New techniques in trajectory optimization and guidance

by

Mohammed Asif Khan

A Thesis Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

Department: Aerospace Engineering and Engineering Mechanics
Major: Aerospace Engineering

Iowa State University
Ames, Iowa
1992

In the name of Allah, Most Gracious, Most Merciful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my deep gratitude and sincere thanks to my major professor, Dr. Ping Lu. I am very grateful for his guidance and encouragement, discussions and suggestions, and his friendship. I would also like to thank Dr. Bion L. Pierson and Dr. Ron M. Nelson for serving on my committee. Thanks are also due to Laith Zori and Nick Thorp for their help through all stages of this work. Most of all, I would like to give special thanks to my family for their support and encouragement.

AIRCRAFT NOMENCLATURE

b	wing span, ft
C	force or moment coefficient
C_D	coefficient of drag
C_L	coefficient of lift
C_l	coefficient of rolling moment
C_m	coefficient of pitching moment
C_n	coefficient of yawing moment
C_y	coefficient of sideforce
$\bar{c}$	mean chord, ft
D	drag force, lb
g	acceleration due to gravity, ft/s^2
h	altitude, ft
h_f	final altitude, ft
h_i	initial altitude, ft
h_{min}	minimum altitude, ft
I	aircraft inertia tensor, $slug/ft^2$
I_x,I_y,I_z	moments of inertia, $slug/ft^2$
I_{xz},I_{xy},I_{yz}	products of inertia, $slug/ft^2$

L	lift force, lb
m	total aircraft mass, $slug$
P_a	percentage afterburner
P_c	percentage core engine
p	roll rate, deg/s
q	pitch rate, deg/s
q_f	final pitch rate, deg/s
r	yaw rate, deg/s
S	wing area, ft^2
T	thrust per each engine, lb
T_{idl}	idle thrust, lb
T_{max}	maximum thrust, lb
T_{mil}	military thrust, lb
t	time, s
t_f	time-of-flight, s
V	velocity, ft/s
W	weight, lb
X_b	aircraft x-body axis
X_T	total thrust force along x-body axis, lb
x	downrange distance, ft
Y	sideforce, lb
Y_b	aircraft y-body axis
Y_T	total thrust force along y-body axis, lb
z	cross-range distance, ft

Z_b	aircraft z-body axis
Z_T	total thrust force along z-body axis, lb

Greek Symbols

α	angle of attack, deg
β	angle of sideslip, deg
γ	flight path angle, deg
γ_f	final flight path angle, deg
δ_A	aileron deflection, deg
δ_D	differential stabilator deflection, deg
δ_H	symmetric stabilator deflection, deg
δ_R	rudder deflection, deg
δ_T	throttle setting, deg
θ	pitch angle, deg
ΣL	total body axis rolling moment, $ft - lb$
ΣM	total body axis pitching moment, $ft - lb$
ΣN	total body axis yawing moment, $ft - lb$
ϕ	roll angle, deg
ϕ_f	final roll angle, deg
ψ	heading angle, deg
ψ_f	final heading angle, deg

CHAPTER 1. INTRODUCTION

1.1 Background

In the past several decades trajectory optimization has become an extensively researched field. Some of the most famous work has been done in the context of minimum-time trajectories such as minimum time-to-climb and minimum time-to-turn problems. Well known work in this area includes Bryson's and Denham's [1] development of a gradient based steepest ascent method which is used to obtain the minimum time-to-climb trajectory of a supersonic fighter. Other researchers have applied a variety of techniques such as singular perturbation methods [2] and sequential quadratic programming [3]. Optimal trajectories have also been successfully obtained by Hargraves et al. [4] using a collocation method in which discretized control and state time histories are represented by Chebychev polynomials. This entire spectrum of work however is limited to the use of point-mass aircraft models. Only recently, Stalford and Hoffman [5] have computed optimal trajectories for a rigid-body dynamic aircraft model. The applicability of this research is still restricted to aircraft models that are defined by smooth analytical expressions. As most aerodynamic and propulsion data commonly available for an aircraft are from experimental sources and are in tabulated form, there is a need for a method of performing trajectory optimization for such non-smooth models. This study addresses the aforementioned problem

by presenting a robust optimization technique that is insensitive to the complexity of the aircraft model.

1.2 Trajectory Optimization Method

Historically, optimal trajectories generated for practical implementation [6] are obtained using gradient based algorithms. In all realistic situations, no analytical aircraft model is available. Consequently gradient algorithms fail to give solutions as the gradient information is usually inaccurate or simply unattainable. Most non-gradient search methods are also inadequate since their results are often initial guess dependent and are only locally optimal. As many practical aerospace trajectory optimization problems are inherently non-smooth, there exists a need for a technique that can solve these problems. To address this requirement, the current study proposes the use of a newly developed continuous simulated annealing algorithm. The algorithm is successfully used to produce minimum time-to-climb, minimum time-to-half-loop and minimum time-to-turn trajectories. A number of other algorithms applied to these problems failed to yield solutions. The simulated annealing method is described in detail in Chapter 2.

1.3 Guidance Law Development

A new technique for developing guidance laws is tested on a complex aircraft model. The model was made available by the American Institute of Aeronautics and Astronautics and was originally intended for use in the 1991–92 Controls Design Challenge [7]. Though this study is not intended to directly participate in the challenge, a substantial amount of guidance law development is carried out. These guidance

laws meet a number of the requirements of the design challenge. In addition, they provide controllers for tracking the optimal trajectory solutions in the presence of disturbances. The controllers are nonlinear in state but assume linearity in control. A detailed account of the guidance law development and implementation is given in Chapter 5.

CHAPTER 2. SIMULATED ANNEALING

Simulated annealing is a class of stochastic algorithms which search for global optimal solutions. The algorithms simulate the physical phenomenon of annealing. The relationship between simulated and physical annealing and the development of the specific algorithm used in this study is described in the following sections.

2.1 Annealing

Simulated annealing algorithms were originally developed for use on combinatorial optimization problems. A combinatorial optimization problem is one in which the design vector has finite or countably infinite configurations [11]. The simulated annealing algorithm is essentially an imitation of physical annealing. Physical annealing is the process by which a solid is first heated until it melts, and then gradually cooled until it crystallizes into a state with a perfect lattice. The final state this process attains is the configuration which minimizes the free energy of the solid. This final state can be viewed as the solution to a combinatorial optimization problem in that the annealing process has found a state, from a vast selection of configurations, that has the minimum free energy.

Annealing is based on the principle that at any given temperature T, free atoms occupy a state which satisfies the Boltzmann's probability distribution. This distri-

bution can be expressed as the probability of a system being in a state r is given by,

$$P(r) \sim e^{-E(r)/k_b T} \tag{2.1}$$

where $E(r)$ is the magnitude of the energy associated with state r, and k_b is the Boltzmann's constant [12]. Hence, at equilibrium, the most probable state of the system is that of the lowest energy. Simulated annealing uses this principle of physical annealing to solve optimization problems. The following is an outline of a generic simulated annealing algorithm.

1. Choose an initial design vector.
2. Generate a candidate next iteration vector by some random process.
3. If the candidate vector is an improvement in the cost function, accept it as the next vector.
4. If the candidate vector is a deterioration of the cost function, apply the Metropolis criterion to determine if it is accepted as the next vector. If it is not accepted, the current design vector is retained.
5. Lower the temperature parameter according to a pre-determined cooling schedule.
6. If the termination criteria has not been met, return to step 2.

Simulated annealing is a general optimization methodology rather than a completely specified algorithm. The distinguishing characteristic between specific annealing algorithms is, in general, the choice of the cooling schedule and the process by

Table 2.1: Corresponding terms in physical and simulated annealing

Physical annealing	Simulated annealing
State	Design vector
Energy	Cost function
Boltzmann's distribution	Metropolis criterion
Minimum energy state	Optimal solution

which candidate vectors are generated. The specific algorithm used in this study is described further on in this chapter. To illustrate the connection between physical and simulated annealing, the corresponding terms of the two annealing processes are shown in Table 2.1. The Metropolis criterion referred to in the algorithm and the table is defined in a following section.

2.2 Development and Past Applications

Simulated annealing has received wide attention from researchers since being presented by Kirkpatrick et al. in 1983 [13]. Kirkpatrick applies a discrete multivariate or combinatorial optimization algorithm to design an integrated circuit and to solve the famous *traveling salesman problem.* The *traveling salesman problem* is to find the shortest itinerary for a traveling salesman who must visit each of N cities in turn. The dimension of this problem is extraordinary as for just 20 cities the number of feasible tours is an 18-digit number [14]. Furthermore, since a configuration is discrete, there is no notion of a gradient or down-hill direction. Kirkpratrick efficiently used a simulated annealing algorithm to solve this problem for 400 cities. Discrete annealing has been applied to many optimization problems with varying degrees of

success. Among them is telephone network design, test pattern generation, logic minimization and image restoration. The methodology is also being extensively used in the field of statistics.

Most applications to date have been restricted to combinatorial optimization problems. For trajectory optimization purposes, a continuous optimization method is required. Formulations of continuous simulated annealing algorithms have been proposed by various authors; however, in most cases they lack solid theoretical foundation. In this study, a new continuous simulated annealing algorithm proposed by Romeijn and Smith [15], is used. This algorithm is called Hide-and-Seek and is proven to converge in probability to global optimal solutions.

2.3 Hide-and-Seek

Hide-and-Seek is an efficient and easily implemented continuous simulated annealing algorithm [16]. It is capable of finding the global maximum of a continuous function which may have many locally optimal solutions. This algorithm has two distinguishing features: a continuous random walk process and an adaptive cooling schedule. These features are unique to this annealing algorithm. Hide-and-Seek is very simple to apply as it only requires a bounded design space and a method for evaluating the cost function for a given feasible design vector. Within the bounded space any design vector can be declared infeasible. This allows the easy enforcement of inequality constraints.

The Hide-and-Seek algorithm proceeds as follows:

1. Choose a very high (positive) initial value for the temperature parameter.

2. Generate a random initial feasible design vector X_0 and set $k = 0$.

3. Choose a direction vector Θ_k from a uniform direction distribution.

4. Choose λ_k from the uniform distribution of feasible points along the direction vector Θ_k.

5. Define the candidate for the next iteration vector by $Y_{k+1} = X_k + \lambda_k \Theta_k$.

6. Calculate the acceptance probability function β_k by applying the Metropolis criterion.

7. With probability β_k, set $X_{k+1} = Y_{k+1}$. Otherwise, set $X_{k+1} = X_k$.

8. If the objective function value of X_{k+1} is greater than any previous objective function, decrease the temperature parameter according to the cooling schedule.

9. If the termination criteria is not met, set $k = k + 1$ and go to step 3.

The Metropolis criterion [17] determines the acceptance of a candidate next iteration vector. It is given by,

$$\beta_k = min\{\ 1.0\ ,\ e^{(f(Y_{k+1})-f(X_k))/T}\} \tag{2.2}$$

$$X_{k+1} = \begin{cases} Y_{k+1} & if \quad V_k \leq \beta_k \\ X_k & if \quad V_k > \beta_k \end{cases} \tag{2.3}$$

where V_k is a random variable with a uniform distribution on [0,1]. It should be noted that from Equation 2.2, even if the candidate vector represents a deterioration in the objective function (i.e., $f(Y_{k+1}) < f(X_k)$), the initial high temperature parameter

results in a high probability of acceptance. This acceptance probability for deteriorations decreases as the algorithm progresses, due to the decrease in the temperature parameter. Additionally, all objective function improvements are accepted, as in such cases $\beta_k = 1.0$.

The temperature parameter is monotonically decreased according to an adaptive cooling schedule given by Equation 2.4. The temperature update only occurs when the objective function for X_{k+1} is greater than all previous objective functions. This cooling schedule allows the temperature to decrease according to the real-time progress of the algorithm.

$$T = \frac{2(f^* - f(X_{k+1}))}{\chi^2_{1-p}(n)} \tag{2.4}$$

where n is number of design variables, $\chi^2_{1-p}(n)$ is the $100(1-p)$ percentile point of the chi-squared distribution with n degrees of freedom, and f^* is the optimal objective function value. $\chi^2_{1-p}(n)$ is a constant for a given problem. As for most practical problems f^* is not known in advance, the authors of Hide-and-Seek have derived the following heuristic estimator ($\hat{f}$).

$$\hat{f} = f_1 + \frac{f_1 - f_2}{(1-p)^{\frac{-n}{2}} - 1} \tag{2.5}$$

where f_1 and f_2 are the current largest two objective functions, respectively. The parameter p corresponds to the probability that $\hat{f}$ will be less than f^*. As $f_1 > f_2$, $\hat{f}$ will always be greater than the current best objective function value. This assures a monotonically increasing $\hat{f}$ that in the limit will approach f^*.

The algorithm is terminated when for an $\epsilon > 0$,

$$\hat{f} - f_1 < \epsilon \tag{2.6}$$

This termination criteria will always be met as in the limit, f_1 approaches $\hat{f}$.

2.4 Testing Hide-and-Seek

To verify the global convergence of Hide-and-Seek, it is tested on three stringent problems. The test problems used are the Beale function [18], the Goldstein/Price (GP) function [16], and a function referred to in this study as TF [19]. The Beale function has multiple local maxima and a unique global maximum. It is a two-variable function given by,

$$\begin{aligned} f(x_1,x_2) &= -\Big((1.5-x_1(1-x_2))^2+(2.25-x_1(1-x_2^2))^2 \\ &\quad +(2.625-x_1(1-x_2^3))^2\Big) \end{aligned} \tag{2.7}$$

The Goldstein/Price function also has two independent arguments. It has three local and one global maxima over the interval $-2 \leq x_i \leq 2$. It is defined by Equation 2.8.

$$\begin{aligned} f(x_1,x_2) &= -0.1\Big(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2 \\ &\quad -14x_2+6x_1x_2+3x_2^2)\Big)\Big(30+(2x_1-3x_2)^2\cdot \\ &\quad (18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\Big) \end{aligned} \tag{2.8}$$

TF is a variable dimension function. Over the interval $-10 \leq x_i \leq 10$, $i = 1,..,n$, it has approximately 10^n local maxima, one of which is global. The function is tested using two, three, and four variables. A surface plot of TF with two variables is shown in Figure 2.1. TF is defined by the following equation.

$$\begin{aligned} f(x_1,x_2,...,x_n) &= -\frac{\pi}{n}\Big\{10sin(\pi x_1)^2+ \\ &\quad +\sum_{i=1}^{n-1}(x_i-1)^2\left[1+10sin(\pi x_{i+1})^2\right] \\ &\quad +(x_n-1)^2\Big\} \end{aligned} \tag{2.9}$$

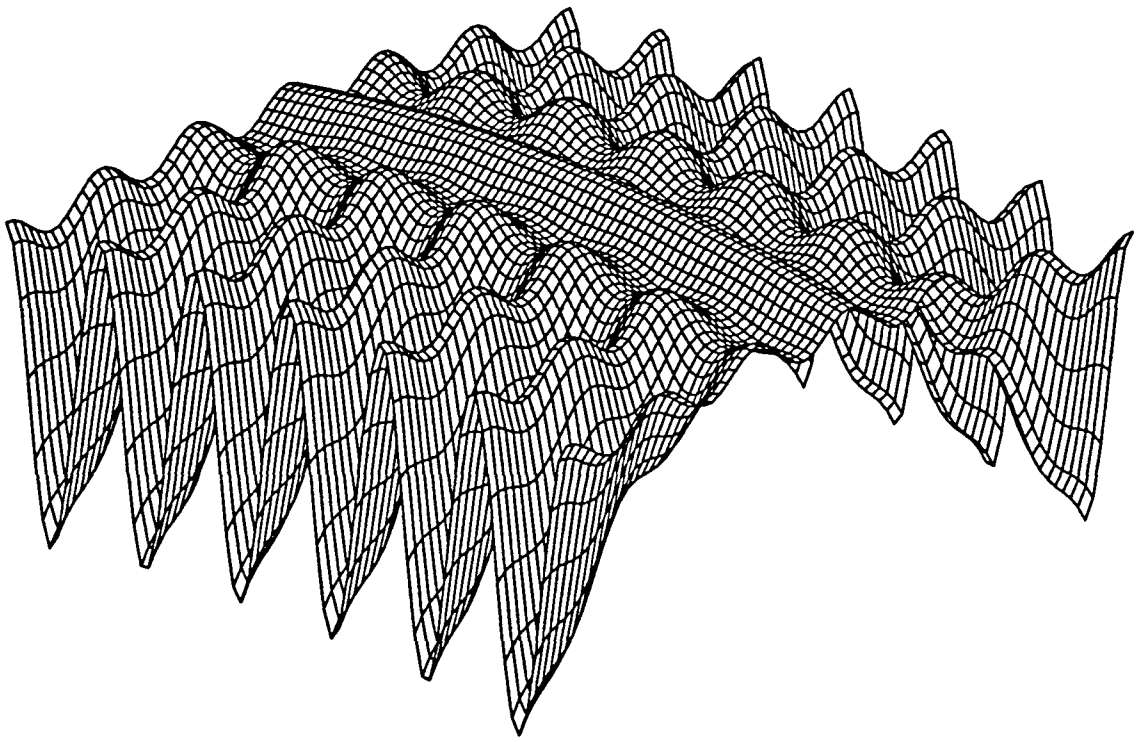

Figure 2.1: Test function TF

Table 2.2: The number of function evaluations for Hide-and-Seek and Hybrid

Function	Hide-and-Seek with $\hat{f}$	Hide-and-Seek with f^*	Hybrid with $\hat{f}$	Hybrid with f^*
Beale	1,131	494	1,445	528
GP	2,536	548	5,975	662
TF(2)	11,213	3,063	11,581	3,093
TF(3)	13,738	5,191	14,041	5,248
TF(4)	6,616	7,663	7,788	8,727

The performance of Hide-and-Seek is compared against the Nelder-Mead Simplex algorithm [22] and the Principal Axis method [21]. Both techniques are well-known non-gradient local search methods. The Principal Axis method is a modified version of Powell's method. In addition, a hybrid routine ("Hybrid") using Hide-and-Seek for the initial global search and Nelder-Mead Simplex for the final local search is tested.

The Principal Axis and Nelder-Mead Simplex techniques failed to converge on the global solutions for most initial guess vectors. The global solutions were only found when an extremely 'good' initial guess was given. The test results from Hide-and-Seek and the Hybrid method are given in Table 2.2. The number of function evaluations referred to in Table 2.2 is an average over ten runs. The Hybrid method found the same solutions as Hide-and-Seek but converged to a higher degree of accuracy. This, however, is at a higher cost in terms of function evaluations. The test results show various beneficial aspects of Hide-and-Seek which are discussed in detail in the following section.

2.5 Assessing Hide-and-Seek

The test results and the earlier description of Hide-and-Seek make evident a number of advantages and disadvantages of this method. The routine is robust as it can be applied to highly complex functions with a reasonable assurance that it will find the global solution. The functions can also be discontinuous in gradient and value as no gradient information is used. If the optimal function value f^* is known, the test results show that the method can use this information to rapidly converge on the solution. In addition, given f^*, the number of function evaluations required increases approximately linearly with the dimension of the problem. Even if the estimator is used, the function evaluations do not rapidly increase with the number of design variables. Unlike most other non-gradient search methods, Hide-and-Seek demands no initial guess. The only inputs it requires are a termination tolerance and lower and upper bounds on the design variables. As any design vector within this bounded space can be declared infeasible, inequality constraints can effortlessly be enforced. Equality constraints on the other hand, can only be sanctioned using penalty functions or variable transformations. The authors of Hide-and-Seek [16] have shown that this technique is insensitive to the complexity of the feasible region. Some drawbacks of Hide-and-Seek are that it is expensive in terms of function evaluations and its convergence rate does not increase in the proximity of the optimal solution.

For aerospace engineering, the extensive advantages of Hide-and-Seek make it an attractive method for obtaining optimal trajectories for complex aircraft models. This problem is undertaken in Chapter 4.

CHAPTER 3. AIRCRAFT MODEL

The aircraft model used in this study was primarily developed for use in the 1991–92 AIAA Controls Design Challenge [7]. The model is a generic, state-of-the-art, high-performance aircraft. It has nonlinear aerodynamics and propulsion, defined over the entire operational flight envelope of the aircraft. Though it contains certain aspects of existing fighters, it is in no way a representation of a specific aircraft. It is rather an amalgamation of various aircraft components. The model has six-degrees-of-freedom and is capable of flight in the high angle-of-attack regime. A full three-dimensional description of the model is given in this chapter.

3.1 Aircraft Characteristics

The configuration of the aircraft is shown in Figure 3.1. The aircraft is representative of a modern-day, high-performance, supersonic fighter. It weighs $45,000\ lb$ and has a wing area of $608\ ft^2$. In dimensions, it is approximately the same size as the F-15 Eagle. Details of the aircraft characteristics are given in Table 3.1. The primary control surfaces of the aircraft are two stabilators, two ailerons, and a rudder on the single vertical tail. The aircraft has two engines and has a maximum speed of Mach 2.5. The absolute ceiling of the aircraft is approximately $60,000\ ft$.

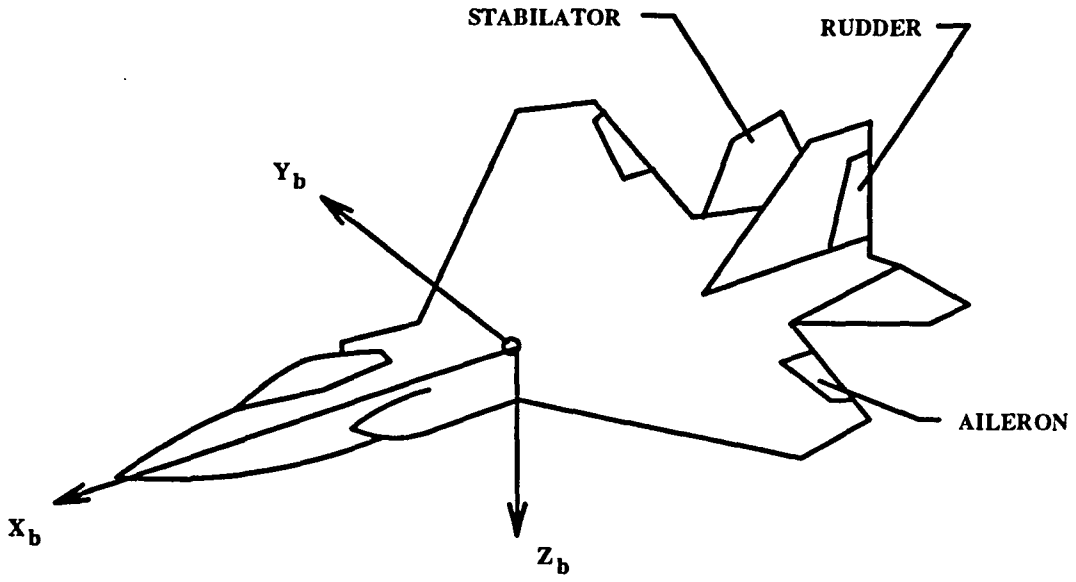

Figure 3.1: Aircraft configuration

Table 3.1: Aircraft model dimensions

Attribute	Symbol	Value
Weight	W	$45,000.0\ lb$
Wing area	S	$608.0\ ft^2$
Wing span	b	$42.8\ ft$
Mean chord	$\bar{c}$	$15.95\ ft$
Moments of inertia	I_x	$28,700.0\ slug/ft^2$
	I_y	$165,100.0\ slug/ft^2$
	I_z	$187,900.0\ slug/ft^2$
Products of inertia	I_{xz}	$-520.0\ slug/ft^2$
	I_{xy}	$0.0\ slug/ft^2$
	I_{yz}	$0.0\ slug/ft^2$

3.2 Propulsion Model

The aircraft has two turbofan engines equipped with afterburners. Each engine is capable of producing a maximum thrust of approximately $30,000\ lb$. The engines produce an identical thrust for a given throttle setting. This results in there being no yawing moment due to the propulsion system. The thrust acts along a vector parallel to the X-body axis and at point $10\ ft$ behind the aircraft center of gravity and $4\ ft$ lateral of the vertical plane of symmetry. At a given altitude and Mach number, the engine has three thrust settings: the idle thrust (T_{idl}), the military thrust (T_{mil}) and the maximum (afterburner) thrust (T_{max}) . The magnitude of these thrust settings is obtained by interpolating tabular data. The thrust produced by each engine is a linear function of the throttle position (δ_T) and is given by Equations 3.1 and 3.2. The throttle has range of 20° to 135° with the afterburner engaged beyond 83°.

If $20° \leq \delta_T \leq 83°$,

$$\begin{aligned} P_c &= (\delta_T - 20.0)/63.0 \\ T &= (T_{mil} - T_{idl})\ P_c + T_{idl} \end{aligned} \tag{3.1}$$

If $83° < \delta_T \leq 135°$,

$$\begin{aligned} P_a &= (\delta_T - 83.0)/44.0 \\ T &= (T_{max} - T_{mil})\ P_a + T_{mil} \end{aligned} \tag{3.2}$$

As no fuel flow rates are available for the engines, the fuel consumed during flight cannot be properly ascertained. For this reason, the aircraft is assumed to have a constant weight of $45,000\ lb$.

Table 3.2: Control surface limits and sign conventions

Control surface	Symbol	Limit (°)	Positive deflection
Symmetric stabilator	δ_H	$+15/-25$	Trailing edge down
Differential stabilator	δ_D	± 20	Left trailing edge down
Aileron	δ_A	± 20	Left trailing edge down
Rudder	δ_R	± 30	Trailing edge left

3.3 Aerodynamic Model

The aircraft is equipped with five aerodynamic control surfaces. It has two stabilators capable of symmetric and differential motion, two conventional ailerons and a single rudder. The sign convention for the displacement of the surfaces and their position limits are shown in Table 3.2.

The aircraft has nonlinear aerodynamic coefficients that are computed by performing multi-dimensional, linear interpolation of tabular data. This interpolation in general is dependent on the current Mach number, the angle of attack, the slideslip angle, and the symmetric stabilator deflection. The longitudinal parameters are given with respect to the stability axis, while the lateral-directional parameters are referenced to the aircraft body axis. The following equations define the aerodynamic coefficients:

$$C_L = C_{L_{BASIC}} \tag{3.3}$$

$$C_m = C_{m_{BASIC}} + \frac{\bar{c}}{(2V)}(C_{m_q} q + C_{m_{\dot{\alpha}}} \dot{\alpha}) \tag{3.4}$$

$$C_D = C_{D_{BASIC}} \tag{3.5}$$

Table 3.3: Aerodynamic coefficients

Coefficient	Source	Function of
$C_{L_{BASIC}}$	Table	M, α, δ_H
$C_{m_{BASIC}}$	Table	M, α, δ_H
C_{m_q}	Table	M, α
$C_{m_{\dot{\alpha}}}$	Table	M, α
$C_{D_{BASIC}}$ $(\alpha < 32)$	Table	$C_{L_{BASIC}}, M$
$(32 < \alpha < 40)$	Table	$C_{L_{BASIC}}, M, \alpha$
$(\alpha > 40)$	Calc	$C_{L_{BASIC}}, \alpha$
$C_{y_{BASIC}}$	Table	M, α, β
$C_{y_{\delta_A}}$	Table	M, α
$C_{y_{\delta_D}}$	Table	M, α
$\Delta C_{y_{\delta_R}}$	Table	M, α, δ_R
$K_{\delta_{R_y}}$	Table	M
$C_{l_{BASIC}}$	Table	M, α, β
$C_{l_{\delta_A}}$	Table	M, α
$C_{l_{\delta_D}}$	Table	M, α
$\Delta C_{l_{\delta_R}}$	Table	M, α, δ_R
$K_{\delta_{R_l}}$	Table	M
C_{l_p}	Table	M, α
C_{l_r}	Table	M, α
$C_{n_{BASIC}}$	Table	M, α, β
$C_{n_{\delta_A}}$	Table	M, α
$C_{n_{\delta_D}}$	Table	M, α
$\Delta C_{n_{\delta_R}}$	Table	M, α, β
$K_{\delta_{R_n}}$	Table	$M.\alpha$
C_{n_p}	Table	M, α
C_{n_r}	Table	M, α

$$C_y = C_{y_{BASIC}} + C_{y_{\delta_A}}\delta_A + C_{y_{\delta_D}}\delta_D - \Delta C_{y_{\delta_R}} K_{\delta_{R_y}} \tag{3.6}$$

$$C_l = C_{l_{BASIC}} + C_{l_{\delta_A}}\delta_A + C_{l_{\delta_D}}\delta_D - \Delta C_{l_{\delta_R}} K_{\delta_{R_l}} + \frac{b}{(2V)}(C_{l_p}p + C_{l_r}r) \tag{3.7}$$

$$C_n = C_{n_{BASIC}} + C_{n_{\delta_A}}\delta_A + C_{n_{\delta_D}}\delta_D + \Delta C_{n_{\delta_R}} K_{\delta_{R_n}} + \frac{b}{(2V)}(C_{n_p}p + C_{n_r}r) \tag{3.8}$$

$$\tag{3.9}$$

Table 3.3 shows the source of the aerodynamic coefficients on the right-hand side of the equations.

3.4 Equations of Motion

The aircraft dynamics are defined by nonlinear six-degree-of-freedom equations of motion. The equations assume a rigid aircraft flying over a non-rotating flat Earth, in a stationary atmosphere. They are given with respect to the aircraft wind-axes and are detailed in Etkin [8] and Blakelock [9]. The following equations define the derivative of each state variable.

The translational acceleration equations are,

$$\begin{aligned}\dot{V} = {} & [-D\ cos\beta + Y\ sin\beta + X_T\ cos\alpha\ cos\beta + Y_T\ sin\beta \\ & + Z_T\ sin\alpha\ cos\beta - mg(sin\theta\ cos\alpha\ cos\beta - cos\theta\ sin\phi\ sin\beta \\ & - cos\theta\ cos\phi\ sin\alpha\ cos\beta)]/m\end{aligned} \tag{3.10}$$

$$\begin{aligned}\dot{\alpha} = {} & [-L + Z_T\ cos\alpha - X_T\ sin\alpha + mg(cos\theta\ cos\phi\ cos\alpha \\ & + sin\theta\ sin\alpha)]/Vm\ cos\beta + q - tan\beta\ (p\ cos\alpha + r\ sin\alpha)\end{aligned} \tag{3.11}$$

$$\begin{aligned}\dot{\beta} = {} & [D\ sin\beta + Y\ cos\beta - X_T\ cos\alpha\ sin\beta + Y_T\ cos\beta \\ & - Z_T\ sin\alpha\ sin\beta + mg(sin\theta\ cos\alpha\ sin\beta + cos\theta\ sin\phi\ cos\beta\end{aligned} \tag{3.12}$$

$$-cos\theta\ cos\phi\ sin\alpha\ sin\beta)]/Vm + p\ sin\alpha - r\ cos\alpha$$

The vehicle attitude rates are,

$$\dot{\theta} = q\ cos\phi - r\ sin\phi \tag{3.13}$$

$$\dot{\psi} = q\ sin\phi\ sec\theta + r\ cos\phi\ sec\theta \tag{3.14}$$

$$\dot{\phi} = p + q\ sin\phi\ tan\theta + r\ cos\phi\ tan\theta \tag{3.15}$$

The Earth-relative velocities are,

$$\begin{aligned}\dot{h} &= V(cos\beta\ cos\alpha\ sin\theta - sin\beta\ sin\phi\ cos\theta \\ &\quad -cos\beta\ sin\alpha\ cos\phi\ cos\theta)\end{aligned} \tag{3.16}$$

$$\begin{aligned}\dot{x} &= V[cos\beta\ cos\alpha\ cos\theta\ cos\psi + sin\beta\ (sin\phi\ sin\theta\ cos\psi \\ &\quad -cos\phi\ sin\psi) + cos\beta\ sin\alpha\ (cos\phi\ sin\theta\ cos\psi \\ &\quad +sin\phi\ sin\psi)]\end{aligned} \tag{3.17}$$

$$\begin{aligned}\dot{y} &= V[cos\beta\ cos\alpha\ cos\theta\ sin\psi + sin\beta\ (sin\phi\ sin\theta\ sin\psi \\ &\quad +cos\phi\ cos\psi) + cos\beta\ sin\alpha\ (cos\phi\ sin\theta\ sin\psi \\ &\quad -sin\phi\ cos\psi)]\end{aligned} \tag{3.18}$$

The rotational accelerations are,

$$\begin{aligned}\dot{p} &= [(\Sigma L)I_1 + (\Sigma M)I_2 + (\Sigma N)I_3 - p^2(I_{xz}I_2 - I_{xy}I_3) \\ &\quad +pq(I_{xz}I_1 - I_{yz}I_2 - D_zI_3) - pr(I_{xy}I_1 + D_yI_2 - I_{yz}I_3)\end{aligned} \tag{3.19}$$

$$
\begin{aligned}
&+q^2(I_{yz}I_1 - I_{xy}I_3) - qr(D_xI_1 - I_{xy}I_2 + I_{xz}I_3) \\
&-r^2(I_{yz}I_1 - I_{xz}I_2)]/det\ I \\
\dot{q} \;=\; &[(\Sigma L)I_2 + (\Sigma M)I_4 + (\Sigma N)I_5 - p^2(I_{xz}I_4 - I_{xy}I_5) \qquad (3.20) \\
&+pq(I_{xz}I_2 - I_{yz}I_4 - D_zI_5) - pr(I_{xy}I_2 + D_yI_4 - I_{yz}I_5) \\
&+q^2(I_{yz}I_2 - I_{xy}I_5) - qr(D_xI_2 - I_{xy}I_4 + I_{xz}I_5) \\
&-r^2(I_{yz}I_2 - I_{xz}I_4)]/det\ I \\
\dot{r} \;=\; &[(\Sigma L)I_3 + (\Sigma M)I_5 + (\Sigma N)I_6 - p^2(I_{xz}I_5 - I_{xy}I_6) \qquad (3.21) \\
&+pq(I_{xz}I_3 - I_{yz}I_5 - D_zI_6) - pr(I_{xy}I_3 + D_yI_5 - I_{yz}I_6) \\
&+q^2(I_{yz}I_3 - I_{xy}I_6) - qr(D_xI_3 - I_{xy}I_5 + I_{xz}I_6) \\
&-r^2(I_{yz}I_3 - I_{xz}I_5)]/det\ I
\end{aligned}
$$

Where,

$$
\begin{aligned}
det\ I &= I_xI_yI_z - 2I_{xy}I_{xz}I_{yz} - I_xI_{yz}^2 - I_yI_{xz}^2 - I_zI_{xy}^2 && (3.22) \\
I_1 &= I_yI_z - I_{yz}^2 && (3.23) \\
I_2 &= I_{xy}I_z + I_{yz}I_{xz} && (3.24) \\
I_3 &= I_{xy}I_{yz} + I_yI_{xz} && (3.25) \\
I_4 &= I_xI_z - I_{xz}^2 && (3.26) \\
I_5 &= I_xI_{yz} + I_{xy}I_{xz} && (3.27) \\
I_6 &= I_xI_y - I_{xy}^2 && (3.28) \\
D_x &= I_z - I_y && (3.29) \\
D_y &= I_x - I_z && (3.30)
\end{aligned}
$$

$$D_z = I_y - I_x \tag{3.31}$$

During a trajectory simulation, the dynamic equations are integrated using a fixed step 4th-order Runge-Kutta integration routine [10].

The atmospheric parameters used by the model are based on linear interpolation of tables from the U.S. Standard Atmosphere (1962).

3.5 Modifications

The following is a list of the major modifications made to the original model [7]. In general, these changes were made to simplify the guidance law development and to allow the use of a larger integration step-size during trajectory simulations.

1. Actuators attached to all the control surfaces are removed, however the deflection limits are retained. The actuators had time constants of 0.05 s and rate limits of 24 $°/s$.

2. First-order dynamics in the throttle setting is deleted.

3. The rate limiter and sequencing logic for the afterburner are removed.

4. Additional terms in the aerodynamic coefficients that are functions of altitude and engine thrust are dropped. These additional terms had a negligible effect on the overall magnitude of the coefficients.

5. The upper limit of the throttle range is extended from 127° to 135°.

CHAPTER 4. TRAJECTORY OPTIMIZATION

In this chapter, minimum-time optimization problems for the aircraft model are formulated. These problems are parameterized and solved using Hide-and-Seek. The resulting optimal trajectories and control histories are also presented.

4.1 Problem Statement

A general optimal control problem can be stated as follows.

Find the input $u(t)$, over the interval $t_0 \leq t \leq t_f$, which minimizes the cost function:

$$J = \Phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t), t)dt \tag{4.1}$$

subject to a nonlinear time-varying dynamic plant,

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

given initial states and terminal constraints,

$$x(t_0) = x_0 \quad , \quad x(t_f) = x_f \tag{4.3}$$

and inequality control constraints,

$$u_{min} \leq u(t) \leq u_{max} \tag{4.4}$$

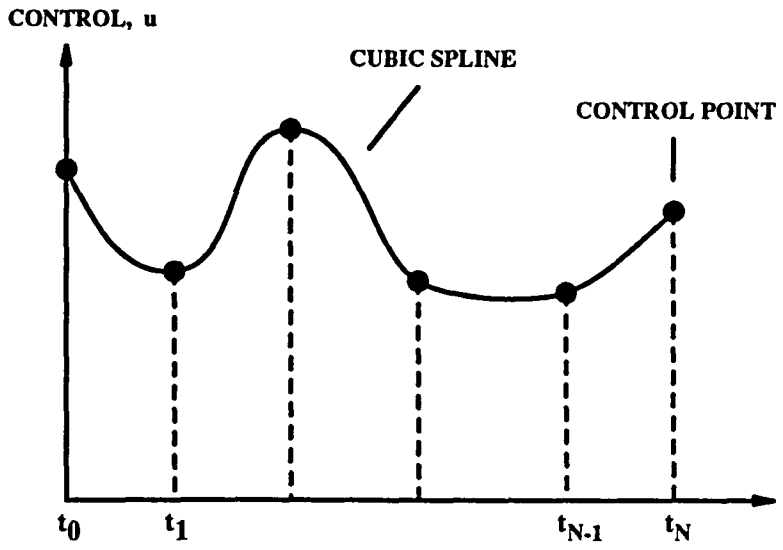

Figure 4.1: Discretization of the control

In this study only minimum-time problems are considered. This allows the performance index to be simplified to,

$$J = t_f \tag{4.5}$$

The optimal control problem presented above is an infinite-dimensional problem, in that the control $u(t)$ is to be found at each instant in the interval $t_0 \leq t \leq t_f$. A number of techniques exist for solving such problems, however they are in general cumbersome and require extensive problem preparation. A simple and yet effective approach is to transform the exercise into a parameter optimization problem by discretizing the control history. Figure 4.1 shows how the control history is defined in terms of discrete control nodes or points. The performance index can be evaluated for a specific set of nodes by the following procedure: 1) Fit the control nodes with cubic splines to obtain a continuous control history, 2) Beginning at the specified initial conditions, integrate the differential equations governing the aircraft trajectory,

3) Compute the performance index and the terminal constraints. These nodes thus make up the set of parameters that are adjusted to minimize the performance index, while satisfying the terminal constraints.

This investigation implements the simulated annealing algorithm Hide-and-Seek to optimize the cost function. As Hide-and-Seek contains no mechanism for handling terminal constraints, they are incorporated into the performance index as penalty functions. This results in the following augmented performance index.

$$\acute{J} = t_f + \sum_{i=1}^{l} K_i |g_i| \tag{4.6}$$

where K_i is the positive penalty constant for the g_ith constraint residual. Note that as Hide-and-Seek is a maximization routine, the performance index used by it is the negative of Equation 4.6.

In the following sections a number of classical optimal control problems are presented. These problems are solved using Hide-and-Seek and two standard non-gradient methods. Attempts were made to solve the problems by applying two sequential quadratic programming techniques: the SQP [20] algorithm and the NPSOL routine in the NAG Fortran Library. Both algorithms failed because of the high sensitivity of the problems and the inaccurate gradient information.

Table 4.1: 2-D half-loop initial conditions

State	Value
Velocity	636.4 ft/s (Mach 0.6)
Angle of attack	2.3 deg
Pitch angle	2.3 deg
Pitch rate	0.0 deg/s
Altitude	15,000.0 ft
Downrange distance	0.0 ft

4.2 Minimum Time-to-Half-Loop

4.2.1 2-D Minimum Time-to-Half-Loop

The time-optimal half-loop maneuver is investigated here. For the two dimensional case, only symmetric stabilator deflection and throttle setting are available as control inputs. The two control histories are parameterized by cubic splines over five equal time intervals bounded by six control nodes. With the addition of time-of-flight as a variable, the problem has a total of thirteen design parameters. The initial conditions used for the maneuver are given in Table 4.1. The trajectory is required to satisfy the following terminal constraints. As mentioned earlier the terminal equality constraints are enforced using penalty functions.

$$\gamma_f = 180\ deg \tag{4.7}$$

$$q_f = 0\ deg/s \tag{4.8}$$

$$h_f > h_i \tag{4.9}$$

Table 4.2: 2-D half-loop results

Method	Time-of-flight	Function evaluations
Hide-and-Seek	24.9 s	5,518
Nelder-Mead	39.4 s	2,084
Principal Axis	35.1 s	4,063

To evaluate the performance of Hide-and-Seek, the Nelder-Mead Simplex and the Principal Axis algorithms are also applied to this problem. The results presented for the latter two methods are the best solutions obtained from 20 different initial guesses. These results are summarized in Table 4.2 and the trajectories are displayed in Figure 4.2.

The number of function evaluations given for Hide-and-Seek is an average over five solutions. The results show that Hide-and-Seek finds the lowest time-of-flight. The Nelder-Mead and Principal Axis solutions have time-of-flights that are greater by 58 % and 41 %, respectively. Due to the use of a strict convergence criteria, the Hide-and-Seek trajectory is essentially guaranteed to be the global optimal solution. This flight path is however found at a higher cost in terms of function evaluations relative to the other two methods. Figures 4.3–4.7 display the state and control histories of the Hide-and-Seek trajectory. The angle-of-attack history shows that the optimal solution requires the aircraft to fly in the high angle-of-attack regime. Due to the proximity of stall, this region of the flight envelope contains numerous discontinuities. As Nelder-Mead and Principal Axis are incapable of traversing discontinuities in the feasible design space, the solutions they acquire are only locally optimal. It should be noted that as there is no constraint on the Mach number (Figure 4.7) the aircraft

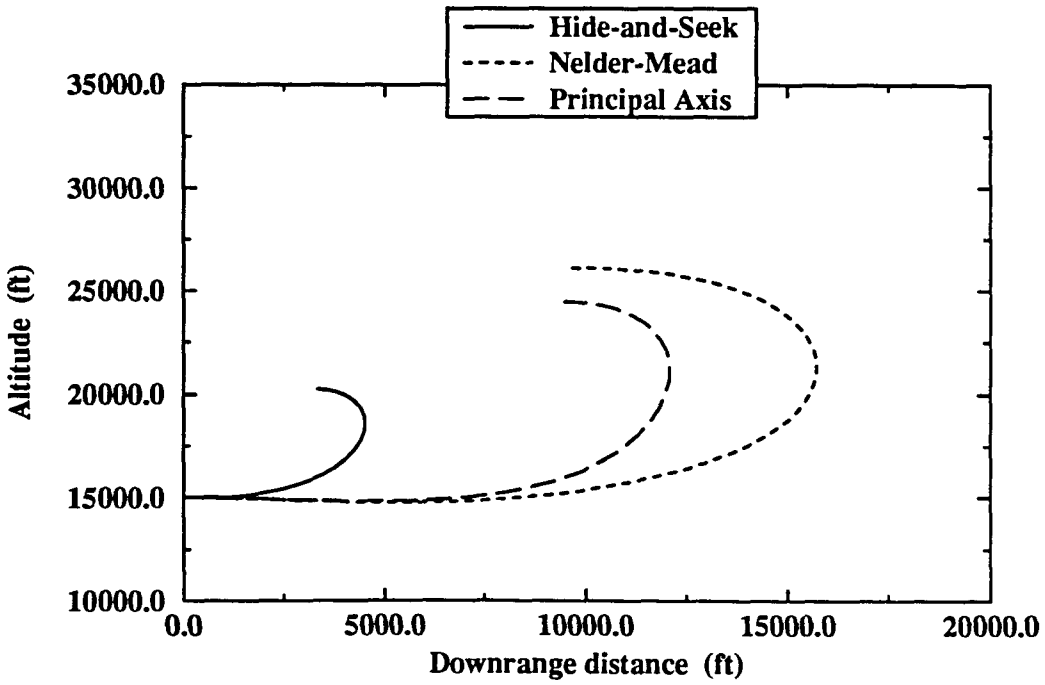

Figure 4.2: 2-D half-loop trajectories

tends to loose its kinetic energy in favor of potential energy. This is done to satisfy the constraints as rapidly as possible.

To investigate the effects of the dimensionality of the problem, the exercise is repeated using larger numbers of design variables. The time-of-flight is divided into ten and fifteen equal intervals, bounded by eleven and sixteen nodes per control input. With the addition of time-of-flight, these supplementary problems have twenty-three

Table 4.3: Number of function evaluations for 2-D half-Loop

No. of variables	Hide-and-Seek	Hybrid
13	5,518	7,215
23	5,713	11,614
33	10,015	$< 100,000$

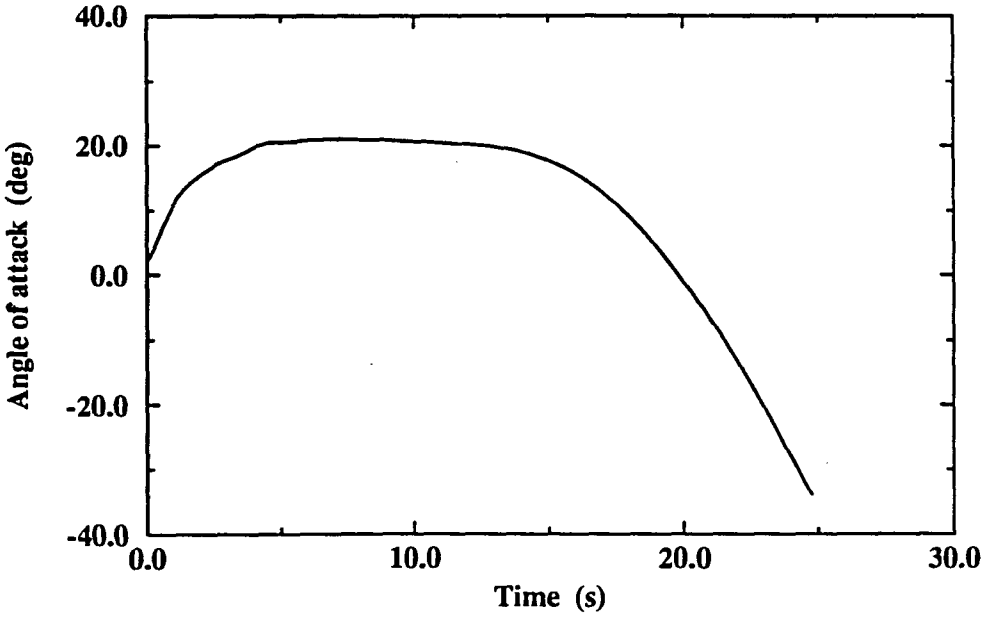

Figure 4.3: 2-D half-loop angle-of-attack history

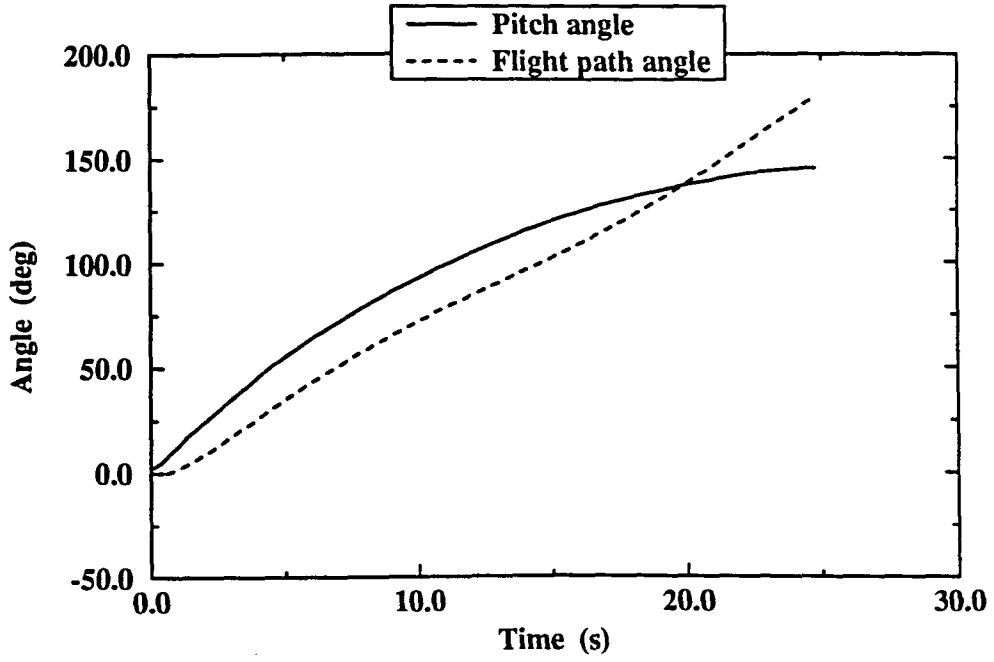

Figure 4.4: 2-D half-loop pitch and flight path angle history

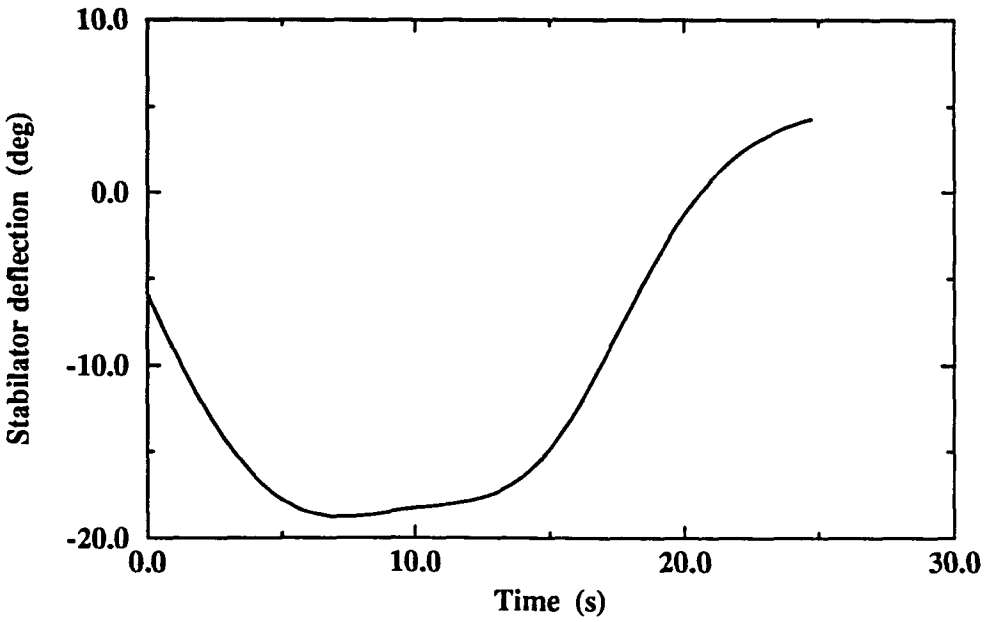

Figure 4.5: 2-D half-loop stabilator deflection history

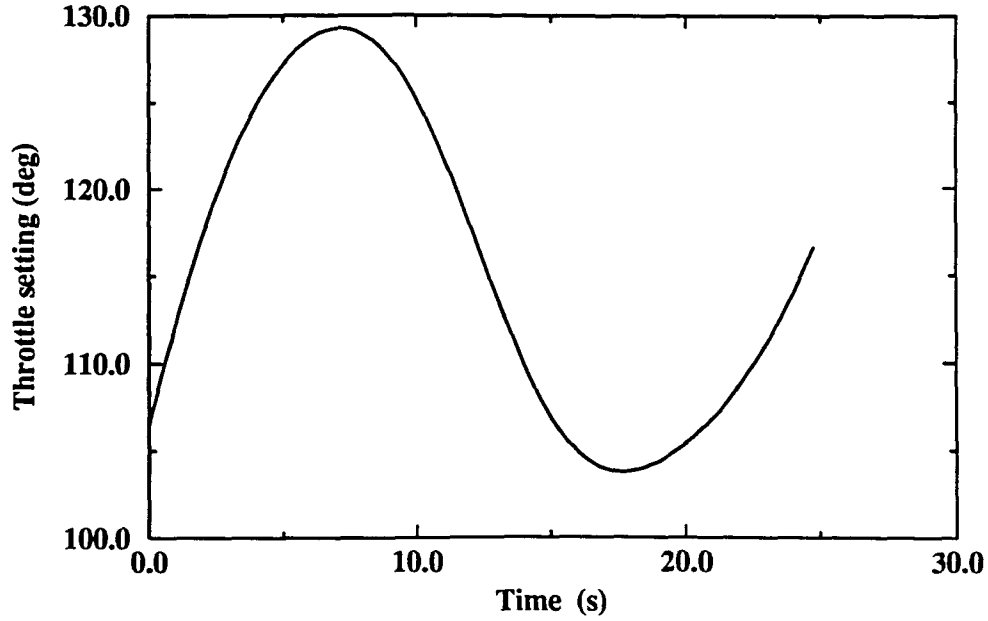

Figure 4.6: 2-D half-loop throttle control history

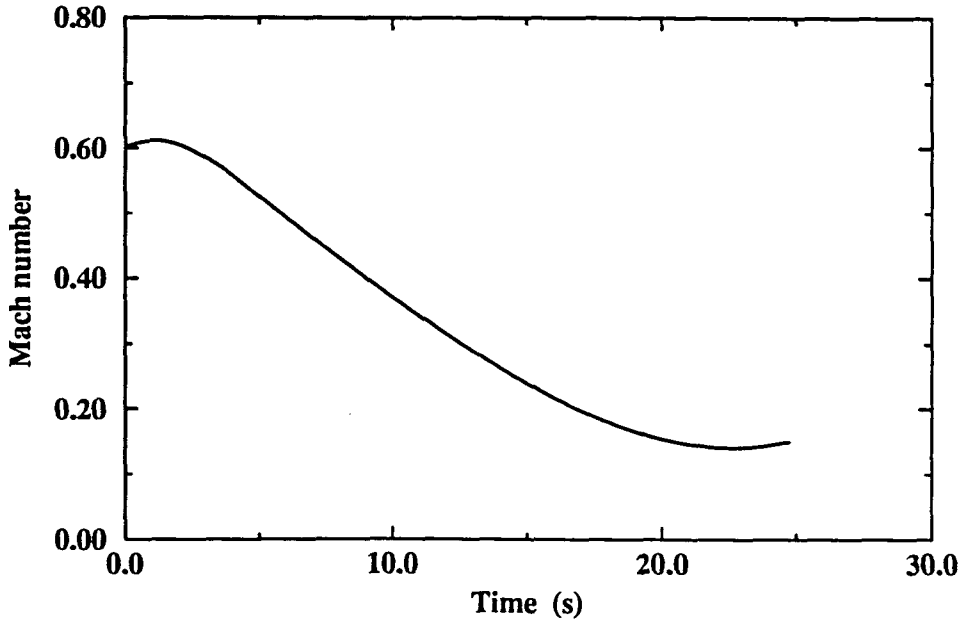

Figure 4.7: 2-D half-loop Mach number history

and thirty-three design variables. A summary of the results is presented in Table 4.3. The flight times of these solutions range from 24.7 to 25.0 seconds. These findings show that the number of function evaluations required does not rapidly increase with the dimension of the problem.

The Hybrid method is also applied on these problems. In the Hybrid method, the solution from Hide-and-Seek acts as the initial guess for the Nelder-Mead Simplex technique. In all cases, the Hybrid method solutions are approximately the same as the Hide-and-Seek solutions, with the minor difference being that the constraints are better satisfied. This however is at a much higher cost in terms of function evaluations. Note that due to the Nelder-Mead section of the Hybrid method, for the thirty-three design variables case, in excess of 100,000 function evaluations were completed without convergence. The process was deemed not converging and terminated.

4.2.2 3-D Minimum Time-to-Half-Loop

The three-dimensional half-loop maneuver is a significantly more complex problem to solve than the two-dimensional case. During three-dimensional flight, the aircraft has five control inputs: the symmetric and differential stabilators, the aileron, the rudder, and the throttle setting. For this problem the time-of-flight is divided into five equal intervals. This produces six design points for each aircraft control, giving a total of thirty-one design parameters including the time-of-flight. The initial conditions used for this problem are shown in Table 4.4. Due to the different ranges of various angles in three-dimensional flight, a different set of constraints must be satisfied. These constraints are given below.

$$\gamma_f = 0 \; deg \tag{4.10}$$

$$\psi_f = 180 \; deg \tag{4.11}$$

$$\phi_f = 180 \; deg \tag{4.12}$$

$$q_f = 0 \; deg/s \tag{4.13}$$

$$h_f > h_i \tag{4.14}$$

The optimal solution found by Hide-and-Seek is 21.7 s. This result is obtained after completing an average of 29,246 function evaluations. The trajectory is shown in two and three dimensions in Figures 4.8 and 4.9. Though the optimal trajectory is approximately within the vertical plane (the maximum cross-range distance is only 21.5 ft), with the additional control surfaces the aircraft is able to reduce the time-of-flight by 14.7 % as compared to the two-dimensional case. The aircraft does not significantly go out of the initial plane of motion as out-of-plane motion requires the expenditure of additional energy, and hence tends to extend the flight time.

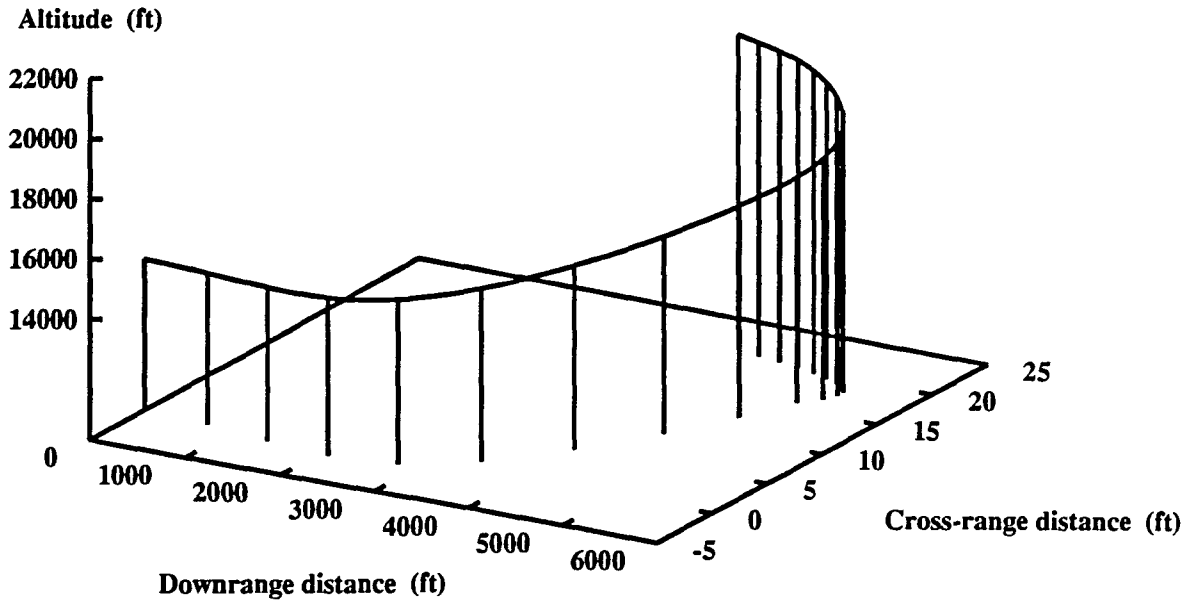

Figure 4.8: 3-D half-loop trajectory (3-D view)

Table 4.4: 3-D half-loop initial conditions

State	Value
Velocity	636.4 ft/s (Mach 0.6)
Angle of attack	2.3 deg
Sideslip angle	0.0 deg
Pitch angle	2.3 deg
Heading angle	0.0 deg
Roll angle	0.0 deg
Pitch rate	0.0 deg/s
Yaw rate	0.0 deg/s
Roll rate	0.0 deg/s
Altitude	15,000.0 ft
Downrange distance	0.0 ft
Cross-range distance	0.0 ft

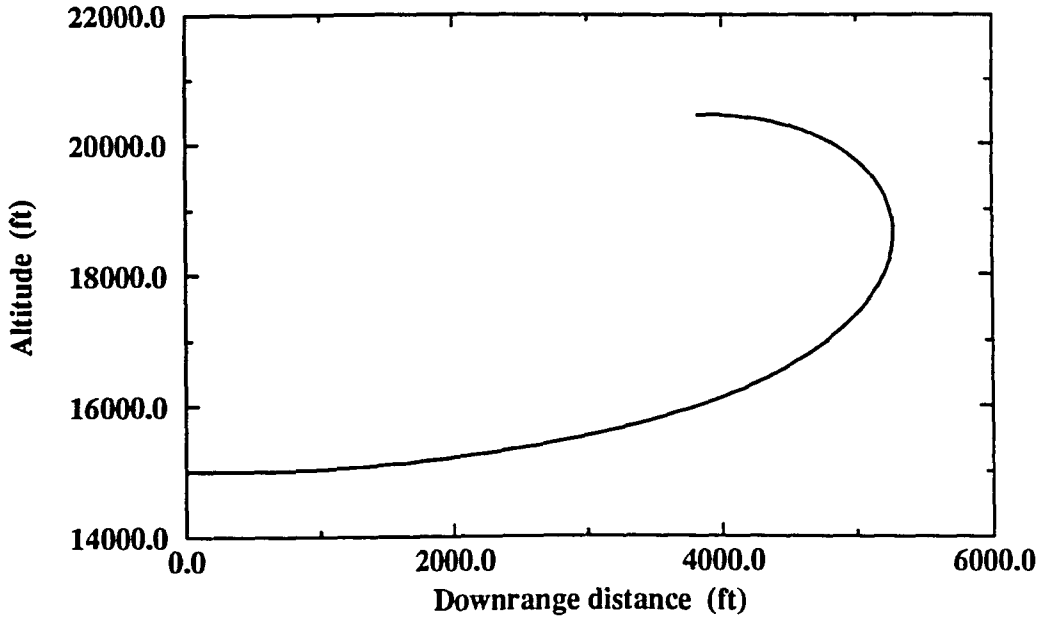

Figure 4.9: 3-D half-loop trajectory (vertical plane view)

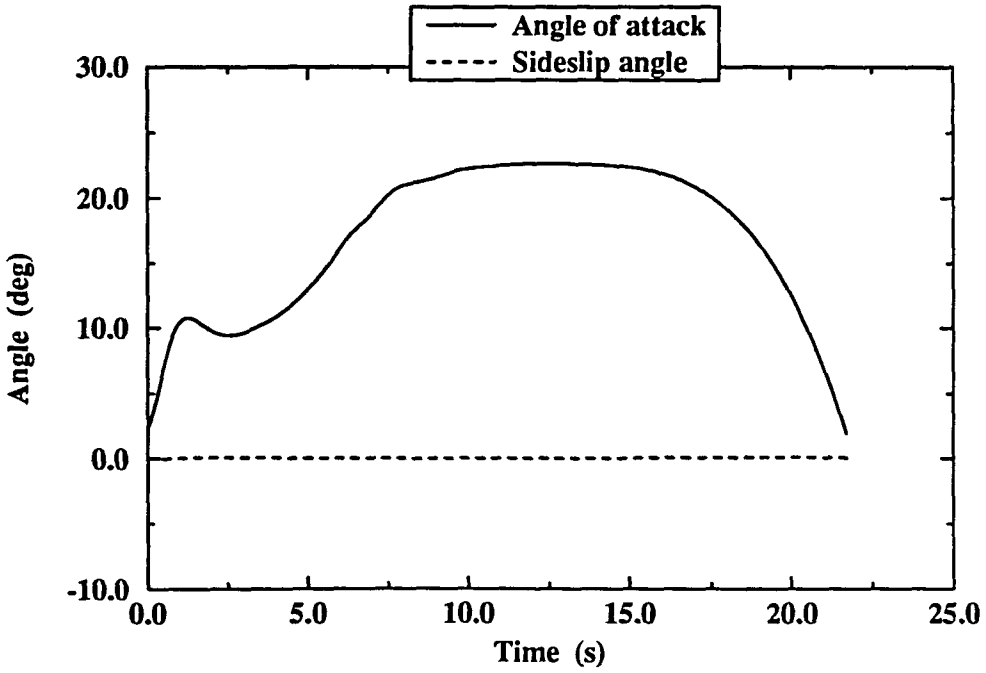

Figure 4.10: 3-D half-loop angle-of-attack and sideslip history

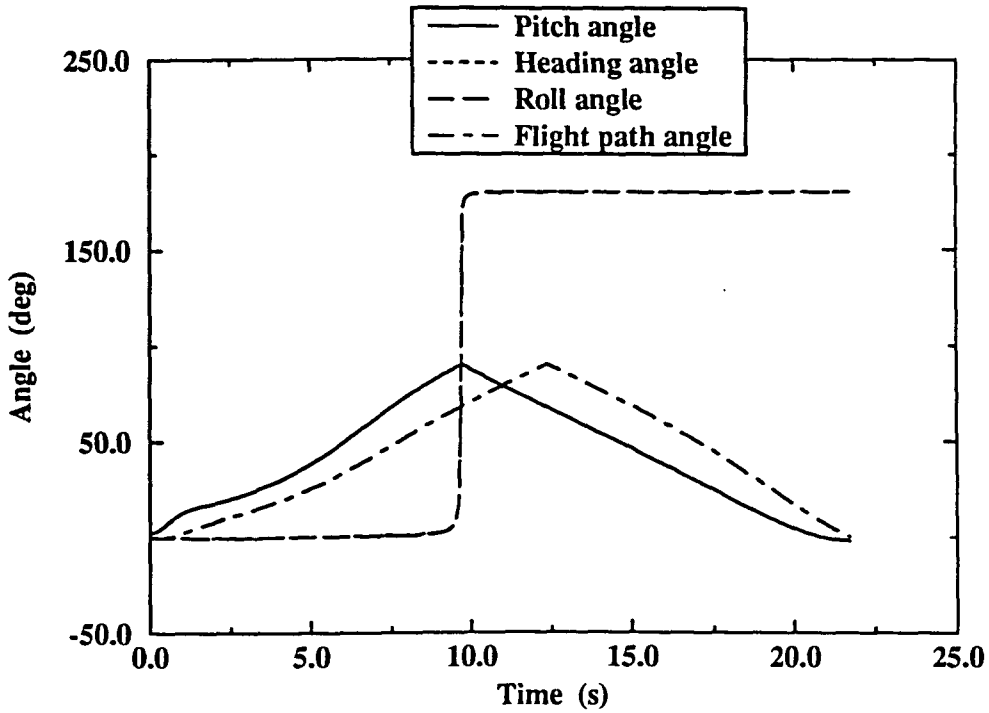

Figure 4.11: 3-D half-loop attitude angles history

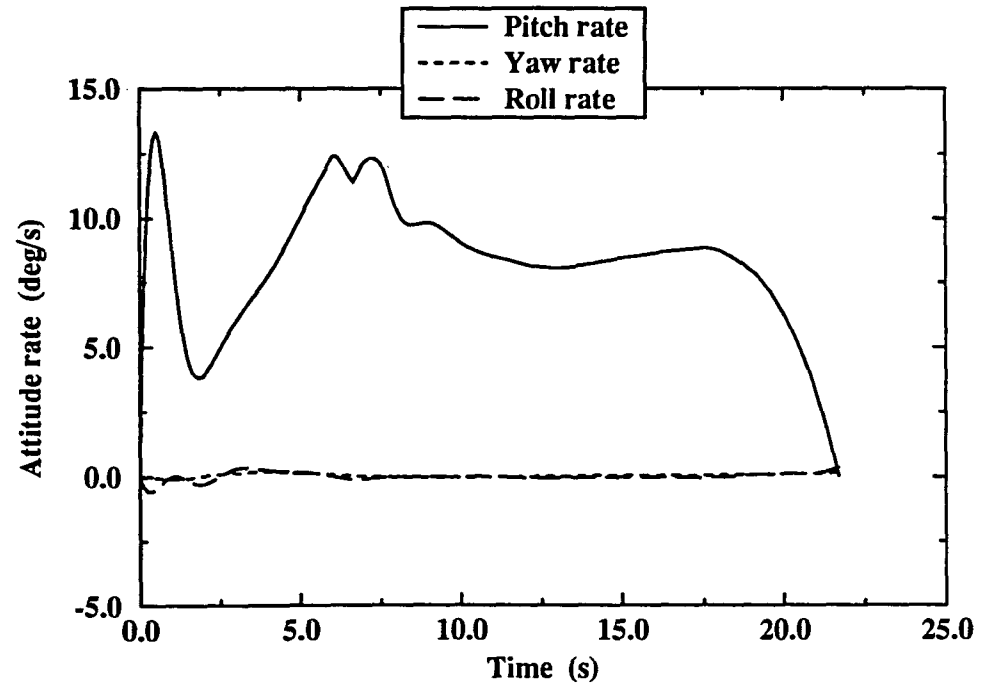

Figure 4.12: 3-D half-loop attitude rates history

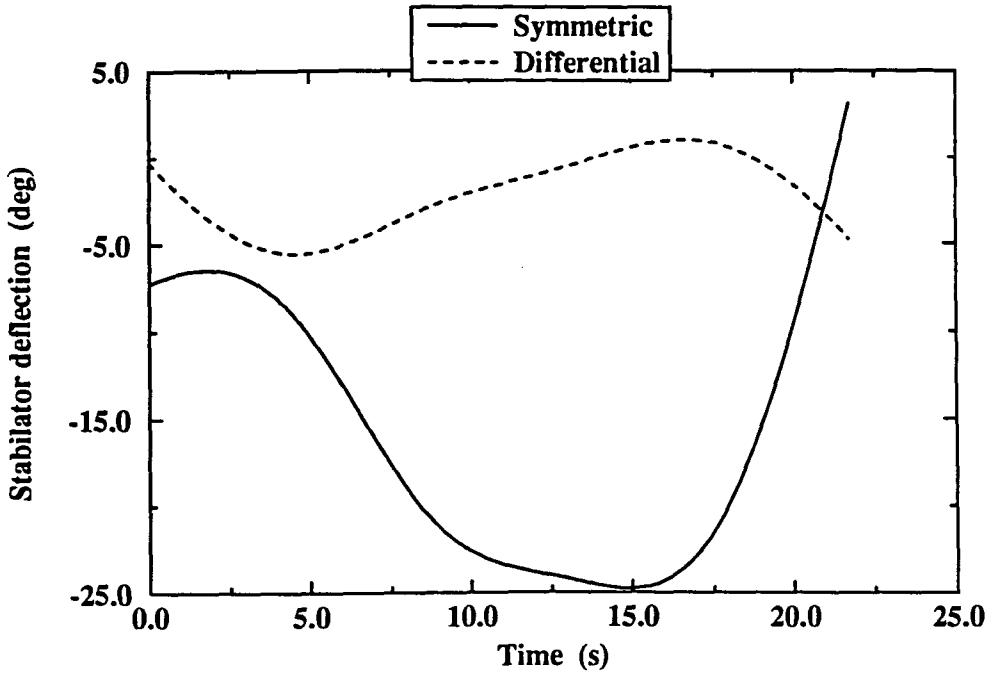

Figure 4.13: 3-D half-loop stabilator deflection history

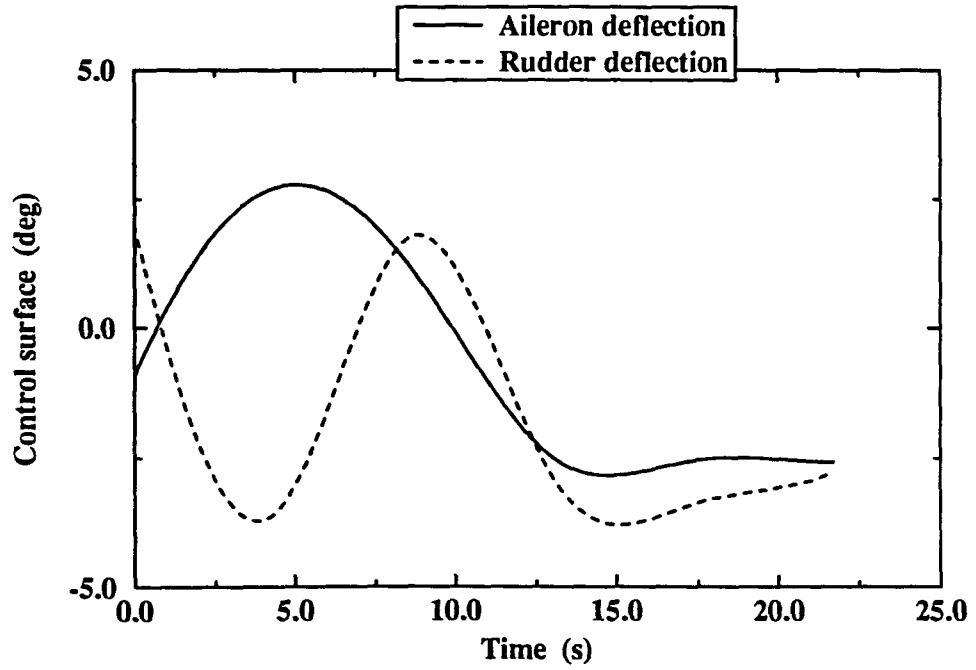

Figure 4.14: 3-D half-loop aileron and rudder history

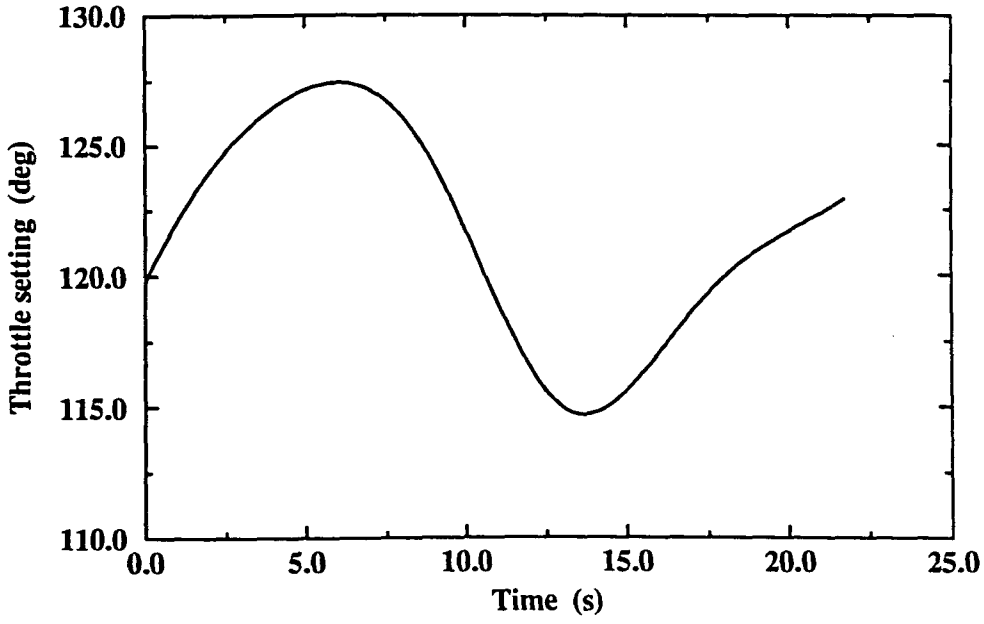

Figure 4.15: 3-D half-loop throttle setting history

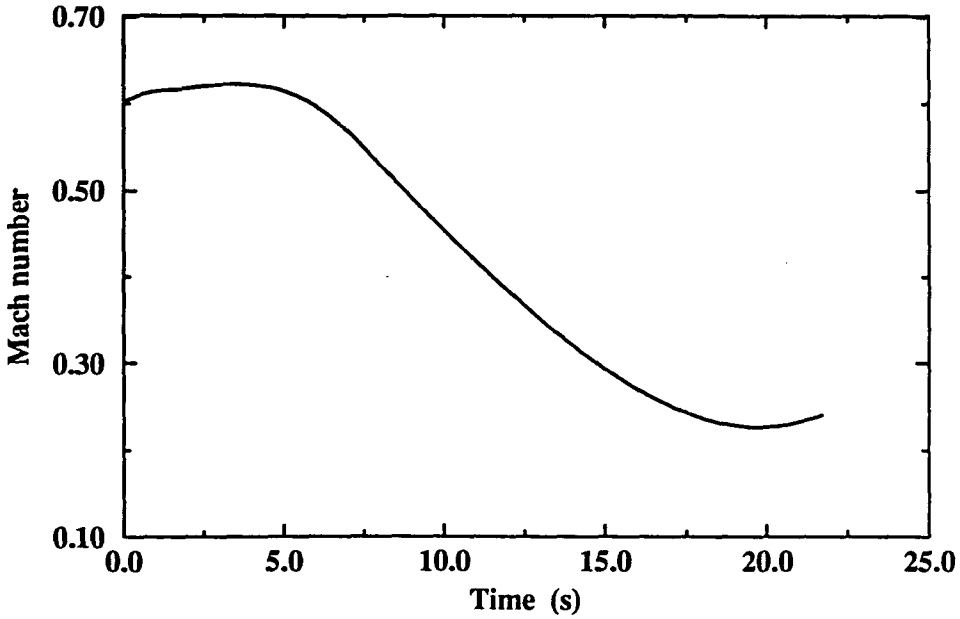

Figure 4.16: 3-D half-loop Mach number history

Figures 4.10–4.16 display the state and control histories of this trajectory. Note that in Figure 4.11 the histories of heading and roll angle almost coincide. The Nelder-Mead Simplex and Principal Axis methods are not used for this problem because from the two-dimensional case we have already seen that for problems of this size, both methods will use an extremely excessive number of function evaluations. In addition, if they converge at all, they will converge on locally optimal solutions. This makes both methods impractical for application to three-dimensional flight problems.

4.3 Minimum Time-to-Turn

The minimum time-to-turn trajectory is studied for three-dimensional flight. The initial conditions and the design parameters used in this problem are identical to those employed for the three-dimensional half-loop maneuver. The constraints

that define the trajectory are as follows:

$$\gamma_f = 0\ deg \tag{4.15}$$

$$\psi_f = 180\ deg \tag{4.16}$$

$$\phi_f = 0\ deg \tag{4.17}$$

$$h_f = h_i \tag{4.18}$$

Hide-and-Seek obtained an optimal flight time of 30.6 s. This solution is found at an average (over four runs) cost of 51,258 function evaluations. A three-dimensional view of the optimal trajectory is shown in Figure 4.17. Unlike the half-loop maneuver, this flight path contains considerable lateral motion. To produce this motion, the aircraft applies large rudder deflections. Figures 4.19–4.25 display the state and control histories of the turn maneuver. It should noted that though this is an optimal trajectory, none of the control inputs reach saturation. Comparing Figures 4.16 and 4.25 one can observe the difference in Mach number histories. During the half-loop maneuver the Mach number declines rapidly, while during the turn maneuver the Mach number is always greater than the initial value.

4.4 Minimum Time-to-Climb

The time-optimal climb is a classical aircraft performance problem. The flight paths investigated here are limited to two-dimensional flight in the vertical plane. Solutions to this problem are obtained using Hide-and-Seek and the Nelder-Mead Simplex algorithm. Since in the past, it has been found that time-optimal flight paths operate at or close to the maximum throttle setting, for this set of problems the throttle is fixed at 127 *deg*. This setting allows some leeway for guidance con-

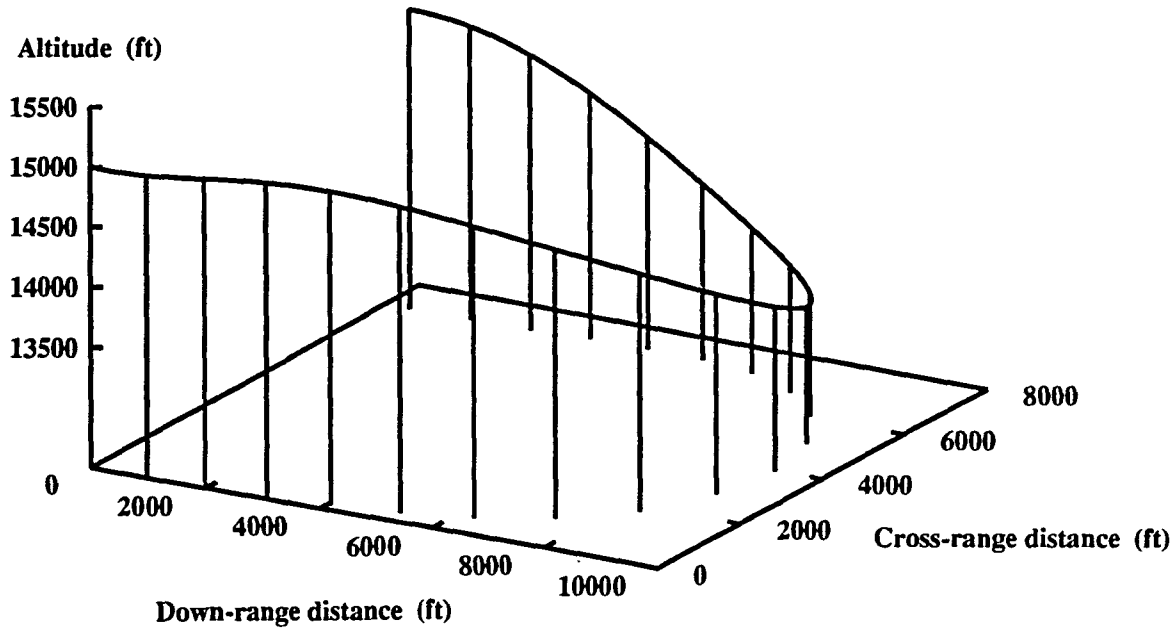

Figure 4.17: 3-D turn trajectory (3-D view)

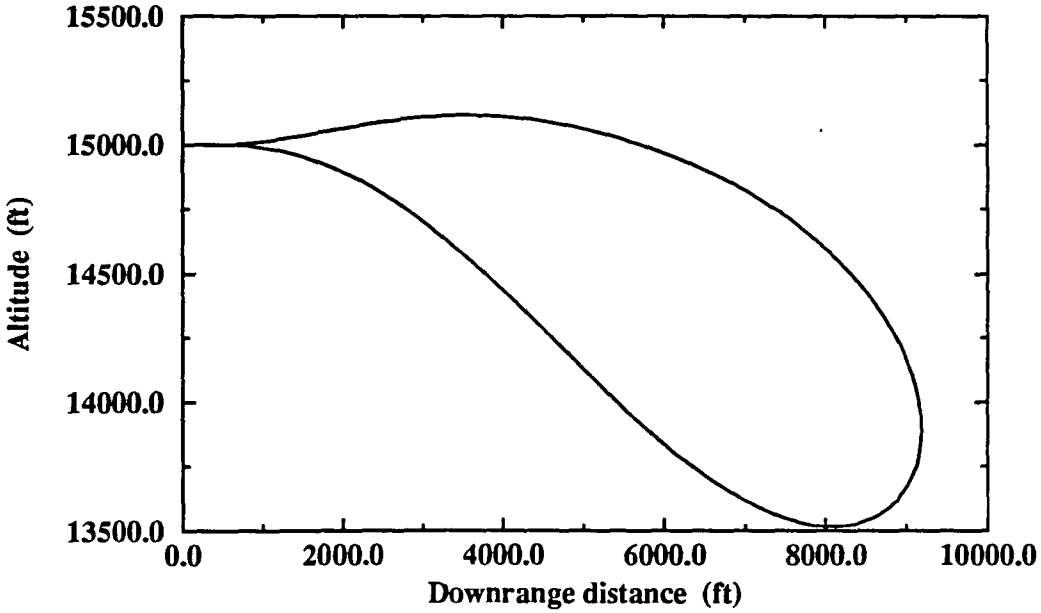

Figure 4.18: 3-D turn trajectory (vertical plane view)

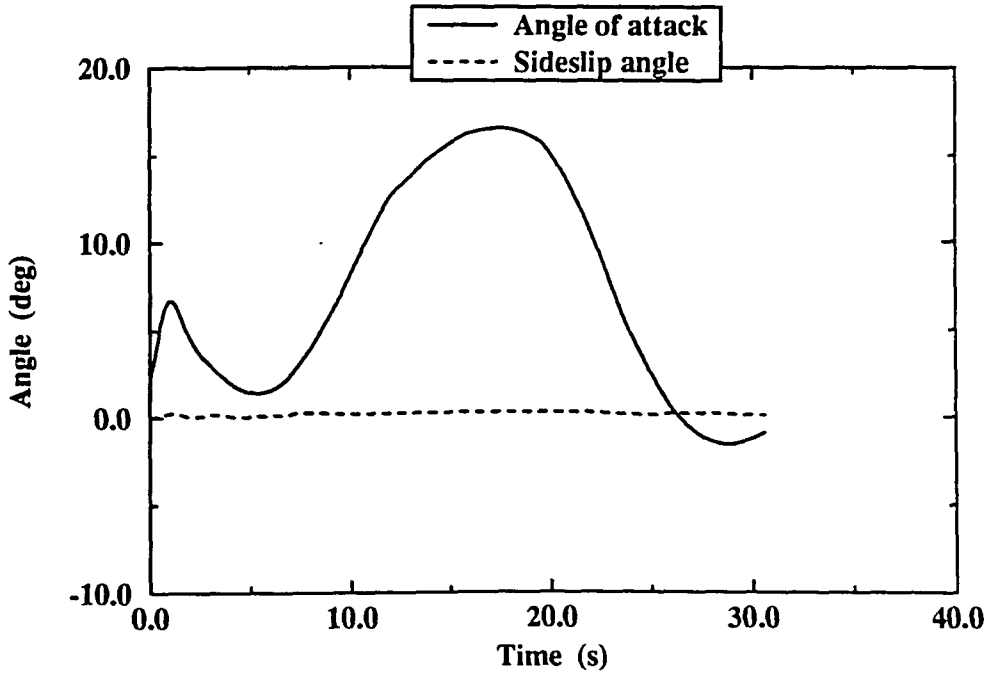

Figure 4.19: 3-D turn angle-of-attack and sideslip history

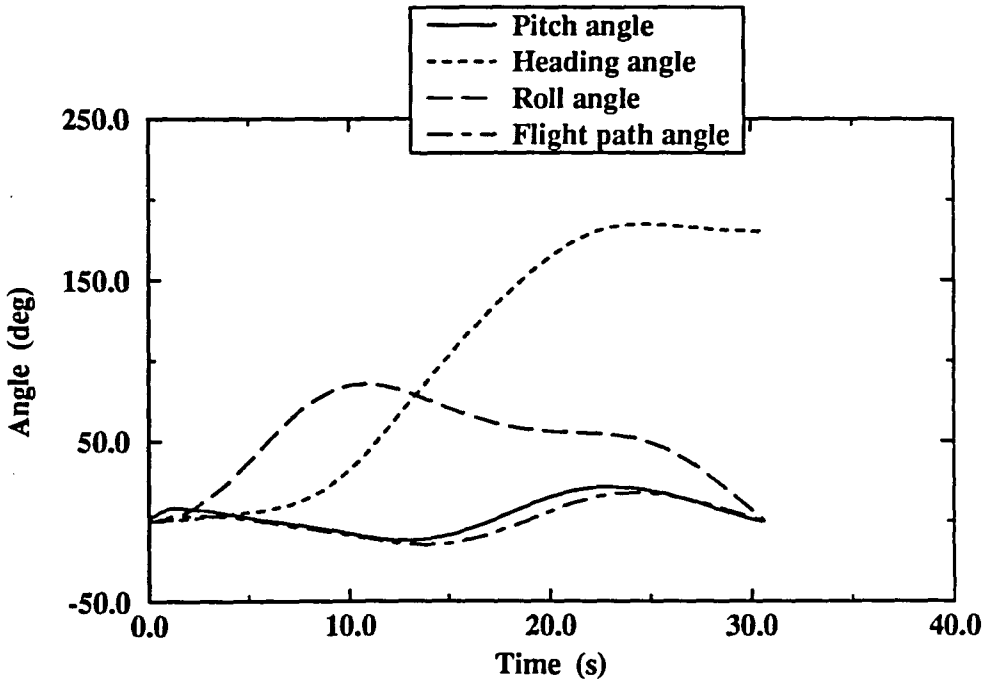

Figure 4.20: 3-D turn attitude angles history

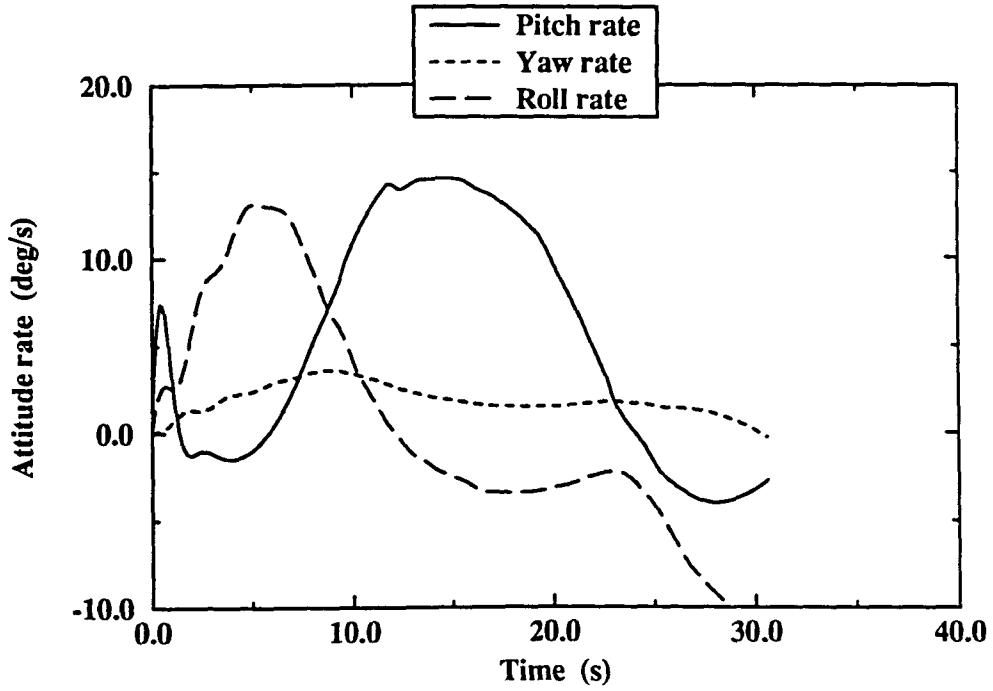

Figure 4.21: 3-D turn attitude rates history

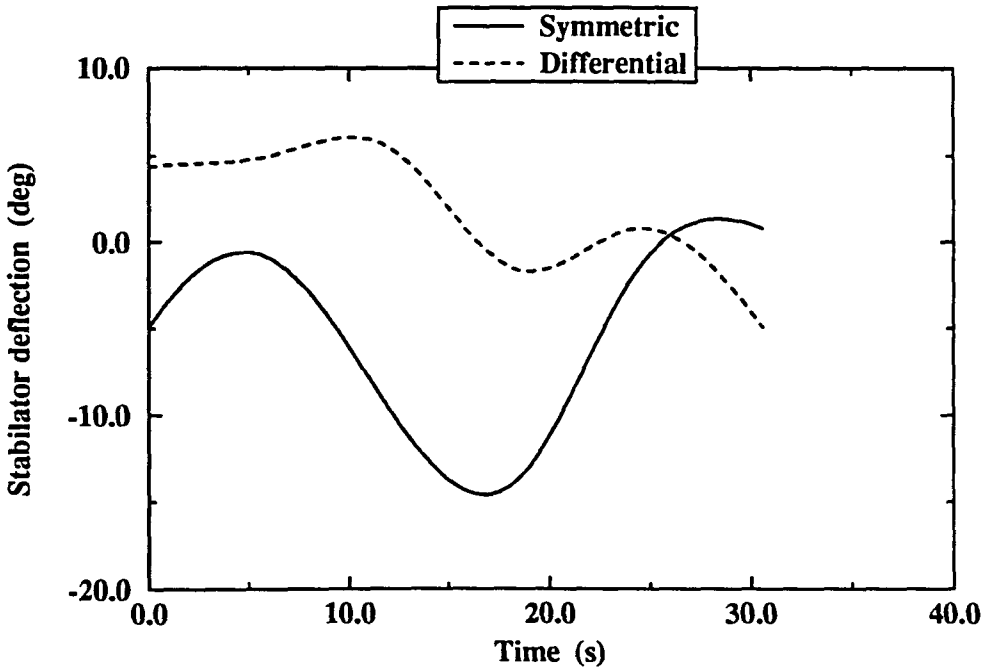

Figure 4.22: 3-D turn stabilator deflection history

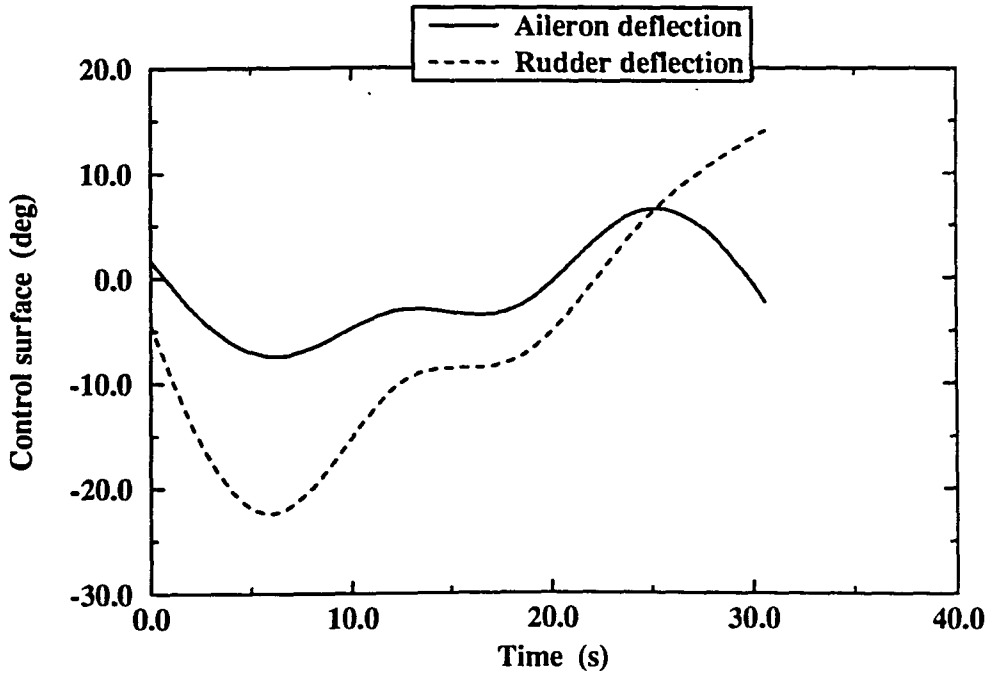

Figure 4.23: 3-D turn aileron and rudder history

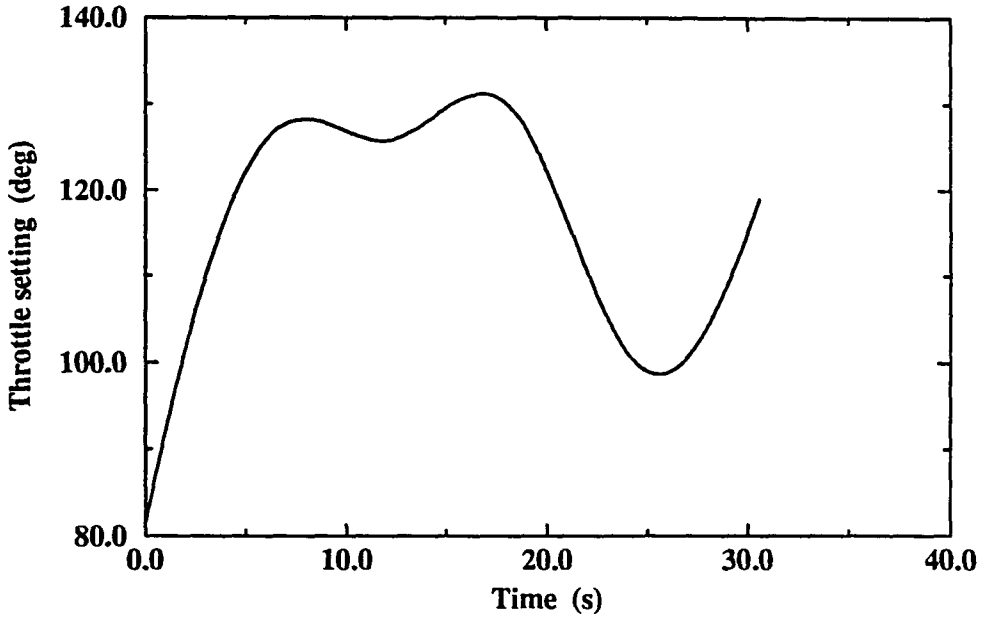

Figure 4.24: 3-D turn throttle setting history

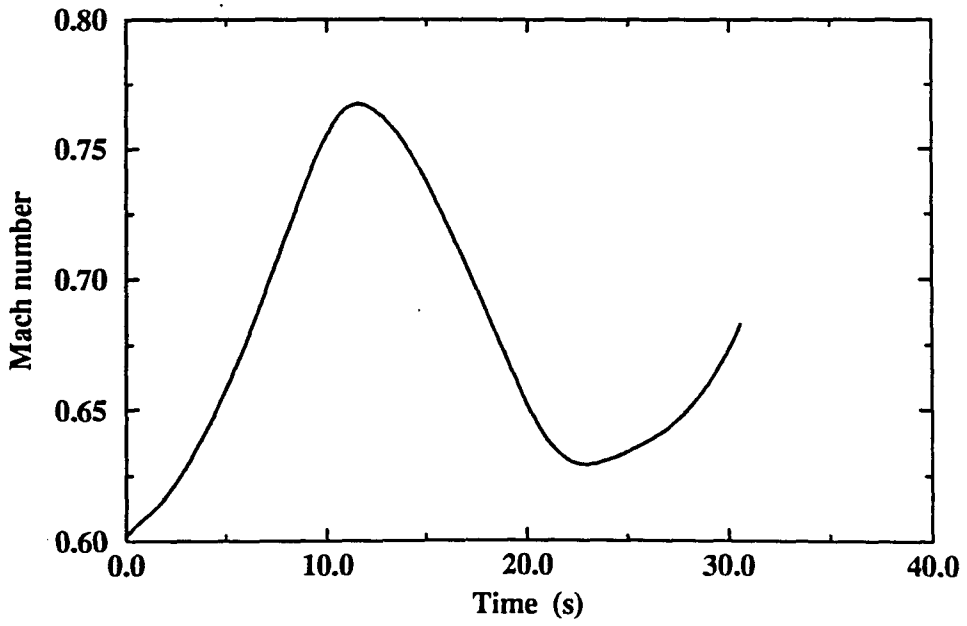

Figure 4.25: 3-D turn Mach number history

siderations. The time history of the stabilator deflection is discretized into ten equal intervals. This gives a total of twelve design parameters including the time-of-flight. The climb problem is investigated with two (4.19 and 4.20), three (4.19–4.21) and four (4.19–4.22) constraints. The first constraint specifies that the minimum altitude should be greater than zero. As this is an inequality constraint, the standard way to enforce it in Hide-and-Seek is to declare any trajectory that violates the constraint to be infeasible. The drawback with this approach is that as the flight path begins at a zero altitude, a very large percentage of the possible aircraft trajectories become infeasible. To avoid this problem, the inequality constraint is converted into the following terminal constraint residual.

$$g_1 = |min\{0.0, h_{min}\}| \tag{4.19}$$

The remaining constraints are as follows.

$$h_f = 50{,}000\ ft \tag{4.20}$$

$$M_f = 1.0 \tag{4.21}$$

$$\gamma_f = 0.0\ deg \tag{4.22}$$

The initial conditions for these problems are given in Table 4.5. The results from both Hide-and-Seek and Nelder-Mead Simplex are summarized in Tables 4.6 and 4.7. Each result presented for Nelder-Mead is the best solution obtained from 10 different initial guesses. The number of function evaluations given for Hide-and-Seek is an average over four runs. The solutions show that Hide-and-Seek found flight times that are 19.4 % to 44.8 % lower than the best results from Nelder-Mead. Note that Nelder-Mead uses less function evaluations for these problems. It is expected though

Table 4.5: Minimum time-to-climb initial conditions

State	Value
Velocity	400.0 ft/s
Angle of attack	5.0 deg
Pitch angle	5.0 deg
Pitch rate	0.0 deg/s
Altitude	0.0 ft
Downrange distance	0.0 ft

Table 4.6: Minimum time-to-climb trajectories: Flight times

Constraints	Nelder-Mead (12 variables)	Hide-and-Seek (12 variables)	Hide-and-Seek (15 variables)
2	171.3 s	118.3 s	117.3 s
3	186.3 s	156.0 s	153.2 s
4	199.3 s	161.2 s	160.4 s

that when the size of the problems increase, Nelder-Mead will rapidly become too expensive. The optimal trajectories from both methods are displayed in Figures 4.26 and 4.27. The flight paths marginally satisfy the minimum altitude constraint. To allow a greater degree of freedom in the initial stages of the climb, the first time-of-flight interval is further divided into four segments. This gives a total of fifteen design parameters. The resulting trajectories are shown in Figure 4.28. The additional three variables produced a negligible improvement in the flight times but did improve the ground clearance of the trajectories.

Figures 4.29–4.37 display the state and control histories of the Hide-and-Seek fifteen variable solutions. Bryson [1] and Ong [3] found that for certain aircraft

Table 4.7: Minimum time-to-climb trajectories: Function evaluations

Constraints	Nelder-Mead (12 variables)	Hide-and-Seek (12 variables)	Hide-and-Seek (15 variables)
2	16,844	26,030	21,270
3	14,157	33,775	37,896
4	14,207	31,280	37,170

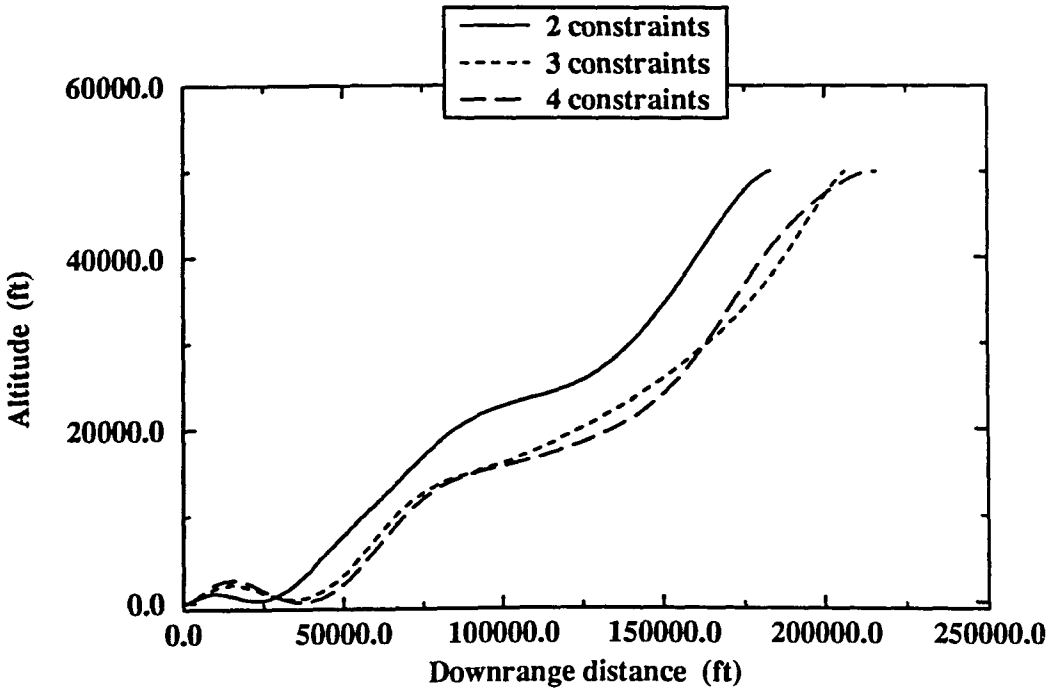

Figure 4.26: Nelder-Mead climb trajectories with 12 variables)

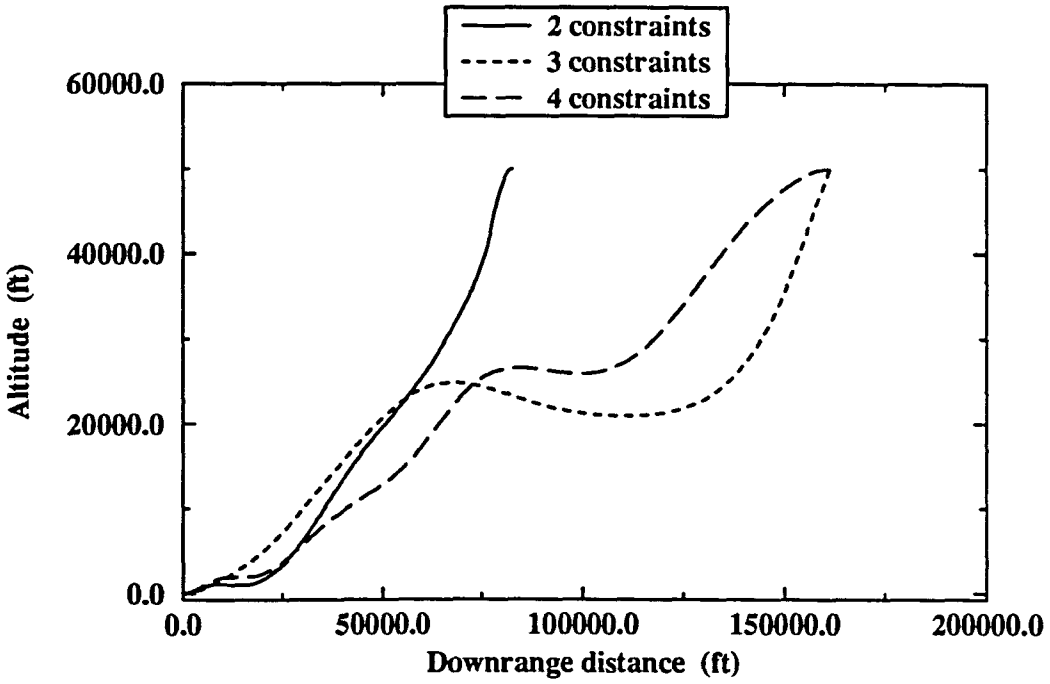

Figure 4.27: Hide-and-Seek climb trajectories with 12 variables

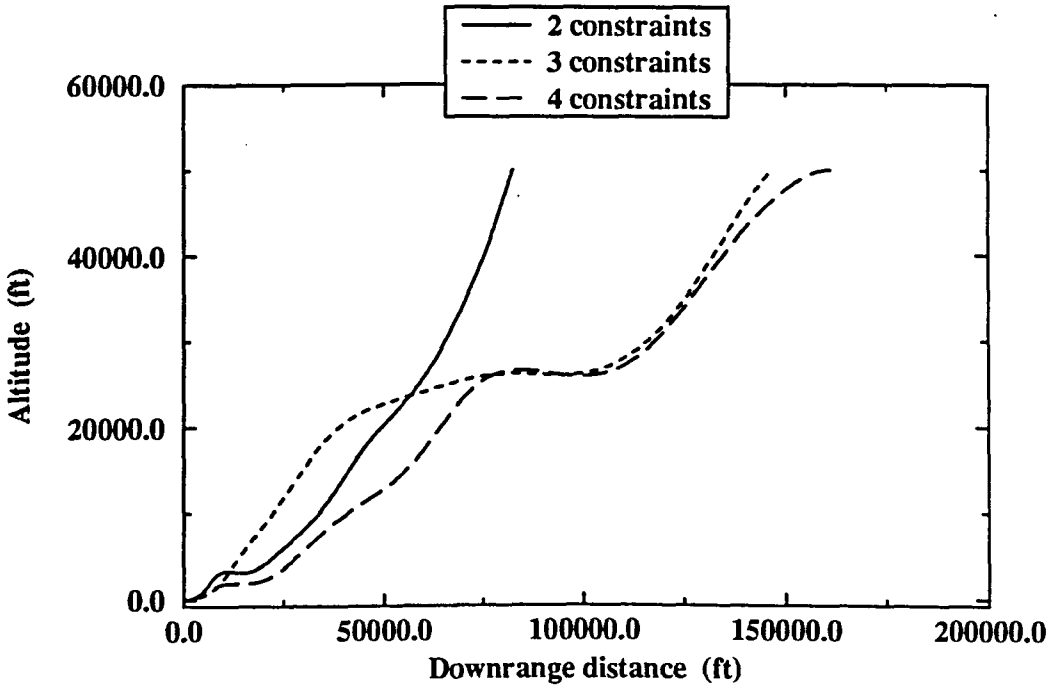

Figure 4.28: Hide-and-Seek climb trajectories with 15 variables

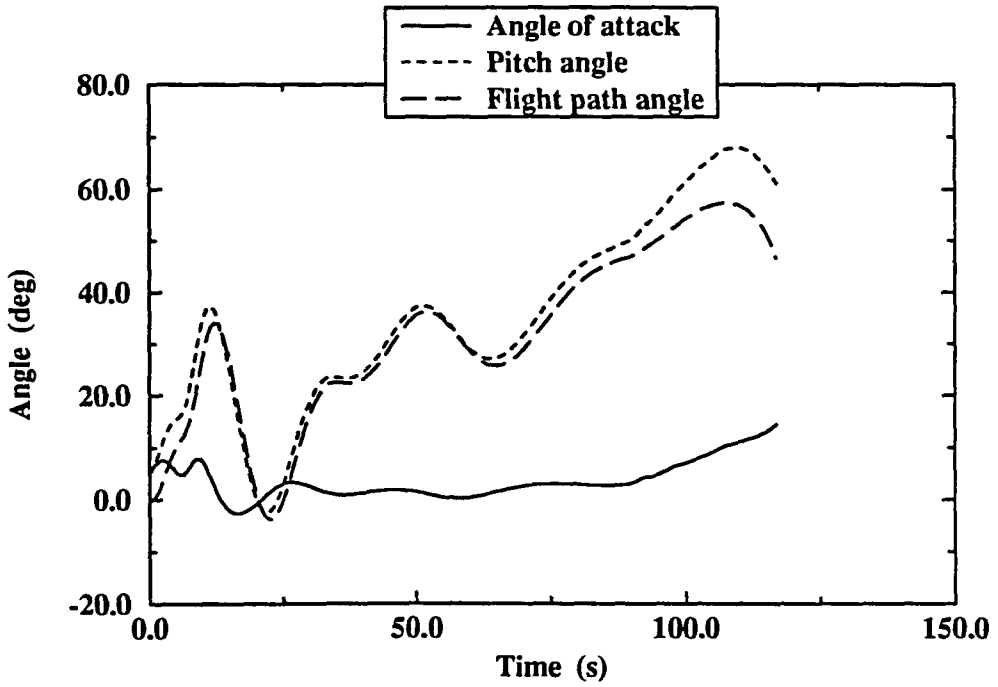

Figure 4.29: Angle histories for 2 constraint 15 variable climb

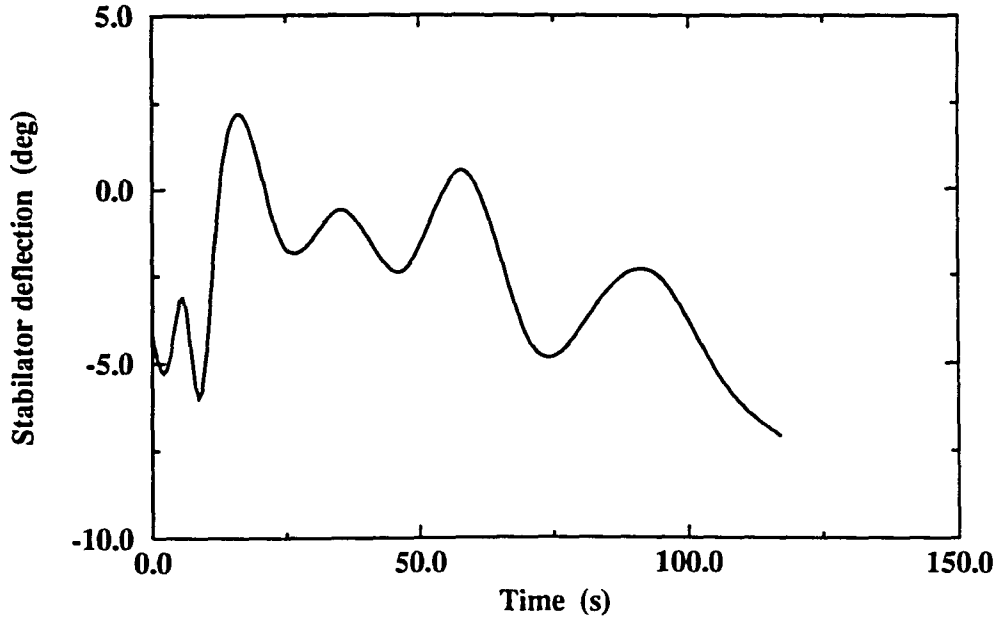

Figure 4.30: Control history for 2 constraint 15 variable climb

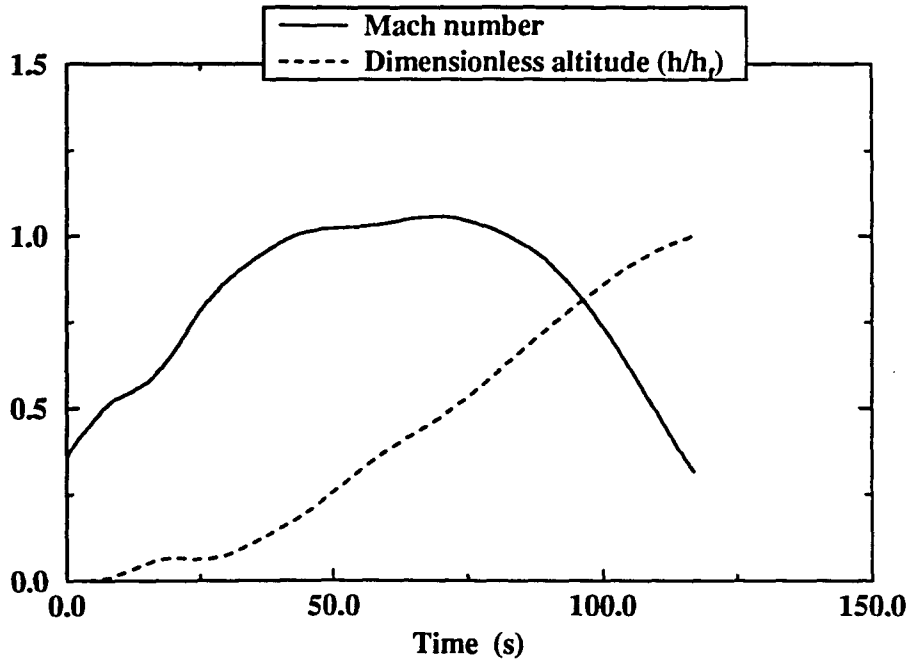

Figure 4.31: Mach number history for 2 constraint 15 variable climb

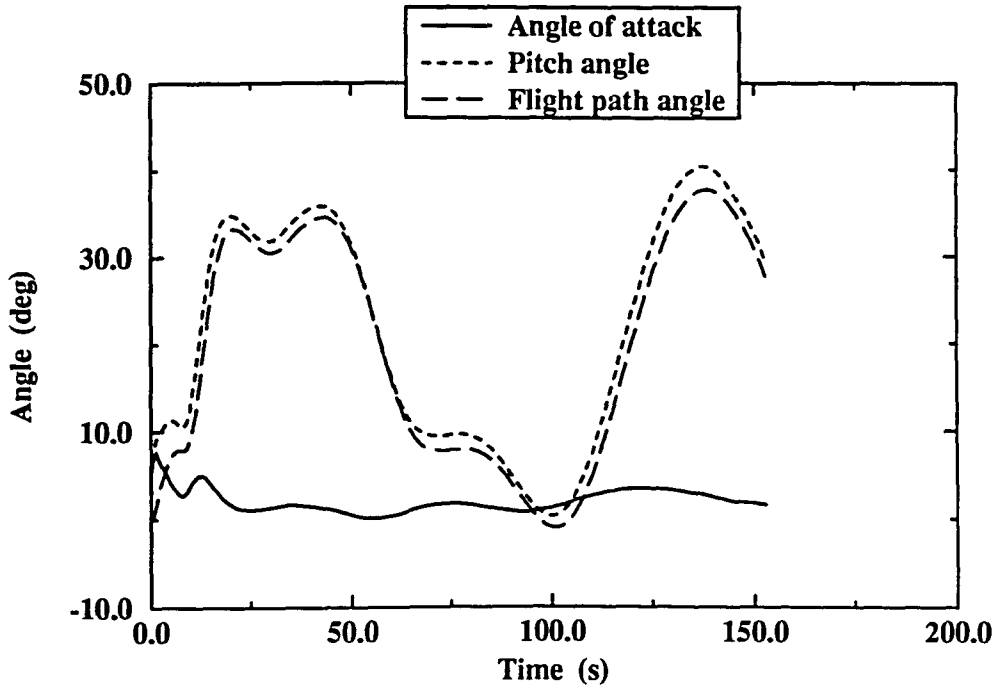

Figure 4.32: Angle histories for 3 constraint 15 variable climb

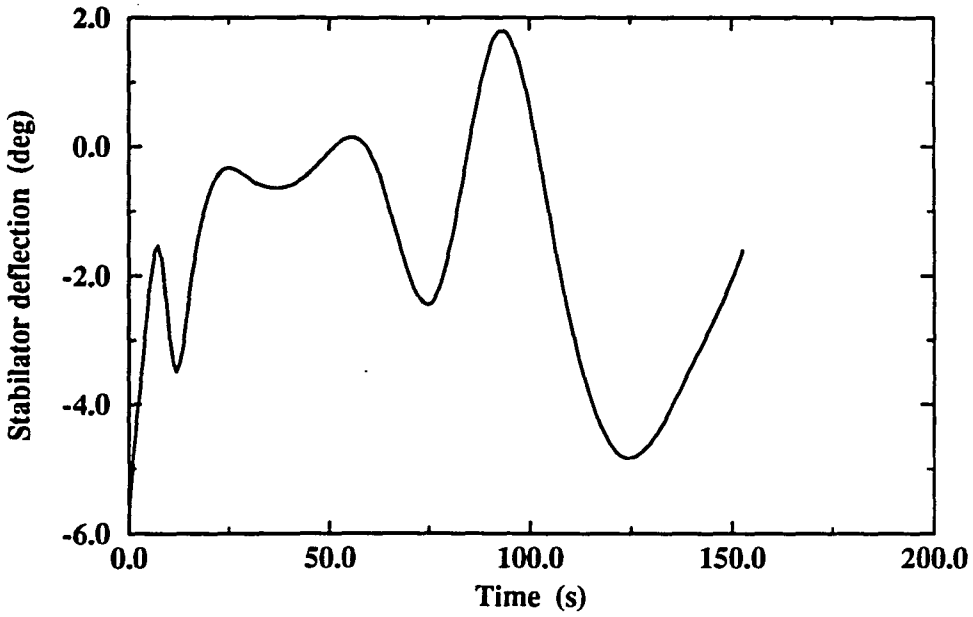

Figure 4.33: Control history for 3 constraint 15 variable climb

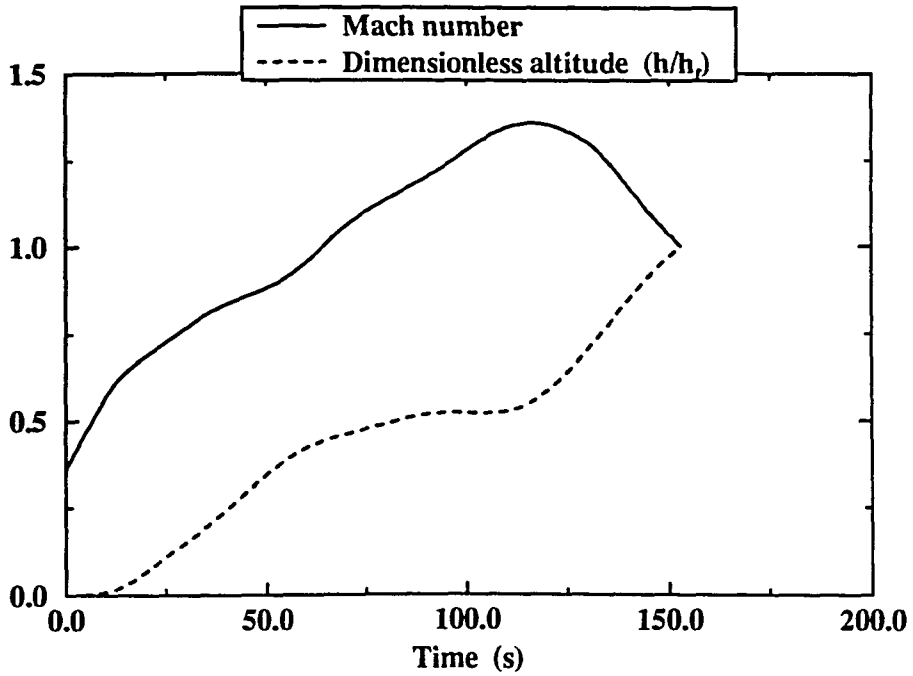

Figure 4.34: Mach number history for 3 constraint 15 variable climb

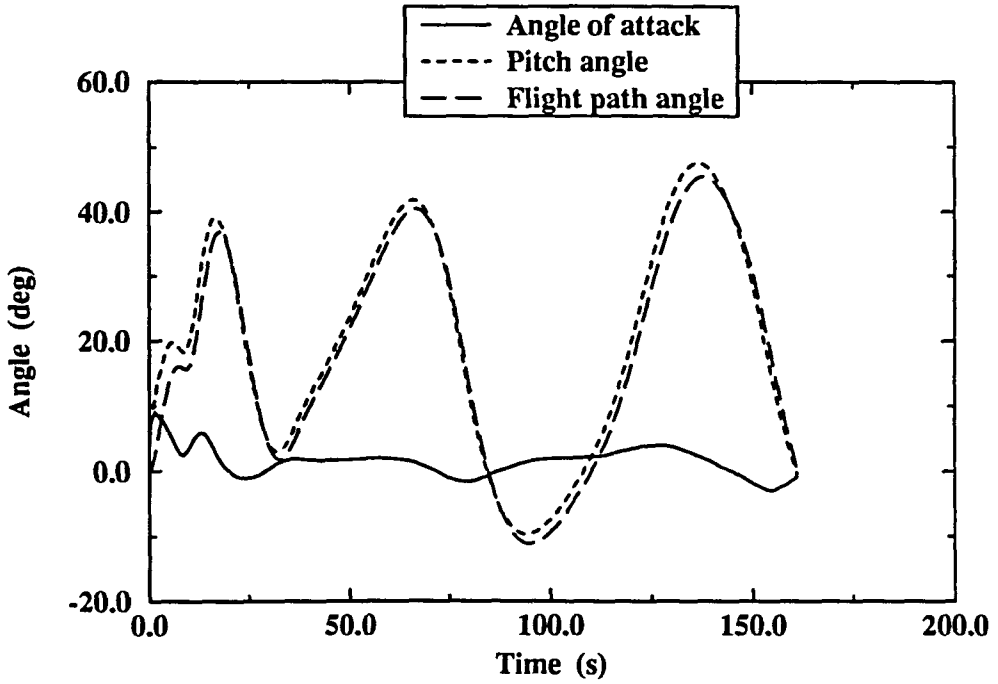

Figure 4.35: Angle histories for 4 constraint 15 variable climb

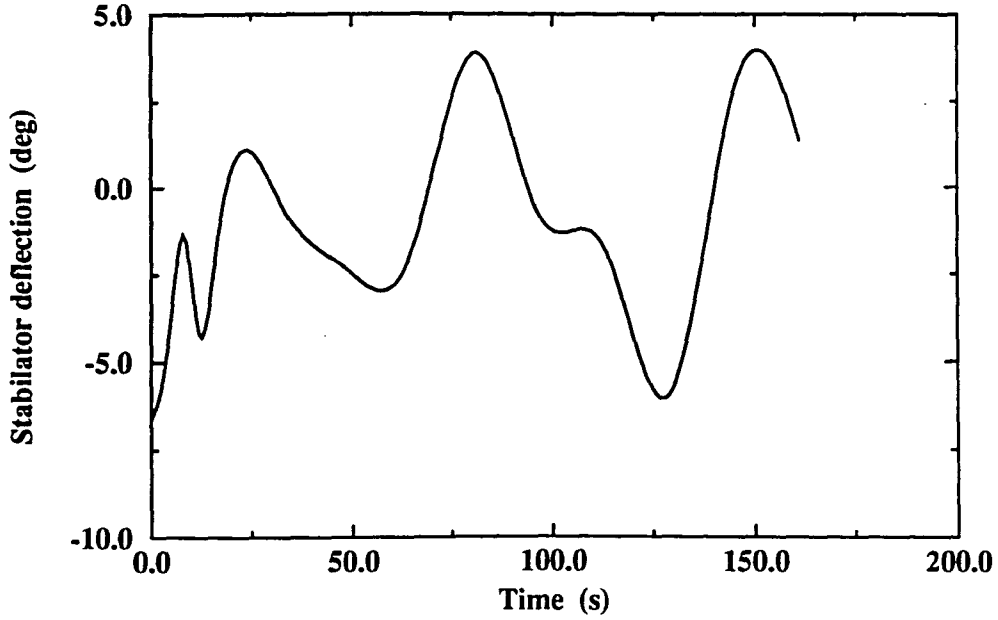

Figure 4.36: Control history for 4 constraint 15 variable climb

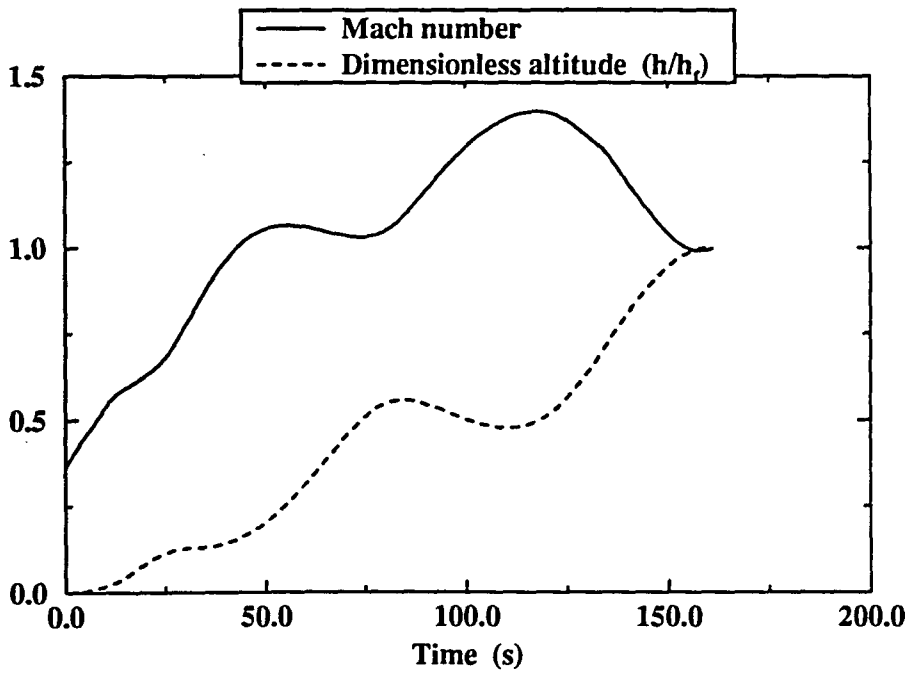

Figure 4.37: Mach number history for 4 constraint 15 variable climb

models, the minimum time-to-climb trajectory required the aircraft to dive in order to break the sound barrier. The optimal trajectories obtained here did not show this feature. This is probably due to the more powerful engines (the thrust-to-weight ratio is greater than one) and the better aerodynamic characteristics of the aircraft.

4.5 Recommendations for Hide-and-Seek Implementation

During the use of Hide-and-Seek, it was found that by applying various problem formulation techniques, the efficiency of the algorithm can be enhanced. These modifications are in no way required for Hide-and-Seek implementation. The following is a brief list of the recommended procedures.

1. Choose variable bounds that largely result in feasible trajectories.

2. Whenever possible, narrow the variable bounds to reduce the search area.

3. Scale the design variables to have ranges of approximately the same magnitude.

4. Scale the objective function to avoid a very slow or very rapid decrease in the temperature parameter. During the optimization process, the objective function should at most change by three orders of magnitude.

CHAPTER 5. GUIDANCE LAW DEVELOPMENT

Control histories obtained by parameter optimization are open-loop in nature. If internal or external disturbances are present in the system, the application of these open-loop control histories will result in trajectories that diverge from the optimal flight paths. Consequently to follow these trajectories a guidance scheme is needed. The guidance scheme must generate the guidance commands required to fly the aircraft so as to follow a given reference flight path. In this chapter a new nonlinear feedback controller is implemented. Lu [23] recently developed this controller for application on a broad class of dynamic systems. The controller is based on the minimization of predicted tracking errors. During the course of this study, the controller is further developed to enhance its performance on aircraft trajectory tracking problems.

5.1 Controller I: Basic Controller Design

The following is the derivation of the basic controller as described by Lu [23].

Suppose the dynamic system equations for $x \in \mathbf{R}^n$ have the form,

$$\dot{x}_1 = f_1(x) \tag{5.1}$$

$$\dot{x}_2 = f_2(x) + B(x)u \tag{5.2}$$

where $x_1 \in \mathbf{R}^{n_1}$, $x_2 \in \mathbf{R}^{n_2}$ and $n_1 + n_2 = n$. Partition the reference trajectory $s(t)$

accordingly,

$$s(t) = \begin{pmatrix} s_1(t) \\ s_2(t) \end{pmatrix} \tag{5.3}$$

Suppose at $t_k \in (t_0, t_f)$, $x(t_k)$ is known. Consider $x(t_{k+1}) = x(t_k + \hbar)$, where $\hbar > 0$. Applying the Runge-Kutta 2nd order scheme to Equation 5.1 and the Euler method to Equation 5.2 gives,

$$x_{1_{k+1}} = x_{1_k} + \frac{\hbar}{2}\left[f_1(x_k) + f_1\left(x_k + \hbar \begin{pmatrix} f_1(x_k) \\ f_2(x_k) + B(x_k)u_k \end{pmatrix}\right)\right] \tag{5.4}$$

$$x_{2_{k+1}} = x_{2_k} + \hbar(f_2(x_k) + B(x_k)u_k) \tag{5.5}$$

Expanding the last term in Equation 5.4,

$$f_1\left(x_k + \hbar \begin{pmatrix} f_1(x_k) \\ f_2(x_k) + B(x_k)u_k \end{pmatrix}\right) = f_1(x_k) + \hbar F_{11} f_1(x_k) + \hbar F_{12}(f_2(x_k) + B(x_k)u_k) \tag{5.6}$$

where $F_{11} = \frac{\partial f_1}{\partial x_1}(x_k)$ and $F_{12} = \frac{\partial f_1}{\partial x_2}(x_k)$. Hence, Equation 5.4 can be rewritten as,

$$x_{1_{k+1}} = x_{1_k} + \hbar f_1(x_k) + \frac{\hbar^2}{2}(F_{11}f_1(x_k) + F_{12}f_2(x_k) + F_{12}B(x_k)u_k) \tag{5.7}$$

It is assumed that $F_{12}B(x_k)$ has no zero rows. This assumption is in general satisfied by all mechanical systems if no actuator dynamics are considered. To minimize the predicted tracking error, we can define the following performance index,

$$J_k = \frac{1}{2}e_{1_{k+1}}^T Q_1 e_{1_{k+1}} + \frac{1}{2}e_{2_{k+1}}^T \hbar^2 Q_2 e_{2_{k+1}} + \frac{1}{2}u_k^T \hbar^4 R u_k \tag{5.8}$$

where $e_{1_{k+1}} = x_{1_{k+1}} - s_{1_{k+1}}$ and $e_{2_{k+1}} = x_{2_{k+1}} - s_{2_{k+1}}$. Q_1, Q_2 and R are positive semi-definite square matrices of the appropriate dimensions. The future reference

states $s_{1_{k+1}}$ and $s_{2_{k+1}}$ are given by,

$$s_{1_{k+1}} = s_{1_k} + \hbar \dot{s}_{1_k} + \frac{\hbar^2}{2} \ddot{s}_{1_k} \tag{5.9}$$

$$s_{2_{k+1}} = s_{2_k} + \hbar \dot{s}_{2_k} \tag{5.10}$$

The performance index J_k is a quadratic function in u_k when $x_{1_{k+1}}$ and $x_{2_{k+1}}$ are approximated by Equations 5.7 and 5.5. As it can be shown that $\frac{\partial^2 J_k}{\partial u_k^2} \geq 0$, J_k has a unique minimum at $\frac{\partial J_k}{\partial u_k} = 0$. Solving for the u_k that satisfies this expression gives,

$$u_k = -W^{-1} \left\{ \frac{1}{2\hbar^2} G^T Q_1 P_1 + \frac{1}{\hbar} B(x_k)^T Q_2 P_2 \right\} \tag{5.11}$$

where the following substitutions and expansions have been made:

$$G = F_{12} B(x_k) \tag{5.12}$$

$$W = \frac{1}{4} G^T Q_1 G + B(x_k)^T Q_2 B(x_k) + R \tag{5.13}$$

$$P_1 = e_{1_k} + \hbar \dot{e}_{1_k} + \frac{\hbar^2}{2} \left(F_{11} f_1(x_k) + F_{12} f_2(x_k) - \ddot{s}_{1_k} \right) \tag{5.14}$$

$$P_2 = e_{2_k} + \hbar (f_2(x_k) - \dot{s}_{2_k}) \tag{5.15}$$

Since t_k can be arbitrarily chosen as any point in the interval (t_0, t_f), this controller can more generally be written as,

$$u(t) = -W^{-1} \left\{ \frac{1}{2\hbar^2} G^T Q_1 P_1 + \frac{1}{\hbar} B(x)^T Q_2 P_2 \right\} \tag{5.16}$$

This controller has various desirable characteristics. It involves only the Jacobians of $f_1(x)$. Typically $f_1(x)$ specifies the kinematic relations of the system and thus is often explicitly known. This results in F_{11} and F_{12} being simple to obtain. The variable $\hbar$ is not required to be the "integration step size". It can be treated as an additional control parameter that is adjusted to improve the performance of the controller. An

analysis of the controller equation reveals several additional properties. When the initial error is zero, the trajectory is perfectly tracked (i.e., $e(t) \equiv 0$ for all $t \in (t_0, t_f)$) and the control command produced is exactly the nominal control. If the number of tracked variables is equal to that of controls, asymptotic tracking is guaranteed, provided no control saturation occurs. It can also be established that the controller is robust in the presence of a class of modeling uncertainties. To simplify notation, this controller is henceforth referred to as Controller I.

In this study, the application of the controller is limited to two-dimensional aircraft trajectories. To define the aircraft model in terms of a dynamic system described by Equations 5.1 and 5.2, the state vector is partitioned as follows.

$$x_1 = \begin{pmatrix} \theta \\ h \\ x \end{pmatrix} \tag{5.17}$$

$$x_2 = \begin{pmatrix} V \\ \alpha \\ q \end{pmatrix} \tag{5.18}$$

The two control inputs for the x_2 vector are symmetric stabilator deflection and throttle setting. To make the controller applicable to the model, the dynamic equations are linearized with respect to the control. The linearization is performed at each instant in time, about the current nominal control and the given state variables. This procedure yields two aircraft models: 1) Model A: the original model and 2) Model B: a linearized model that uses the linear in control dynamic equations to generate the aircraft trajectory. The original model uses the linearized dynamics in the controller calculations, but then applies the control commands to the original

dynamic equations. The controller is tested on both aircraft models. Model B gives a fairer assessment of the performance of the controller, while testing on Model A expounds the suitability of the controller for aircraft guidance.

The testing of the controller is conducted by using the fifteen variable, two constraint minimum time-to-climb solution obtained in Chapter 4 as the reference trajectory. Guidance along this flight path is a very difficult task. Even in nominal conditions the aircraft is operating close to the maximum throttle setting. In addition, a large portion of the trajectory is flown at low Mach numbers and high altitudes. In such flight conditions the effectiveness of the control surfaces is considerably reduced. For testing the controller, off-nominal conditions are created by introducing perturbations into the initial conditions of the state variables. The performance of the controller is judged on how well the altitude history is followed. The results for a number of initial perturbations are given in Tables 5.1 and 5.2. The maximum error is the greatest difference in the nominal and guided altitude histories at any given time. The mean error is defined by,

$$\bar{E} = \frac{\int_{t_0}^{t_f} |h_{actual} - h_{ref}| \, dt}{t_f} \tag{5.19}$$

The testing found that to track the reference trajectory only two or three state variable errors should be penalized in the weighting matrices Q_1 and Q_2. As the controller is being evaluated on it's ability to follow the altitude history, altitude errors must be tracked. Of the remaining penalized state errors, at least one must be an angle or angular rate. The altitude and control histories of two Model B test cases are shown in Figures 5.1–5.6. The results show that the controller provides fair tracking of the nominal trajectory. The largest altitude error is less than 7.5% of the total height climbed. This is an exceptional achievement considering that

Table 5.1: Controller I: Model A results

Perturbation	Maximum error (ft)	Mean error (ft)
+80 ft/s velocity	1,020	226
−80 ft/s velocity	1,206	230
+5 deg angle of attack	43.1	5.3
−5 deg angle of attack	1,560	822
+5 deg pitch angle	1,880	1,030
−5 deg pitch angle	79.3	9.5
+8 deg/s pitch rate	17.5	1.84
−8 deg/s pitch rate	5.7	1.0
+200 ft altitude	3,100	1,790

the perturbations in several cases are 100% of the initial state value. Figures 5.1 and 5.3 demonstrate the feedback nature of the controller. At the instant the aircraft "overshoots" the nominal trajectory, the throttle command attempts to reduce the aircraft velocity by switching from the upper to the lower limit. Immediately after the 20 s mark, the throttle setting also switches to the lower limit for a brief period of time. This allows the aircraft to rapidly pitch up to a moderate climb attitude. For most test cases though, the saturation of the throttle control leads to a steady altitude error.

Against expectations, Model A achieved slightly better results than Model B. This is possibly just an uncharacteristic occurance. In the graphs, the trajectory referred to as the "perturbed nominal" is the flight path the aircraft follows if the nominal control history is applied to the perturbed initial conditions. The perturbed nominal trajectory is not shown for the −80 ft/s velocity error case as it crashes a few seconds after takeoff.

During the testing it was observed that the best controller performance is ob-

Table 5.2: Controller I: Model B results

Perturbation	Maximum error (ft)	Mean error (ft)
+80 ft/s velocity	1,110	239
−80 ft/s velocity	1,570	248
+5 deg angle of attack	43.9	6.7
−5 deg angle of attack	2,000	1,070
+5 deg pitch angle	2,330	1,290
−5 deg pitch angle	80.9	14.4
+8 deg/s pitch rate	18.6	2.0
−8 deg/s pitch rate	10.0	1.2
+200 ft altitude	3,640	2,160

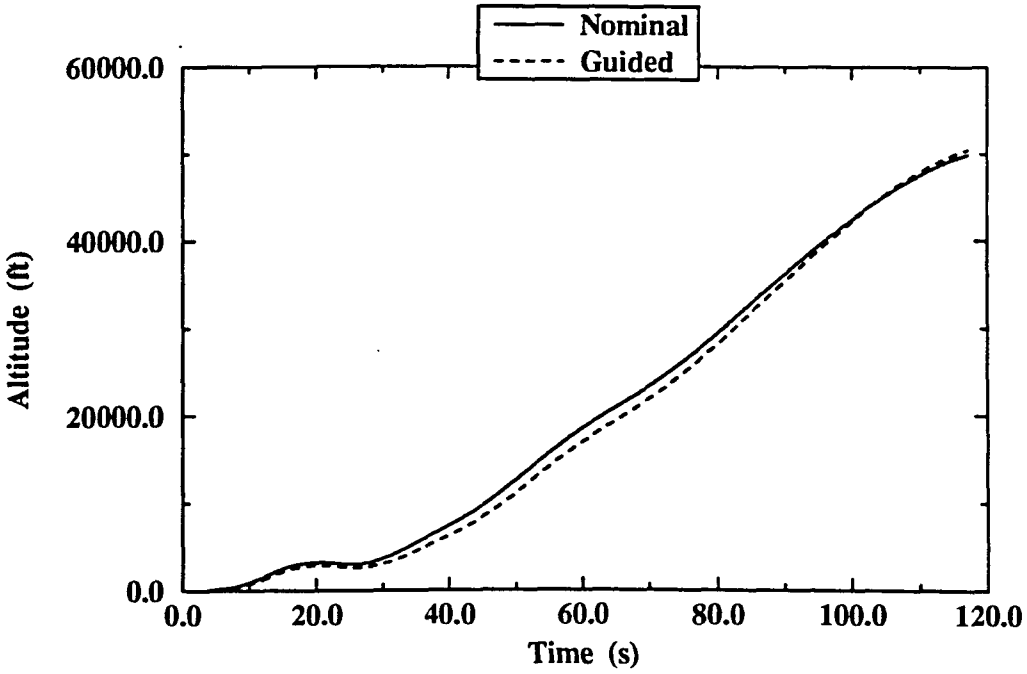

Figure 5.1: Controller I: Altitude history with −80 ft/s velocity perturbation

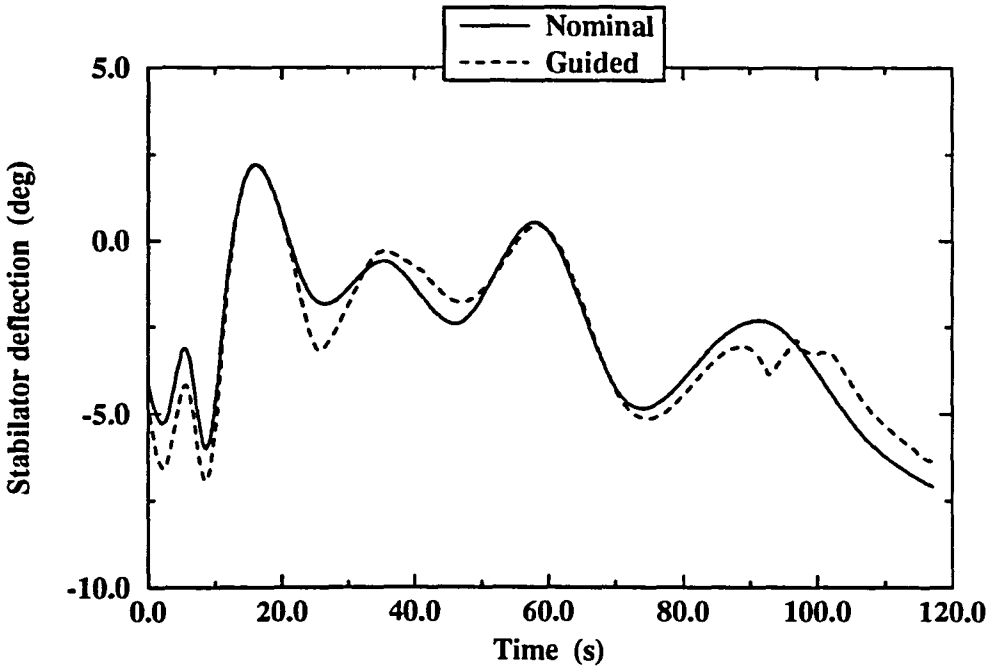

Figure 5.2: Controller I: Stabilator history with $-80\ ft/s$ velocity perturbation

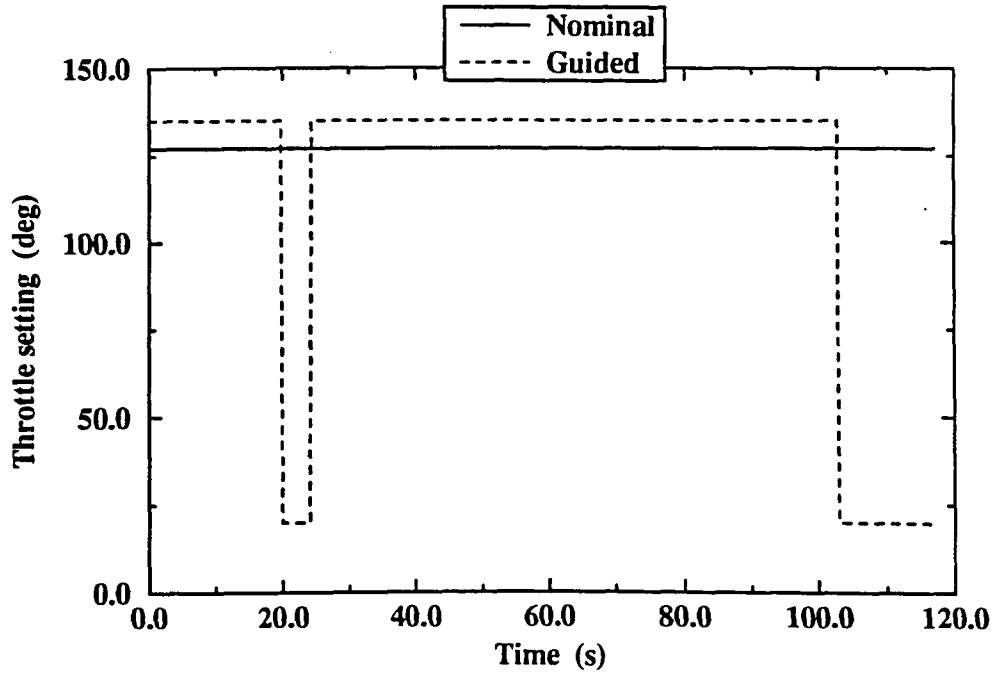

Figure 5.3: Controller I: Throttle history with $-80\ ft/s$ velocity perturbation

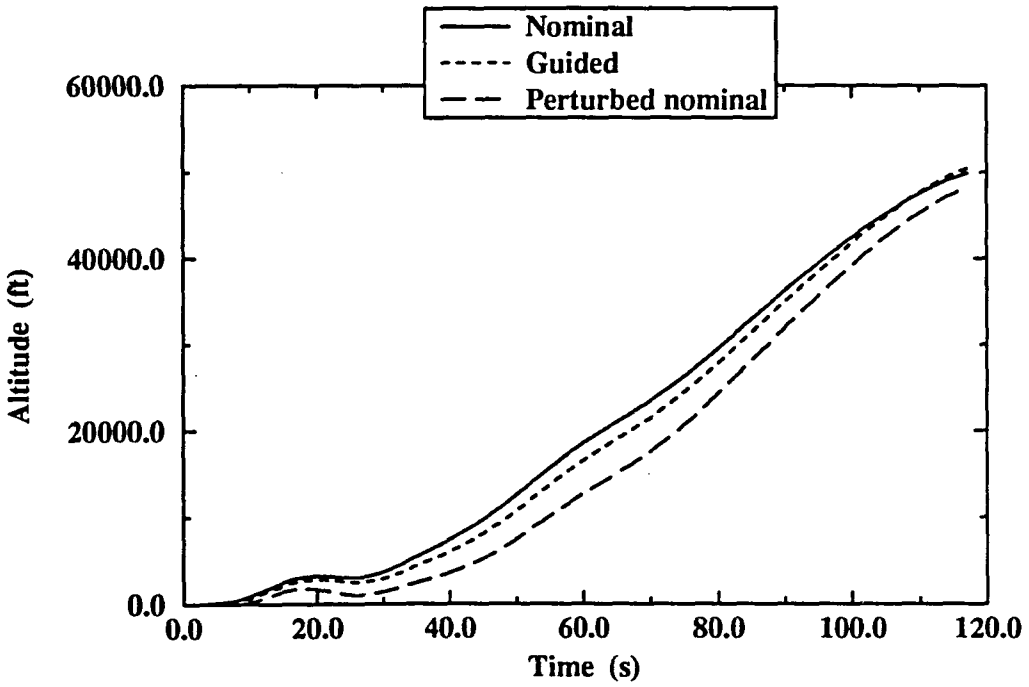

Figure 5.4: Controller I: Altitude history with -5 *deg* angle of attack perturbation

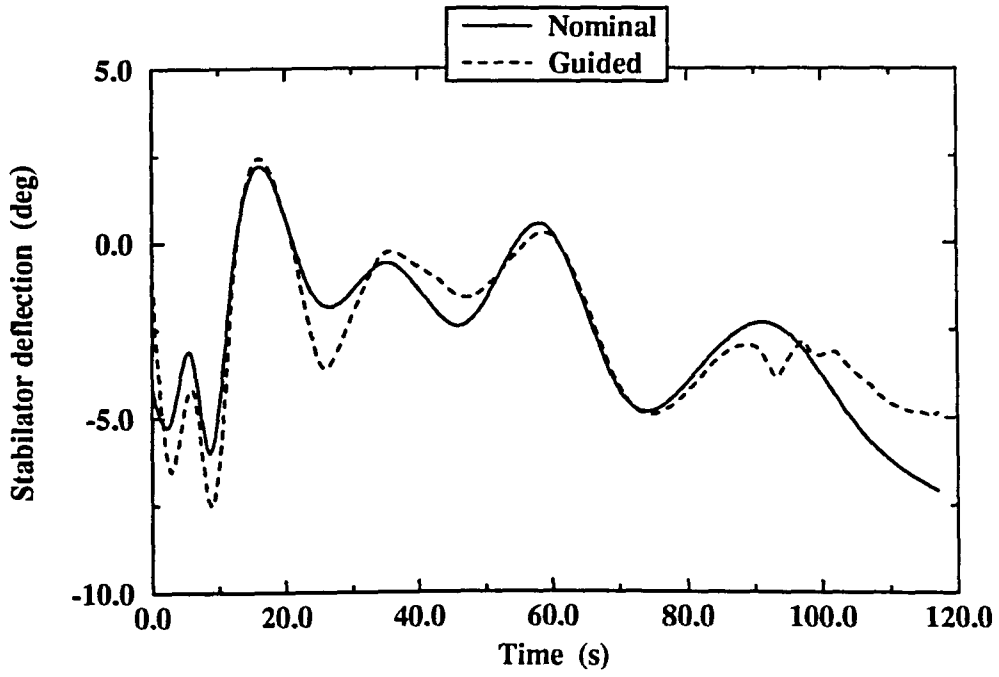

Figure 5.5: Controller I: Stabilator history with -5 *deg* angle of attack perturbation

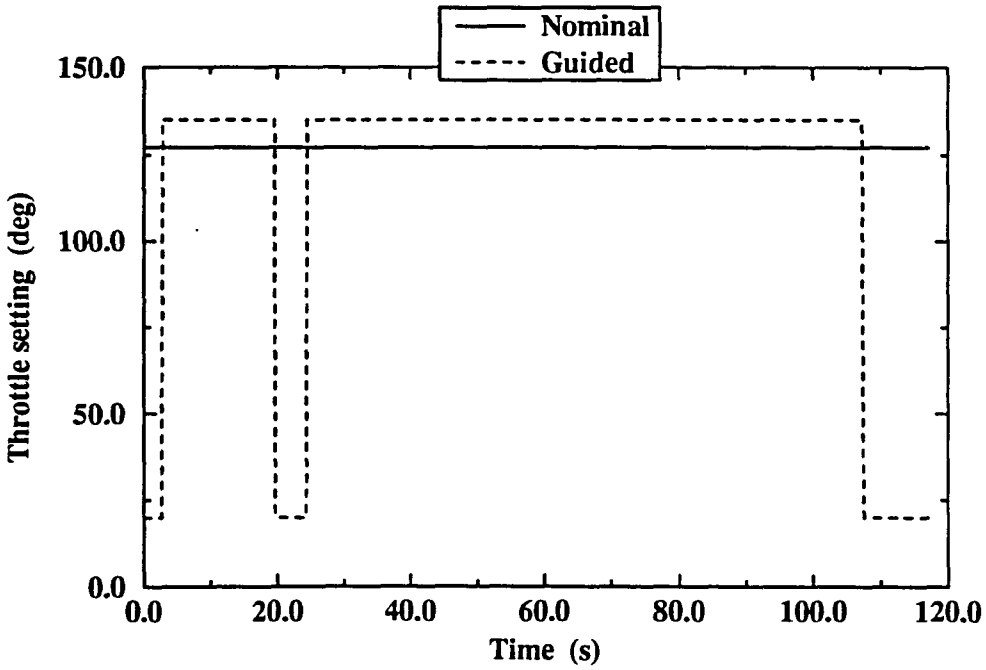

Figure 5.6: Controller I: Throttle history with -5 *deg* angle of attack perturbation

tained by:

1. Carefully choosing the $\hbar$ parameter.

2. Placing no penalty on the control (i.e., $R \equiv 0$).

3. Tracking only the altitude and the pitch angle errors.

Based on these observations, various modifications to the basic controller are made in the following sections.

5.2 Controller II: Optimal $\hbar$ Modification

During the preceding testing it was observed that the choice of the $\hbar$ parameter greatly effected the performance of the controller. For Controller I, $\hbar$ is taken to

Table 5.3: Controller II: Model A results

Perturbation	Maximum error (ft)	Mean error (ft)
$-80\ ft/s$ velocity	1,210	564
$-5\ deg$ angle of attack	1,350	655
$+5\ deg$ pitch angle	1,250	574
$+200\ ft$ altitude	1,130	860

be a constant through out the time-of-flight of the trajectory. Here the controller is modified to allow a time-varying $\hbar$. From the discussion in Section 5.1, we see that J_k in Equation 5.8 is a function of $\hbar$ with x_{1_k} and x_{2_k} approximated by Equations 5.7 and 5.5, and u_k specified by Equation 5.11. In this section, the $\hbar$ parameter is optimally chosen to minimize the performance index J_k at each instant in time. The one-dimensional minimization is performed by using Brent's algorithm [24]. This modified controller is denoted Controller II.

The "worst" cases from the Controller I testing are repeated here. The results are given in Tables 5.3 and 5.4. The altitude and control histories of two Model B test cases are shown in Figures 5.7–5.12. If the $\hbar$ parameter is allowed to be the true optimal of J_k, at each instant in time, the throttle setting frequently switches between the upper and lower extremes. By bounding the parameter in a reasonable range, this "bang-bang" effect is significantly reduced. A fluctuation in the throttle setting however continues to occur due the $\hbar$ optimization. With the exception of one case, Controller II shows a notable improvement over Controller I. This enhancement in the tracking performance is very evident when comparing Figures 5.4 and 5.10.

Table 5.4: Controller II: Model B results

Perturbation	Maximum error (ft)	Mean error (ft)
$-80\ ft/s$ velocity	1,570	796
$-5\ deg$ angle of attack	543	182
$+5\ deg$ pitch angle	1,810	892
$+200\ ft$ altitude	1,220	544

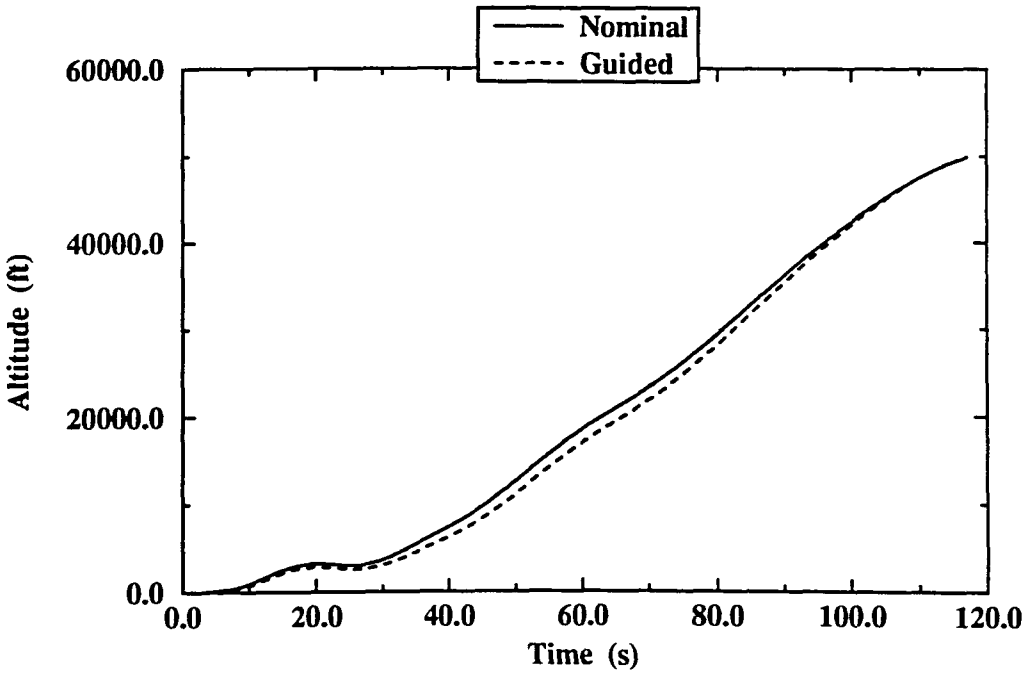

Figure 5.7: Controller II: Altitude history with $-80\ ft/s$ velocity perturbation

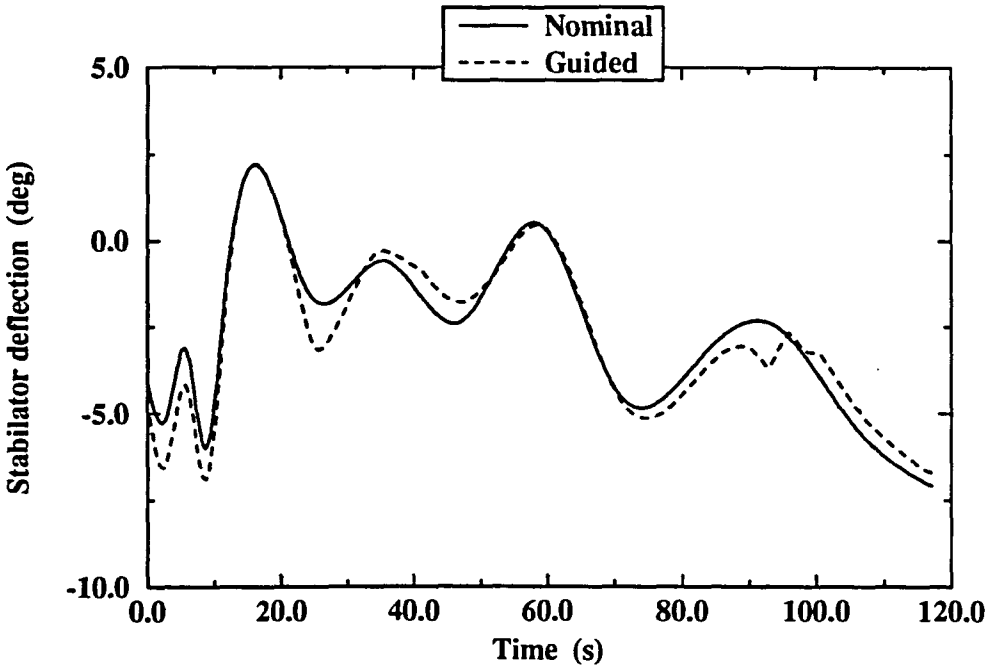

Figure 5.8: Controller II: Stabilator history with -80 ft/s velocity perturbation

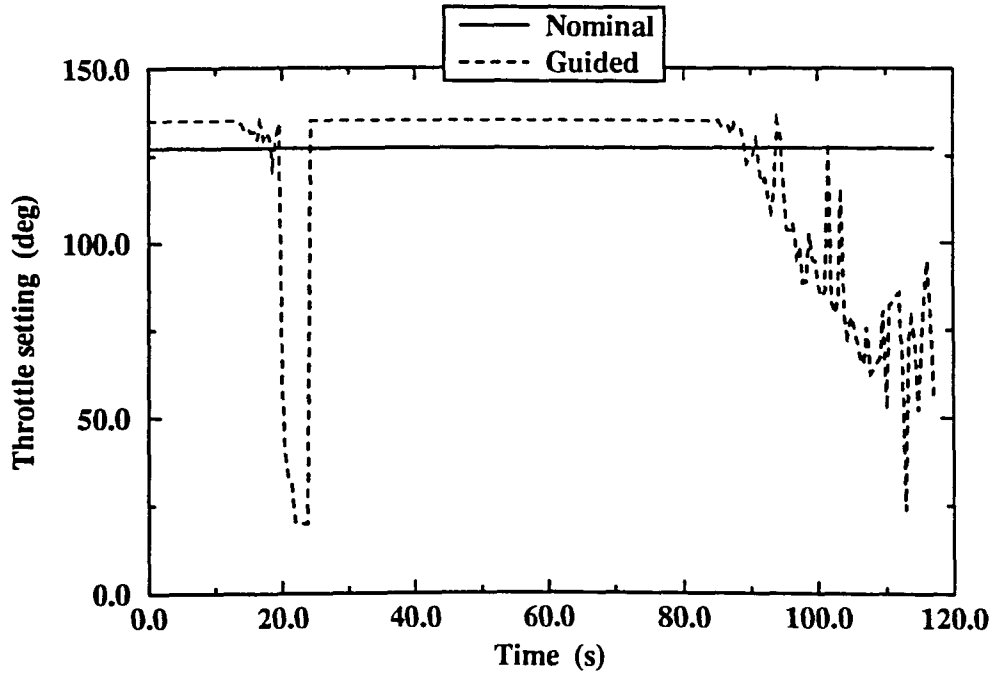

Figure 5.9: Controller II: Throttle history with -80 ft/s velocity perturbation

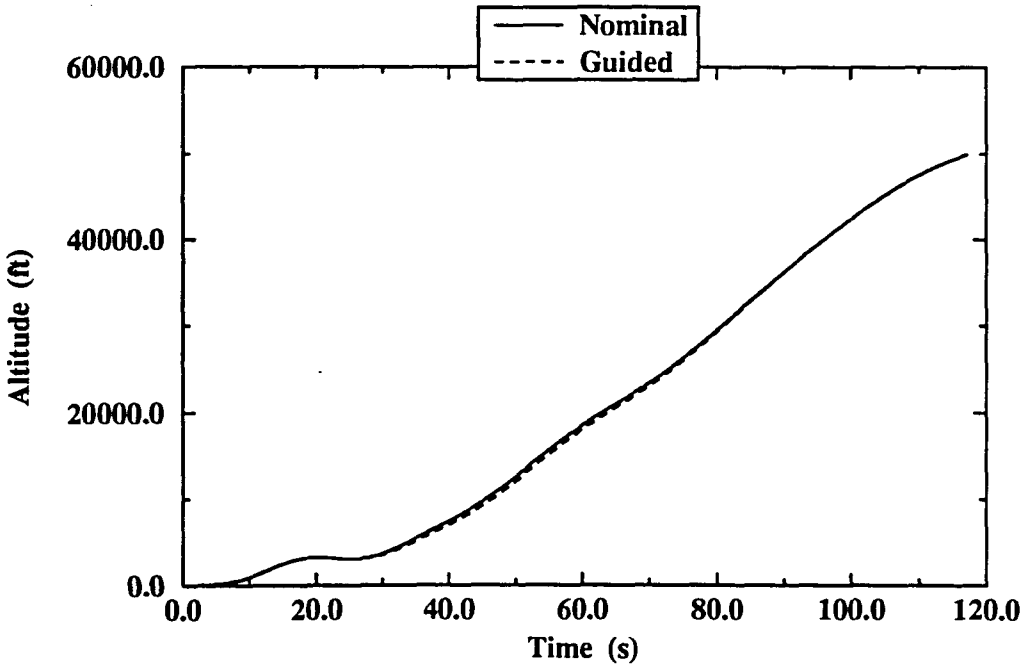

Figure 5.10: Controller II: Altitude history with -5 *deg* angle of attack perturbation

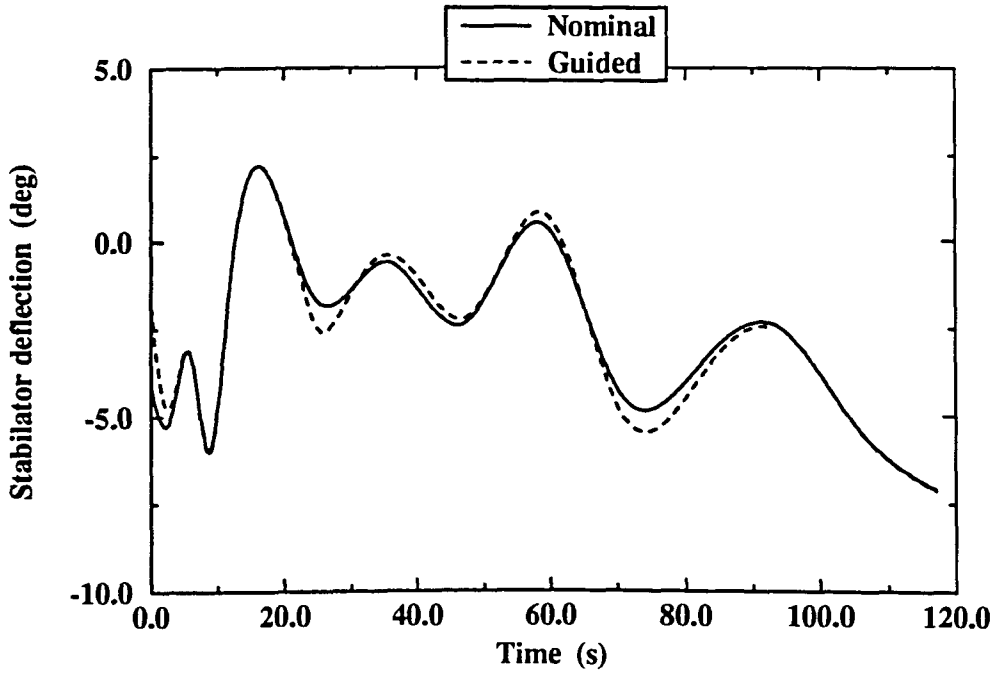

Figure 5.11: Controller II: Stabilator history with -5 *deg* angle of attack perturbation

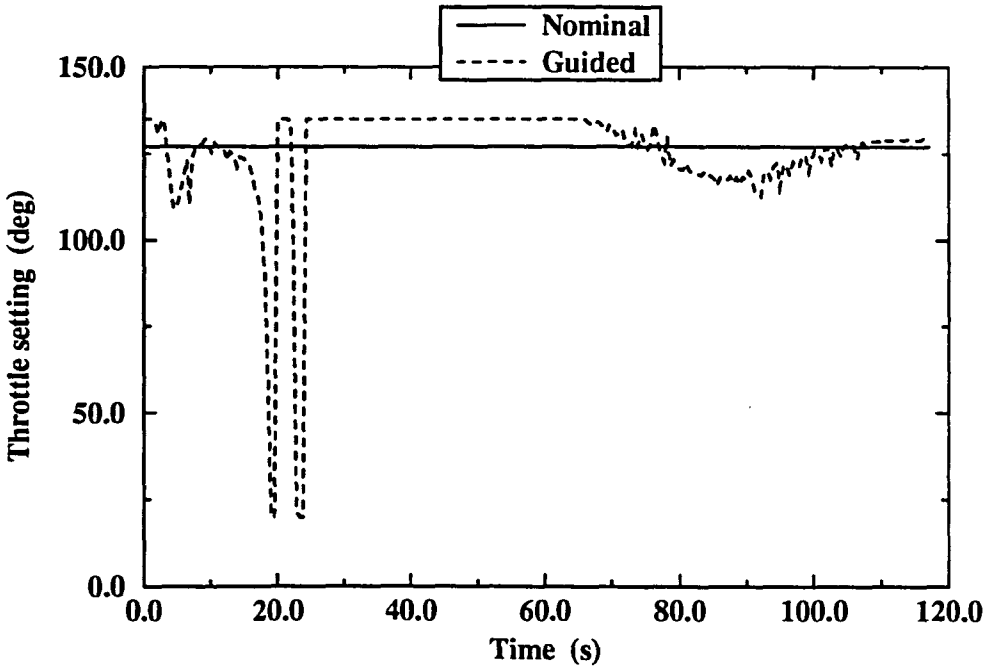

Figure 5.12: Controller II: Throttle history with -5 *deg* angle of attack perturbation

5.3 Controller III: Error Rate Modification

The Controller I testing showed that the best tracking is obtained by penalizing only x_1 terms in the performance index. Keeping this in mind, the performance index can be extended to include the error rate $\dot{e}_1$. Therefore,

$$J_k = \frac{1}{2} e_{1_{k+1}}^T Q_1 e_{1_{k+1}} + \frac{1}{2} \dot{e}_{1_{k+1}}^T Q_2 \dot{e}_{1_{k+1}} \tag{5.20}$$

where Q_2 is a positive semi-definite matrix. The rational for this modification is that incorporating the error rate gives the controller anticipatory information for future errors. This additional information may further improve the response of the controller. Repeating the procedure carried out to derive Controller I, the following control law is obtained.

$$u(t) = -W^{-1} \left\{ \frac{\hbar^2}{2} G^T Q_1 P_1 + \hbar G^T Q_2 P_2 \right\} \tag{5.21}$$

Table 5.5: Controller III: Model A results

Perturbation	Maximum error (ft)	Mean error (ft)
$-80\ ft/s$ velocity	487	433
$-5\ deg$ angle of attack	167	152
$+5\ deg$ pitch angle	249	126
$+200\ ft$ altitude	251	89.0
$+1{,}000\ ft$ altitude	1,000	218

where the following substitutions have been made:

$$G = F_{12}B(x) \tag{5.22}$$

$$W = G^T\left(\frac{\hbar^4}{4}Q_1 + \hbar^2 Q_2\right)G \tag{5.23}$$

$$P_1 = e_1 + \hbar\dot{e}_1 + \frac{\hbar^2}{2}P \tag{5.24}$$

$$P_2 = \dot{e}_1 + \hbar P \tag{5.25}$$

$$P = F_{11}f_1(x) + F_{12}f_2(x) - \ddot{s}_1 \tag{5.26}$$

This control law is denoted Controller III. The test results for the controller are given in Tables 5.5 and 5.6. Figures 5.13–5.18 display the altitude and control histories of two Model B test cases. The results show a marked reduction in the maximum and mean errors compared to Controller I. In some cases the errors have been reduced by as much as 90%. Combining Controllers II and III is not expected to further reduce the errors as the control inputs are virtually saturated when the reference trajectory is not exactly followed. To demonstrate the potential of this controller, it is tested with very large initial perturbations (e.g. a $-20\ deg$ angle of attack perturbation). Even in these extreme cases, the largest deviation is only 2.0% of the final altitude.

Table 5.6: Controller III: Model B results

Perturbation	Maximum error (ft)	Mean error (ft)
-80 ft/s velocity	480	429
-5 deg angle of attack	183	167
-20 deg angle of attack	547	460
$+5$ deg pitch angle	211	191
$+50$ deg pitch angle	628	520
$+200$ ft altitude	399	147
$+1,000$ ft altitude	1,000	206

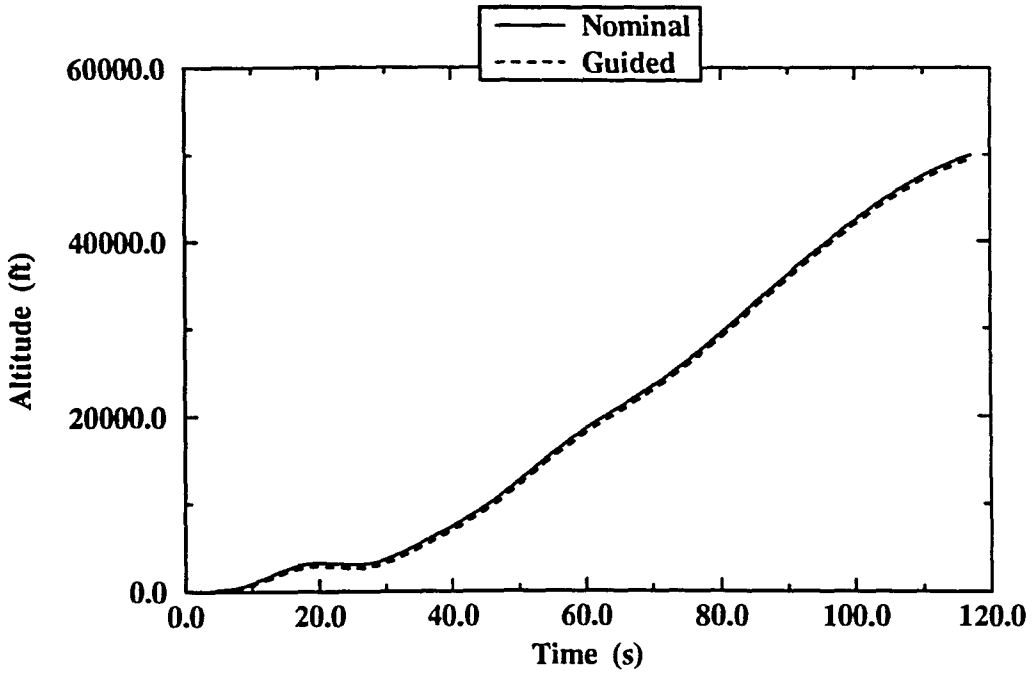

Figure 5.13: Controller III: Altitude history with -80 ft/s velocity perturbation

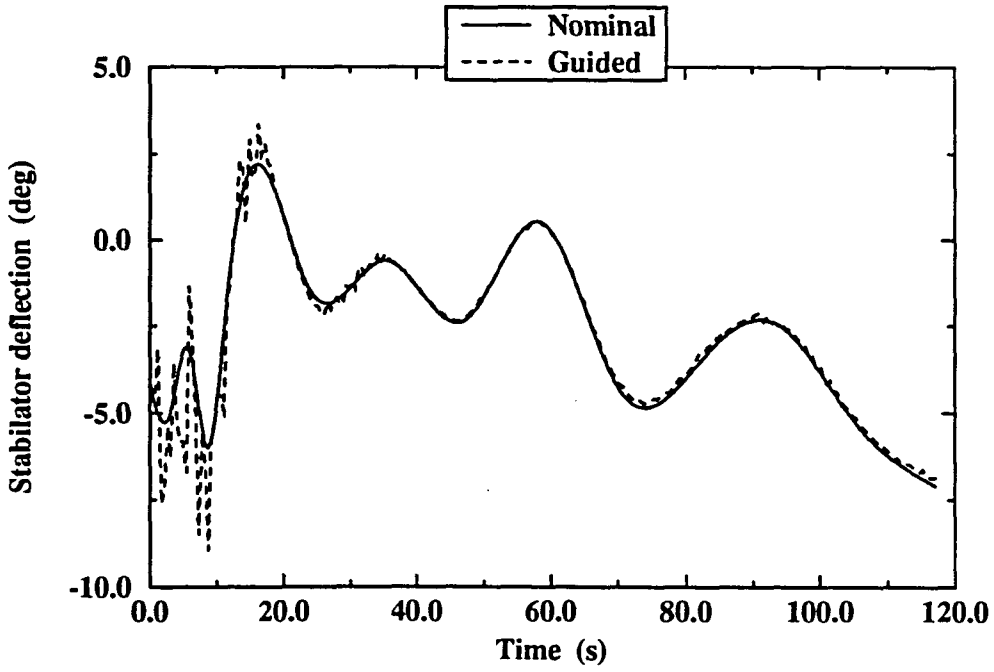

Figure 5.14: Controller III: Stabilator history with $-80\ ft/s$ velocity perturbation

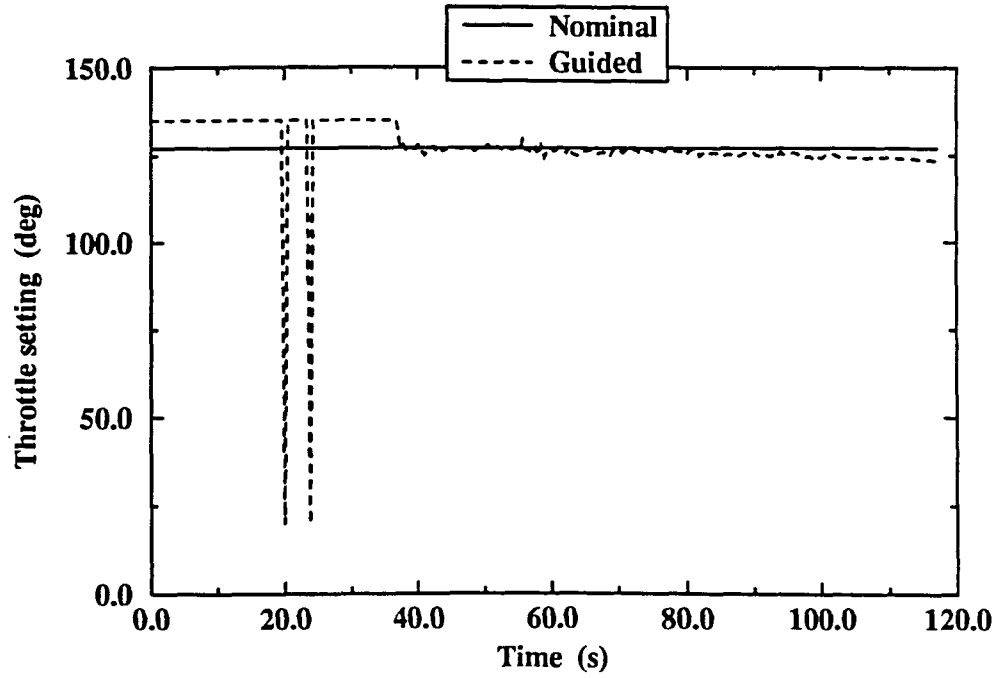

Figure 5.15: Controller III: Throttle history with $-80\ ft/s$ velocity perturbation

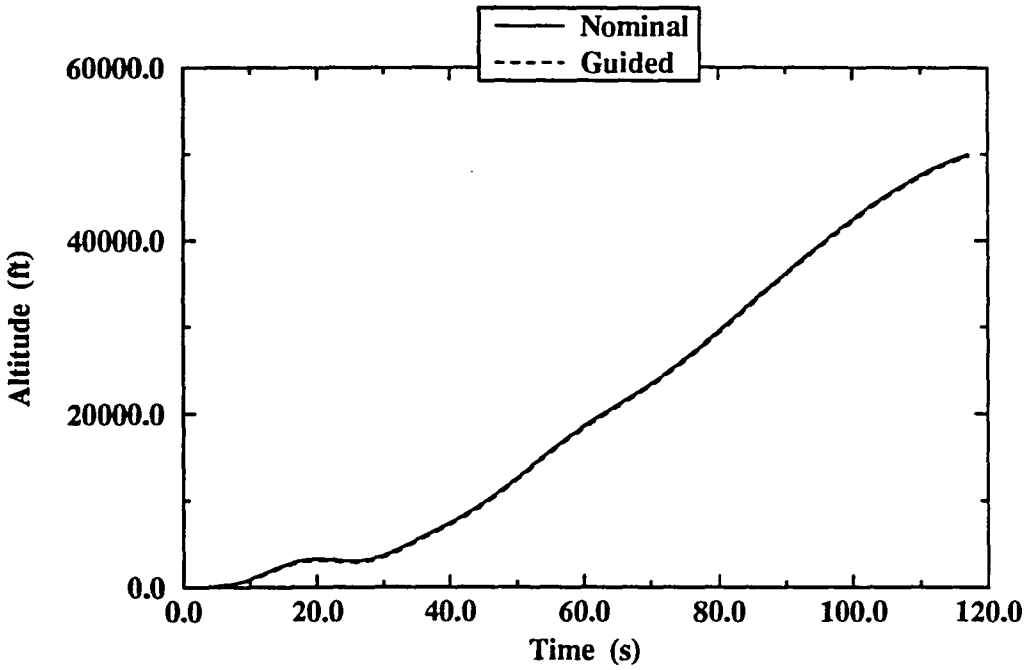

Figure 5.16: Controller III: Altitude history with -5 *deg* angle of attack perturbation

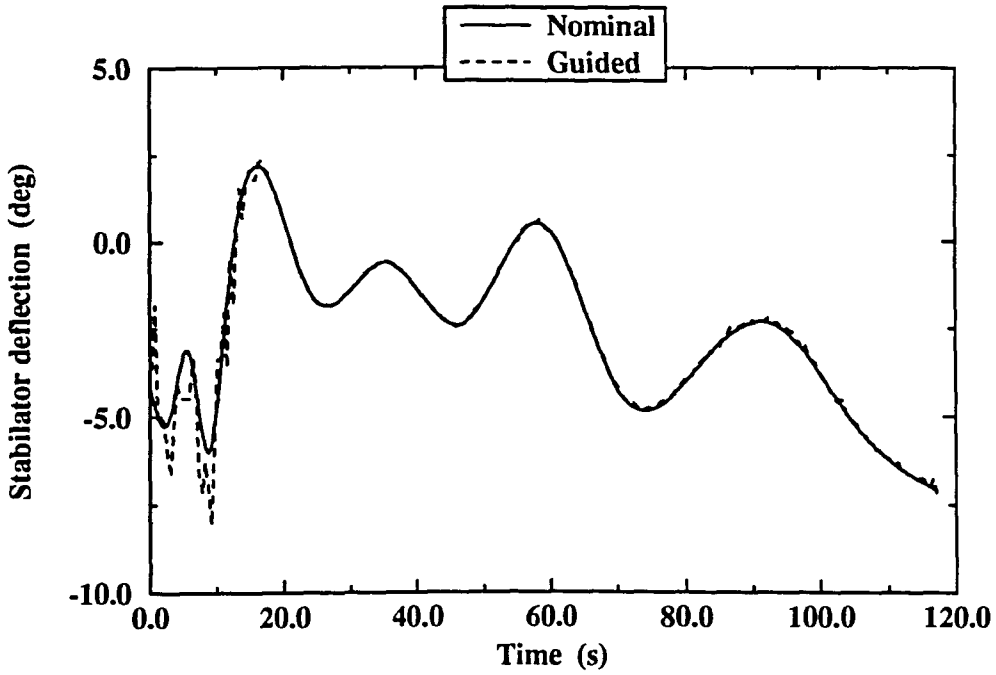

Figure 5.17: Controller III: Stabilator history with -5 *deg* angle of attack perturbation

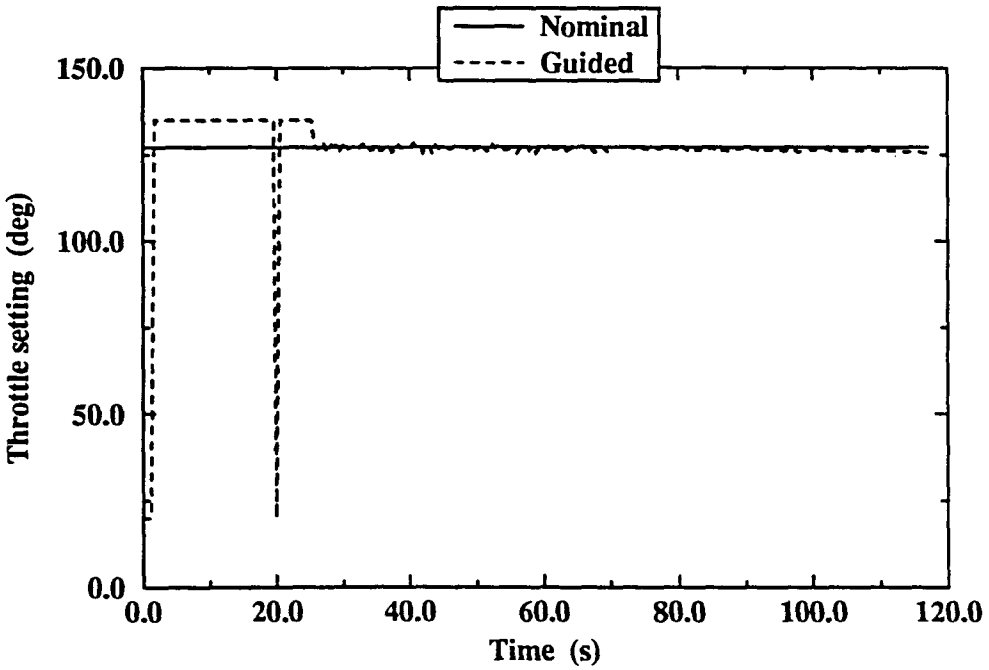

Figure 5.18: Controller III: Throttle history with -5 *deg* angle of attack perturbation

5.4 Controller Robustness

In the presence of system uncertainties and disturbances, it is desired that the controller maintain satisfactory performance. The feedback nature of the controller is expected to provide a certain degree of robustness. Some robustness properties of the controller have been established by Lu [23]. To evaluate this robustness, the tracking ability of the controller is first tested in the presence of wind. The velocity of the wind is modeled as a function of altitude and its direction is horizontal. The magnitude of the wind speed is given by,

$$W_x = \begin{cases} 80\ cos\left(\frac{h\pi}{10{,}000}\right) & if\ h \leq 25{,}000\ ft \\ 0.0 & otherwise \end{cases} \tag{5.27}$$

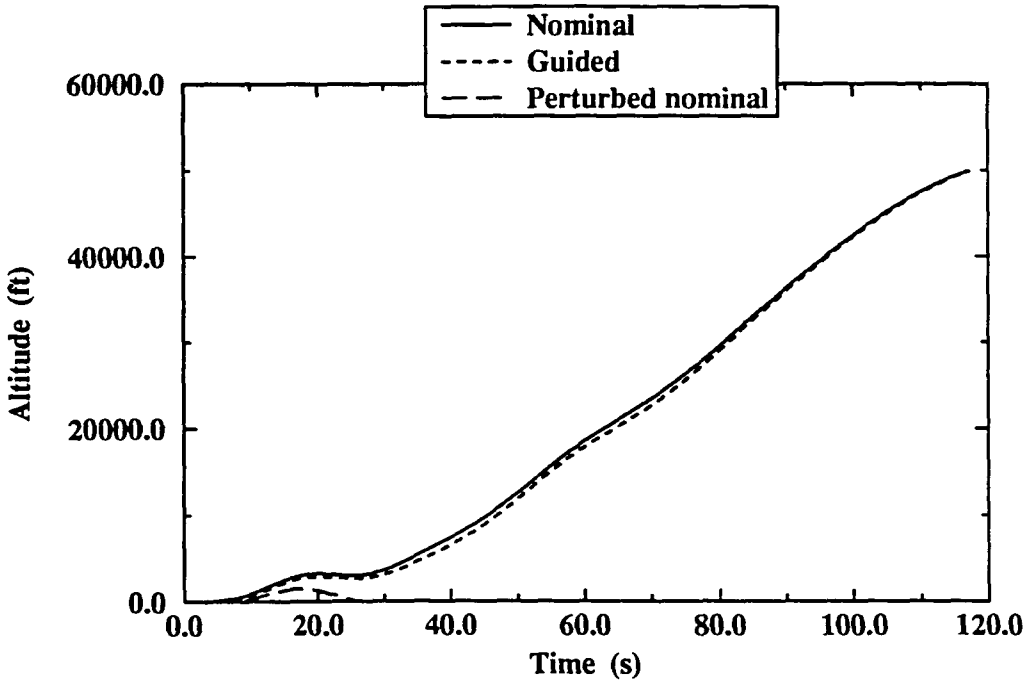

Figure 5.19: Controller III: Wind disturbed altitude history

To account for the presence of wind, several terms need to be added to the dynamic equations of the aircraft model [25]. The modified equations are as follows.

$$\dot{V} = [T\,cos\alpha - D - mgsin(\theta - \alpha)]/m - \dot{W}_x\,cos(\theta - \alpha) \quad (5.28)$$

$$\dot{\alpha} = [-T\,sin\alpha - L + mgcos(\theta - \alpha)]/Vm - \dot{W}_x\,sin(\theta - \alpha)/V + q \quad (5.29)$$

$$\dot{x} = V\,cos\alpha + W_x \quad (5.30)$$

where,

$$\dot{W}_x = \frac{\partial W_x}{\partial h}\dot{h} \quad (5.31)$$

The model represents a strong 80 ft/s (55 mph) wind that modulates from tailwind to headwind. The initial relative velocity of the aircraft is taken to be the nominal inertial velocity minus the 80 ft/s tailwind. The reference trajectory is the one used in the preceding sections. The test is conducted using Controller III on

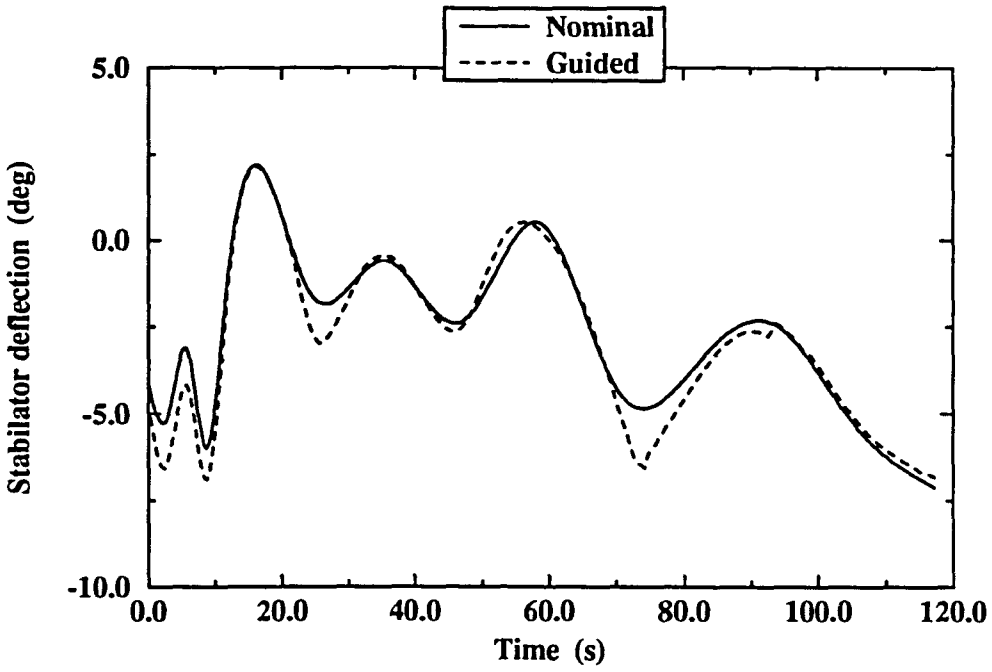

Figure 5.20: Controller III: Wind disturbed stabilator history

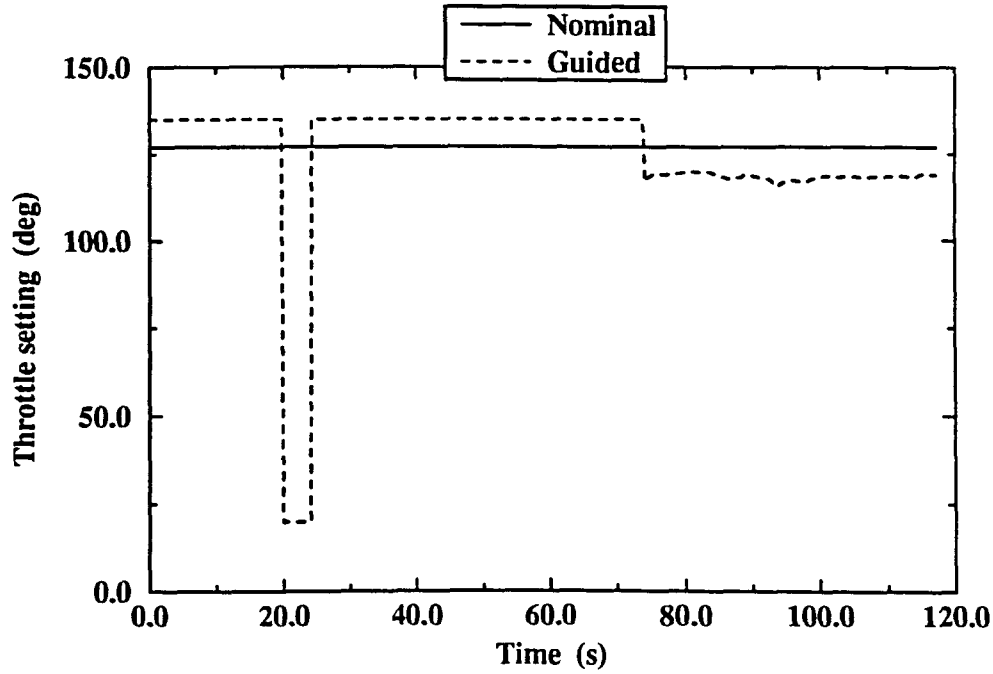

Figure 5.21: Controller III: Wind disturbed throttle history

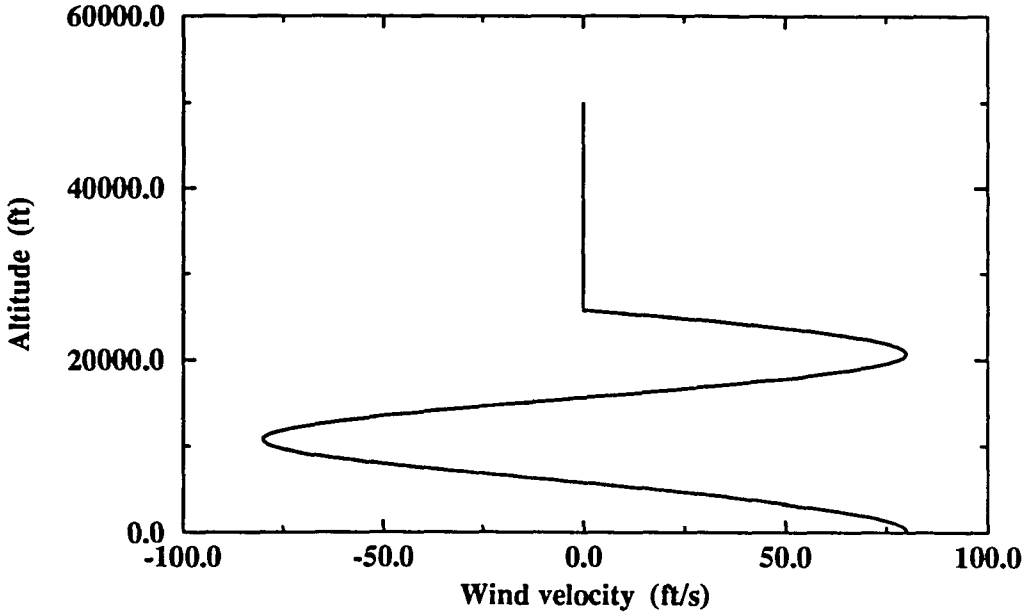

Figure 5.22: Wind profile

Model B. The controller, which is based on the nominal model, is given no information regarding the wind. The resulting trajectory and control histories are displayed in Figures 5.19–5.22. The guided trajectory has a maximum error of 857 ft and a mean error of 460 ft. These errors are very small when compared to the final altitude the aircraft climbs to. Notice that without the guidance of the controller, the perturbed nominal trajectory crashes shortly after takeoff. The test was repeated using an optimal $\hbar$ in combination with Controller III. This resulted in the same maximum error but a reduced mean error of 403 ft.

To evaluate the performance of the controller in the face of system uncertainties, the controller is tested with errors in the aerodynamic coefficients. The exercise is performed assuming that a $\pm 30\%$ error exits in all the stored aerodynamic data. The error is taken to simultaneously occur in the coefficient of lift, the coefficient of drag and the coefficient of pitching moment. The testing is executed on Model B,

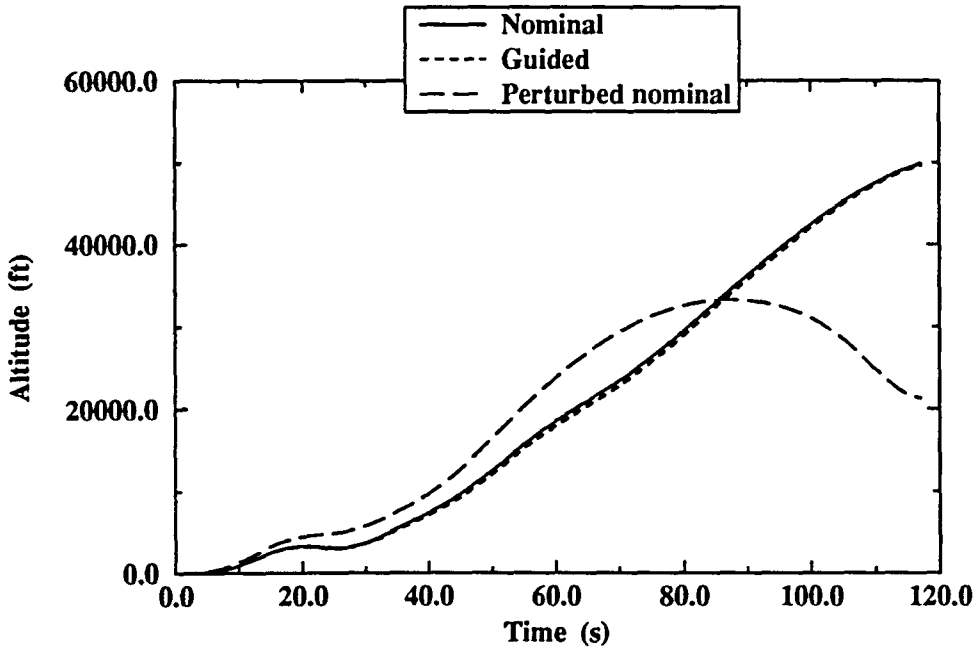

Figure 5.23: Controller III: +30% aerodynamic coefficient perturbation altitude history

using Controller III. The results are shown in Figures 5.23–5.28. The +30% case has a maximum error of 628 ft and a mean error of 346 ft. The −30% case has a larger maximum error of 1,170 ft and a mean error of 436 ft. These results show the impressive ability of the controller to track the reference trajectory even in the presence of large system uncertainties. The control histories show considerable modulation about the nominal. This is due to the controller attempting to reduce the error, but continuously 'over-shooting' because of the erroneous information. This fluctuation in the control setting is not excessive as it falls well below modern actuator rate limits.

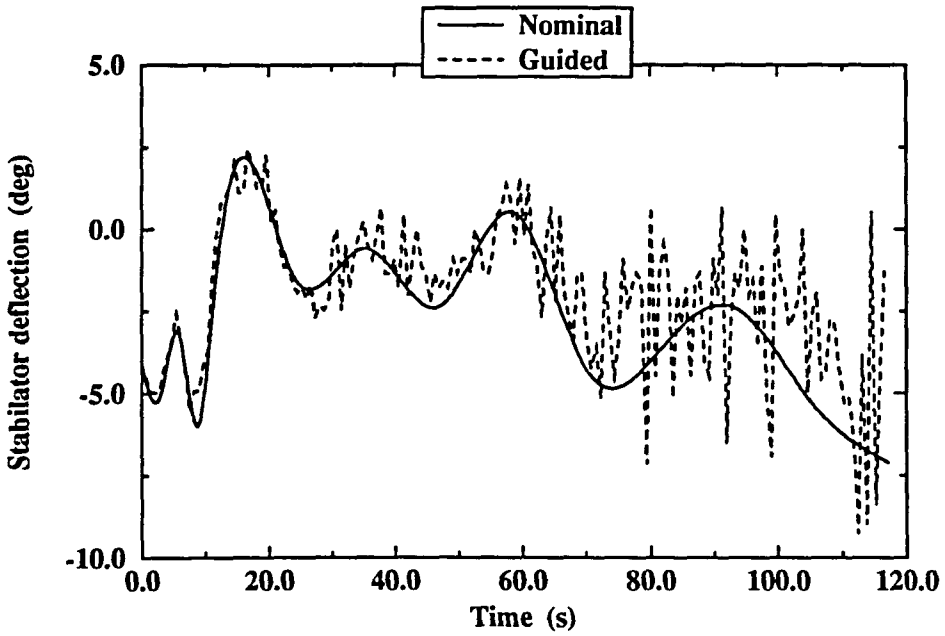

Figure 5.24: Controller III: +30% aerodynamic coefficient perturbation stabilator history

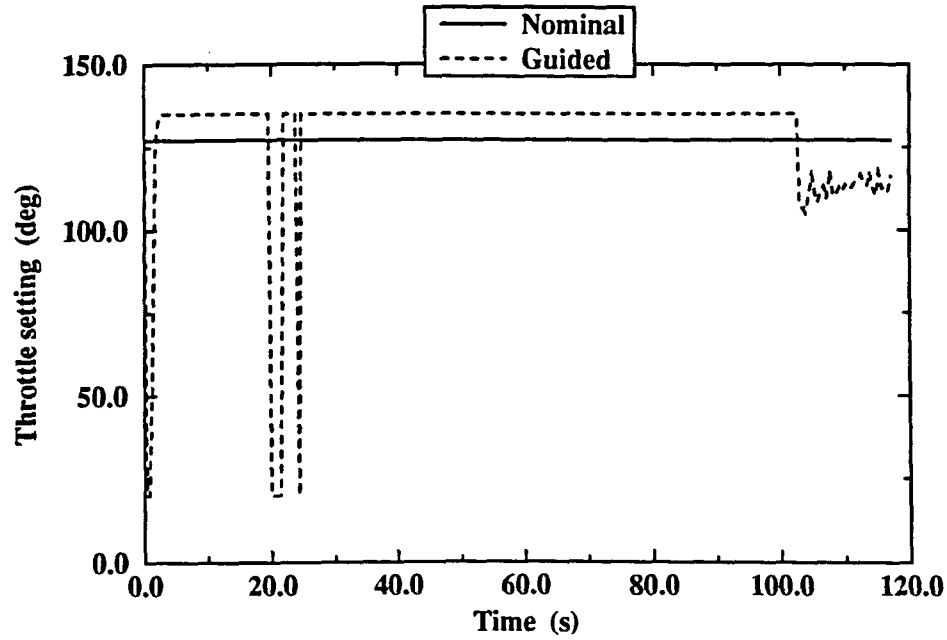

Figure 5.25: Controller III: +30% aerodynamic coefficient perturbation throttle history

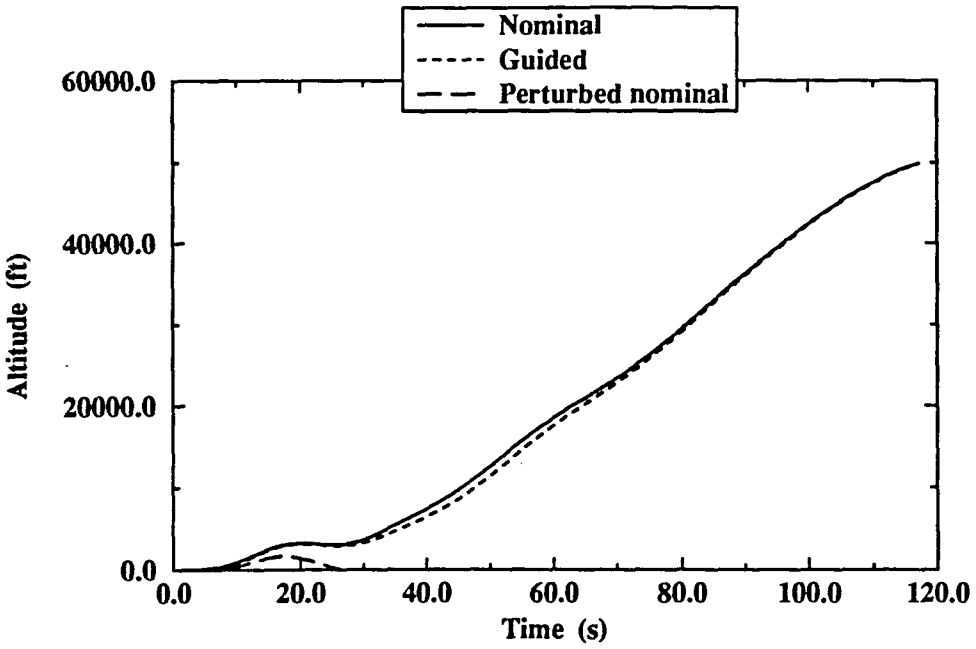

Figure 5.26: Controller III: −30% aerodynamic coefficient perturbation altitude history

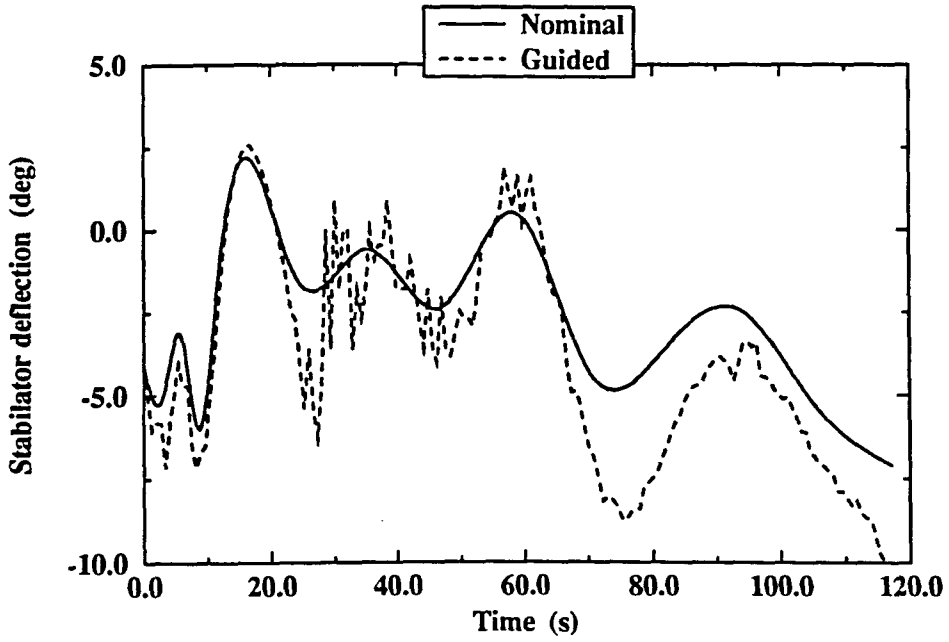

Figure 5.27: Controller III: −30% aerodynamic coefficient perturbation stabilator history

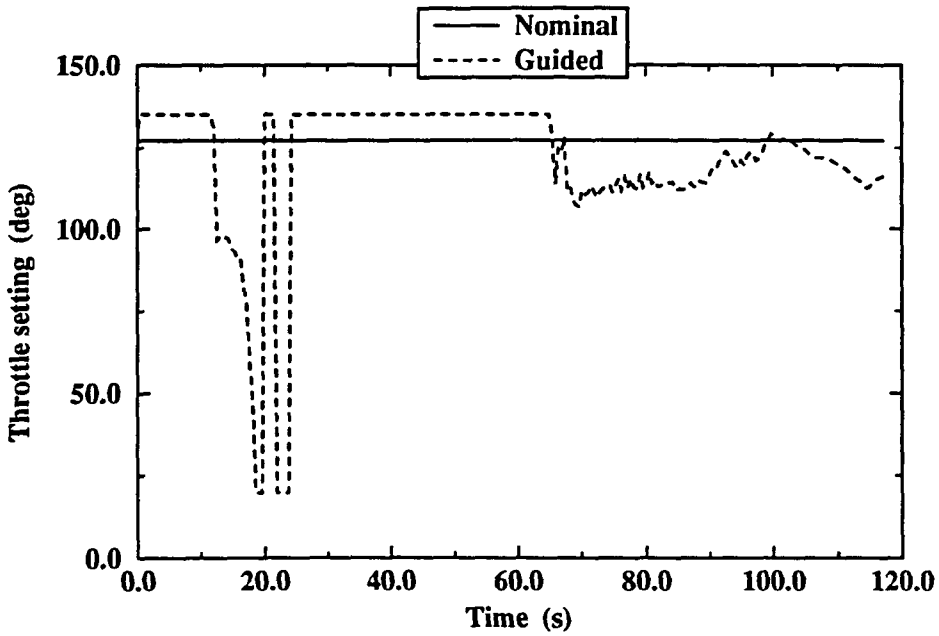

Figure 5.28: Controller III: −30% aerodynamic coefficient perturbation throttle history

5.5 Trajectory Design

In the preceding sections the controller is used to track a well-defined nominal trajectory that is prior obtained. The architecture of the controller is such that no knowledge of the nominal control is required. Hence, by defining desired reference state histories and penalizing appropriate terms in the weighting matrices, Q_1 and Q_2, the controller can be used to generate nominal control histories. This allows the design of virtually any trajectory.

As mentioned in Chapter 1, the aircraft model was originally intended for use in the 1991–92 Controls Design Challenge [7]. With the ability to design trajectories, these controllers can be used to meet a number of the requirements of the controls challenge. The tasks addressed here are:

1. Beginning at $9,800\ ft$ and Mach 0.5, climb $200\ ft$ and hold altitude and Mach number, simultaneously.

2. Beginning at $39,800\ ft$ and Mach 1.4, climb $200\ ft$ and hold altitude and Mach number, simultaneously.

3. Accelerate from Mach 0.5 to Mach 1.4 while holding altitude at $30,000\ ft$.

The challenge requires the conditions to be met within an altitude error of $\pm 50\ ft$ and a Mach number error of within ± 0.01.

As a first step, the tasks 1 and 2 are attempted as only altitude hold exercises. This requires the tracking of two state variables, altitude and an angle or an angular rate. An angular measurement must be tracked to give the aircraft an attitude reference. The reference angle is chosen to be the aircraft pitch. It was found that satisfactory performance can obtained by tracking any constant pitch angle. The choice of the reference altitude history though is not as trivial. If the desired final altitude is taken as the reference trajectory, the large initial error results in excessive control saturation. This problem can be avoided by choosing a reference altitude history given by the following equation.

$$h_{ref} = (h_i - h_f)e^{-t/\tau} + h_f \tag{5.32}$$

Reference state histories chosen in this fashion have the advantage of yielding no errors at the initial and final time. A time constant τ of $1.5\ s$ proved to be adequate. For the complete task of holding both altitude and Mach number, a third state variable, velocity must be tracked. The reference velocity is taken to be the velocity at the current altitude that corresponds to the required Mach number. As there

exists a unique trim pitch angle for holding a specified altitude and Mach number, the reference angle must be the trim pitch angle. This angle is computed in advance by solving the respective dynamic equations. Like altitude, creating a reference pitch angle history in the form of Equation 5.32 improves the controller response.

The third task similarly requires the tracking of altitude, pitch and velocity. In this case, the reference altitude is a constant $30,000\ ft$. The reference velocity and pitch histories are taken to be exponential functions in the form of Equation 5.32, with time constants of 20 s. The final reference pitch angle is the trim condition value and is obtained by solving the dynamic equations at the desired final Mach number and altitude.

Figures 5.29–5.40 display the the resulting state and control histories for the three tasks. All the solutions fall well within the challenge requirements. This demonstrates the remarkable ability of the controller to follow arbitrarily designed trajectories

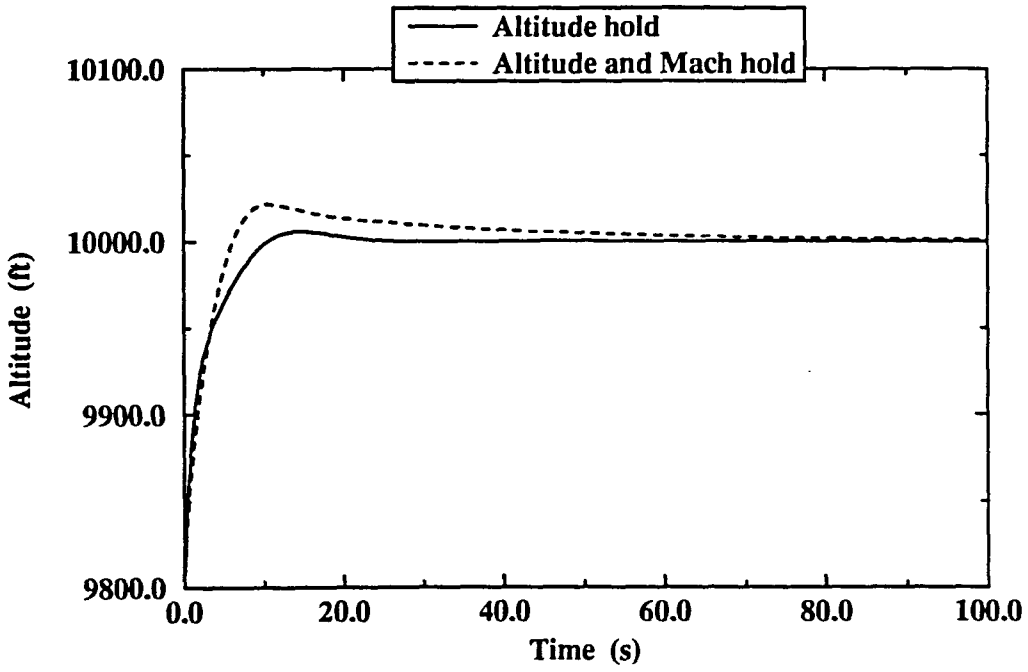

Figure 5.29: Altitude history for Mach hold trajectory (at 10,000 ft)

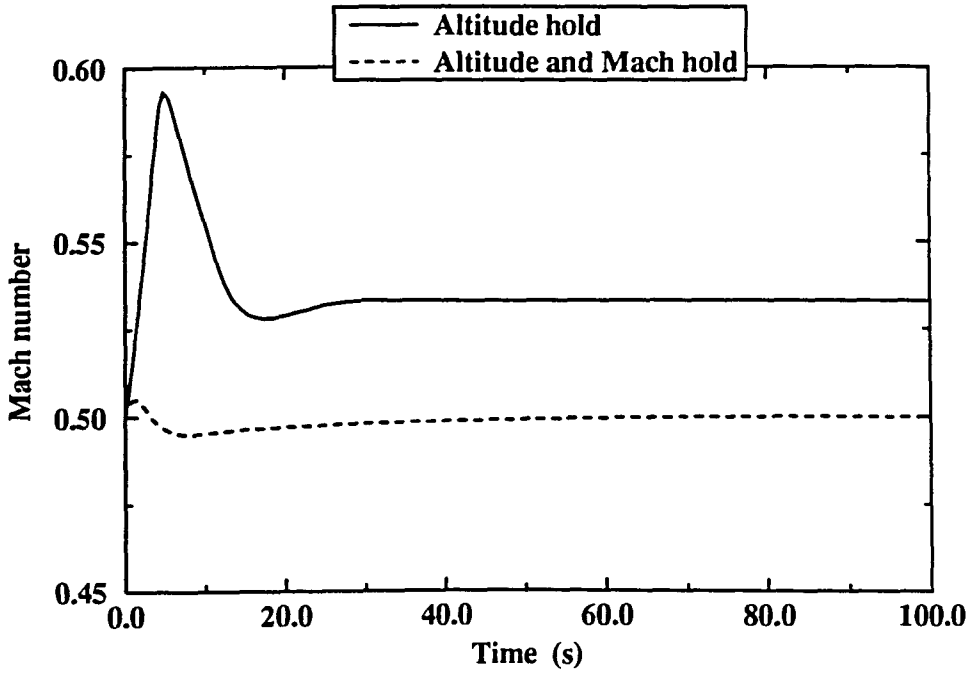

Figure 5.30: Mach history for Mach hold trajectory (at 10,000 ft)

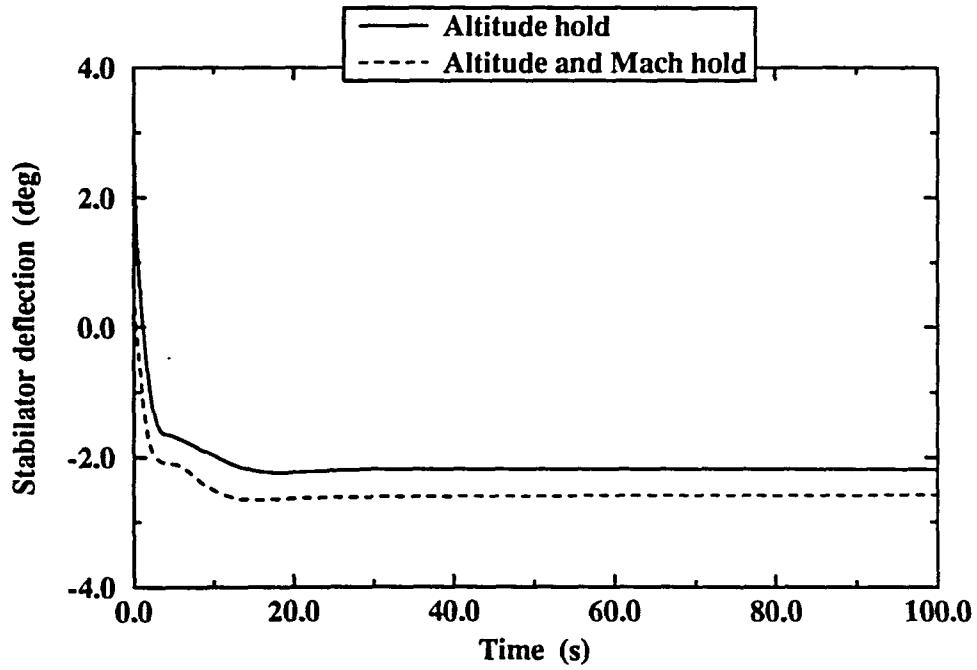

Figure 5.31: Stabilator history for Mach hold trajectory (at 10,000 ft)

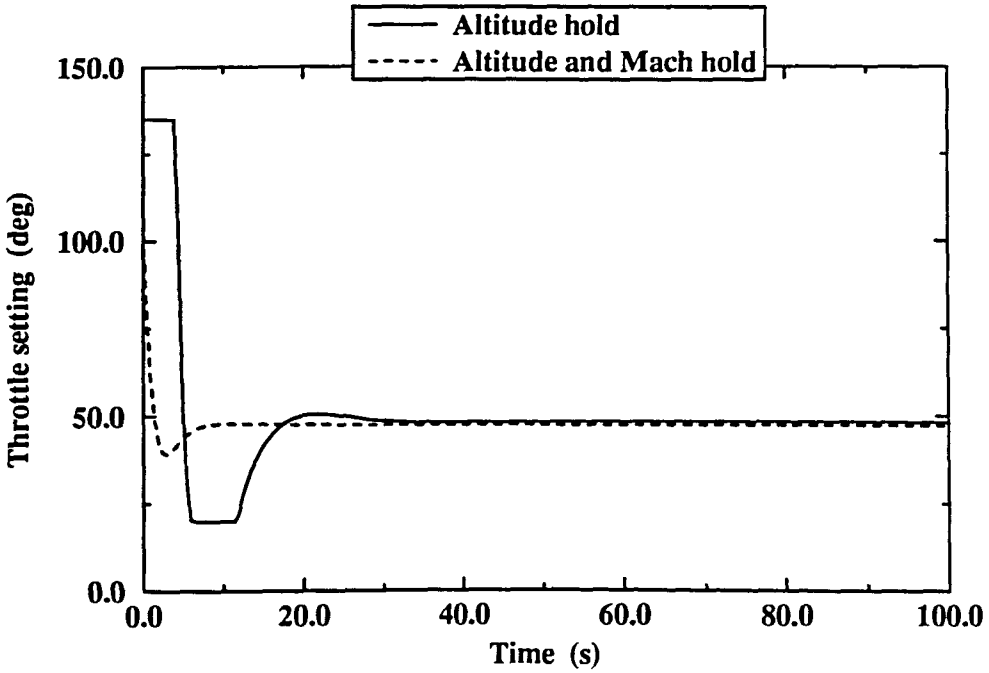

Figure 5.32: Throttle history for Mach hold trajectory (at 10,000 ft)

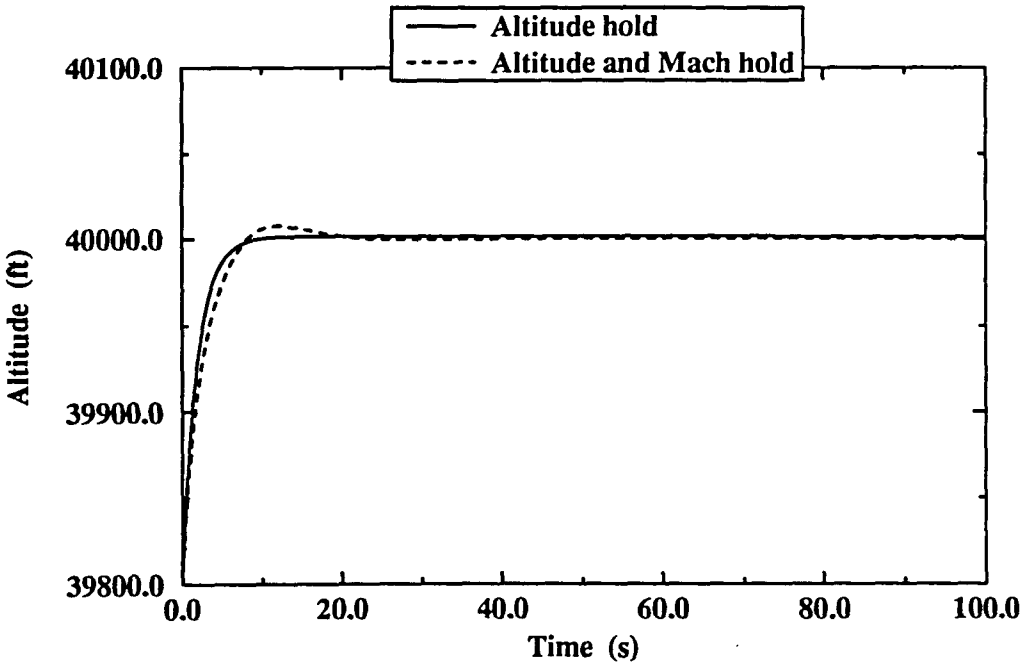

Figure 5.33: Altitude history for Mach hold trajectory (at 40,000 ft)

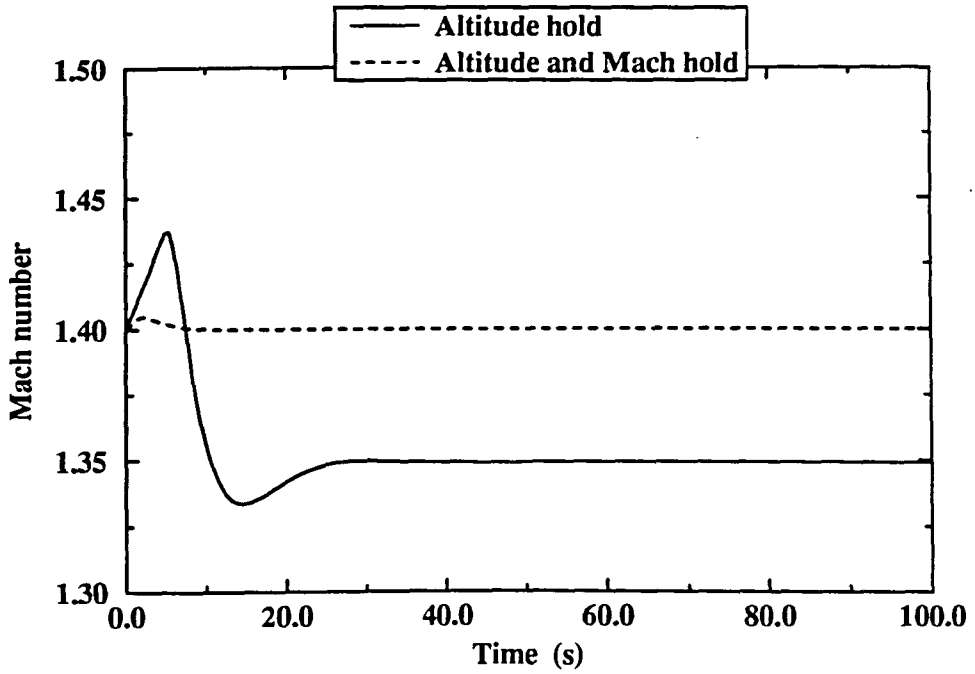

Figure 5.34: Mach history for Mach hold trajectory (at 40,000 ft)

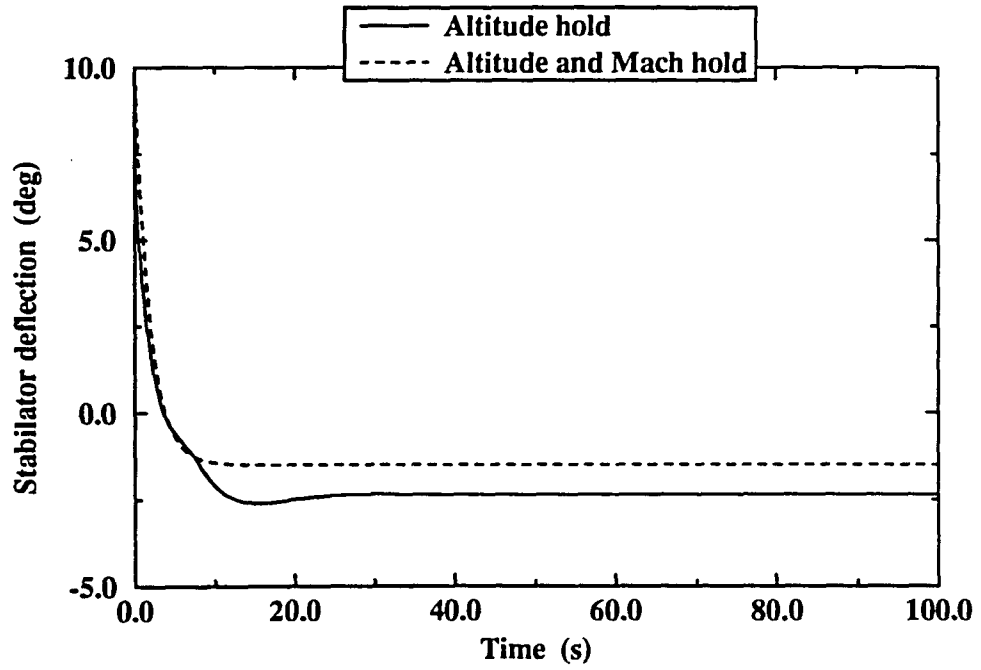

Figure 5.35: Stabilator history for Mach hold trajectory (at 40,000 ft)

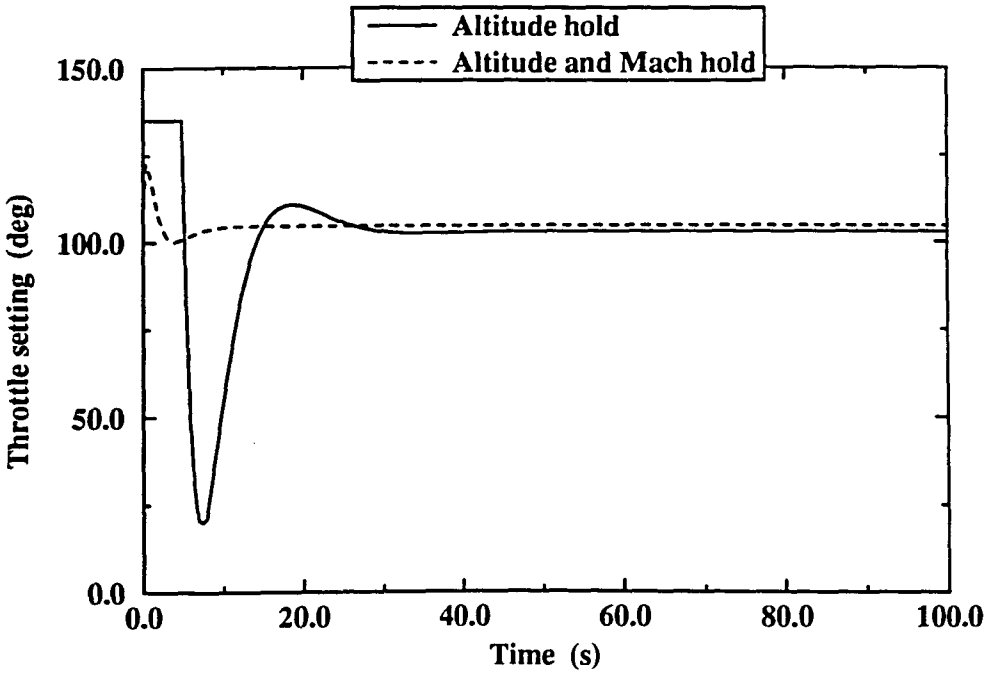

Figure 5.36: Throttle history for Mach hold trajectory (at 40,000 ft)

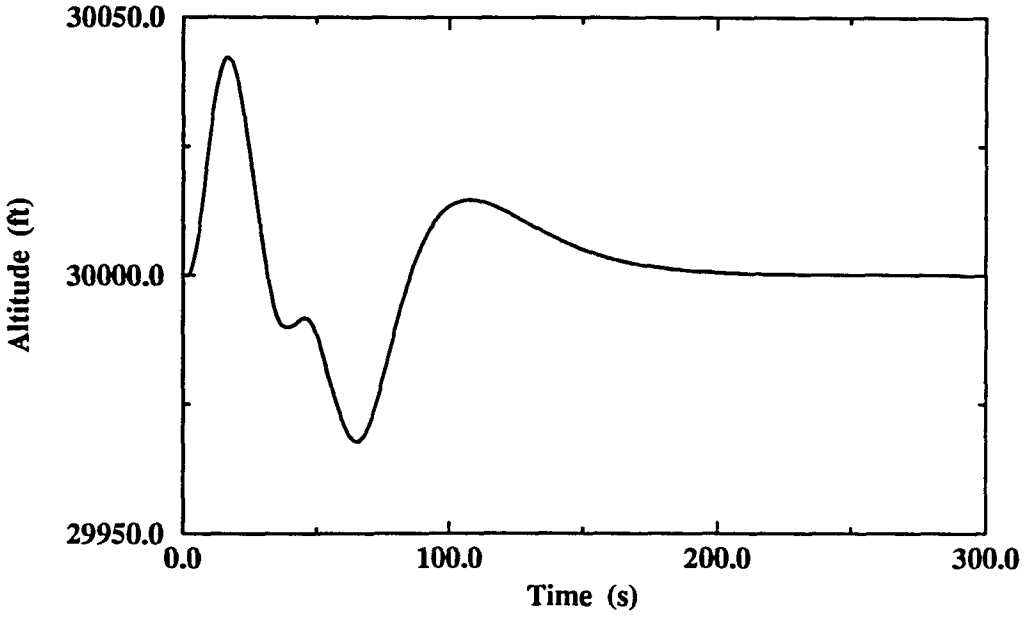

Figure 5.37: Altitude history for level acceleration

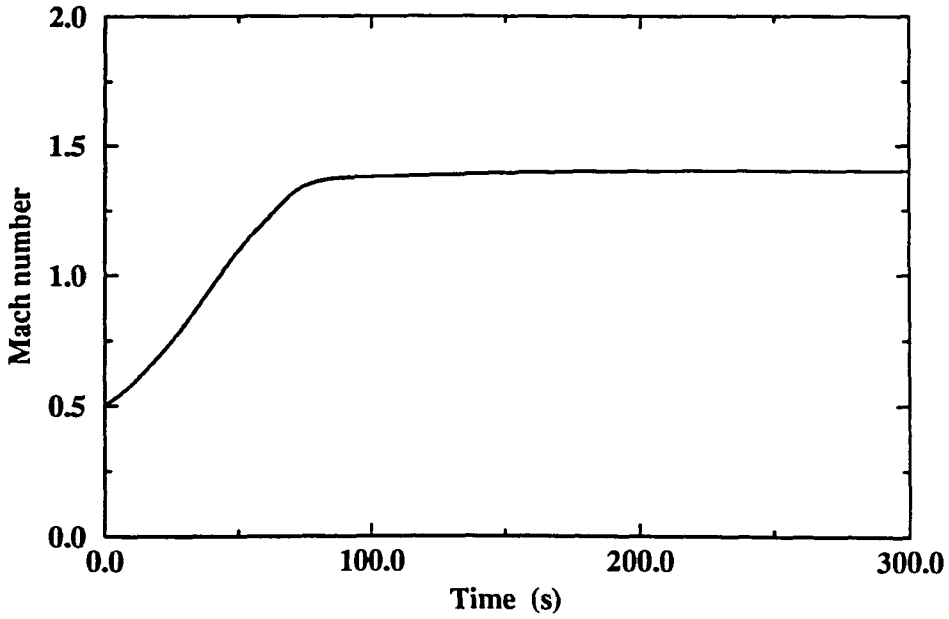

Figure 5.38: Mach history for level acceleration

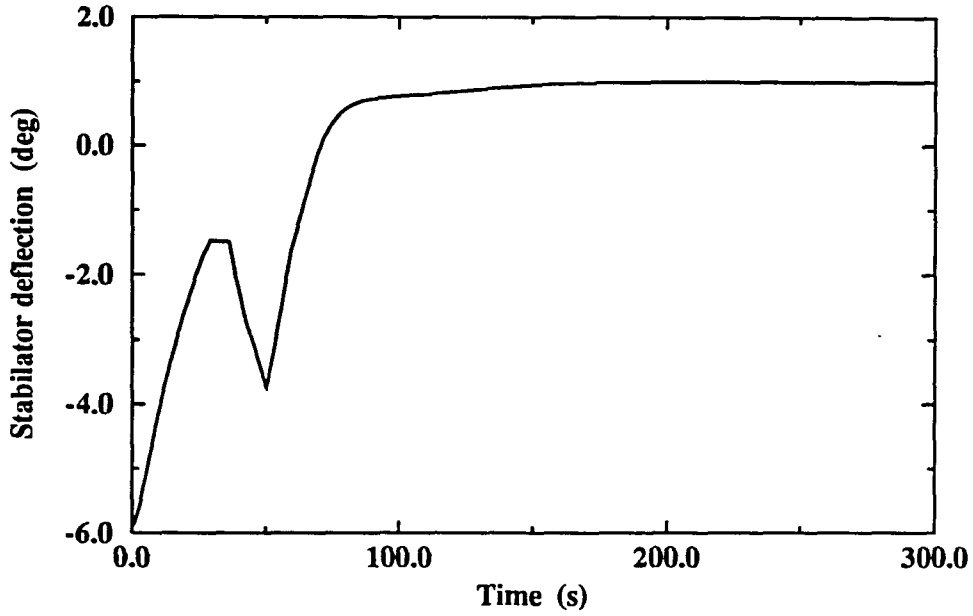

Figure 5.39: Stabilator history for level acceleration

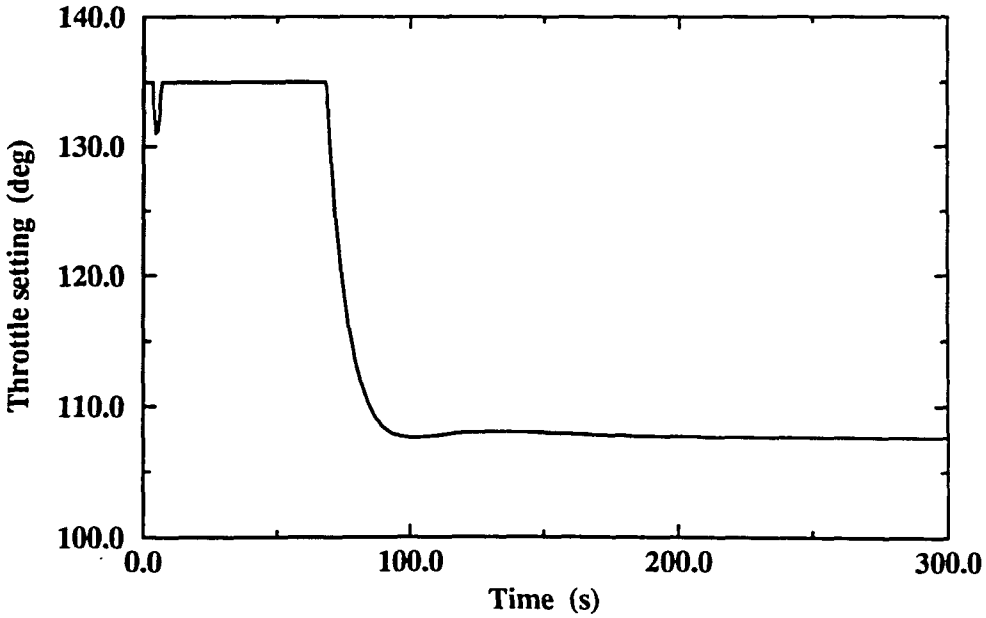

Figure 5.40: Throttle history for level acceleration

CHAPTER 6. SUMMARY

New methods for trajectory optimization and nonlinear guidance law development are presented. Optimal trajectories are obtained using a continuous simulated annealing algorithm called Hide-and-Seek. The algorithm is capable of finding the global maximum of functions with many local optimal solutions. It is implemented on optimal control problems for a complex, high-performance aircraft model. The aircraft model has six-degrees-of-freedom and nonlinear aerodynamics and propulsion. To apply Hide-and-Seek, the continuous optimal control exercises are discretized to finite-dimension parameter optimization problems. The algorithm is successfully used to produce both two and three dimensional minimum time-to-half-loop, minimum time-to-turn, and minimum time-to-climb trajectories. The method proves to be efficient and very robust. It is insensitive to the complexity of the function being optimized or the bounded feasible region. Testing shows that unlike traditional methods, the number of function evaluations required for Hide-and-Seek does not rapidly increase with the dimension of the problem. The simplicity of implementing Hide-and-Seek allows it to be an effective method for solving a wide range of optimization problems.

Various areas of the Hide-and-Seek method and its application can be further studied. The effect of "guessing" an optimal function value versus using the optimal

estimator can be investigated. Indirect optimization problems solving for costates and stochastic optimization problems can also be researched.

The development of a nonlinear feedback controller is also presented. The controller is based on the minimization of predicted tracking errors. It is evaluated on its ability to track a strenuous optimal flight path. The controller exhibits excellent performance for a whole range of initial state perturbations. The basic controller is modified to include a time-varying optimal $\hbar$ parameter. This results in a modest improvement in the controller performance. A second modification minimizing the tracking error rate in addition to the absolute error, results in a significant enhancement over the basic controller. To test the robustness of the control law, it is evaluated in the presence of large external and internal system disturbances. In all cases the controller provides stable high-grade tracking of the reference trajectory. In addition, the controller is used in a unique fashion to design desired trajectories. This ability is used to generate Mach number hold and level acceleration flight paths. These applications demonstrate the vast possibilities for practical use of the controller. Several extensions to the control law research can be made. The effect of actuator dynamics on the performance of the controller can be investigated. In this study, the application is limited to two-dimensional aircraft trajectories. Further investigation can be performed on three-dimensional flight paths.

REFERENCES

[1] Bryson, A.E. and Denham, W. F. "A Steepest-Ascent Method for Solving Optimum Programming Problems," *Journal of Applied Mechanics*, Vol 29, pp. 247-257, 1967.

[2] Ardema, M.D. "Solution of the Minimum Time-To-Climb Problem by Matched Asymptotic Expansions," *AIAA Journal*, Vol 14, pp. 843-850, 1976.

[3] Ong, S.Y. "A Model Comparison of a Supersonic Aircraft Minimum Time-to-Climb Problem," M.S. Thesis, Iowa State University, Ames, 1986.

[4] Hargraves, C., Johnson, F., Paris, S. and Rettie, I. "Numerical Computation of Optimal Atmospheric Trajectories," *Journal of Guidance and Control*, Vol 4, pp. 406-414, 1981.

[5] Stalford, H. and Hoffman, E. "Maximum Principle Solutions for Time-Optimal Half-Loop Maneuvers of a High Alpha Fighter Aircraft," Presented at the *American Control Conference*, Pittsburgh, Pennsylvania, June 1989.

[6] Bryson, A.E. "Applications of Optimal Control Theory in Aerospace Engineering," *Journal of Spacecraft and Rockets*, Vol 4, pp. 545-553, 1967.

[7] Brumbaugh, R.W. "An Aircraft Model for the AIAA Controls Design Challenge," Presented at the *AIAA Guidance, Navigation and Control Conference*, New Orleans, Louisiana, August 1991.

[8] Etkin, B. "Dynamics of Flight," 2nd edition, John Wiley & Sons, Inc., New York, 1982.

[9] Blakelock, J.H. "Automatic Control of Aircraft and Missiles," 2nd edition, John Wiley & Sons, Inc., New York, 1991.

[10] Chapra, S.C. and Canale, R.P. "Numerical Methods for Engineers," 2nd edition, McGraw-Hill, New York, pp. 596-609, 1988.

[11] Otten, R.H.J.M. and van Ginneken, L.P.P.P. "The Annealing Algorithm," Kluwer Academic Publishers, Boston, pp. 1-19, 1989.

[12] Wong, D.F., Leong, H.W. and Liu, C.L. "Simulated Annealing for VLSI Design," Kluwer Academic Publishers, Boston, pp. 1-7, 1988.

[13] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. "Optimization by Simulated Annealing," *Science*, Vol 220, pp. 671-680, 1983.

[14] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. "Numerical Recipes," Cambridge University Press, New York, pp. 283-334, 1989.

[15] Romeijn, H.E. and Smith, R.L. "Sampling through Random Walks," Technical Report 90-02, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1990.

[16] Bélisle, C.J.P., Romeijn, H.E. and Smith, R.L. "Hide-and-Seek : A Simulated Annealing Algorithm for Global Optimization," Technical Report 90-25, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, 1990.

[17] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. "Equations of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, Vol 21, pp. 1087-1092, 1953.

[18] More, J.J., Garbow, B.S. and Hillstrom, K.E. "Testing Unconstrained Optimization Software," *ACM Transactions on Mathematical Software*, Vol 7, pp. 17-41, 1981.

[19] Aluffi-Pentini, F., Parisi, V. and Zirilli, F. "Global Optimization and Stochastic Differential Equations," *The Journal of Optimization Theory and Applications*, Vol 47, pp. 1-16, 1985.

[20] Pouliot, M. R., "CONOPT2: A Rapidly Convergent Constrained Trajectory Optimization Program for TRAJEX," Report No. GDC-SP-82008, General Dynamics, Convair Division, San Diego, California, 1982.

[21] Brent, R.P. "Algorithms for Minimization without Derivatives," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 128-133, 1973.

[22] Parkinson, J.M. and Hutchinson, D. "Numerical methods in Nonlinear Optimization," Academic Press, Boston, pp. 99-114, 1972.

[23] Lu, Ping "Nonlinear Tracking Controllers Design," Submitted to the *American Control Conference*, San Francisco, California, June 1993.

[24] Kahaner, D., Moler, C. and Nash, S. "Numerical Methods and Software," Prentice Hall, Inc., Englewood Cliffs, New Jersey, pp. 361-363, 1989.

[25] Yiyuan, Z. and Bryson, A.E. "Approach Guidance in a Downburst," *Journal of Guidance and Control*, Vol 15, pp. 893-900, 1992.