$. The convergence history is shown in Fig. 5.2.

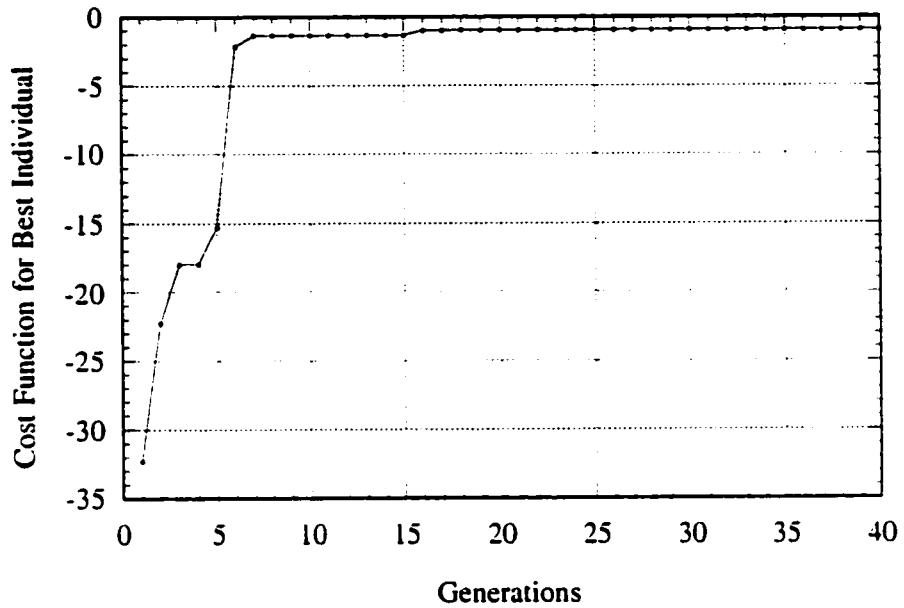


Figure 5.2 Convergence History of GA Pre-Processing for Simplified Two-Dimensional Air Combat

The semi-DCNLP algorithm yields a convergent solution for this initial condition. It requires 18 major iterations of NZSOL, which starts from the initial guess provided by the simple GA. Figure 5.3 shows the histories of the adjoint variables to validate that the obtained trajectories satisfy the optimality condition. The line shows the histories of the modified adjoint variables, (5.29) and (5.30), using the semi-DCNLP formulation, whereas the symbols show the Lagrange multipliers which are obtained from NZSOL directly. The semi-DCNLP method provides the correct saddle-point

trajectory numerically when the symbols lie on the lines. Hence, Fig. 5.3 shows the obtained trajectories are numerically correct.

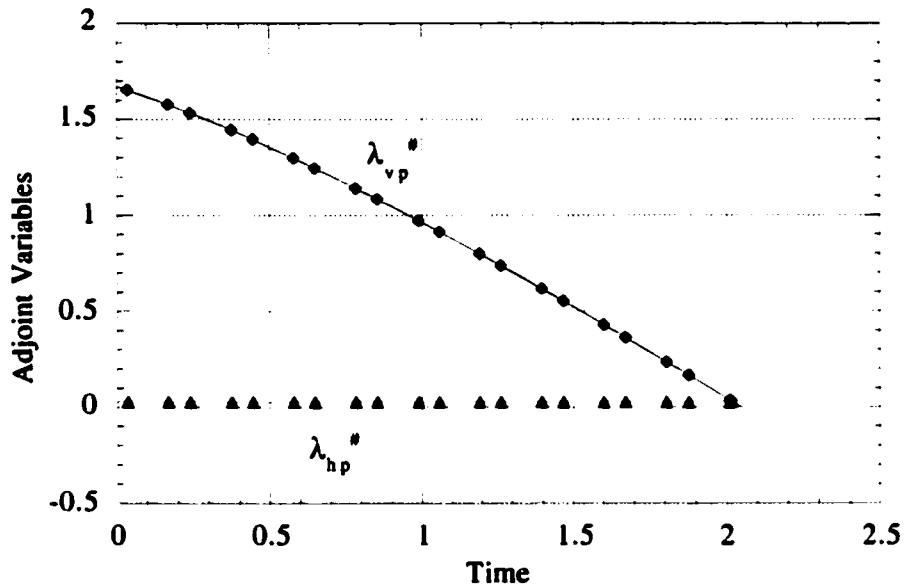


Figure 5.3 Histories of Adjoint Variables of Saddle-Point Trajectories for Two-Dimensional Simplified Air Combat (Nominal Case)

Saddle-point trajectories for the simplified two-dimensional air combat problem are provided in Fig. 5.4. Also, the histories of the flight path angles are shown in Fig. 5.5. These trajectories show that the best maneuver for the simplified two-dimensional air combat is a dive maneuver.

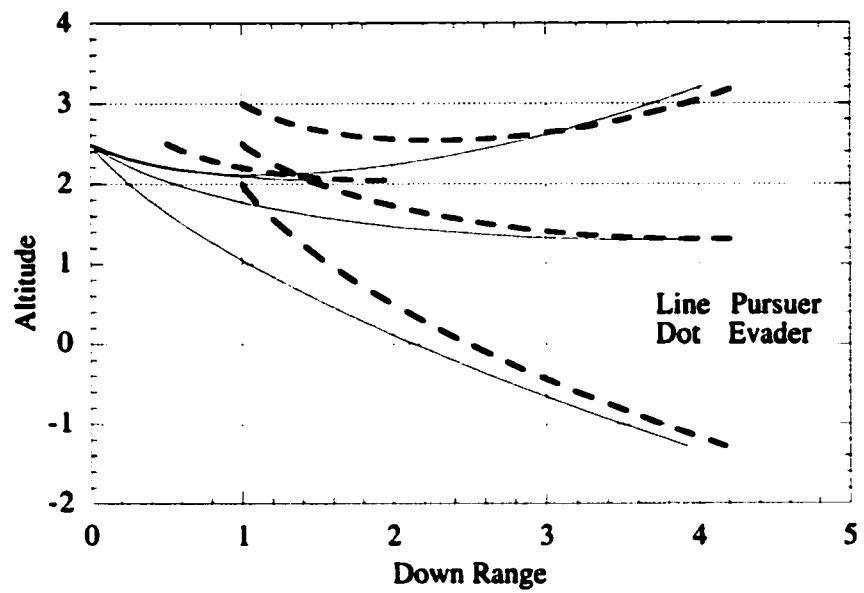


Figure 5.4 Saddle-Point Trajectories for Simplified Two-Dimensional Air Combat

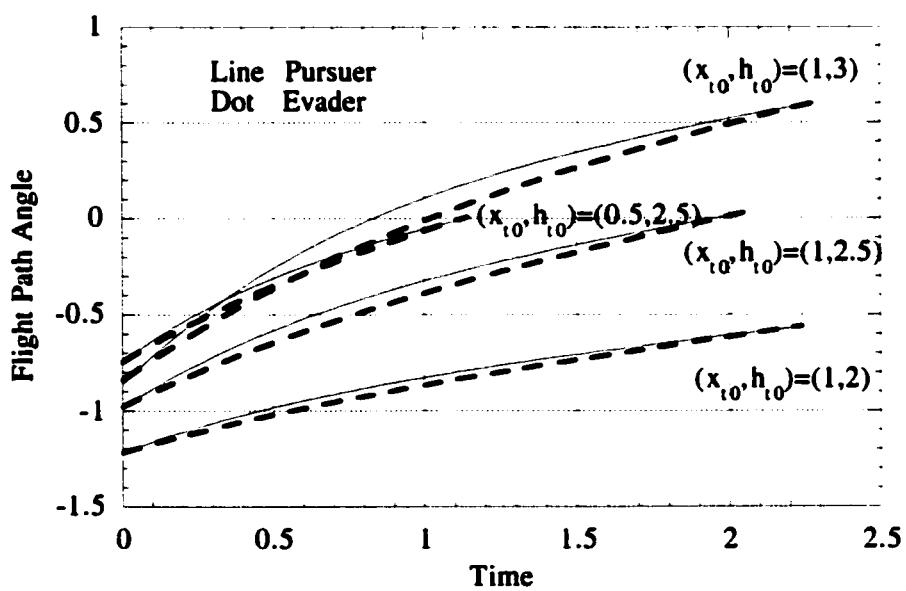


Figure 5.5 Histories of Flight Path Angle for Simplified Two-Dimensional Air Combat

5.2.2 Three-Dimensional Problem

Air combat in the real world involves three-dimensional maneuvering. In this subsection, the simplified problem is extended from two-dimensional space to three-dimensional space. We keep many of the assumptions of Sec. 5.2.1, however the heading angle and the bank angle must be treated differently. A new assumption applied is that the aircraft can take arbitrary bank angle, have normal force as required and be back to level flight instantly. This means the heading angle is added as a control variable and then (5.3) is eliminated from the set of equations of motion. With these new assumptions the set of equations of motion is:

$$\frac{dv_i}{dt} = \frac{1}{m} (T - \frac{1}{2} \rho v_i^2 S C_{D_0} - \frac{2km^2 g^2}{\rho v_i^2 S} \cos^2 \gamma_i) - g \sin \gamma_i \quad (5.37)$$

$$\frac{dx_i}{dt} = v_i \cos \gamma_i \cos \psi_i \quad (5.38)$$

$$\frac{dy_i}{dt} = v_i \cos \gamma_i \sin \psi_i \quad (5.39)$$

$$\frac{dh_i}{dt} = v_i \sin \gamma_i \quad (5.40)$$

All of the aircraft parameters required for (5.37) - (5.40) are set to the same values as they were in Sec. 5.2.1.

The terminal condition in the problem is interception, i.e. the position of the pursuer is the same as that of the evader at the terminal time:

$$x_e = x_p \quad (5.41)$$

$$y_e = y_p \quad (5.42)$$

$$h_e = h_p \quad (5.43)$$

The problem is characterized as a two-sided flight path optimization problem:

$$V = \min_{\gamma_p, \psi_p} \max_{\gamma_e, \psi_e} t_f \quad (5.44)$$

subject to (5.37) - (5.43) for given initial conditions.

The semi-DCNLP formulation requires the adjoint equations and the Pontryagin principle for one player, in this case the pursuer. The Hamiltonian of the problem is:

$$\begin{aligned} H = & \lambda_{vp} \left\{ \frac{1}{m} \left(T_p - \frac{1}{2} \rho v_p^2 S C_{D_0} - \frac{2km^2 g^2}{\rho v_p^2 S} \cos^2 \gamma_p \right) - g \sin \gamma_p \right\} \\ & + \lambda_{xp} v_p \cos \gamma_p \cos \psi_p + \lambda_{yp} v_p \cos \gamma_p \sin \psi_p + \lambda_{hp} v_p \sin \gamma_p + H_e \end{aligned} \quad (5.45)$$

Differential game theory provides adjoint equations and the Pontryagin principle as:

$$\begin{aligned} \frac{d\lambda_{vp}}{dt} = & - \frac{\partial H}{\partial v_p} = \frac{\lambda_{vp}}{m} \left(\rho v_p S C_{D_0} - \frac{4km^2 g^2}{\rho v_p^3 S} \cos^2 \gamma_p \right) \\ & - \lambda_{xp} \cos \gamma_p \cos \psi_p - \lambda_{yp} \cos \gamma_p \sin \psi_p - \lambda_{hp} \sin \gamma_p \end{aligned} \quad (5.46)$$

$$\frac{d\lambda_{xp}}{dt} = - \frac{\partial H}{\partial x_p} = 0 \quad (5.47)$$

$$\frac{d\lambda_{yp}}{dt} = - \frac{\partial H}{\partial y_p} = 0 \quad (5.48)$$

$$\frac{d\lambda_{hp}}{dt} = - \frac{\partial H}{\partial h_p} = 0 \quad (5.49)$$

$$\begin{aligned}\frac{\partial H}{\partial \gamma_p} &= \lambda_{vp} \left\{ \frac{2km^2g^2}{\rho v_p^2 S} \sin(2\gamma_p) - g \cos \gamma_p \right\} - \lambda_{xp} v_p \sin \gamma_p \cos \psi_p \\ &\quad - \lambda_{yp} v_p \sin \gamma_p \sin \psi_p + \lambda_{hp} v_p \cos \gamma_p = 0\end{aligned}\quad (5.50)$$

$$\frac{\partial H}{\partial \psi_p} = -\lambda_{xp} v_p \cos \gamma_p \sin \psi_p + \lambda_{yp} v_p \cos \gamma_p \cos \psi_p = 0 \quad (5.51)$$

The terminal conditions of the adjoint variables are:

$$\Phi = t_f + v_1(x_e - x_p) + v_2(y_e - y_p) + v_3(h_e - h_p) \quad (5.52)$$

$$\lambda_{vp}(t_f) = \frac{\partial \Phi}{\partial v_p(t_f)} = 0 \quad (5.53)$$

$$\lambda_{xp}(t_f) = \frac{\partial \Phi}{\partial x_p(t_f)} = -v_1 \quad (5.54)$$

$$\lambda_{yp}(t_f) = \frac{\partial \Phi}{\partial y_p(t_f)} = -v_2 \quad (5.55)$$

$$\lambda_{hp}(t_f) = \frac{\partial \Phi}{\partial h_p(t_f)} = -v_3 \quad (5.56)$$

Furthermore, the technique applied in Sec. 5.2.1 to reduce the set of the equations, dividing the other adjoint variables by the constant λ_{vp} , is applied. The set of equations for the optimal control for the pursuer become:

$$\begin{aligned}\frac{d\lambda_{vp}''}{dt} &= \frac{\lambda_{vp}''}{m} (\rho v_p S C_{D_0} - \frac{4km^2g^2}{\rho v_p^3 S} \cos^2 \gamma_p) \\ &\quad - \cos \gamma_p \cos \psi_p - \lambda_{yp}'' \cos \gamma_p \sin \psi_p - \lambda_{hp}'' \sin \gamma_p\end{aligned}\quad (5.57)$$

$$\frac{d\lambda_{yp}''}{dt} = 0 \quad (5.58)$$

$$\frac{d\lambda_{hp}''}{dt} = 0 \quad (5.59)$$

$$\begin{aligned}\lambda_{vp}'' \left\{ \frac{2kmg^2}{\rho v_p^2 S} \sin(2\gamma_p) - g \cos \gamma_p \right\} - v_p \sin \gamma_p \cos \psi_p \\ - \lambda_{yp}'' v_p \sin \gamma_p \sin \psi_p + \lambda_{hp}'' v_p \cos \gamma_p = 0\end{aligned}\quad (5.60)$$

$$-\cos \gamma_p \sin \psi_p + \lambda_{yp}'' \cos \gamma_p \cos \psi_p = 0 \quad (5.61)$$

$$\lambda_{vp}''(t_f) = 0 \quad (5.62)$$

Finally the semi-DCNLP formulation becomes:

$$J = \max_{\gamma_e, \psi_e} t_f \quad (5.63)$$

subject to (5.37) - (5.43), (5.57) - (5.62) for given initial conditions.

Figures 5.6 - 5.11 show solutions of the problem with the same initial separation and altitude, but different line of sight (LOS); i.e., separation is 1, altitude is 2.5 and LOS is either 30°, 45° or 60°. Also, both pursuer and evader are flying along the x axis (heading angle is zero) at the initial time. For the initial heading angle of the fighters. Fig. 5.6 shows how both evader and pursuer change their heading angle instantaneously. Also, from Fig. 5.6 and Fig. 5.9, it is obvious that the trajectories are fundamentally in two-dimensional space. The instantaneous change of the heading angle and the two-dimensional vertical maneuvering provide identical flight path angle histories in Fig. 5.10 in spite of the difference in initial horizontal position and LOS. This suggests that the altitude and velocity histories are also identical.

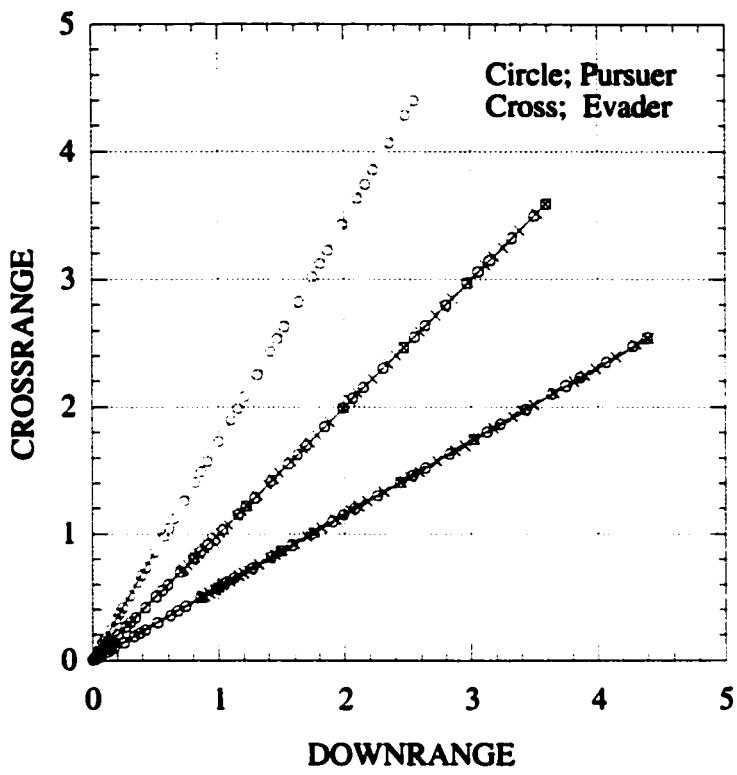


Figure 5.6 Horizontal Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Relative Positions

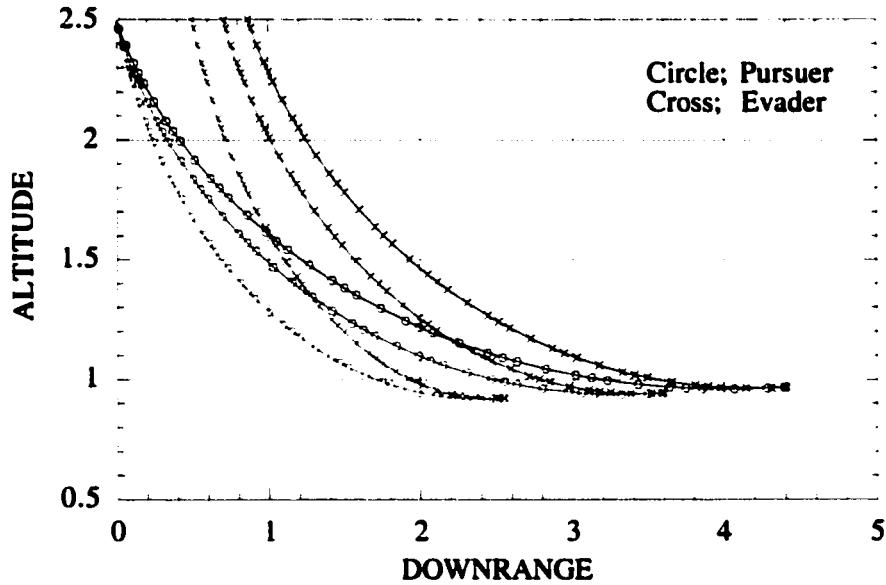


Figure 5.7 Vertical (X-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Relative Positions

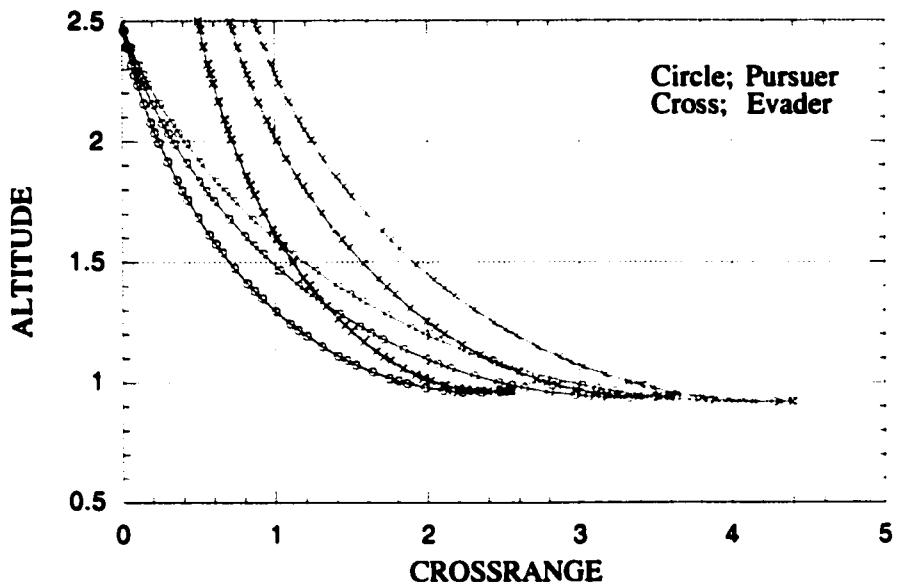


Figure 5.8 Vertical (Y-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Relative Positions

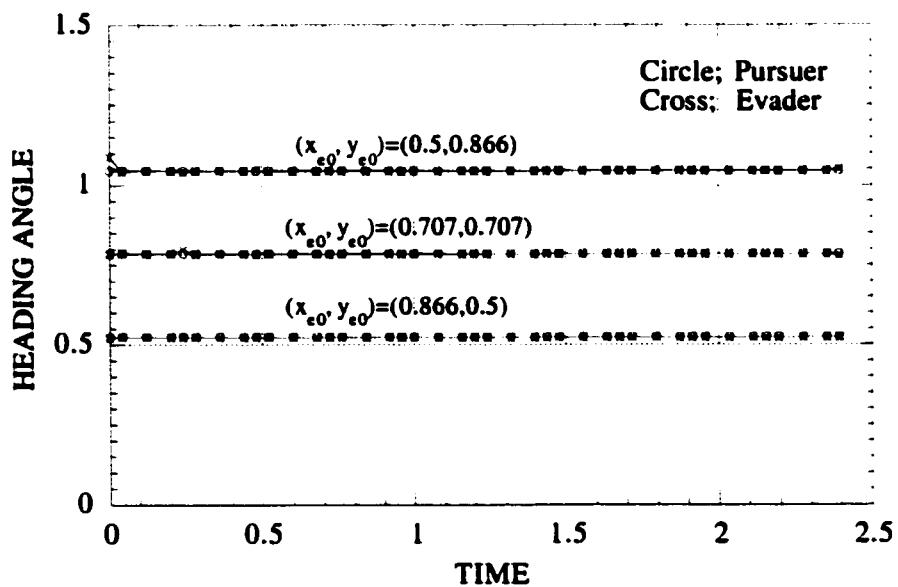
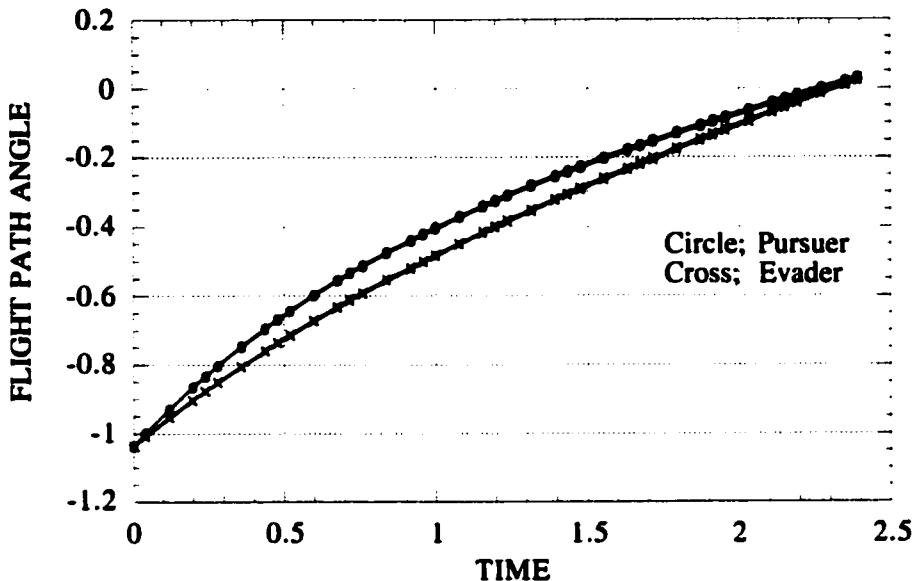


Figure 5.9 Heading Angle Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Relative Positions



(Trajectories for all 3 initial conditions are shown, but are indistinguishable)

Figure 5.10 Flight Path Angle Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Relative Positions

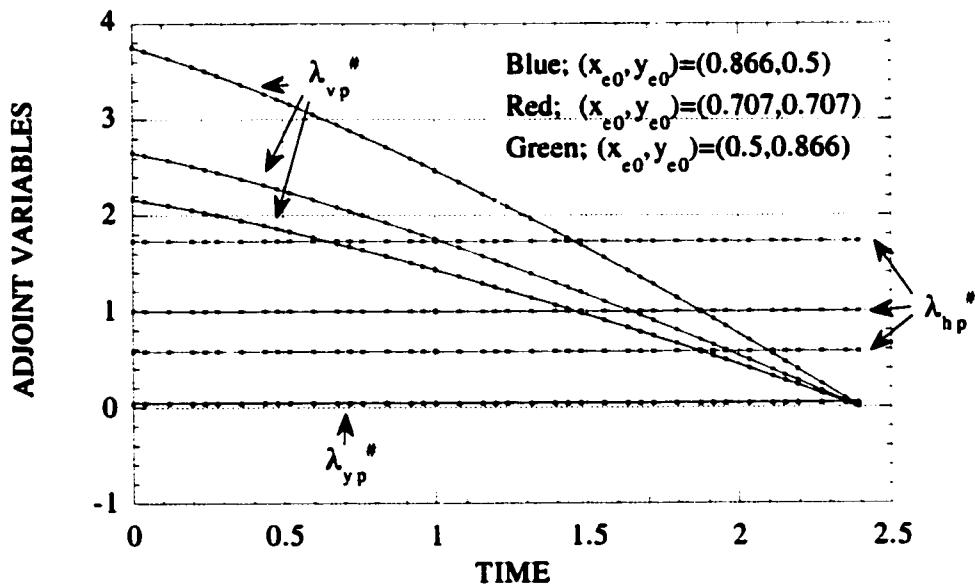


Figure 5.11 Adjoint Variable Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Relative Positions

Figures 5.12 - 5.17 show aspects of the solution of the problem with the same initial separation and LOS, but different altitude. The same initial conditions in the vertical-plane as used in Sec. 5.2.1 are used here. Figures 5.12 - 5.17 are very similar to the corresponding results from Sec. 5.2.1, although the difference in the terminal condition brings slightly different trajectories; i.e., the terminal condition is interception in this problem whereas in Sec. 5.2.1 the aircraft have a small separation at the terminal time.

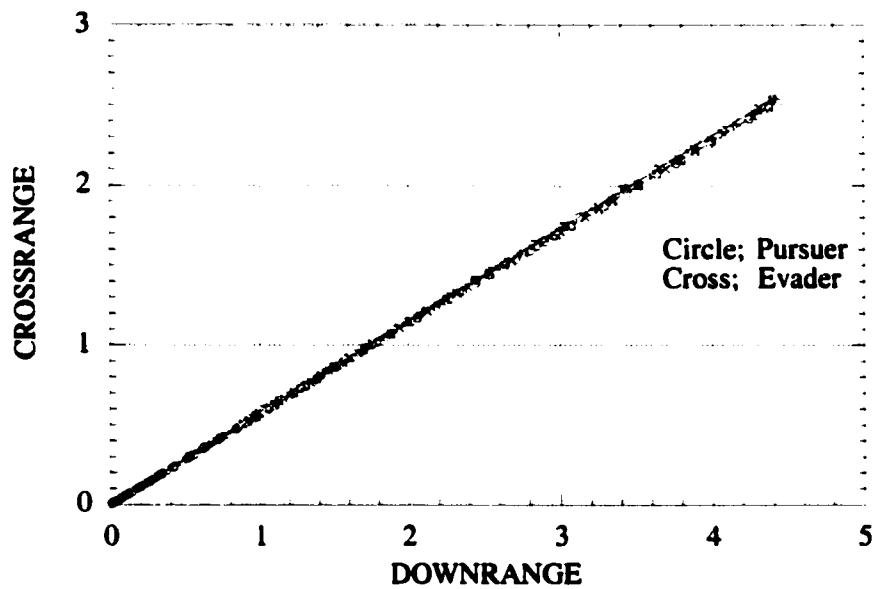


Figure 5.12 Horizontal Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Altitudes

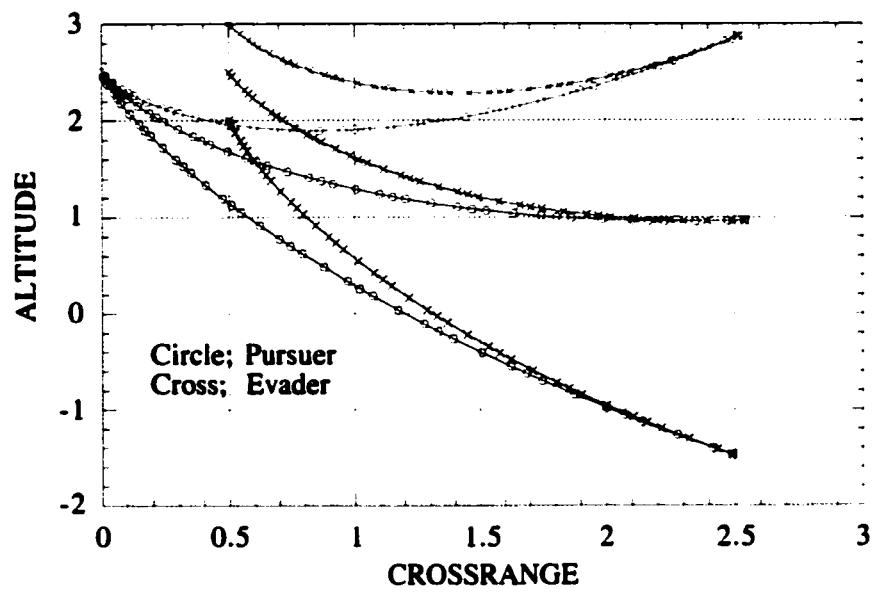


Figure 5.13 Vertical (X-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Altitudes

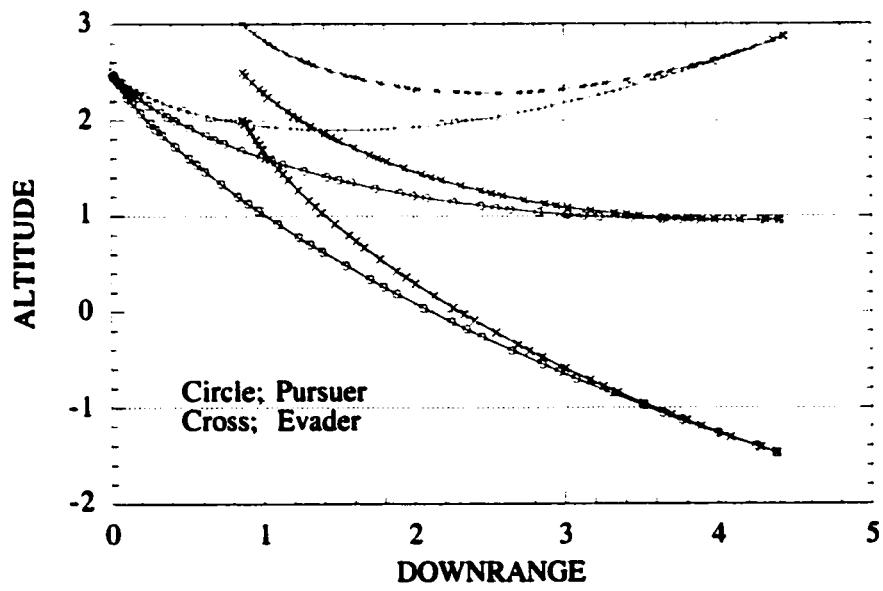
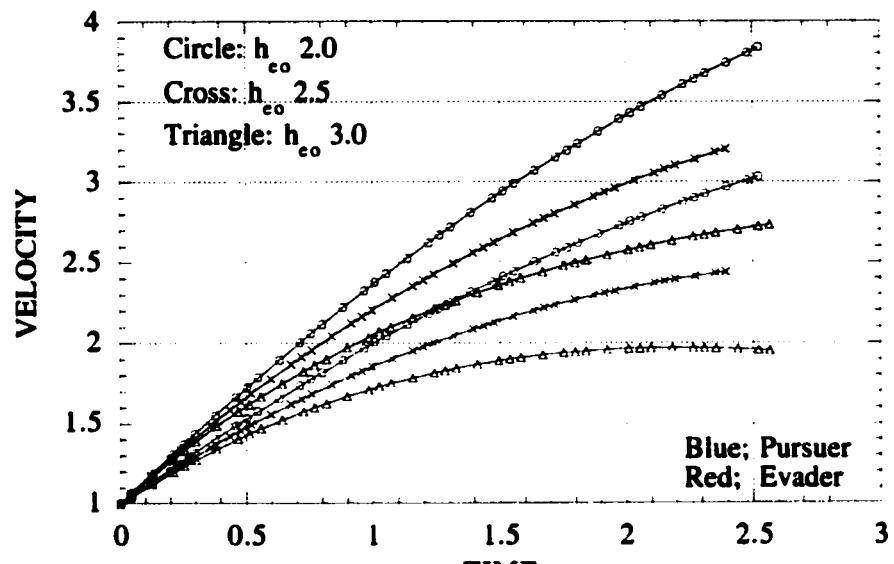


Figure 5.14 Vertical (Y-Z) Saddle-Point Trajectories for Simplified Three-Dimensional Air Combat for Various Initial Altitudes



(Pursuer always begins from an altitude of 2.5 units)

Figure 5.15 Flight Path Angle Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Altitudes

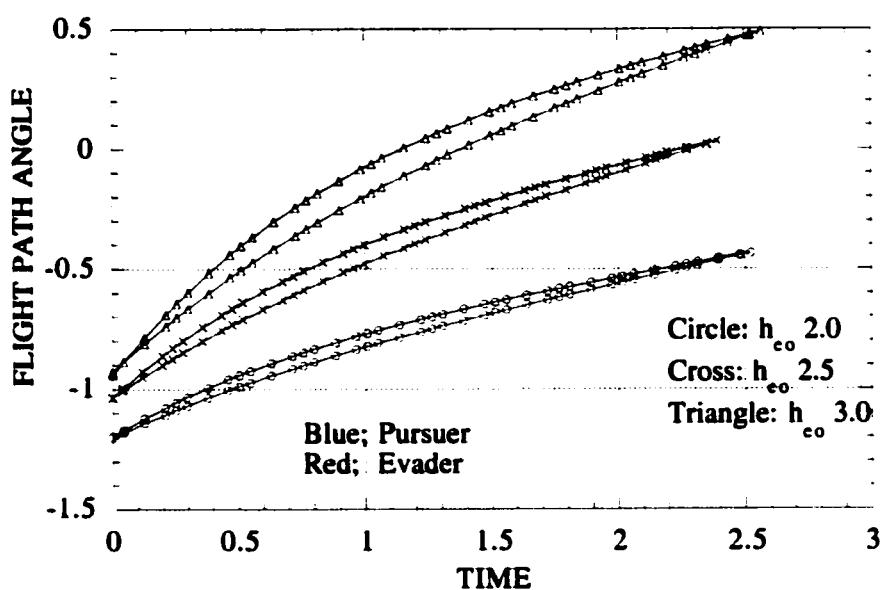


Figure 5.16 Velocity Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Altitudes

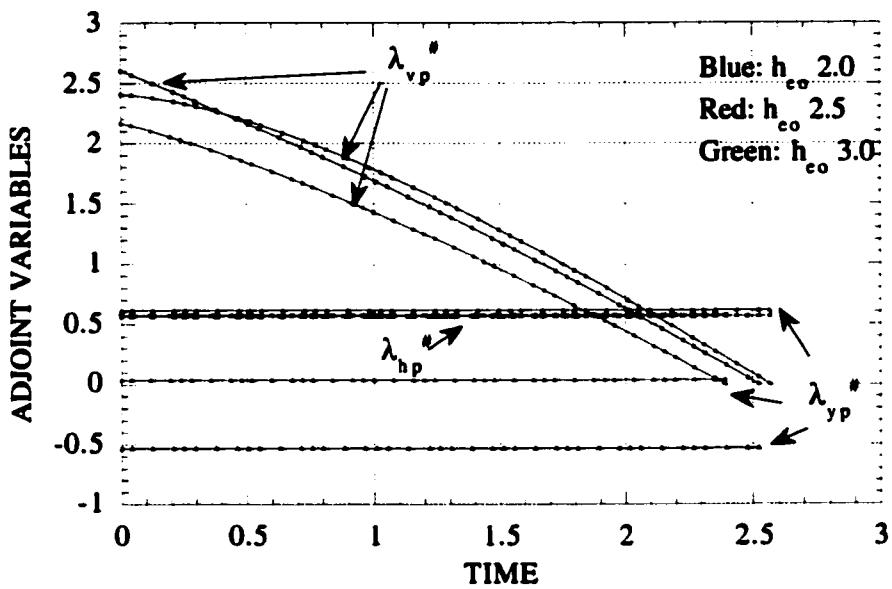


Figure 5.17 Adjoint Variable Histories for Saddle-Point Trajectories of Simplified Three-Dimensional Air Combat for Various Initial Altitudes

In the simplified three-dimensional air combat problem, dive maneuvers and initial instantaneous heading change are observed. This says the obtained maneuvers are qualitatively the same as those in the two-dimensional case except for the initial heading change, i.e the optimal maneuvers consist of two-phases, instantaneous horizontal maneuvering followed by continuous vertical dive maneuvering.

5.3 Realistic Air Combat

5.3.1 Two-Dimensional Problem

This section considers the solution of the problem described in Sec. 5.1. As the final step for that, in this sub-section the only assumption introduced is that maneuvers of the aircraft lie on a vertical plane; i.e. the heading angle and the bank angle of both aircraft are equal to zero. Applying the assumption makes eqns. (5.3) and (5.5) unnecessary. Then, the set of equations of motion (5.1) - (5.6) is re-written as follows:

$$\frac{dv_i}{dt} = \frac{l}{m}(T_i \cos\alpha_i - D_i) - g \sin\gamma_i \quad (5.64)$$

$$\frac{d\gamma_i}{dt} = \frac{l}{mv_i}(T_i \sin\alpha_i + L_i) - \frac{g}{v_i} \cos\gamma_i \quad (5.65)$$

$$\frac{dx_i}{dt} = v_i \cos\gamma_i \quad (5.66)$$

$$\frac{dh_i}{dt} = v_i \sin\gamma_i \quad (5.67)$$

Although the aerodynamic coefficients are expressed in (5.9) and (5.10), range of the angle of attack of the pursuer and the evader are set as follows to allow high angle of attack flight ability, especially for the pursuer:

$$0 \leq \alpha_p \leq \pi/2 \quad (5.68)$$

$$0 \leq \alpha_e \leq \pi/6 \quad (5.69)$$

The atmospheric density is approximated [26] using the same model as in Sec. 2.3:

$$\rho(h_i) = \rho_s \left(1 - 0.00688 \left(\frac{h_i}{1000} \right) \right)^{4.256} \quad (5.70)$$

Other parameters required to solve (5.64) - (5.67) are set to the same values they had in Sec. 5.2.2.

The same terminal condition as employed in Sec. 5.2.1 is applied; the pursuer is to approach to 0.25 non-dimensional range units behind the evader and be at the same altitude. Then,

$$x_e - x_p = r_{\text{shoot}} \quad (5.71)$$

$$h_e - h_p = 0 \quad (5.72)$$

The problem is then to find:

$$V = \min_{\alpha_p} \max_{\alpha_e} t_f \quad (5.73)$$

subject to (5.64)-(5.67), for given initial conditions and for terminal conditions (5.71) and (5.72).

To use the semi-DCNLP solution method we first require the system Hamiltonian:

$$\begin{aligned} H = & \lambda_{vp} \left\{ \frac{1}{m} (T_p \cos \alpha_p - \frac{1}{2} \rho(h_p) v_p^2 S_{CD}(\alpha_p)) - g \sin \gamma_p \right\} \\ & + \lambda_{yp} \left\{ \frac{1}{mv_p} (T_p \sin \alpha_p + \frac{1}{2} \rho(h_p) v_p^2 S_{CL}(\alpha_p)) - \frac{g \cos \gamma_p}{v_p} \right\} \\ & + \lambda_{xp} v_p \cos \gamma_p + \lambda_{hp} v_p \sin \gamma_p + H_e \end{aligned} \quad (5.74)$$

The system adjoint equations are:

$$\begin{aligned}\frac{d\lambda_{vp}}{dt} = -\frac{\partial H}{\partial v_p} &= \frac{\lambda_{vp} \rho(h_p) v_p^2 SC_D(\alpha_p)}{m} \\ &+ \lambda_{\gamma p} \left\{ \frac{1}{m} \left(\frac{T_p \sin \alpha_p}{v_p^2} - \frac{1}{2} \rho(h_p) SC_L(\alpha_p) \right) - \frac{g \cos \gamma_p}{v_p^2} \right\} \\ &- \lambda_{xp} \cos \gamma_p - \lambda_{hp} \sin \gamma_p\end{aligned}\quad (5.75)$$

$$\frac{d\lambda_{\gamma p}}{dt} = -\frac{\partial H}{\partial \gamma_p} = \lambda_{vp} g \cos \gamma_p - \frac{\lambda_{\gamma p} g \sin \gamma_p}{v_p} + \lambda_{xp} v_p \sin \gamma_p - \lambda_{hp} v_p \cos \gamma_p \quad (5.76)$$

$$\frac{d\lambda_{xp}}{dt} = -\frac{\partial H}{\partial x_p} = 0 \quad (5.77)$$

$$\frac{d\lambda_{hp}}{dt} = -\frac{\partial H}{\partial h_p} = 0 \quad (5.78)$$

The optimal angle of attack for the pursuer is found analytically through the necessary condition:

$$\alpha_p = \arg \min_{\alpha_p} H \quad (5.79)$$

Eqn. (5.79) cannot be expressed as an algebraic equation if the optimal angle of attack for the pursuer is at a boundary. It is easier to obtain the optimal control for an unbounded control variable because only stationary points are satisfied with the first-order optimality condition. Thus, the angle of attack for the pursuer is transformed into a new, unbounded variable, τ_{ap} , using the following relationship:

$$\alpha_p = \frac{\pi}{4} (1 - \cos \tau_{ap}) \quad (5.80)$$

Using this variable, the optimality condition becomes:

$$\begin{aligned}\frac{\partial H}{\partial \tau_{ap}} &= \frac{\partial H}{\partial \alpha_p} \frac{\partial \alpha_p}{\partial \tau_{ap}} = \frac{\pi}{4} \sin \tau_{ap} \left\{ \frac{\lambda_{vp}}{m} (-T_p \sin \alpha_p - \frac{1}{2} \rho(h_p) v_p^2 S \frac{dC_D(\alpha_p)}{d\alpha_p}) \right. \\ &\quad \left. + \frac{\lambda_{rp}}{mv_p} (T_p \cos \alpha_p + \frac{1}{2} \rho(h_p) v_p^2 S \frac{dC_L(\alpha_p)}{d\alpha_p}) \right\} = 0\end{aligned}\quad (5.81)$$

The terminal conditions of the adjoint variables are derived as follows:

$$\Phi = t_f + v_1(x_e - x_p - r_{shoot}) + v_2(h_e - h_p) \quad (5.82)$$

$$\lambda_{vp}(t_f) = \frac{\partial \Phi}{\partial v_p(t_f)} = 0 \quad (5.83)$$

$$\lambda_{rp}(t_f) = \frac{\partial \Phi}{\partial v_p(t_f)} = 0 \quad (5.84)$$

$$\lambda_{vp}(t_f) = \frac{\partial \Phi}{\partial x_p(t_f)} = -v_1 \quad (5.85)$$

$$\lambda_{hp}(t_f) = \frac{\partial \Phi}{\partial h_p(t_f)} = -v_2 \quad (5.86)$$

Eqn. (5.85) and (5.86) include unknown variables v_1 and v_2 . Then, only (5.83) and (5.84) are used as boundary conditions.

Furthermore, the same technique used in Sec. 5.2.1 to reduce the number of the equations, i.e. having the constant λ_{vp} divide the other adjoint variables, is applied.

Finally, the set of equations for determining the optimal control for the pursuer are:

$$\begin{aligned}\frac{d\lambda_{vp}''}{dt} &= \frac{\lambda_{vp}'' \rho(h_p) v_p^2 S C_D(\alpha_p)}{m} + \lambda_{rp}'' \left\{ \frac{1}{m} \left(\frac{T_p \sin \alpha_p}{v_p^2} - \frac{1}{2} \rho(h_p) S C_L(\alpha_p) \right) \right. \\ &\quad \left. - \frac{g \cos \gamma_p}{v_p^2} \right\} - \cos \gamma_p - \lambda_{hp}'' \sin \gamma_p\end{aligned}\quad (5.87)$$

$$\frac{d\lambda_{vp}''}{dt} = \lambda_{vp}'' g \cos \gamma_p - \frac{\lambda_{vp}'' g \sin \gamma_p}{v_p} + v_p \sin \gamma_p - \lambda_{hp}'' v_p \cos \gamma_p \quad (5.88)$$

$$\frac{d\lambda_{hp}''}{dt} = 0 \quad (5.89)$$

$$\begin{aligned} \sin \tau_{ap} \left\{ \frac{\lambda_{vp}''}{m} \left(-T_p \sin \alpha_p - \frac{1}{2} \rho(h_p) v_p^2 S \frac{dC_D(\alpha_p)}{d\alpha_p} \right) \right. \\ \left. + \frac{\lambda_{vp}''}{mv_p} \left(T_p \cos \alpha_p + \frac{1}{2} \rho(h_p) v_p^2 S \frac{dC_L(\alpha_p)}{d\alpha_e} \right) \right\} = 0 \end{aligned} \quad (5.90)$$

$$\lambda_{vp}''(t_f) = 0 \quad (5.91)$$

$$\lambda_{hp}''(t_f) = 0 \quad (5.92)$$

The semi-DCNLP formulation for this problem becomes:

$$V = \max_{\gamma_e} t_f \quad (5.93)$$

subject to (5.64) - (5.69), and (5.87) - (5.92) for given initial conditions.

The simple GA is applied to find an initial guess of the solution for the semi-DCNLP problem. A fitness function for this problem is:

$$J_{fit} = -100 \{ (x_e - x_p - r_{shoot})^2 + (h_e - h_p)^2 + \lambda_{vp}^{+2} + \lambda_{vp}^{-2} \} |_{t=t_f} \quad (5.94)$$

Also, the properties of the genetic algorithm are provided as n=1000, l=40, p_c=0.7 and p_m=0.001. Unknown parameters are t_f, λ_{vp}⁺(0), λ_{vp}⁻(0) and λ_{vp}^{''}(0). All parameters are decoded from a string of 10 digits each. The control variables for the pursuer are fixed because of the simplification of the problem.

A simple genetic algorithm program provides a convergent solution for the initial condition, $v_{p,0}=v_{e,0}=1.0$, $\gamma_{p,0}=\gamma_{e,0}=0$, $h_{p,0}=h_{e,0}=2.5$, $x_{p,0}=0$ and $x_{e,0}=1.0$.

After getting an initial guess for the solution of the problem using GA, the nonlinear programming problem solver used by the semi-DCNLP formulation is able to converge to a solution. The problem is solved for initial altitudes of the evader of 2.0 unit and 3.0 unit in addition to the original value of 2.5 unit.

Trajectories obtained numerically are shown in Figures 5.18 to 5.22. Figure 5.18 shows the histories of the adjoint variables, with the corresponding variables obtained from NZSOL, to verify that the obtained trajectories are the saddle-point trajectories. Thus, the semi-DCNLP method provides the solution for the realistic air combat problem.

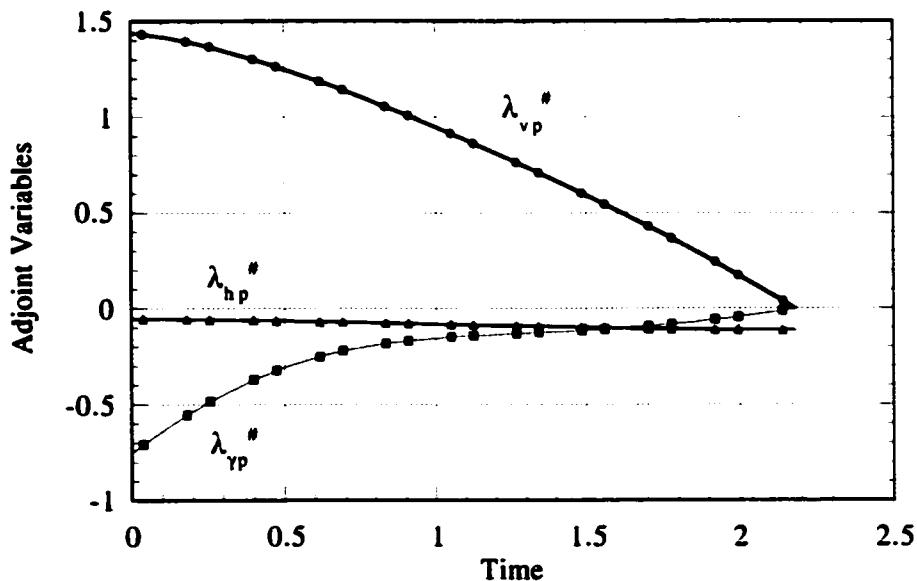


Figure 5.18 Adjoint Variable Histories for Saddle-Point Trajectories of Realistic Two-Dimensional Air Combat

Figures 5.19, 5.20 and 5.21 show the saddle-point trajectories, the histories of the velocities and the flight path angle of the pursuer and evader. It is observed from these figures that both evader and pursuer increase dive angle initially, then decrease the dive angle. Although the simplified air combat problem also yields a dive maneuver as the optimal maneuver, the optimal maneuvering in the simplified case is qualitatively different. The diving angle in the simplified case is a maximum at the initial time and then shallows gradually (Figures 5.4 and 5.5). However, this phenomena results from the simplified system, i.e. from using the flight path angle as a control variable rather than a state variable.

Figure 5.22 shows the histories of angle of attack for both the pursuer and evader. In the figure, it is found that the angles of attack are properly constrained at zero for the early phase of the maneuvering. This is because the control variables are required to take a value that increases the dive angle, i.e., zero AOA. This also validates that the control variable is properly constrained despite transforming the constrained control variable to an unconstraint one, using (5.80).

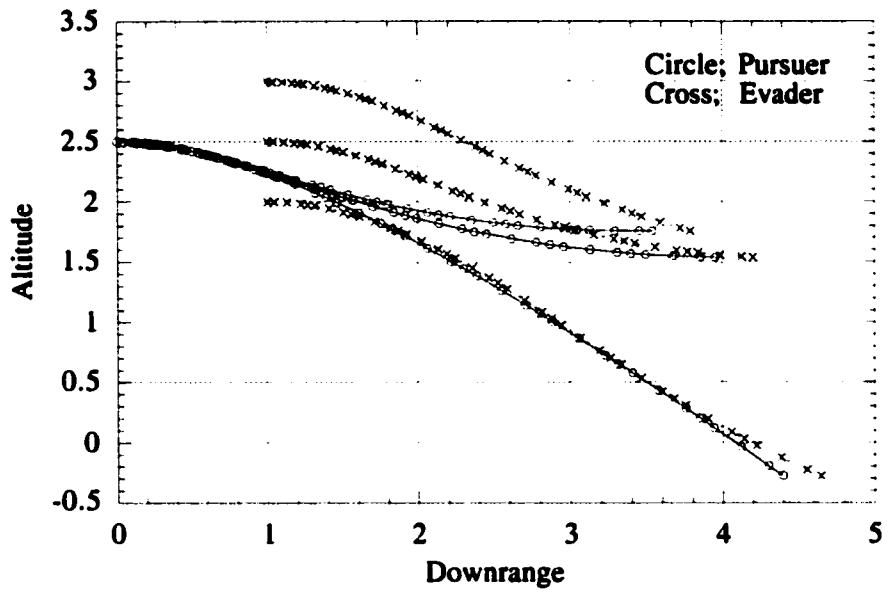


Figure 5.19 Saddle-Point Trajectories for Realistic Two-Dimensional Air Combat

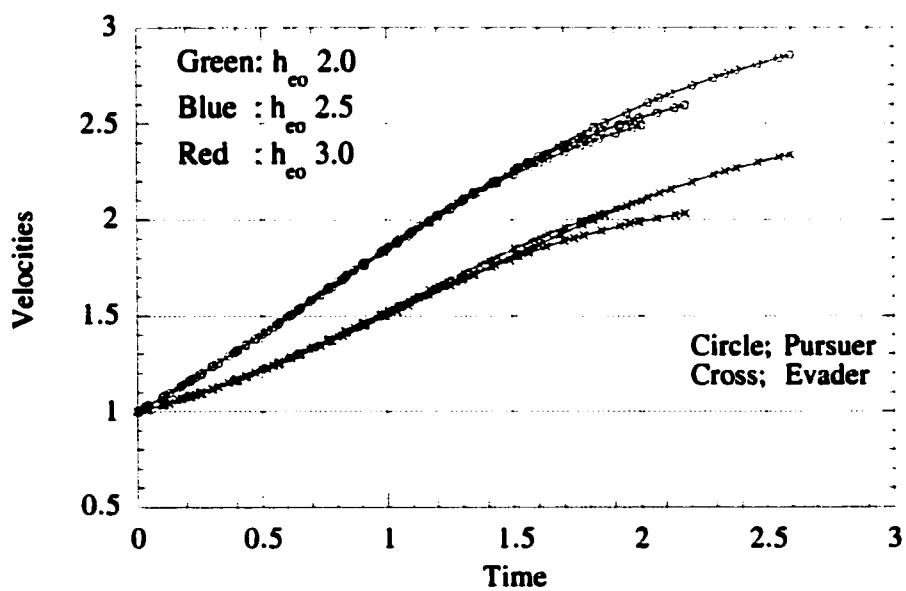
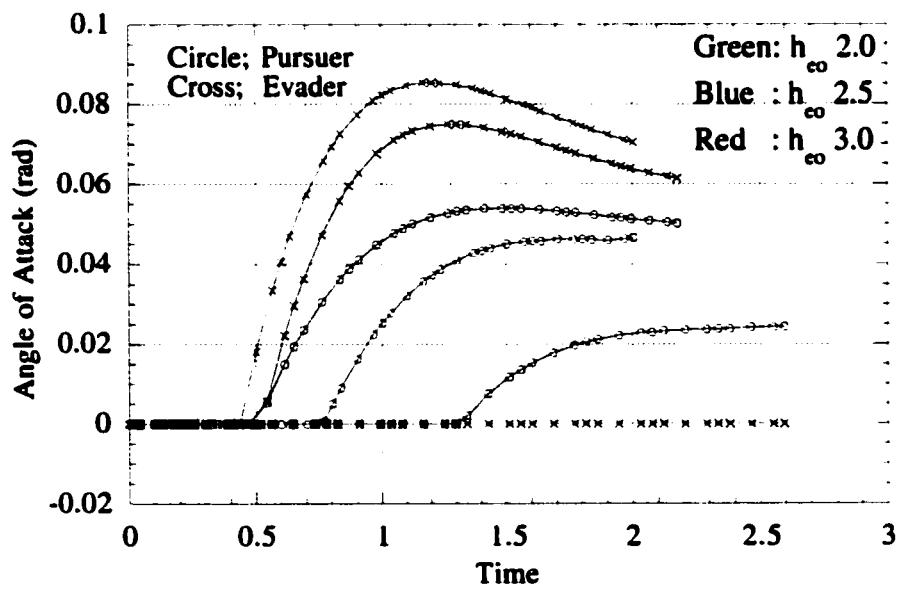
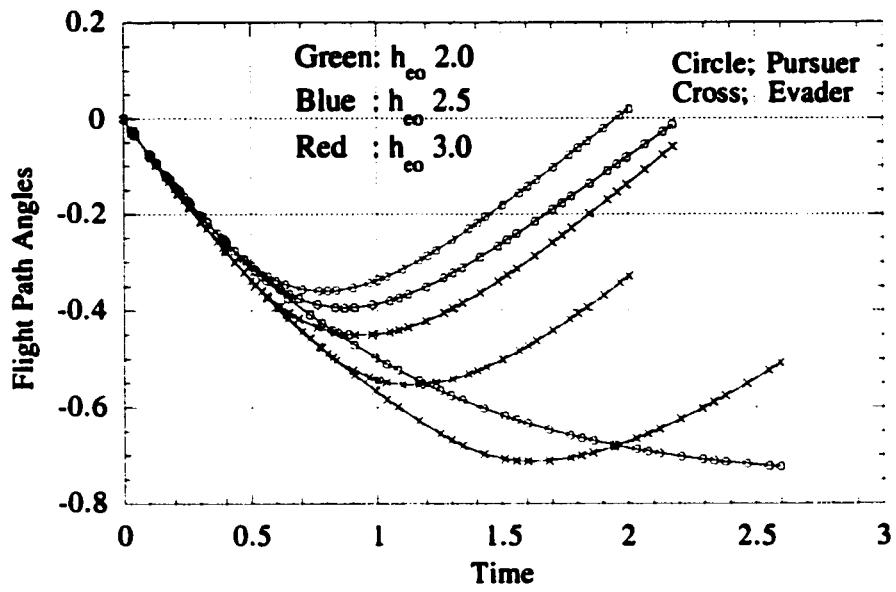


Figure 5.20 Velocity Histories for Saddle-Point Trajectories of Realistic Two-Dimensional Air Combat



5.3.2 Three-Dimensional Problem

In this sub-section, the realistic three-dimensional air combat problem described in Sec. 5.1 is solved in a complete form. In addition, effects of changes in (relative) initial position and (relative) initial motion are evaluated to characterize the optimal air combat.

The problem is restated as follows:

- (1) The set of equations of motion is (5.1) - (5.6) with (5.7) and (5.8),
- (2) The aerodynamic model is (5.9), (5.10), (5.68) and (5.69),
- (3) The atmospheric density model is (5.70).

Other parameters required to solve the problem are specified as they were in Sec. 5.3.1, except for the thrust of the aircraft. The thrust is set on the basis of the actual F-16A aircraft at 0.5 Mach and 20,000 ft altitude, with afterburner for the pursuer, yielding $T_p=12,274\text{lb}$, and without afterburner for the evader, yielding $T_e=6,124\text{lb}$ [37]. Of course the real engine has performance that cannot be represented by a simple function of Mach number and altitude. Therefore, the fixed values above are applied to simplify the solution of the numerical optimization.

The terminal condition is appropriate to the “tail-chase” situation and is modeled as an interception condition with the same heading angle:

$$x_p(t_f) = x_e(t_f) \quad (5.95)$$

$$y_p(t_f) = y_e(t_f) \quad (5.96)$$

$$h_p(t_f) = h_e(t_f) \quad (5.97)$$

$$\psi_p(t_f) = \psi_e(t_f) \quad (5.98)$$

The problem is characterized as a two-sided flight path optimization problem as:

$$V = \min_{\alpha_p, \phi_p} \max_{\alpha_e, \phi_e} t_f \quad (5.99)$$

subject to (5.1) - (5.6) and (5.95) - (5.98) for given initial conditions.

The semi-DCNLP method is applied to solve the problem. Therefore, the adjoint equations and the optimality conditions for the superior fighter aircraft (the pursuer) are required. The Hamiltonian for the problem is:

$$\begin{aligned} H = & \frac{\lambda_{v_p}}{m_p} \left\{ (T_p \cos \alpha_p - D_p) - m_p g \sin \gamma_p \right\} \\ & + \frac{\lambda_{\gamma_p}}{m_p v_p} \left\{ (T_p \sin \alpha_p + L_p) \cos \phi_p - m_p g \cos \gamma_p \right\} \\ & + \frac{\lambda_{\psi_p}}{m_p v_p \cos \gamma_p} (T_p \sin \alpha_p + L_p) \sin \phi_p + \lambda_{x_p} v_p \cos \gamma_p \cos \psi_p \\ & + \lambda_{y_p} v_p \cos \gamma_p \sin \psi_p + \lambda_{h_p} v_p \sin \gamma_p + H_e \end{aligned} \quad (5.100)$$

where H_e means a part of the Hamiltonian which is described for the evader.

By applying the Pontryagin principle, the following adjoint equations are obtained:

$$\begin{aligned} \frac{d\lambda_{v_p}}{dt} = & \frac{2\lambda_{v_p} D_p}{m_p v_p} + \frac{\lambda_{\gamma_p}}{m_p v_p^2} \left\{ (T_p \sin \alpha_p - L_p) \cos \phi_p - m_p g \cos \gamma_p \right\} \\ & + \frac{\lambda_{\psi_p}}{m_p v_p^2 \cos \gamma_p} (T_p \sin \alpha_p - L_p) \sin \phi_p - \lambda_{x_p} \cos \gamma_p \cos \psi_p \\ & - \lambda_{y_p} \cos \gamma_p \sin \psi_p - \lambda_{h_p} \sin \gamma_p \end{aligned} \quad (5.101)$$

$$\frac{d\lambda_{y_p}}{dt} = \lambda_{v_p} g \cos \gamma_p - \frac{\lambda_{y_p} g \sin \gamma_p}{v_p} - \frac{\lambda_{\psi_p} \tan \gamma_p}{m_p v_p \cos \gamma_p} (T_p \sin \alpha_p + L_p) \sin \phi_p \\ + \lambda_{x_p} v_p \sin \gamma_p \cos \psi_p + \lambda_{y_p} v_p \sin \gamma_p \sin \psi_p - \lambda_{h_p} v_p \cos \gamma_p$$
(5.102)

$$\frac{d\lambda_{\psi_p}}{dt} = \lambda_{x_p} v_p \cos \gamma_p \sin \psi_p - \lambda_{y_p} v_p \cos \gamma_p \cos \psi_p$$
(5.103)

$$\frac{d\lambda_{x_p}}{dt} = 0$$
(5.104)

$$\frac{d\lambda_{y_p}}{dt} = 0$$
(5.105)

$$\frac{d\lambda_{h_p}}{dt} = \frac{1}{m_p \rho(h_p)} \left\{ \lambda_{v_p} D_p - \frac{\lambda_{y_p} L_p \cos \phi_p}{v_p} - \frac{\lambda_{\psi_p} L_p \sin \phi_p}{v_p \cos \gamma_p} \right\} \frac{dp}{dh_p}$$
(5.106)

Since λ_{vp} is constant, from (5.104), the adjoint variables are transformed as:

$$\lambda_{vp}'' = \frac{\lambda_{vp}}{\lambda_{x_p}}$$
(5.107)

$$\lambda_{yp}'' = \frac{\lambda_{yp}}{\lambda_{x_p}}$$
(5.108)

$$\lambda_{\psi p}'' = \frac{\lambda_{\psi p}}{\lambda_{x_p}}$$
(5.109)

$$\lambda_{y p}'' = \frac{\lambda_{y p}}{\lambda_{x_p}}$$
(5.110)

$$\lambda_{h p}'' = \frac{\lambda_{h p}}{\lambda_{x_p}}$$
(5.111)

Then the adjoint equations (5.101)-(5.106) are re-written as follows:

$$\begin{aligned}\frac{d\lambda_{vp}^*}{dt} &= \frac{2\lambda_{vp}^* D_p}{m_p v_p} + \frac{\lambda_{yp}^*}{m_p v_p^2} \left\{ (T_p \sin \alpha_p - L_p) \cos \phi_p - m_p g \cos \gamma_p \right\} \\ &\quad + \frac{\lambda_{wp}^*}{m_p v_p^2 \cos \gamma_p} (T_p \sin \alpha_p - L_p) \sin \phi_p - \cos \gamma_p \cos \psi_p \\ &\quad - \lambda_{yp}^* \cos \gamma_p \sin \psi_p - \lambda_{hp}^* \sin \gamma_p\end{aligned}\quad (5.112)$$

$$\begin{aligned}\frac{d\lambda_{yp}^*}{dt} &= \lambda_{vp}^* g \cos \gamma_p - \frac{\lambda_{yp}^* g \sin \gamma_p}{v_p} - \frac{\lambda_{wp}^* \tan \gamma_p}{m_p v_p \cos \gamma_p} (T_p \sin \alpha_p + L_p) \sin \phi_p \\ &\quad + v_p \sin \gamma_p \cos \psi_p + \lambda_{yp}^* v_p \sin \gamma_p \sin \psi_p - \lambda_{hp}^* v_p \cos \gamma_p\end{aligned}\quad (5.113)$$

$$\frac{d\lambda_{wp}^*}{dt} = v_p \cos \gamma_p \sin \psi_p - \lambda_{yp}^* v_p \cos \gamma_p \cos \psi_p \quad (5.114)$$

$$\frac{d\lambda_{hp}^*}{dt} = 0 \quad (5.115)$$

$$\frac{d\lambda_{hp}^*}{dt} = \frac{1}{m_p \rho(h_p)} \left\{ \lambda_{vp}^* D_p - \frac{\lambda_{yp}^* L_p \cos \phi_p}{v_p} - \frac{\lambda_{wp}^* L_p \sin \phi_p}{v_p \cos \gamma_p} \right\} \frac{dp}{dh_p} \quad (5.116)$$

Considering a bounded angle of attack for the pursuer, which is transformed into a new unbounded variable using (5.80), the first order optimality conditions are:

$$\begin{aligned}H_{\tau_{ap}} &= \frac{\pi}{4} \sin \tau_{ap} \left\{ \frac{\lambda_{vp}^*}{m_p} (-T_p \sin \alpha_p - \frac{D_p}{C_D(\alpha)} \frac{dC_D}{d\alpha}) + \frac{\lambda_{yp}^*}{m_p v_p} (T_p \cos \alpha_p \right. \\ &\quad \left. + \frac{L_p}{C_L(\alpha)} \frac{dC_L}{d\alpha}) \cos \phi_p + \frac{\lambda_{wp}^*}{m_p v_p \cos \gamma_p} \left(T_p \cos \alpha_p + \frac{L_p}{C_L(\alpha)} \frac{dC_L}{d\alpha} \right) \sin \phi_p \right\} = 0\end{aligned}\quad (5.117)$$

$$H_{\phi_p} = -\frac{\lambda_{yp}^*}{m_p v_p} (T_p \sin \alpha_p + L_p) \sin \phi_p + \frac{\lambda_{wp}^*}{m_p v_p \cos \gamma_p} (T_p \sin \alpha_p + L_p) \cos \phi_p = 0 \quad (5.118)$$

The boundary conditions corresponding to the adjoint equations are obtained through the following process:

$$\Phi = -t_f + v_1(x_p - x_e) + v_2(y_p - y_e) + v_3(z_p - z_e) + v_4(\psi_p - \psi_e) \Big|_{t=t_f} \quad (5.119)$$

$$\lambda_{x_p}(t_f) = \frac{d\Phi}{dx_p} \Big|_{t=t_f} = v_1 \quad (5.120)$$

$$\lambda_{v_p}''(t_f) = \frac{1}{\lambda_{x_p}(t_f)} \frac{d\Phi}{dv_p} \Big|_{t=t_f} = 0 \quad (5.121)$$

$$\lambda_{y_p}''(t_f) = \frac{1}{\lambda_{x_p}(t_f)} \frac{d\Phi}{dy_p} \Big|_{t=t_f} = 0 \quad (5.122)$$

$$\lambda_{z_p}''(t_f) = \frac{1}{\lambda_{x_p}(t_f)} \frac{d\Phi}{dz_p} \Big|_{t=t_f} = \frac{v_4}{v_1} \quad (5.123)$$

$$\lambda_{\psi_p}''(t_f) = \frac{1}{\lambda_{x_p}(t_f)} \frac{d\Phi}{d\psi_p} \Big|_{t=t_f} = \frac{v_2}{v_1} \quad (5.124)$$

$$\lambda_{h_p}''(t_f) = \frac{1}{\lambda_{x_p}(t_f)} \frac{d\Phi}{dh_p} \Big|_{t=t_f} = \frac{v_3}{v_1} \quad (5.125)$$

As boundary conditions of the adjoint variables in the semi-DCNLP problem, only (5.121) and (5.122) are applied because v_1, v_2, v_3 and v_4 are unknown parameters.

The semi-DCNLP formulation of the air combat problem is thus:

$$J = \max_{\gamma_e, \psi_e} t_f \quad (5.126)$$

subject to (5.1)-(5.6) and (5.121)-(5.125) as differential equations, (5.108)-(5.111) and (5.121)-(5.122) as terminal conditions, and (5.107)-(5.108) as algebraic constraints, for given initial condition.

To make the air combat problem realistic, we choose initial conditions representative of real air combat, and to make the problem interesting the aircraft need

altitude to maneuver. In this research, 20,000 ft altitude and 400 ft/sec velocity are selected as a nominal, initial condition.

To evaluate three-dimensional pursuit-evasion maneuvering, the relative initial position and relative motion between the pursuer and the evader are given out-of-plane components. The nominal, initial condition is that: (1) the evader flies on the right side of the pursuer with a line of sight 30 deg from forward (2) both aircraft fly in the same direction (3) both aircraft are in level flight and (4) the distance between the pursuer and the evader is 4,000 ft.

The pursuer is put at the origin of the co-ordinate system at the initial time. Thus the initial condition for nominal case is:

$$\begin{aligned} & [v_p, \gamma_p, \psi_p, x_p, y_p, h_p, v_e, \gamma_e, \psi_e, x_e, y_e, h_e] \\ & = \left[1.0, 0.0, 0.0, 0.0, 0.0, 5.0, 1.0, 0.0, 0.0, \cos(-\frac{\pi}{6}), \sin(-\frac{\pi}{6}), 5.0 \right] \end{aligned} \quad (5.127)$$

Using an initial guess on the basis of a solution of the two-dimensional problem, from Sec. 5.3.1, the semi-DCNLP method finds the saddle-point trajectory for the nominal case as shown in Figs. 5.23 - 5.32.

We observe that:

(1) The trajectories (Fig. 5.23 - 5.25) show the three-dimensional realistic air combat problem (pursuit-evasion problem) has two phases, qualitatively speaking. In the

first phase, aircraft are steered in three-dimensional space so that their maneuvers will be in a vertical plane in a second phase. In the second phase, the aircraft mainly do dive maneuvers in a vertical plane. These characteristics of the two phases are observed in Fig. 5.26, Fig. 5.27, Fig. 5.30 and Fig. 5.31. In these figures, the characters of the time histories change at approximately 0.9 sec. These characteristics are qualitatively observed in simpler problems in this chapter and in Sec. 3.4. For example, the "Homicidal Chauffeur Problem" has a similar two phase structure.

(2) The bank angle in the second phase shows chattering control (Fig. 5.26).

This is supported by the fact that λ_{rp}'' and $\lambda_{\psi\text{p}}''$ become zero (Fig. 5.32), i.e., (5.118) cannot determine the bank angle. This was also observed in the homicidal chauffeur problem in Sec. 3.4. Except for this observation, the bank angle control is consistent with Figs. 5.23 - 5.25, i.e., the aircraft takes a bank angle in the first phase but does not in the second phase.

(3) Figs. 5.27 and 5.28 show that the aircraft uses feasible vertical maneuvering.

The normal acceleration ranges from 1G through 5G throughout the time history. Also, the AOA is below 15 degrees. This means that the pursuer does not use post-stall flight capability in this case.

(4) Fig. 5.32 shows the histories of the adjoint variables. The line means the solution of (5.112)-(5.116) obtained by the semi-DCNLP method, whereas the symbol shows the Lagrange multipliers corresponding to (5.1) - (5.6). The semi-DCNLP

solution provides the correct saddle-point trajectory of the pursuit-evasion problem when the symbols lie on the lines. The symbols in Fig. 5.32 are on the line from the initial to the mid-time. Some differences from mid-time to terminal time are explained from observation (2), i.e., the bank angle history has a lot of chattering. Also, the adjoint variables for velocity, flight path angle and altitude are qualitatively consistent with those of the realistic two-dimensional air combat problem. Thus, we believe the solution is valid numerically.

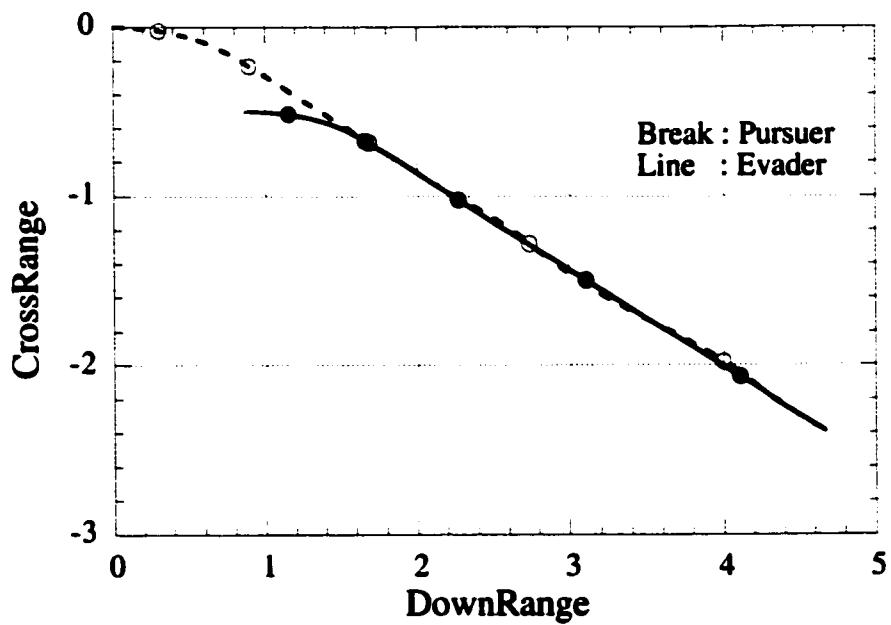


Figure 5.23 Saddle-Point Trajectories on a Horizontal Plane for Three-Dimensional Realistic Air Combat (Nominal Case)

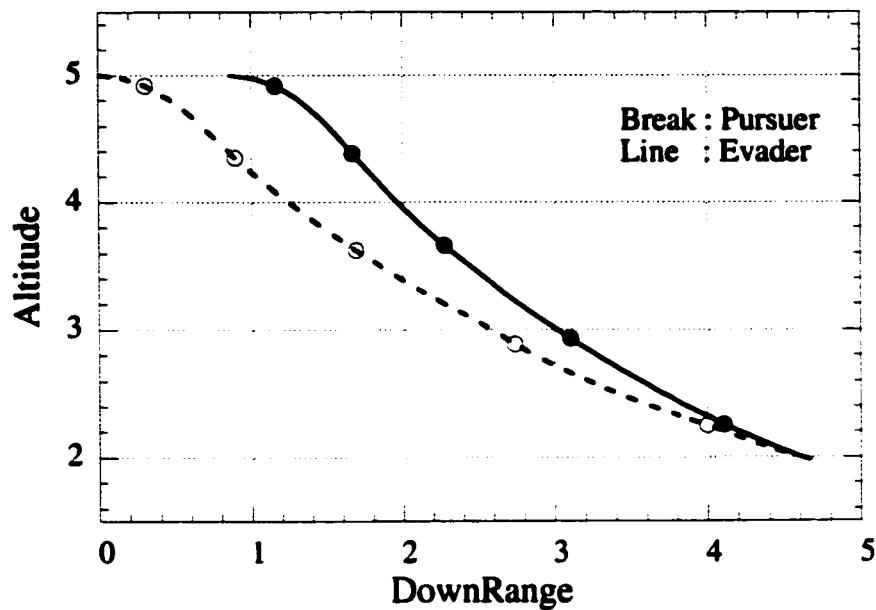


Figure 5.24 Saddle-Point Trajectories on a Vertical (X-Z) Plane for Three-Dimensional Realistic Air Combat (Nominal Case)

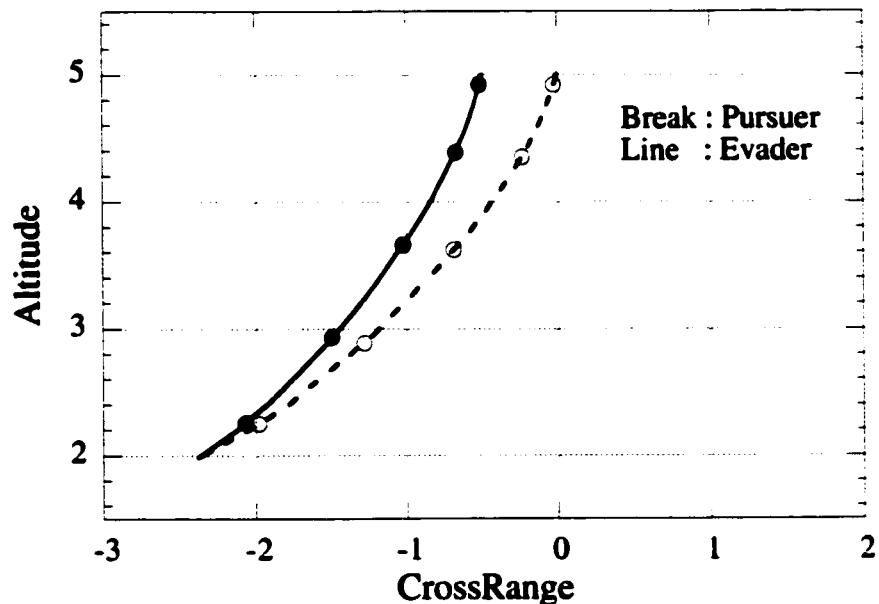


Figure 5.25 Saddle-Point Trajectories on a Vertical (Y-Z) Plane for Three-Dimensional Realistic Air Combat (Nominal Case)

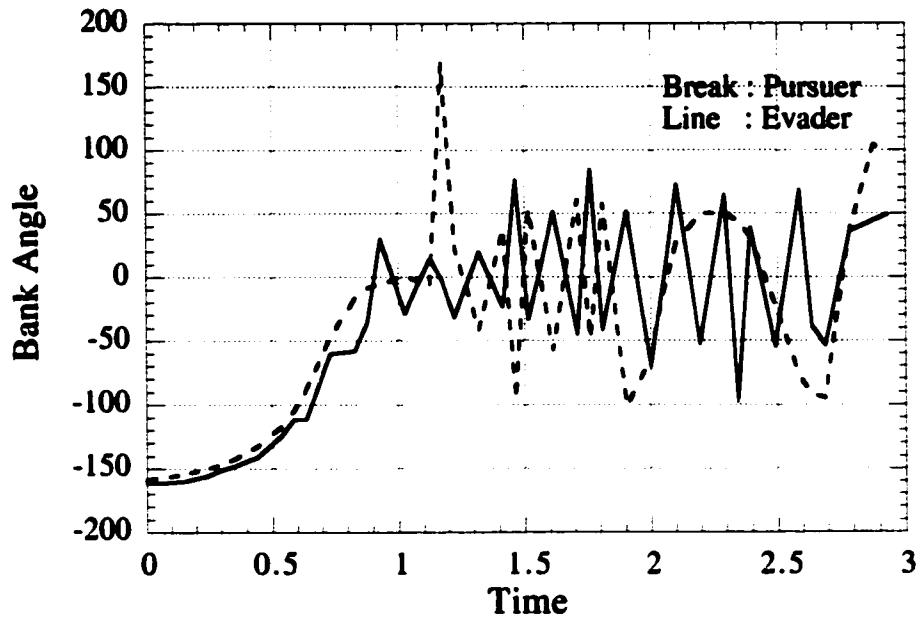


Figure 5.26 Histories of Bank Angle of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

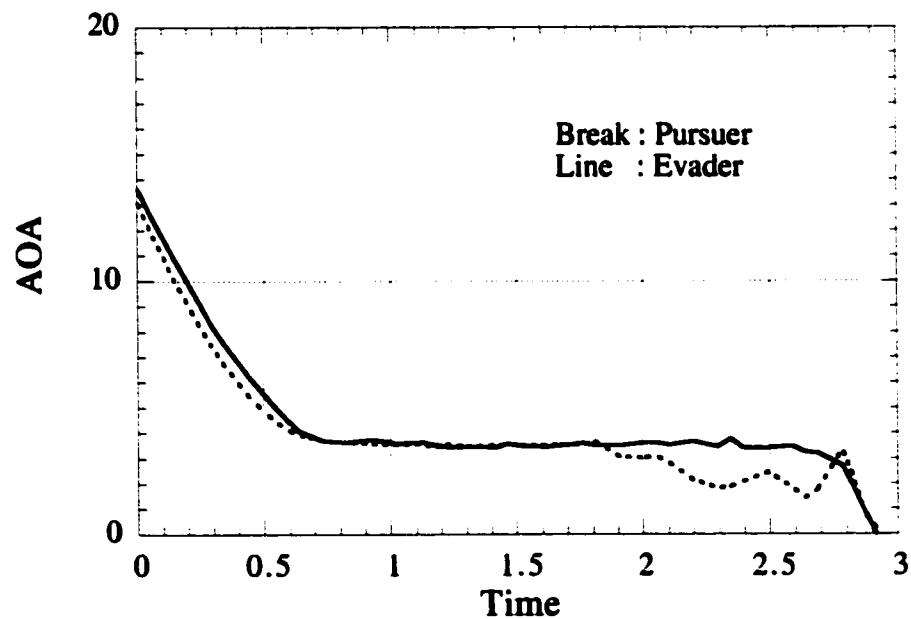


Figure 5.27 Histories of Angle of Attack of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

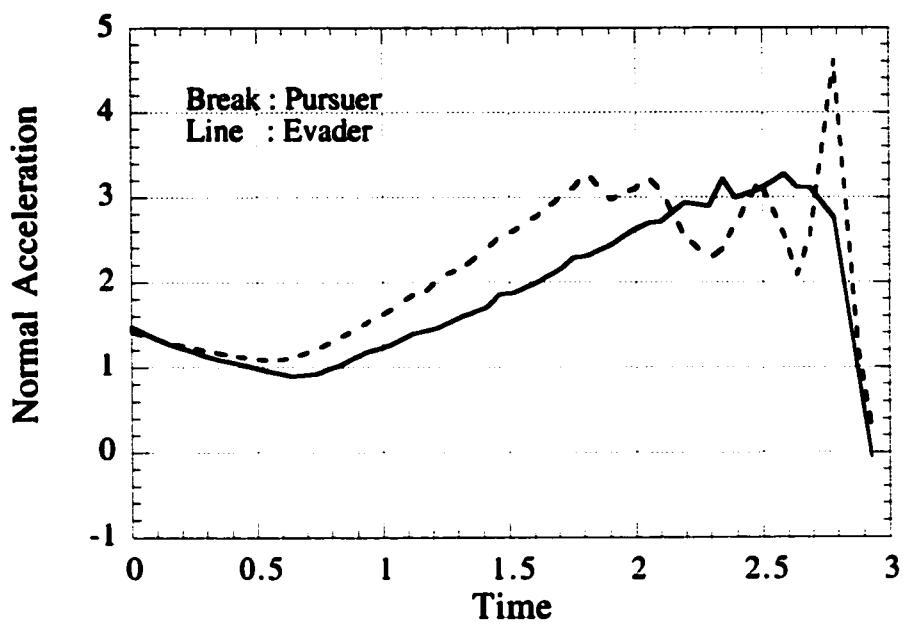


Figure 5.28 Histories of Normal Acceleration of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

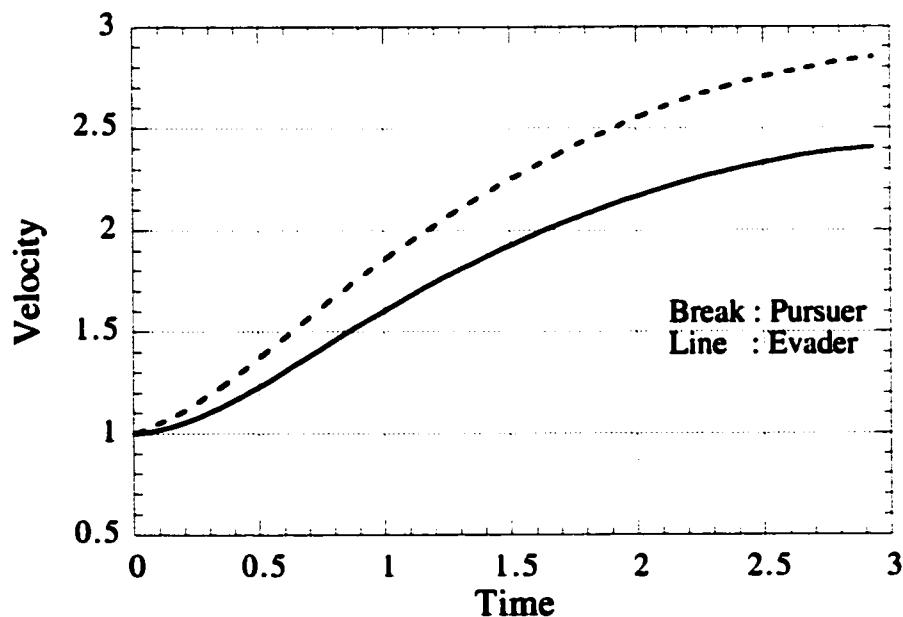


Figure 5.29 Histories of Velocity of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

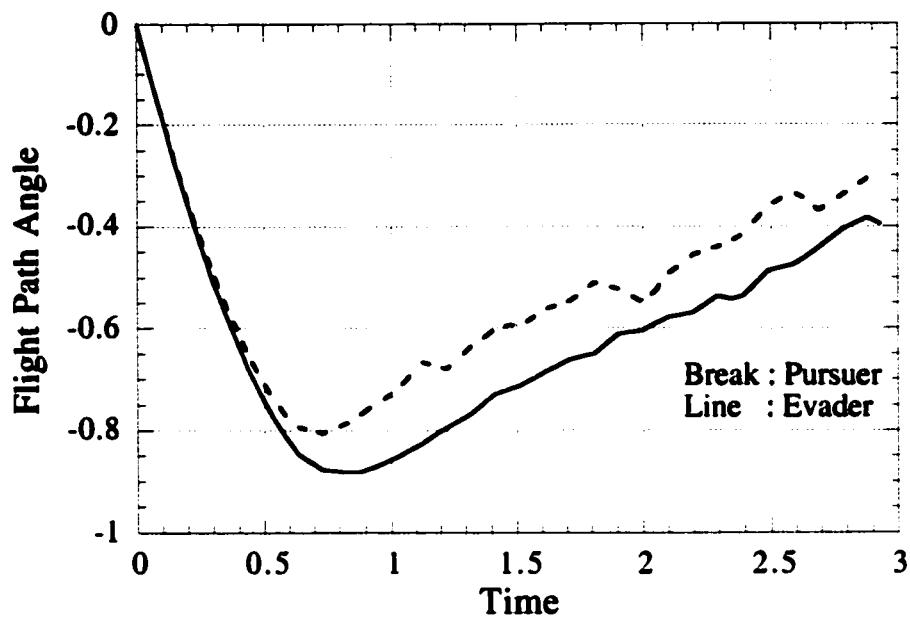


Figure 5.30 Histories of Flight Path Angle of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

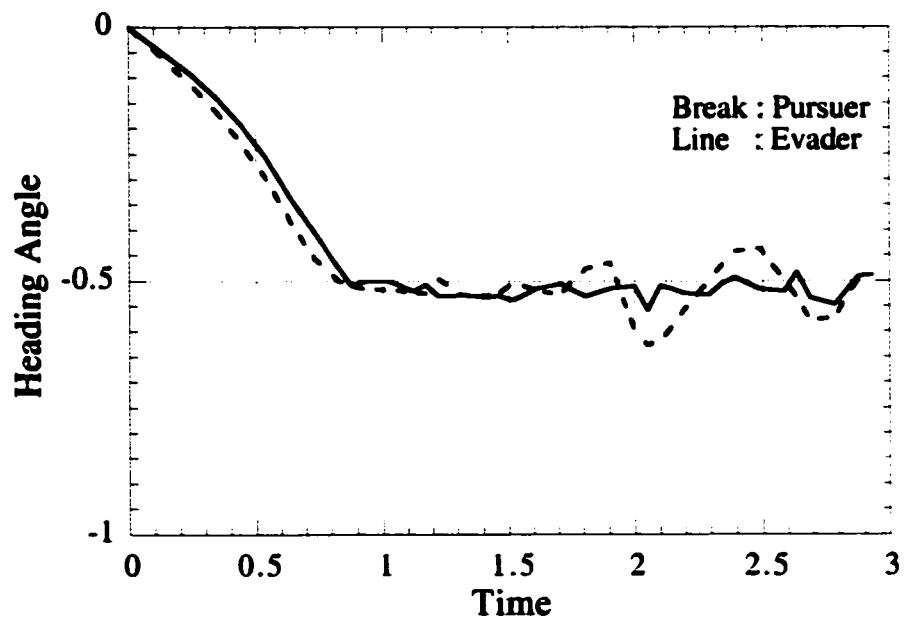


Figure 5.31 Histories of Heading Angle of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

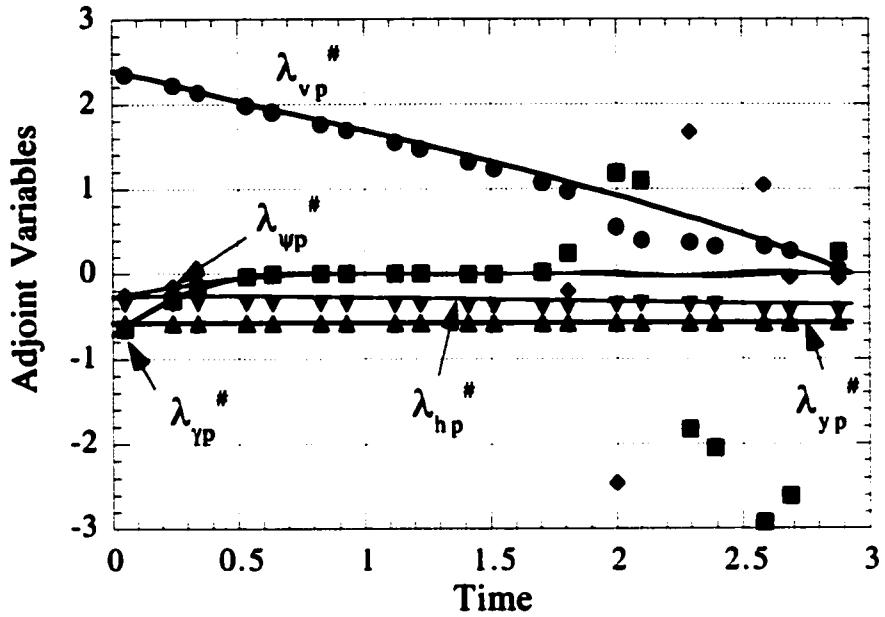
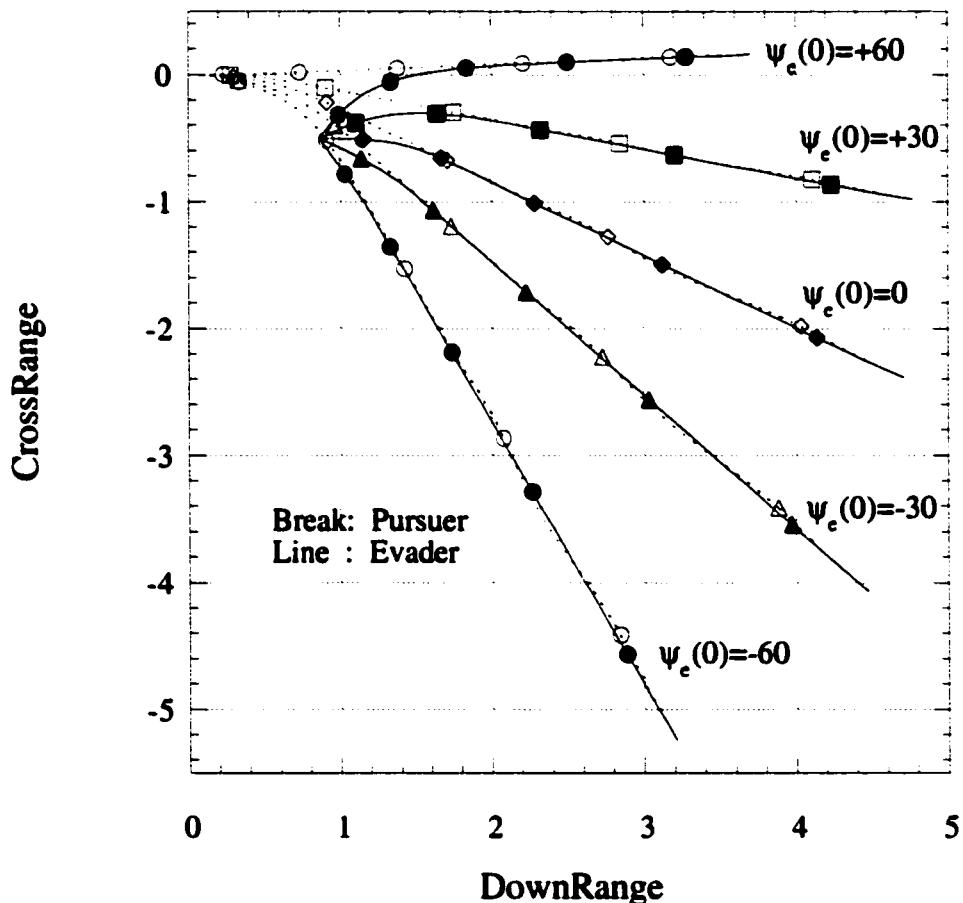


Figure 5.32 Histories of Adjoint Variables of Saddle-Point Trajectories for Three-Dimensional Realistic Air Combat (Nominal Case)

To evaluate the effect of relative motion, the initial heading angle of the evader is varied while maintaining the same initial condition as the nominal case for the other parameters. The semi-DCNLP method yields convergent solutions for the cases ranging from -60° to $+60^\circ$ of the initial evader's heading angle.

Figs. 5.33 - 5.36 show the group of the saddle-point trajectories. All the trajectories obtained are qualitatively the same as the nominal case (in which the initial heading angle of the evader is zero). Also, the smaller the initial heading angle, the smaller the change of the heading angle is. This tendency is recognized in Fig. 5.38, the

relationship between the initial and the final heading angle of the evader. The figure shows the relationship is linear and predicts that the initial angle is equal to the final angle around -75° . It is predicted that the evader maneuvering changes qualitatively beyond this point, otherwise the magnitude of the initial heading angle of the evader becomes larger than the magnitude of the final heading angle and in that case the evader would approach the pursuer.



**Figure 5.33 Saddle-Point Trajectories on a Horizontal Plane
(Parameter: Initial Heading Angle of Evader)**

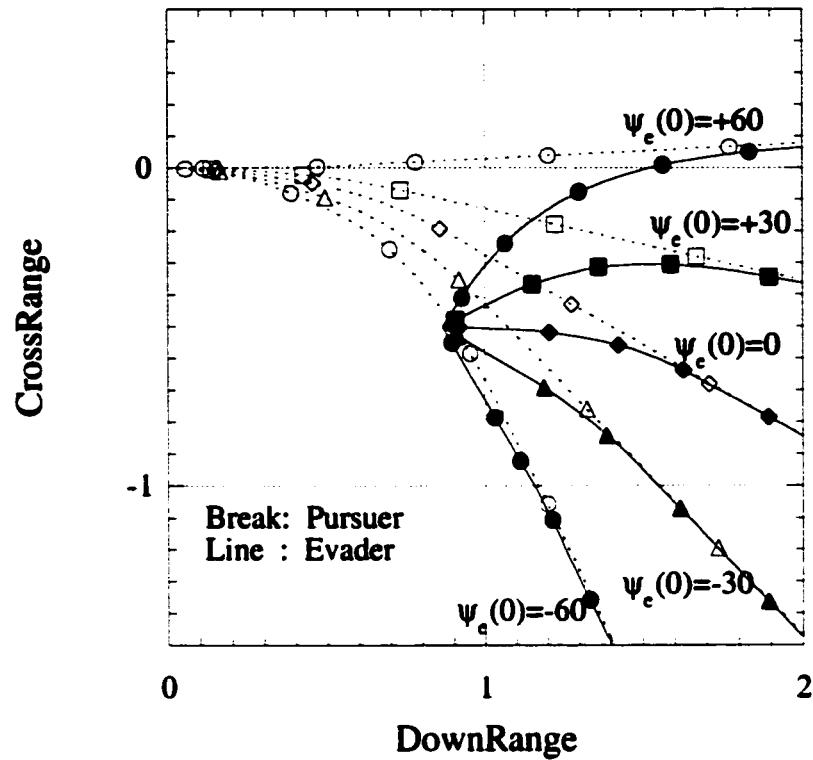


Figure 5.34 Saddle-Point Trajectories on a Horizontal Plane - Detail
(Parameter: Initial Heading Angle of Evader)

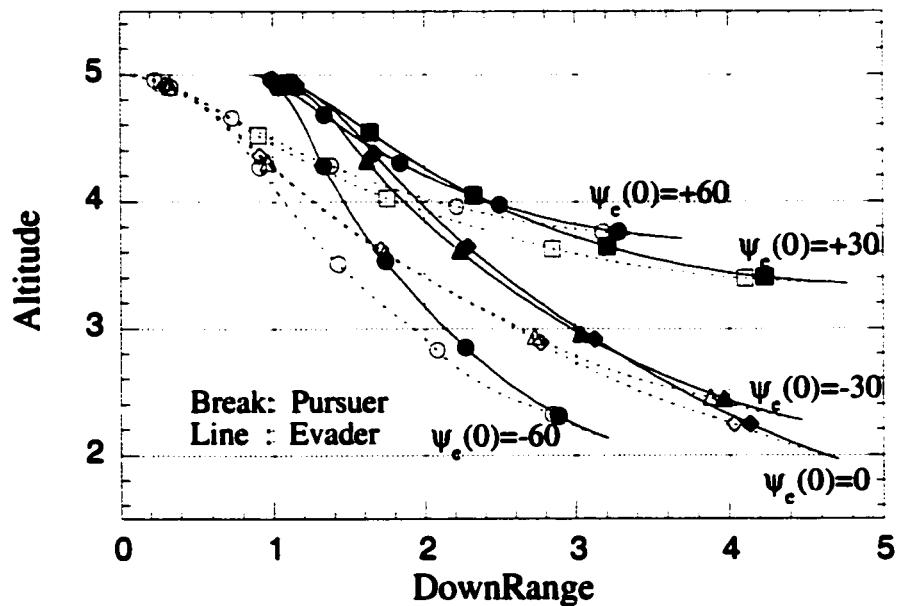
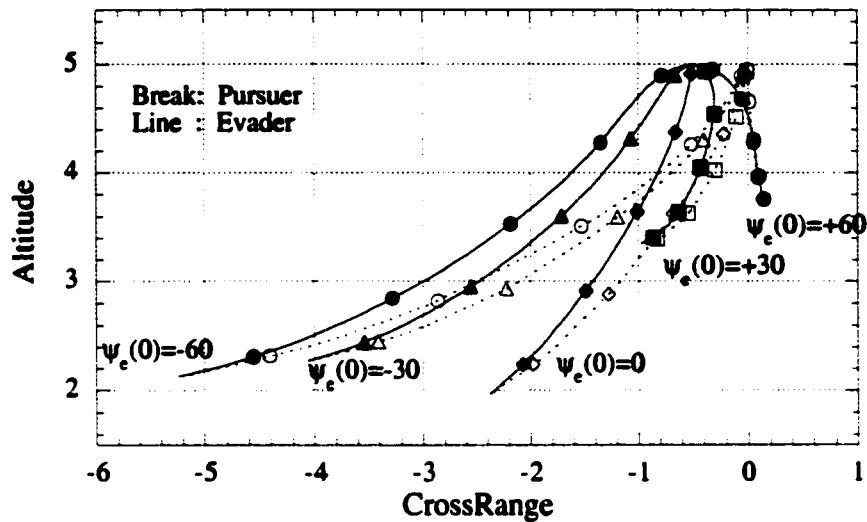


Figure 5.35 Saddle-Point Trajectories on a Vertical (X-Z) Plane
(Parameter: Initial Heading Angle of Evader)



**Figure 5.36 Saddle-Point Trajectories on a Vertical (Y-Z) Plane
(Parameter: Initial Heading Angle of Evader)**

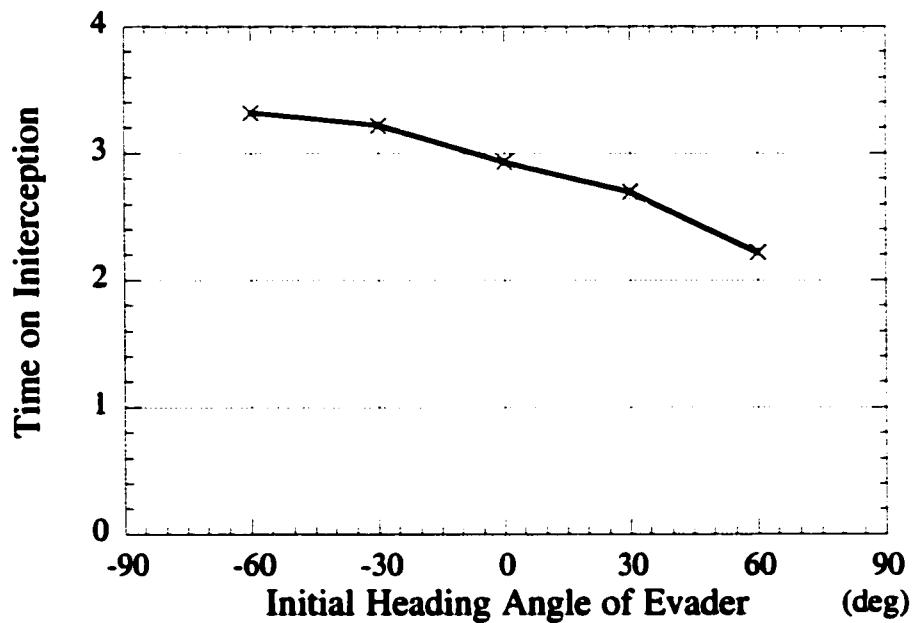
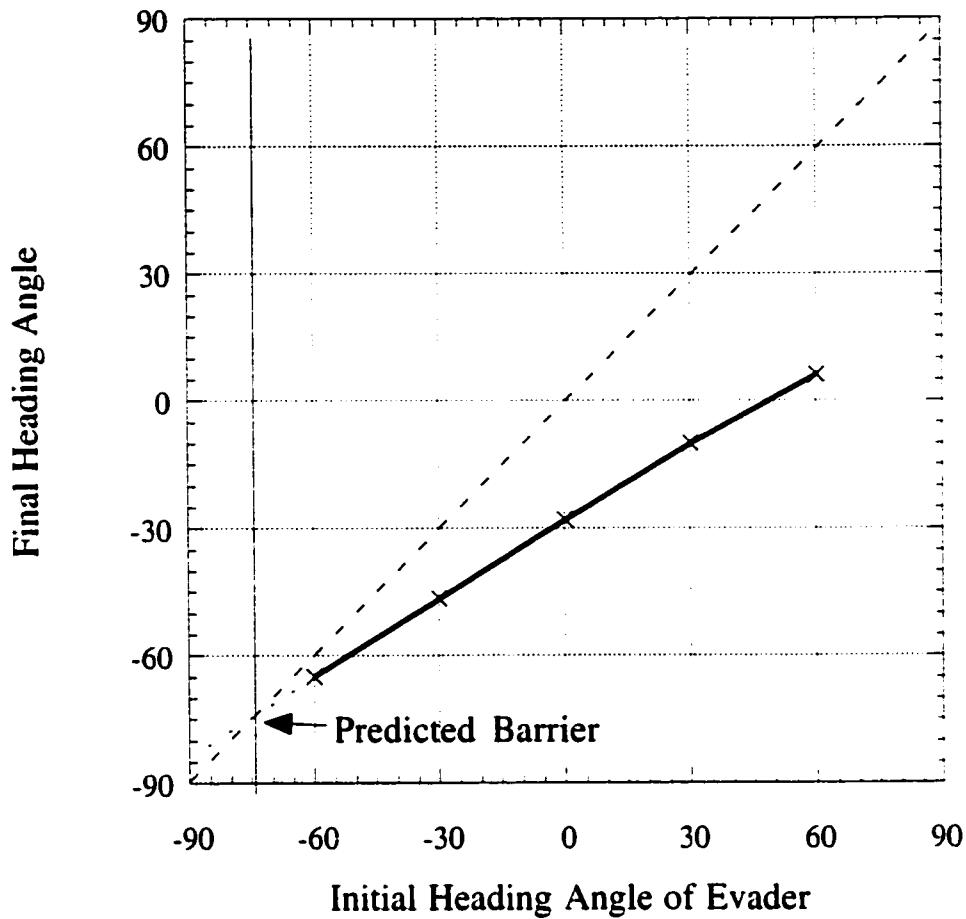


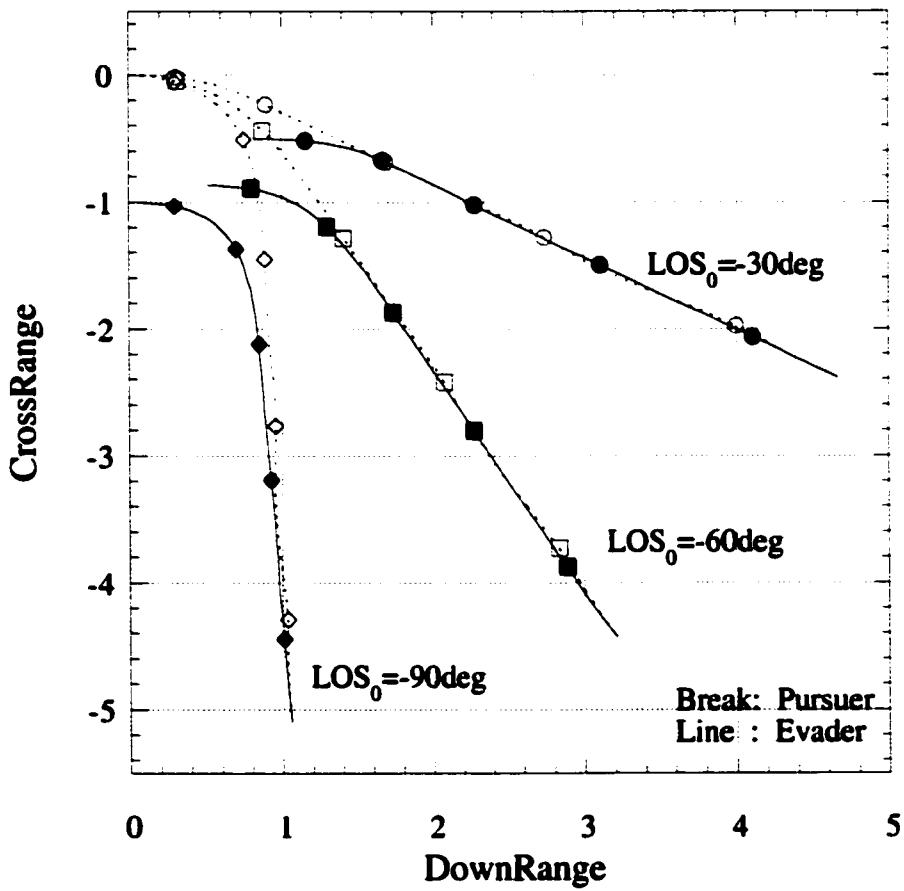
Figure 5.37 Time to Interception (Parameter: Initial Heading Angle of Evader)



**Figure 5.38 Initial Heading Angle vs. Final Heading Angle
(Parameter: Initial Heading Angle of Evader)**

To evaluate the effect of initial relative position, the initial Line of Sight (LOS) is varied while maintaining the same initial condition as the nominal case for the other parameters. The semi-DCNLP method yields convergent solutions for the cases ranging from -30° to -90° of the initial LOS.

Figs. 5.39 - 5.42 show a group of saddle-point trajectories. Also, Fig. 5.43 describes the relationship between the initial LOS and the final heading angle. These figures suggest an interesting result; that the initial LOS is close to the final heading angle. Fig. 5.44 shows the relationship between the initial LOS and the time to interception, the cost function. It is observed from Fig. 5.44 that the initial LOS hardly affects the value of the cost function.



**Figure 5.39 Saddle-Point Trajectories on a Horizontal Plane
(Parameter: Initial LOS)**

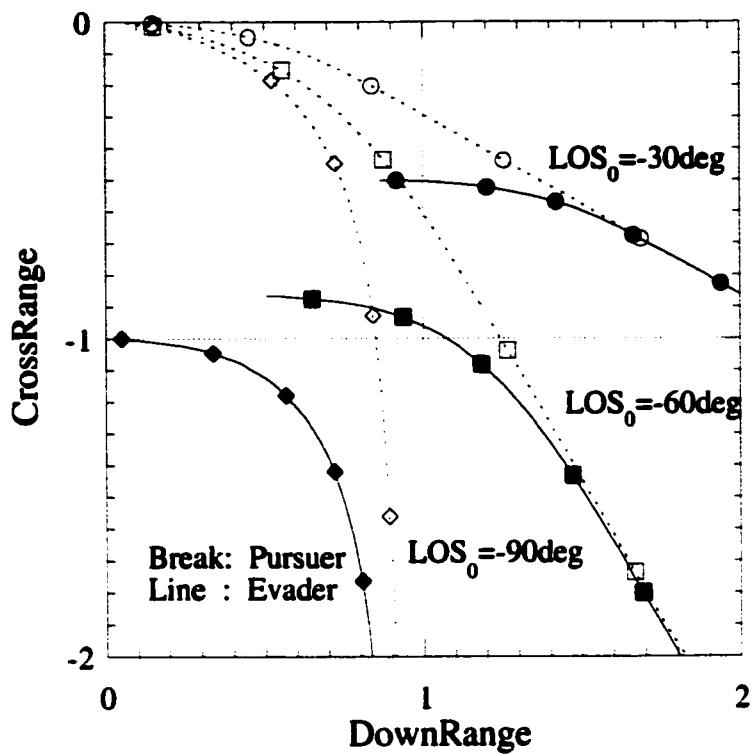


Figure 5.40 Saddle-Point Trajectories on a Horizontal Plane - Detail
(Parameter: Initial LOS)

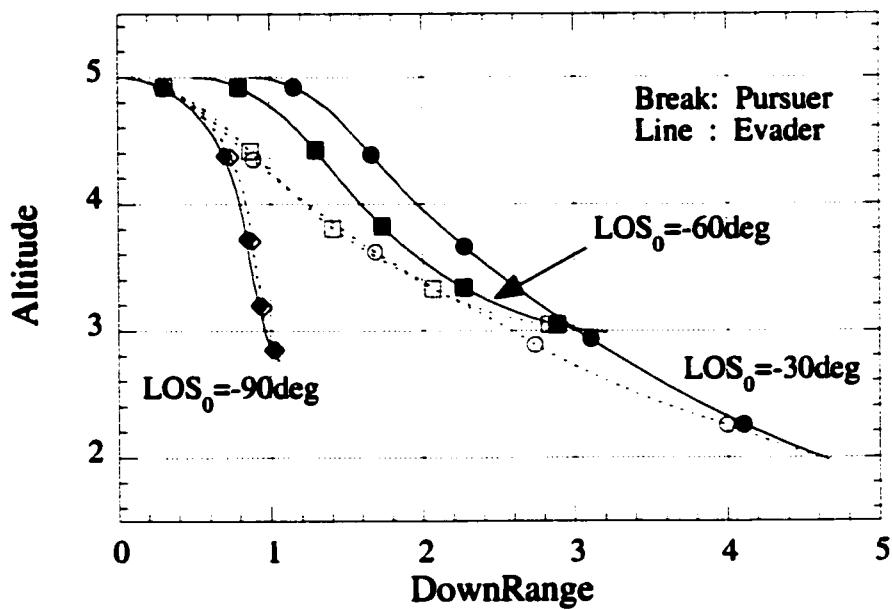
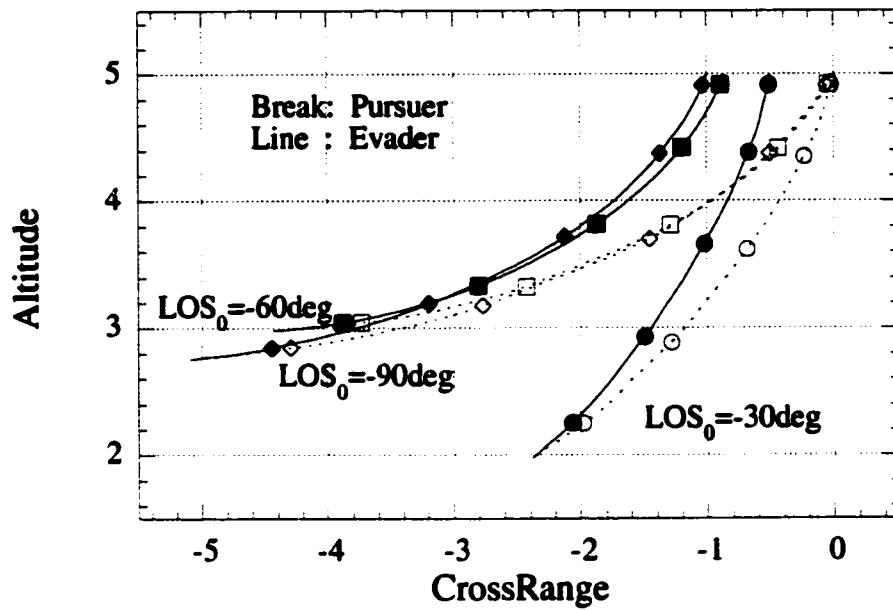


Figure 5.41 Saddle-Point Trajectories on a Vertical (X-Z) Plane
(Parameter: Initial LOS)



**Figure 5.42 Saddle-Point Trajectories on a Vertical (Y-Z) Plane
(Parameter: Initial LOS)**

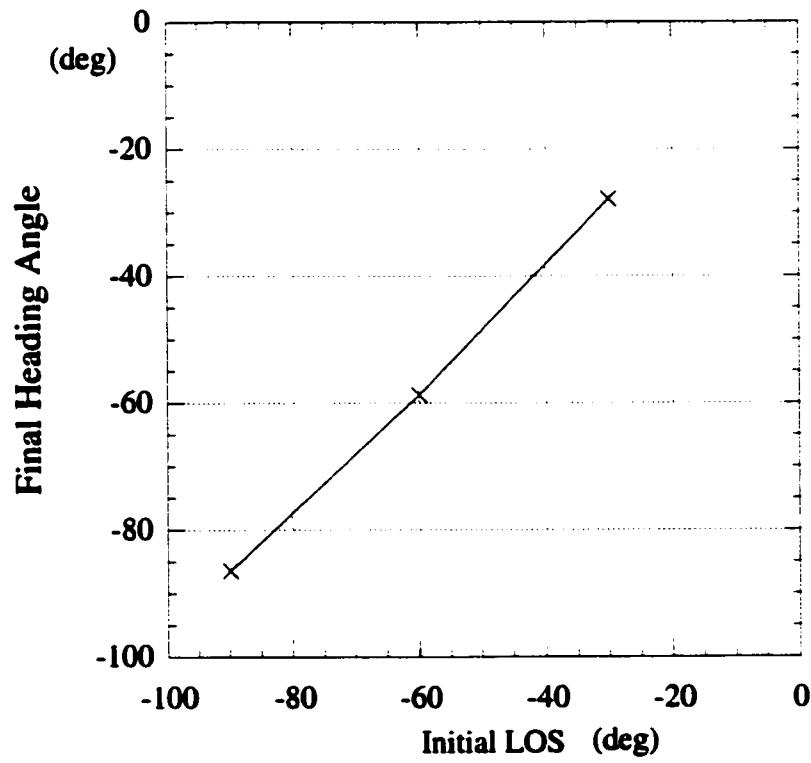


Figure 5.43 Initial LOS vs. Final Heading Angle (Parameter: Initial LOS)

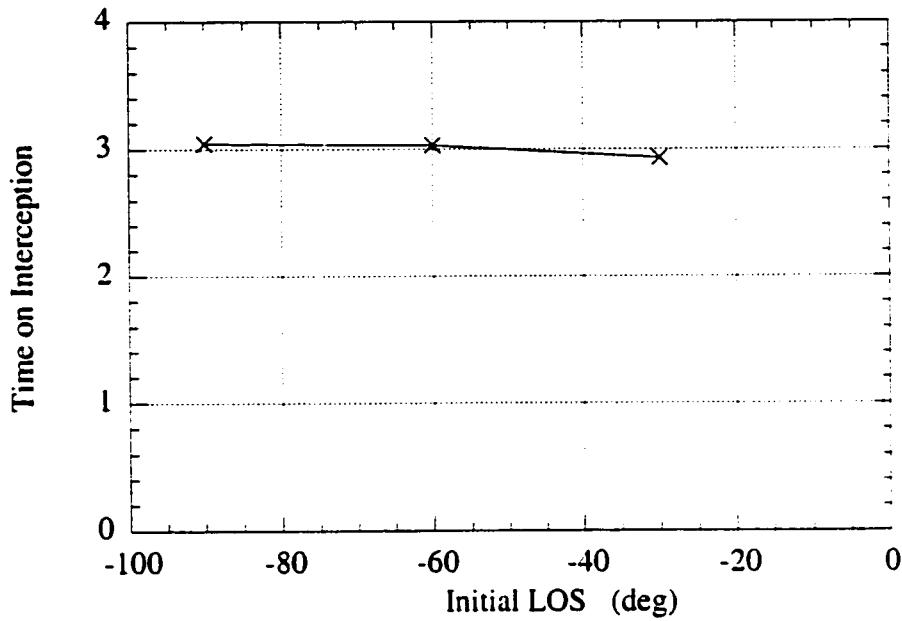


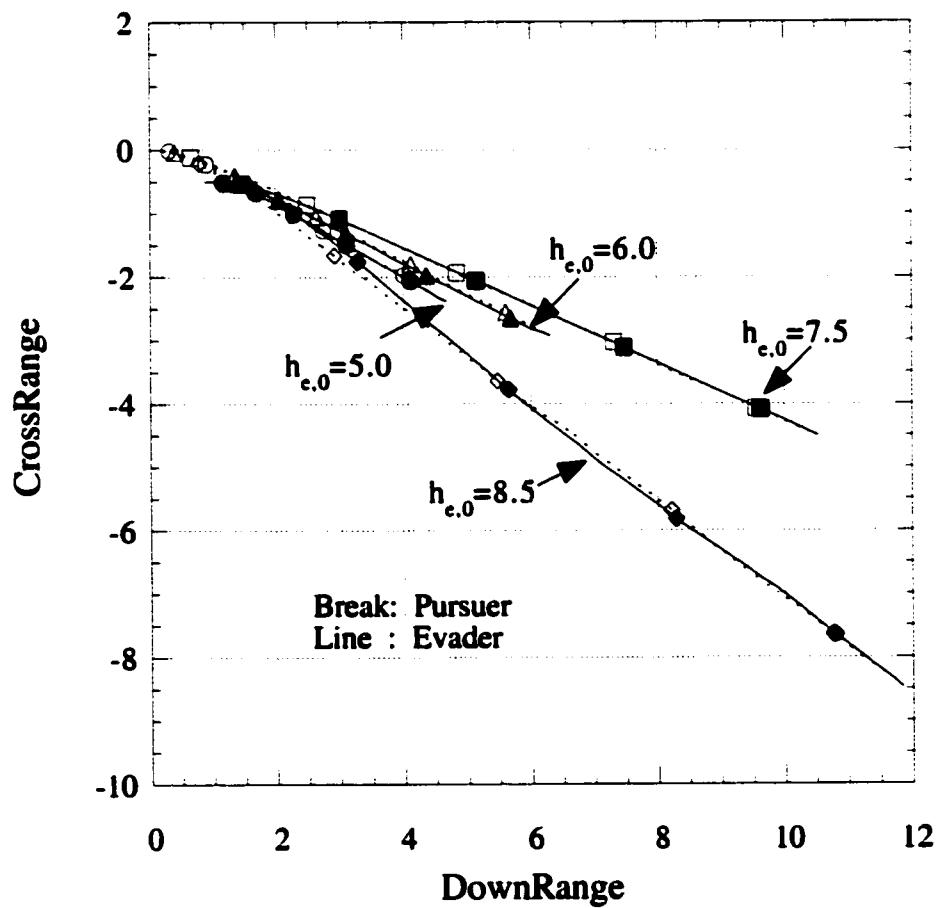
Figure 5.44 Time to Interception (Parameter: Initial LOS)

The initial altitude of the evader is varied to evaluate the effect of altitude differences while maintaining the same initial condition as in the nominal case for the other parameters. The evader' inferior thrust does not change to avoid complexity.

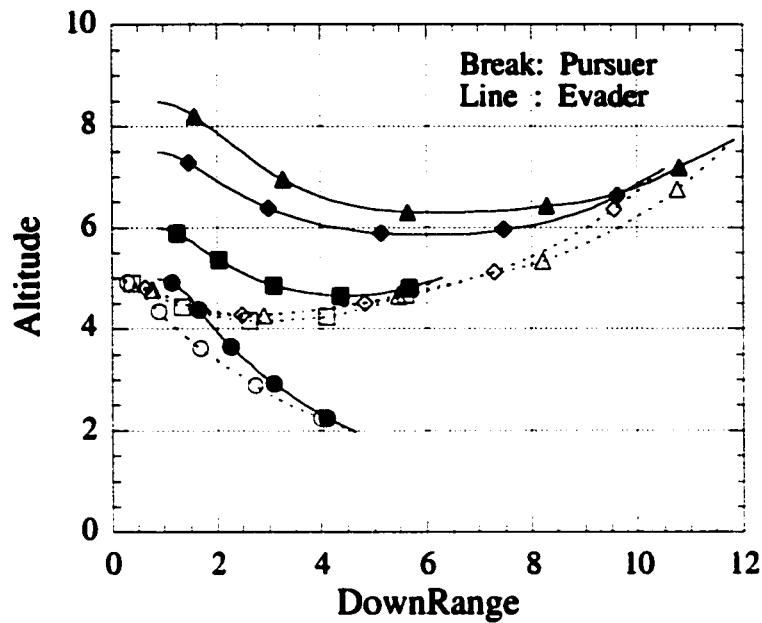
Figs. 5.45 - 5.47 show a group of saddle-point trajectories. No obvious relationship is observed between the altitude difference and the final heading angle from Fig. 5.45. On the other hand, Fig. 5.46 and Fig. 5.47 indicate that both aircraft dive once during the pursuit-evasion maneuvers. This can be explained by the fact that the aircraft increase their velocities by losing altitude at first to satisfy each purpose as discussed in Sec. 5.2.1. Also, we believe that the evader gains advantage by climbing directly beyond

a certain initial altitude because he can then separate from the pursuer. Therefore, the solution cannot be obtained beyond 34,000 ft altitude of the evader.

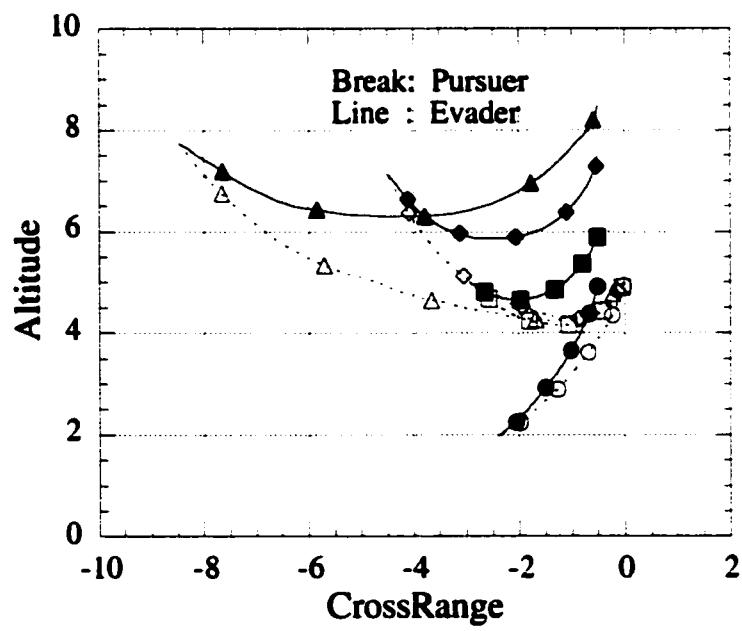
The value of the cost function increases linearly as the altitude difference increases, as seen in Fig. 5.48. This is a straightforward result because the altitude difference increases the distance between the pursuer and the evader.



**Figure 5.45 Saddle-Point Trajectories on a Horizontal Plane
(Parameter: Initial Altitude of Evader)**



**Figure 5.46 Saddle-Point Trajectories on a Vertical (X-Z) Plane
(Parameter: Initial Altitude of Evader)**



**Figure 5.47 Saddle-Point Trajectories on a Vertical (Y-Z) Plane
(Parameter: Initial Altitude of Evader)**

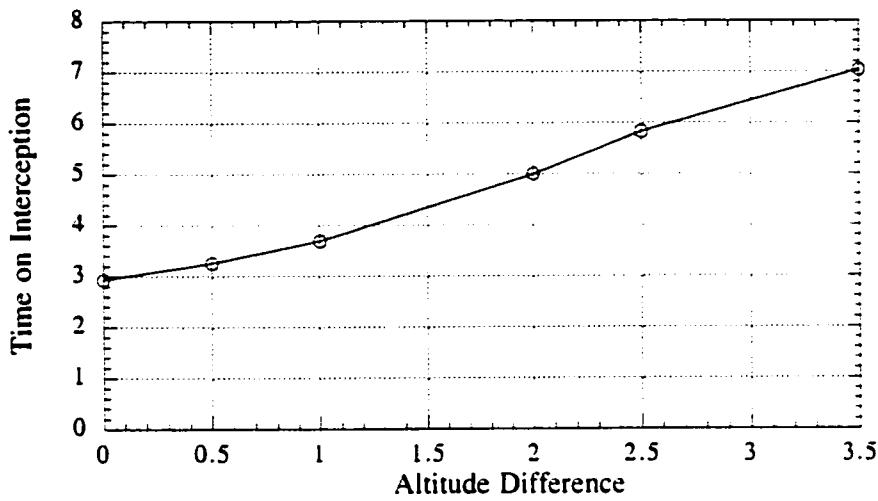


Figure 5.48 Altitude Difference vs. Time to Interception

In conclusion, the three-dimensional realistic air combat problem has been formulated as a kind of pursuit-evasion problem and solved using the semi-DCNLP method. The resulting trajectories consist of two phases of maneuvers. The maneuvering in the first phase is out-of-plane maneuvering whereas in the second phase the maneuvering is almost two-dimensional, taking place on a vertical-plane. The fact that the maneuvers have two qualitatively different phases is characteristic of the simpler air combat problem and problems such as the "Homicidal Chauffeur".

The problem has been studied for various initial LOS, heading angle and altitude of the evader. The initial LOS hardly affects the value of the cost function and is close to the final heading angle. On the other hand, the value of the cost function increases as the

initial heading angle decreases because the evader escapes from the pursuer in this case.

A larger initial altitude difference also increases the final time, as one would expect since this increases the initial separation.

The semi-DCNLP successfully solves the complicated realistic air combat problem. There are very few solutions of such complicated air combat problems in the literature [15, 16]. They typically use a well known multiple shooting method or iteration of a DCNLP, which decomposes the problem into two sub-problems for optimal control and solves a sequence of sub-problems with a pre-specified capture point of one player. The semi-DCNLP method has not experienced the sensitivity to the initial guess characteristic of a shooting method and has easily handled the problem as has been the case when the DCNLP method is applied. The semi-DCNLP is thus believed to be an excellent numerical method for complicated two-sided flight path optimization because of its robustness and relative ease of use.

Chapter 6: Conclusions

6.1 Summary of Research

The main contribution of this work is in the development of a new numerical method for solving two-sided optimization problems (i.e. pursuit-evasion games) using collocation with nonlinear programming. Solution of this problem required an extension of the calculus of variations to such problems and also required development of a new method, based on genetic algorithms (GA), for finding approximate optimal trajectories.

In Chapter 2, a numerical method for one-sided flight path optimization is described and then evaluated as the optimizer for a problem regarding optimal air combat maneuvering. The direct collocation with nonlinear programming method (DCNLP) is introduced as a robust solver for a one-sided flight path optimization. Then the method is applied to air combat maneuvers in a vertical-plane in which the pursuing aircraft maintains level flight. The optimal maneuver is obtained for various required terminal conditions, related to placing the airplane in a position from which it may now attack its pursuer, and this maneuver is found to have the characteristics of the "cobra" maneuver, perhaps the most well-known of the post-stall maneuvers. This work demonstrates that the DCNLP method is capable of finding optimal air combat trajectories.

A new method of numerical analysis is developed in Chapter 3 for a two-sided flight path optimization. The algorithm is developed on the basis of the DCNLP method by incorporating some of the system adjoint equations, i.e. some of the analytical

necessary conditions, into the system equation of the problem; solving for the control variables of one player using the Pontryagin principle but finding those for the other player via nonlinear programming. We refer to this new method as the semi-DCNLP method since it is no longer a purely “direct” method. Thereafter, it is verified that the semi-DCNLP method is capable of finding an open-loop representation of an optimal saddle-point trajectory under some conditions.

Several differential games are solved using the semi-DCNLP method to evaluate its abilities. First it is shown that the semi-DCNLP method can find an optimal saddle-point trajectory for a simple differential game by solving the *dolichobrachistochrone* problem. Next we show that the semi-DCNLP method is capable of solving a pursuit-evasion problem in fixed coordinates by applying it to the ballistic interception problem. As a third example, the famous *homicidal chauffeur* problem is solved to verify that the semi-DCNLP solution identifies some of the singular surfaces. The trajectories obtained correctly include a universal surface and/or a switching surface. This verifies that the semi-DCNLP method is capable of identifying singular surfaces, which are often included in the saddle-point trajectories. Finally, a problem of minimum-time spacecraft interception with an optimally evasive target is solved to verify the capability of the semi-DCNLP method as a solver of a realistic dynamic system. We conclude from these four successful applications that the semi-DCNLP method has the ability to find open-loop representations of feedback saddle-point trajectories.

In Chapter 4, a pre-processing algorithm is developed to provide an initial guess of the solution required when one is using the semi-DCNLP method. The desired pre-processing algorithm must be able to find an initial guess in or near the feasible region without any *a priori* information regarding the optimal trajectories. For the homicidal chauffeur problem, a simple genetic algorithm (GA) based pre-processor provides a satisfactory initial guess, that is, the initial guess yields a convergent solution of the NLP problem solver used by the semi-DCNLP method. The efficiency of the pre-processor developed here is evaluated by comparing it with a standard hybrid GA/local optimizer. The new method is more efficient than the standard hybrid GA with regard to actual operation time. However the standard hybrid algorithm converges in fewer generations than the new method.

Finally, in Chapter 5, the capstone problem, an air combat problem with realistic aerodynamic and aircraft models, between a superior fighter and an inferior fighter, is modeled using a pursuit-evasion game approach and is solved using the semi-DCNLP method. Solutions of this problem are rarely seen in the literature because the problem is very complicated and difficult to solve using extant methods. However the semi-DCNLP method solves the problem easily and robustly. For the two-dimensional case in which both airplanes maneuver only in a vertical plane, both the pursuer and the evader dive when the evader is in front of the pursuer. The result suggests that the optimal maneuver uses gravity in order to increase the velocity of each aircraft. For three-dimensional air

combat we find that the optimal trajectory consists qualitatively of two phases. The aircraft first maneuver out of the original plane of their motion, followed by a second phase in which the motion is mainly confined to a vertical-plane. It is also found that change in the initial line-of-sight changes the value of cost function only slightly whereas change in the initial heading angle and altitude of the evader change the value of the cost function significantly.

6.2 Recommendations for Future Research

The numerical methods, the DCNLP and the new semi-DCNLP, greatly enrich the research field of flight path optimization. Several interesting research areas are recommended for the future based on this work.

Optimization of precise one-sided air combat maneuvers: This thesis research is mainly focused on the two-sided optimization problem. However, one-sided optimization is useful to evaluate the performance of fighter aircraft still in the design phase because indices of performance, such as minimum-time turning, are usually defined using single aircraft maneuvers. We recommend that the aircraft model for the optimization be extended from three degree-of-freedom motion to six degree-of-freedom motion in order to evaluate and identify the characteristics of the optimal maneuvers more precisely. This work will likely meet with difficulties due to the complexity caused

by having more system variables, and by constraints with regard to dynamic stability and control surface effectiveness.

Extension of the semi-DCNLP for path constrained problems: The semi-DCNLP method has been developed here for a two-sided optimization problem with no path constraints. It would be desirable to extend the method to a problem including path constraints because flight path optimization problems often require them, for example constraints on aerodynamic heating rate. A suggestion is to incorporate conditions for discontinuities of the adjoint variables for one player into the semi-DCNLP formulation, in addition to the other optimality conditions for that player. A theoretical development and some tested applications for the path-constrained problem are required to verify this suggestion.

Hybrid GA/semi-DCNLP: This research suggests the potential of the hybrid GA/semi-DCNLP method to have excellent convergent characteristics. A potential problem found in this research is computation time, which is serious in the present but may not be in the future. An application of the hybrid method to several simple two-sided optimization problems using a standard CPU and to more complicated problems using a supercomputer might identify severe obstacles of the method not found in this research.

Realistic air combat analysis: This research has identified some characteristics of pursuit-evasion type air combat using the semi-DCNLP approach. Two different

directions of approach are recommended as extensions of this research. One direction is to use a more accurate model and then identify more precisely the characteristics of the air combat. Another direction would be to evaluate the qualitative differences in optimal air combat maneuvering among a variety of models and thus find the most cost-effective model. The first approach is useful for the evaluation of the combat ability of a specific fighter aircraft. The second approach conserves calculation time and provides the saddle-point trajectories in just sufficient quality. This is perhaps more useful for trade-off studies in the conceptual design phase of fighter aircraft.

Another interesting possibility would be to apply the results of an optimal feedback strategy to controlling the enemy fighter aircraft in a flight simulator and conduct flight simulation testing. Construction of the opponent's proper strategy may yield qualitative and practical flight simulation testing. In addition, knowledge of the optimal strategies will be useful for on-board integrated fire/flight control computers in the future.

References

1. Isaacs, R., Games of Pursuit, the Rand Corporation, Santa Monica, P-257, 1951.
2. Bryson, A.E. and Ho, Y.C., Applied Optimal Control, Hemisphere, New York, 1975.
3. Ryan III, G.W. and Downing, D.R., "Evaluation of Several Agility Metrics for Fighter Aircraft Using Optimal Trajectory Analysis", Journal of Aircraft, Vol.32, No.4, 1995, pp. 732-738.
4. Well, K.H., Faber, B. and Berger, E., "Optimization of Tactical Aircraft Maneuvers Utilizing High Angle of Attack", Journal of Guidance, Control and Dynamics, Vol.5, No.2, 1982, pp. 131-137.
5. Murayama, K. and Hull, D.G., "The Cobra Maneuver as Minimum Time Problem". AIAA Atmospheric Flight Mechanics Conference, New Orleans, AIAA-97-3586, 1997.
6. Isaacs, R., Differential Games, Wiley, New York, 1965.
7. Breakwell, J.V. and Merz, A.W., "Minimum Required Capture Radius in a Coplanar Model of the Aerial Combat Problem", AIAA Journal, Vol.15, No.8, 1977, pp. 1089-1094.
8. Segal, A. and Miloh, T., "Novel Three-Dimensional Differential Game and Capture Criteria for a Bank-to-Turn Missile", Journal of Guidance, Control and Dynamics, Vol.17, No.5, 1994, pp. 1068-1074.
9. Guelman, M., Shinar, J. and Green, A., "Qualitative Study of a Planar Pursuit Evasion Game in the Atmosphere", Journal of Guidance, Control and Dynamics, Vol.13, No.6, 1990, pp. 1136-1142.
10. Gill, P.E., Murray, W. and Saunders, M.A., User's Guide for SNOPT 5.3: A Fortran Package for Large-Scale Nonlinear Programming, Stanford University, Stanford, 1998.

11. Hillberg, C. and Järmark, B., "Pursuit-Evasion Between Two Realistic Aircraft". AIAA Atmospheric Flight Mechanics Conference, Gatlinburg, AIAA-83-2119, 1983.
12. Breitner, M., Grimm, W. and Pesch, H.J., "Barrier Trajectories of a Realistic Missile/Target Pursuit-Evasion Game", In: R.P.Hamalainen and H.K.Ehtamo, ed. Differential Games: Developments in Modeling and Computation, Springer-Verlag, Berlin, 1991, pp. 48-57.
13. Breitner, M.H., Pesch, H.J. and Grimm, W., "Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 1: Necessary Conditions for Open-Loop Strategies", Journal of Optimization Theory and Applications, Vol.78, No.3, 1993, pp. 419-441.
14. Breitner, M.H., Pesch, H.J. and Grimm, W., "Complex Differential Games of Pursuit-Evasion Type with State Constraints, Part 2: Numerical Computation of Open-Loop Strategies", Journal of Optimization Theory and Applications, Vol.78, No.3, 1993, pp. 443-463.
15. Lachner, R., Breitner, M.H. and Pesch, H.J., "Three-Dimensional Air Combat: Numerical Solution of Complex Differential Games", In: G.J. Olsder, ed. Advances in Dynamic Games and Applications, Birkhäuser, Boston, 1995, pp. 165-190.
16. Raivio, T. and Ehtamo, H., "Visual Aircraft Identification as a Pursuit-Evasion Game", Journal of Guidance, Control and Dynamics, Vol. 23, No. 4, 2000, pp. 701-708.
17. Dickmanns, E.D. and Well, K.H., "Approximate Solution of Optimal Control Problems Using Third-Order Hermite Polynomial Functions", the 6th Technical Conference on Optimization Techniques, Springer-Verlag, New York, IFIP-TC7, 1975.
18. 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, 1987, pp. 338-342.
19. Enright, P.J. and Conway, B.A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming", Journal of Guidance, Control and Dynamics, Vol.14, No.5, 1991, pp. 981-985.

20. Herman, A.L. and Conway, B.A., "Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules", Journal of Guidance, Control and Dynamics, Vol.19, No.3, 1996, pp. 592-599.

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

22. Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H., User's Guide for NPSOL (Version 4.0): A Fortran Package for Nonlinear Programming, Stanford University, Stanford, SOL 86-2, 1986.

23. Raivio, T. and Ehtamo, H., "On Numerical Solution of a Class of Pursuit-Evasion Games", Annals of the International Society of Dynamic Games, Vol. 5, 2000, Birkhäuser, Boston, pp. 177-192.

24. Gill, P.E., Saunders, M.A. and Murray,W., User's Guide for NZOPT 1.0: A Fortran Package For Nonlinear Programming, McDonnell Douglas Aerospace, Huntington Beach, 1993.

25. Vinh, N.X., Optimal Trajectories in Atmospheric Flight, Elsevier, Amsterdam, Netherland. 1981.

26. McCormick, B.W., Aerodynamics, Aeronautics and Flight Mechanics, John Wiley & Sons. New York, 1994.

27. Zaginov,G., "High Maneuverability. Theory and Practice", AIAA 8th Aircraft Design, Systems and Operations Meeting, Monterey, AIAA-93-4737, 1993.

28. Basar, T. and Olsder, G. J., Dynamic Noncooperative Game Theory, SIAM, Philadelphia, 1999.

29. Goldberg, D., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, 1989.

30. Carroll, D.L., Fortran Genetic Algorithm (GA) Driver, <http://www.staff.uiuc.edu/~carroll/ga.html>, September 30, 1998.

31. Smith, R.E. and Dike, B.A., "An Application of Genetic Algorithms to Air Combat Maneuvering", Handbook of Evolutionary Computation, Oxford University Press, 1997.
32. Hartmann, J.W., Coverstone-Carroll, V. and Williams, S.N., "Optimal Interplanetary Spacecraft Trajectories via a Pareto Genetic Algorithm", Proceedings of AAS/AIAA Space Flight Mechanics Meeting, Monterey, AAS-98-202, 1998.
33. Heath, M. T., Scientific Computing: An Introductory Survey, McGraw-Hill, New York, 1996.
34. Goldberg, D. E., and Voessner, S., "Optimizing Global-Local Search Hybrids", Proceedings of the Genetic and Evolutionary Computation Conference 1999 (GECCO-99), Morgan Kaufmann, San Francisco, 1999.
35. Hinton, G.E. and Nowlan, S.J., "How Learning can Guide Evolution", Complex Systems, Vol.1, 1987, pp. 495-502.
36. Miele, A., Flight Mechanics, Volume I, Theory of Flight Paths, Addison-Wesley, Reading, 1962.
37. Gilbert, W.P., Nguyen, L.T. and Van Gunst, R.W., Simulator Study of the Effectiveness of an Automatic Control System Designed to Improve the High-Angle-of-Attack Characteristics of a Fighter Airplane, Langley Research Center, NASA TN D-8176, 1976.

Vita

Kazuhiro Horie is a graduate student at the University of Illinois at Urbana-Champaign and has recently finished his Ph.D. thesis research. He graduated with B.Engr. degree from Nagoya University, Japan, in March, 1987. After one year studying as a graduate student, he left Nagoya University and joined the Technical Research and Development Institute (TRDI), Japan Defense Agency. He took a position in a section of flight mechanics and flight control, Third Research Center, TRDI. While in this position, he was in charge of development of engineering models for TFMS, tactical flight management system, and SRFCS, self-repairing flight control system, in addition to joining two developments: flight testing of the XSH-60J, anti-submarine helicopter, and design of flight control law of the FS-X (current F-2), Japanese new jet fighter. In August 1994, he was re-assigned to FS-X development office, TRDI as a staff engineer. He was the coordinator in review team for aerodynamics and flight control system of FS-X and the primary working-level negotiator with US government. In August 1996, he was selected in TRDI abroad research program and began graduate study at University of Illinois at Urbana-Champaign. In January 1998, he received M.S. degree and identified research topic for Ph.D. thesis research. Although he left USA and was back to Japan in 1999, he continued his research while working for TRDI. After finishing his graduate work, he will continue his current assignment in TRDI, senior research scientist, aircraft systems section, Third Research Center, and focus on research on flight mechanics.