An Approach to the Use of Neural Network in the Analysis of a Class of Flight Vehicles

by

Agus Budiyono

B.Eng., Bandung Institute of Technology (1992)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1998

Author ....
Department of Aeronautics and Astronautics
December 26, 1997

Certified by ....
Rudrapatna V. Ramnath
Adjunct Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by ....
Jaime Peraire
Associate Professor
Chairman, Departmental Committee on Graduate Students

An Approach to the Use of Neural Network in the Analysis of a Class of Flight Vehicles

by

Agus Budiyono

Submitted to the Department of Aeronautics and Astronautics
on December 26, 1997, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This work presents the possible implementation of an artificial neural network in an Automatic Flight Control System. This new generation of control systems is currently still in the experimental stage and some aspects of the application were examined. The neural networks approach is used to develop a system identification model that imitates the dynamics of a class of flight vehicles. The networks are trained with the simulation data of the vehicle dynamics along a prescribed trajectory. The use of the neural networks in the system identification of the vehicles is considered for use in the design of a neuro-controller. A preliminary effort is made to incorporate the neural network model in this context. The robustness of the neural networks is tested by introducing uncertainties, changes in parameters and time delays in the control system. The overall performance in this illustration is evaluated and compared to that of classical PI control and adaptive control.

Thesis Supervisor: Rudrapatna V. Ramnath
Title: Adjunct Professor of Aeronautics and Astronautics

Acknowledgement

The author would like to express his gratitude to Prof. R.V. Ramnath for his guidance through intensive discussions and lectures which have been inspiring and have shaped continuous motivation throughout the entire period of research.

His thanks also goes to his longtime colleagues, seniors and friends: Prof. Harijono Djojodihardjo, Sc.D. and Said D. Jenie, Sc.D. for their supports without which his work and study at MIT would have been impossible. Financial assistance from Prof. O. Diran and Satrio Sumantri B., Ph.D. from Bandung Institute of Technology which paved his way to MIT has also been always appreciated.

Many have contributed to keep his environment supportive. The unconditional supports from his entire family have been indispensable.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AFCS	Automatic Flight Control System
AI	Artificial Intelligent
ANN	Artificial Neural Networks
BITE	Built In Test Equipment
FDIE	Fault Detection and Isolation Equipment
GHAME	Generic Hypersonic Aerodynamic Model Example
IAC	Indirect Adaptive Control
LEO	Low-Earth-Orbit
LTI	Linear Time Invariant
LTV	Linear Time Varying
NNC	Neural Networks Controller
NNM	Neural Networks Model
MRAC	Model Reference Adaptive control
MSE	Mean Square Error
PI	Proportional Integral
SSTO	Single-Stage-to-Orbit
SSV	Single Stage-Orbiter Vehicle
STOL	Short TakeOff and Landing
TPS	Thermal Protection System
VTOL	Vertical TakeOff and Landing

General Nomenclature

$\dot{x}$	Time derivative of x
x'	Derivative with respect to ξ
$\hat{x}$	Estimate of x
Δx	Discrete difference in x
x_c	Command to x

Neural Network Nomenclature

$f(x)$	Neuron activation function
m	Number of input neurons
n	Number of output neurons
$N+n$	Total number of neurons in the networks
X_i	Input neuron(s)
Y_i	Output neuron(s)
x_i	*Hidden* neurons
$x_{j,i}$	Neuron i, layer j input signal
$y_{j,i}$	Neuron i, layer j activation level
$w_{j,i\times k}$	Weight connecting neurons i and k
	Neuron i in in layer j, neuron k is in layer $j-1$
ϵ	learning rate

Aircraft Dynamics Nomenclature

a	Speed of sound
b	Reference span
c	Reference chord
D	Drag
e	Span efficiency
g	Gravitational acceleration
h	Altitude
i_w	Wing tilt angle
$I_{xx}, I_{yy}, I_{Izz}, I_{xz}$	Moments of inertia
k_x, k_y, k_z	Radii of gyration
L	Lift, in longitudinal equations
	Rolling moment, in lateral-directional equations
M	Mach number
	Pitching moment, in lateral-directional equations

N	Yawing moment
m	Mass of the aircraft
p	Roll rate
q	Pitch rate
r	Yaw rate
R	Radial distance from center of the Earth
S	Reference area
T	Absolute temperature
	Thrust, in longitudinal equations
u, v, w	Perturbation velocity in x, y, z direction
V	Flight speed
W	Weight
y	Generic variable
α	Angle of attack
β	Side-slip angle
δ	Nondimensional mass of atmosphere
δ_N	Control deflection
	Subscript $'N'$:
	e : elevator
	r : rudder
	T : thrust
ϵ	Gradient of atmosphere absolute temperature
γ	Flight path angle
ν	Ratio of moment of inertia
ρ	Air mass density
σ	Inverse nondimensional pitching moment of inertia
ϕ	Euler angle, roll
θ	Euler angle, pitch
ψ	Euler angle, yaw
ξ	Vehicle lengths along trajectory

Chapter 1

Introduction to Neural Networks

The ubiquity of the concept, design and application of intelligent systems in recent years has displayed signs that the approach to the engineering problems has changed. People are attempting to forge biologically-inspired methods into many aspects of engineering problems. Those efforts are focused on the exploration of human-like systems typified by some distinct features not found in traditional logic. Much of engineering today is still based on the knowledge of unchanging objects where decisions are made largely from facts. This is safe and —at times— accurate. But it hinders creativity and thus the horizon of new potential; as well as being inadequate logic system for certain problems, particularly ones which involve uncertainties, time-varying or unknown parameters.

Presently we have the requisite hardware, software and sensor technologies at our disposal for assembling intelligent systems. All the progress owes to the evolution of computing methodologies —as the backbone of such systems— dating back to the 1940s when the first neuron model was devised by McCulloch and Pitts. Neural networks is one of the main constituents of modern computing methodology; the other ones being the fuzzy set theory and genetic algorithm/simulated annealing.

Neural networks innovate the realm of information processing and computation with its distinct feature of learning and adaptation. The method is based on the biological process in the human brain. The brain as a source of natural intelligence has both strengths and weaknesses compared to modern computer. It process incomplete information obtained by perception at an incredibly rapid rate. Nerve cells function about 10^6 times slower that electronic circuit gates, but human brain processes visual and auditory information much faster than modern computers [4]. Neural network methods explore the brain internal mechanism to simulate its powerful functionality.

A neural network is a nonalgorithmic computation technique in which the connectionist architecture of the brain is simulated with a continuous-time nonlinear dynamic system to mimic intelligent behavior. Such connectionism substitutes symbolically structured representations with distributed representations in the form of weights between a massive set of interconnected *neurons* (or *processing units*) and a surprisingly simple processor. It does not need critical decision flows in its 'algorithm'. We do not need to have a detailed process to algorithmically convert an input to an output. Rather, all that one need for most networks is collection of representative examples of the desired translation. The power of an artificial neural network (ANN) approach lies not necessarily in the elegance of the particular solution, but rather in the generality of the networks to find *its own solution* to particular problems, given only instances of the desired behavior.

As a novel computing methodology, neural networks approach is aimed at solving real-world decision making, modeling and control problems. One area of particular significance is system identification and control system design for aerospace vehicles whereby some or all of the pilot's ability is replaced or augmented. During its evolution in the past neural networks have not been used as an acceptable possibility since engineers prefer to use well-defined systems due to the stringent requirement of safety. That is one of the reasons why linear control has occupied large areas in the engineering systems. A certain class of (flight) dynamics problems represents a complex behavior characterized by:

□ Nonlinear dynamics

□ Unknown parameters or uncertainties

□ Time-varying parameters in vehicle plant and environment

□ Time lags

□ Noise inputs from sensors and from air disturbances

Conventional technology could not in general cope with most of these problems. The use of neural networks provides a potential method by which the problems can be approached, analyzed and overcome.

1.1 Basic Perspective of Neural Networks

1.1.1 Definition and Remarks

What are neural networks? There have been numerous definitions that is representative in describing the ANN. Some conceptual as well as technical notions from researchers are listed below.

> A neural network is a parallel, distributed information processing structure consisting of processing elements (which can posses a local memory

> and carry out localized information processing operations) interconnected together with unidirectional signal channels called *connections.* Each processing element has single output connection which branches ('fans out') into as many collateral connections as desired (each carrying the same signal — the processing element output signal). The processing element output signal can be of any mathematical type desired. All of the processing that goes on with each processing element must be completely local *i.e.*, it must depend only upon the current values of the input signal arriving at the processing element via impinging connections and upon values stored in the processing element's local memory [13]
>
> Artificial neural networks are methods of computing that are designed to exploit the organizational principles that are felt to exist in biological neural systems [12].
>
> Neural networks are massively parallel systems that rely on dense arrangements of interconnections and surprisingly simple processors [10].
>
> Neural networks are self-organizing information systems in which information organizes itself [9].

Those attributes illustrate the novelty of neural networks in contrast to traditional computing and information systems. It is the self-organizing characteristic resulting solely from its internal dynamics that gives neural networks great capabilities.

1.1.2 Milestones on Neural Networks Research

Artificial neural networks method is still sometimes referred as new method though it has evolved for more than 50 years along with other 'soft' computing constituents (see Table 1.1). The list below gives the brief historical notes on neural networks research.

- □ 1943. W. McCulloch and W.Pitts designed the first neuron model. Their paper – "A Logical calculus of the ideas immanent in nervous activity"– presents the mathematical formalization of a neuron, in which a weighted sum of input signals is compared to a threshold to determine wheter or not the neuron fires. It shows that simple neural networks can compute any arithmetic or logical function.

- □ 1948. Weiner's seminal book *Cybernetics* posed that cybernetics as the study of information and control in humans and machines was the root of computing

Table 1.1: A historical sketch of ANN research as part of soft computing evolution [4]

Year	Conventional AI	ANN	Fuzzy systems	Other methods
1940s	1947. Cybernetics	1943 McCulloch -Pitts neuron model		
1950s	1956. Artificial Intelligence	1957. Perceptron		
1960s	1960. Lisp Language	1960s Adaline/ Madaline	1965. Fuzzy sets	
1970	mid-1970s. Knowledge Engineering (Expert systems)	1974. Birth. of backpropagation algorithm. 1975. Cognitron and Neocognitron	1974. Fuzzy controller	1970s. Genetic Algorithm
1980		1980. Self-organizing map. 1982. Hopfield Net. 1983 Boltzmann machine. 1986 Backpropagation algorithm boom	1985 Fuzzy modeling	mid-1980s Artificial life Immune modeling
1990			1990. Neuro-fuzzy modeling. 1991. ANFIS. 1994. CANFIS.	Genetic programming

sciences, control and neurobiology. As we know today, those constituents have tended to go their own separate ways and at times creates a barrier to effective interchange of ideas between the disciplines.

□ 1949. D. O. Hebb introduced a important learning method in his book *The Organization of Behavior* which is used in some form in most learning rules used today. The major proposal of this book is that behavior can be explained by the neuron activities.

□ 1951. Minsky and Edmonds devised the first hardware realization of a neural network.

□ 1958. F. Rosenblatt presented the first practical ANN called perceptron, a basic unit for many other ANN. A form of Hebbian learning rules were used in the learning techniques for perceptron.

□ 1960. B. Widrow and M.E. Hoff introduced an adaptive perceptron-like network that can learn quickly and accurately. Known today as Adaline, the system has inputs and a desired output classification for each input. The well-known gradient descent method is used to minimize the least mean square error as for weights adjustment rules.

□ 1969. M. Minsky and Seymour Papert caused widespread pessimism —causing inhibition of research funding and activities for several years— among the ANN researches with their book *Perceptrons.* The book rigorously describes the limitation of perceptron learning capability and is closed with the remark that field of ANN is a dead end. Though this has been proved to be not entirely true, the premise has showed the kind of problems that can or can not be solved by perceptron.

□ 1972. Teuko Kohonen and J. A. Anderson worked independently to come to closely related results. Kohonen proposed a correlation matrix model for associative memory, while Anderson did for a 'linear associator' for the same purpose. A generalized Hebbian learning rules were used to correlate input and output vectors. While Kohonen's work gave emphasize on mathematical structure, Anderson's remarks put weight on physiological plausibility of the network.

□ 1974. P.J. Werbos brought new ideas in error minimization algorithm of ANN learning rules. His Ph.D. thesis "Beyond regression: new tools for prediction and analysis in the behavior sciences" was published and gave birth to a now famous learning rule — backpropagation.

Table 1.2: Neural networks as a constituent of soft computing [4]

Methodology	Strength
Neural network	Learning and adaptation
Fuzzy set theory	Knowledge representation via fuzzy if-then rules
Genetic algorithm and simulated annealing	Systematic random search
Conventional artificial intelligence	Symbolic manipulation

- □ 1976. S. Grossberg proposed a self-organizing neural network. The idea was based on the visual system. The network is characterized by internal continuous-time competition. This gives the basis for concepts of the adaptive resonance theory (ART) networks.

- □ 1982. J.J. Hopfield, in his paper "Neural networks and physical systems with emergent collective computation abilities", described a content-addressable neural network. This recurrent networks (today called Hopfield Net) was considered highly influential in bringing about the renaissance of neural network research that year.

At present research on neural networks is still growing. Some of it is still directed to the learning method refinement (that once caused backpropagation algorithm boom in the late 1980's). On the other side, people have tended to resort to the idea of cybernetics where computation, information systems, control and neurobiology meet. This tendency has created a new discipline called 'soft computing' consisting a mix of the separated fields. In this discipline neural network method has blended with the other methodologies: fuzzy logic, probabilistic reasoning and genetic algorithm. Each of the elements has its own distinction. So they are complementary rather than competitive with each other. The significant contribution of neural networks is methodology for system identification, learning and adaptation; that of fuzzy logic is computing with words; that of probabilistic reasoning is propagation of belief; and that of genetic algorithm is systematized random search and optimization. Table 1.2 enumerates the strength of each soft computing elements.

Along with the unfolding of the neural networks research, industry has responded with various applications in different fields ranging from consumer electronics and industrial process control to decision support systems and financial trading. With many real-world problems that are imprecisely defined, contain uncertainties and require human intervention, neural networks method is likely to play an increasingly important roles today and in the future.

1.2 The Characteristics of Neural Networks

There are many reasons why neural networks are a likely candidate for an attractive new design of system identification and control of aerospace vehicles. They stem from ANN's intrinsic properties; they are summarized below from various references.

□ **Potential of intelligent and natural control**. Inspired by biological neural networks, ANN are employed to deal with pattern recognition, and nonlinear regression and classification problems. It can also imitate human behavior if desired.

□ **Real-time parallel processing**. Neural networks have a highly parallel structure which lends itself immediately to parallel implementation. Such an implementation can be expected to achieve a higher degree of fault tolerance than conventional schemes. The basic processing element in a neural network has a very simple structure. This, in conjunction with parallel implementation, results in very fast overall processing.

□ **Nonlinear systems**. Neural networks have the greatest promise in the realm of nonlinear control problems. This emanates from their theoretical ability to approximate arbitrary nonlinear mappings. These networks are likely to also achieve more parsimonious modelingthan alternative approximation schemes.

□ **Resistance to noisy data**. This capability comes from its ability to learn a general pattern not necessarily an exact figures.

□ **Multivariable System**. Neural networks naturally process many inputs and have many outputs; they are readily applicable to multivariable systems.

□ **Computational efficiency**. Without assuming to much background knowledge of the problem being solved, neural networks rely heavily on high-speed number-crunching computation to find rules or regularity in data sets. This is a common feature of all areas of computational intelligence.

□ **Fault Tolerance**. The structure of neural networks depicts its inherent fault tolerance characteristic. ANN architectures encode information in a distributed fashion. Typically the information that is stored in neural networks is shared by many of its processing elements. This kind of structure provides redundant information representation. The deletion of some neurons does not necessarily destroy the system. Instead, the system continues to work with gracefully degraded performance; thus the result is a naturally fault —or error— tolerant system.

□ **Quickly integrated into systems**. Control systems using neural networks, for example, has fixed software/hardware architecture. Changing the control algorithm amounts to changing the weights of the neural connections and not the structure of the controller.

□ **No requirement of explicit programming**. Neural networks are not programmed; they learn by example. Typically, a neural network is presented with a training set consisting of a group of examples from which the network can learn. The training set, in the form of input-output value pairs is known as the *patterns*. The absence of human development of algorithms and programs suggests that time and human effort can be saved.

□ **The ability of universal approximation**. The nonlinear activation function gives the neural network the capability of forming any nonlinear functional representation. It should be noted that there are other universal approximation methods. For example Taylor series, which use polynomial approximation and Fourier series, which use the sine and cosine trigonometric functions to approximate any continuous function. This ability is particularly useful in the case when internal dynamics of a system under study are not clear, unavailable, or nonlinear.

□ **Associative memory**. The information in neural networks is stored in associative memory in contrast to digital computer's addressed memory where particular pieces of information are stored in particular locations of memory. This, besides giving a fault-tolerant property, is used to generalize between training examples.

Note that all of these characteristics of neural networks can be clarified through the simple mathematical structure of a neural networks model. Though the broad behavioral terms such as *learn*, *generalize* and *adapt* are used, the neural network behavior is simple and quantifiable at each node. In short, their internal mechanism is clear and tractable, but produces complex macroscopic behavior.

1.3 Artificial Neural Networks Applications in Aerospace Engineering

There have been numerous research on ANN applications in aerospace field. There are at least two major reasons why bringing new technology is attractive. Firstly, performance gain over conventional approach is desirable; and secondly novel approaches can engender wider horizon of possibilities. Several key considerations driving ANN application to aerospace have been elaborated in the previous section dealing with ANN characteristics.

Universities, aircraft companies and research centers have been involved in the research of ANN for use in aerospace thus far. Some of them are summarized as follows:

- □ **Flight Control**. Jorgensen and Schley (1990) propose the use of ANN in autolanding systems, to increase the operational safety envelope and provide methods of exploring alternative control systems capable of dealing with turbulent, possibly nonlinear control conditions. The method used was to set up a neural network to copy pilot's actions and hence attempt to obtain 'flight sense'.

- □ **FDIE and BITE**. Notable work in the area of Fault Detection and Isolation Equipment (FDIE) as well as Built In Test Equipment (BITE) was published by Barron et al. (1990), where ANN were used successfully in the development of an FDIE and BITE system for an aircraft with reconfigurable flight controls.

- □ **Pilot Aid**. Numerous researchers have been used ANN in developing pilot aids, for example in pilot decision support systems (Seidman 1990); and flight management systems (Burgin & Schnetzler 1990).

- □ **Image Recognition**. ANN have been used for image processing as well as recognition, with application to identification of aircraft from visual or radar signals. Research in this area is abundant, and is described as a typical application of ANN (Freeman & Skapura 1991).

- □ **Manufacturing**. ANN are being investigated for use in process optimization and control by several major aerospace company.

In the realm of control of aircraft or spacecraft, ANN will function either as the following roles [33]:

Controller. An ANN controller would perform mappings of state variables (as input vectors) into control variables (output vectors)

Critic. The critic networks can be trained to monitor to aircraft's dynamic state, mission profile and hence evaluate system performance. Information from this critic can then be used to adapt a conventional adaptive controller or a Neural Networks Controller (NNC). Pattern classification networks would then be suitable for this function.

Gain programmer. An ANN can be used as a gain scheduler/programmer, to link with a conventional controller, under adaptive learning or static programming. This technique has many advantages since by limiting the output gain range, the Automatic Flight Control System (AFCS) can have 'proven' stability using conventional linear stability analysis techniques.

Monitoring, fault detection. A suitable application for ANN in AFCS is in monitoring systems and Fault Detection and Isolation Equipment (FDIE). This is because ANN generally have a good fuzzy knowledge ability, can operate well with noisy data, and have good generality.

Plant mimic. Of perhaps more use in the design of conventional and modern controllers, an ANN can be used to model plant (e.g. aircraft/spacecraft, environment, pilot) in order to optimize performance, find system inverse, dynamic derivatives, etc.

Not only could an ANN function as a mimic of the plant, which is really in the system identification role, but a network could be set up to model a reference plant with desired properties for a model reference adaptive controller (MRAC). This is however not recommended since for MRAC type systems it is customary to choose a reference model with well known properties: such as a linear model

1.4 Neural Networks Architecture

The network architecture is defined by the basic *processing elements* and the way in which they are interconnected. The basic processing element of the ANN architecture is often called a *neuron*, but other names such as *unit, node, perceptron* and *adaline* are also used. The neurons are connected by *weights*, also referred to as *connections* or *synapses*, which convey the information. Neurons are also often collected into groups called *layers* or *slabs* within which the neurons have a similar function or structure. Depending on their function in the net, one can distinguish three types of layers. The layers whose activations are input receptive for the net are called *input* layers, similarly layers whose activations represent the output of the net, *output* layers. The remaining layers are called *hidden* layers, because they are 'invisible' from outside.

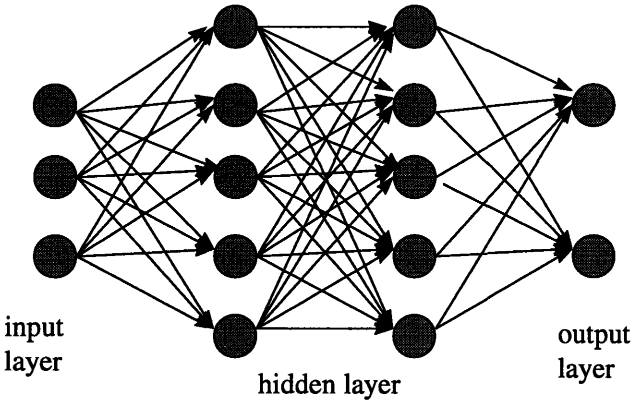


Figure 1-1: A Neural network structure with 2 hidden layers

In general, there can be any number of layers, and neurons in any layer can be connected to neurons in any other layer. Some neurons have no input from other neurons, and are known as *bias*, or *threshold*, unit. The memory content or signal strength of neuron is referred to as the *activation level.* Usually, the activation levels of the input and output units are scaled such that the activation levels are appropriate to neuron function and the observable values are given a physical meaning.

1.4.1 Neuron Model and Network Structure

Let us consider a single neuron from the networks structure which is depicted in Fig. 1-1. An individual neuron has many inputs, dependent on the number of incoming connections. Each connection to the neuron has a weight associated with it. After the total neuron input signal is calculated, that is given by the sum of the product of inputs and weights:

$$x_{j,i} = \sum_{k=1}^{n} w_{j,i \times k} y_{j-1,k} \tag{1.1}$$

this signal is converted into an activation value through a functional relationship. The nonlinear processing power lies within this transfer function.

$$y_{j,i} = f(x_{j,i}) \tag{1.2}$$

There is a wide class of arbitrary functions which can be used for some ANN. The commonly used functions are listed in Table 1.3

The output of the neuron is obtained from the activation value also through a functional relationship. The identity function is usually used in this step. In general, the selection of the function that is used in neural networks depends on the type of patterns (input-output value pairs). At present, the selection of activation function is more art than science and the subject of much research. The important point to remember is that any nonlinear function will provide network with the capability of representing any nonlinear functional mappings.

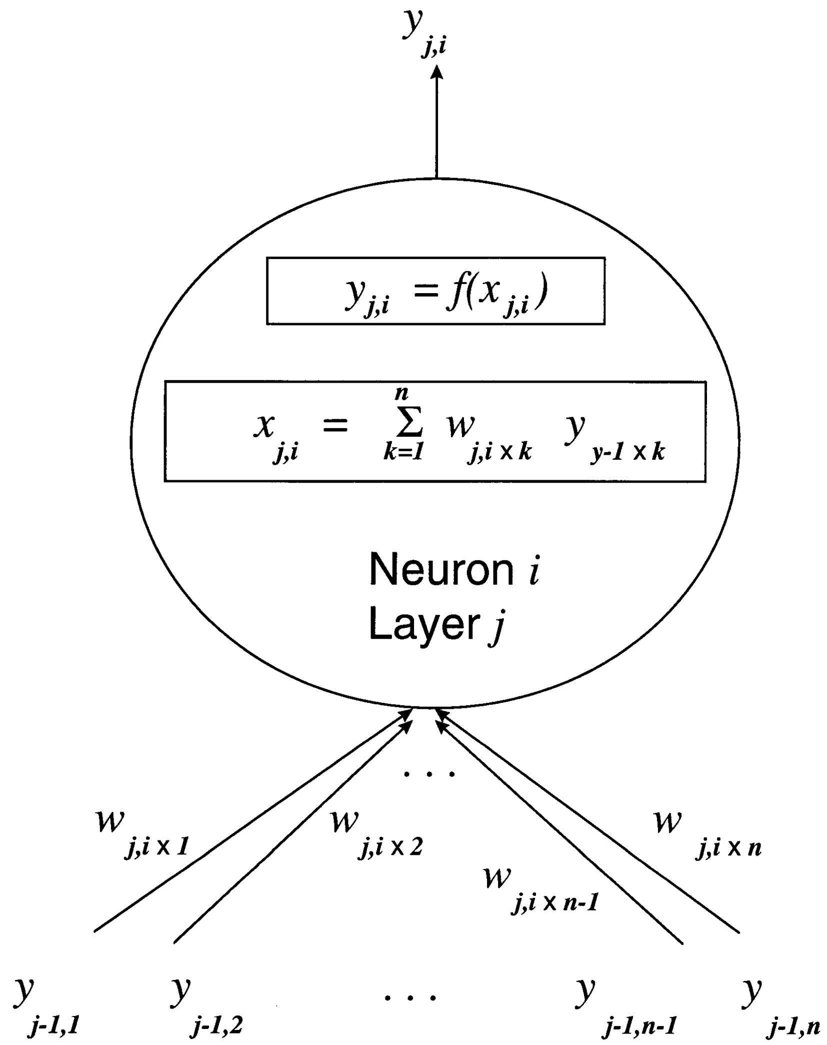


Figure 1-2: A Single Neuron Mathematical Model

Table 1.3: Activation function for neuron

Function name	Expression	Annotation
Sigmoidal Function	$f(x) = \frac{1}{1+e^{-\alpha x}}$	This differentiable, step-like, positive (bounded by $(0,1)$) function is the most commonly used activation function for neural networks
Hyperbolic Tangent	$f(x) = tanh(x) = \frac{e^{x}-e^{-\alpha x}}{e^{x}+e^{-\alpha x}}$	Its characteristics resemble those of the sigmoid except that it is zero-mean function bounded by$(-1,1)$
Threshold	$f(x) = H(x) = \begin{cases} +1, & x \geq 0 \\ 0, & x < 0 \end{cases}$	It is also called Heaviside step function and is non-differentiable, step-like and positive.
Signum Function	$f(x) = sgn(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$	It is nondifferentiable, step-like and zero-mean.
Ramp Function/ Linear Scaling Function	$f(x) = \alpha x$	This unbounded function can be useful in some particular conditions. Note that it is a linear function.

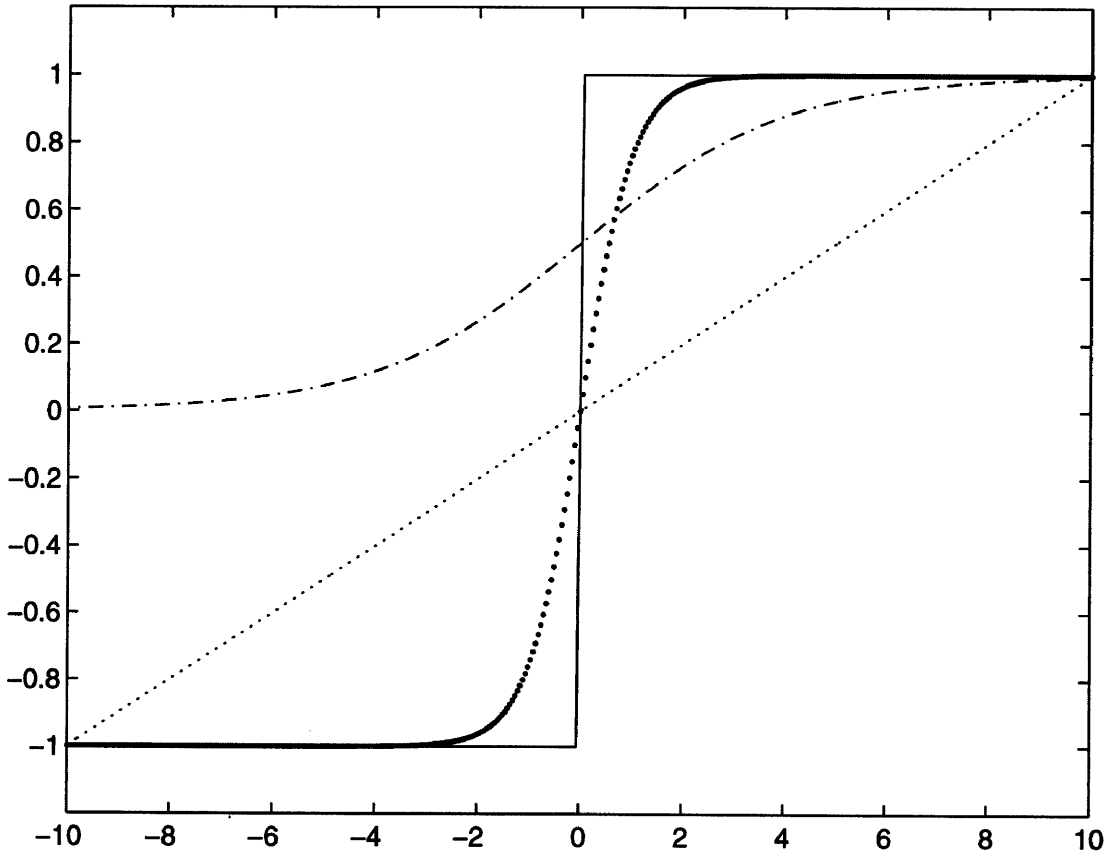


Figure 1-3: Activation functions in ANN neurons

.... ≡ Ramp function /linear function
— · — · —· ≡ Sigmoidal function
· · · · · · ≡ Hyperbolic Tanh
———— ≡ Threshold /Sign function

1.4.2 Types of Artificial Neural Networks

There are in general three different architectures of ANN. Each of them exhibits a particular way of processing the information throughout the networks. Though all these networks can perform various tasks, each of them is generally well-suited only for a certain class of problem. Ref. [14] provides a rich array of networks that has been developed during recent years. They are listed under the corresponding rubric below.

Feedforward Networks

In these networks the output is computed directly from the input in one pass; no feedback is involved. Included in this category are perceptron (multi-layer perceptron), linear associator and adaline. Feedforward networks are used for pattern recognition and for function approximation. Examples of application areas for function approximation are adaptive filtering and automatic control. Some feedforward (and their variant) networks are listed as the following:

- Radial Basis Networks (RBN). RBN may require more neurons than standard feed-forward backpropagation networks, but often they can be designed in a fraction of the time it takes to train standard feed-forward networks. The activation function used is called radial basis function. It has a maximum of 1 when its input is 0; and approaches 0 when its input approaches ∞ or $-\infty$. The RBN work best when many training vector are available.

- Adaptive Critic. This networks are generally used in the reinforcement learning controller (similar to direct control concepts of traditional adaptive control). In this scheme a neural network is used as a judgement network capable of differentiating between good performance and bad performance.

Competitive Networks

These types of networks are characterized by two properties. First, they compute some measure of distance between stored prototype patterns and the input pattern. Second, they perform a competition to determine which neuron represents the prototype pattern closest to the input. In the competitive networks the prototype patterns are adjusted as new inputs that are applied to the network. These adaptive networks learn to cluster the inputs into different categories.

Recurrent (Dynamic Associative Memory) Networks

As opposed to feedforward network, this type of network has feedback that associates stored data with input data. In other words, they are used as associative memories in which data is recalled in association with input rather than by an address. Hopfield network is an example of recurrent networks.

1.5 Learning Algorithm

Learning is one of the basic features of intelligence. The concept of learning machines comes from biological models. It is an effective self-modification of the organism that lives in a complex and changing environment. Learning is a directed change in knowledge structure that improves the performance [5]. This section describes different learning rules in the *training* of neural networks. The word *training* here means the process of minimization error between the networks output and the desired one. The discussion will be restricted to the supervised feedforward model which is so far the most tractable and most applied network model [6].

Learning rules govern the modification of connectivity as a function of experience in neural network. Various learning rules have been developed and today the research in that field is still active. Basically, most of the learning rules is inspired by Hebb's law. Thus, we begin with the overview of the Hebb's law. Note that the list is not intended as rigorous or comprehensive. Rather, it is given as a synopsis for introductory purposes. For a more complete list, the reader can see Ref. [14]

1.5.1 Hebb's Law [2]

Hebb's law is the fundamental psychophysical law of associative learning. It serves as both the basic law of psychology and artificial neural systems. The essence of Hebb's law is formulated as follows:

> "If neuron A repeatedly contributes to the firing of neuron B, then A's efficiency in firing B increases"

The combination of Hebb's law and Grossberg's neural modeling theory explain the learning process in *Pavlovian* experiments:

> When a dog is presented with food it salivates. When the dog hears a bell it does not salivate initially. But after hearing the bell simultaneously with presentation of food on several consecutive occasions, the dog is subsequently found to be salivate when it hears the bell alone.

In general terms, when a conditioned stimulus (such as bell) is repeatedly paired with an unconditioned stimulus (such as food) which evokes an unconditioned response (such as salivation), the conditioned stimulus gradually acquires the ability to evoke the unconditioned response.

Hestenes in ref. [5] concludes that the association strength between stimulus and response that psychologist infer from their experiments is a crude measure of the synaptic coupling strength between neurons in the central nervous system. The same can be said about all association among ideas and actions. Thus, the full import of Hebb's law is this:

> "All associative (long-term) memory resides in synaptic connections of the central nervous system, and learning consists of changes in synaptic coupling strengths"

The strengthening of specific synapses within neural circuits is the most accepted theory for learning and memory in the brain.

1.5.2 Generalized Delta Rule

The learning process (training) begins when error between ANN and the desired output is obtained. The main problem of learning in the network with three or more layers is how to modify an inner or hidden layer. The first answer is unsupervised competitive learning, which generates useful hidden-unit connection. The second answer is to assume a hidden-unit connection matrix, on *a priori* grounds. The third possible way is the modification of the hidden units through the *backpropagation* error. The determination of error starts with the output units, and then propagates to the next hidden layer, until it reaches the input units. This kind of learning is called generalized Delta Rule.

1.5.3 Sigma-Pi Units

This learning algorithm is applied for more general form of multilayer perceptron called Sigma-Pi network. The net input is given by $\sum w_{ij} \Pi a_{i1} a_{i2} \cdots a_{ik}$ where i indexes the conjuncts impinging on unit j and $a_{i1}, a_{i2}, \cdots, a_{ik}$ are the k units in the conjunct. Thus, the net input to each neuron is equal to weighted sum of all signals impinging on that neuron, as well as weighted sum of selected products of these signals. The more elaborate usage of Sigma-Pi units includes:

- Gates; weighted connections

- Dynamically programmable networks in which the activation value of some units determine the function of other units

– Mimicking different monotonic activation and interconnection functions

1.5.4 The Boltzman Machine Learning Rules

A Hopfield network will converge to a local minimum of the Lyapunov function, but there is no guarantee that it will converge to a global minimum [14]. In the Boltzman machine, noise is used in an attempt to reach the global minimum. The technique is called *simulated annealing* and is analogous to metallurgical annealing, in which a body of a metal is heated to near melting and then slowly cooled according to a specified schedule. The high temperatures cause thermal agitation, which prevents the metal from becoming frozen in a high energy (brittle) state. The global "energy" of the system is defined as:

$$E = -\sum_{i<j} w_{ij} s_i s_j + \sum \theta_i s_i \tag{1.3}$$

$$\Delta E_k = \sum_i w_{ki} s_i - \theta_k \tag{1.4}$$

where
s_i is the state of i-th unit (-1 or $+1$)
θ_i is the threshold
ΔE_k is the difference between the energy of the whole system with the k-th hypothesis false and its energy with the k-th hypothesis true.

While the binary threshold in perceptron is deterministic, in Boltzmann machine it is probabilistic:

$$P_i = P\left(\Delta E_i\right) = \frac{1}{1 + e^{\Delta E_i / T}} \tag{1.5}$$

where
P_i is the probability for the i-th unit to be in state
$P(x)$ is a sigmoidal probability function
T is a parameter analogous to temperature and is a measure of the noise introduced into the decision.

The total output to the unit is

$$\Delta E_i = \sum w_{ij} s_j \tag{1.6}$$

The Boltzman learning algorithm is in this case closely related to the maximum likelihood methods. The learning is supervised i.e. the input units are clamped to a particular pattern, while the network relaxes into a state of low energy in which the output units have the correct values. Due to the symmetry ($w_{ij} = w_{ji}$), the energy gradient with respect to w_{ij} depends only on the behavior of i-th and j-th units and not on the whole network. This fact helps in updating input, output and hidden units.

1.5.5 Backpropagation

The backpropagation technique is first introduced in the Ph.D. work of Werbos in 1974. Since then numerous variants of backpropagation have been developed. The name backpropagation is actually the popular name for the generalized Delta Rule mentioned previously. It is probably the most commonly used learning algorithm for the training of neural networks. The principle of backpropagation is summarized from [8] as follows.

In basic backpropagation, how the output of a neural network depend on its inputs and weights is defined using the following logic:

$$x_i = X_i, \quad 1 \le i \le m \tag{1.7}$$

$$net_i = \sum_{j=1}^{i-1} W_{ij} x_i, \quad n < m < i \le N + n \tag{1.8}$$

$$x_i = s\left(net_i\right), \quad n < i \le N + n \tag{1.9}$$

$$Y_i = x_{i+N}, \quad 1 \le i \le n \tag{1.10}$$

where

X_i	is the input neuron(s)
Y_i	is the output neuron(s)
x_i	is the *hidden* neurons
m	is the number of input neurons
n	is the number of output neurons
$N + n$	is the total number of neurons in the networks
s	is commonly *sigmoidal* function.

In that structure, the networks is *fully connected* in the extreme. Most researchers prefer to use layered networks, in which all connection weights w_{ij} are zeroed out, except for those going from one layer to the next layer.

The weights in this learning algorithm are adjusted according to the following procedures. First, we adapt weights w_{ij}, so as to minimize square error over training set:

$$E = \sum_{t=1}^{T} E\left(t\right) = \sum_{t=1}^{T} \sum_{i=1}^{n} \left(\hat{Y}_i\left(t\right) - Y_i\left(t\right)\right)^2 \tag{1.11}$$

This is simply a special case of the well-known method of least-squares. The uniqueness of backpropagation lies in how this expression is minimized. In standard backpropagation, we start with output values Y_i for the weights w. Next, we calculate the output $\hat{Y}\left(t\right)$ and the errors $E\left(t\right)$ for the set of the weights. Then we calculate the derivatives of E with respect to all of the weights. If increasing a weight leads to a less error, we adjust it upwards. After adjusting all the weights up and down, we start all over and keeping or going through this process until the weights and the errors settle down. The uniqueness of backpropagation lies in the method used to calculate the derivatives exactly for all of the weights in only a single pass through the system. Secondly, the backward pass for the ANN starts with computing partial derivative of error with respect to the output:

$$\frac{\partial E}{\partial \hat{Y}_i\left(t\right)} = \hat{Y}_i\left(t\right) - Y_i\left(t\right), \quad 1 \leq i \leq n \tag{1.12}$$

Using the definition of *ordered derivatives* [8], we can write:

$$\frac{\partial^{+} E}{\partial X_i\left(t\right)} = \frac{\partial E}{\partial X_i\left(t\right)} + \sum_{j=i+1}^{N+n} \frac{\partial^{+} E}{\partial X_j\left(t\right)} \frac{\partial X_j\left(t\right)}{\partial X_i\left(t\right)}, \quad N + n \geq i \geq 1 \tag{1.13}$$

where

$$\frac{\partial X_j\left(t\right)}{\partial X_i\left(t\right)} = \begin{cases} 0 & 1 \leq i \leq m \\ \frac{\partial E}{\partial \hat{Y}_{i-N}\left(t\right)} & N + n \geq i \geq m + 1 \end{cases} \tag{1.14}$$

and

$$\frac{\partial X_j\left(t\right)}{\partial X_i\left(t\right)} = s'\left(net_j\right) w_{ji} \text{ with } s'\left(\right) = \frac{\partial s\left(\right)}{\partial\left(\right)} \tag{1.15}$$

Now the ordered derivatives for the weights can be calculated as:

$$\frac{\partial^{+} E}{\partial w_{ij}\left(t\right)} = \frac{\partial^{+} E}{\partial X_i\left(t\right)} s'\left(net_i\right) X_j\left(t\right) \tag{1.16}$$

Finally, the weights are updated using:

$$\text{new}w_{ij} = w_{ij} - \epsilon \frac{\partial^+ E}{\partial w_{ij}(t)}, \quad 1 \leq i \leq N+n; 1 \leq j \leq N+n \tag{1.17}$$

where ϵ is defined as the *learning rate* which is some small constant chosen on an *ad hoc* basis.

The main problem with basic backpropagation is a slow convergence. Since backpropagation was first popularized, there has been considerable work on methods to accelerate the convergence of the algorithm. Together with this efforts was the development of many variants of backpropagation. With respect to convergence improvements, there are two approaches: heuristic methods and standard numerical optimization methods [14].

- **Heuristic Methods**. Under this category are backpropagation with momentum (MOBP) and variable learning rate backpropagation (VLBP). These two methods significantly accelerate the convergence. MOBP is simple to implement, can be used in batch mode or incremental mode. It does require the selection of the momentum coefficient γ, but γ is limited to the range $[0, 1]$ and the algorithm is not extremely sensitive to this choice [14]. The VLBP algorithm is faster than MOBP but must be used in batch mode. For this reason it requires more storage.

- **Standard Numerical Procedures**. Two well-known example of this technique are conjugate gradient backpropagation (CGBP) and Levenberg - Marquardt backpropagation (LMBP). CGBP is generally faster than VLBP. It is a batch mode algorithm, which requires a linear search at each iteration, but its storage requirements are not significantly different from that of VLBP. The LMBP algorithm is one of the fastest algorithm for training multilayer networks of moderate size, even though it requires a matrix inversion at each iteration. The main drawback of LMBP is the storage requirement.

- **Other variants**. The area of backpropagation variants has probably been the most active area of neural network research since 1986. A few of the more successful backpropagation approaches are: quickprop, Rprop, Cascade-Correlation, Network-Pruning, Regularization and Stopped Training.

1.5.6 Discussion of Some Related Issues

Active versus passive learning from samples

Active learning from an environment without necessarily assuming the availability of training samples is much more demanding and difficult than passive learning from samples. Conventional supervised and unsupervised learning are examples of passive learning, whereas distal supervised learning is an example of active learning. In this case the learner must be active in getting training samples from the environment. It is worth pointing out that supervised learning is closely related to parameter estimation in system identification and adaptive filtering [11]

On-line learning versus off-line learning

Off-line learning is carried out by presenting the network to a specified data set (training set), and through learning the network's structure is reorganized. After the training goal achieved the network can then be used on-line, but the network is prevented from learning any further. Thus ANN performance will have a static performance over time provided the plant does not change. On-line learning, in contrast, is carried out by presenting the network to its intended operation. Thus it learns continuously during operation. On-line or real-time learning in control is not only normally desirable but also sometimes indispensable. For this to be possible, learning algorithm must have simplicity and efficiency so that it can operate in the real-time. In application, combining off-line learning and on-line learning is possibly very useful. The systems first will be trained off-line (at least at a short period of time) to ensure it has an adequate starting performance before it must go on-line.

Dynamic variable-structure learning versus static fixed-structure learning

Conventional neural networks training is fixed-structure learning. Structure parameters such as the number of neurons and layers remain unchanged during training. The learning implies a determination of connection weights only, with an implicit assumption that the present structure is capable of representing in some sense the desired functional relationship. In contrast, dynamic variable-structure learning is capable of self-determining some structure parameters such as the number of neurons in response to incoming data, thereby making the net structure dynamically variable. This feature is sometimes referred to as self-organizing for obvious reasons [11].

Global learning versus specific learning

Depending on the usage, ANN can be made purposely to be globally or specifically learning by supplying global or specific knowledge to the learner. If ANN is used for a forward model, global learning is possible and normally necessary. In contrast, if it is employed as a direct controller it is unlikely that the controller will be able to be global, capable of dealing with a variety of different situations which may be unforeseen and time-varying [11]

Spatial learning versus temporal learning

An ANN controller may be constructed via spatial learning or temporal learning. A spatial learning scheme typically involves the extensive use of past experience to improve the present and future performance. The past experience can be cumulated during several past trials along a spatial direction in a time period of interest. Memorizing past control knowledge and repeatedly interacting with the controller environment are typical characteristics of a spatial learning. Compared with spatial learning, temporal learning emphasizes learning accruing along a time direction only by using the information gained at immediate past time-instants and reacting to temporal variations [11].

Chapter 2

ANN for the GHAME Vehicle Dynamics Simulation

2.1 Introduction

One of the most challenging tasks in aerospace engineering is problem involving time-varying *(nonautonomous)* systems. This type of problems requires a careful analysis since the use of 'standard' technique might indicate some *counter-intuitive* and peculiar figures. This can cause misleading identification that in turn may result in an inefficient design or otherwise inaccurate control decision. A better method is therefore desired to meet the higher demand of performance and the more stringent requirements for safety. To deal with time-varying system, many (if not most of) engineers today still tacitly use a 'frozen' analysis where the systems are assumed to be slowly varying. Under this assumption, the time-varying system is 'frozen' over various time intervals and treated as a constant coefficient system during each interval. While this approach may be acceptable for some slowly varying system, it will be inadequate for time-varying (or highly varying) systems in general. The latter category includes: dynamics of *reentry* vehicles such as space shuttle, hypersonic vehicles, ballistic missiles, etc.; dynamics Vertical/Short Take-Off and Landing $V/STOL$ aircraft; and space vehicles in general. A reentry vehicle undergoes variation in density with altitude and aerodynamic parameters which renders the system highly time-varying. Similarly, in the transition from hover to cruise, a VTOL aircraft exhibits time-varying aerodynamic properties. A large flexible space structures operating in years obviously falls under the class of nonlinear time-varying systems. During long time operation it may have experienced structural characteristic evolution that changes its dynamics.

To cope with time-varying problems, a system that can 'follow' the changing dynamics is thus desired. Artificial neural networks that provide a learning and adaptation ability will be one of the attractive approaches in handling such problems. Not only will it adapt to the changing system and environment but further also pave the way for the intelligent system design.

This chapter describes the use of the ANN for the simulation of second order model of GHAME vehicle. The case for the fourth order model is given in Chapter 3.

2.2 GHAME Vehicle Geometric and Trajectory Parameters

GHAME (Generic Hypersonic Aerodynamic Model Example) is a mathematical model of hypersonic vehicle that was developed in NASA to facilitate the increasing demand for more accurate and realistic aerodynamic data in that flight regime. The program was precursor of the well-known NASP project. The data contained in the GHAME provides the requisite of realistic simulation for the design of control and guidance systems as well as trajectory optimization. Those data are for a particular generic vehicle geometry and were developed as a combination of existing aircraft and theories. Actual data from vehicles such as the Space Shuttle Orbiter, lifting body type, as well as theories as the modified Newton impact flow method were employed in developing the final GHAME aerodynamic data [18]

The GHAME data was developed for a flight regime of single-stage-to-orbit (SSTO) mission. This mission entails the vehicle taking-off horizontally from conventional runaways accelerating to orbital velocities as an airbreathing aircraft and insertion into a low-Earth-orbit (LEO). Upon completion of this mission the aircraft would re-enter the atmosphere and maneuver to power-off horizontal landing. The GHAME basic configuration and parameters are given by [18].

In this study, the dynamics of the GHAME vehicle are examined as it traverses a prescribed trajectory returning it into the Earth's atmosphere. The trajectory employed is one which was originally designed to minimize the thermal-protection-system (TPS) weight of the Space Shuttle Orbiter 049 vehicle. The TPS of the Space Shuttle Orbiter consists of a collection of 22 metallic panels of varying composition and thickness. In order to obtain the optimal trajectory, the method of steepest descent was applied iteratively to minimize the total heat load at the stagnation point. The optimal trajectory produced a minimum TPS weight of 30,700 *lbs*.

Referring to [27], the optimal Shuttle trajectory is detailed in the following figures. The figures show the angle of attack, velocity, altitude, Mach number, flight path angle and some other related parameters as functions of the non-dimensional variable ξ. This non-dimensional variable is the number of vehicle lengths traversed along the trajectory. The Space Shuttle reentry trajectory covers a range of 165,000 to 290,000 vehicle lengths traversed and a descent from 225,245 ft at Mach number 20 to 100,000 ft at Mach number 3.2. The maximum acceleration does not exceed 3 $g's$. Angle of attack values vary between 20^0 to 30^0 while flight path angles vary from 0^0 to -4^0. The variation of real time with respect to ξ is nonlinear. The total time of the trajectory under consideration is approximately 950 seconds.

2.2.1 Related Parameters

The calculation starts by calculating the absolute temperature T as a function of ξ. Our objective is to have mach number M as a function of ξ. This needs information on the speed of sound $a\,(T)$. Fig. 3-3 of ref. [21] gives the absolute temperature as a function of altitude, $T = T\,(h)$. Using this information and having the $h = h\,(\xi)$ from the prescribed trajectory, we can calculate $T = T\,(\xi)$. The result is shown in Fig. 2-1 (top). Temperature variation against ξ is then used to calculate a by the relation:

$$a = \sqrt{\gamma R T} \tag{2.1}$$

where

$$\begin{aligned} \gamma &= \frac{C_p}{C_v} = \frac{\text{specific heat at constant pressure}}{\text{specific heat at constant volume}} \\ R &= \text{gas constant} \\ T &= \text{absolute temperature} \end{aligned}$$

For the air, Eqn. 2.1 can be expressed as [22],

$$a = 65.8\sqrt{T} ft/s \tag{2.2}$$

The result is depicted in Fig. 2-1 (bottom). Note that the flat part of the curve is due to the corresponding isothermal layer at altitude around 47 – 53 km in the atmosphere. Mach number M can finally be calculated by the following relation:

$$M = \frac{V}{a} \tag{2.3}$$

where the flight speed V is known from the prescribed trajectory, Fig. 2-2 (top). The bottom figure shows the corresponding mach number M.

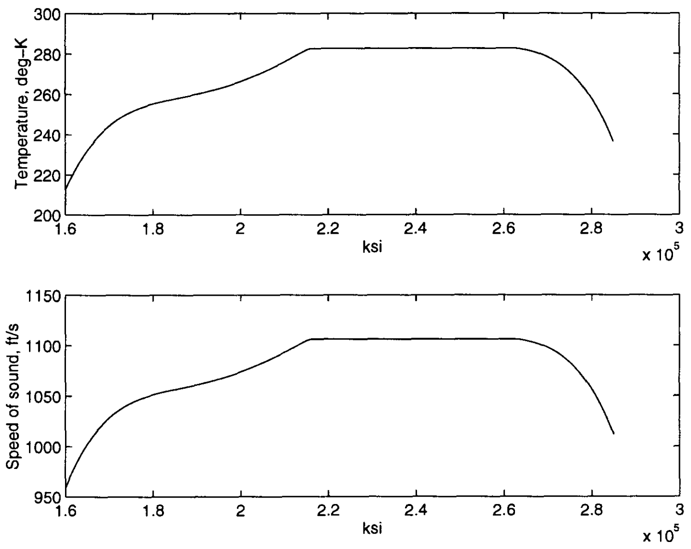


Figure 2-1: Absolute temperature T and speed of sound a as a function of ξ

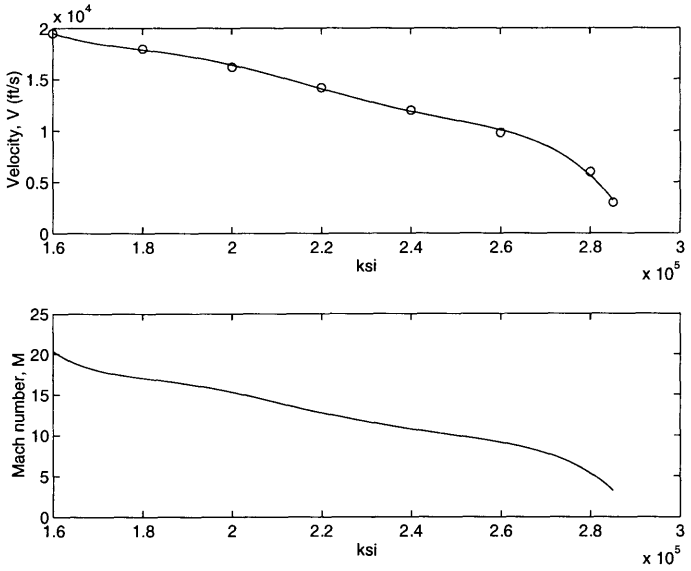


Figure 2-2: Velocity V and Mach number M as a function of ξ

The air density is calculated using the following formula [25]:

$$\rho(h) = \rho_0 \left(1 - \frac{\epsilon h}{T_0}\right)^{-1+g/R\epsilon} \tag{2.4}$$

in which ρ_0 and T_0 are density and absolute temperature at sea level. Parameter ϵ specifies the temperature gradient in the temperature versus altitude curve and is given in Fig. 3-3 of Ref. [21]. Parameter g is the acceleration of gravity which is in general a function of altitude h. Overall, we can express ρ variation against ξ since ρ_0, T_0 and ϵ are known and $h = h(\xi)$ is available from the prescribed trajectory, Fig. 2-3 (top). The air density variation along the prescribed trajectory is given in Fig. 2-3 (bottom).

2.3 Stability Derivatives

The linearized equations of aircraft motion can be expressed in terms of stability derivatives. These derivatives represent the changes in aerodynamic forces and moments due to small changes in the perturbation variables. As an illustration, the α derivatives describe the changes that take place in the forces and moments when the angle of attack is increased. This normally results in an increase in drag and a negative pitching moment. The stability derivatives are defined in terms of partial derivatives and expressed in terms of elementary aerodynamics parameters for simulation. The nondimensional stability derivatives is obtained by normalizing the derivatives with mass or the moments of inertia of the vehicle.

As an example, the L_{rl} derivatives are calculated below. $L_{rl} = C_l \frac{1}{2}\rho V^2 Sb$. Hence

$$\begin{aligned} L_v I_{xx} &= \left(\frac{\partial L_{rl}}{\partial v}\right)_0 = \frac{1}{2}\rho V^2 Sb \left(\frac{\partial C_l}{\partial v}\right)_0 \\ &= \frac{1}{2}\rho V Sb \left(\frac{\partial C_l}{\partial \beta}\right)_0 \\ &= \frac{1}{2}\rho V^2 Sb C_{l_\beta} \end{aligned}$$

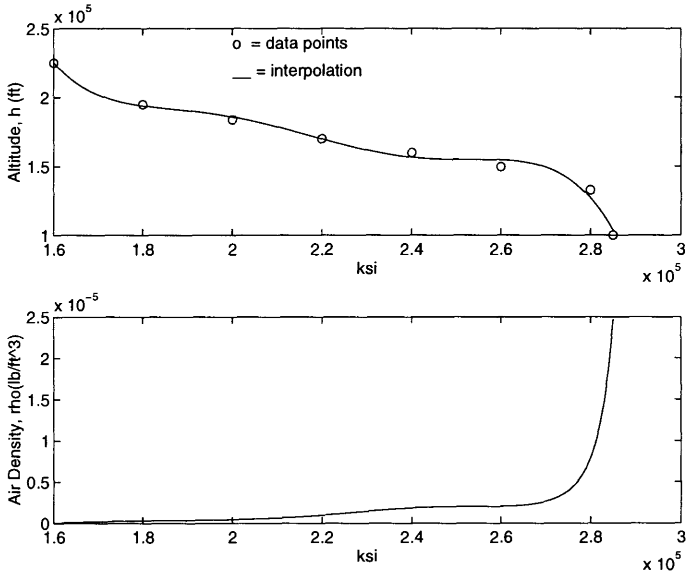


Figure 2-3: Altitude h and air density ρ as a function of ξ

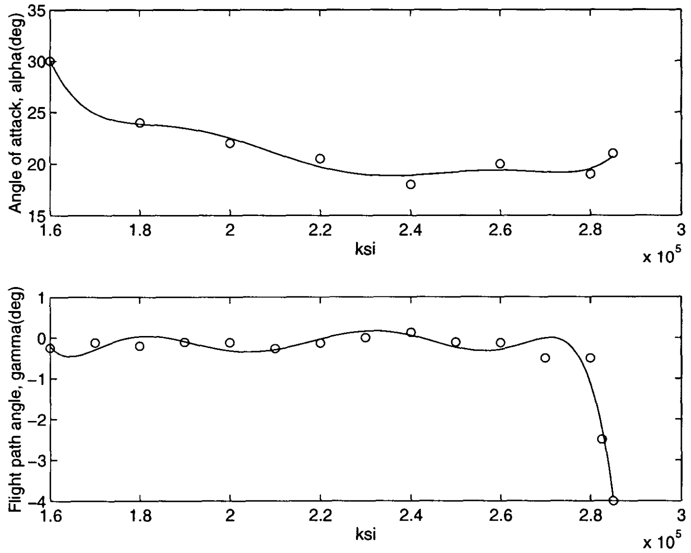


Figure 2-4: Angle of attack α_0 and flight path angle γ as a function of ξ

And

$$\begin{aligned} L_p I_{xx} &= \left(\frac{\partial L_{rl}}{\partial p}\right)_0 = \frac{1}{2}\rho V^2 Sb\left(\frac{\partial C_l}{\partial p}\right)_0 \\ &= \frac{1}{4}\rho V Sb\left(\frac{\partial C_l}{\partial \hat{p}}\right)_0 \\ &= \frac{1}{4}\rho V^2 Sb^2 C_{l_p} \end{aligned}$$

Similarly

$$\begin{aligned} L_r I_{xx} &= \left(\frac{\partial L_{rl}}{\partial r}\right)_0 = \frac{1}{2}\rho V^2 Sb\left(\frac{\partial C_l}{\partial r}\right)_0 \\ &= \frac{1}{4}\rho V^2 Sb^2 C_{l_r} \end{aligned}$$

Subscripts $-_0$ indicates the value is given at the equilibrium point. The remaining stability derivatives are evaluated in similar fashion and the complete results are listed in Table 2.1.

Table 2.1: Stability derivatives

Stability Derivative	Definition	Expression
D_u	$\frac{1}{m}\frac{\partial D}{\partial u}$	$\frac{1}{m}\rho VSC_D$
D_α	$\frac{1}{m}\frac{\partial D}{\partial \alpha}$	$\frac{1}{m}\rho VSC_{D_\alpha}$
L_u	$\frac{1}{m}\frac{\partial L}{\partial u}$	$\frac{1}{m}\rho VSC_L$
L_α	$\frac{1}{m}\frac{\partial D}{\partial \alpha}$	$\frac{1}{m}\rho VSC_{L_\alpha}$
M_u	$\frac{1}{I_{yy}}\frac{\partial M}{\partial u}$	$\frac{2gc_{m.a.c}}{VC_L{k_y}^2}\left(C_M + \frac{M}{2}\frac{\partial C_M}{\partial M}\right)$
M_α	$\frac{1}{I_{yy}}\frac{\partial M}{\partial \alpha}$	$\frac{1}{2I_{yy}}\rho V^2 Sc_{m.a.c.}C_{M_\alpha}$
$M_{\dot\alpha}$	$\frac{1}{I_{yy}}\frac{\partial M}{\partial \dot\alpha}$	$\frac{gc^2_{m.a.c}}{VC_L{k_y}^2}\frac{l_t}{c_{m.a.c}}\frac{d\epsilon}{d\alpha}C_{m_i t}$
$M_{\dot\theta}$	$\frac{1}{I_{yy}}\frac{\partial M}{\partial \dot\theta}$	$\frac{1}{4I_{yy}}\rho V^2 Sc_{m.a.c.}C_{M_q}$
Y_v	$\frac{1}{m}\frac{\partial Y}{\partial v}$	$\frac{g}{VC_L}C_{y_\beta}$
L_v	$\frac{1}{I_{xx}}\frac{\partial L_{rl}}{\partial v}$	$\frac{1}{2I_{xx}}\rho VSbC_{l_\beta}$
L_r	$\frac{1}{I_{xx}}\frac{\partial L_{rl}}{\partial r}$	$\frac{1}{4I_{xx}}\rho VSb^2C_{l_r}$
L_p	$\frac{1}{I_{xx}}\frac{\partial L_{rl}}{\partial p}$	$\frac{1}{4I_{xx}}\rho VSb^2C_{l_p}$
N_v	$\frac{1}{I_{zz}}\frac{\partial N}{\partial v}$	$\frac{1}{2I_{zz}}\rho VSbC_{n_\beta}$
N_r	$\frac{1}{I_{zz}}\frac{\partial N}{\partial r}$	$\frac{1}{4I_{zz}}\rho VSb^2C_{n_r}$
N_p	$\frac{1}{I_{zz}}\frac{\partial N}{\partial p}$	$\frac{1}{4I_{zz}}\rho VSb^2C_{n_p}$

2.4 Atmospheric Entry Equation of Motion: 2nd Order Longitudinal Dynamics

2.4.1 Perturbation Equations

This section primarily describes the equations of motion in the plane of symmetry. It is assumed that the vehicle experiences lift but does not roll and the planet is spherical. The axis system through the center of mass of the vehicle is such that the x - axis is always tangential to the instantaneous flight path. The equations of motion for arbitrary flight path angles and zero thrust are given by [26], [27].

Assuming that the slope of the lift curve is approximately independent of flight speed and Mach number at high supersonic speeds, the aerodynamics coefficients are linearized through a Taylor Series expansion about a nominal trajectory. After eliminating θ and V from the equations and change of the independent variable from t to ξ according to

$$L\dot{\xi} = V\left(t\right) \tag{2.5}$$

the equations of motion is transformed into the equation for perturbation angle-of-attack α after linearizing the aerodynamic coefficients. The result reads:

$$\alpha'' + \omega_1\left(\xi\right)\alpha' + \omega_0\left(\xi\right)\alpha = f\left(\xi\right) \tag{2.6}$$

where

$$\begin{aligned}
\omega_1\left(\xi\right) &= \delta\left[C_{L_\alpha} - \sigma\left(C_{m_\alpha} + C_{m_q}\right)\right] + \frac{V'}{V} \\
\omega_0\left(\xi\right) &= -\delta\left(\sigma C_{m_\alpha} + \frac{gl}{V^2}C_{D_\alpha}\cos\gamma\right)\delta' C_{L_\alpha} + \\
&\quad \delta\frac{V'}{V}C_{L_\alpha} - \delta^2\left[C_{L_\alpha}\left(\sigma C_{m_q} + C_{D_0}\right) + C_{L_0}C_{D_\alpha}\right] + \\
&\quad \frac{3l}{R}\left(\frac{gl}{V^2}\right)\nu\cos^2\left(\gamma + \alpha_0\right)
\end{aligned}$$

$$\begin{aligned} f\left(\xi\right) = \delta\left(\frac{gl}{V^2}\right)\left[C_{D_O} - \sigma C_{M_q}\left(1 - \frac{V^2}{gr}\right)\right]\cos\gamma - \delta' C_{L_O} + \\ \delta^2 C_{L_0}\left(C_{D_0} + \sigma C_{m_q}\right) - \\ \frac{gl}{V^2}\left[\left(\frac{3L}{r} - \frac{gl}{V^2}\right)\sin 2\delta + \frac{3L}{2r}\nu\sin 2\left(\gamma + \alpha_0\right)\right] \end{aligned}$$

The non-dimensional parameter are defined by

$$\delta = \frac{\rho S l}{2m}, \qquad \nu = \frac{I_{xx} - I_{zz}}{I_{yy}}, \qquad \sigma = \frac{ml^2}{I_{yy}} \tag{2.7}$$

The primes represent differentiation with respect to the new independent variable ξ. The terms ω_1, ω_0, and f are functions of parameters which depend on instantaneous flight conditions, and can be calculated if the trajectory of the center of mass is explicitly known. This implies that the effect of the angle-of-attack perturbations on the trajectory itself is negligible. Eqn. 2.6 is a second order linear differential equation with variable coefficients. It is, in general, impossible to be solved exactly. The arrangement of numerical data for calculating the instantaneous flight conditions is detailed in the following paragraphs.

2.4.2 Numerical Data

The numerical data for the GHAME are made available in Appendix D. The data from NASA were given as matrix representing aerodynamic coefficients as a function of mach number and angle of attack. The mach number were varied from 0.4 to 24 and angle of attack from -3^o to 21^o. The trajectory data from Ref. [27] is used in conjunction with the available data. It were originally given for the GHAME vehicle entering atmosphere at altitude $4 \times 10^5 ft.$ with the angle of attack around 54^o at the speed $2.6 \times 10^4 ft./s$. Only part of the data, where angle of attack data extrapolation is still considered valid, is used. This corresponds to range of variable ξ from 1.6×10^5 to 2.85×10^5. This flight segment describes the vehicle entering atmosphere at altitude $2.4 \times 10^5 ft.$ with angle of attack α around 30^o . The α varies in the range of 19^o – 30^o and the mach number M from 3 – 20. The available aerodynamic coefficient data is first extrapolated up to $\alpha = 30^o$. The bisection interpolation is then used to refine the data in the flight segment under interest. This is done to facilitate the two-dimensional polynomial fitting for every aerodynamic coefficient.

As an illustration, the calculation of C_L is outlined as follows. The three degree polynomial of α and M is used to fit the data in a least-squares sense. The resulted coefficients of the polynomial are in the form of matrix below:

```
p9 =

  2.3743e-009 -1.0528e-007  1.7013e-006 -2.1377e-005
 -1.5620e-007  6.9259e-006 -1.1192e-004  1.4063e-003
  9.2581e-007 -4.1050e-005  6.6337e-004 -8.3355e-003
 -3.4802e-005  1.8201e-003 -3.1509e-002  1.8989e-001
```

Thus, the three degree polynomial expression for C_L in α and M is

$$\begin{aligned} C_L = {} & 2.374 \times 10^{-9} M^3\alpha^3 - 1.0528 \times 10^{-7} M^3\alpha^2 + 1.701 \times 10^{-6} M^3\alpha - 2.138 \times 10^{-5} M^3 \\ & - 1.562 \times 10^{-7} M^2\alpha^3 + 6.9259 \times 10^{-6} M^2\alpha^2 - 1.119 \times 10^{-4} M^2\alpha + 1.406 \times 10^{-3} M^2 \\ & + 9.258 \times 10^{-7} M\alpha^3 - 4.105 \times 10^{-5} M\alpha^2 + 6.6337 \times 10^{-4} M\alpha - 8.3355 \times 10^{-3} M \\ & - 3.480 \times 10^{-5} \alpha^3 + 1.820 \times 10^{-3} \alpha^2 - 3.1509 \times 10^{-2} \alpha + 1.8989 \times 10^{-1} \end{aligned}$$

Similar procedures were done for all other aerodynamic coefficients. The degree of the polynomial for the fitting depends on the nature of coefficient variation. In general, for more complex variation, a higher degree polynomial is necessary to achieve a satisfactory result. The two-dimensional polynomials approximation for each coefficient are plotted below. Fig. 2-5 through Fig. 2-10 show the variation of aerodynamic coefficients against M and α in the flight segment under interest.

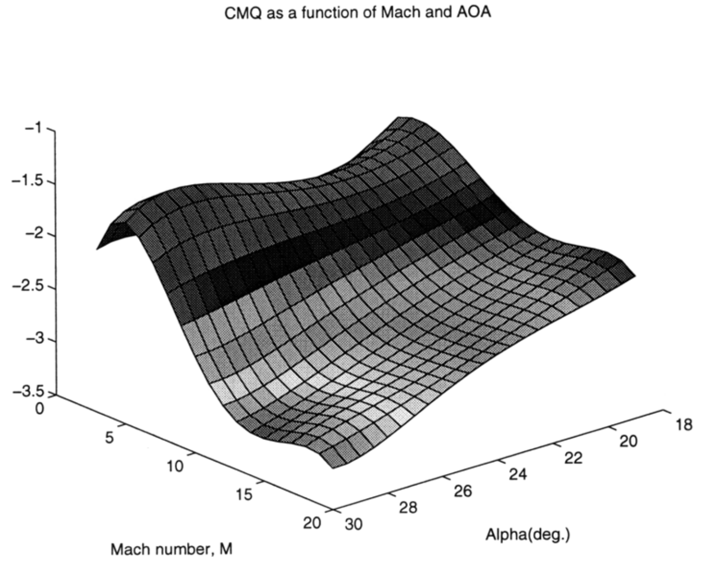


Figure 2-5: Aerodynamic coefficient C_{mq} as a function of M and α

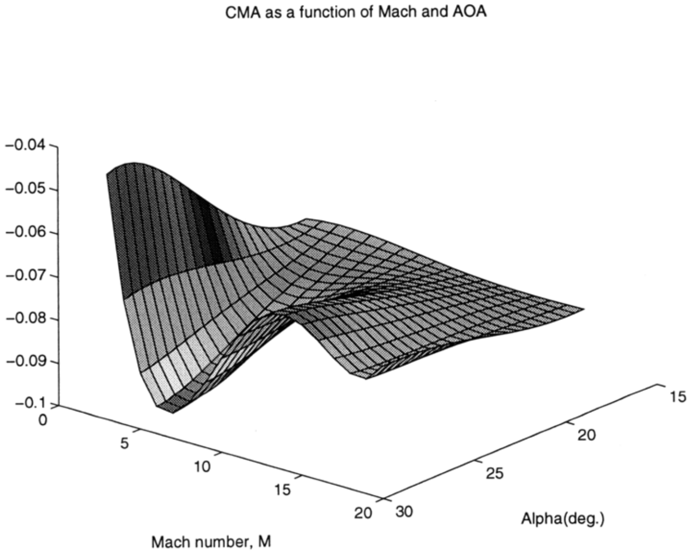


Figure 2-6: Aerodynamic coefficient $C_{m\alpha}$ as a function of M and α

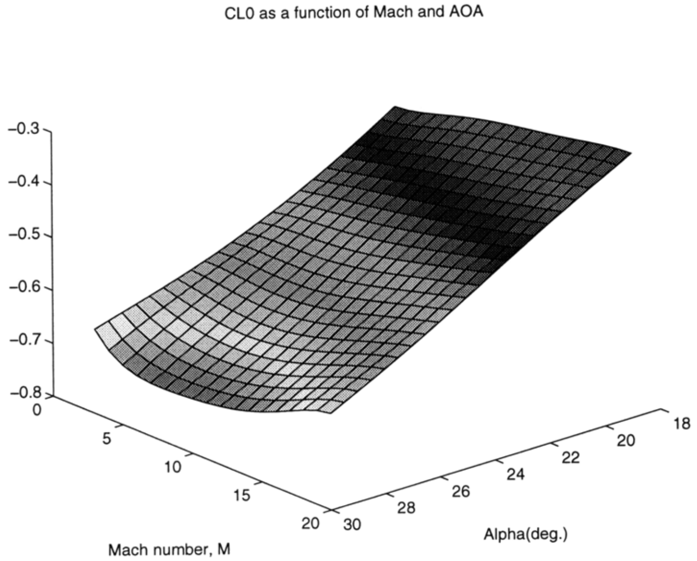


Figure 2-7: Aerodynamic coefficient C_{L0} as a function of M and α

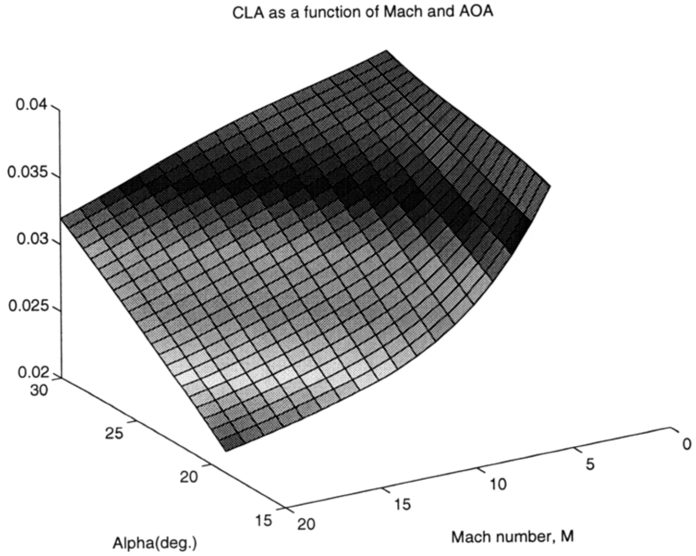


Figure 2-8: Aerodynamic coefficient $C_{L\alpha}$ as a function of M and α

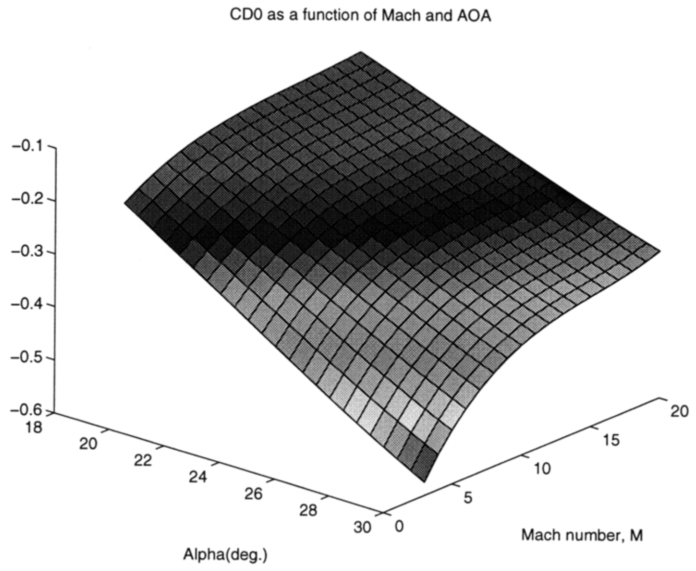


Figure 2-9: Aerodynamic coefficient C_{D0} as a function of M and α

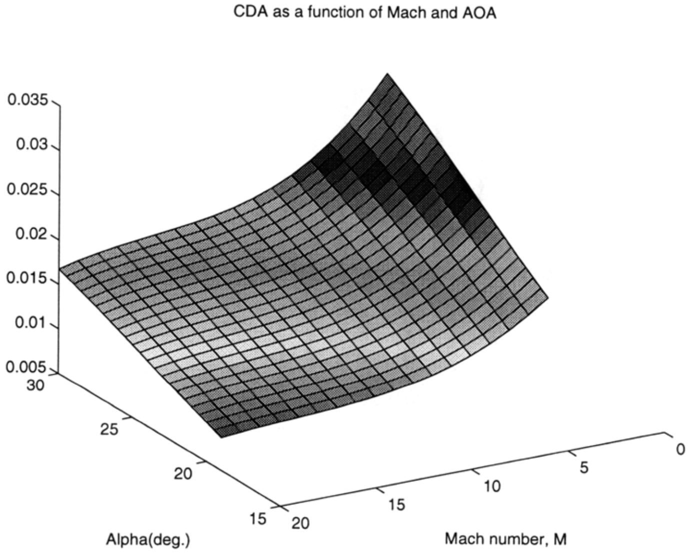


Figure 2-10: Aerodynamic coefficient $C_{D\alpha}$ as a function of M and α

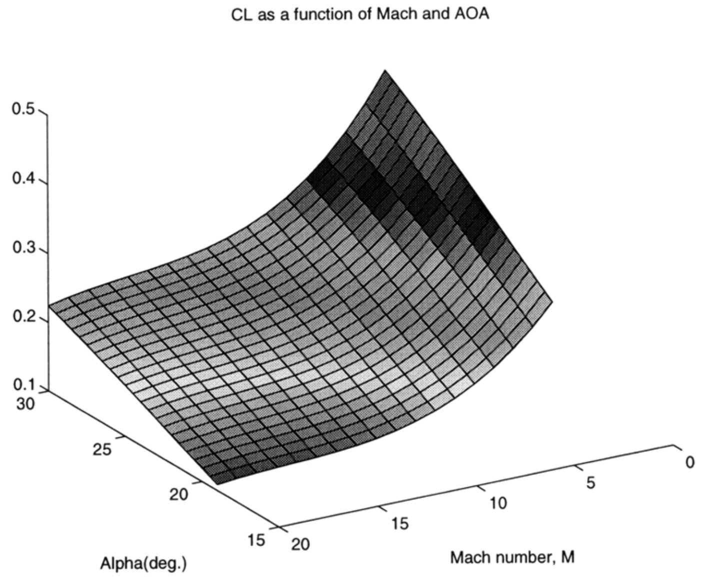


Figure 2-11: Aerodynamic coefficient C_L as a function of M and α

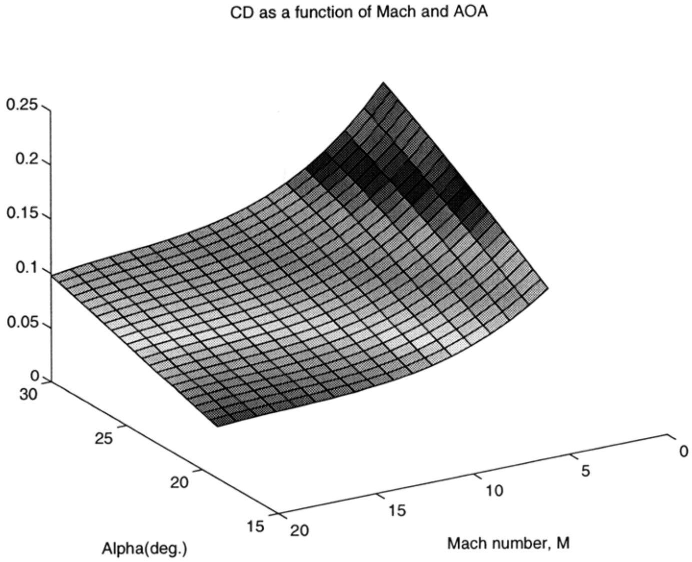


Figure 2-12: Aerodynamic coefficient C_D as a function of M and α

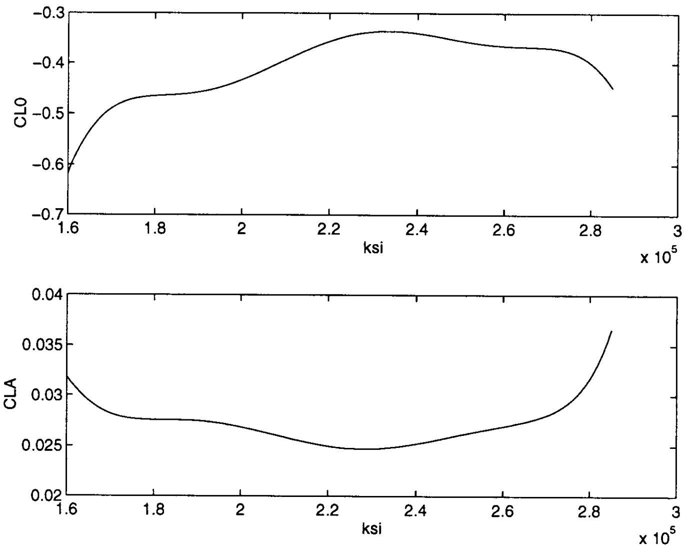


Figure 2-13: C_{L_0} and C_{L_α} as a function of ξ

Upon knowing the aerodynamic coefficients as functions of M and α, we can express them further in variable ξ. This can be done since we have calculated M and α as functions of ξ (See Fig. 2.3 and Fig. 2-4). The aerodynamic coefficients are plotted against ξ in the subsequent figures. Since flight conditions have been all expressed in ξ, we have all the necessary terms for calculating $\omega_0(\xi)$ and $\omega_1(\xi)$. Fig. 2-17 depicts the variation of $\omega_0(\xi)$ and $\omega_1(\xi)$ in the flight segment. Equation 2.6 is then solved numerically using Runge-Kutta. The integration is executed with SIMULINK and the diagram is displayed in Fig. 2-18. The numerical data yielded from the Runge-Kutta integration is saved for the purpose of neural networks training.

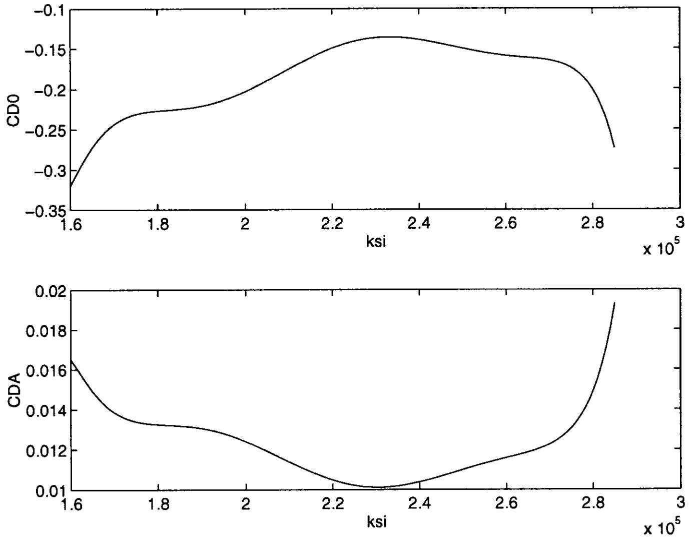


Figure 2-14: C_{D0} and $C_{L\alpha}$as a function of ξ

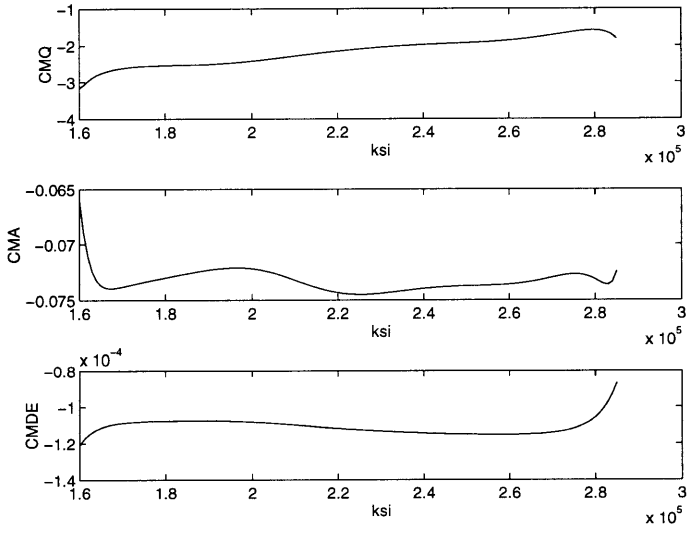


Figure 2-15: C_{m0} and $C_{m\alpha}$as a function of ξ

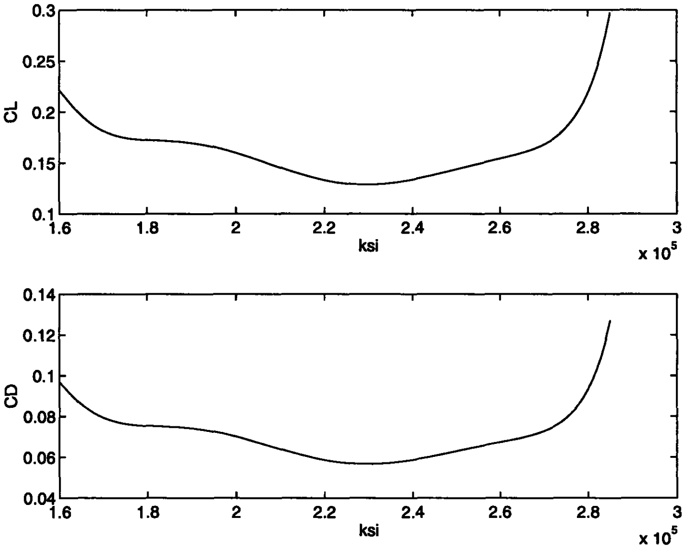


Figure 2-16: Aerodynamics coefficient C_L and C_D as a function of ξ

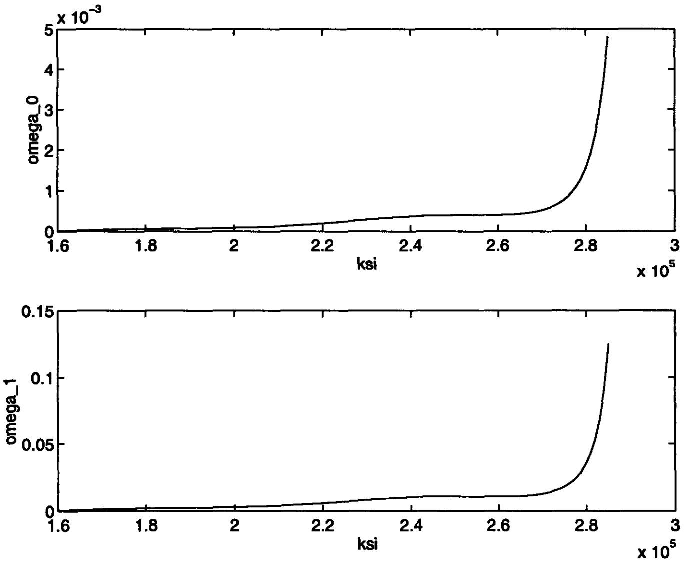


Figure 2-17: ω_1 and ω_0 as a function of ξ

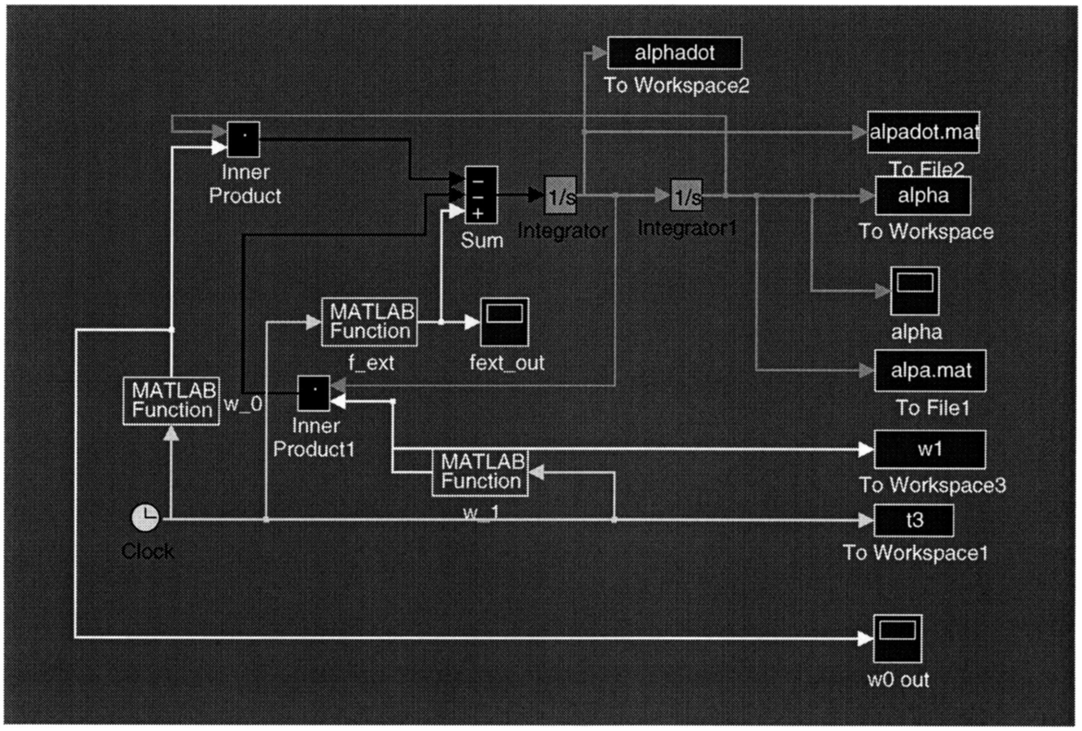


Figure 2-18: Simulation block diagram for 2-nd order longitudinal dynamics

2.5 Root Locus Analysis

The stability analysis is done by sketching the rootlocus and observing the corresponding time response behavior. Since the coefficients appearing in the differential equation of variable α is a function of ξ, the root locus of the system will also be a function of ξ.

2.5.1 Root Locus

The root locus of the second order model of GHAME is given in Fig. 2-19. At the early phase of the trajectory, the conjugate roots are located near the origin. As the vehicle transverses the prescribed flight trajectory, the roots move to the left half plane. In general the longitudinal dynamics of the vehicle is stable and characterized by damped oscillation. As can be observed from the figure, the damping and the frequency of the oscillation increase along the trajectory.

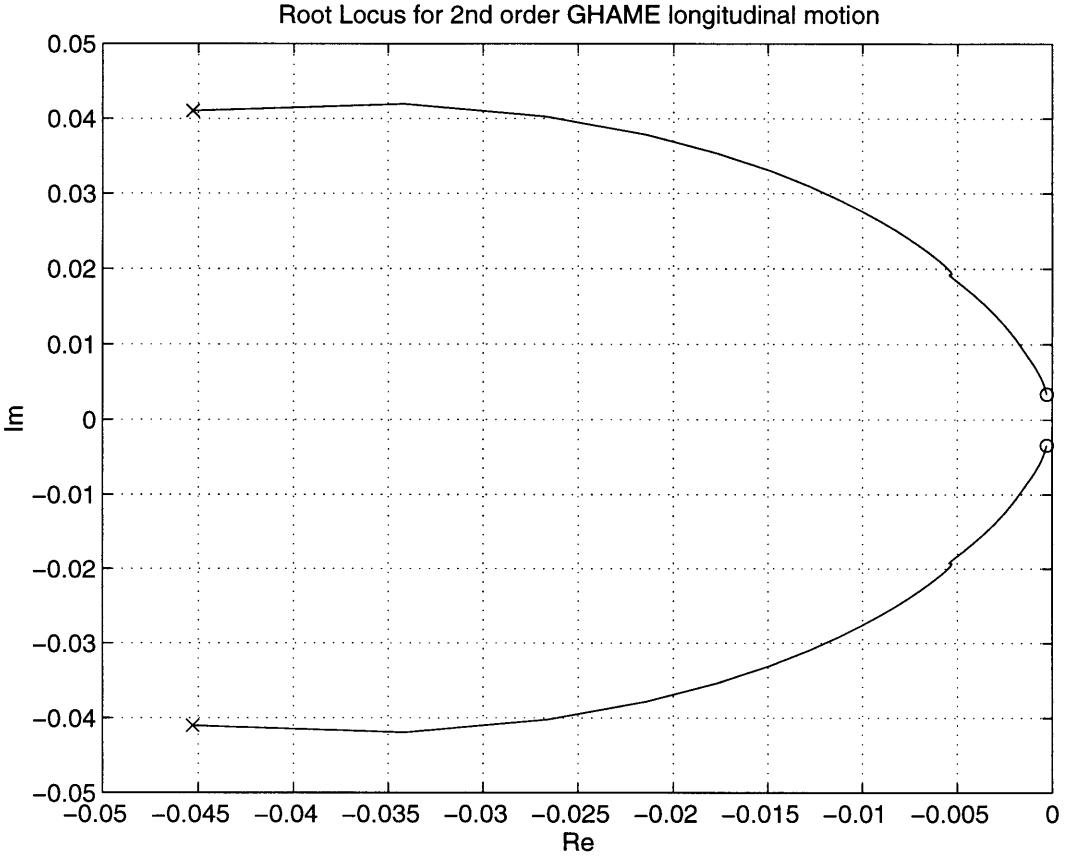


Figure 2-19: Root locus variation for 2-nd order longitudinal dynamics
○ ≡ beginning of the trajectory $\xi = 1.6.10^5$
× ≡ end of the trajectory $\xi = 2.85.10^5$

2.5.2 Time Response

The time responses for the 2nd order longitudinal dynamics are given by Fig. 2-20 – Fig. 2-23. The fourth order Runge-Kutta integration method is used in generating the time response of the system. The scheme of the simulation is depicted in Fig. 2-18. In agreement with the root locus analysis, the results show the damped oscillatory behavior of the system. The changes of the frequency is also present in the early stage of oscillation.

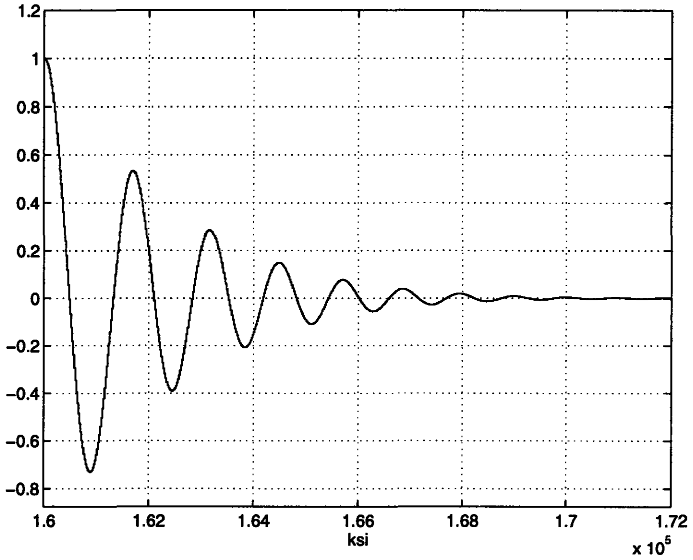


Figure 2-20: Variable α time response for 2-nd order longitudinal dynamics

$$\begin{aligned}\alpha\,(1.6.10^5) &\equiv 1\\ \dot{\alpha}\,(1.6.10^5) &\equiv 0\end{aligned}$$

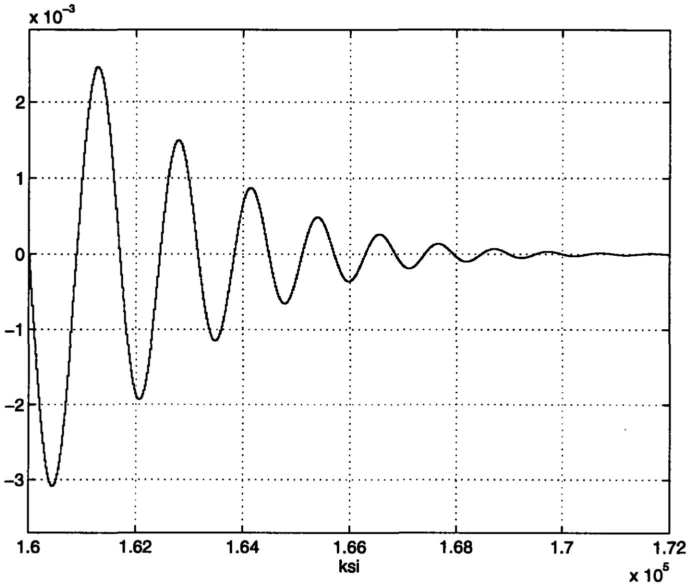


Figure 2-21: Variable $\dot{\alpha}$ time response for 2-nd order longitudinal dynamics

$$\begin{aligned}\alpha\,(1.6.10^5) &\equiv 1\\ \dot{\alpha}\,(1.6.10^5) &\equiv 0\end{aligned}$$

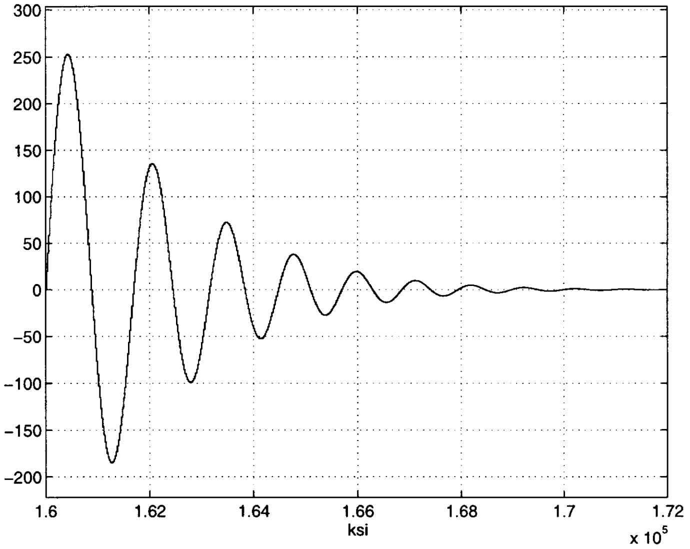


Figure 2-22: Variable α time response for 2-nd order longitudinal dynamics

$$\begin{aligned}\alpha\,(1.6.10^5) &\equiv 0\\ \dot{\alpha}\,(1.6.10^5) &\equiv 1\end{aligned}$$

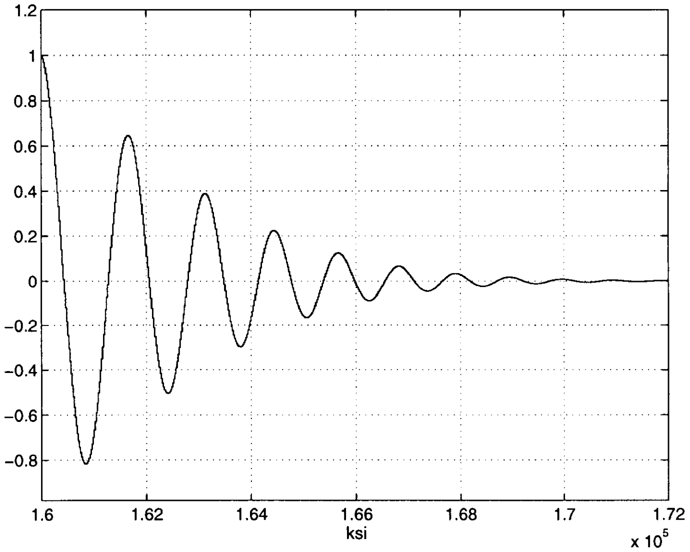


Figure 2-23: Variable $\dot{\alpha}$ time response for 2-nd order longitudinal dynamics

$$\begin{aligned}\alpha\,(1.6.10^5) &\equiv 0\\ \dot{\alpha}\,(1.6.10^5) &\equiv 1\end{aligned}$$

2.6 Sensitivity Analysis

In dynamics analysis of vehicles, it is often important to know how each parameter affects its motion. This study is called a sensitivity analysis. It studies the effect of an incremental change in a certain physical parameter to the motion of the vehicle. If the motion of the vehicle is described by state variable x, then we can write the sensitivity variable s as partial derivative of x with respect to a certain parameter p.

$$s = \frac{\partial x}{\partial p} \tag{2.8}$$

If the dynamics of the vehicle is written in the state space form,

$$\dot{x} = A\left(p\right)x \tag{2.9}$$

we can write the sensitivity equation as

$$\begin{aligned} \dot{s} &= As + A_p x \\ \dot{x} &= Ax \end{aligned}$$

Note that the above expressions are given in matrix equation. For a certain purpose (e.g. simulation) it is easier to write the scalar version.

The sensitivity of the GHAME vehicle second order angle-of-attack perturbation is carried out in that manner. The effects of aerodynamic coefficients C_{L_α}, C_{m_α}, and C_{m_q} are studied due to their known importance in the longitudinal dynamics. For the sake of completeness, the sensitivity of the longitudinal dynamics due to all other aerodynamic coefficients is also included to observe their relative significance to the above-mentioned coefficients.

The sensitivity equation of the GHAME can be derived starting from the general equation of motion:

$$\alpha'' + \omega_1\left(\xi\right)\alpha' + \omega_0\left(\xi\right)\alpha = 0 \tag{2.10}$$

It can be rearranged to include the sensitivity variable s as follows:

$$
\begin{aligned}
\alpha'' &= -\omega_1\left(\xi\right)\alpha' - \omega_0\left(\xi\right)\alpha \\
s'' &= -\omega_{1_p}\left(\xi\right)\alpha' - \omega_{0_p}\left(\xi\right)\alpha - \omega_1\left(\xi\right)s' - \omega_0\left(\xi\right)s \\
s'' &= -\omega_1\left(\xi\right)s' - \omega_0\left(\xi\right)s - \omega_{1_p}\left(\xi\right)\alpha' - \omega_{0_p}\left(\xi\right)\alpha
\end{aligned}
$$

In solving the differential equation, we need to first calculate the partial derivatives of the time-varying coefficients $\omega's$ with respect to the aerodynamic coefficients. As an illustration for C_{m_α}, the partial derivatives can be calculated as:

$$
\begin{aligned}
\frac{\partial \omega_1}{\partial C_{m_\alpha}} &= -\delta\sigma \\
\frac{\partial \omega_0}{\partial C_{m_\alpha}} &= -\delta\delta'\sigma C_{L_\alpha}
\end{aligned}
$$

After calculating all the necessary partial derivatives, again we can use the Runge-Kutta method to integrate the differential equation of variable s. Note that the initial conditions for the sensitivity differential equation are only given to the state variable α. The initial conditions for s are all zero. For comparison, notice that upon having the simulation scheme, we can use Eqn. 2.8 directly to find the sensitivity of each parameter. It can be carried out by perturbing the system with Δp where p is the aerodynamic coefficients. We can then take the output difference between the perturbed system and the original unperturbed system. Dividing the difference by the incremental perturbation Δp , we will get the sensitivity of parameter p. The following figures give the sensitivity of all the aerodynamic coefficients to the longitudinal motion of the GHAME vehicle. The two mentioned methods are compared.

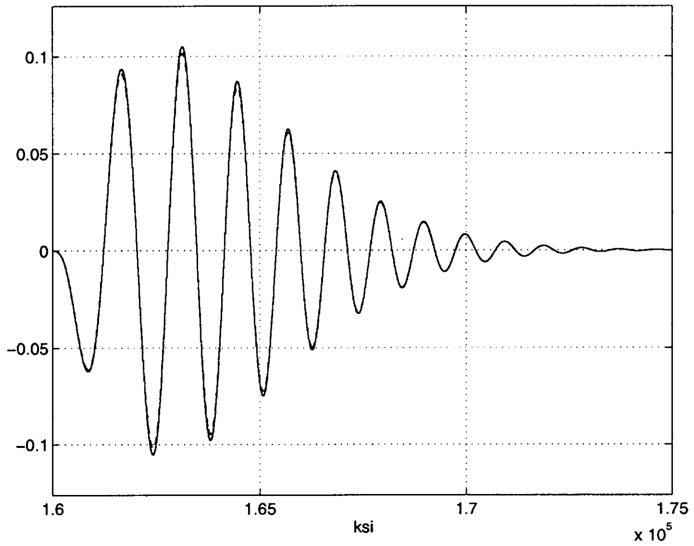


Figure 2-24: Sensitivity of 2-nd order longitudinal dynamics to C_{m_α}

—— $\equiv$ using the sensitivity differential equation

$- \cdot - \cdot -$ $\equiv$ using direct perturbation

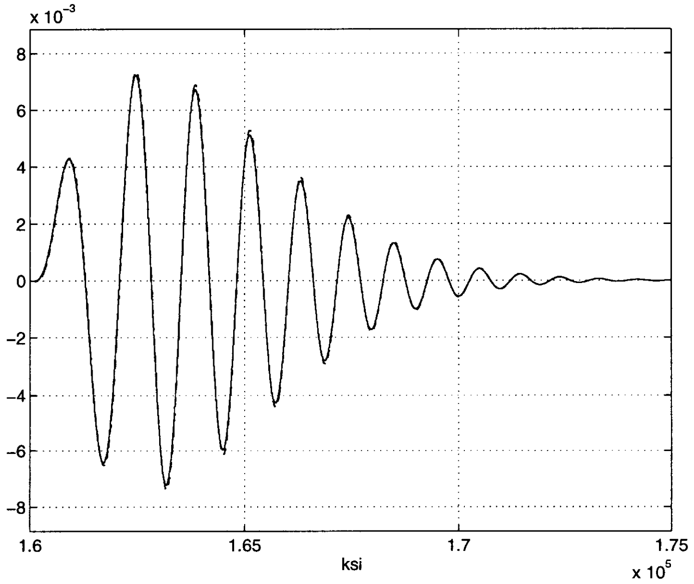


Figure 2-25: Sensitivity of 2-nd order longitudinal dynamics to C_{L_α}

—— $\equiv$ using the sensitivity differential equation

$- \cdot - \cdot -$ $\equiv$ using direct perturbation

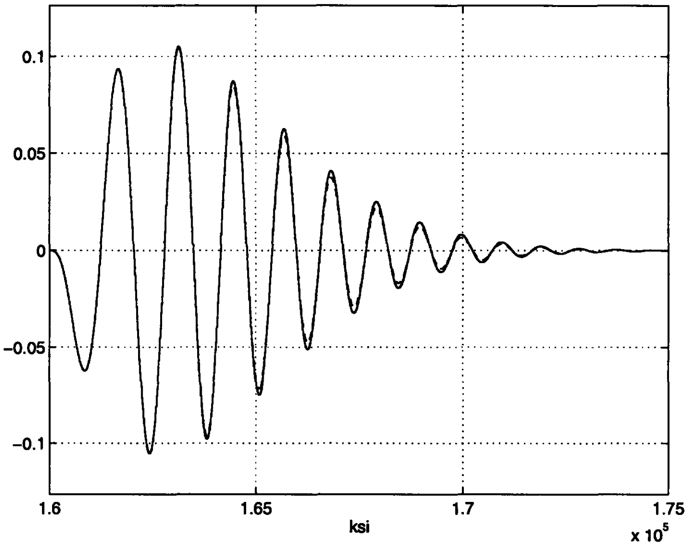


Figure 2-26: Sensitivity of 2-nd order longitudinal dynamics to C_{m_q}

—— ≡ using the sensitivity differential equation

— · — · — ≡ using direct perturbation

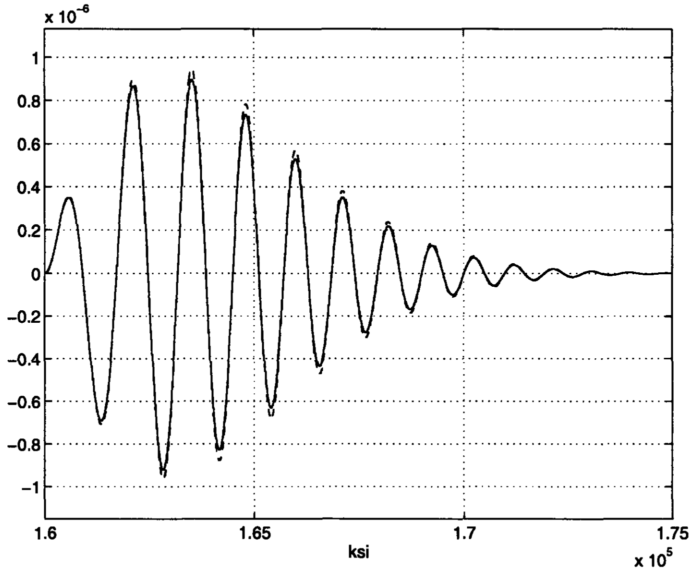


Figure 2-27: Sensitivity of 2-nd order longitudinal dynamics to C_{d_0}

—— ≡ using the sensitivity differential equation

— · — · — ≡ using direct perturbation

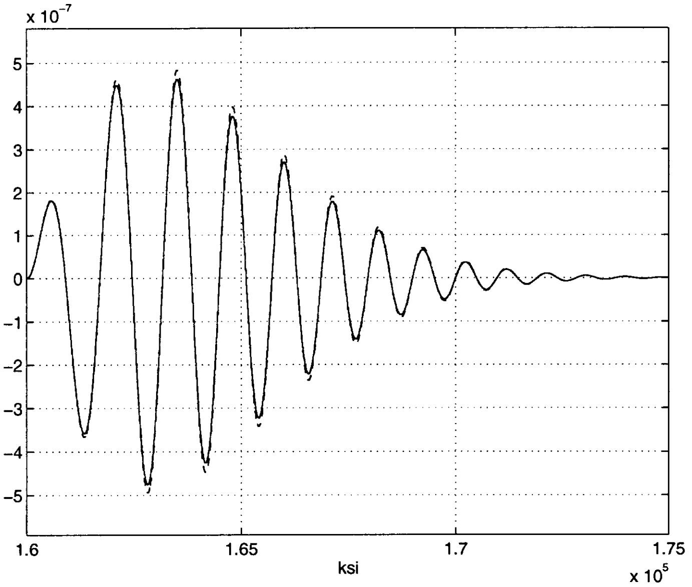


Figure 2-28: Sensitivity of 2-nd order longitudinal dynamics to C_{L0}

——— $\equiv$ using the sensitivity differential equation

— · — · — $\equiv$ using direct perturbation

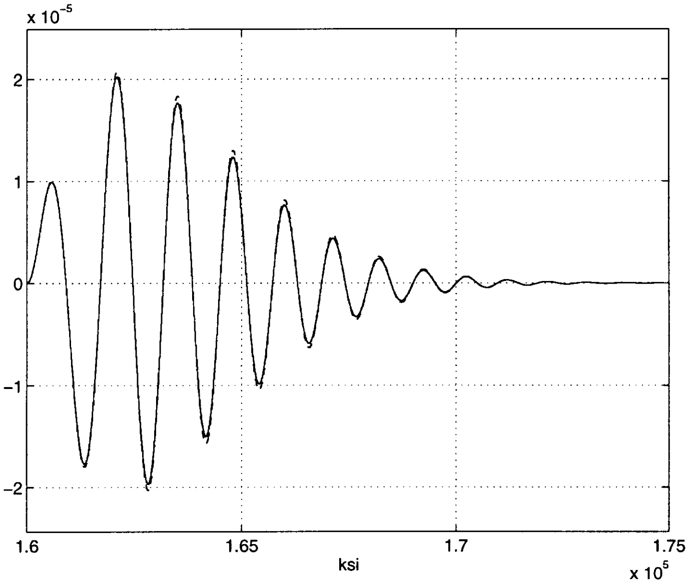


Figure 2-29: Sensitivity of 2-nd order longitudinal dynamics to C_{d_α}

——— $\equiv$ using the sensitivity differential equation

— · — · — $\equiv$ using direct perturbation

It can be observed from those plots, that the three dominant aerodynamic coefficients mentioned earlier indeed give higher order effects to the longitudinal dynamics. Sensitivity of C_{L_α} is of order 10^{-3}, whereas C_{m_α} and C_{m_q} are of order 10^{-1}. The other coefficients give much lower sensitivity, i.e $<< \mathcal{O}\,(10^{-6})$. In general, however, the reentry sensitivity to all aerodynamic coefficients behave in similar manner. They oscillate with the same frequency and all reach the maximum amplitude at $\xi = 1.62.10^5$, or approximately 3,000 vehicle lengths from the initial trajectory. This is followed by an amplitude decay until it decreases to zero after the vehicle travels about 15,000 vehicle lengths.

In conclusion, it is clearly evident that when concerned with the effect of changes in aerodynamic coefficients on GHAME vehicle longitudinal dynamics, greater consideration should be given to C_{m_α} and C_{m_q} than the other aerodynamic parameters.

2.7 The Use of ANN for System Identification

The ANN is used to copy the behavior of the second order GHAME vehicle dynamics. As alluded in previous chapter, the success of the control system employing neural network will be dependent on the success of using ANN as the system identification. In this work, we are just concerned with the ANN use for dynamics identification. The ANN model is trained using data from the prescribed (known) dynamics that has been derived in detail. Note that in reality, we can use ANN for the lesser known dynamic or system with uncertainties. During the training, ANN just need the vector pair of input and output of the system (the so called pattern) to learn the dynamic. Thus, it is clear that the performance of the neural networks in mimicking the dynamics will depend on the way the training material is given and the range of the patterns given during the training.

2.7.1 Network Training

The ANN is used to model the angle-of-attack perturbation variable α dynamics. The networks is inputted with two delayed signal of $\alpha\,(\xi)$ and the target (or output) is the signal $\alpha\,(\xi)$. Since the system is linear, the linear network is used. This kind of network can be quickly trained and suitable for the adaptive system. The results are presented in the following plot.

The figures show that the trained network gives very good performance in mimicking the true behavior of the second order GHAME vehicle dynamic. In the following section, we will test the network for the presence of the uncertainty in the system.

2.7.2 Network Testing

To test the robustness of the network model, the trained network is tested with the uncertainty appearing in the damping coefficient ω_1 (ξ). It is assumed that the system deviates from the behavior that was learned by the network during the training period. This is done by letting the ω_1 (ξ) increase by 10%. The trained network is now tested with this changing system. The result is presented in the following figure.

The figure shows that the neural network model still captures the dynamics of the GHAME vehicle though the ω_1 (ξ) increases by 10% from its original value. However, there is a certain limit at which the network will fail to cope with the changing system i.e. when the system deviates too far from the neighborhood of the training domain. To observe this phenomenon the similar network is again tested by the GHAME vehicle dynamics with ω_1 (ξ) increases by 20% . Figure 2-32 indicates that the networks can still follow the dynamics at the right frequency but fails to capture the amplitude.

To improve the performance of the networks, the material of the training is changed to include the information that some degree of uncertainty exists within the system. This can be likened to telling the networks about the presence of uncertainty. To emulate this, we introduce a white noise to ω_1 (ξ) for the training material of the networks. The behavior of the ω_1 (ξ) can be observed in the Fig. 2-33. Now, after training the network with white noise added to the ω_1 (ξ), the network is tested for system with 15% increase in ω_1 (ξ) and further with 20% increase in ω_1 (ξ). The results are subsequently shown in the following two figures.

The results indicates only slight improvement on the network performance in learning the dynamic of the system with 20% increase in ω_1 (ξ). This suggests that using a white noise during training is useful in giving the information about the presence of uncertainty to the network though it might not be satisfactory for the case when the system has larger scale of uncertainty. In real application, where the network has to learn online the uncertainties can be dealt inherently since any change present in the system will be captured during identification process. In the realm of neural network training, this can be done, for example, by back propagation through time learning method.

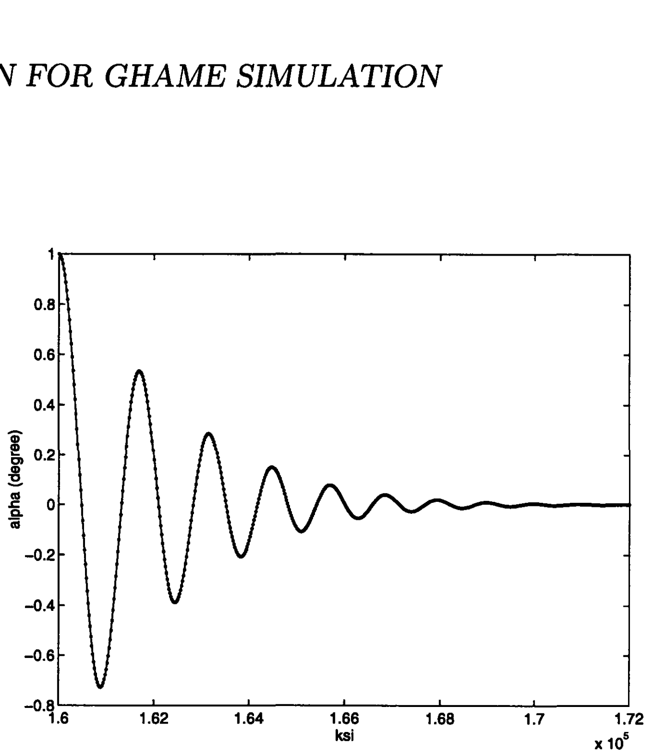


Figure 2-30: Neural Network Model for 2nd order GHAME

—— ≡ GHAME output

— · — · — ≡ Neural network model output

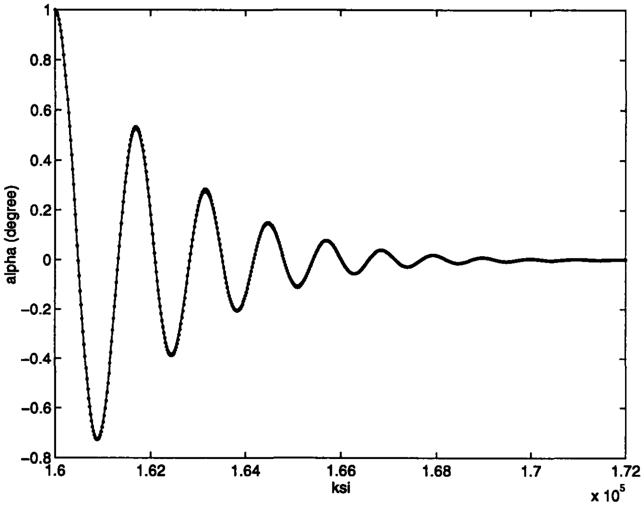


Figure 2-31: Neural Network Model for 2nd order GHAME with 10% increase in $\omega_1 (\xi)$

—— ≡ GHAME with 10% increase in $\omega_1 (\xi)$

— · — · — ≡ Neural network model output

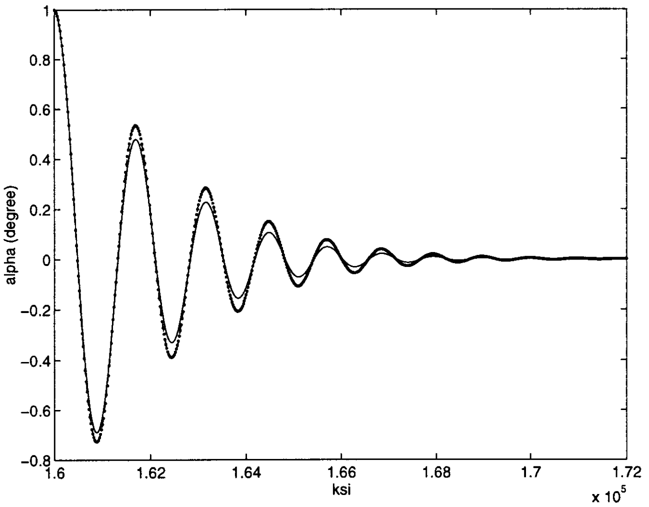


Figure 2-32: Neural Network Model for 2nd order GHAME with 20% increase in $\omega_1(\xi)$

——	≡	GHAME with 20% increase in $\omega_1(\xi)$
– · – · –	≡	Neural network model output

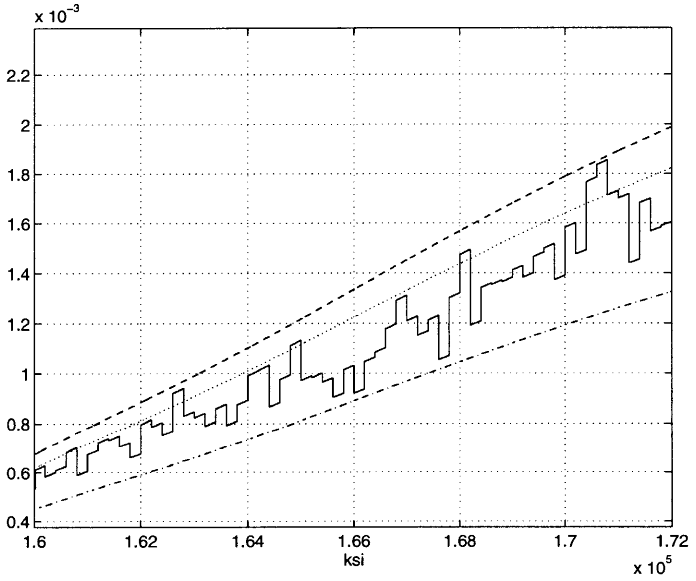


Figure 2-33: Damping Coefficient $\omega_1(\xi)$ with 20% uncertainty

——	≡	±20% uncertainty in $\omega_1(\xi)$
– · – · –	≡	20% decrease in $\omega_1(\xi)$
– – –	≡	20% increase in $\omega_1(\xi)$
....	≡	10% increase in $\omega_1(\xi)$

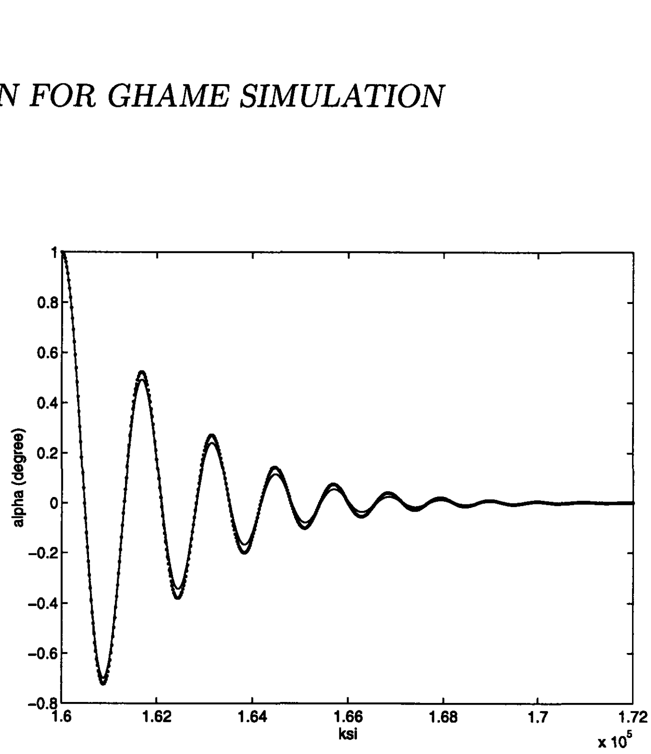


Figure 2-34: Neural Network Model for 2nd order GHAME with 15% increase in $\omega_1\,(\xi)$

—— ≡ GHAME with 15% increase in $\omega_1\,(\xi)$
— · — · — ≡ Neural network model output

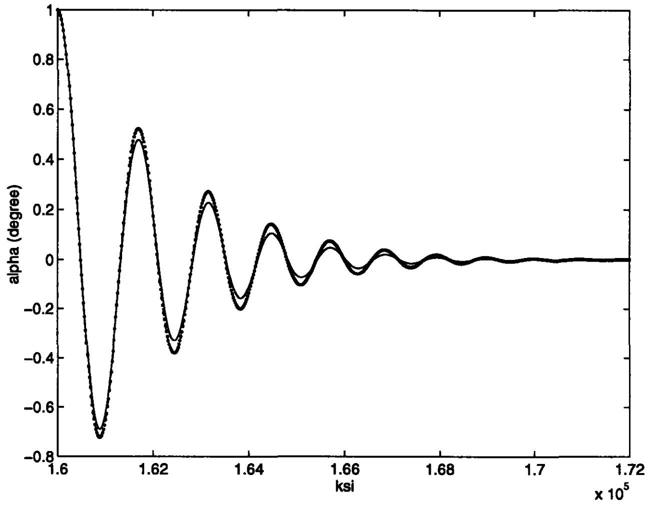


Figure 2-35: Neural Network Model for 2nd order GHAME with 20% increase in $\omega_1\,(\xi)$

—— ≡ GHAME with 20% increase in $\omega_1\,(\xi)$
— · — · — ≡ Neural network model output

Chapter 3

GHAME Vehicle Fourth Order Longitudinal Dynamics

The GHAME vehicle dynamics are described by nonlinear and time-varying differential equation. To reduce complexity, the equations are linearized about a specific α for a given Mach number at zero β. The process of linearization, leading to the use of the stability derivatives, are detailed in the following sections. Note that the linearized motion will sufficiently represent the true non-linear behavior if the motion of the vehicle is in the neighborhood of the equilibrium point or the perturbation variable are small. For large amplitude perturbation (like in spin and high angle of attack maneuver), nonlinear analysis is required.

3.1 Equations of Motion

Following Ref. [29], the general longitudinal dynamics of a flight vehicle can be described by three equations which are obtained through balancing the lift and drag forces as well as moments acting on the vehicle, below:

$$-m\dot{V} + (T - D) - W(\theta - \alpha) = 0 \tag{3.1}$$

$$L - W - mV\left(\dot{\theta} - \dot{\alpha}\right) = 0 \tag{3.2}$$

$$M_y - I_{yy}\ddot{\theta} = 0 \tag{3.3}$$

Eqn. 3.1 is the drag equation and represents a balancing of forces in the direction of flight. Lift equation expressed in Eqn. 3.2 describes the sum of forces in a direction perpendicular to the flight path. And the moment equation shown in Eqn. 3.3 balances the moments experienced by the vehicle about its center of mass. In order to linearize these equations of motion, the flight parameters α, V, and θ are represented as perturbations about a nominal flight condition, as follows:

$$\alpha = \alpha_0 + \Delta\alpha \tag{3.4}$$
$$V = V_0 + \Delta V \tag{3.5}$$
$$\theta = \theta_0 + \Delta\theta \tag{3.6}$$

Here, α_0, V_0, and θ_0 are the nominal flight conditions. The forces of thrust, lift and drag as well as the aerodynamic moment are expressed about a nominal flight condition through expansion in a convergent Taylor series. Taylor series expansion allows the forces and moment to be written as

$$L = L_0 + \frac{\partial L}{\partial \alpha}\Delta\alpha + \frac{\partial L}{\partial V}\Delta V + \ldots \tag{3.7}$$
$$T = T_0 + \frac{\partial T}{\partial V}\Delta V + \ldots \tag{3.8}$$
$$D = D_0 + \frac{\partial D}{\partial \alpha}\Delta\alpha + \frac{\partial D}{\partial V}\Delta V + \ldots \tag{3.9}$$
$$M = M_0 + \frac{\partial M}{\partial \alpha}\Delta\alpha + \frac{\partial M}{\partial V}\Delta V + L_0 + \frac{\partial M}{\partial \dot{\alpha}}\Delta\dot{\alpha} + \frac{\partial M}{\partial \dot{\theta}}\Delta\dot{\theta} + \ldots \tag{3.10}$$

L_0, T_0, D_0, and M_0 are nominal values of the aerodynamic forces and moment which produce the equilibrium flight condition given by α_0, V_0, and θ_0. Eqn. 3.1 – 3.10 are substituted into the general nonlinear equations of motion and after some manipulation results in the following equations:

$$\Delta\dot{V} + \Delta V\,(D_V - T_V) + \Delta\alpha\,(D_\alpha - g) + g\Delta\theta = 0 \tag{3.11}$$
$$\left(\frac{L_V}{V_0}\right)\Delta V + \Delta\dot{\alpha} + \left(\frac{L_\alpha}{V_0}\right)\Delta\alpha - \Delta\dot{\theta} = 0 \tag{3.12}$$
$$-M_V\Delta V - M_{\dot{\alpha}}\Delta\dot{\alpha} - M_\alpha\Delta\alpha + \Delta\ddot{\theta} - M_{\dot{\theta}}\Delta\dot{\theta} = 0 \tag{3.13}$$

The parameters D_V, D_α, T_V, L_V/V_0, L_α/V_0, M_V, M_α, and $M_{\dot{\theta}}$ appearing in the equations above are longitudinal stability derivatives of the vehicle and vary with time as the flight conditions change along the reentry trajectory. These derivatives have been defined in Table 2.1. They are basically partial derivatives of aerodynamic force or moment with respect to the flight parameter in question normalized by mass or moment of inertia.

3.2 Longitudinal Stability Derivatives

Referring to Table 2.1, the first eight stability derivatives are longitudinal stability derivatives. In the analysis of vehicle dynamics and stability some of the these derivatives are subdominant to the other terms and thus are assumed to be zero. In our analysis, these include the thrust velocity derivative, T_V, the angle-of-attack damping derivative, $M_{\dot{\alpha}}$ and speed stability term, M_V. Note that $u = v$, and it explains the notation M_u, D_u and L_u.

From the expression, it is obvious that the stability derivatives are functions of vehicle geometric parameters, trajectory parameters and vehicle aerodynamic characteristic. The first class of parameters is assumed to be constant. The others are varying with time as the vehicle moves along the reentry trajectory. The variation of trajectory parameters has been shown in Fig. 2.3 – Fig. 2-4. The figures describe the velocity V, air density ρ and other related parameters as a function of ξ, the number of vehicle lengths traversed along the trajectory. The variation of aerodynamic characteristic against ξ are shown subsequently in Fig. 2-13 – Fig. 2-15. As a result, we have all the requisite to express the stability derivatives as functions of ξ. The following figures display the longitudinal stability derivatives of the GHAME vehicle along the prescribed reentry trajectory.

There are some useful "rules of thumb" for obtaining practical insight from the vehicle stability derivatives. This rules are basically a rough measure of stability based on the accumulated experiences with (conventional) aircraft design. For conventional flight vehicle, D_V , D_α and L_α are usually positive and of small magnitude. L_V is larger than the latter and positive. In this respect, the GHAME vehicle does not exhibit anomalies. All value of those derivatives are positive. It is also evident that L_V is dominant over other parameters. The tendency of those derivatives, shows a close resemblance to that of air density variation along reentry trajectory. This means that the air density plays a major influence in determining these derivatives.

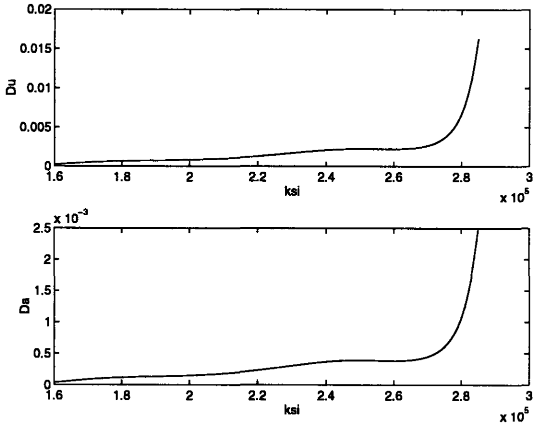


Figure 3-1: Stability derivative D_u and D_α as a function of ξ

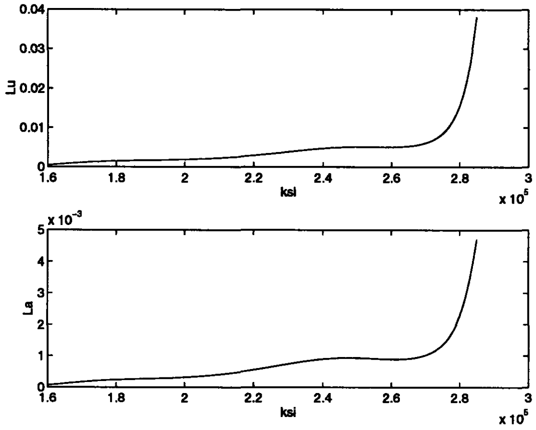


Figure 3-2: Stability derivative L_u and L_α as a function of ξ

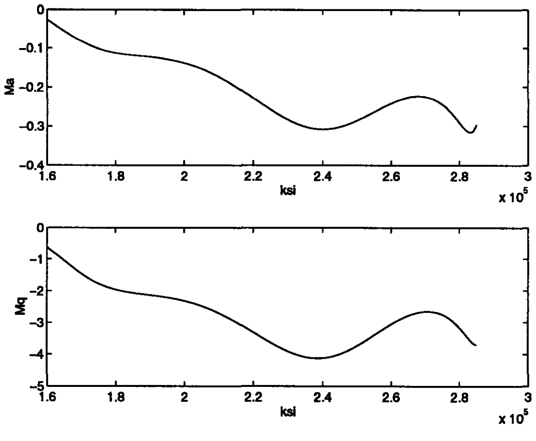


Figure 3-3: Stability derivative M_α and M_q as a function of ξ

Three stability derivatives that significantly determine vehicle longitudinal stability are M_α, $M_{\dot\theta}$, and M_V. For conventional vehicle configuration these parameters are usually negative. If M_α is positive the vehicle is statically unstable. On the other hand, the vehicle is statically unstable when M_V is negative and dynamically unstable when M_V is positive. Thus, M_V is desired to be of very small number or close to zero. In our analysis M_V is of $\mathcal{O}\left(10^{-6}\right)$ and thus neglected. Fig. 3-3 shows that both M_α and $M_{\dot\theta}$ (or M_q) are negative. Both exhibit similar tendency. In the early phase of reentry, their values are close to zero and they increase as the vehicle flies along the prescribed trajectory.

3.3 Solution to the Equations of Motion

To solve the equation of motion, Eqns. 3.11 – 3.13 are first simplified and written into the standard state space form:

$$\begin{bmatrix} s+D_V-T_V & D_\alpha - g & g \\ L_V/V_0 & s+L_\alpha/V_0 & -s \\ -M_V & -\left(M_{\dot\alpha}s+M_\alpha\right) & s\left(s-M_{\dot\theta}\right) \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta\alpha \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.14}$$

Variable s denotes the derivative operator $\frac{d}{dt}$. Again, stability derivatives appearing in the matrix vary with time leading to a time-varying system. It can not be solved with the standard method for solving the time-invariant (constant coefficient) cases.

Solution to the equations of motion is obtained by calculating the determinant of the main matrix in the above equation. Ignoring the derivatives T_V , we have

$$det\begin{bmatrix} s+D_V-T_V & D_\alpha - g & g \\ L_V/V_0 & s+L_\alpha/V_0 & -s \\ -M_V & -M_\alpha & s\left(s-M_{\dot\theta}\right) \end{bmatrix} = s^4+\omega_3 s^3+\omega_2 s^2+\omega_1 s+\omega_0 \tag{3.15}$$

For a steady flight condition ω_3, ω_2, ω_1 and ω_0 are given by:

$$\omega_3 = L_\alpha/V_0 - M_{\dot\theta} + D_V \tag{3.16}$$

$$\omega_2 = D_V\left(L_\alpha/V_0\right) - D_V M_{\dot\theta}\left(L_\alpha/V_0\right) - M_\alpha - D_\alpha\left(L_\alpha/V_0\right) + g\left(L_V/V_0\right) \tag{3.17}$$

$$\omega_1 = M_V D_\alpha - M_\alpha D_V - D_V M_{\dot\theta}\left(L_\alpha/V_0\right) + D_\alpha M_{\dot\theta}\left(L_V/V_0\right) - gM_{\dot\theta}\left(L_V/V_0\right) \tag{3.18}$$

$$\omega_0 = g\left[M_V\left(L_\alpha/V_0\right) - M_\alpha\left(L_V/V_0\right)\right] \tag{3.19}$$

and s is again the derivative operator $\frac{d}{dt}$. Under the assumption that the longitudinal stability derivatives in the above equations are constant, the full longitudinal response of the vehicle is described by setting the above determinant equal to zero and replacing the higher order s with their corresponding higher order derivatives. When flight conditions are changing, the decoupled scalar equations for ΔV, $\Delta\theta$, and $\Delta\alpha$ are of the general form (3.20) with different $\omega_i(t)$.

$$\frac{d^4y}{dt^4} + \omega_3(t)\frac{d^3y}{dt^3} + \omega_2(t)\frac{d^2y}{dt^2} + \omega_1(t)\frac{dy}{dt} + \omega_0(t)\,y = 0 \qquad (3.20)$$

This equation is dominant approximation to the actual longitudinal behavior and is made under the assumption that the stability derivatives in Eqns. 3.16 – 3.19 are constant. It turns out that the coefficients appearing in Eqn. 3.20 are actually time varying. For slowly varying flight conditions the coefficients along the reentry trajectory are shown in the following figures.

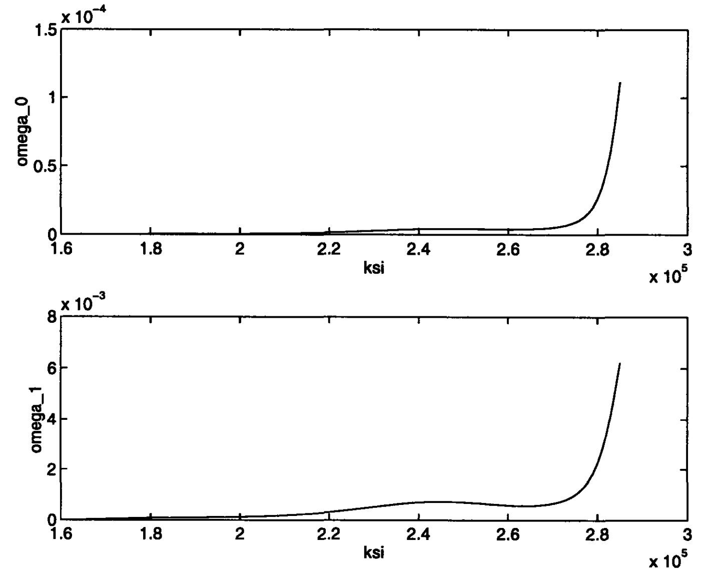


Figure 3-4: Time-varying coefficient ω_0 and ω_1 as a function of ξ

Numerical integration using Adam-Gear technique is employed to solve the above linear time-varying equation. The technique is typically used for what so called stiff system, where the characteristic roots are widely separated. The block diagram for the simulation of longitudinal dynamics of the GHAME vehicle during atmospheric reentry is shown in Fig. 3-6

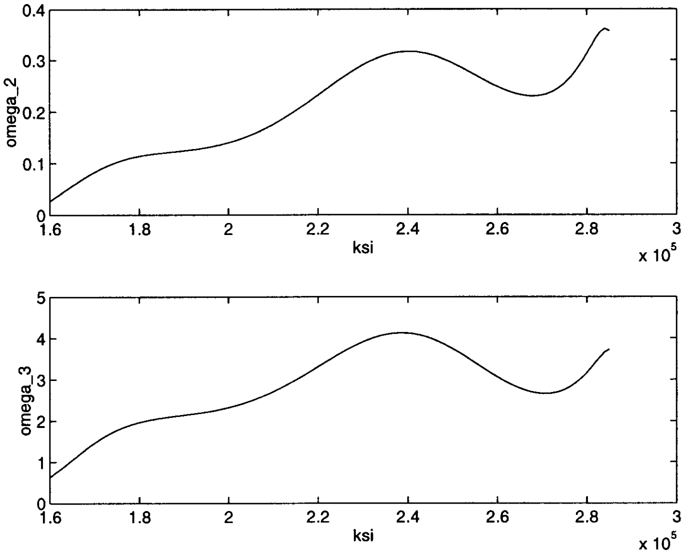


Figure 3-5: Time-varying coefficient ω_2 and ω_3 as a function of ξ

3.4 Root Locus Analysis

Like in the case of 2nd order GHAME vehicle dynamics, the stability analysis is done by sketching the rootlocus and observing the corresponding time response behavior. Since the coefficients appearing in the differential equation of variable α is a function of ξ, the root locus of the system will also be a function of ξ.

3.4.1 Root Locus

The root locus diagrams of the fourth order model of GHAME are given in Fig. 3-7 and Fig. 3-8. The figures depict the nature of a stiff system i.e. the characteristic roots are widely separated. At the early phase of the trajectory, there is a pair of conjugate roots located near the origin. These conjugate roots represent the phugoid mode which is usually characterized by slow and lightly damped motion. Two other roots are located in negative real axis around -0.045 and -0.6 respectively. As the vehicle transverses the prescribed flight trajectory, the roots move to the left half plane. The frequency of the phugoid mode increases from around 0.001 up to 0.015, while the damping rises from around zero to -0.008. The real roots move further to about -0.1 and -3.5 which gives substantial increment in the damping of this mode.

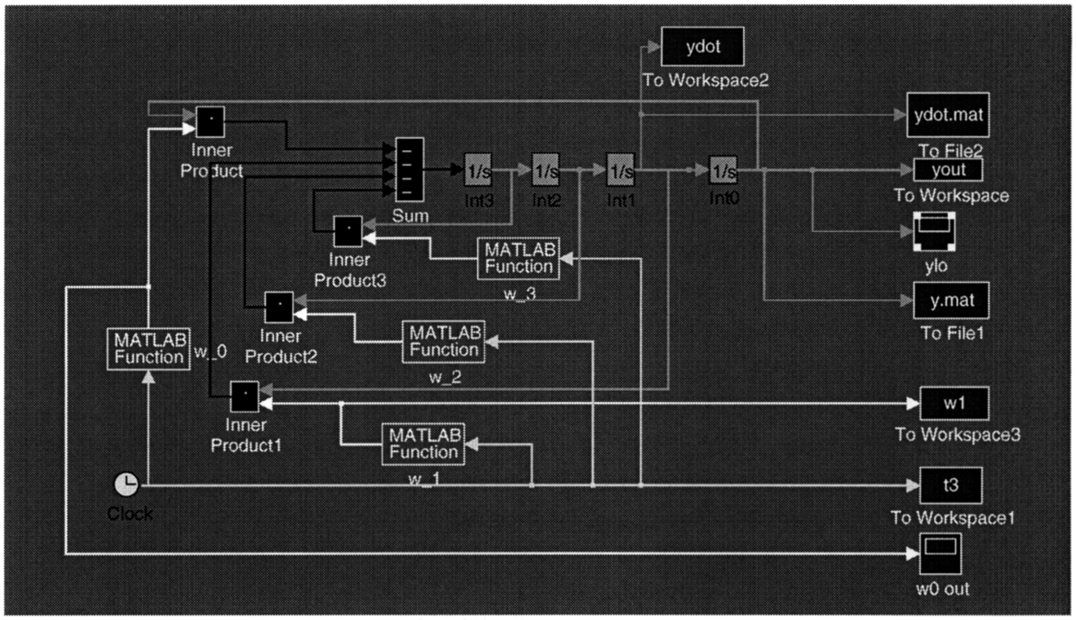


Figure 3-6: Simulation block diagram for 4-th order longitudinal dynamics

In general the longitudinal dynamics of the vehicle are stable and are characterized by damped oscillations. As can be observed from the figure, the damping and the frequency of the oscillation increase along the trajectory.

3.4.2 Time Response

The time responses of the 4th order longitudinal dynamics for different initial conditions are given by Fig. 3-9 – Fig. 3-16. As mentioned earlier, the Adam-Gear integration method is used in generating the time response of the system. The scheme of the simulation is depicted in Fig. 3-6. In agreement with the root locus analysis, the results show the damped oscillatory behavior of the system. The frequency changes in the early stage of oscillation.

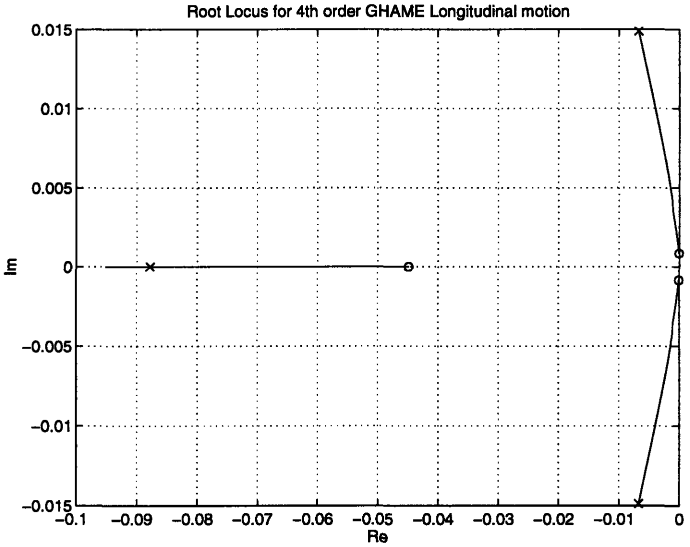


Figure 3-7: Root locus variation for 4-th order longitudinal dynamics: plot of phugoid roots and one real root

$\circ \quad \equiv \quad$ beginning of the trajectory $\xi = 1.6.10^5$

$\times \quad \equiv \quad$ end of the trajectory $\xi = 2.85.10^5$

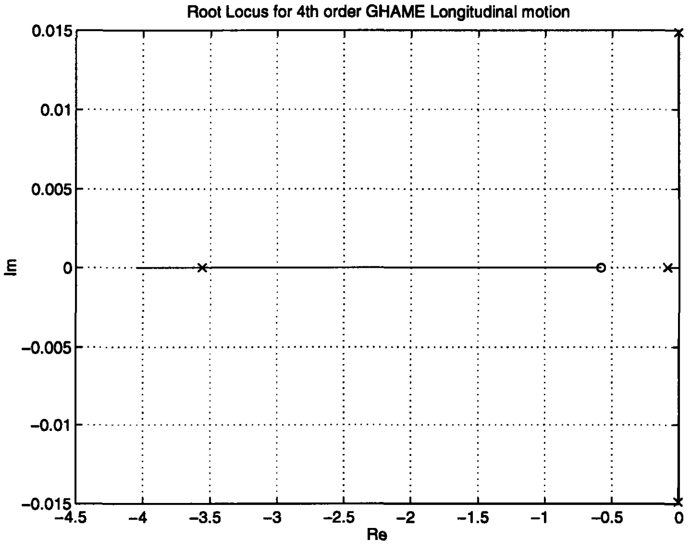


Figure 3-8: Root locus variation for 4-th order longitudinal dynamics: plot of the complete roots

$\circ \quad \equiv \quad$ beginning of the trajectory $\xi = 1.6.10^5$

$\times \quad \equiv \quad$ end of the trajectory $\xi = 2.85.10^5$

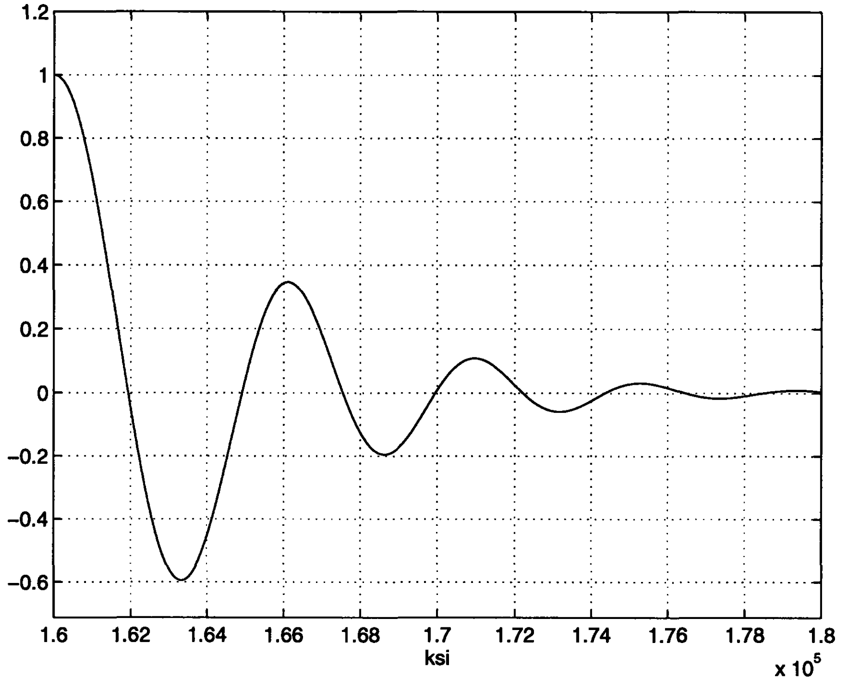


Figure 3-9: Time response of y for 4th order longitudinal dynamics

$$\begin{aligned} y\,(1.6.10^5) &\equiv 1 \\ y'\,(1.6.10^5),\ y''\,(1.6.10^5),\ y'''\,(1.6.10^5) &\equiv 0 \end{aligned}$$

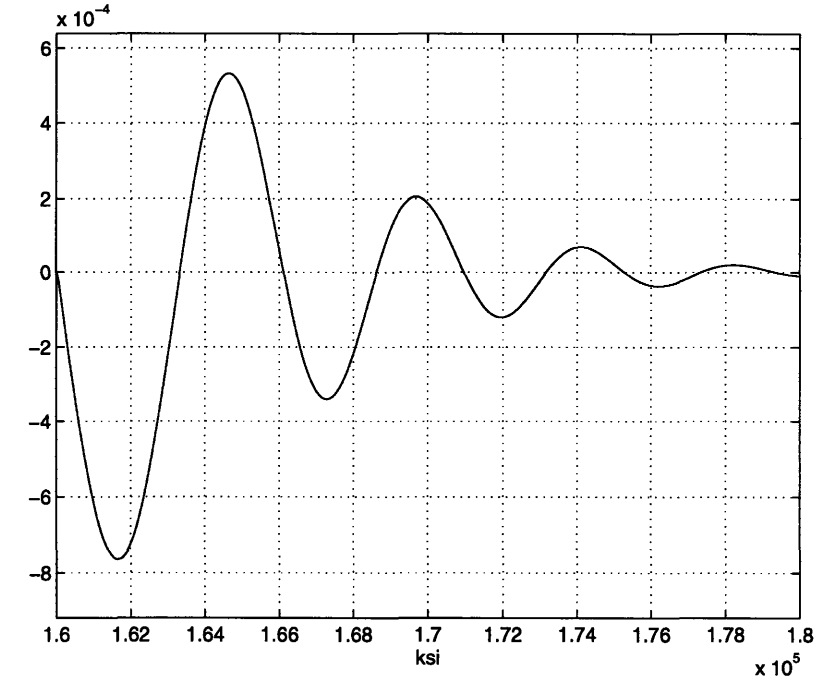


Figure 3-10: Time response of y' for 4th order longitudinal dynamics

$$\begin{aligned} y\,(1.6.10^5) &\equiv 1 \\ y'\,(1.6.10^5),\ y''\,(1.6.10^5),\ y'''\,(1.6.10^5) &\equiv 0 \end{aligned}$$

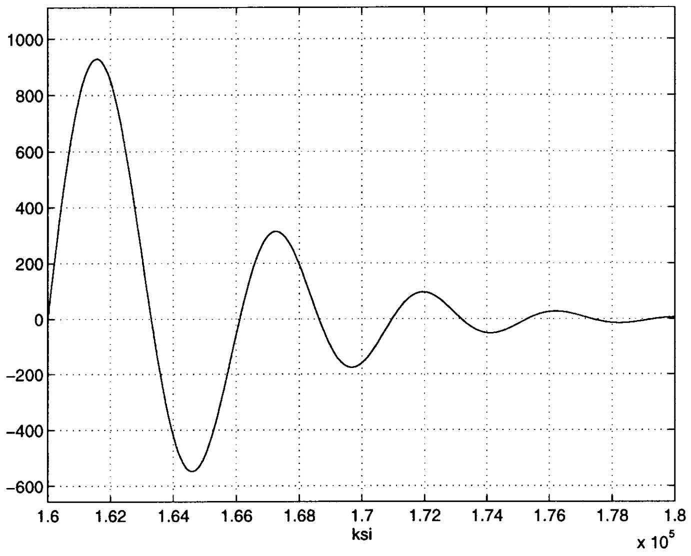


Figure 3-11: Time response of y for 4th order longitudinal dynamics

$$\begin{aligned} y'\,(1.6.10^5) &\equiv 1 \\ y\,(1.6.10^5),\ y''\,(1.6.10^5),\ y'''\,(1.6.10^5) &\equiv 0 \end{aligned}$$

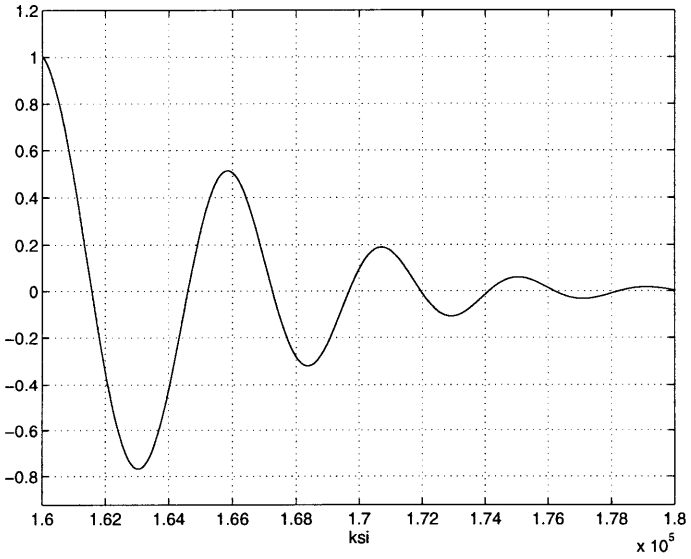


Figure 3-12: Time response of y' for 4th order longitudinal dynamics

$$\begin{aligned} y'\,(1.6.10^5) &\equiv 1 \\ y\,(1.6.10^5),\ y''\,(1.6.10^5),\ y'''\,(1.6.10^5) &\equiv 0 \end{aligned}$$

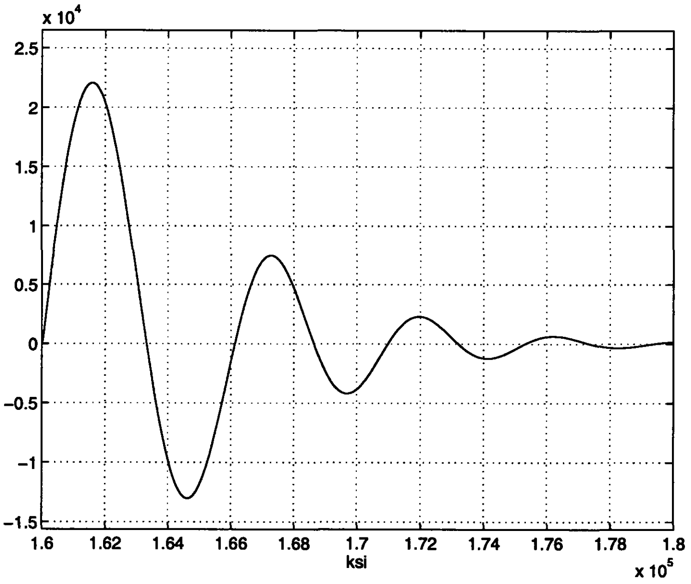


Figure 3-13: Time response of y for 4th order longitudinal dynamics

$$\begin{aligned} y''(1.6.10^5) &\equiv 1 \\ y'(1.6.10^5),\ y'(1.6.10^5),\ y'''(1.6.10^5) &\equiv 0 \end{aligned}$$

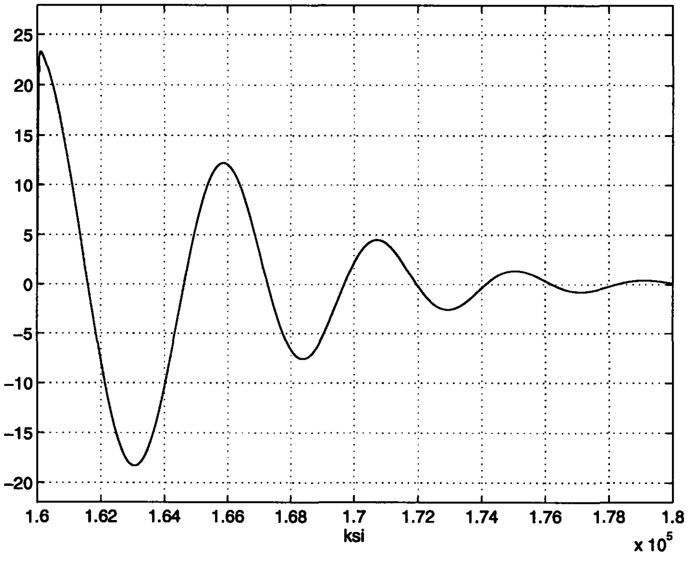


Figure 3-14: Time response of y' for 4th order longitudinal dynamics

$$\begin{aligned} y''(1.6.10^5) &\equiv 1 \\ y'(1.6.10^5),\ y'(1.6.10^5),\ y'''(1.6.10^5) &\equiv 0 \end{aligned}$$

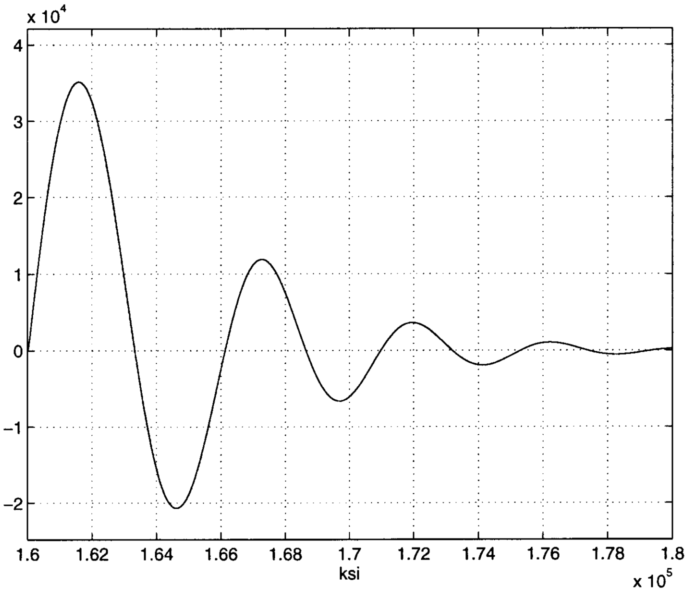


Figure 3-15: Time response of y for 4th order longitudinal dynamics

$$\begin{array}{lcl} y'''(1.6.10^5) & \equiv & 1 \\ y(1.6.10^5),\, y'(1.6.10^5),\, y''(1.6.10^5) & \equiv & 0 \end{array}$$

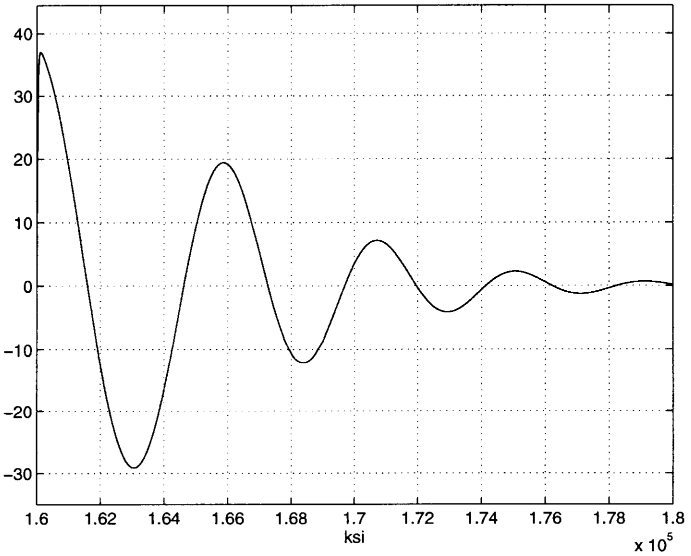


Figure 3-16: Time response of y' for 4th order longitudinal dynamics

$$\begin{array}{lcl} y'''(1.6.10^5) & \equiv & 1 \\ y(1.6.10^5),\, y'(1.6.10^5),\, y''(1.6.10^5) & \equiv & 0 \end{array}$$

3.5 Sensitivity Analysis

The sensitivity of the fourth order GHAME vehicle dynamics is carried out in similar manner as that of the second order case. For the sake of completeness, the sensitivity of the general response of the state variable to all stability derivatives are studied to observe their corresponding effects. This includes vertical dampingL_α, D_α, angle-of-attack static stability parameter M_α, lift-velocity derivative L_u, drag stability derivative D_u, and pitch damping M_q

The sensitivity equation of the 4th order GHAME vehicle dynamics can be derived starting from the general equation of motion:

$$\frac{d^4y}{dt^4} + \omega_3 \frac{d^3y}{dt^3} + \omega_2 \frac{d^2y}{dt^2} + \omega_1 \frac{dy}{dt} + \omega_0 y = 0 \tag{3.21}$$

or can be also written as

$$y'''' + \omega_3 y''' + \omega_2 y'' + \omega_1 y' + \omega_0 y = 0 \tag{3.22}$$

It can be rearranged to include the sensitivity variable s as follows:

$$\begin{aligned}
y'''' &= -\omega_3 y''' - \omega_2 y'' - \omega_1 y' - \omega_0 y \\
s'''' &= -\omega_{3_p} y''' - \omega_{2_p} y'' - \omega_{1_p} y' - \omega_{0_p} y \\
&\quad - \omega_3 y'''_p - \omega_2 y''_p - \omega_1 y'_p - \omega_0 y_p \\
s'''' &= -\omega_3 s''' - \omega_2 s'' - \omega_1 s' - \omega_0 s - \omega_{3_p} y''' - \omega_{2-p} y'' - \omega_{1_p} y' - \omega_{0_p} y
\end{aligned}$$

In solving the differential equation, we need to first calculate the partial derivatives of the time-varying coefficients $\omega' s$ with respect to the stability derivatives. After calculating all the necessary partial derivatives, again we can use the Adam-Gear method to integrate the differential equation of variable s. Note that the initial conditions for the sensitivity differential equation are only given to the state variable y. The initial conditions for s are all zero. The following figures give the sensitivity of the longitudinal motion of the GHAME vehicle to all the aerodynamic coefficients.

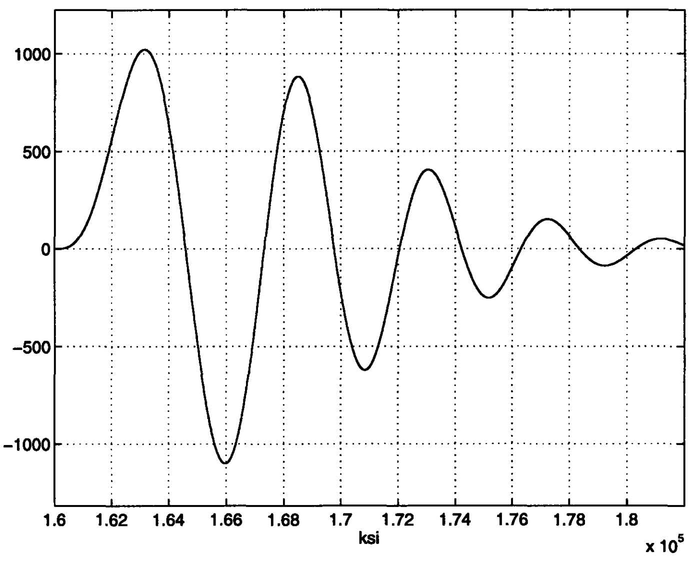


Figure 3-17: Sensitivity of 4th order longitudinal dynamics to D_u

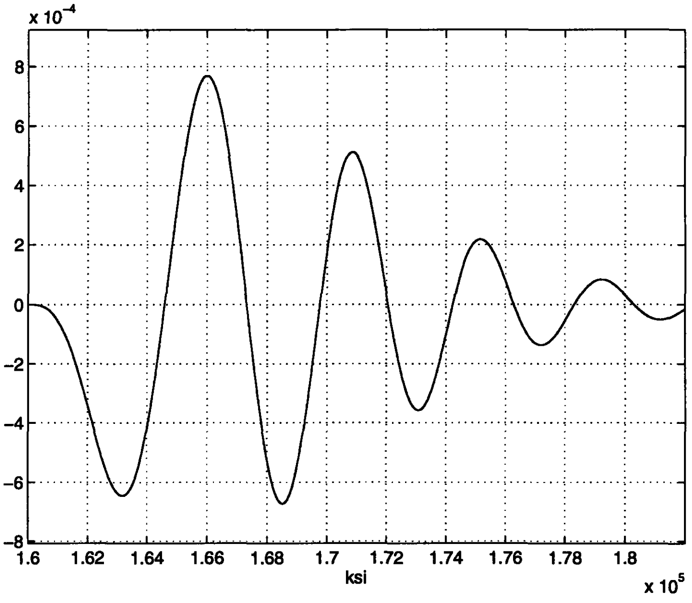


Figure 3-18: Sensitivity of 4th order longitudinal dynamics to D_α

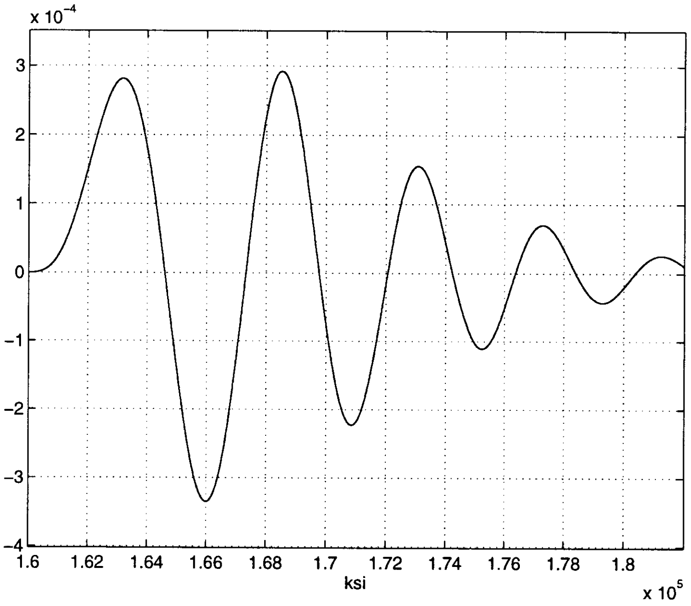


Figure 3-19: Sensitivity of 4th order longitudinal dynamics to L_{α}

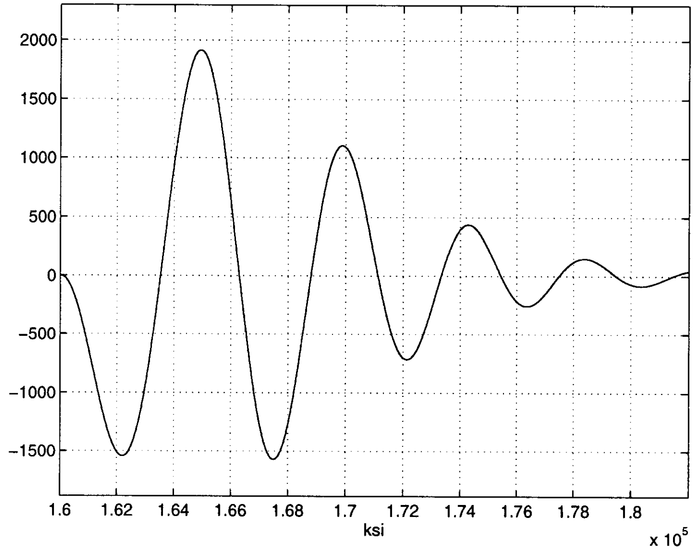


Figure 3-20: Sensitivity of 4th order longitudinal dynamics to L_u

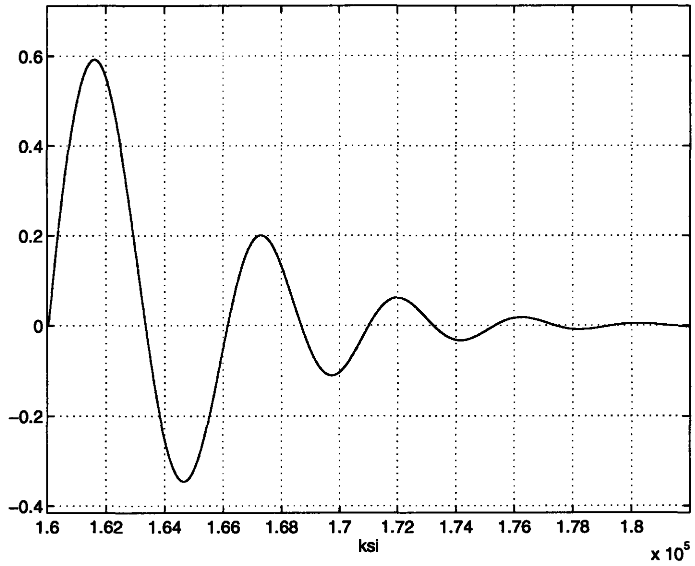


Figure 3-21: Sensitivity of 4th order longitudinal dynamics to M_α

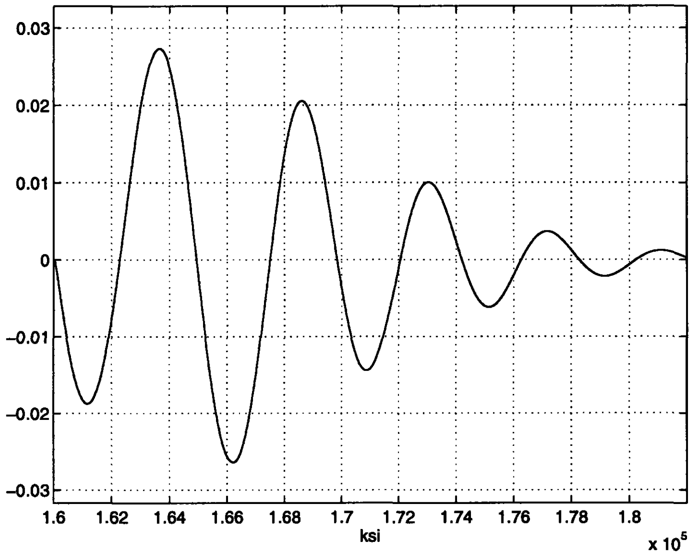


Figure 3-22: Sensitivity of 4th order longitudinal dynamics to M_q

3.6 The Use of ANN for System Identification

Analogous to the 2nd order case, the ANN will be used to design a system identification for 4th order GHAME vehicle longitudinal dynamics. The same technique of learning and test of robustness is used for the 4th order case. The results are shown in the following figures.

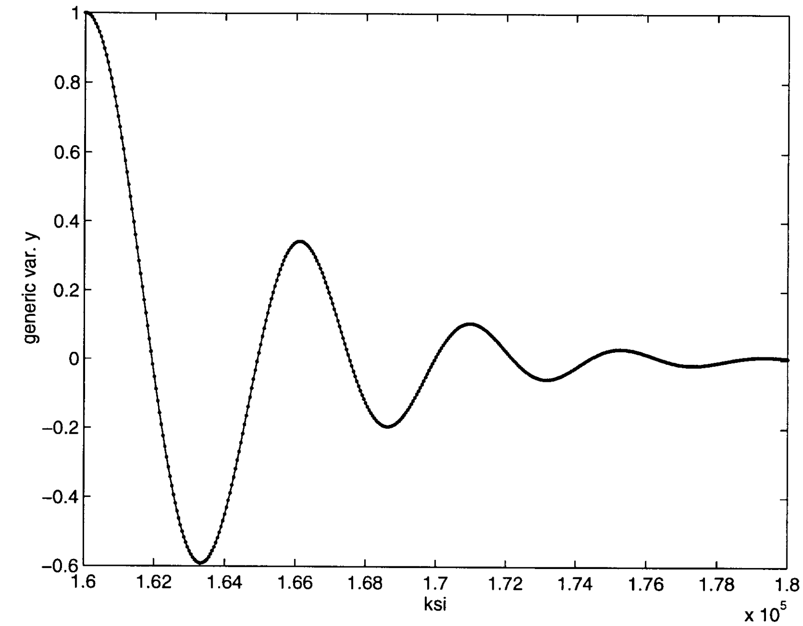


Figure 3-23: Neural Network Model for 4th order GHAME, training without white noise

——	≡	GHAME output
— · — · —	≡	Neural network model output

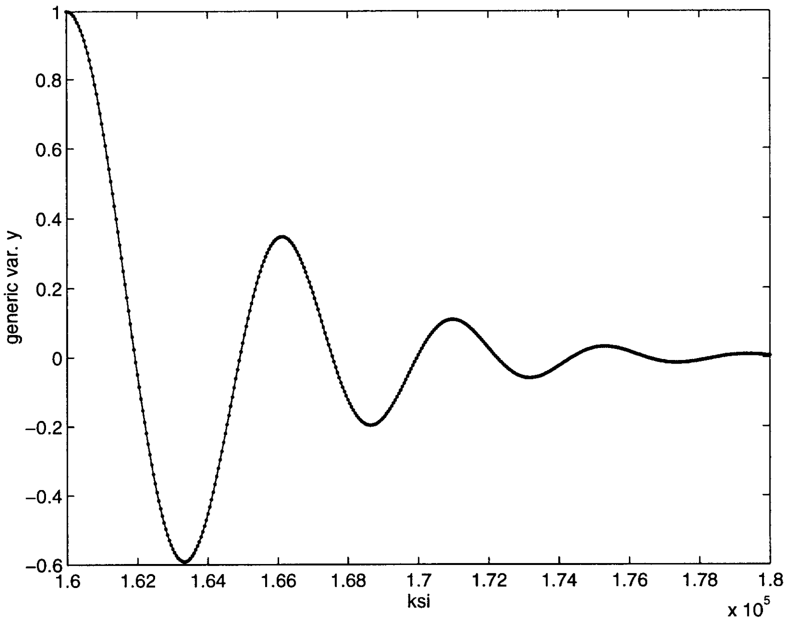


Figure 3-24: Neural Network Model for 4th order GHAME, training with white noise

——	≡	GHAME output
— · — · —	≡	Neural network model output

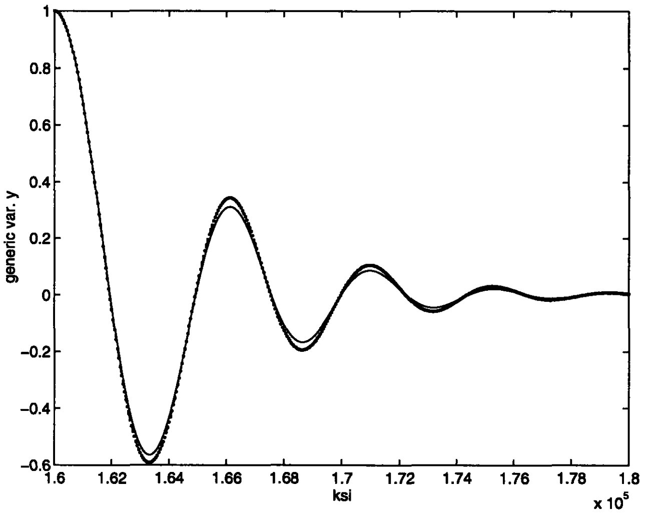


Figure 3-25: Neural Network Model for 4thorder GHAME with 10% increase in ω_1 (ξ)

—— $\equiv$ GHAME with 10% increase in ω_1 (ξ)

— · — · — $\equiv$ Neural network model output

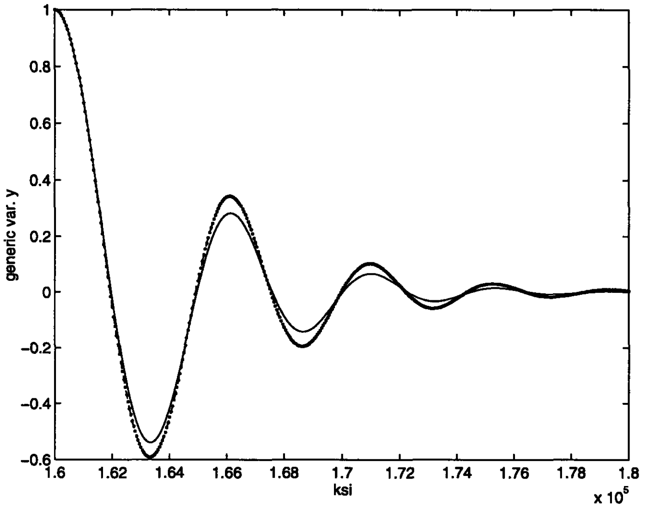


Figure 3-26: Neural Network Model for 4thorder GHAME with 20% increase in ω_1 (ξ)

—— $\equiv$ GHAME with 20% increase in ω_1 (ξ)

— · — · — $\equiv$ Neural network model output

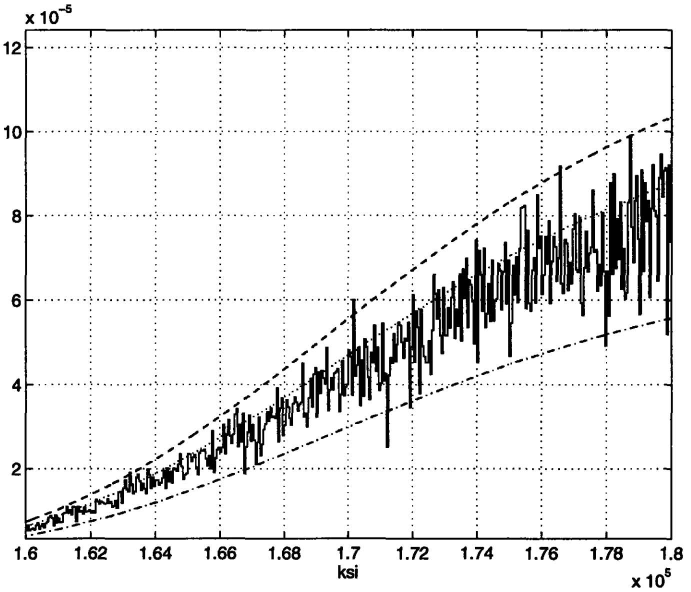


Figure 3-27: Damping Coefficient $\omega_1(\xi)$ with 20% uncertainty

——	≡	±20% uncertainty in $\omega_1(\xi)$
— · — · —	≡	20% decrease in $\omega_1(\xi)$
— — —	≡	20% increase in $\omega_1(\xi)$
....	≡	10% increase in $\omega_1(\xi)$

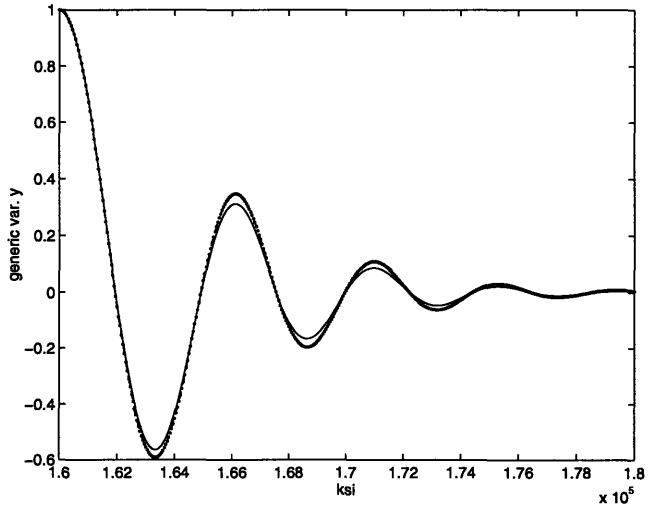


Figure 3-28: Neural Network Model (trained with white noise) for 4thorder GHAME with 10% increase in $\omega_1(\xi)$

——	≡	GHAME with 10% increase in $\omega_1(\xi)$
— · — · —	≡	Neural network model output

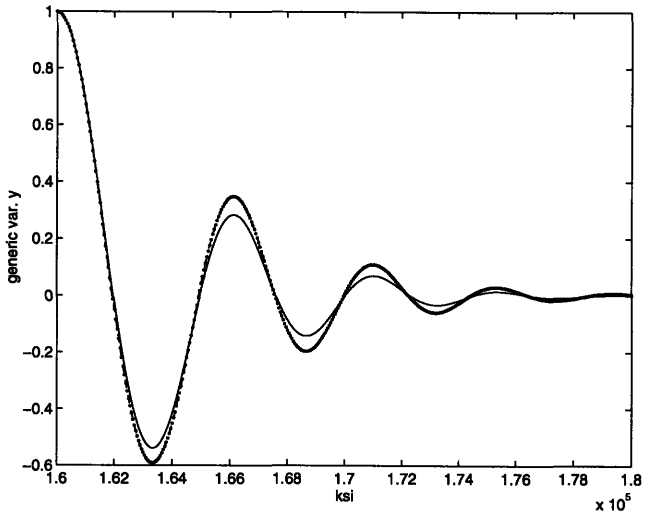


Figure 3-29: Neural Network Model (trained with white noise) for 4thorder GHAME with 20% increase in ω_1 (ξ)

—— ≡ GHAME with 20% increase in ω_1 (ξ)
— · — · — ≡ Neural network model output

Like the 2nd order case, again the results indicate only a slight improvement on the network performance in learning the dynamics of the system with 20% increase in ω_1 (ξ). This suggests that using a white noise during training is useful in giving the information about the presence of uncertainty to the network though it might no longer be satisfactory for the case when the system has larger scale of uncertainty.

Chapter 4

GHAME Vehicle Fourth Order Lateral-Directional Dynamics

The analysis of the 4th order lateral-directional dynamics of the GHAME vehicle is parallel with its longitudinal counterpart. The notion of stability from *Poincarè - Lyapunov* theorem is again used in conjunction with the linearization of the equations of motion.

4.1 Equations of Motion

Analogous to longitudinal dynamics, the differential equations of motion describing the lateral-directional dynamics of a flight vehicle are generally nonlinear and time-varying. Hence, to obtain the solution, the equations of motion are linearized about a nominal flight condition. The nonlinear equations of motion describing the lateral-directional dynamics of the GHAME vehicle are given by [29]

$$Y - m\dot{v} - mVr + mg\phi = 0 \tag{4.1}$$

$$L - I_{xz}\dot{r} + I_{xx}\ddot{\phi} = 0 \tag{4.2}$$

$$N - I_{zz}\dot{r} + I_{xz}\ddot{\phi} = 0 \tag{4.3}$$

where Y, L, and N are the aerodynamic side force, rolling moment, and yawing moment on the vehicle, respectively. Note that the variable v represents the component of velocity V perpendicular to the flight path, while r and ϕ are the yaw rate and roll angle of the vehicle. Eqns. 4.1 through 4.3 are obtained by "trimming" the vehicle at a nominal flight. At the trimmed condition, the aerodynamic and inertial forces as well as moments are balanced out. Eqn. 4.1 is attained by balancing the forces in the direction perpendicular to the flight path. While Eqns. 4.2 and 4.3 are the result of balancing the rolling and yawing moments of the vehicle respectively. These three equations are treated equally to their longitudinal counterparts. The three parameters v, r, and ϕ are represented as perturbations about a nominal value. The aerodynamic forces and moment are then expressed in the form of a Taylor series expansion as in the longitudinal case. The Taylor series representations and the perturbation forms of v, r, and ϕ are plugged into the nonlinear equations of motion given by Eqn. 4.1 – 4.3. The result is the standard state space form for the linearized lateral-directional equations of motion [29].

$$\begin{bmatrix} s - Y_v & V & -g \\ -L_v & -\frac{I_{xz}}{I_{xx}}s - L_r & s^2 - L_p s \\ -N_v & s - N_r & -\frac{I_{xz}}{I_{zz}}s^2 - N_p s \end{bmatrix} \begin{bmatrix} \Delta v \\ \Delta r \\ \Delta \phi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.4}$$

where variable s is as before the derivative operator $\frac{d}{dt}$. The parameters Y_v, L_v, L_r, L_p, N_v, N_r, and N_p appearing in the above state space equation are the lateral-directional stability derivatives for the flight vehicle. Their expressions have been elaborated in Table 2.1. From their expressions, it is obvious that those parameters for GHAME vehicle vary with time as the vehicle travels along the reentry trajectory. The influence of variation in V and ρ has been alluded in Chapter 3. The more subtle contributions come from the non-dimensional stability derivatives C_{y_β}, C_{l_β}, C_{l_r}, C_{l_p}, C_{n_β}, C_{n_r}, and C_{n_p}. They are all functions of the angle-of-attack α and Mach number M. The technique to express them in high order two variable polynomial has been detailed in Chapter 3. The results of the polynomial approximation of these parameters are shown in the succeeding figures. Again, having α and M as a function of ξ, we can express all the above non-dimensional stability derivatives in the variable ξ leading to having the parameters Y_v, L_v, L_r, L_p, N_v, N_r, and N_p as functions of ξ. These are all the requisite of the calculation of the time-varying coefficients appearing in the linearized differential equation to be detailed in the next paragraphs.

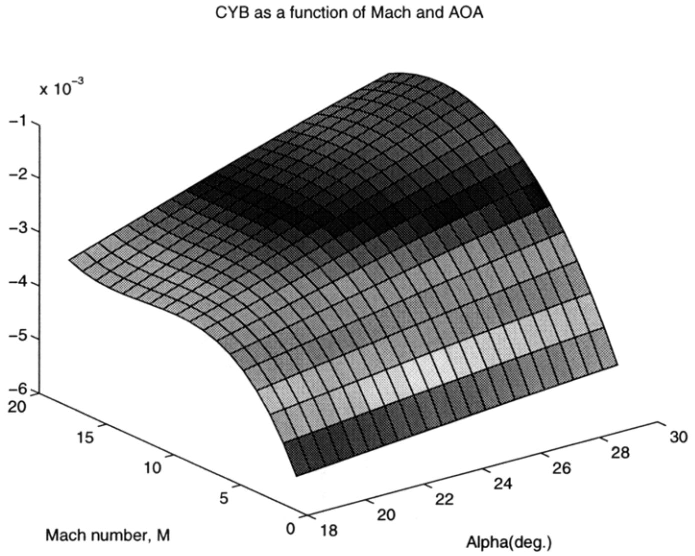


Figure 4-1: Aerodynamic coefficient C_{y_β} as a function of M and α

4.2 Lateral-Directional Stability Derivatives

Similar to the corresponding section for longitudinal dynamics, this section elaborates the stability derivatives for the lateral-directional motion. Stability derivative Y_v corresponds to the derivative C_{y_β} that is called side-force derivative. It gives the force that acts in the y direction (right) when the vehicle has a positive β or v. C_{y_β} is usually negative, and frequently small enough to be neglected entirely. The main contributions are those of the body and vertical tail, although the wing, and wing-body interference, may modify it significantly [24]. Fig. 4-8 (top) shows that GHAME vehicle has a small negative Y_v. The value increases in magnitude along the flight trajectory.

Stability derivative L_v corresponds to the derivative C_{l_β} called the dihedral effect. Its importance is related to the notion of *roll stiffness*, *i.e.* the flight vehicle tends to fly with wings level. The primary contribution to C_{l_β} is from the wing. The dihedral angle, aspect ratio and sweep angle of the wing all are important parameters. As shown in Fig. 4-8 (bottom), the value of L_v is small almost through out the entire flight trajectory. The sharp increment in the end is due to the increasing air density. Since GHAME vehicle wings do not have dihedral angle, the likely main contribution

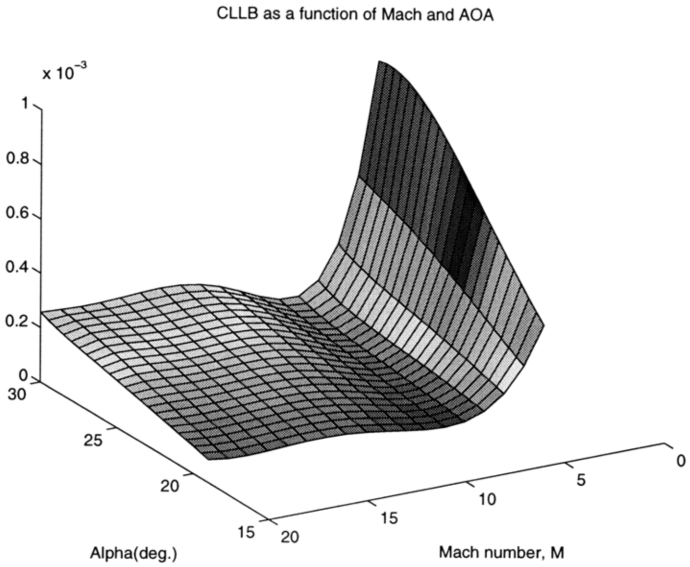


Figure 4-2: Aerodynamic coefficient C_{l_β} as a function of M and α

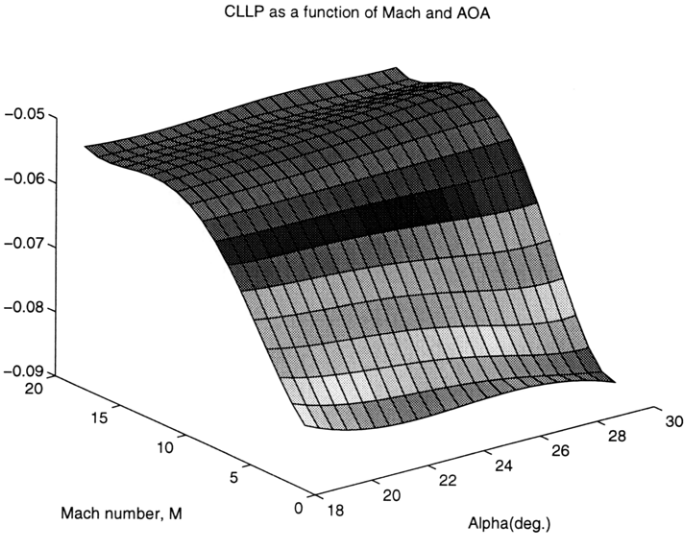


Figure 4-3: Aerodynamic coefficient C_{l_p} as a function of M and α

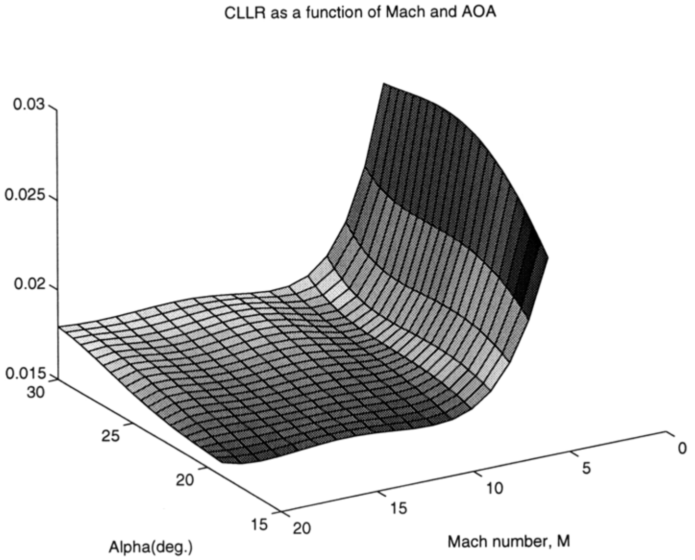


Figure 4-4: Aerodynamic coefficient C_{l_r} as a function of M and α

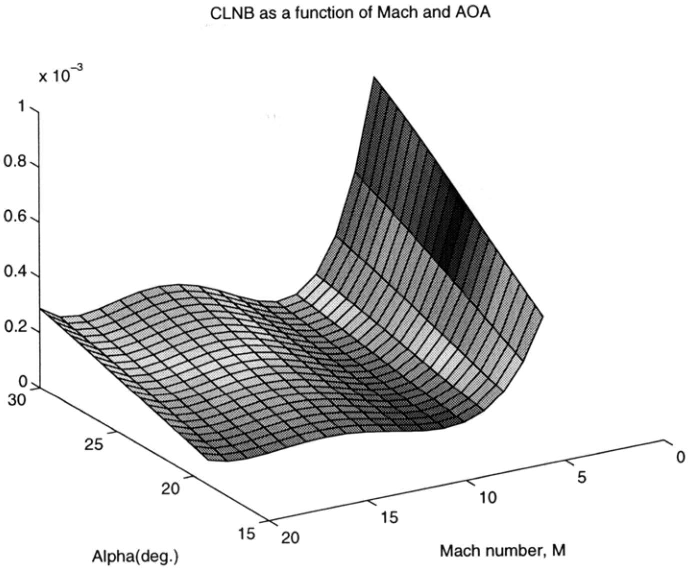


Figure 4-5: Aerodynamic coefficient C_{n_β} as a function of M and α

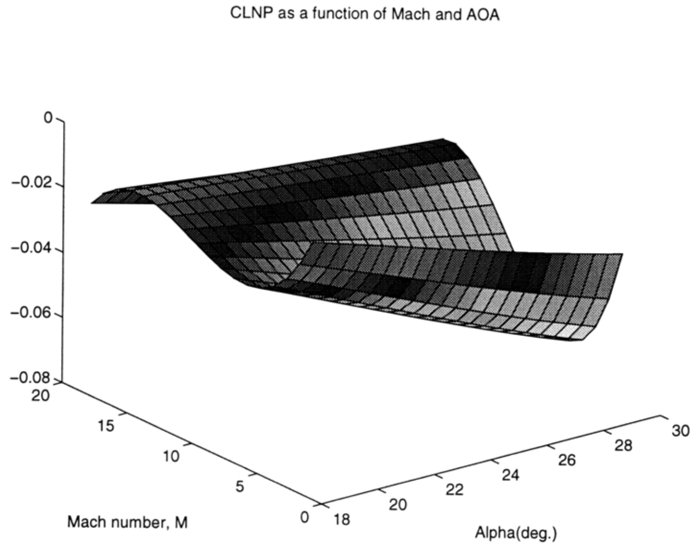


Figure 4-6: Aerodynamic coefficient C_{n_p} as a function of M and α

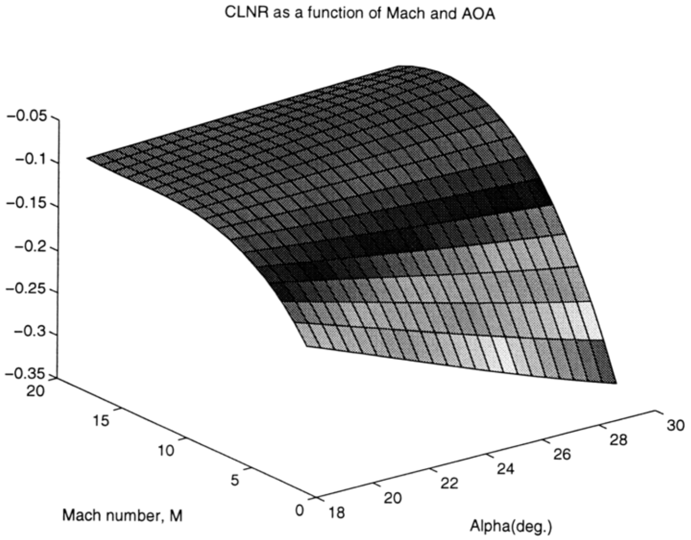


Figure 4-7: Aerodynamic coefficient C_{n_r} as a function of M and α

to the derivative is the sweep angle, wing-fuselage interference and vertical tail effect. Another important cross stability derivative is L_r which is related to derivative C_{l_r}. This describes the rolling moment due to yawing. The increase in lift on the left wing, and the decrease on the right wing combine to produce rolling moment proportional to the original lift coefficient C_L. The wing aspect ratio, taper ratio and sweepback all give paramount effects. If the vertical tail is large, it can give significant contribution. Fig. 4-9 (top) describes this derivative for GHAME vehicle. It is of positive large value and has tendency similar to variation of air density. The bottom figure shows the L_p stability derivative which corresponds to derivative C_{l_p} better known as *damping-in-roll* derivative. It expresses the resistance of the flight vehicle to rolling. For the conventional configuration, only the wing contribute significantly to this derivative. The value is usually negative. The GHAME vehicle exhibit no unusual properties with respect to this derivative. Along the reentry trajectory, L_p remains of negative value with tendency again resembles that of air density variation.

The directional stability derivative N_v corresponding to C_{n_β} is also of paramount important. This parameter represents the tendency of flight vehicles to yaw into the relative wind and largely caused by the change in oncoming airflow angle experienced by the vertical tail. This tendency gives the stabilizing effect called *weathercock* stability. In contrast with its analogous longitudinal stability C_{m_α}, the wing has little influence in most cases, and the center of gravity location is a weak parameter. Note that whether or not a positive value of C_{n_β} will produce lateral stability can only be determined by a full dynamic analysis [24]. The value of this derivative for GHAME vehicle (shown in Fig. 4-10) is unusually small enough although still positive.

The next directional derivative is N_r. The related derivative C_{n_r} is often referred to as *damping-in-yaw* derivative, and is always negative. The body does not have critical contribution to this derivative except when it is very large. The significant contributions come from wing and tail. The increases in both the profile and induced drag on the left wing and the decreases on the right wing give a negative yawing moment and hence resistance to the motion. The aspect ratio, taper ratio and sweepback again determine the magnitude of the effect. Fig. 4-10 (middle) shows the parameter N_r which is of negative value throughout the entire flight segment.

The directional stability that is coupled with lateral stability is N_p. This so called cross derivative represents the yawing moment produced by the rolling motion. The wing and tail both influence the derivative C_{n_p}. Fig. 4-10 (bottom) depicts the corresponding stability derivative of the GHAME vehicle along the reentry trajectory.

4.3 Solution to the Equation

The equation of motion described by Eqn. 4.4 is solved in similar manner as that of its longitudinal counterpart. First, it is assumed that the stability derivatives described in previous paragraphs are of constant value throughout the flight trajectory. Under such an assumption, the three independent variables of the lateral-directional equations of motion Δv, Δr, and $\Delta\phi$ have the same response. They are replaced by generic variable y whose dynamics are described by the determinant of the main matrix in Eqn. 4.4. Taking the determinant equal to zero we get the characteristic equation in variable s.

$$s^4 + \omega_3 s^3 + \omega_2 s^2 + \omega_1 s + \omega_0 = 0 \tag{4.5}$$

Noting that s is a differential operator $\frac{d}{dt}$, the equation can be rewritten as

$$\frac{d^4y}{dt^4} + \omega_3\left(t\right)\frac{d^3y}{dt^3} + \omega_2\left(t\right)\frac{d^2y}{dt^2} + \omega_1\left(t\right)\frac{dy}{dt} + \omega_0\left(t\right)y \tag{4.6}$$

For a steady flight condition ω_3, ω_2, ω_1 and ω_0 are given by:

$$\omega_3\left(t\right) = \left(aN_p + bL_r + N_r + L_p - abY_v + Y_v\right)/\left(ab-1\right) \tag{4.7}$$

$$\omega_2\left(t\right) = \left(L_rN_p - L_pN_r - aY_vN_p - bY_vL_r - Y_vN_r - Y_vL_p - bL_vV - N_vV\right)/\left(ab-1\right) \tag{4.8}$$

$$\omega_1\left(t\right) = \left(L_vg + Y_vL_pN_r - Y_vL_rN_p - L_vN_pV + N_vL_pV + agN_v\right)/\left(ab-1\right) \tag{4.9}$$

$$\omega_0\left(t\right) = g\left(N_vL_r - L_vN_r\right)/\left(ab-1\right) \tag{4.10}$$

and

$$a = \frac{I_{xz}}{I_{xx}} \qquad\qquad b = \frac{I_{xz}}{I_{zz}} \tag{4.11}$$

Fig. 4-11 and 4-12 show the coefficients of the GHAME vehicle lateral directional dynamic for slowly varying flight conditions. Solution to this time-varying system is obtained via numerical integration using Runge-Kutta method. The block diagram for the system is depicted in Fig. 4-13.

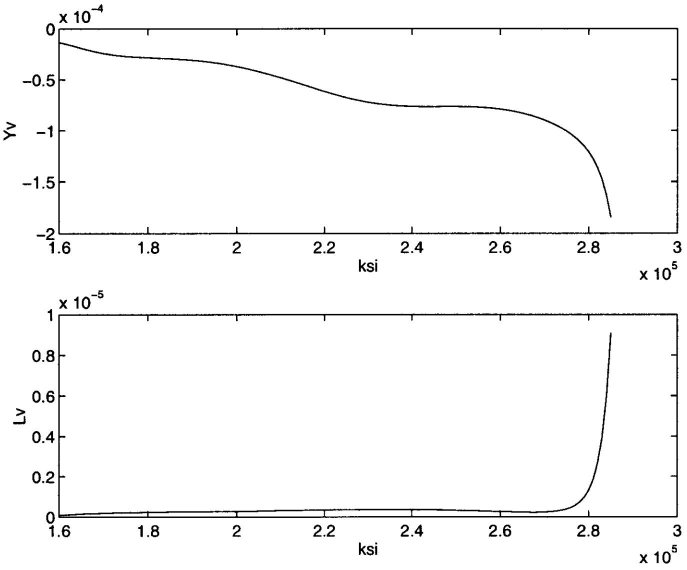


Figure 4-8: Stability derivative Y_v and L_v as a function of ξ

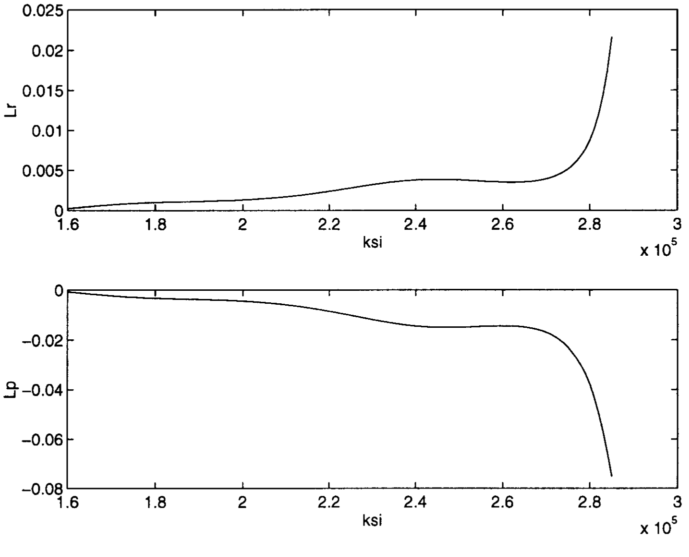


Figure 4-9: Stability derivative L_r and L_p as a function of ξ

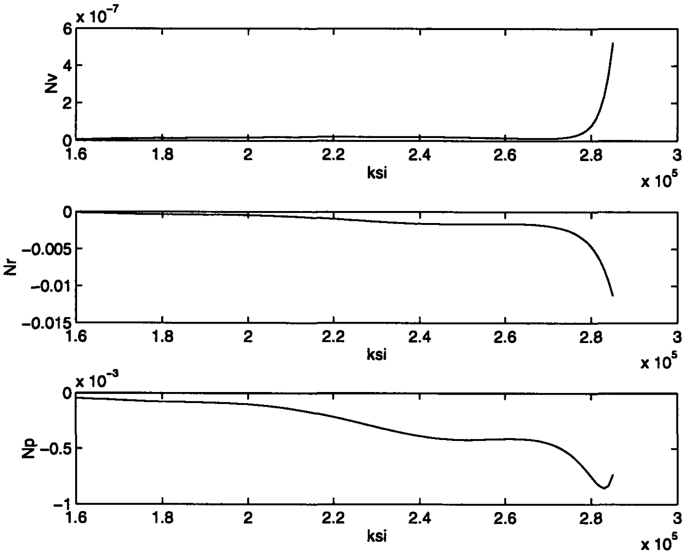


Figure 4-10: Stability derivative N_v, N_p and N_r as a function of ξ

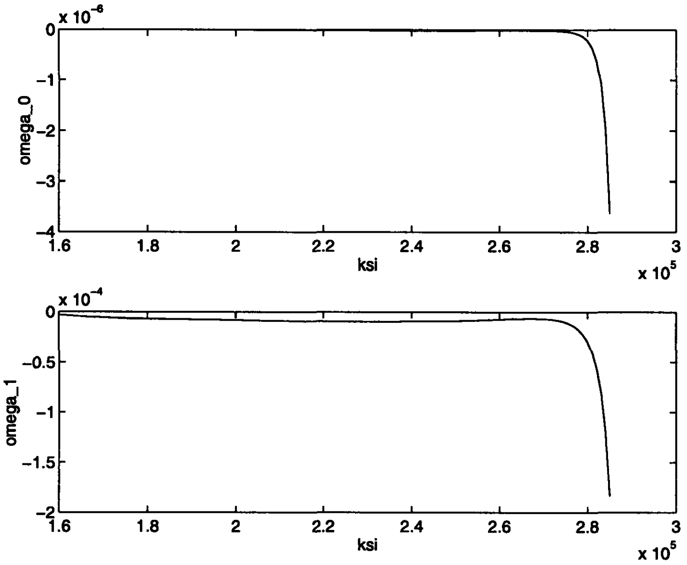


Figure 4-11: Time-varying coefficient ω_0 and ω_1 as a function of ξ

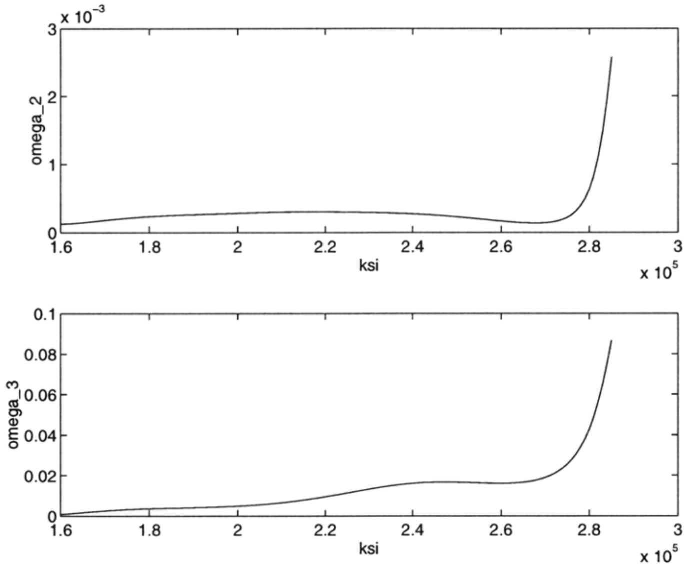


Figure 4-12: Time-varying coefficient ω_2 and ω_3 as a function of ξ

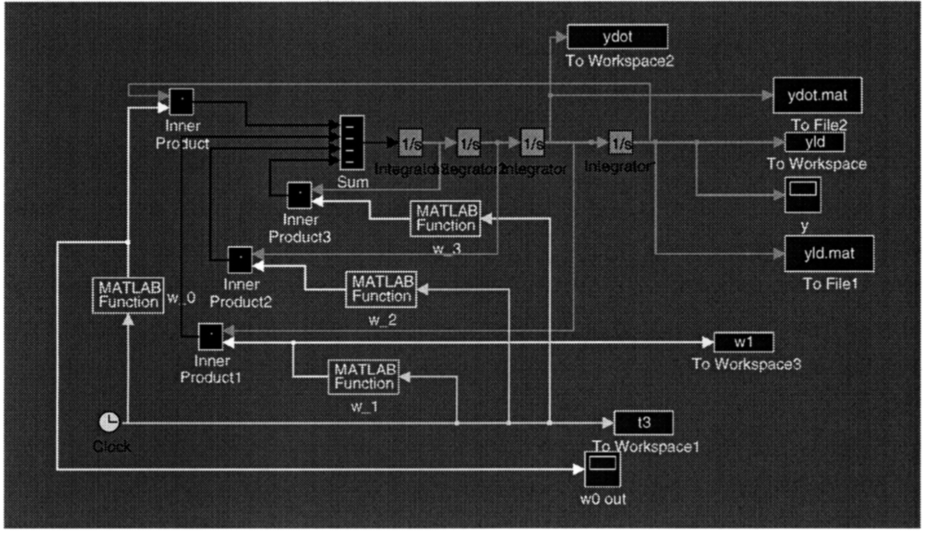


Figure 4-13: Simulation block diagram for 4-th order lateral-directional dynamics

4.4 Root Locus Analysis

Like its longitudinal counterpart,, the stability analysis is done by sketching the root-locus and looking at the corresponding time response behavior. Since the coefficients appearing in the differential equation of variable α is a function of ξ, the root locus of the system will also be a function of ξ.

4.4.1 Root Locus

The root locus of the fourth order lateral directional motion of GHAME are given in Fig. 4-14 and Fig. 4-15. At the early phase of the trajectory, there is a pair of conjugate roots located near the origin. These conjugate roots represent the dutch-roll mode which is usually characterized by high frequency lightly damped oscillatory motion. Two other roots are located in positive real axis; one is around 0.01 and the other very close to zero. As the vehicle transverses the prescribed flight trajectory, the conjugate roots move to the left half plane. The frequency of the dutch-roll-like mode increases from around 0.015 up to 0.05, while the damping increases by order of 10 from around -0.005 to -0.05. The real root which is close to zero moves further to the left which gives substantial increment in the damping of this mode. However another real root moves from 0.01 to about 0.03 which causes the system to be unstable. In general the lateral directional dynamics of the vehicle is unstable.

4.4.2 Time Response Analysis

The time response of the 4th order lateral-directional dynamics for different initial conditions are given by Fig. 4-16 – Fig. 4-23. As mentioned earlier, the Adam-Gear integration method is used in generating the time response of the system. The scheme of the simulation is depicted in Fig. 4-13. In agreement with the root locus analysis, the results show the unstable behavior of the system.

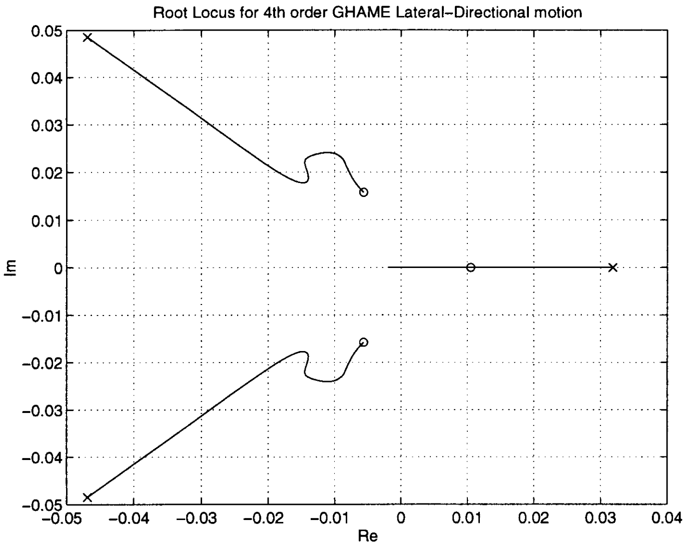


Figure 4-14: Root locus variation for 4-th order lateral-directional dynamics: plot of dutch-roll-like roots and one real root

○	≡	beginning of the trajectory $\xi = 1.6.10^5$
×	≡	end of the trajectory $\xi = 2.85.10^5$

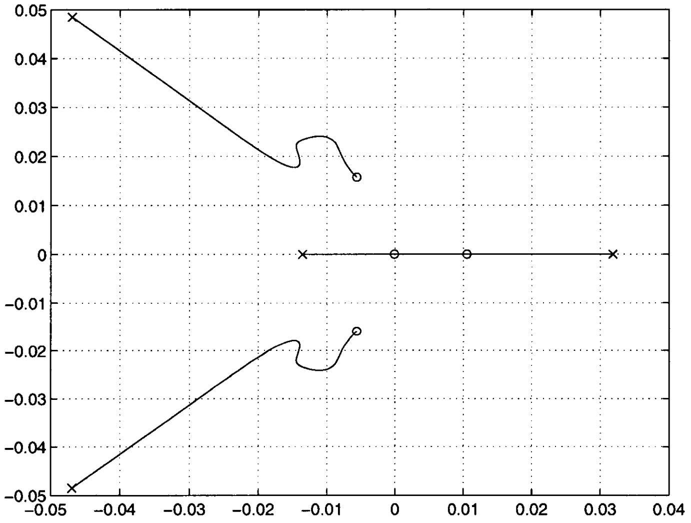


Figure 4-15: Root locus variation for 4-th order lateral-directional dynamics: complete roots

○	≡	beginning of the trajectory $\xi = 1.6.10^5$
×	≡	end of the trajectory $\xi = 2.85.10^5$

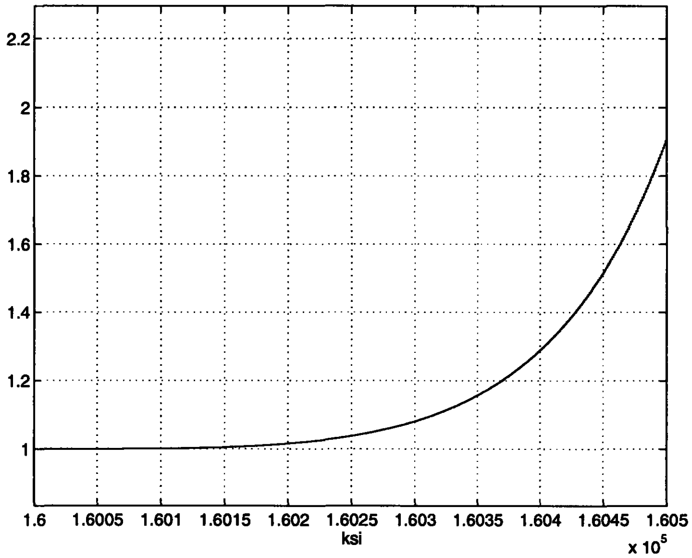


Figure 4-16: Time response of y for 4th order lateral-directional dynamics

$$\begin{aligned} y\,(1.6.10^5) &\equiv 1 \\ y'\,(1.6.10^5),\ y''\,(1.6.10^5),\ y'''\,(1.6.10^5) &\equiv 0 \end{aligned}$$

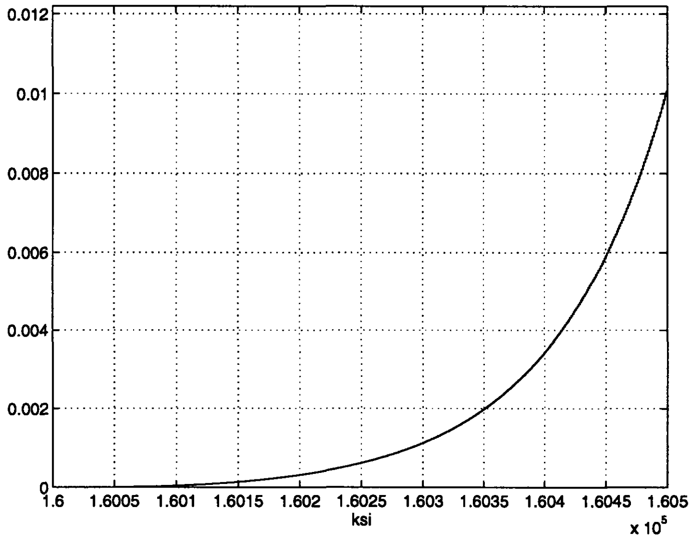


Figure 4-17: Time response of y' for 4th order lateral-directional dynamics

$$\begin{aligned} y\,(1.6.10^5) &\equiv 1 \\ y'\,(1.6.10^5),\ y''\,(1.6.10^5),\ y'''\,(1.6.10^5) &\equiv 0 \end{aligned}$$

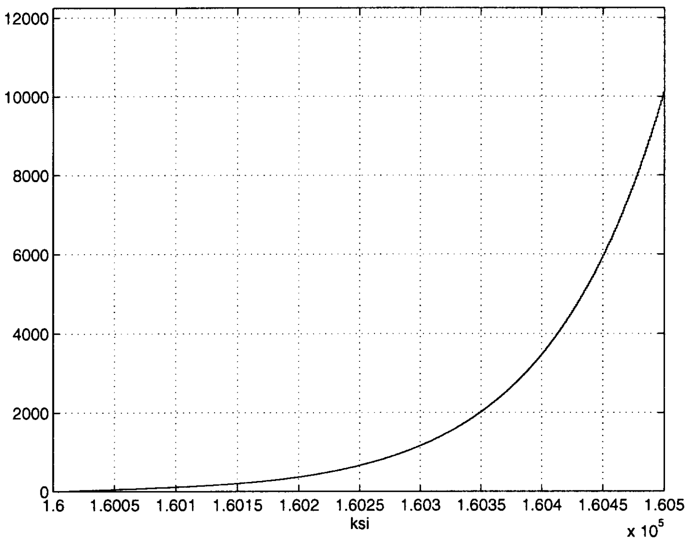


Figure 4-18: Time response of y for 4th order lateral-directional dynamics

$$\begin{aligned} &y'(1.6.10^5) &&\equiv 1 \\ &y(1.6.10^5),\ y''(1.6.10^5),\ y'''(1.6.10^5) &&\equiv 0 \end{aligned}$$

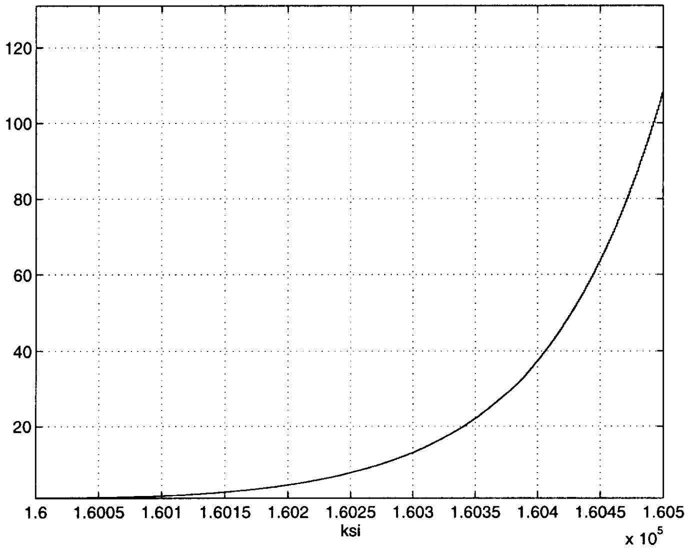


Figure 4-19: Time response of y' for 4th order lateral-directional dynamics

$$\begin{aligned} &y'(1.6.10^5) &&\equiv 1 \\ &y(1.6.10^5),\ y''(1.6.10^5),\ y'''(1.6.10^5) &&\equiv 0 \end{aligned}$$

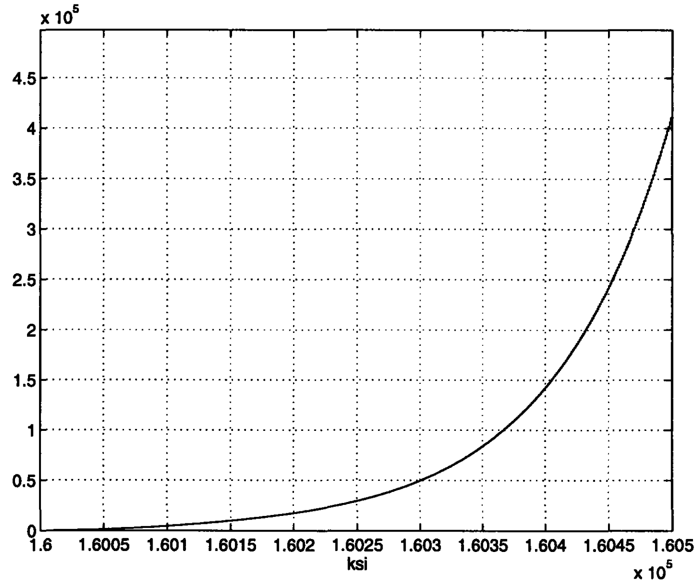


Figure 4-20: Time response of y for 4th order lateral-directional dynamics

$$\begin{aligned} y''(1.6.10^5) &\equiv 1 \\ y'(1.6.10^5),\ y'(1.6.10^5),\ y'''(1.6.10^5) &\equiv 0 \end{aligned}$$

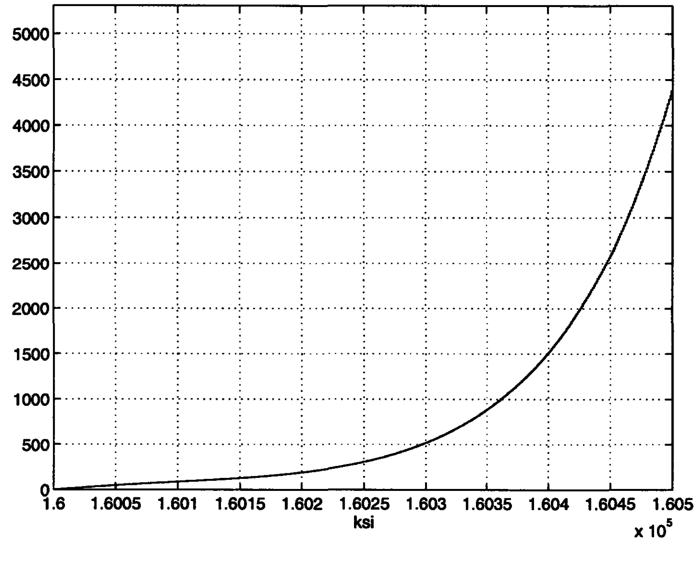


Figure 4-21: Time response of y' for 4th order lateral-directional dynamics

$$\begin{aligned} y''(1.6.10^5) &\equiv 1 \\ y'(1.6.10^5),\ y'(1.6.10^5),\ y'''(1.6.10^5) &\equiv 0 \end{aligned}$$

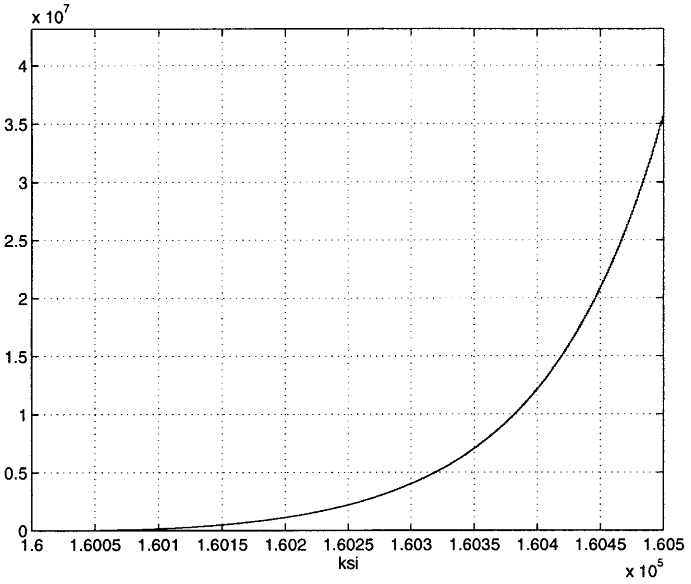


Figure 4-22: Time response of y for 4th order lateral-directional dynamics

$$\begin{aligned} y'''(1.6.10^5) &\equiv 1 \\ y(1.6.10^5),\ y'(1.6.10^5),\ y''(1.6.10^5) &\equiv 0 \end{aligned}$$

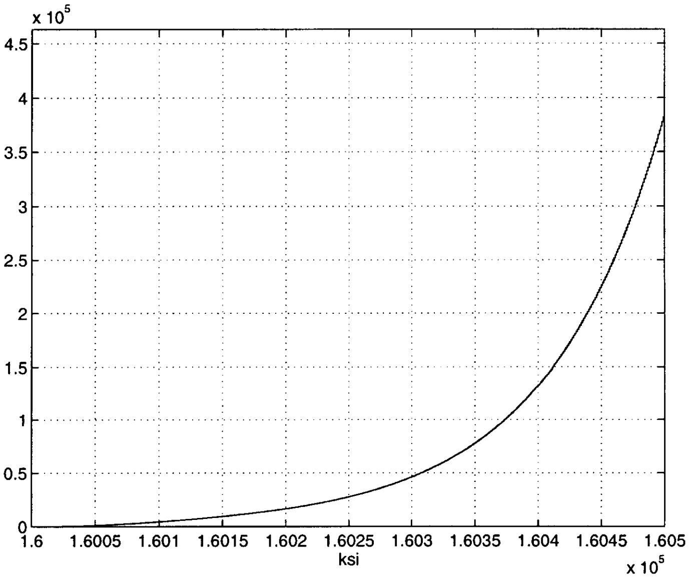


Figure 4-23: Time response of y' for 4th order lateral-directional dynamics

$$\begin{aligned} y'''(1.6.10^5) &\equiv 1 \\ y(1.6.10^5),\ y'(1.6.10^5),\ y''(1.6.10^5) &\equiv 0 \end{aligned}$$

Chapter 5

Vertical Take-Off and Landing (VTOL) Aircraft Dynamics

We now consider the longitudinal dynamics of a VTOL aircraft during the hover-forward flight transition. The model taken into account is the dynamics of XC-142 aircraft. The model, approach and solutions are based on Ramnath [30]. At hover the vehicle behaves like a helicopter and is dynamically unstable. At the forward flight condition it is dynamically stable. Linearization of the equations of motion about the transition trajectory leads to the time-varying coefficients in the differential equations. For our purpose, we just consider the dependence of the stability derivatives on V (the main one). Based on the work of Ramnath [30] at Princeton University, they are given in Table 5.1

The V is a function of time and the following two variation are studied:

$$V\left(t\right)=\left\{\begin{array}{l}\frac{150t}{10+t}ft/sec\\ \frac{150t}{20+t}ft/sec\end{array}\right.$$

5.1 Third Order Longitudinal Dynamics

5.1.1 Equations of Motion

As shown by Ramnath [30], the decoupled linearized perturbation equations of motion of a VTOL during the transition valid in the vicinity of hover are given by:

Table 5.1: VTOL Stability Derivatives

Stability Derivative	Expression
$-X_u$	0.2
$-Z_w$	$0.1 + 0.0004V$
$-Z_u$	$\frac{0.25V}{10+V}$
$-M_q$	$0.1 + 0.0034V$
$-M_u$	$0.015\left(-1 + \frac{V}{150}\right)$
$-M_w$	$\begin{cases} -0.02 + 0.00025\frac{V}{150}_2 \\ 0.005 + 0.015\left(\frac{V}{150}\right)^2 \end{cases}$

$$\dddot{u} - (X_u + M_q)\,\ddot{u} + \left(X_u M_q - 2\dot{X}_u\right)\dot{u} + \left(gM_u + \dot{X}_u M_q - \ddot{X}_u\right)u = 0 \tag{5.1}$$

$$\dddot{\theta} - \left(X_u + M_q + \frac{\dot{M}_u}{M_u}\right)\ddot{\theta} + \left(X_u M_q - \dot{M}_q + M_q\frac{\dot{M}_u}{M_u}\right)\dot{\theta} + gM_u\theta = 0 \tag{5.2}$$

where u and θ are perturbations in forward velocity and pitch attitude.

Note that since we are dealing with linear time-varying (LTV) system, the equation describing the two state variables have different forms. This may lead to a different stability behavior i.e. one of the states might be stable while the other might be unstable. This phenomenon can not happen in the LTI system. In the LTI case, the above equations will be the same.

Since the V is a function of time and the stability derivatives are functions of V, they can now be written as functions of time and depend on the transition trajectory. For a specific prescribed trajectory the decoupled equations for the perturbation variables u and θ are:

$$(1 + 0.1t)\,\dddot{u} + (0.3 + 0.081t)\ddot{u} + (0.02 + 0.01222t)\dot{u} + 0.48u = 0 \tag{5.3}$$

$$\begin{aligned}(1 + 0.1t)^2\,\dddot{\theta} + (0.4 + 0.081t)(1 + 0.1t)\ddot{\theta} + (0.081 + 0.01833t \\ + 0.00122t^2)\dot{\theta} + 0.48(1 + 0.1t)\theta = 0\end{aligned} \tag{5.4}$$

5.1.2 Stability Analysis

The root locus method is again used to observe the stability behavior of the system. Figure 5-1 and 5-2 show the variation of the root locus along the flight trajectory starting from $t = 0$ to $t = 150$ sec. for the variable u and θ. In general they consist of a pair of conjugate roots and one real root. In the beginning of the trajectory, the conjugate roots lie within the right half plane making the system unstable. As the VTOL transverses along its flight path, these roots move to the left and cross the imaginary axis around $t = 60$ sec. From then on, all the roots are in the left half plane and so the system is stable. Thus, in that segment of trajectory, the system experiences a change in the stability behavior that happens around $t = 60$ sec.

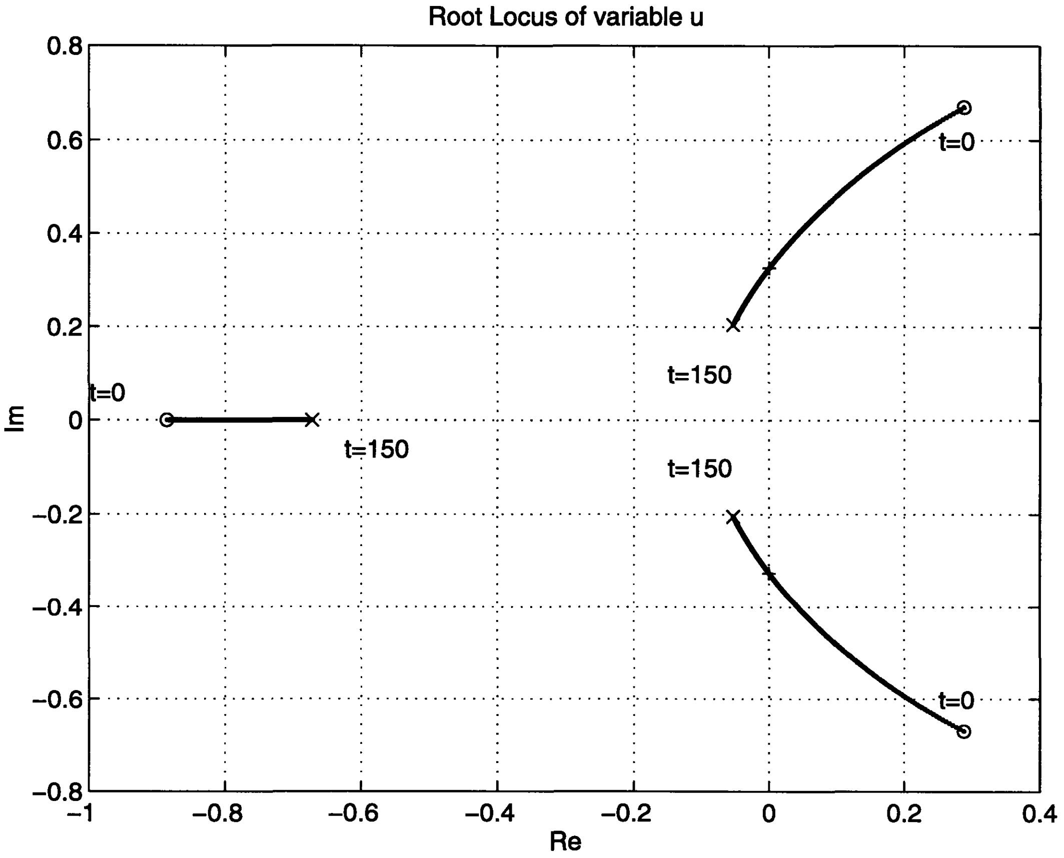


Figure 5-1: Root locus variation for 3rd order VTOL longitudinal dynamics: var. u
○ ≡ beginning of the trajectory $t = 0$
× ≡ end of the trajectory $t = 150$ sec.

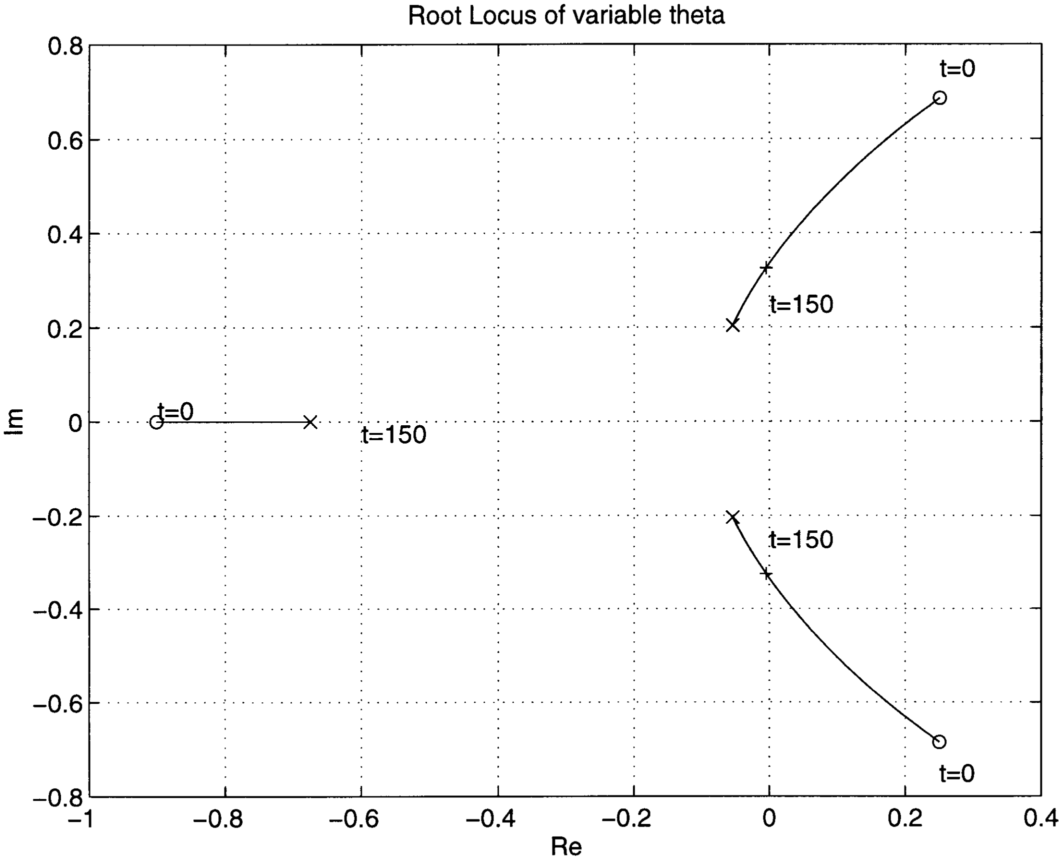


Figure 5-2: Root locus variation for 3rd order VTOL longitudinal dynamics: var. θ

$\circ \equiv$ beginning of the trajectory $t = 0$

$\times \equiv$ end of the trajectory $t = 150$ sec.

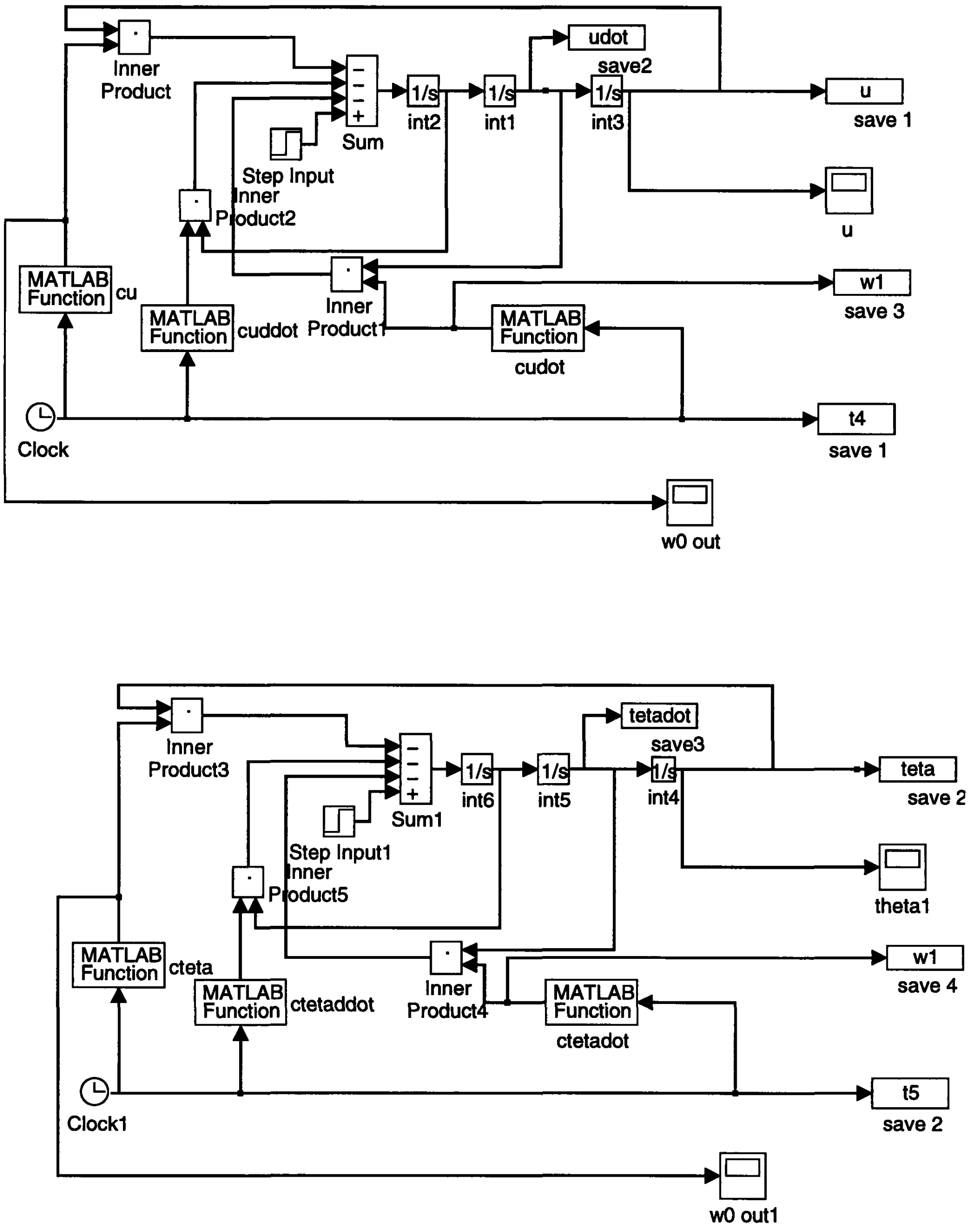


Figure 5-3: Simulation block diagram for perturbation variable $u(t)$ and $\theta(t)$ dynamics of VTOL aircraft

To get the time response behavior, the two decoupled perturbation equation are then integrated using Runge Kutta fourth order method. SIMULINK block diagrams for the simulation are shown in Fig(5-3). In finding the numerical solution, the perturbation equation is rewritten in the following form:

$$u^{(3)} + c_2(t)u^{(2)} + c_1(t)u^{(1)} + c_0(t)u = 0$$

The time responses for different initial conditions for the perturbation variable u are shown from Fig. 5-4 to Fig. 5-7 and from Fig. 5-8 to Fig. 5-11 for θ.

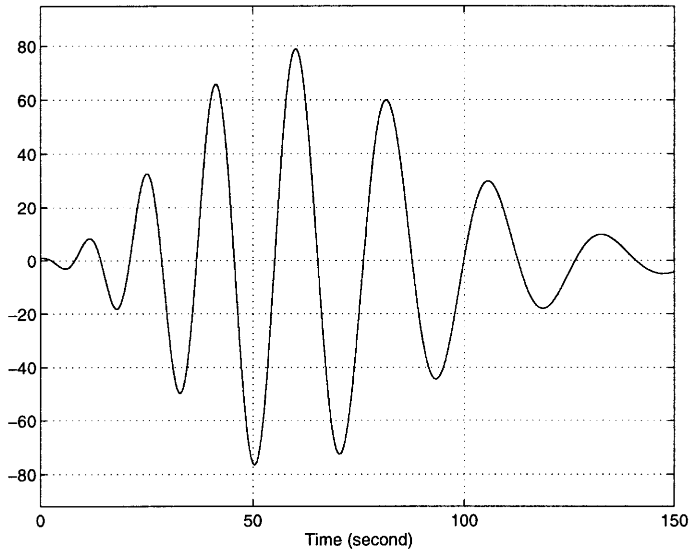


Figure 5-4: Time response of u for 3rd order VTOL longitudinal dynamics model

$$\begin{aligned} u\,(0) &\equiv 1 \\ \dot{u}\,(0),\, \ddot{u}\,(0) &\equiv 0 \end{aligned}$$

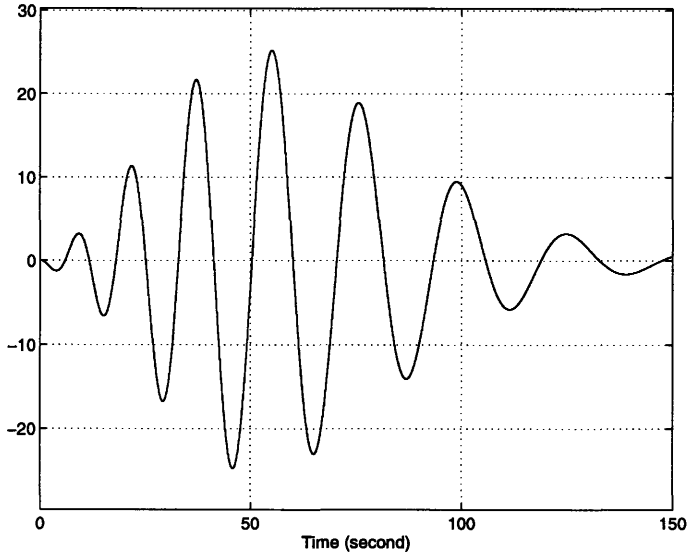


Figure 5-5: Time response of $\dot{u}$ for 3rd order VTOL longitudinal dynamics model

$$\begin{array}{lcl} u\,(0) & \equiv & 1 \\ \dot{u}\,(0),\, \ddot{u}\,(0) & \equiv & 0 \end{array}$$

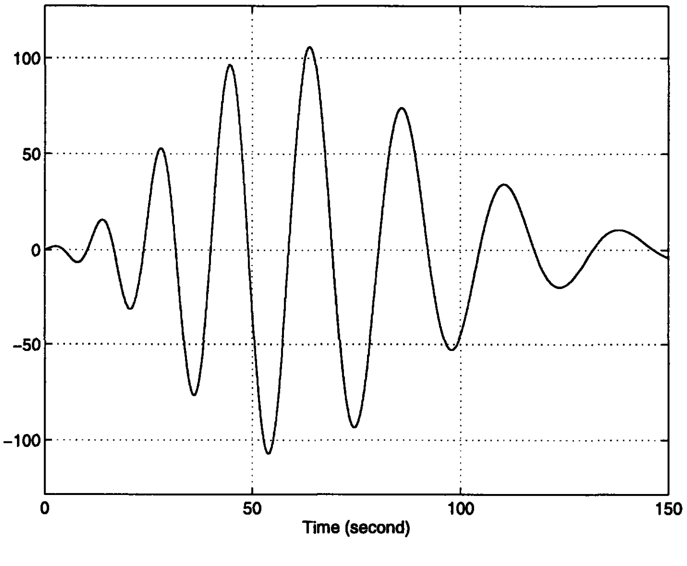


Figure 5-6: Time response of u for 3rd order VTOL longitudinal dynamics model

$$\begin{array}{lcl} \dot{u}\,(0) & \equiv & 1 \\ u\,(0),\, \ddot{u}\,(0) & \equiv & 0 \end{array}$$

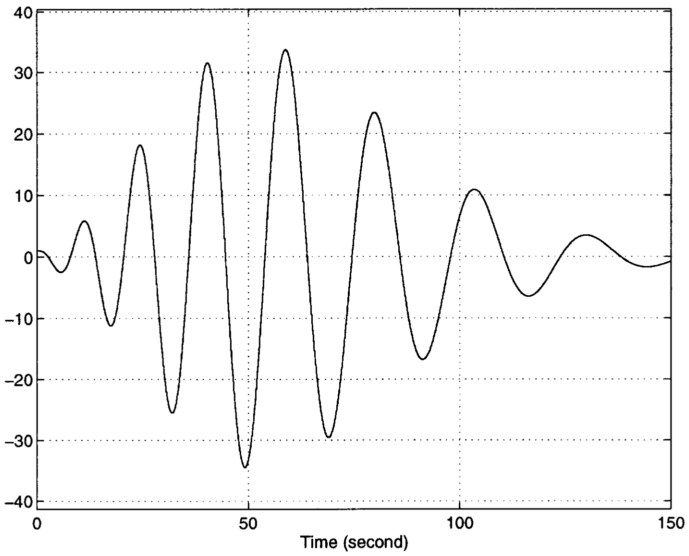


Figure 5-7: Time response of $\dot{u}$ for 3rd order VTOL longitudinal dynamics model

$$\begin{aligned} \dot{u}\,(0) &\equiv 1 \\ u\,(0),\, \ddot{u}\,(0) &\equiv 0 \end{aligned}$$

From Fig. 5-4 to 5-7, we can observe that from the beginning of the trajectory until reaching $t = 60$ sec. the perturbation variable u is divergent. This correspond to the roots that are located in the right half plane. Afterwards, the roots move to the left half plane and this is reflected in the convergent time response. Similar behavior is also shown by perturbation variable θ. The corresponding time response behavior for θ is displayed in the following figures.

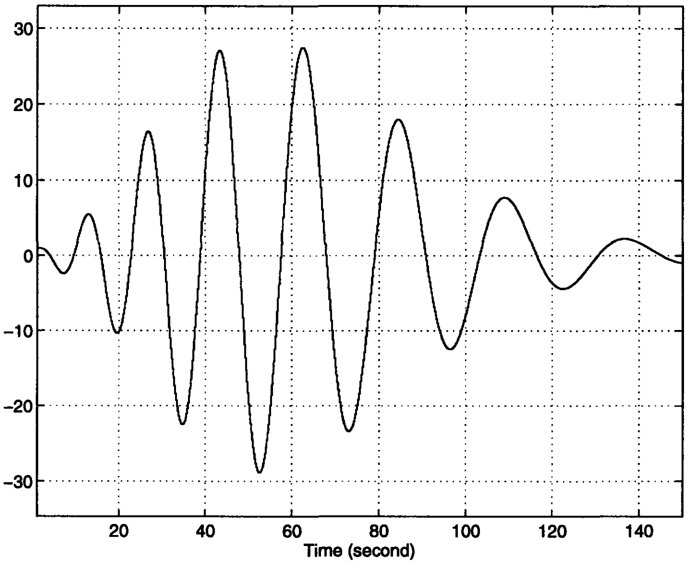


Figure 5-8: Time response of θ for 3rd order VTOL longitudinal dynamics model

$$\begin{aligned} \theta\,(0) &\equiv 1 \\ \dot{\theta}\,(0), \ddot{\theta}\,(0) &\equiv 0 \end{aligned}$$

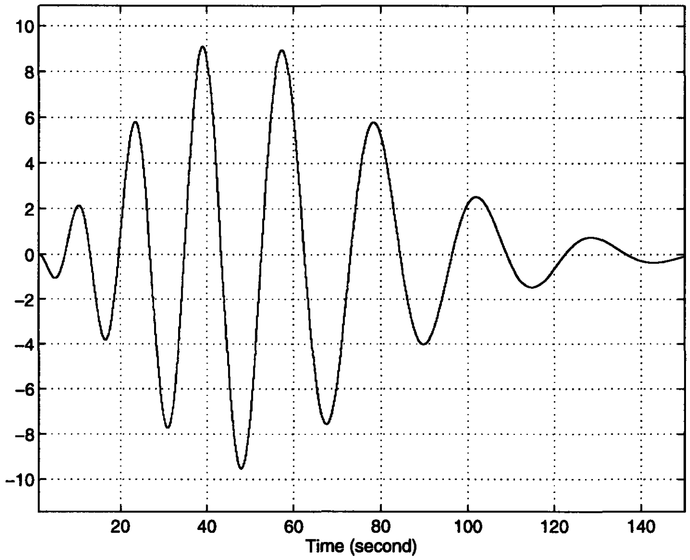


Figure 5-9: Time response of $\dot{\theta}$ for 3rd order VTOL longitudinal dynamics model

$$\begin{aligned} \theta\,(0) &\equiv 1 \\ \dot{\theta}\,(0), \ddot{\theta}\,(0) &\equiv 0 \end{aligned}$$

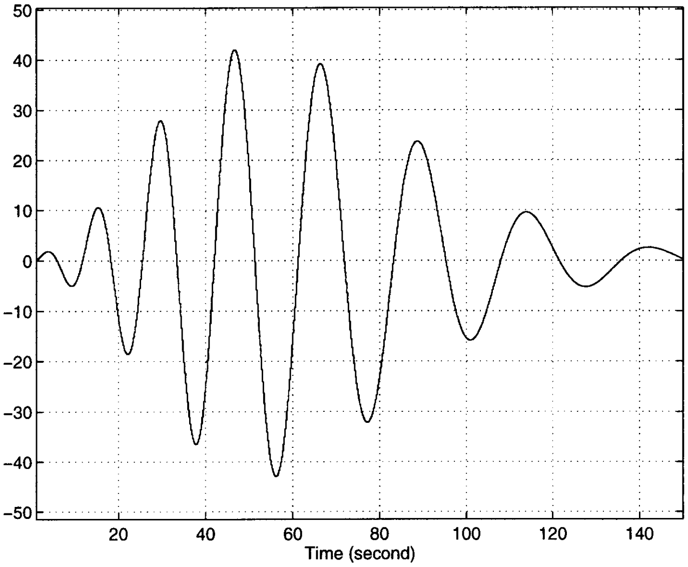


Figure 5-10: Time response of θ for 3rd order VTOL longitudinal dynamics model

$$\begin{aligned} \dot{\theta}(0) &\equiv 1 \\ \theta(0), \ddot{\theta}(0) &\equiv 0 \end{aligned}$$

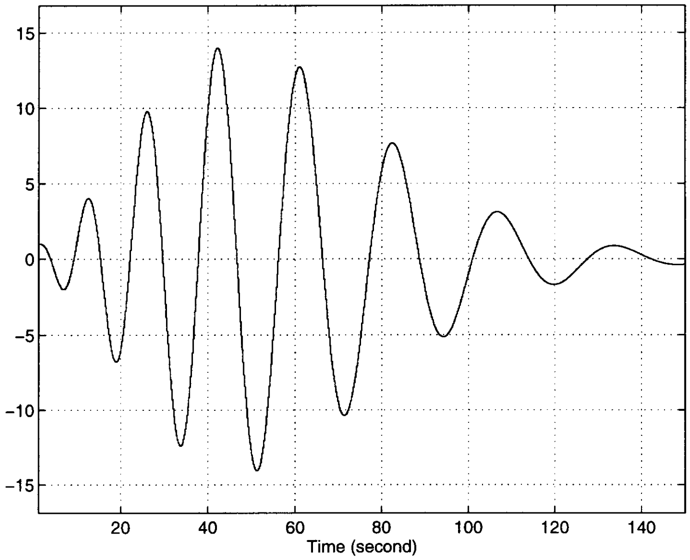


Figure 5-11: Time response of $\dot{\theta}$ for 3rd order VTOL longitudinal dynamics model

$$\begin{aligned} \dot{\theta}(0) &\equiv 1 \\ \theta(0), \ddot{\theta}(0) &\equiv 0 \end{aligned}$$

5.1.3 Neural Network for System Identification

The ANN is used to design a system identification for the 3rd order model of VTOL aircraft longitudinal dynamics. Only the variable u behavior is considered since θ practically exhibits a similar behavior. The same technique of learning and test of robustness are used as for the GHAME vehicle case. The results are shown in the following figures.

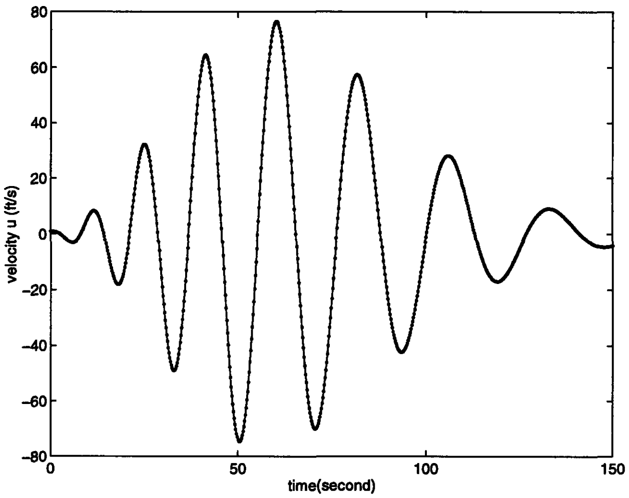


Figure 5-12: Neural Network Model for 3rd order VTOL, training without white noise
—— ≡ VTOL output
– · – · – ≡ Neural network model output

Fig. 5-12 shows the results of the training of neural network without adding any white noise. The trained neural network is tested with the original system. The figure shows the neural network accurately learns the dynamic of the system. The trained network is then tested for the system with the presence of uncertainty. Fig. 5-13 and Fig. 5-14 show the results of the test for 10% and 20% uncertainty in the system respectively.

The second training of the neural network includes white noise. The variation of $c_2(t)$ along with its perturbed value is shown in Fig. 5-15. The result of the second training and the test for the trained network are given from Fig. 5-16 to Fig. 5-19. The results indicate that using the white noise during training can improve the performance of the network in anticipating the presence of uncertainty.

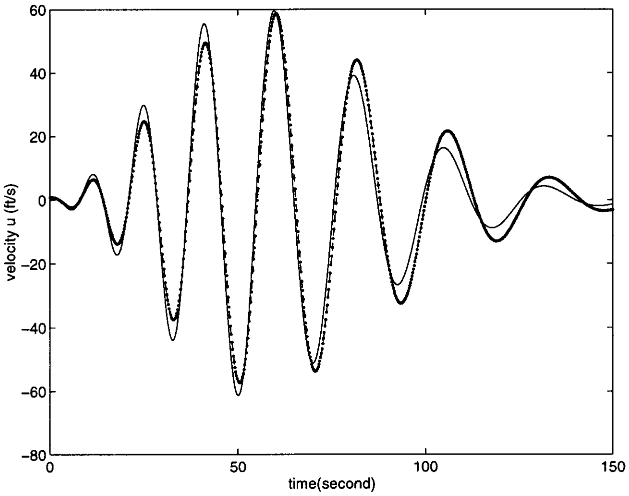


Figure 5-13: Neural Network Model for 3rd order VTOL with 10% increase in $c_2\,(t)$

—— ≡ VTOL with 10% increase in $c_2\,(t)$

— · — · — ≡ Neural network model output

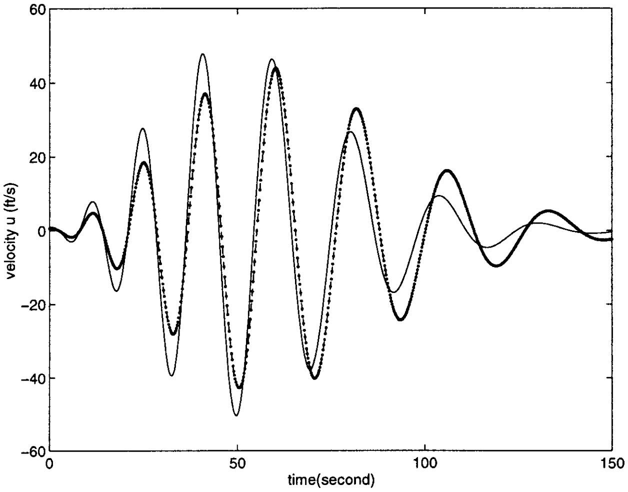


Figure 5-14: Neural Network Model for 3rd order VTOL with 20% increase in $c_2\,(t)$

—— ≡ VTOL with 20% increase in $c_2\,(t)$

— · — · — ≡ Neural network model output

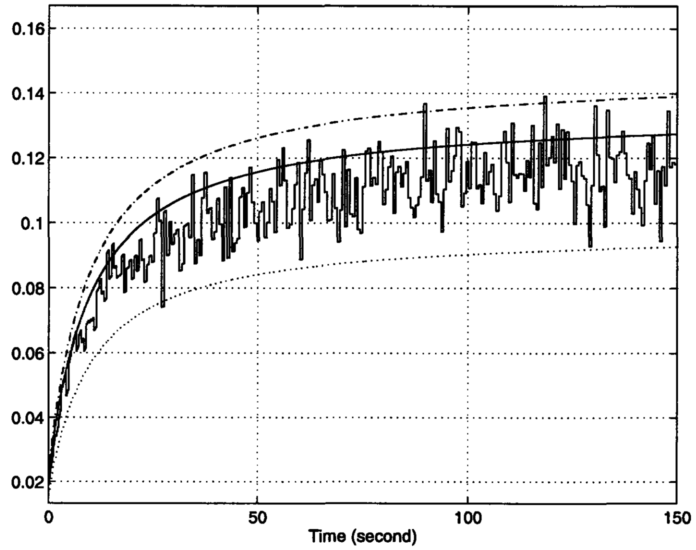


Figure 5-15: Damping Coefficient $c_2\,(t)$ with 20% uncertainty

——	≡	±20% uncertainty in $c_2\,(t)$
— · — · —	≡	20% decrease in $c_2\,(t)$
— — —	≡	20% increase in $c_2\,(t)$
....	≡	10% increase in $c_2\,(t)$

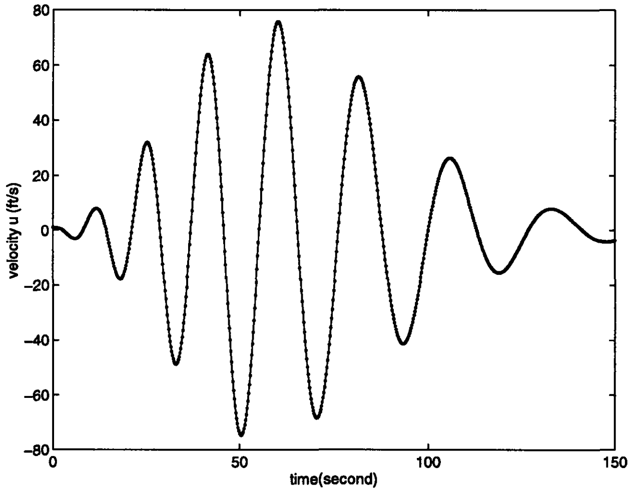


Figure 5-16: Neural Network Model for 3rd order VTOL, training with white noise

——	≡	VTOL output
— · — · —	≡	Neural network model output

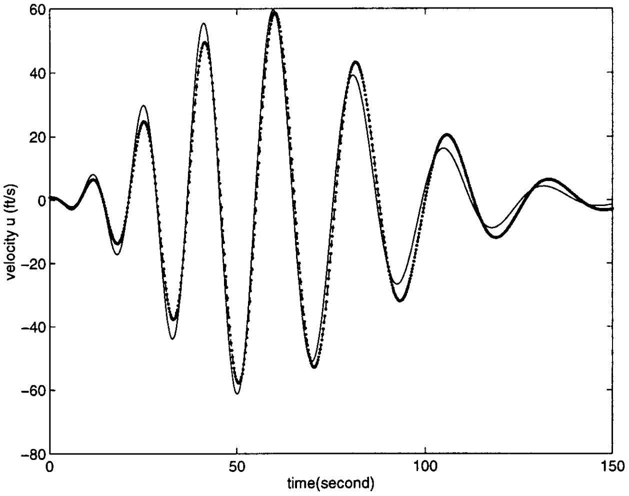


Figure 5-17: Neural Network Model for 3rd order VTOL with 10% increase in $c_2(t)$ (training with white noise)

——	≡	VTOL with 10% increase in $c_2(t)$
— · — · —	≡	Neural network model output

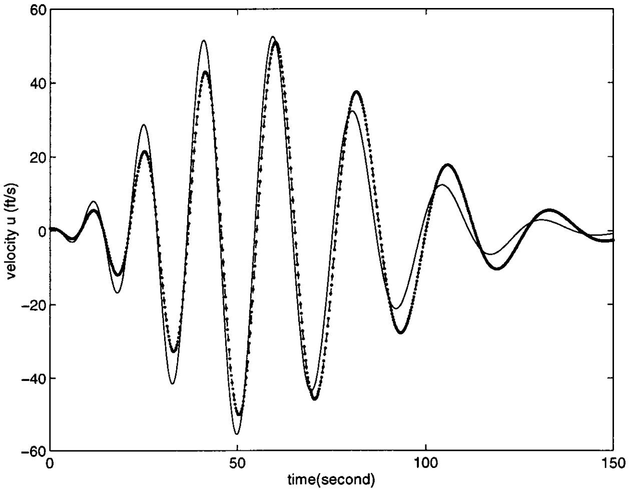


Figure 5-18: Neural Network Model for 3rd order VTOL with 15% increase in $c_2(t)$ (training with white noise)

——	≡	VTOL with 15% increase in $c_2(t)$
— · — · —	≡	Neural network model output

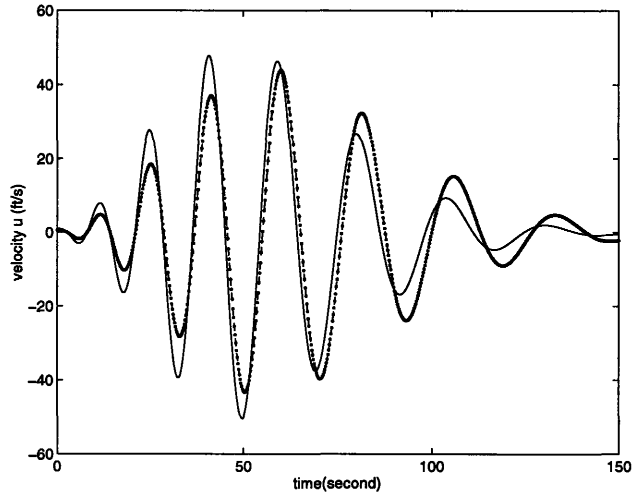


Figure 5-19: Neural Network Model for 3rd order VTOL with 20% increase in $c_2(t)$ (training with white noise)

—— ≡ VTOL with 20% increase in $c_2(t)$

— · — · — ≡ Neural network model output

5.2 Fourth Order Longitudinal Dynamics

5.2.1 Perturbation Equation

The study of the longitudinal dynamics is also carried for the 3 degree of freedom case. As shown by Ramnath [30], the decoupled equation for the perturbed variable u is given by:

$$p_4 u^{(4)} + p_3 u^{(3)} + p_2 u^{(2)} + p_1 u^{(1)} + p_0 u = 0 \tag{5.5}$$

where

$$\begin{aligned}
p_4 &= (10+t)^3(25+5t+t^2)(5+8t) \\
p_3 &= (100+29.25t+8.85t^2+0.97t^3)(25+5t+t^2)(10+t)^2(5+8t) \\
p_2 &= 4(631.25+715t+250.2875t^2+70.6175t^3+9.121t^4+0.8129t^5)(5+8t) \\
p_1 &= 8(1483.75+644t+152.4475t^2+20.85925t^3+1.4096t^4+0.07744t^5)(5+8t) \\
p_0 &= 4.83(10+t)(250+551.25t+68.5t^2+41.95t^3+3.78t^4+0.25t^5)
\end{aligned}$$

Note that in finding the numerical solution the equation needs to be rewritten as

$$u^{(4)} + c_3(t)u^{(3)} + c_2(t)u^{(2)} + c_1(t)u^{(1)} + c_0(t)u = 0$$

where the new coefficients are resulted from normalization by the coefficient p_4.

5.2.2 Root Locus Analysis

The stability is analyzed using root locus method. Since we have two forms of V variation and stability derivative M_w variation, we have four possible root locus configurations corresponding to the combination of V and M_w. The combination is again written for the sake of clarity.

$$-M_w = \begin{cases} -0.02 + 0.00025\frac{V}{150} \\ 0.005 + 0.015\left(\frac{V}{150}\right)^2 \end{cases}$$

$$V(t) = \begin{cases} \frac{150t}{10+t} ft/sec \\ \frac{150t}{20+t} ft/sec \end{cases}$$

Fig. 5-20 represents the root locus for the system using the first form of both M_w and V. In the early phase of the trajectory, there are two real roots in the left half plane and a pair of conjugate roots in the right half plane. As the time increases, the conjugate roots move to the left half plane but at the same time one of the real roots moves to the right half plane. The system is thus unstable. Fig. 5-21 represents the root locus for the system using the second form of V and the first form of M_w. The system exhibits the same stability behavior as that of the first case. Therefore, the system is also unstable.

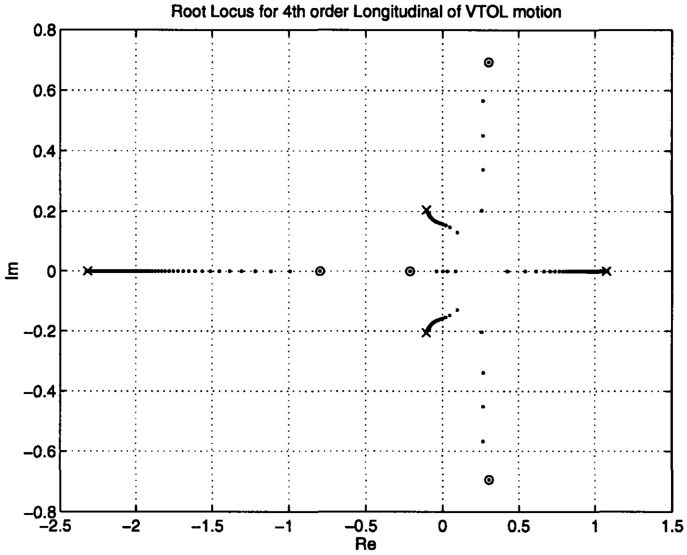


Figure 5-20: Root locus variation for 4th order VTOL longitudinal dynamics: $V(1)$ and $M_w(1)$

$\circ \equiv$ beginning of the trajectory $t = 0$

$\times \equiv$ end of the trajectory $t = 50$ sec.

Fig. 5-22 describes the root locus for the system using the second form of both M_w and V. The system is unstable in the early period of trajectory. This is shown by the presence of conjugate roots in the right half plane. However, these roots move to the left half plane when the vehicle moves along its trajectory. As a result, the system changes from the unstable to the stable condition.

Finally, Fig. 5-23 represents the root locus of the system using the first form of V and the second form of M_w. The behavior is the same as that of the previous

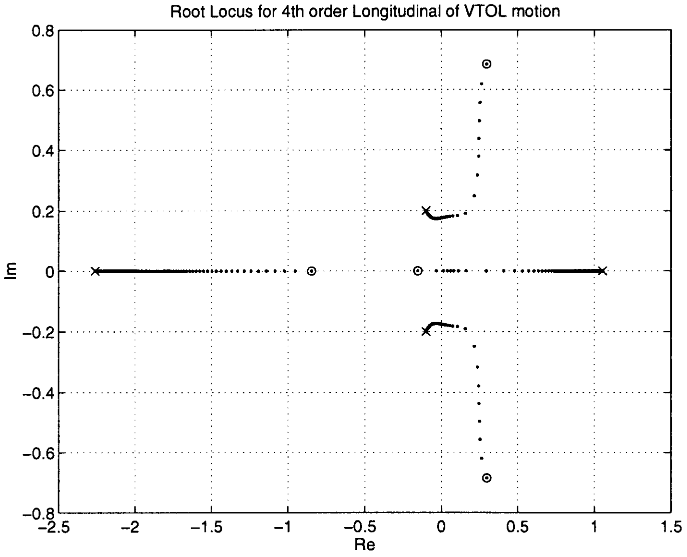


Figure 5-21: Root locus variation for 4th order VTOL longitudinal dynamics: $V(2)$ and $M_w(1)$

$\circ \equiv$ beginning of the trajectory $t = 0$
$\times \equiv$ end of the trajectory $t = 50$ sec.

case. Note that the 4th order VTOL dynamics model exhibits a unique phenomenon. It is characterized by the presence of drastic change in the topology of the solution shown in all of the above root locus. The location in the root locus at which this phenomenon happens is called the turning point. For the first two cases, it happens in the positive real axis around point 0.25. The phenomenon is characterized by a jump in the root locus branches. In the later two cases, the turning point happens in the negative real axis around point −0.5. This is the point where the two real roots in the negative real axis meet and then break up into two parts. This translates into changing the behavior from exponential to oscillatory.

For further analysis, the time response for the above root locus is generated. To specifically observe the turning point, only the latest case is considered. The time responses for different initial conditions are given in Fig 5-24 to Fig. 5-27. We observe some peculiarities in the time response around $t = 12$. This phenomenon corresponds to the turning point.

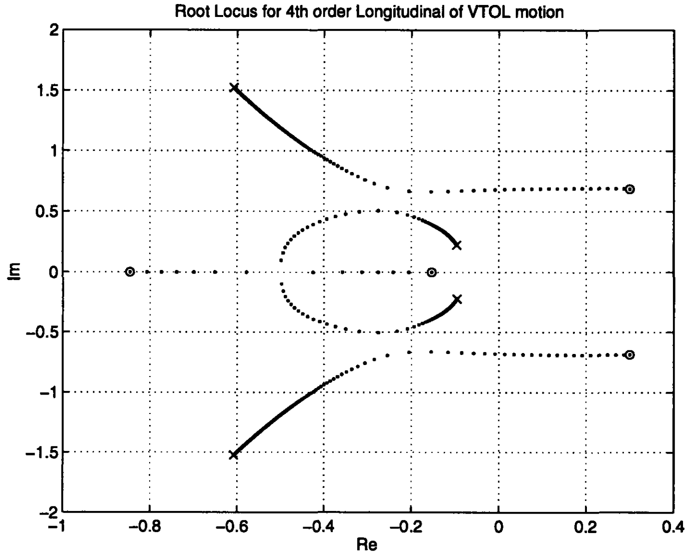


Figure 5-22: Root locus variation for 4th order VTOL longitudinal dynamics: $V(2)$ and $M_w(2)$

$\circ \equiv$ beginning of the trajectory $t = 0$

$\times \equiv$ end of the trajectory $t = 50$ sec.

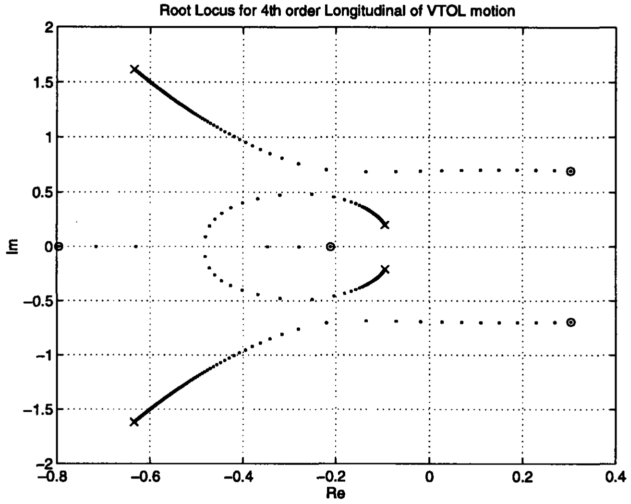


Figure 5-23: Root locus variation for 4th order VTOL longitudinal dynamics: $V(1)$ and $M_w(2)$

$\circ \equiv$ beginning of the trajectory $t = 0$

$\times \equiv$ end of the trajectory $t = 50$ sec.

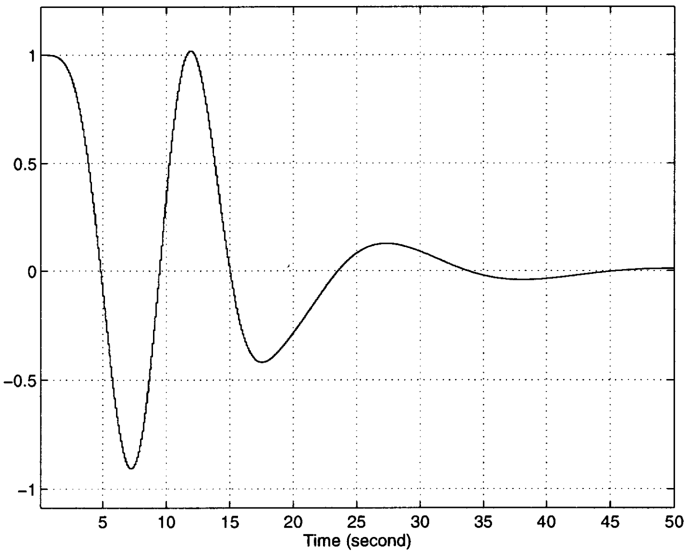


Figure 5-24: Time response of u for 4th order VTOL longitudinal dynamics model

$$\begin{aligned} u\,(0) &\equiv 1 \\ \dot{u}\,(0),\, \ddot{u}\,(0),\, \dddot{u}\,(0) &\equiv 0 \end{aligned}$$

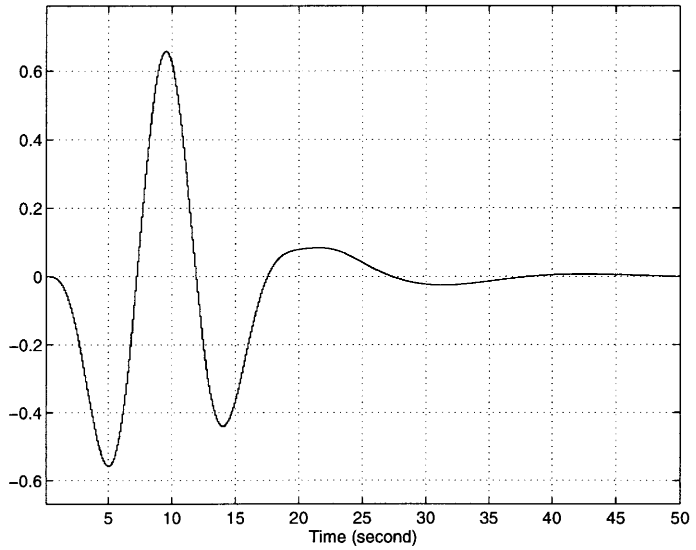


Figure 5-25: Time response of $\dot{u}$ for 4th order VTOL longitudinal dynamics model

$$\begin{aligned} u\,(0) &\equiv 1 \\ \dot{u}\,(0),\, \ddot{u}\,(0),\, \dddot{u}\,(0) &\equiv 0 \end{aligned}$$

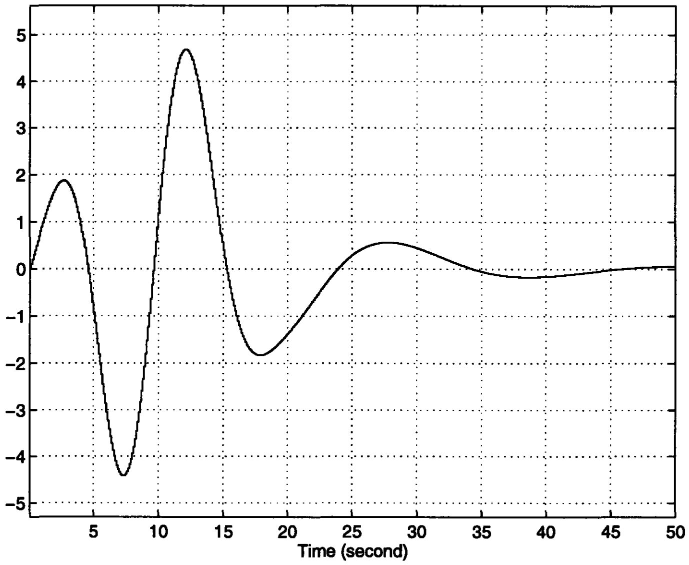


Figure 5-26: Time response of u for 4th order VTOL longitudinal dynamics model

$$\begin{array}{lcl} \dot{u}\,(0) & \equiv & 1 \\ u\,(0),\ \ddot{u}\,(0),\ \dddot{u}\,(0) & \equiv & 0 \end{array}$$

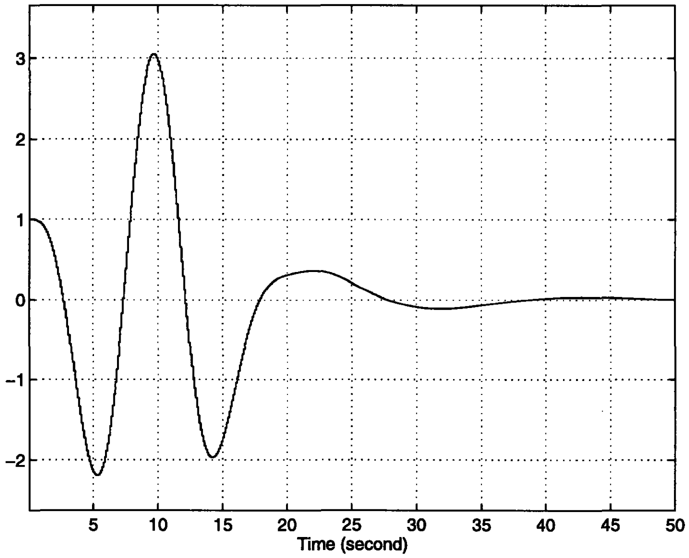


Figure 5-27: Time response of $\dot{u}$ for 4th order VTOL longitudinal dynamics model

$$\begin{array}{lcl} \dot{u}\,(0) & \equiv & 1 \\ u\,(0),\ \ddot{u}\,(0),\ \dddot{u}\,(0) & \equiv & 0 \end{array}$$

5.2.3 Sensitivity Analysis

The sensitivity equation of the 4th order VTOL aircraft dynamics can be derived starting from the general equation of motion:

$$u^{(4)} + c_3(t)u^{(3)} + c_2(t)u^{(2)} + c_1(t)u^{(1)} + c_0(t)u = 0$$

Following the similar technique as for the GHAME sensitivity analysis, the equation is rearranged to include the sensitivity variable s . And the final result can be written as:

$$s^{(4)} = -c_3 \dddot{s} - c_2\ddot{s} - c_1\dot{s} - c_0 s - c_{3_p} \dddot{u} - c_{2-p}\ddot{u} - c_{1_p}\dot{u} - c_{0_p}u$$

The sensitivity of the longitudinal dynamics to all of the stability derivatives is studied. In solving the differential equation, we need to first calculate the partial derivatives of the time-varying coefficients $c's$ with respect to the stability derivatives. As an illustration, the partial derivatives for M_q and M_w can be calculated as:

$$\begin{aligned}
\frac{\partial c_3}{\partial M_q} &= -1 \\
\frac{\partial c_2}{\partial M_q} &= X_u + \frac{\dot{M}_w}{M_w} \\
\frac{\partial c_1}{\partial M_q} &= -X_u Z_w - \frac{\dot{M}_w}{M_w} X_u + 2\dot{X}_u \\
\frac{\partial c_0}{\partial M_q} &= -\dot{X}_u \frac{\dot{M}_w}{M_w} + \ddot{X}_u
\end{aligned}$$

$$\begin{aligned}
\frac{\partial c_3}{\partial M_w} &= \frac{\dot{M}_w}{{M_w}^2} \\
\frac{\partial c_2}{\partial M_w} &= -V - X_u \frac{\dot{M}_w}{{M_w}^2} \\
\frac{\partial c_1}{\partial M_w} &= V X_u + \frac{\dot{M}_w}{{M_w}^2} X_u M_q - 2\dot{X}_u \frac{\dot{M}_w}{{M_w}^2} \\
\frac{\partial c_0}{\partial M_w} &= g Z_u + g \frac{\dot{M}_w}{{M_w}^2} M_u + \dot{X}_u \left(V + \frac{\dot{M}_w}{{M_w}^2} M_q\right) - \ddot{X}_u \frac{\dot{M}_w}{{M_w}^2}
\end{aligned}$$

The results of the sensitivity analysis are shown in the following figures. From the figures, we can observe that the longitudinal dynamics is most sensitive to M_u and M_w. Both are in the order of 10^2. The other stability derivatives do not give substantial effect to the longitudinal dynamics. In particular, the longitudinal dynamics is least sensitive to X_u i.e. the variation of this stability derivative is not significant.

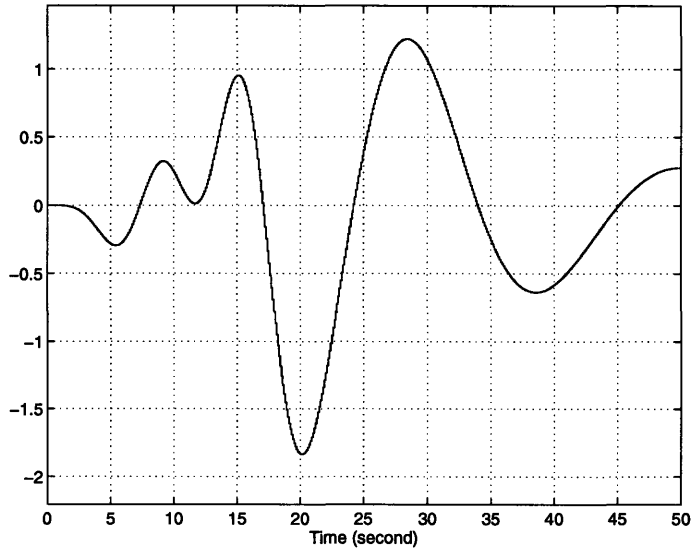


Figure 5-28: Sensitivity of 4th order VTOL longitudinal dynamics to X_u

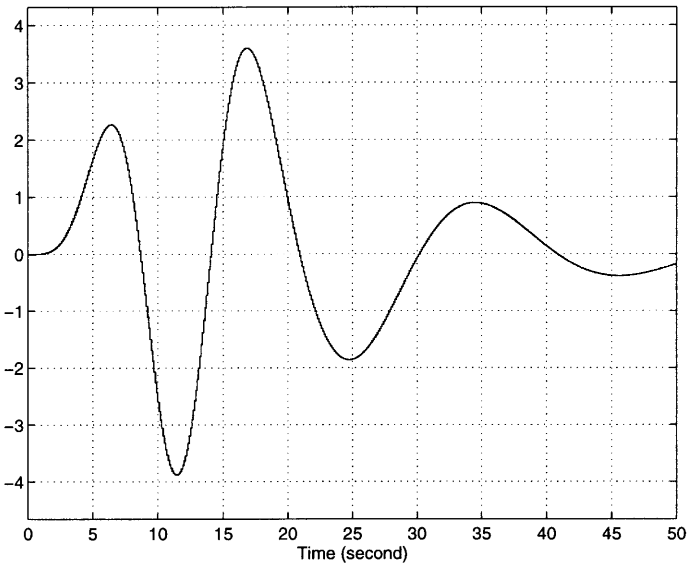


Figure 5-29: Sensitivity of 4th order VTOL longitudinal dynamics to Z_u

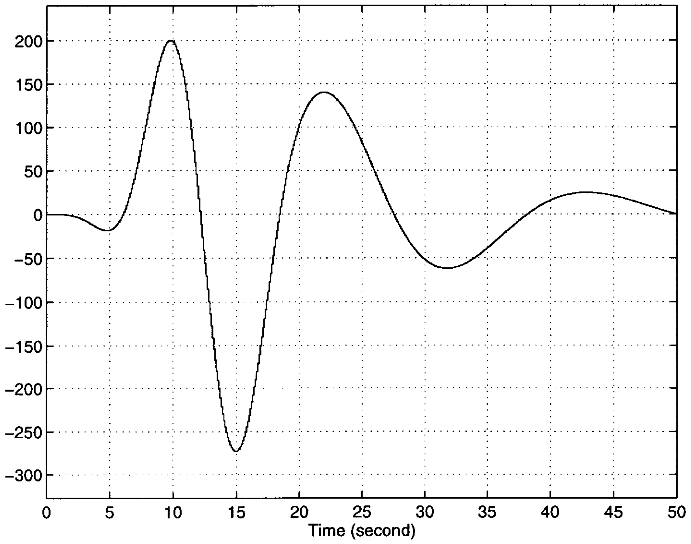


Figure 5-30: Sensitivity of 4th order VTOL longitudinal dynamics to M_u

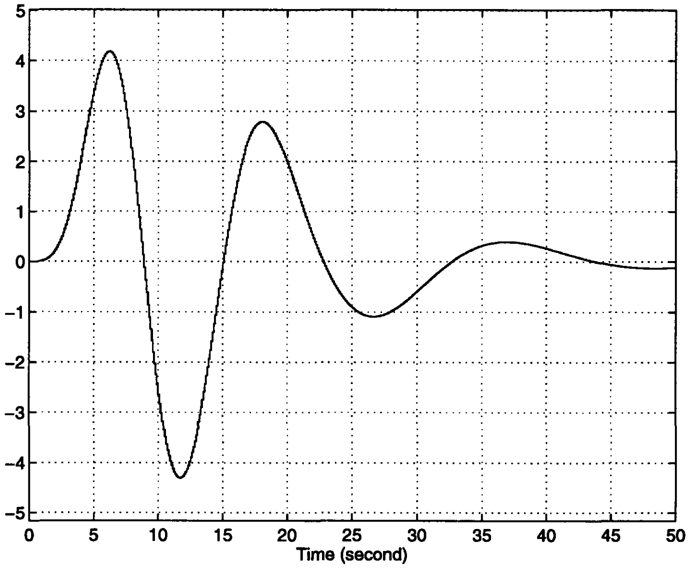


Figure 5-31: Sensitivity of 4th order VTOL longitudinal dynamics to Z_w

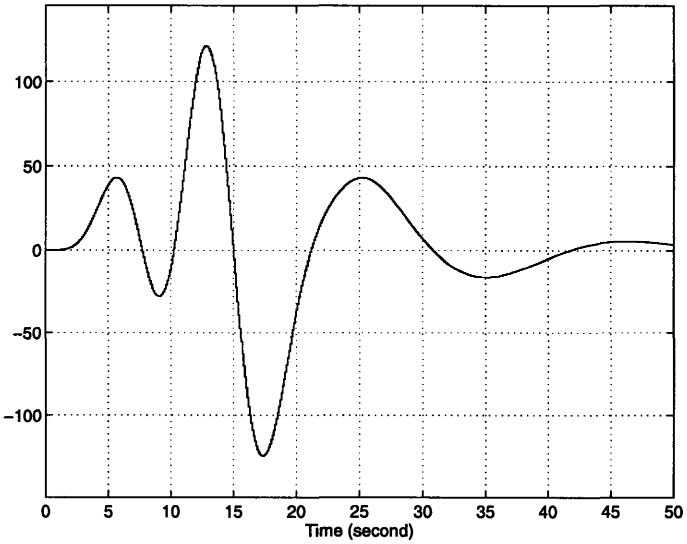


Figure 5-32: Sensitivity of 4th order VTOL longitudinal dynamics to M_w

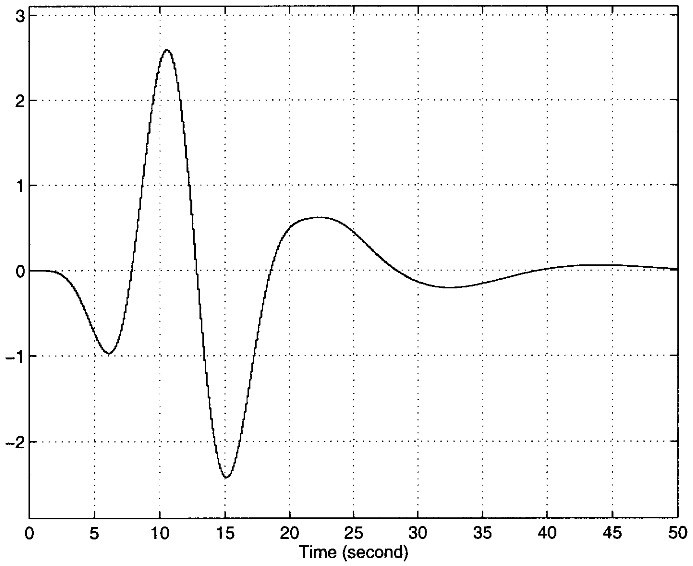


Figure 5-33: Sensitivity of 4th order VTOL longitudinal dynamics to M_q

Chapter 6

Neural Networks Application for the Control of a Time-varying System

This chapter describes the implementation of a neural network for control. As we are dealing with time-varying systems, the notion of on-line learning is first presented. It is critical when we use neural networks for system identification in the control system architecture such as in the indirect adaptive control scheme. In that case, the success of the control scheme will depend largely on the performance of the system identification networks (neural network model). As will be shown later, the neural network model in that scheme acts as a *teacher* that trains the neural network controller to generate a certain desired control command. To evaluate the performance, the neural network controller is compared to two different techniques i.e. PI controller and adaptive controller.

6.1 On-line Learning

6.1.1 Why On-line Learning?

We learn from the previous chapters that in dealing with dynamic systems we can no longer rely on the use of an off-line training. The backpropagation method, discussed in Chapter 1, is simply an efficient method for calculating derivatives of a single target quantity (such as pattern classification error) with respect to a large set of input quantities (such as the parameters or weights in a classification rule). An extension of the backpropagation method is therefore necessary to deal with systems with time-varying parameters or operating in changing environments. The *backprop-*

agation through time allows us to calculate the derivatives needed when optimizing an iterative analysis procedure, a neural network with memory, or a control system which maximizes performance over time [8]. Note that the off-line learning described in Chapter 1 is still useful and relevant to the whole problem of system identification and control. First, it establishes the basis for the training of a multi-layered network and describes how the *ordered derivatives* are used to calculate the error. Secondly, it is often advantageous to train the networks off-line prior to permitting them to perform an on-line learning during real operation.

6.1.2 Backpropagation Through Time [15]

The backpropagation technique is described using a one-layer recurrent network as shown in Figure 6-1. The feed-back block represents the discrete transfer function that delays its input (the output of the network) by one sample interval. All elements of an input are delayed by the same sample delay. The definition of z^{-1} is given by the z-transform below [1]:

$$\mathcal{Z}\left[\delta\left(t-kT\right)\right]=z^{-1} \tag{6.1}$$

where k is any integer and T is the sampling time.

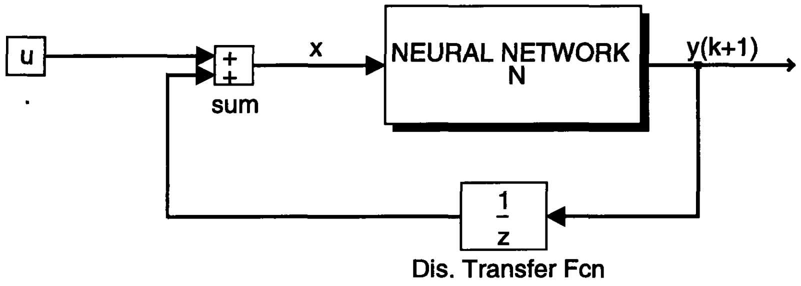


Figure 6-1: N-feed-forward multilayer neural network, z^{-1} – time shift operator

The network can be described by the following state equations:

$$y(t+1)=G(y(t),u(t)) \tag{6.2}$$

where G is non-linear mapping and u is control input. Figure 6-2 presents a block diagram of a system which can be described by Eqn. 6.2.

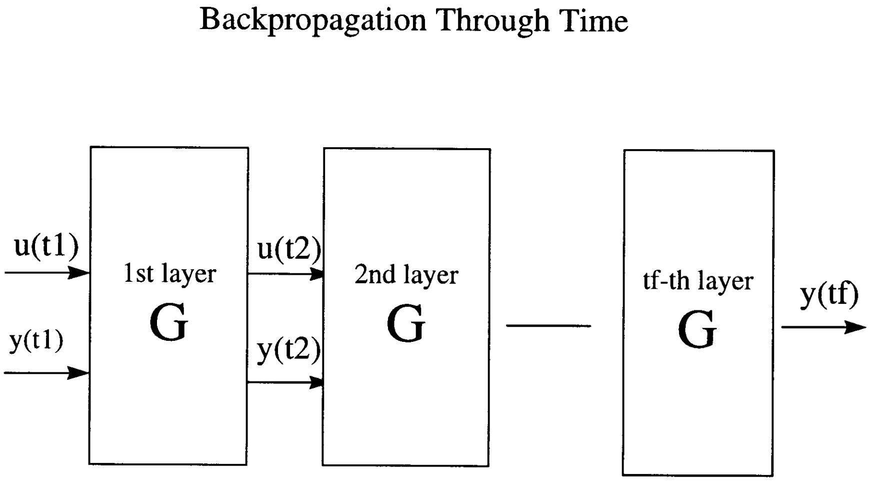


Figure 6-2: Structure of backpropagation through time

The objective of learning in dynamic networks is to present a suitable algorithm to follow a desired trajectory over time (t_0, t_f). For simplicity we shall concentrate only on discrete-time systems. The error is defined as

$$E(t) = d(t) - y(t), \qquad t = t_0, t_0 + 1, ..., t_f \tag{6.3}$$

where $y(t)$ denotes the system output at the time instant t.

The performance criterion is defined as

$$J = \frac{1}{2}\sum_{t_0}^{t_f} E^T(t)\, E(t) \tag{6.4}$$

The idea of backpropagation through time is to unfold the network through time, i.e. replace the one-layer recurrent feed-forward network with t_f layers (see Fig. 6-2) represented by the same neural network modeling the mapping G.

It follows from Eqn. 6.4 that

$$\frac{\partial J(W)}{\partial w_i} = \frac{1}{2}\sum_{t_0}^{t_f} E^T(t)\,\frac{\partial E(t)}{\partial w_i} \tag{6.5}$$

The derivatives $\frac{\partial E(t)}{\partial w_i}$ of the errors at the subsequent time instants with respect to the weight w_i can be computed by applying a static back-propagation scheme (given in Chap.1) at each time instant based on the input produced at the previous time instant and the error corresponding to this time instant. For more complete derivation, the reader can refer to [15] and [8].

6.2 Neural Networks for an Online System Identification

6.2.1 Motivation

It was seen in Chapter 2 that flight vehicles generally exhibit time-varying dynamic properties. Most of them result from plant non-stationarity, non-linearity and random disturbances which affect the plant behavior in the following way:

- *Non-stationary.* The plant is called non-stationary if its dynamics change in time, e.g. as a result of ageing effects after being in operation for a long time.

- *Non-linear.* If the plant is non-linear then the dynamic properties of its linearized model are different in the vicinity of various steady-state points (in normal operating conditions the steady-state point changes).

- *Stochastic.* Stochastic models are used to represent the disturbances acting at the plant output because of the large number and different nature of the factors disturbing the normal plant operation [15].

All the above factors essentially can exist in the operation of the two flight vehicles that we consider in this work. We have observed that the dynamics of those vehicles are described by time-varying differential equation. This factor alone has made the design of control techniques for such system difficult. Others include disturbances affecting the plant, improper plant model structure or a change in the plant parameters. To design a robust control system, we need a system identification approach that has the capability to capture all changes in the plant parameters.

6.2.2 System Identification for VTOL using Neural Networks

The XC-142 aircraft is taken as the plant to be identified. Note that the neural networks do not need the precise representation of the system. The networks can learn the dynamics of the plant by searching the functional relationship between the input and output of the plant. For this purpose, the dynamics of the XC-142 aircraft is represented by the discrete dynamic relation between input u_c and output perturbation velocity u given as:

$$u(i) = f\left(u(i-1), u(i-2), ..., u(i-nA),\right. \tag{6.6}$$

$$\left. u_c(i-1-k), u_c(i-2-k), ..., u_c(i-nB-k)\right) \tag{6.7}$$

where i denotes discrete time and f is an unknown function to be learned by neural networks. The number of input neurons is given by $nA + nB$ and k refers to additional discrete-time delay. For our case, nA and nB is taken to be 3 and 1 respectively. And we assume $k = 0$. Ten hidden neurons are used in the neural networks model. The structure of the networks is given by Fig. 6-4. Note that input and output neurons generally use linear activation function. Following the notations we used in Chapter 1, we can derive the equation that relates the output and inputs of the neural networks model. Neuron i and the layer j is defined by $x_{j,i}$. We have 15 neurons in the networks.

$$x_{1,i} = u_c(t-i), \quad i = 1 \tag{6.8}$$

$$x_{1,i} = u(t-i), \quad 2 \leq i \leq 4 \tag{6.9}$$

$$\phi_i = \sum_{k=1}^{4} W_{i,k} x_{1,k}, \quad 4 < i \leq 14 \tag{6.10}$$

$$y_i = \frac{1}{1+e^{-\phi_i}} \tag{6.11}$$

$$u(t) = x_{3,15} = \sum_{k=1}^{10} W_{i,k} y_{k+4}, \quad i = 15 \tag{6.12}$$

The dimension of the weights matrix relating the input layer and hidden layer is 10×4 and that of the one relating hidden layer and output layer is 10×1. As an illustration their initial values are given in the two matrices below:

Weight matrix relating the input neurons and hidden neurons:

```
0.6762    0.6614    0.0150    0.4558
0.5139    0.4018    0.2742    0.8171
0.7286    0.6056    0.6430    0.0221
```

```
0.7208    0.9865    0.5461    0.1607
0.9448    0.1504    0.9178    0.7069
0.4607    0.6701    0.2666    0.7078
0.9402    0.3438    0.9701    0.4366
0.3216    0.5417    0.2467    0.5824
0.4604    0.5228    0.8440    0.7517
0.5171    0.8292    0.6940    0.9915
```

Weight matrix relating hidden neurons and output neuron:

```
0.6957
0.2795
0.7539
0.1712
0.0043
0.4957
0.0799
0.1631
0.9033
0.7830
```

The XC-142 aircraft is simulated for 150 second during which the aircraft experiences a transition from unstable hover into stable forward flight. The neural network is used as a recursive (on-line) identification scheme. The effects of some learning parameters are investigated. Those include the learning rate, the number of hidden neurons and the activation function for the hidden neurons. The performance of the neural network model is evaluated by the mean square error (MSE) between the actual output and the estimation from the model. However, it turns out that this measure alone does not give sufficient guarantee to the overall model performance. Thus, the evaluation must include the actual behavior of the model in tracking the true dynamics of the system. Figure 6-5 through Fig 6-10 present comparison of the neural network model performance for different learning rates. The simulation is done with the same network structure using ten hidden neurons with sigmoidal activation function. Table 6.1 summarize the MSE comparison. The table indicates that the higher learning rates can give lower MSE. However, as can be observed from Figure 6-9 and 6-10, the network model might exhibit unfavorable behavior if the learning rate is too high. This phenomenon is not uncommon in the system identification and can be explained by the fact that for high learning rate, the model tends to give too *confident* an estimate resulting in an overshoot.

The effect of the number of the hidden neurons used in the networks is shown by Fig. 6-11. The MSE comparison is given by Table 6.2. The number of hidden neurons

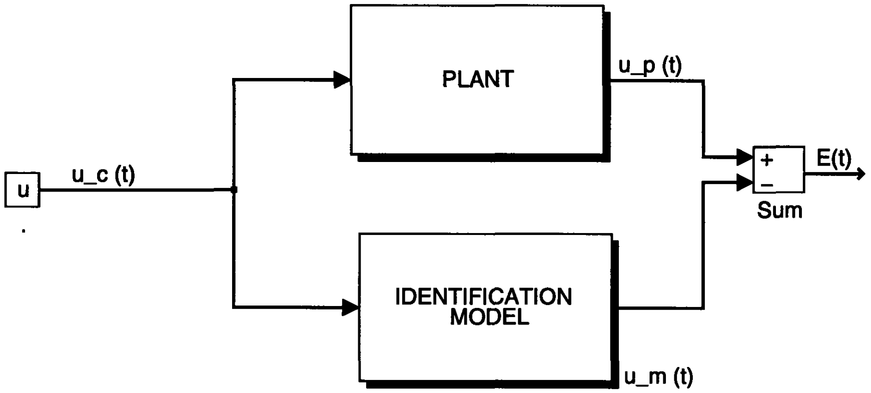


Figure 6-3: System Identification Scheme

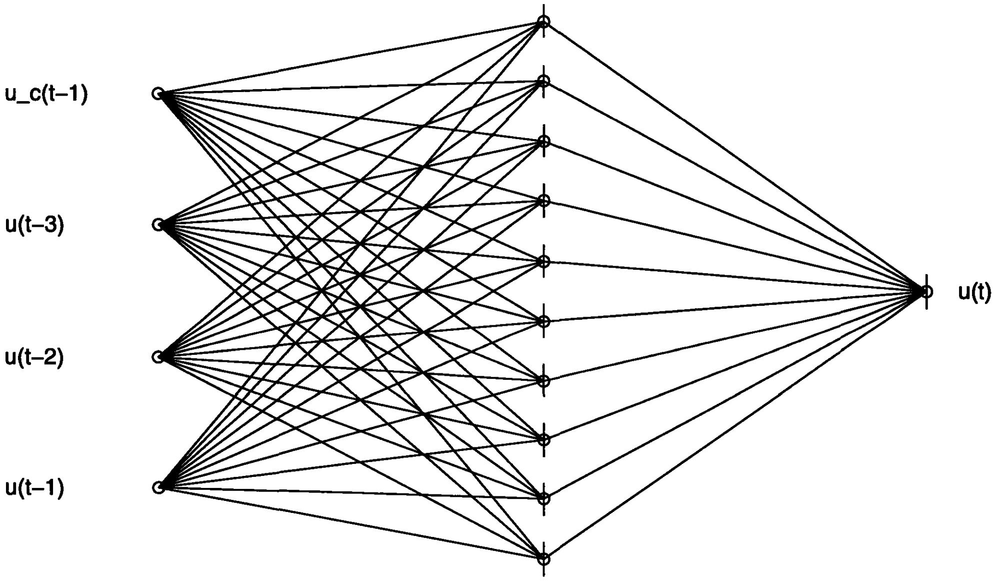


Figure 6-4: Structure of Neural Networks Model

Table 6.1: Mean Square Error comparison for different learning rates

Hidden sigmoidal neuron #	Learning rate	Mean Square Error
10	0.07	2.746×10^{-4}
10	0.1	3.1625×10^{-4}
10	0.17	2.1892×10^{-4}
10	0.2	2.1877×10^{-4}
10	0.4	1.6435×10^{-4}
10	0.5	1.6186×10^{-4}

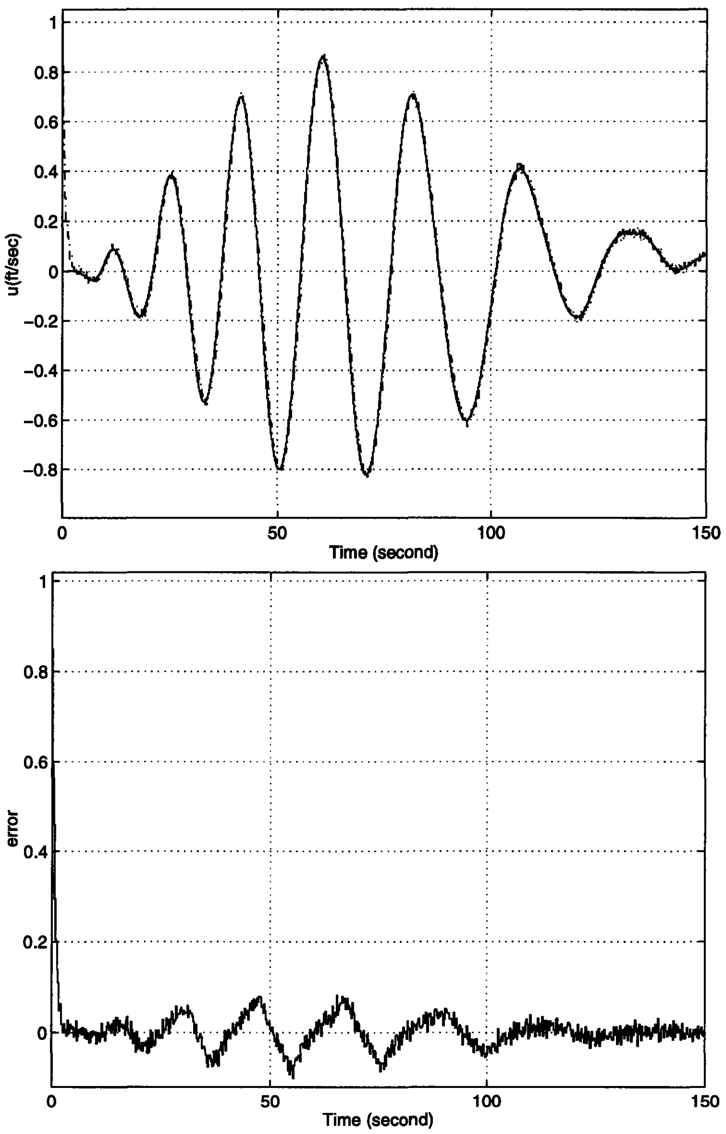
Table 6.2: Mean Square Error comparison for different number of hidden neurons

Hidden sigmoidal neuron #	Learning rate	Mean Square Error
3	0.2	1.4282×10^{-4}
5	0.2	2.0391×10^{-4}
10	0.2	2.1870×10^{-4}
15	0.2	1.5862×10^{-4}
20	0.2	4.8398×10^{-4}

Table 6.3: Mean Square Error comparison for different activation function

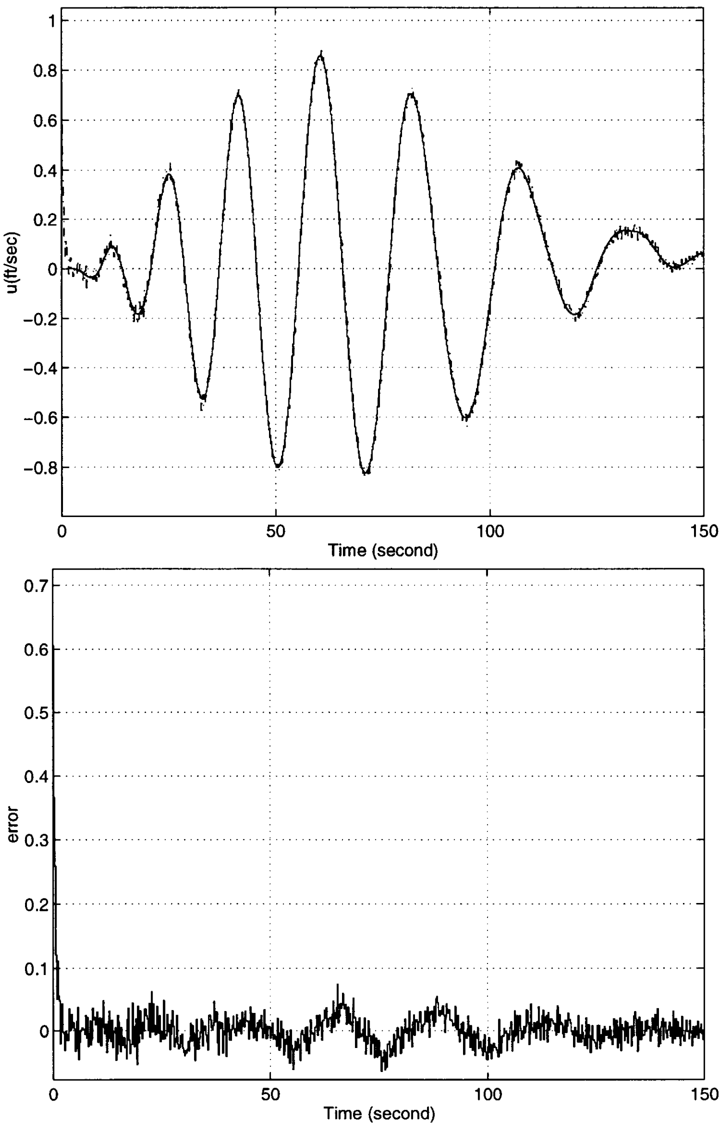
Hidden neuron	Learning rate	Mean Square Error
sigmoidal function	0.2	2.187×10^{-4}
linear	0.2	19.831×10^{-4}

affects the performance of the system identification in a subtle way. From the table we notice that the more neurons used does not always give better performance. In fact, for the case of 20 hidden neurons we get degrading performance reflected both in the MSE and in the *noisy* networks output. There is a common agreement among researchers that the number of neurons used is mainly dictated by the nature of the plant. As a general rule, the figure between as many and twice as many as the number of input neurons will be sufficient for a good system identification performance. Note that low number of hidden neuron can sometime give an acceptable low error. However, we must consider the possibility that the system might be underrepresented. This can lead to unfavorable situation if the system parameter changes beyond the networks identification capability. Finally, we must note the effects of the activation function. In general, sigmoidal function gives a wider region of system representation. Table 6.3 give the error comparison between the networks using sigmoidal function and linear function. Notice that linear time-varying system requires nonlinear representation of neuronal activation function to give an acceptable neural network model performance.



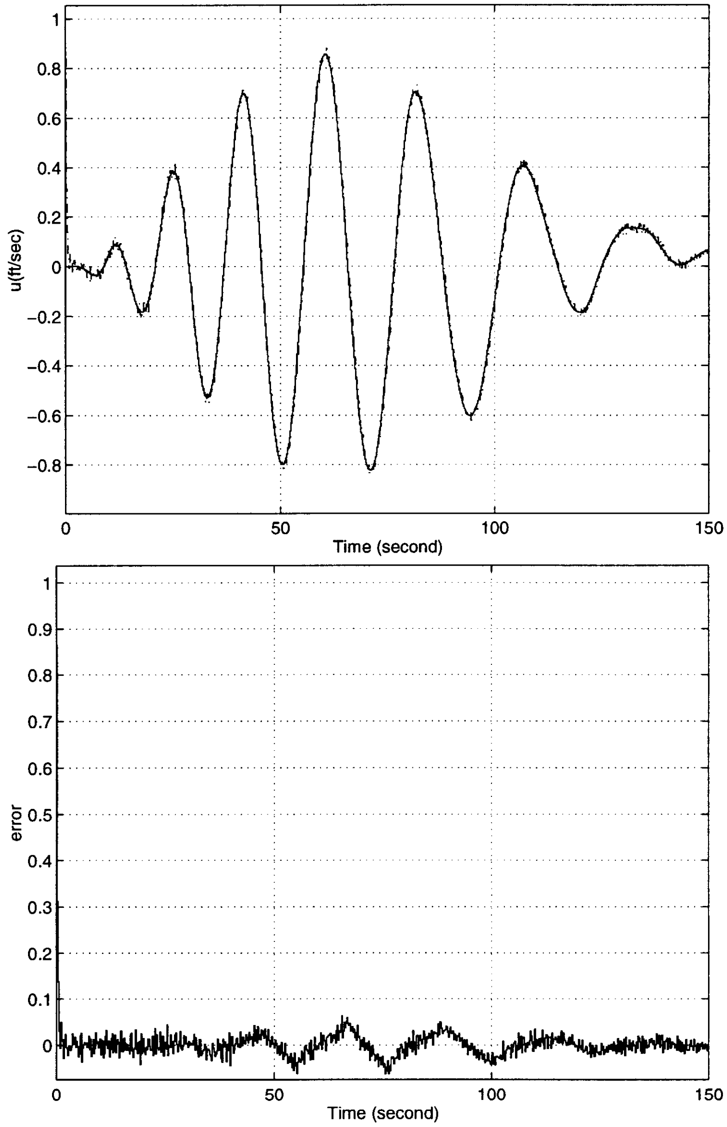
——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron sigmoidal activation function
Learning rate	0.07

Figure 6-5: ANN for an Online System Identification of VTOL dynamics



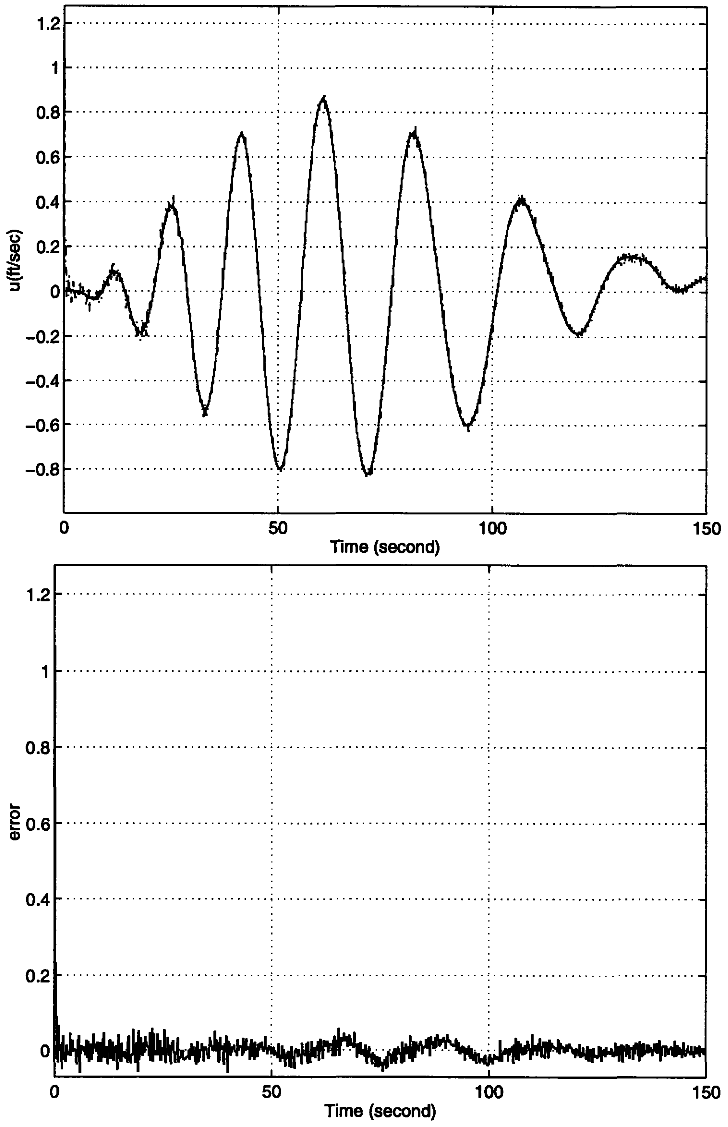
——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron sigmoidal activation function
Learning rate	0.1

Figure 6-6: ANN for an Online System Identification of VTOL dynamics



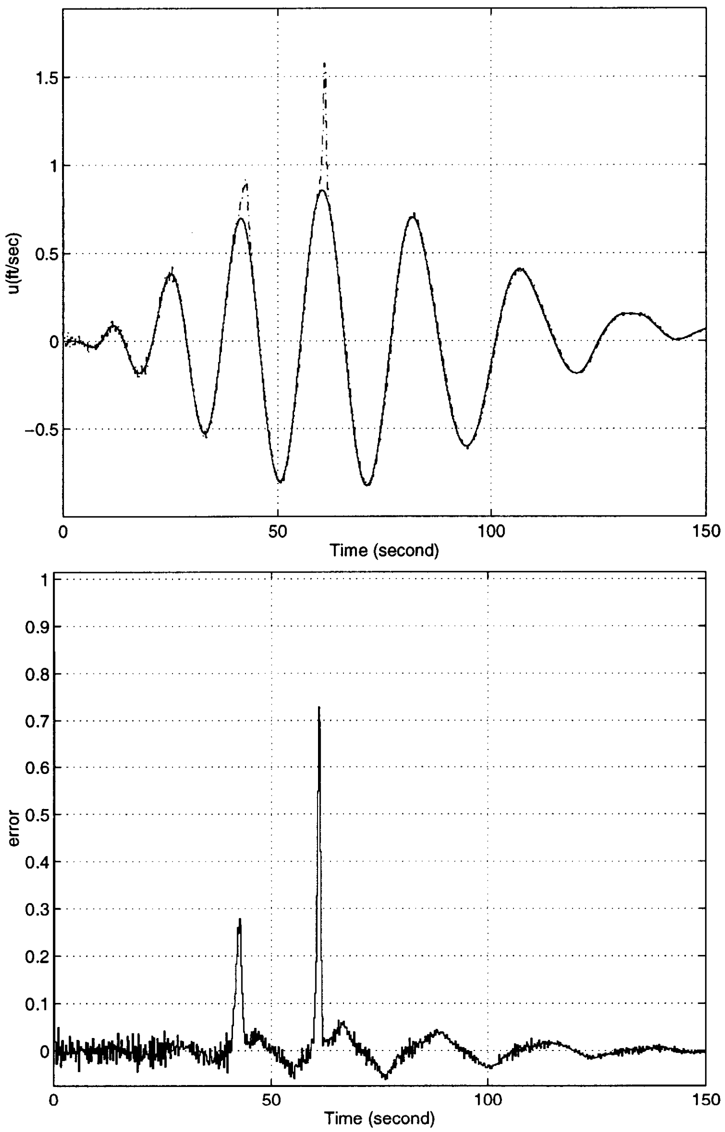
——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron sigmoidal activation function
Learning rate	0.17

Figure 6-7: ANN for an Online System Identification of VTOL dynamics



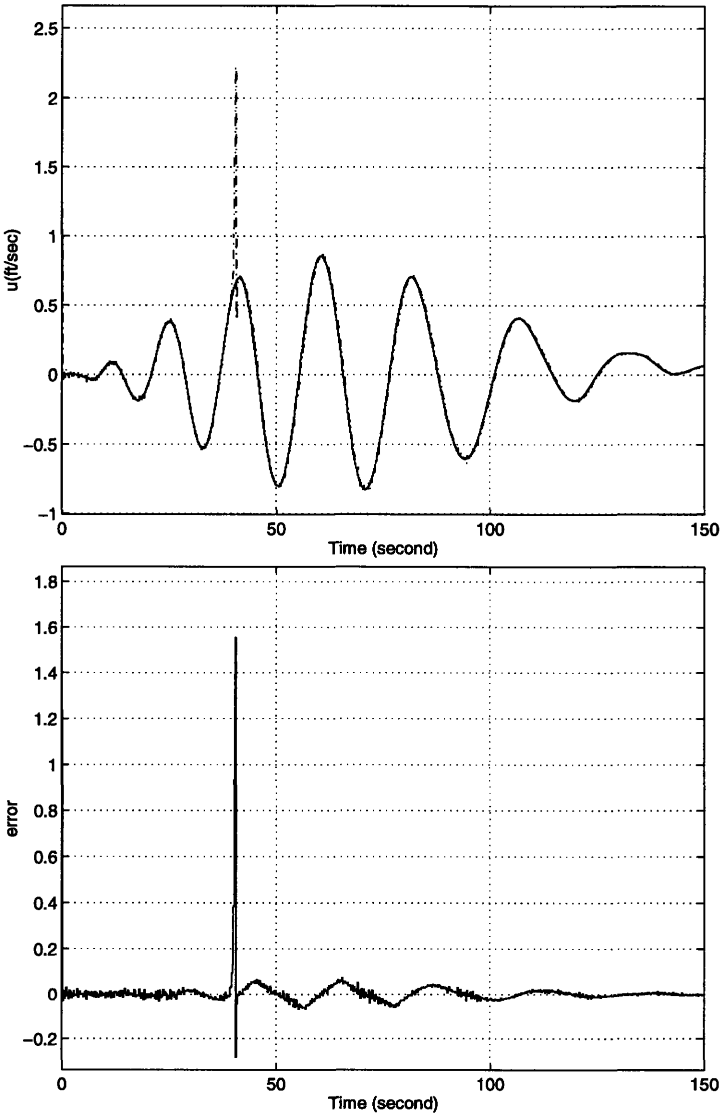
——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron sigmoidal activation function
Learning rate	0.2

Figure 6-8: ANN for an Online System Identification of VTOL dynamics



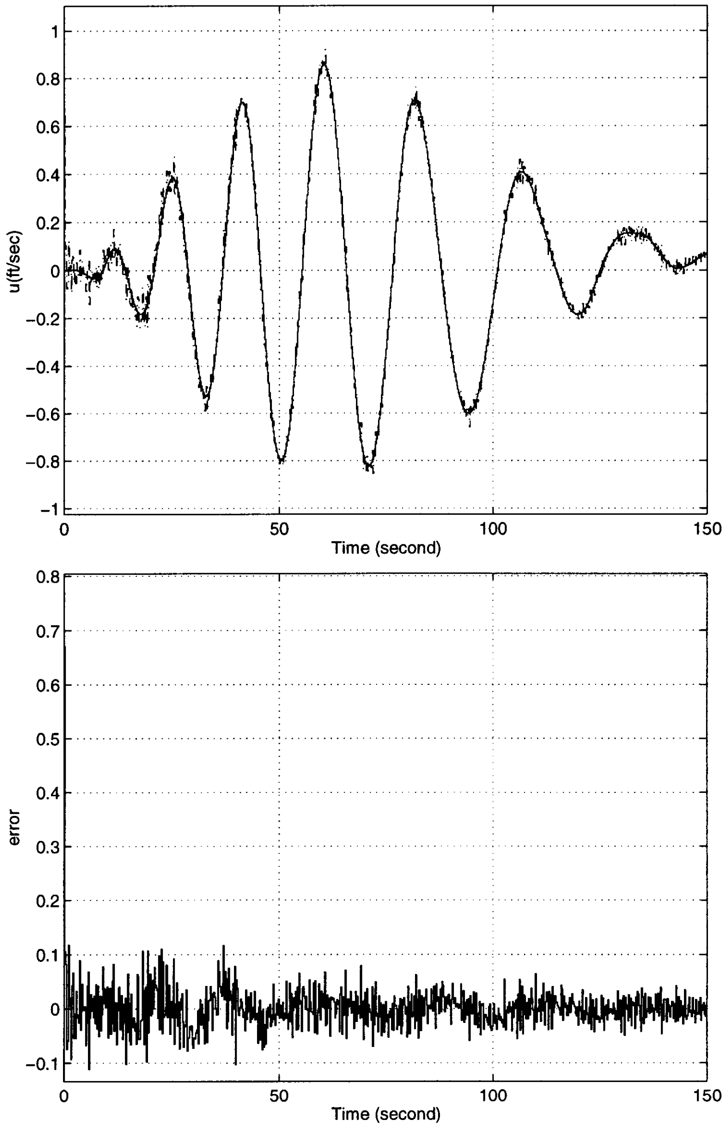
——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron sigmoidal activation function
Learning rate	0.4

Figure 6-9: ANN for an Online System Identification of VTOL dynamics



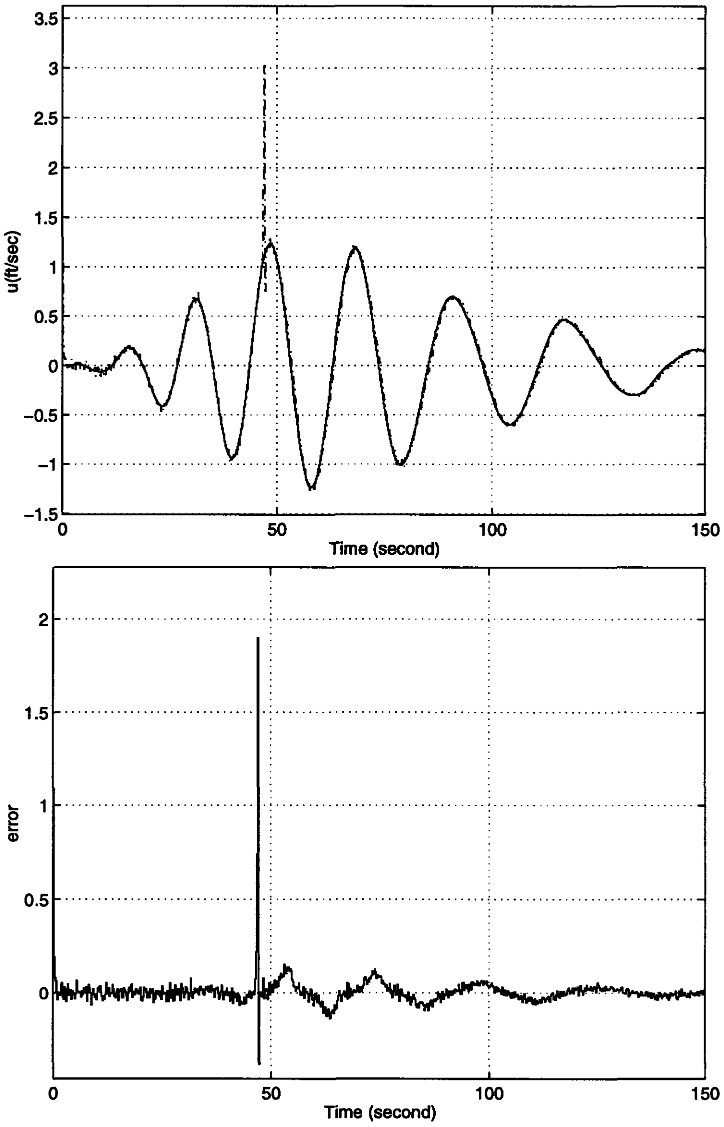
——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron sigmoidal activation function
Learning rate	0.5

Figure 6-10: ANN for an Online System Identification of VTOL dynamics



——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	20 hidden neuron sigmoidal activation function
Learning rate	0.2

Figure 6-11: ANN for an Online System Identification of VTOL dynamics



——	≡ VTOL output
— · — · —	≡Neural network model output
Network Structure	10 hidden neuron linear activation function
Learning rate	0.2

Figure 6-12: ANN for an Online System Identification of VTOL dynamics

Now, we want to examine the robustness of the system identification in response to changes in plant parameters. We will test how the networks can cope with a drastic change in the dynamic of VTOL during transition. In reality, it can correspond to some unknown problems in the rotor system. The problem is emulated by the change in the stability derivatives. For our purpose, we choose the stability derivative M_u for problem emulation due to its profound effect on the vehicle stability. It is assumed that a new problem arises at $t = 50$ second. This is represented by a drastic change in M_u described by Fig. 6-13. The expression for M_u is given as:

$$-M_u = \begin{cases} 0.015\left(-1 + \frac{V}{150}\right) & 0 \le t < 50 \\ 0.0015\left(-1 + \frac{V}{150}\right) & 50 \le t < 150 \end{cases} \tag{6.13}$$

The neural networks model with 10 hidden sigmoidal neurons is now used to identify the system on-line. The performance of the neural networks in coping with the change in plant parameter M_u is shown in Fig. 6-14. It is shown that the networks can successfully capture the change in the dynamics. This feature is essential for the success of designing a robust control system that can cope with all changes in the plant parameters as will be shown in the next section.

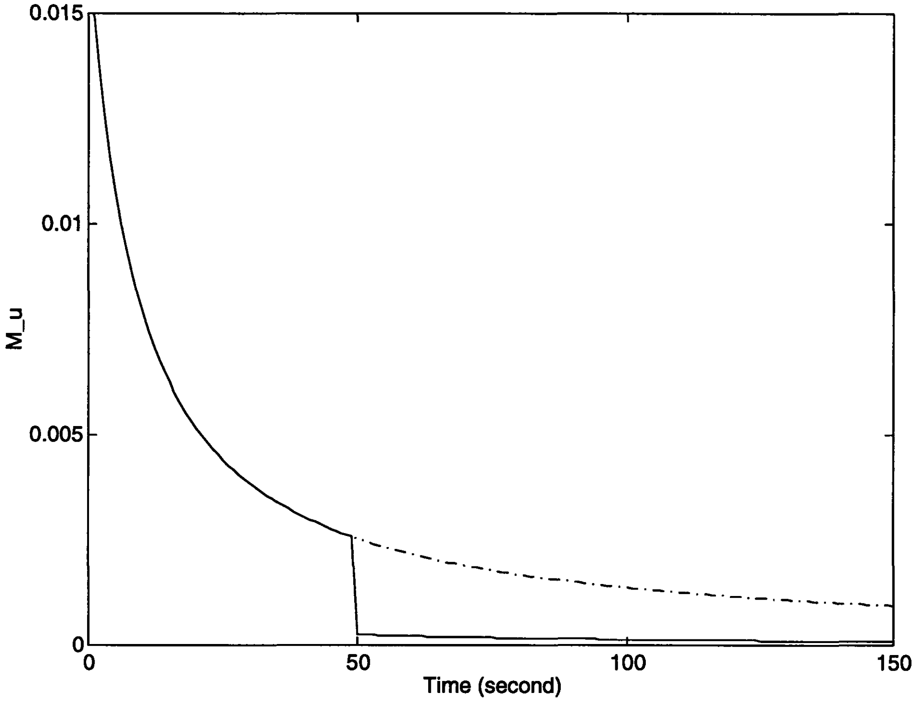


Figure 6-13: Variation of Stability Derivatives M_u

—— ≡ Variation in M_u with change at $t = 50$

— · — · — ≡ Original M_u variation during transition

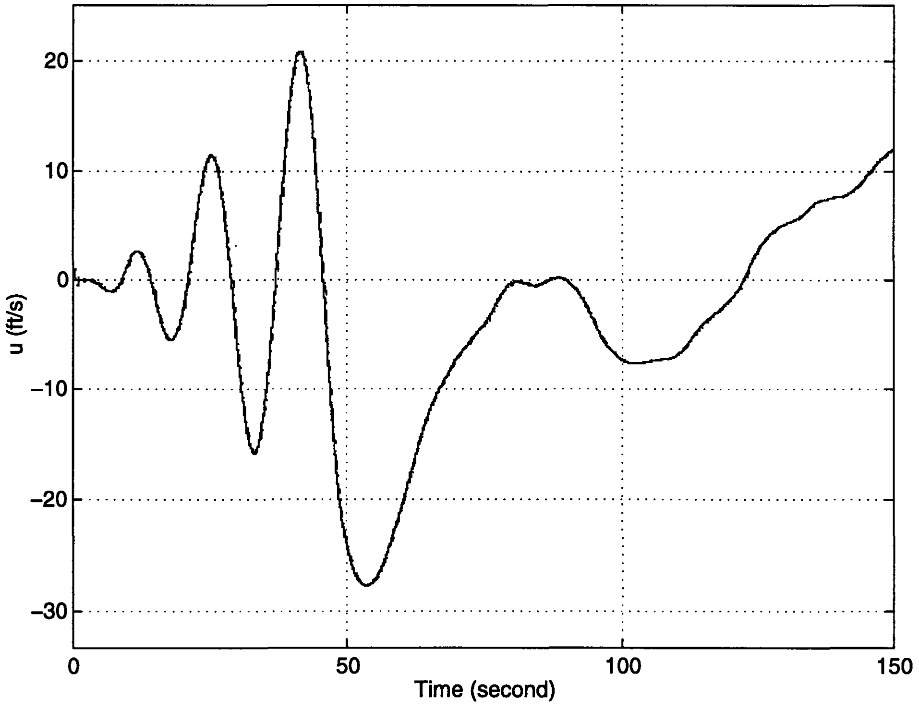


Figure 6-14: Neural Networks for Online System Identification of VTOL dynamics with the presence of change in M_u

—— ≡ VTOL output

— · — · — ≡ Neural network model output

6.3 Neural Networks for the Control of VTOL Aircraft

This section elaborates the use of neural networks for the control of VTOL aircraft. To cope with the change in plant parameters, indirect adaptive control scheme is chosen as the candidate. In this scheme, neural network is used both as system identification and controller. The neural networks model that has been designed previously is used as the system identification.

6.3.1 Indirect Adaptive Control (IAC) Scheme

Models of dynamic system and their inverses have immediate utility for control. In this work, we assume that such models are available in the form of neural networks the structure of which has been outlined above. However we will just focus on one possible architecture of control system involving neural networks. It is beyond the scope of this thesis to provide a complete survey of all neural network based architecture available in the literature. The scheme that we are interested in uses neural network model as a mean of training the neural network controller. Figure 6-15 describe the schematic diagram of the so-called indirect adaptive control. We take the XC-142 as the plant to be controlled. For the sake of simplicity, we just observe the dynamics of perturbation velocity u.

In this approach a neural network model provides a prediction of a plant's future response over a specified state horizon. The predictions are supplied by the network as passed to numerical optimization routine which attempts to minimize a specified performance criterion in the calculation of a suitable control signal. The control signal u_c is chosen to minimize the quadratic performance criterion which compromises between the tracking error and the control cost:

$$J = \sum_{j=N_1}^{N_2} \left(u^r\left(t+j\right) - u^m\left(t+j\right)\right)^2 + \sum_{j=1}^{N_2} \lambda_j \left(u_c\left(t+j-1\right) - u_c\left(t+j-2\right)\right)^2 \tag{6.14}$$

Here the constants N_1 and N_2 define the horizons over which the tracking error and control increments are considered. The values of λ are the control weights. Once the iterative optimization algorithm finds the optimal solution u_c it is applied to the plant. The actual value of the plant output u_p is measured and jointly with the reference signal r it is sent to the neural network controller. This network is trained to produced the same control output u_c as the optimization routine. As the result the nonlinear feedback control law is obtained. Using the above scheme, the control

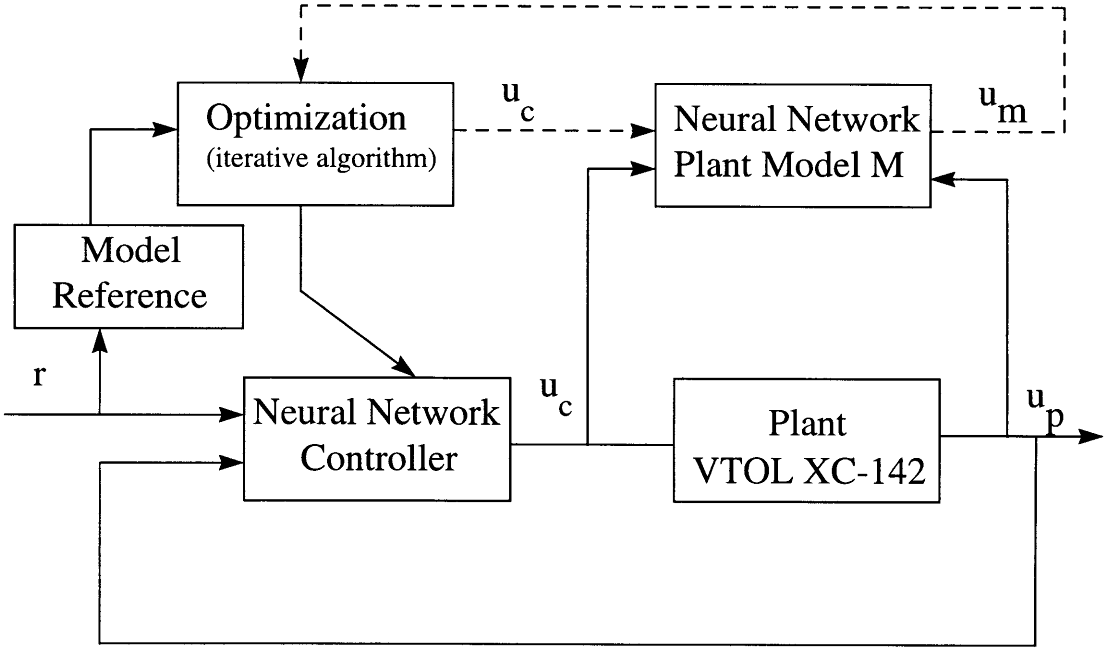


Figure 6-15: Indirect Adaptive Control

to stabilize the VTOL XC-142 during transition is developed. The parameters N_1 and N_2 are taken to be 1 and 4 respectively. The reference signal is square wave with frequency of 0.1 and amplitude of $0.01ft.$. The results are presented in the following section.

6.3.2 Simulation Results

This section summarizes and discusses the results of the indirect adaptive control employing neural networks for XC-142 during transition. The performance of the IAC system is shown in Figure 6-16. After approximately 20 seconds the control system can successfully track the reference signal. To observe the tracking performance the comparison between the reference signal and the VTOL output is shown for the time period after 20 seconds as shown in Fig. 6-17. The figure presents the tracking performance and the amount of control input to achieve it. To investigate the robustness of the control technique some unfavorable factors are introduced into the system. The factors represent realistic aircraft dynamics problems. First factor taken into account is the possibility of the change in plant parameters. This has been emulated in the previous system identification design by the presence of the unprecedented change in M_u during the transition. Now, the IAC is used to cope with the presence of this change. The result is shown in Fig. 6-18. The figure indicates that the IAC can handle the change in system parameter well. We note that this is the advantage of having the system identification that can capture all changes in the plant parameters. Further robustness test includes the time lag introduced in the system besides the change in parameter. The result is depicted in Fig. 6-19. It is shown that the IAC system can cope with change in M_u and time lag simultaneously. Figure 6-20 shows how the IAC system cope with the change in plant parameter together with noise in the sensor system. The noise causes a slight disturbance to the tracking performance especially when the change in M_u happens at $t = 50$ seconds. However the IAC system still works with only gracefully reduced performance.

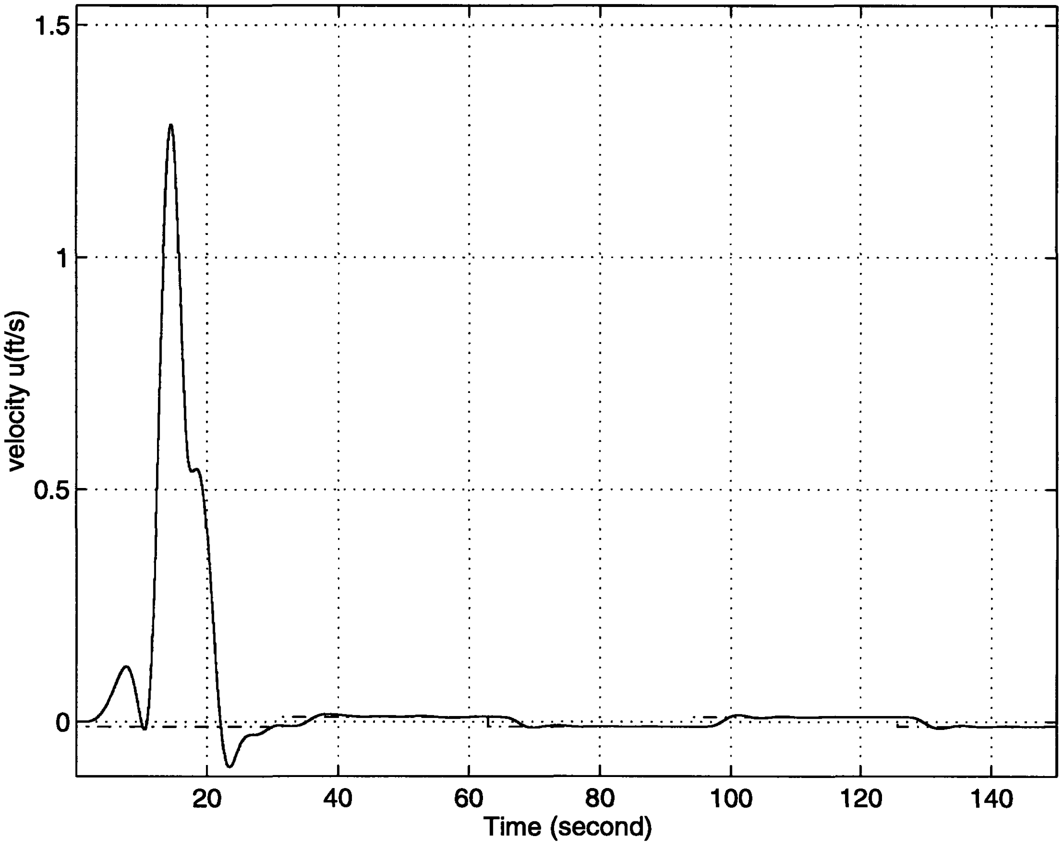
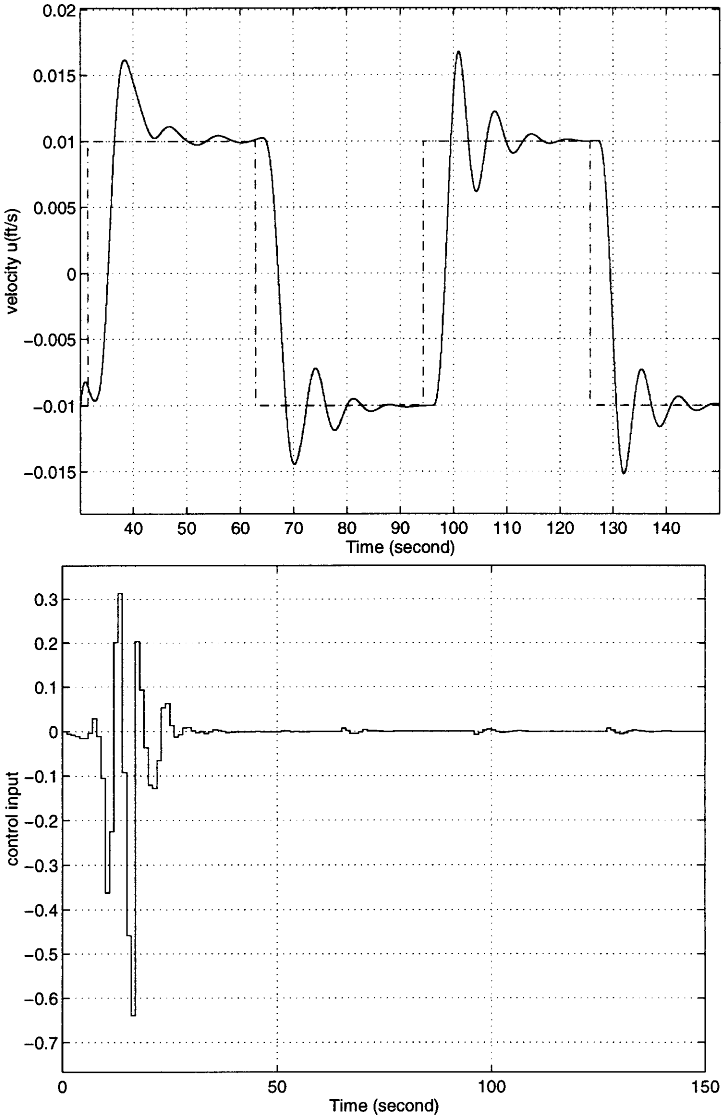
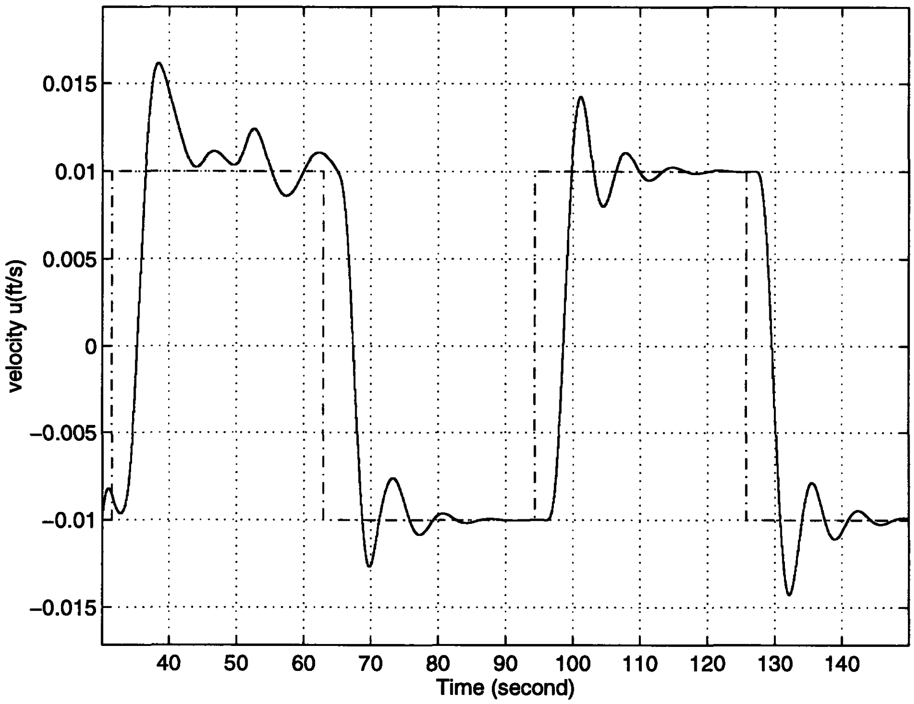


Figure 6-16: Result of IAC for VTOL during Transition



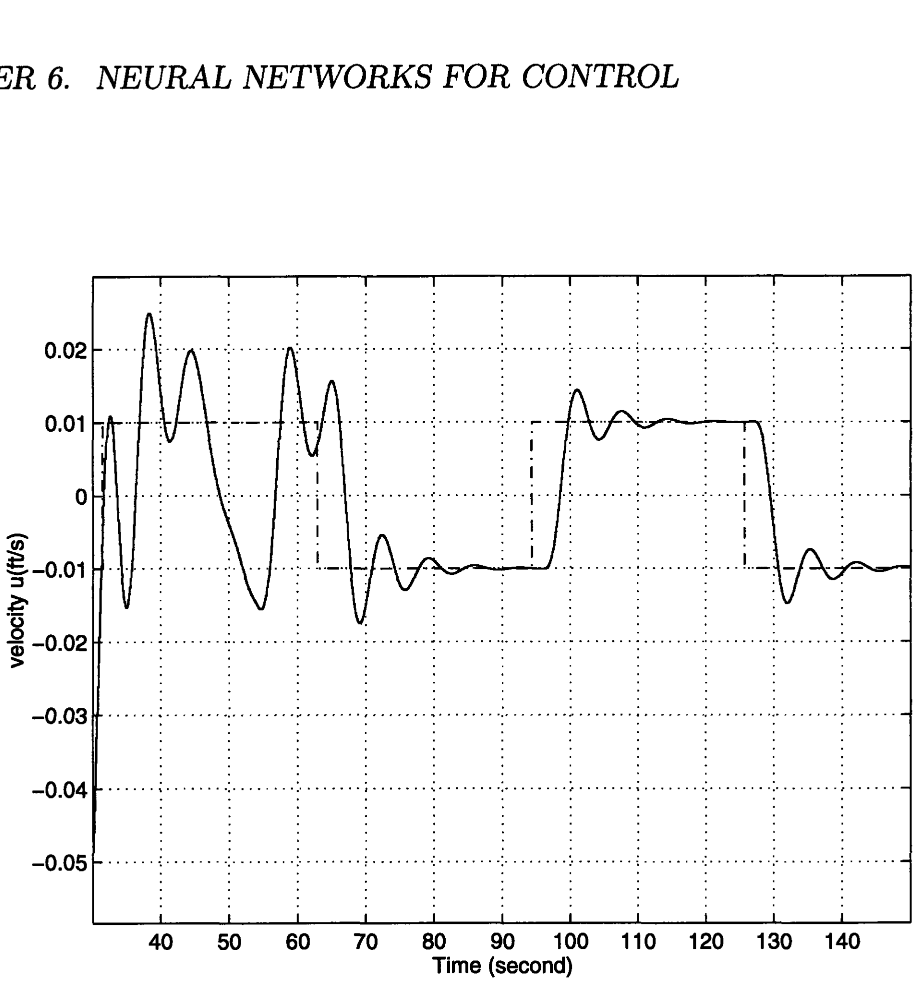
——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	IAC with ANN
Flight segment	
	Transition from hover to forward flight
Condition	
	No change in plant parameters No noise in sensor No time lag

Figure 6-17: IAC for VTOL during Transition



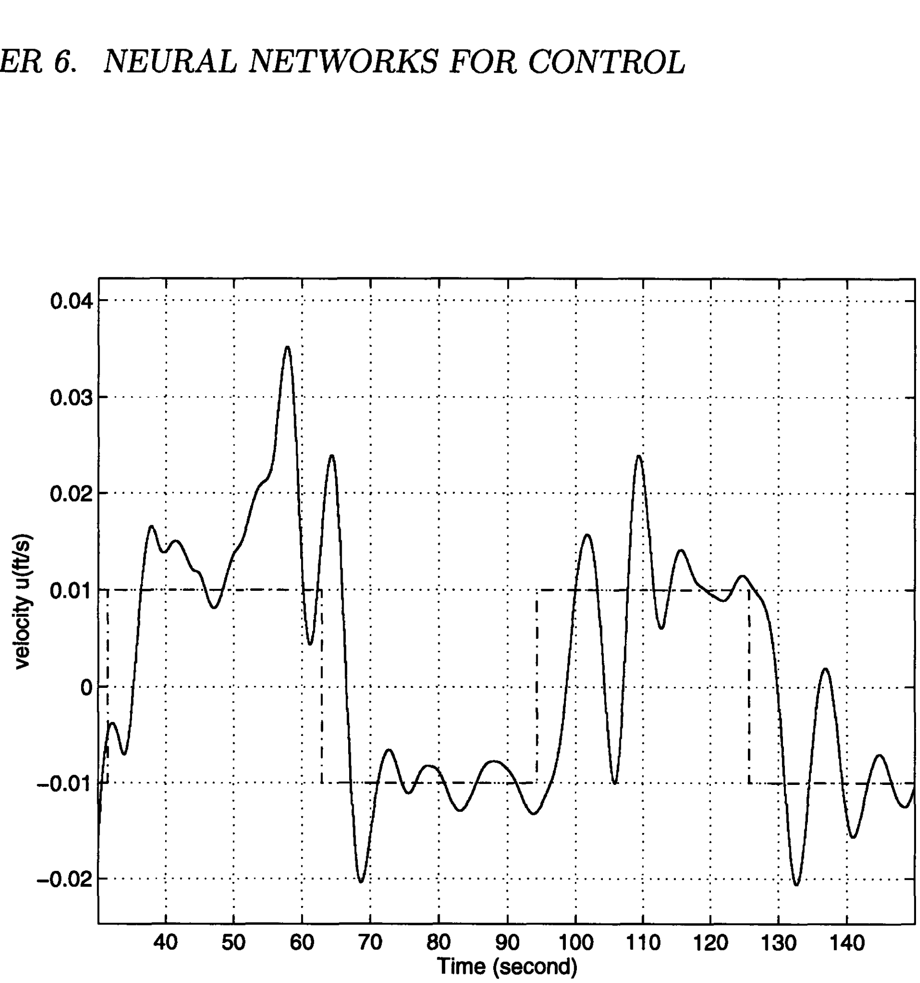
——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	IAC with ANN
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec. No noise in sensor No time lag

Figure 6-18: IAC for VTOL during Transition with change in M_u



———	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	IAC with ANN
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec. No noise in sensor Time lag introduced

Figure 6-19: IAC for VTOL during Transition with change in M_u and the presence of time lag



——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	IAC with ANN
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec. Added noise in sensor No time lag

Figure 6-20: IAC for VTOL during Transition with change in M_u and noise in sensor

6.3.3 Comparison with Conventional PI and Adaptive Controller

Performance Measures

To evaluate the advantages of IAC employing neural networks, the proposed scheme is compared with two other approaches. For our study, the comparison is made to conventional PI controller and adaptive controller. Note that there are many performance measures which effectively give a grade for how well the control system is performing. A list of possible functions is given below. The example functions listed are mean square integral evaluations.

Displacement Error Squared. This error is obviously of importance since it shows how well the system is tracking the reference command.

$$\epsilon = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (r - y)^2 \, dt \tag{6.15}$$

Reference Model Displacement Error. This is the requirement for the IAC type controller.

$$\epsilon = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (e_M)^2 \, dt \tag{6.16}$$

Ride Discomfort Error. This is the function that measures the ride discomfort, or similarly the fatigue loading on the system. These are clearly desirable to minimize in some applications (like AFCS for commercial airplanes) but are not important in others. The function expresses the integral square deviation of the load factor from 1.0.

$$\epsilon = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (1.0 - n_f)^2 \, dt \tag{6.17}$$

Control Energy. This is an important measures in particular for spacecraft systems and missile systems. Minimizing this value means that actuators are not used as much, so the control system is not using as much energy. Note that we use the square of time derivative of the control signal since the energy is proportional to the square of rate of displacement.

$$\epsilon = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (\dot{u}_c)^2 \, dt \tag{6.18}$$

Survival/Time Error. This class of error functions is somewhat specialized quantity. For instance, a pole balancing control system has an error based on how long it can stay upright, as well as the displacement error [33].

$$\epsilon = \int_{t_0}^{t_1} H(t)\, dt \tag{6.19}$$

$$H(t) = \begin{cases} 1 & \text{if pole upright} \\ 0 & \text{pole fallen} \end{cases} \tag{6.20}$$

Similarly, a missile control system could use an error signal based on how quickly it finds its target.

$$\epsilon = \int_{t_0}^{t_1} H(t)\, dt$$

$$H(t) = \begin{cases} 1 & \text{target lost} \\ 0 & \text{target found} \end{cases}$$

Control System Performance Comparison

This section contains the performance comparison between the control system schemes in term of two error measures. The function that measures the tracking performance and the function that gives the corresponding control power used are chosen as the error measures. We define them as ϵ_1 and ϵ_2:

$$\epsilon_1 = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (r - y)^2\, dt \tag{6.21}$$

$$\epsilon_2 = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} (\dot{u}_c)^2\, dt \tag{6.22}$$

The block diagrams of the PI and adaptive controller scheme are given in Fig. 6-21 and Fig. 6-22, respectively.

The result of simulation of PI controller and adaptive control scheme for different conditions are shown in Fig. 6-23 – Fig. 6-27. As shown in the Fig. 6-23 and Fig. 6-24, in general PI controller can not work with a time-varying system. The task of the control system is to drive the system output to follow a certain reference signal. The reference signal in this simulation is defined as the square wave signal with the frequency of 0.1 and an amplitude of 0.01. Even for the nominal case where

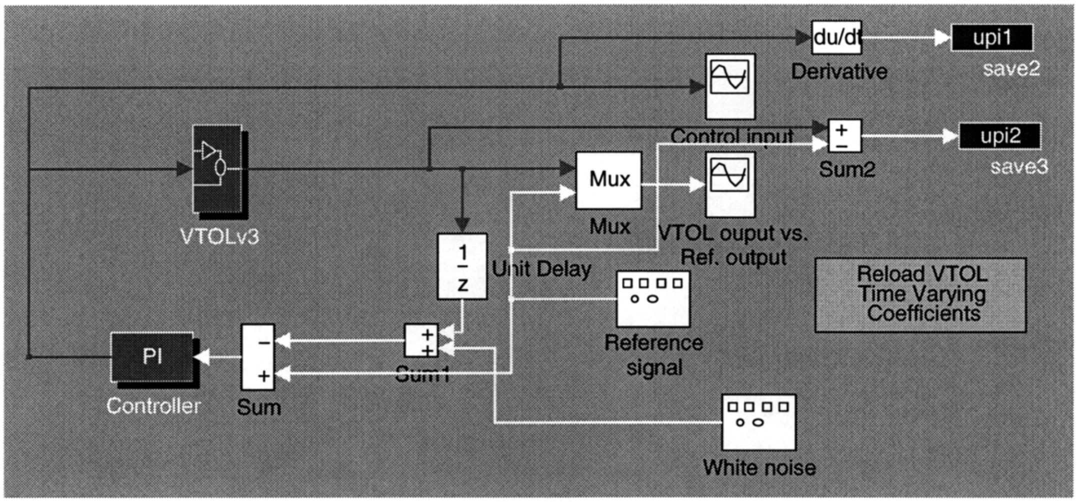


Figure 6-21: SIMULINK block diagram for PI controller

change in plant parameter, time lags, and noise are not introduced, PI controller exhibit unacceptably low tracking performance. The output of the plant fails to follow the square wave reference signal. The worse case happens when an unfavorable disturbance is presented into the system. Based on the simulation (see 6-24), PI controller drives the system to an unstable condition when change in plant parameter is introduced. Thus, the direct use of PI controller is not recommended for such a system. We have to note that this does not indicate that PI controller is not useful for controlling a time-varying system. The useful application will depend on how we introduce the algorithm for adjusting the gain of the PI controller. Ref. [7] is an example of how nonlinear neural network can be utilized for the auto-tuning PI(D) controller.

Fig. 6-25 – Fig. 6-27 presents the result of adaptive control simulation. The algorithm for the adaptive control simulation is given in Ref. [15]. The results indicate that the adaptive control can cope with time-varying system. The change in plant parameter as well as the time lag are handled successfully. The noise causes the tracking performance degradation whilst overall system is still performing adequately well. The comparison using error measures is summarized in Table 6.4 and 6.5 for nominal case and for case with the change in plant parameter M_u. With respect to IAC (using ANN) and adaptive control performance, the comparison suggests that despite the better tracking performance the adaptive control requires much higher control power.

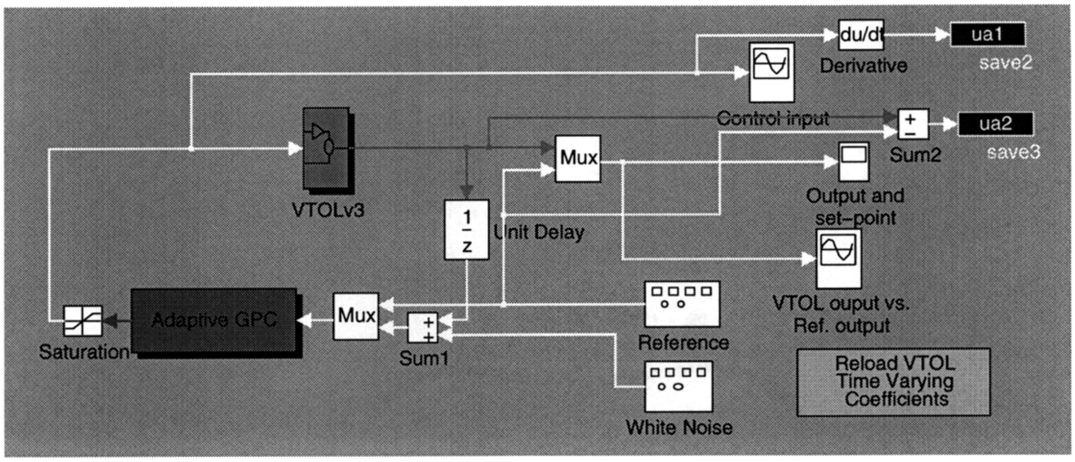


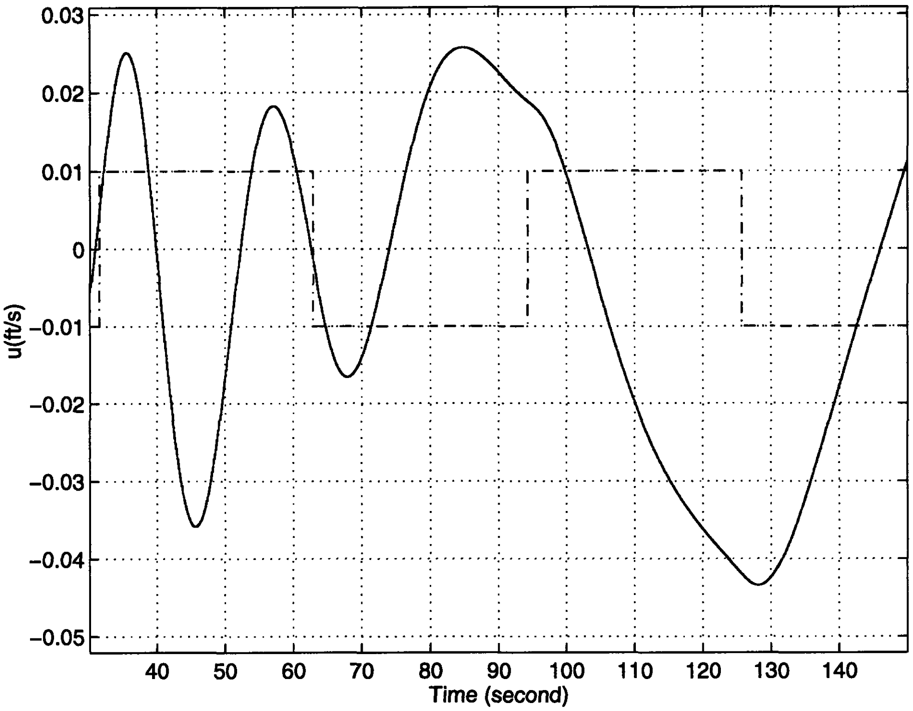
Figure 6-22: SIMULINK block diagram for adaptive control

Table 6.4: Error Comparison (No change in parameters)

Controller	ϵ_1	ϵ_2
PI	6.5571×10^{-4}	1.161×10^{-7}
Adaptive Control	3.7772×10^{-5}	309.2974
IAC with ANN	4.8433×10^{-5}	0.0926

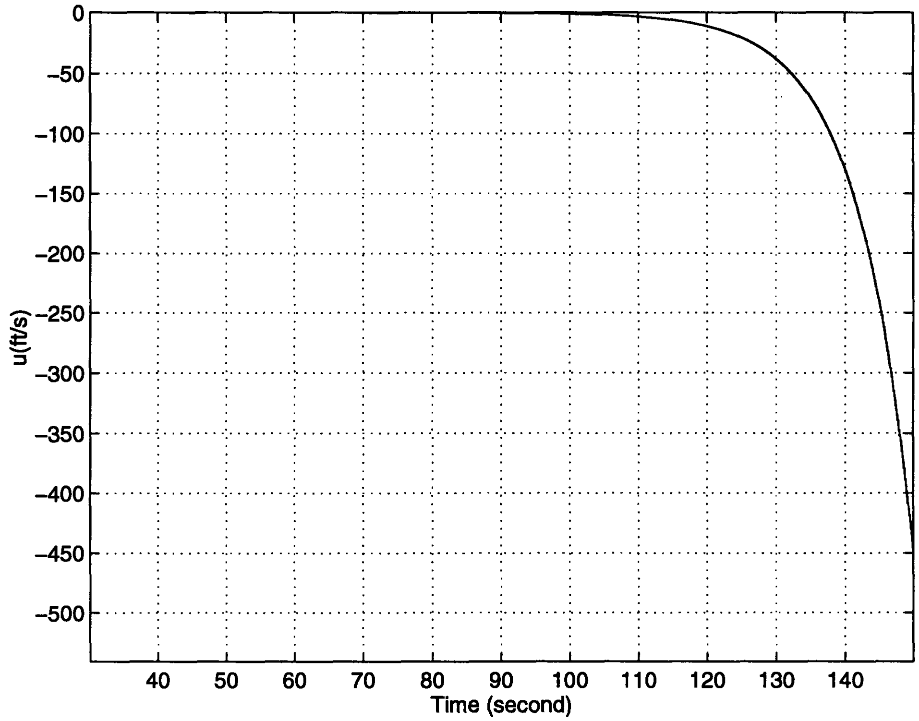
Table 6.5: Error Comparison (change in M_u)

Controller	ϵ_1	ϵ_2
PI	6.9355×10^{3}	0.0812
Adaptive Control	3.7528×10^{-5}	309.2976
IAC with ANN	4.9017×10^{-5}	0.0926



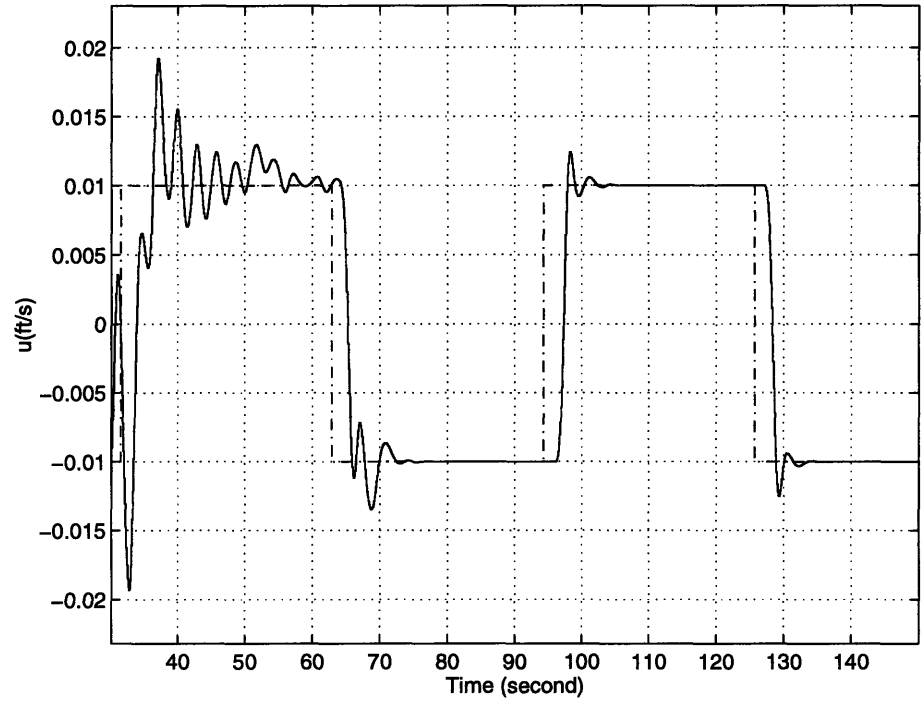
——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	PI Control
Flight segment	
	Transition from hover to forward flight
Condition	
	No change in plant parameters No noise in sensor No time lag

Figure 6-23: PI Control for VTOL during Transition



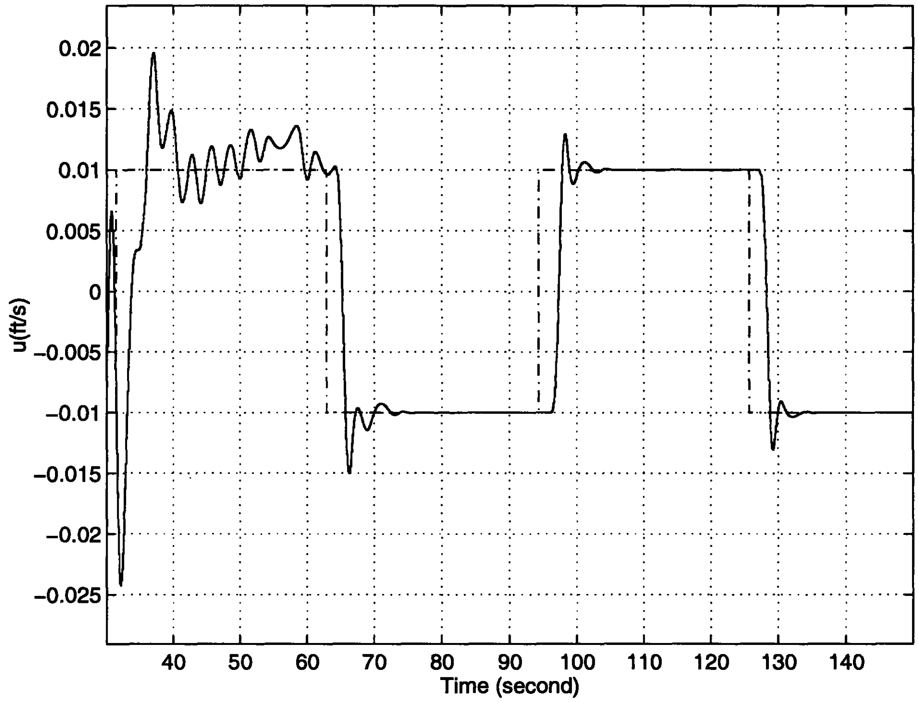
——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	PI Control
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec.
	No noise in sensor
	No time lag

Figure 6-24: PI Control for VTOL during Transition with change in M_u



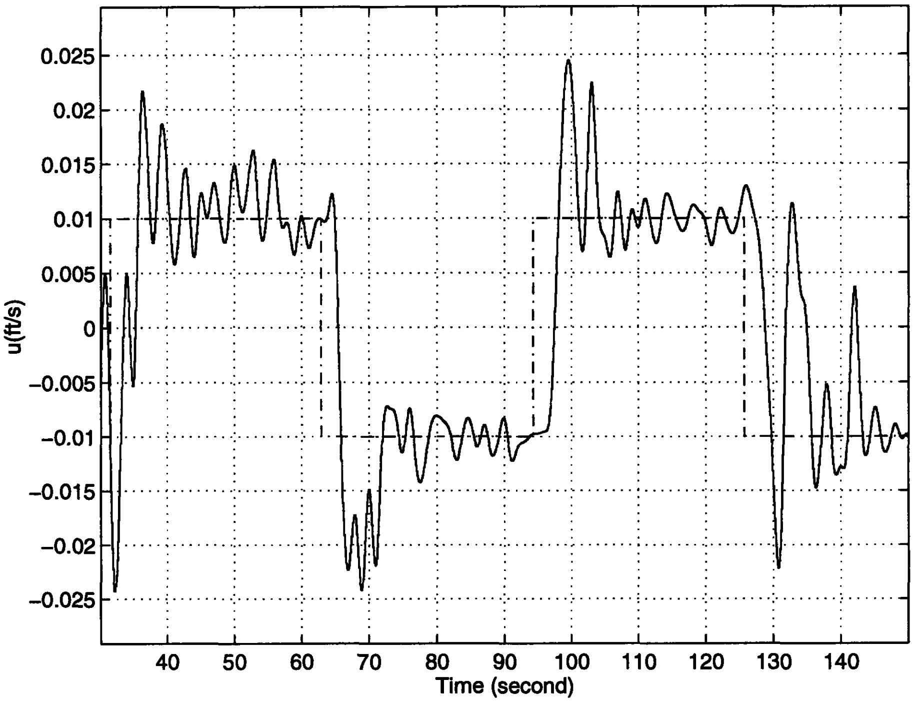
——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	Adaptive Control
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec. No noise in sensor No time lag

Figure 6-25: Adaptive Control for VTOL during Transition with change in M_u



——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	Adaptive Control
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec. No noise in sensor Time lag

Figure 6-26: Adaptive Control for VTOL during Transition with change in M_u and time lag



——	≡ VTOL output
— · — · —	≡ Reference signal
Flight Vehicle	
	XC-142 VTOL Aircraft
AFCS	
	Adaptive Control
Flight segment	
	Transition from hover to forward flight
Condition	
	Change in M_u at $t = 50$ sec. Noise in sensor No time lag

Figure 6-27: Adaptive Control for VTOL during Transition with change in M_u and noise in the sensor system

Chapter 7

Discussion and Conclusions

7.1 Discussion and Conclusions

7.1.1 Time-varying System

The study of flight vehicle dynamics is in general involved with the problem of non-linear time-varying differential equations. Linearization procedures justified by *Poincarè - Lyapunov* theorem are usually used, resulting in a simpler system as an acceptable representation of the original system. If the system is time-varying, however, the dynamics analysis must be conducted carefully as this type of system exhibits some peculiar behaviors not present in the linear time invariant case. Some of the most distinct feature is that the location of the characteristic roots does not necessarily describe the stability of the system. Some time-varying systems might also exhibit a turning point phenomenon. This is characterized by a drastic change in the topology of the solution of the corresponding differential equation.

The differential equation of an LTV system does not, in general, have an exact solution. The approximation to the exact solution can be done analytically and numerically. In practice, these two approaches have their own advantages and disadvantages. The analytical solution gives a better understanding of the related physical phenomena but maybe mathematically demanding. The numerical method, thanks to the rapid development of the number-crunching computer can give a quick and easy approximation to the solution. However this technique is not related directly to the physical understanding of the problem. We should note here that there has been some effort to combine the advantage of the analytical approach and the numerical method. For time-varying and nonlinear systems, the generalized multiple scales (GMS) method [28] is a good example of such an approach.

7.1.2 GHAME Vehicle Dynamics

The GHAME vehicle model provides a realistic representation of an advanced flight dynamic problem. The reentry problem in particular exhibits a rich dynamic phenomenon where many factors such as thermal effects and aerochemistry are involved. In all reentry modes, there are two overriding technical concerns — maximum deceleration and aerodynamic heating. With regard to reentry of manned space vehicle there is an extra consideration. The vehicle must follow a certain trajectory between *overshoot* boundary where it will go shooting past back into outer space; and *undershoot* boundary where maximum deceleration will be too large. Consequently, there is a narrow *reentry corridor* into which the vehicle must be guided for a successful return to the earth's surface.

For an analysis of a space vehicle, the reentry problem can be broken into two parts. First is the motion of a center of mass of the vehicle following a certain boundaries as described above. This is often treated as an optimization problem. The second part is studying the stability of the flight vehicle as it moves along the optimized trajectory. In this work, the reentry dynamics was examined along a trajectory which was originally designed to minimize the thermal-protection-system (TPS) weight of the Space Shuttle Orbiter 049 vehicle [27].

Second order angle-of-attack perturbation model was developed using an appropriate transformation of variable to the equation of motion. The original equation of motion is simplified into a linear variable coefficient differential equation. The variable coefficient is a function of GHAME vehicle geometry and trajectory parameters and aerodynamics stability derivatives. The trajectory parameters are given as the variation of angle-of-attack α, flight path angle γ, flight speed V and altitude h. And aerodynamic derivatives are given as a function of two variable, Mach number M and α. The stability analysis is done by root locus technique and time response analysis. A pair of conjugate roots move from locus near zero to the left half plane. The root locus construction indicates that as the vehicle moves along the trajectory, the frequency and the damping of the motion increase. A noticeable increase occurs in the early phase of the trajectory. The time response analysis confirms that the perturbation variable α oscillation has an increasing frequency at the early stage. The sensitivity analysis is carried to observe the effect of aerodynamic parameter to the longitudinal dynamics. The results show that the longitudinal dynamics is most sensitive to C_{m_q} and C_{m_α}. The GHAME vehicle sensitivity to variation of those two parameters was found to be 14 times as high as change in C_{L_α} and 5000 times as high as variation in C_{D_α}. Variation of C_{L_0} and C_{D_0} was found to be negligible. In the analysis of stability and control, greater attention must be given to the most influential parameter such as C_{m_q} and C_{m_α} since they affect the stability more than any other parameters.

A fourth order longitudinal and lateral-directional reentry dynamics were also analyzed. The approximate solution is predicated on the assumption that the flight conditions are slowly varying. The root locus configuration for the longitudinal case indicates that there is a conjugate roots around zero and there are two real roots in the left half plane. The conjugate roots represent a phugoid mode, characterized by lightly damped low frequency motion. Along the trajectory, this frequency increases as seen by the roots moving upward (downward). The time response shows the stable oscillation with a slight frequency increase in the early stage.

The lateral-directional reentry dynamics was predicated on the same approach as its longitudinal counterpart. The root locus diagram shows that in the early phase of the trajectory there is a pair of conjugate roots located in the left half plane. In the standard flight dynamics analysis, this pair of roots corresponds to the *dutch-roll* mode which is characterized by high frequency and lightly damped motion. The other roots are real and are located near zero and in the right half plane, respectively. As the GHAME vehicle transverses along the trajectory, the *dutch-roll*-like roots move further to the left with substantial increase in frequency. The root near zero moves to the left half plane but the other one moves further to the right resulting in unstable behavior. The time response verifies that the lateral-directional motion of GHAME vehicle is unstable.

7.1.3 VTOL Aircraft Dynamics

Like the problem of reentry dynamics, the VTOL aircraft dynamics exhibit some distinct phenomena. The vehicle flight involves two different segments: hover (like a helicopter) which is dynamically unstable and the forward flight which is stable. In this work, the dynamics of XC-142 aircraft based on a prior study at Princeton was examined. The stability derivatives appearing in the perturbation equation are assumed to be a function of the flight velocitiy V. The analysis was done for two different forms of velocity variation and stability derivative M_w variation.

In the 2 degree of freedom approximation, the *plunging* or vertical motion is suppressed and the resulted equation for drag and moment can then be decoupled. The approach leads to the independent perturbation equation for variable u and θ. The differential equations have a variable coefficient mainly as a function of stability derivative and flight velocity. The velocity is given as a function of time, therefore the final representation of the perturbation equation is a third order time-varying differential equation. The root locus diagram for both perturbation variables reveal that the trajectory begins with unstable motion (hover) indicated by a pair of unstable roots. As the time increases, the unstable roots move quickly to the left half plane. The time response analysis justifies that in the early phase of the trajectory, the motion is unstable. After approximately 60 sec. the divergent motion ceases and the system exhibits stable behavior. In principle the result shows how the VTOL dynamics behave during transition from hover to the forward flight.

The complete 3 degree of freedom analysis involves the intricate *turning point* phenomenon. For simplicity, only the perturbation variable u was examined. The perturbation equation is in the form of 4th order time-varying differential equation. The root locus diagram displays the turning point phenomenon i.e. there is change in the topology of the solution of the dynamic system. Such phenomenon occurs when there are multiple characteristic root. This happens around $t = 12$ seconds.

7.1.4 The Use of Neural Networks

Judging from its increasingly popular use in the realm of dynamic and control, neural network is intended to be used as system identification for the system under study. In this work, a static type of network is considered to be a precursor of further study. This implies that the learning process of the dynamic system is done *off-line*. In this case the data for the training is first obtained via simulation of the vehicle model. Upon gaining the data, the network is then trained using selected input and output pairs called *patterns*. During training the neural network must predict the behavior of the true system. The test is then done to observe the network performance in learning the system dynamics. In general the network shows an accurate prediction of the vehicle dynamics. However, the test by changing the plant parameters, indicates that the trained network can only cope with a slight uncertainty (below 10%). For the presence of a higher uncertainty in the system a different training technique is necessary. It can be said that in general *on-line* type of learning will be necessary to handle the system which is not only time-varying but also contains uncertainties.

The online learning scheme design is presented in Chapter 7. It is motivated by the need to have a system identification that can capture all possible changes in the system. These changes may stem from disturbances affecting the plant, improper plant model structure and a change in plant parameters. The design of the neural network system identification includes the investigation of the networks structure (number of hidden neuron), class of activation function and learning rate parameter. The design is concluded with the neural network model with the optimum number of sigmoidal hidden neuron with moderate learning rate of 0.2. Based on the simulation results, the neural network model is performing well after only a few second in operation without any prior off-line training.

The IAC was designed using the available neural network model. The results indicate that the IAC can cope with various realistic problem including plant parameter changes, time lags, and noise. Comparison with two different approach (PI and adaptive control) was based on two error measures each corresponds to the tracking performance and the required control power. Overall comparison suggests that IAC

Table 7.1: Some of advantages and disadvantages of using ANN for AFCS

Advantages	Disadvantages
Noise and disturbances robustness	Not well understood
Nonlinear capability	No proven closed loop performance
'Intelligent' control possibilities	'New' technology
Faster computation due to paralellity	May require a lot of simulation
Fault tolerance capability	Lack of design tools
Simple, easy integration	
'Natural' control	

system exhibits more favorable performance reflected by the adequate tracking performance with a low required control power. We have to note that serial simulation of the neural networks did not allow exploitation of the speed benefits, but a flexibility in software and ease of integration was enjoyed.

With respect to AFCS, Artificial Neural Networks are so versatile and interesting that they undoubtedly have found many applications. For current using this approach, there are both advantages and disadvantages that we have to consider. Some of them are listed in Table 7.1.

7.2 Recommendations for Further Work

The following is suggested as the area of further research pertaining neural network application and the analysis of flight vehicle in general.

- Investigate the 4th order model of the GHAME vehicle using direct integration of the nonlinear time-varying equation. This will give a true behavior of the system.

- Determine the handling qualities of the GHAME vehicle and VTOL aircraft.

- Investigate other control algorithm using Neural Networks for an AFCS such as internal model control (IMC), self-tuning controller and heuristic reference adaptive critic (HRAC)

- Determine fault tolerance of neural networks

- Investigate ways of using neural networks in BITE and FDIE systems

Bibliography

[1] J. J. D'AZZO & C. H. HOUPIS, *Linear Control System Analysis and Design*, McGraw-Hill, 4th edition, 1995.

[2] J.A. FREEMAN & D.M. SKAPURA , *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison Wesley Publishing Co., USA.

[3] K.J. HUNT, D. SBARBARO, R. ZBIKOWSKI AND P.J. GAWTHROP, *Neural Networks for Control Systems—A Survey*, IFAC conference, 1992

[4] J.-S. R. JANG, C.-T. SUN AND E. MIZUTANI, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Inc., New Jersey, 1997

[5] B. SŏUCEK AND M. SŏUCEK, *Neural and Massively Parallel Computer*, Wiley, 1988.

[6] BART KOSKO, *Neural Networks and Fuzzy Systems.* A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, Englewoods Cliffs, 1992.

[7] M.J. WILLIS AND G.A. MONTAGUE , *Auto-Tuning PI(D) Controllers with Artificial Neural Networks*, Procedding of IFAC Conference, Sydney, 1993

[8] P. J. WERBOS, *Backpropagation Through Time: What It Does and How To Do It*, Proceeding of the IEEE, Vol. 78, No. 10, 1990

[9] EDWARD DEBONO, *I am Right You Are Wrong*, Penguin Books, Great Britain, 1990

[10] JUDITH E. DAYHOFF, *Neural Network Architecture: An Introduction*, Van Nostrand Reinhold, 1990.

[11] J. NIE AND D. LINKENS, *Fuzzy-neural Control.* Principles, Algorithm and Applications, Prentice Hall, UK, 1995.

[12] PHILIP DALE HEERMANN, *Neural Network Techniques for Stable Learning Control of Nonlinear System*, PhD Thesis, The University of Texas at Austin, 1992.

[13] R. HECHT-NIELSEN, *Neurocomputing*, Addison-Wesley, USA, 1990.

[14] H. DEMUTH AND M. BEALE, *Neural Networks Toolbox*, User's Guide, The MathWorks Inc., MA, 1994

[15] J. MOŚCIŃSKI AND Z. OGONOWSKI, *Advanced Control with Matlab & Simulink*, Ellis Horwood Ltd., 1995

[16] M. T. HAGAN, H. B. DEMUTH AND M. BEALE, *Neural Networks Design*, PWS Publishing Company, MA, 1996

[17] K. S. NARENDRA, *Neural Networks for Control: Theory and Practice*, Proceeding of the IEEE, Vol. 84, No. 10, Oct., 1996

[18] A.H. BOWERS ET.AL. , *A Generic Hypersonic Aerodynamic Model Example (GHAME) for Computer Simulation*, A Proposed NASA TM, Dryden Flight Research Facility, Edwards, California, August 5, 1988

[19] HANS R. FRIEDRICH AND FRANK J. DORE, *The Dynamic Motion of a Missile Descending Through the Atmosphere*, Journal of the Aerospace Sciences, Vol 22, No. 9, Sept. 1955, pp. 628 - 632,638

[20] W.H.T. LOH, *A Second Order Theory of Entry Mechanics into a Planetary Atmosphere*, Journal of Aerospace Sciences, Vol. 29, 162, pp. 1210 - 1221, 1237

[21] JOHN D. ANDERSON, *Introduction to Flight*, 3rd Edition, Mc. Graw-Hill, New York, 1989

[22] JOHN E. ALLEN, *Aerodynamics:* The science of air in motion, 3rd Edition, Allen Brothers and Father, Suffolk, 1982

[23] B. ETKIN, *Dynamics of Atmospheric Flight*, Wiley, New York, 1972

[24] BERNARD ETKIN AND LLOYD D. REID, *Dynamics of Flight: Stability and Control*, 3rd Edition, Wiley and Sons, New York, 1996

[25] ARNOLD M. KUETHE AND CHUEN-YEN CHOW *Foundations of Aerodynamics:* Bases of Aerodynamic Design, 3rd Edition, John Wiley & Sons, 1976

[26] N.X. VINH AND E.V. LAITONE, *Longitudinal Dynamic Stability of Shuttle Vehicle*, The Journal of the Aeronautical Sciences, Vol. XIX, No 5, pp. 337 - 363, March - April, 1972

[27] RUDRAPATNA V. RAMNATH AND PRASUN SINHA , *Dynamics of the Space Shuttle during Entry into Earth's Atmosphere*, AIAA Journal, Vol. 13, No. 3, March 1975

[28] R.V. RAMNATH AND G. SANDRI, *A Generalized Multiple Scales Approach to a Class of Linear Differential Equations*, Journal of Mathematical Analysis and Applications 28, 339 - 364, 1969

[29] R.V. RAMNATH, *Advanced Spacecraft Dynamics and Control*, Lecture Notes, Massachusetts Institute of Technology, MA, 1996

[30] RUDRAPATNA V. RAMNATH , *Transition Dynamics of VTOL Aircraft*, AIAA Journal, 1972

[31] M. ABRAMOWITZ AND IRENE A. STEGUN, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, NBS, Washington, 1964

[32] F. B. HILDEBRAND, *Advanced Calculus for Applications*, 2nd. Ed., Prentice-Hall Inc., NJ, 1976

[33] J. M. WHARINGTON, *Neural Networks for Aircraft Flight Control System*, B-Eng Thesis, Volume I, RMIT, 1992.

Appendix A

Aerodynamics Coefficients 2-D Data

A.1 Longitudinal Aerodynamics Coefficients

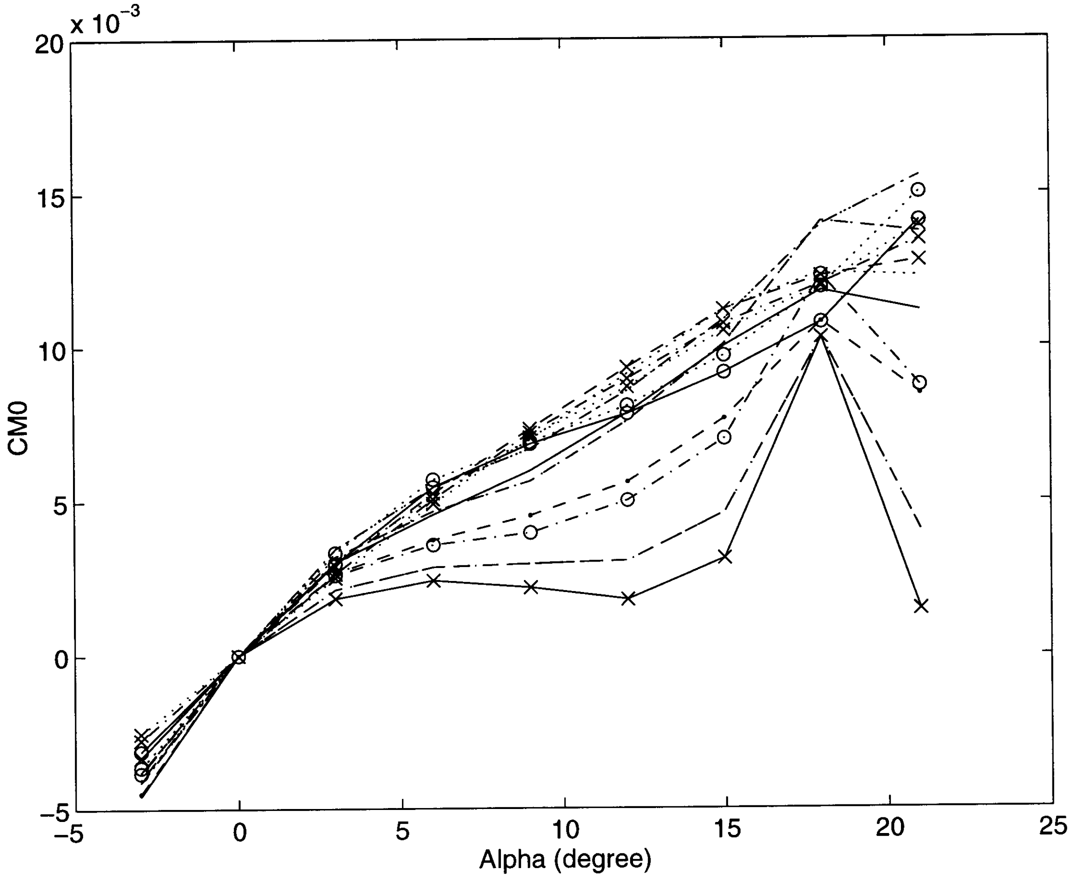


Figure A-1: Aerodynamics coefficient C_{M0} as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
··· o ···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
··· × ···	≡	$M = 24$			

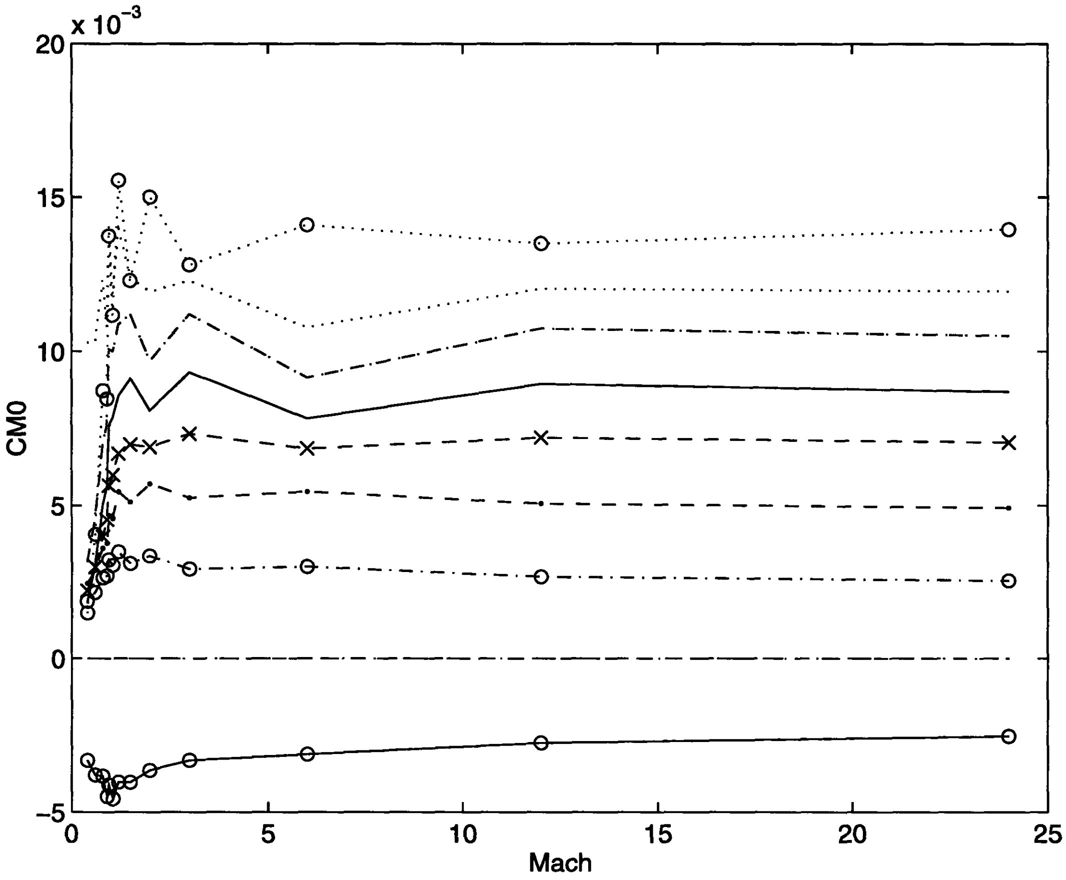


Figure A-2: Aerodynamics coefficient C_{M0} as a function Mach number M

—o—	$\equiv$	$\alpha = -3^o$	– –.	$\equiv$	$\alpha = 0^o$
—.o–	$\equiv$	$\alpha = 3^o$	– ● –	$\equiv$	$\alpha = 6^o$
.–×–	$\equiv$	$\alpha = 9^o$	——	$\equiv$	$\alpha = 12^o$
–.–.	$\equiv$	$\alpha = 15^o$	· · ·	$\equiv$	$\alpha = 18^o$
· · · o· · ·	$\equiv$	$\alpha = 21^o$			

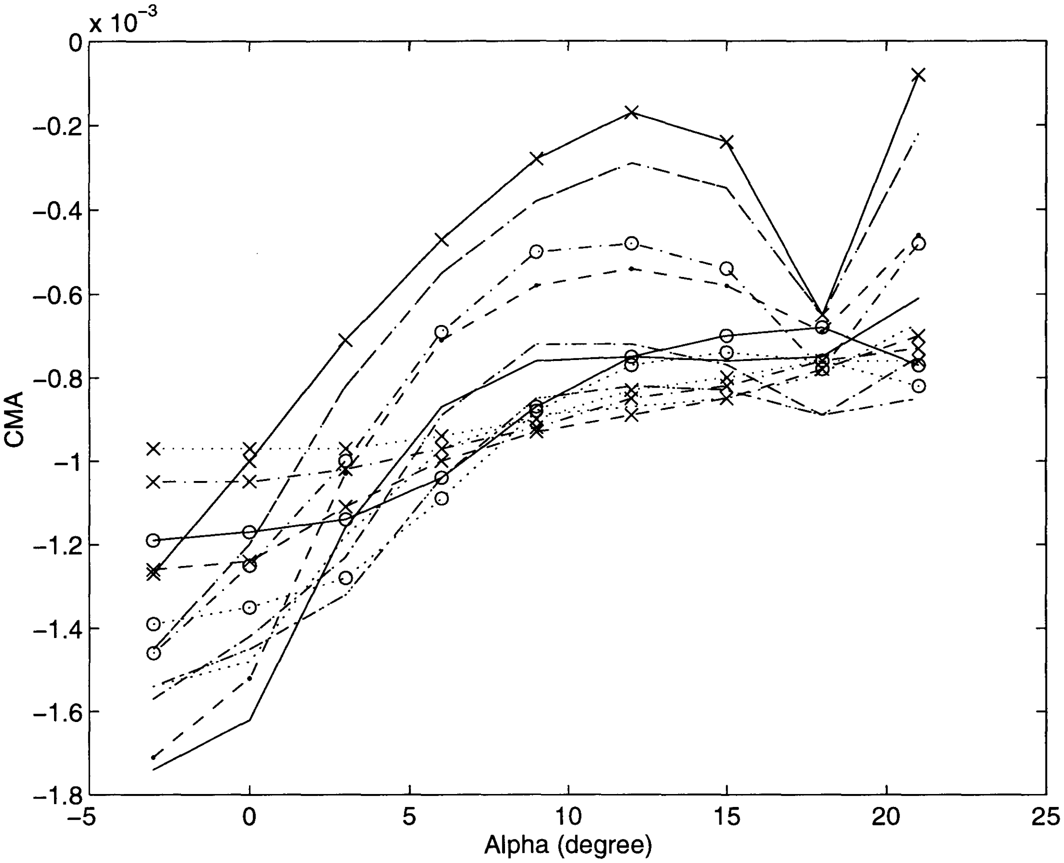


Figure A-3: Aerodynamics coefficient C_{MA} as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.○–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	......	≡	$M = 1.5$
···○···	≡	$M = 2$	–×–×–	≡	$M = 3$
—○—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

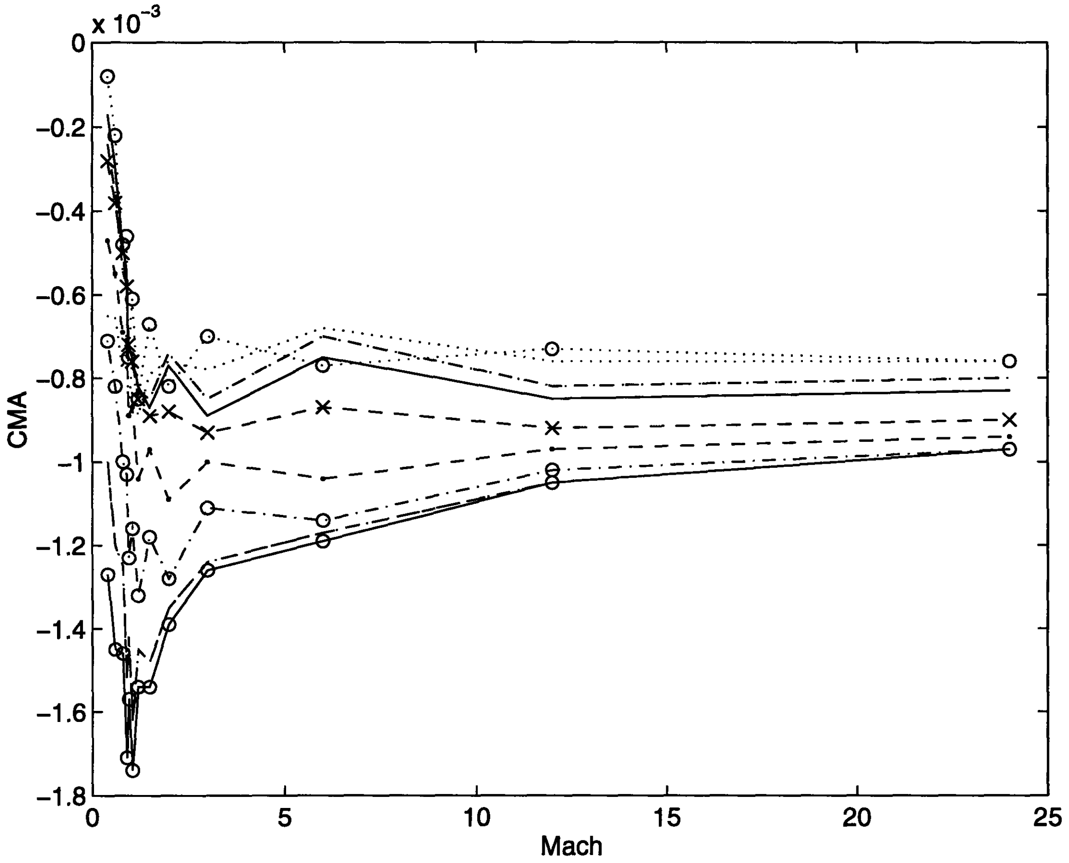


Figure A-4: Aerodynamics coefficient C_{MA} as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

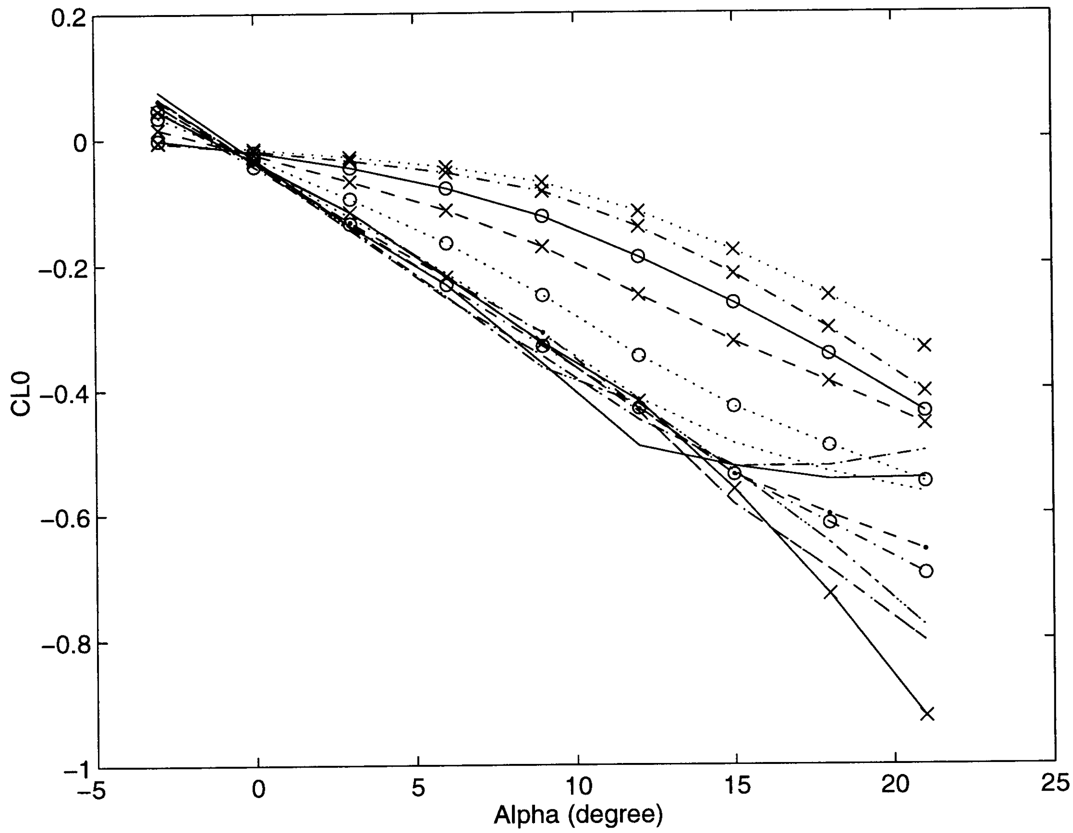


Figure A-5: Aerodynamics coefficient C_{L0} as a function of angle of attack α

—×—	$\equiv$	$M = .4$	– –.	$\equiv$	$M = .6$
—.o–	$\equiv$	$M = .8$	– ● –	$\equiv$	$M = .9$
–.–.	$\equiv$	$M = .95$	——	$\equiv$	$M = 1.05$
–..–	$\equiv$	$M = 1.2$	······	$\equiv$	$M = 1.5$
···o···	$\equiv$	$M = 2$	–×–×–	$\equiv$	$M = 3$
—o—	$\equiv$	$M = 6$	–.×–.	$\equiv$	$M = 12$
···×···	$\equiv$	$M = 24$			

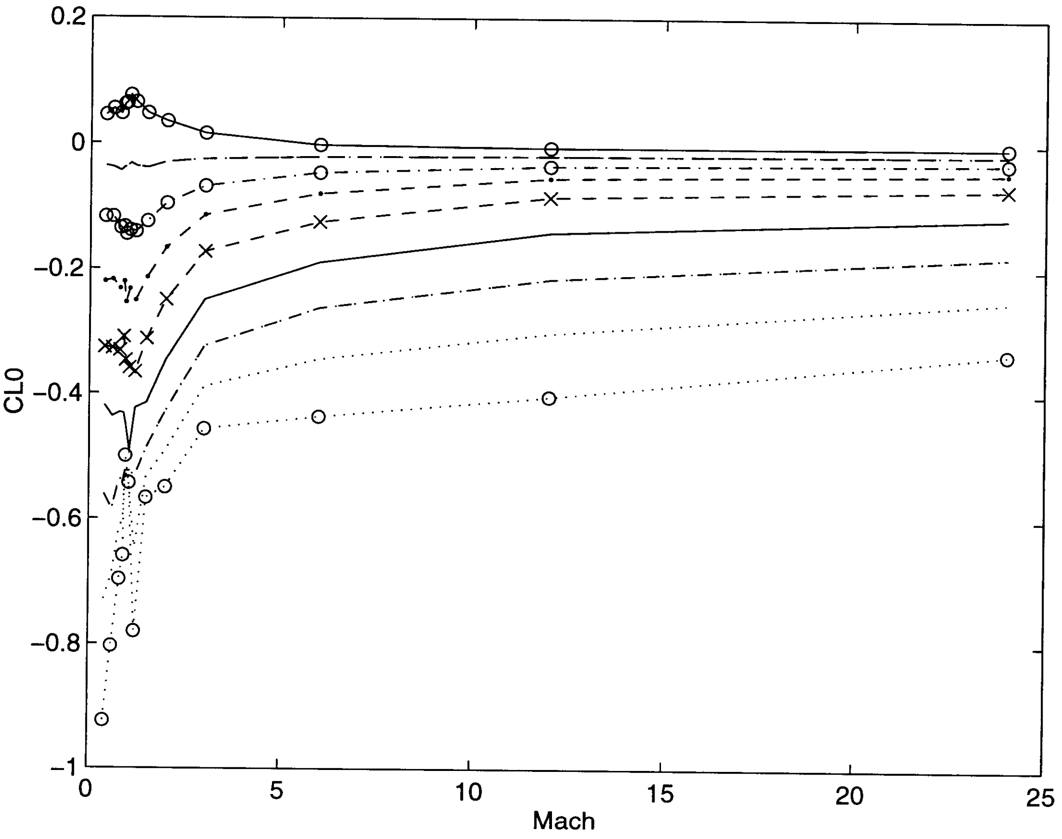


Figure A-6: Aerodynamics coefficient C_{L0} as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	———	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

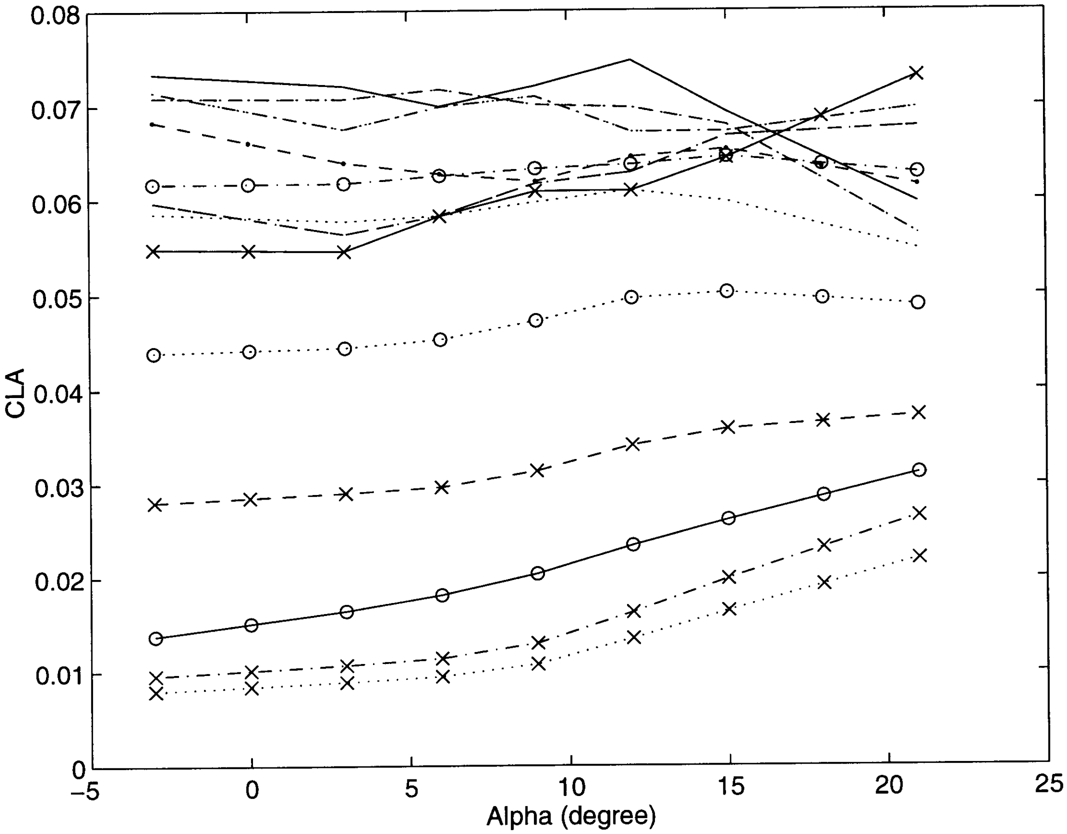


Figure A-7: Aerodynamics coefficient C_{LA} as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.○–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···○···	≡	$M = 2$	–×–×–	≡	$M = 3$
—○—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

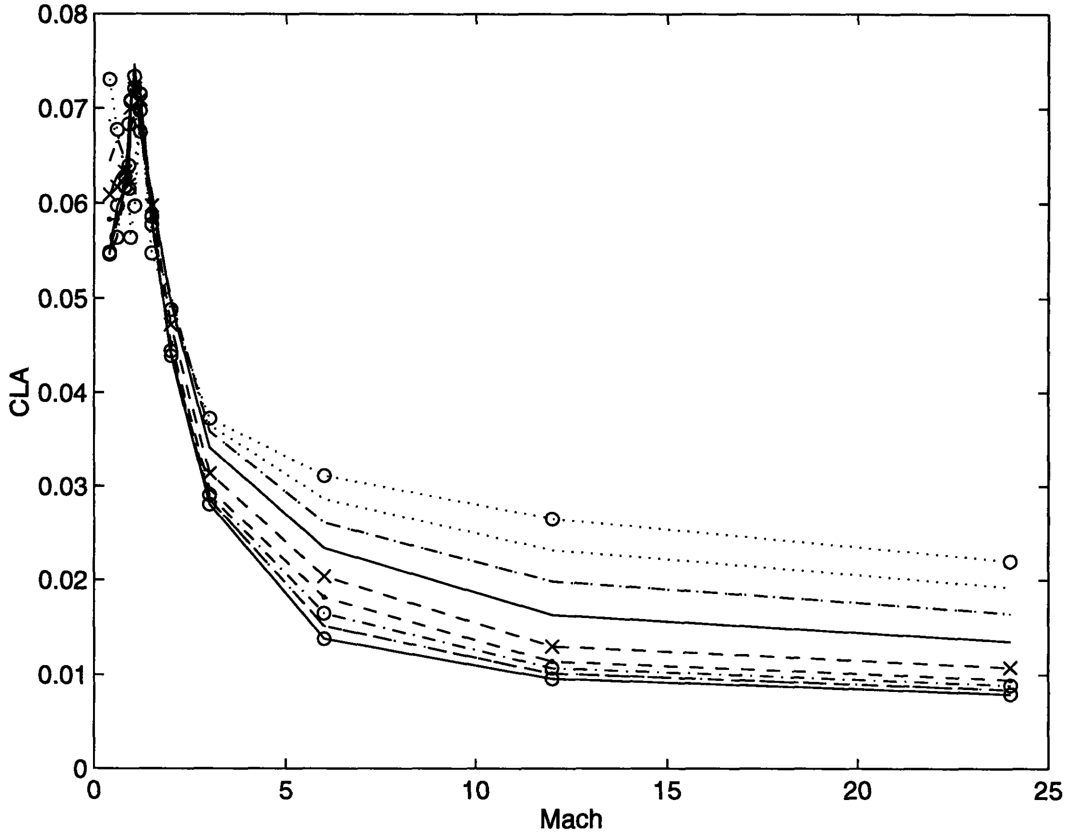


Figure A-8: Aerodynamics coefficient C_{LA} as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

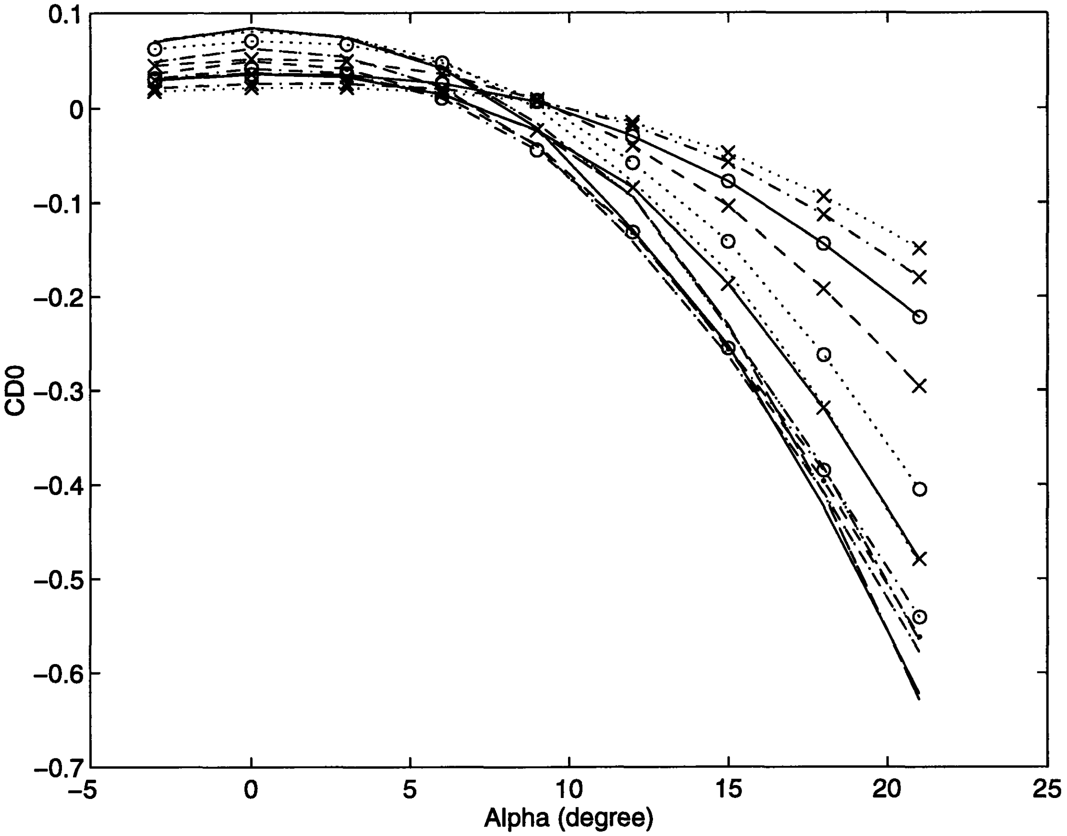


Figure A-9: Aerodynamics coefficient C_{D0} as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

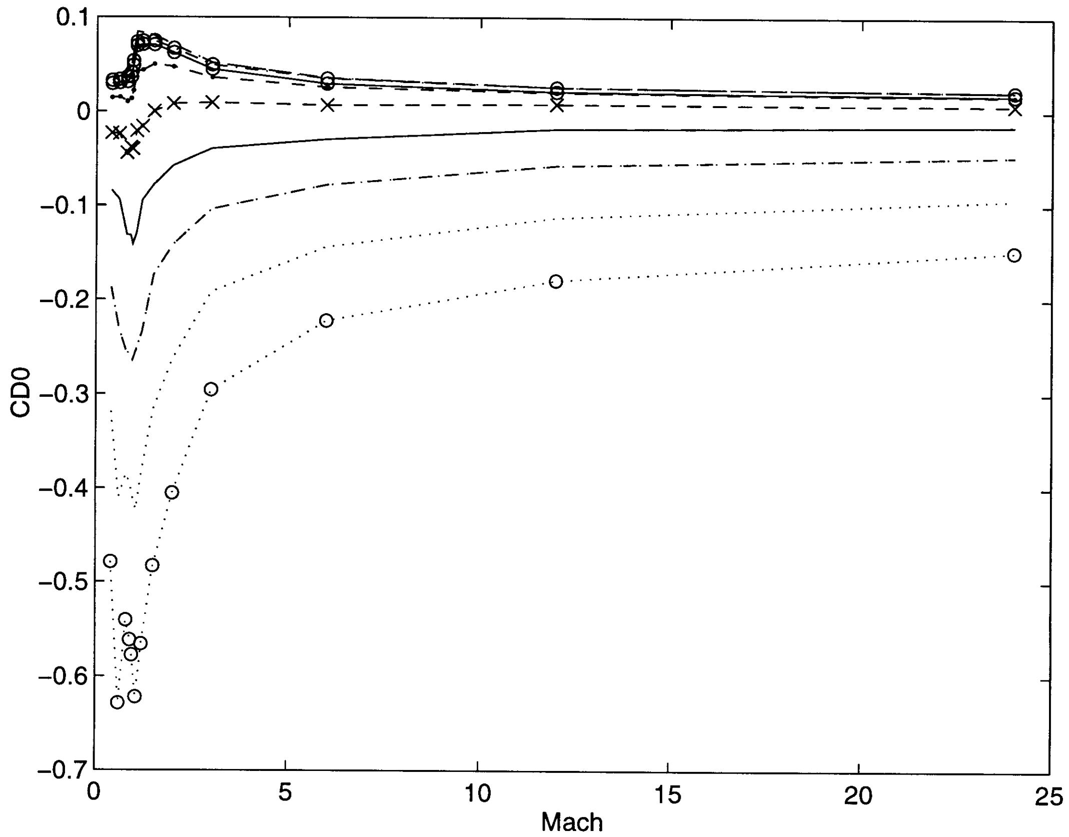


Figure A-10: Aerodynamics coefficient C_{D0} as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	———	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

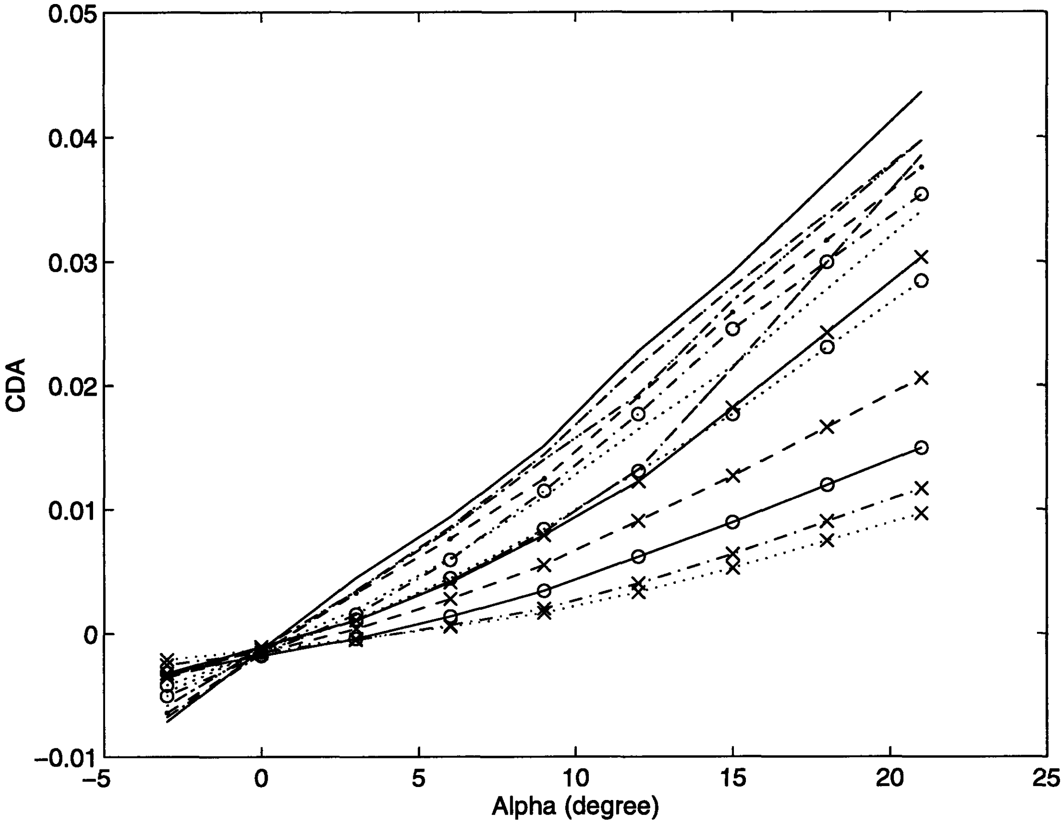


Figure A-11: Aerodynamics coefficient C_{DA} as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

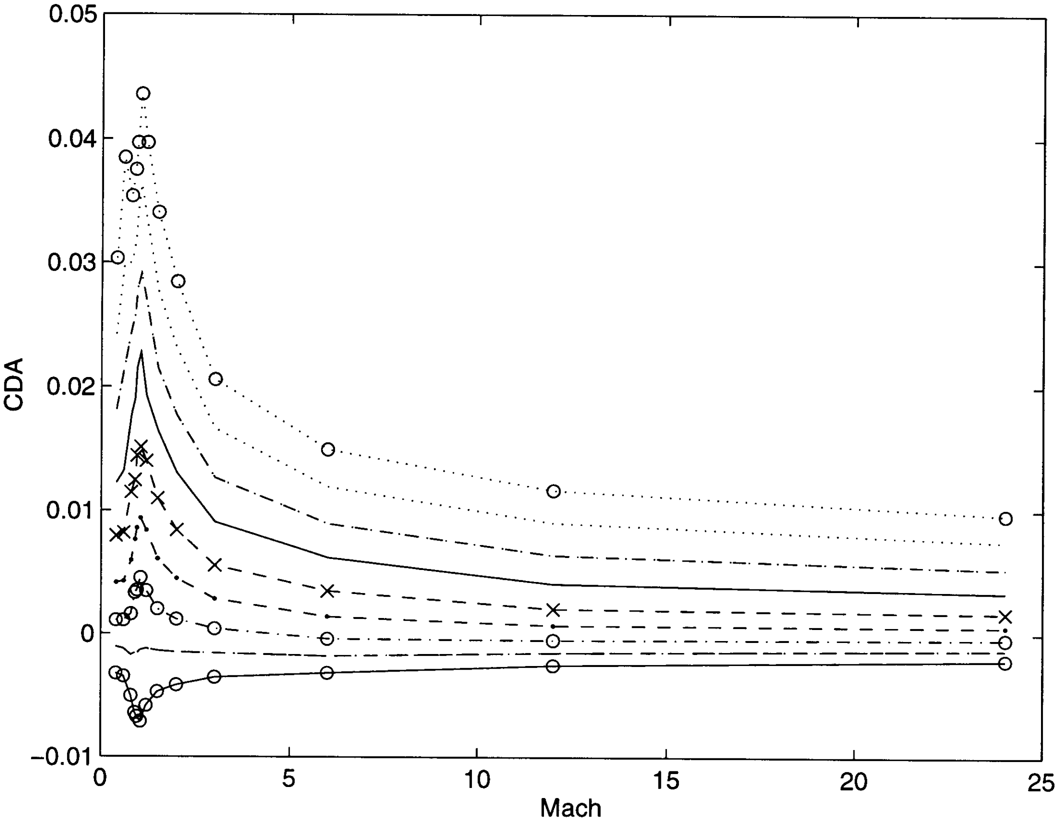


Figure A-12: Aerodynamics coefficient C_{DA} as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	· · ·	≡	$\alpha = 18^o$
· · · o· · ·	≡	$\alpha = 21^o$			

A.2 Lateral Aerodynamics Coefficients

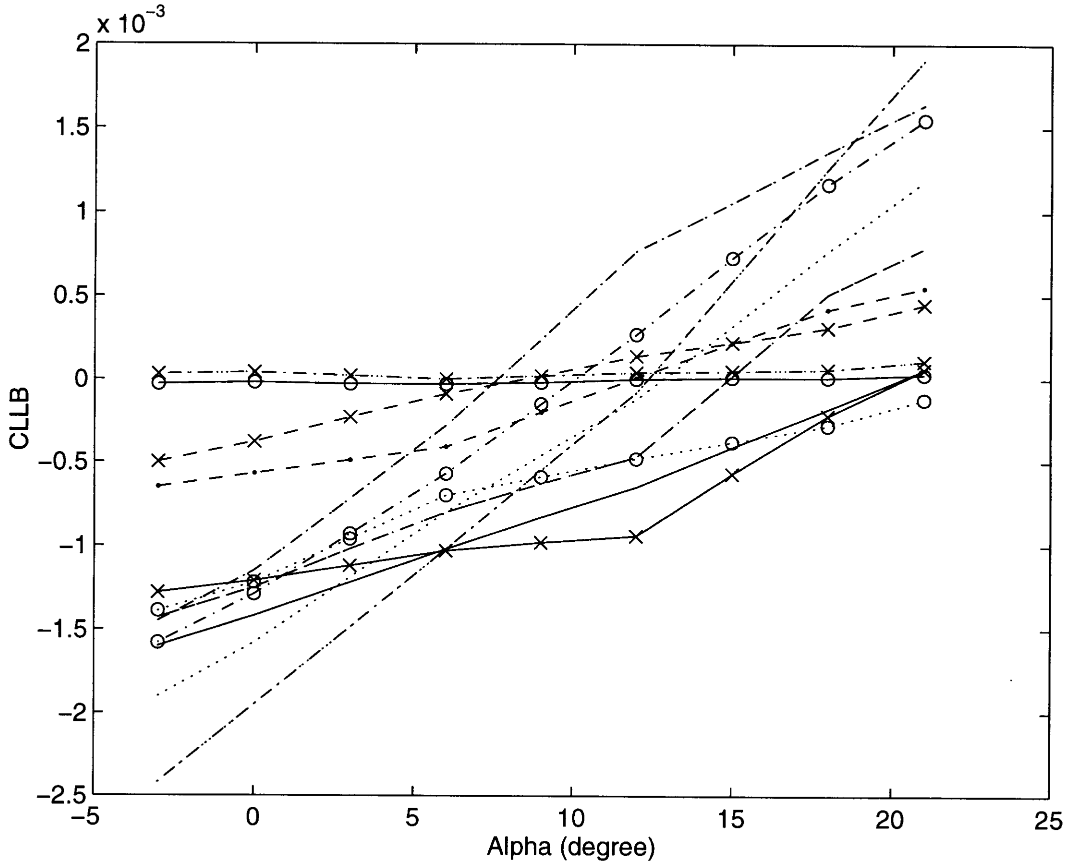


Figure A-13: Aerodynamics coefficient $C_{L_{LB}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.○–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···○···	≡	$M = 2$	–×–×–	≡	$M = 3$
—○—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

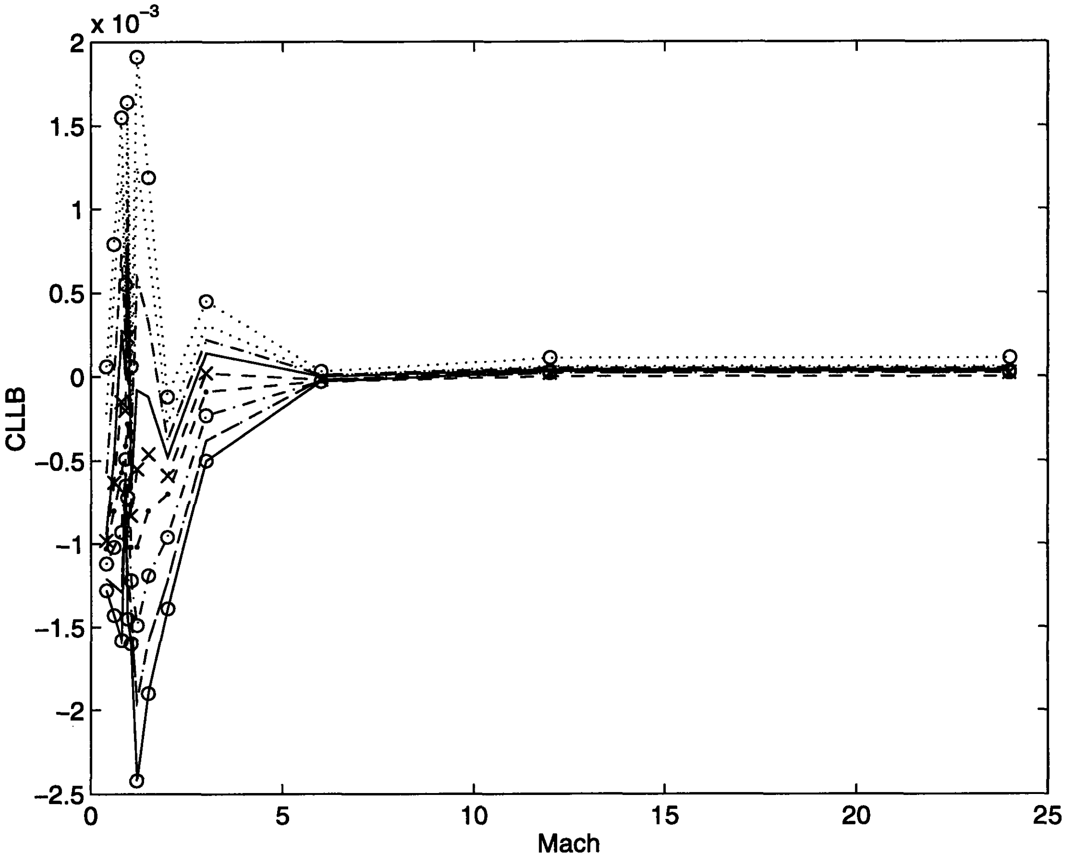


Figure A-14: Aerodynamics coefficient $C_{L_{LB}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

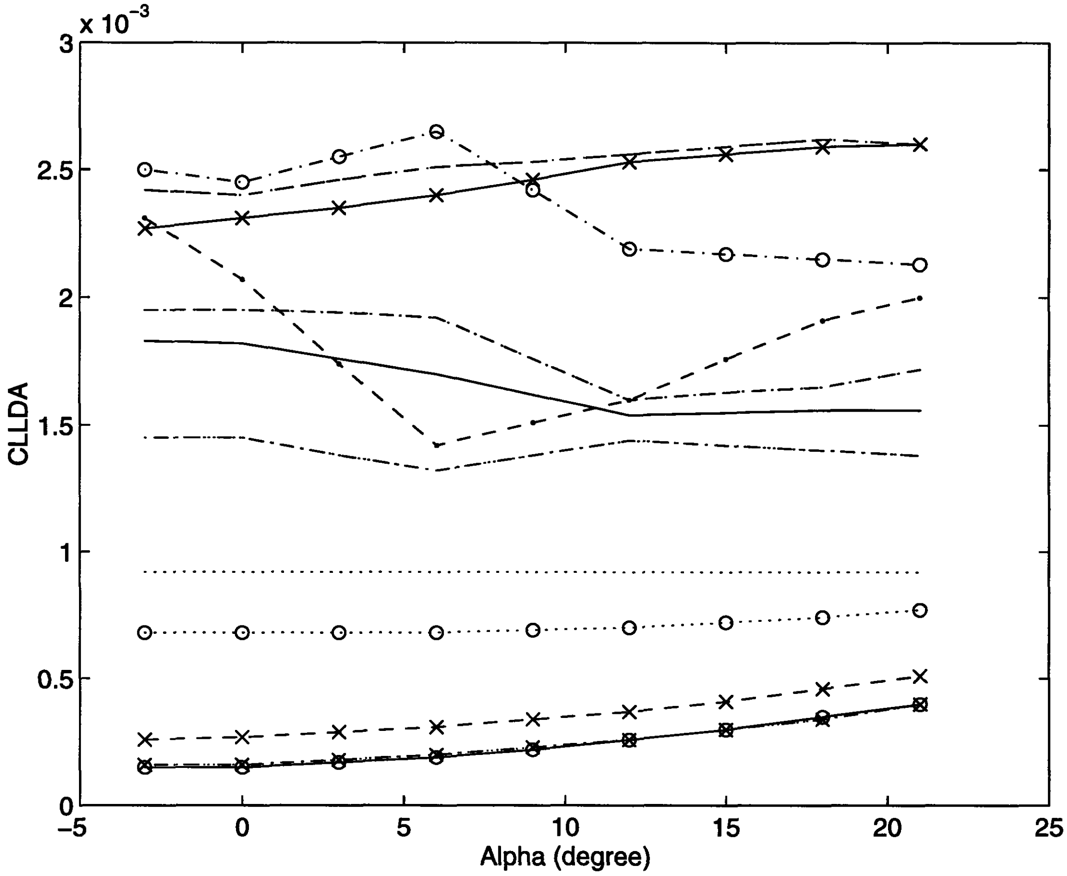


Figure A-15: Aerodynamics coefficient $C_{L_{LDA}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

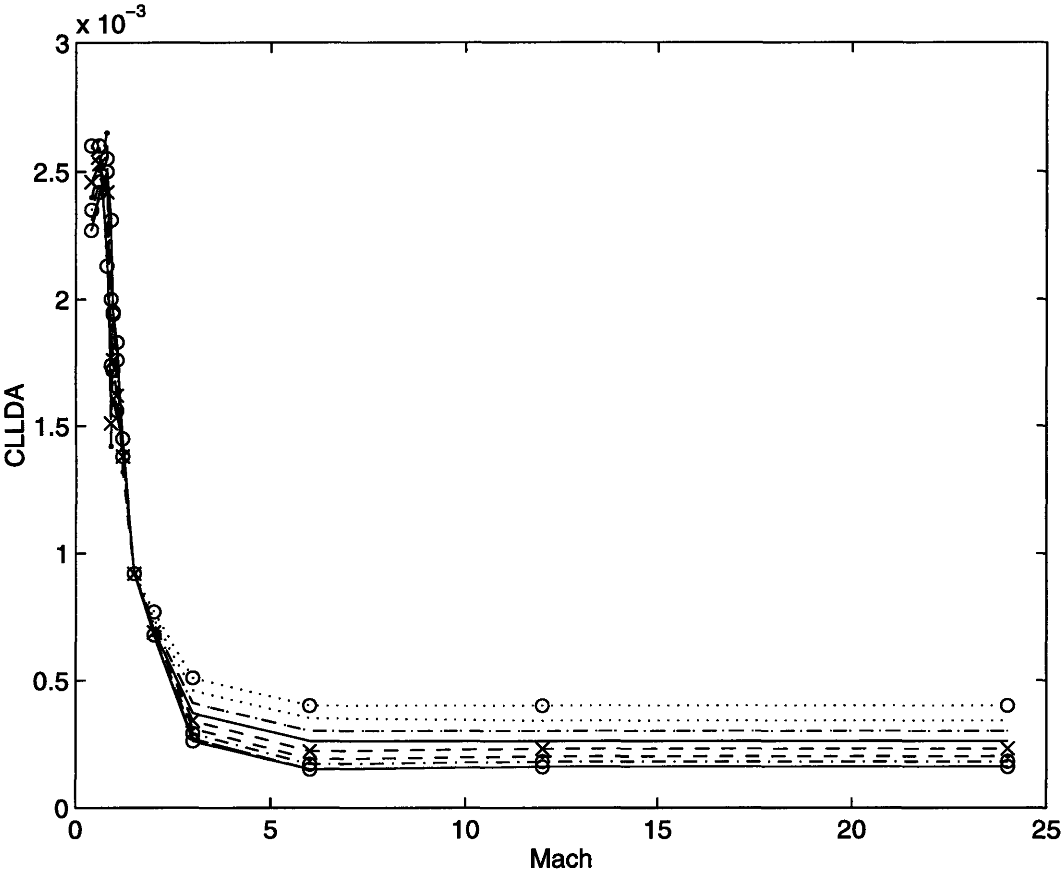


Figure A-16: Aerodynamics coefficient $C_{L_{LDA}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

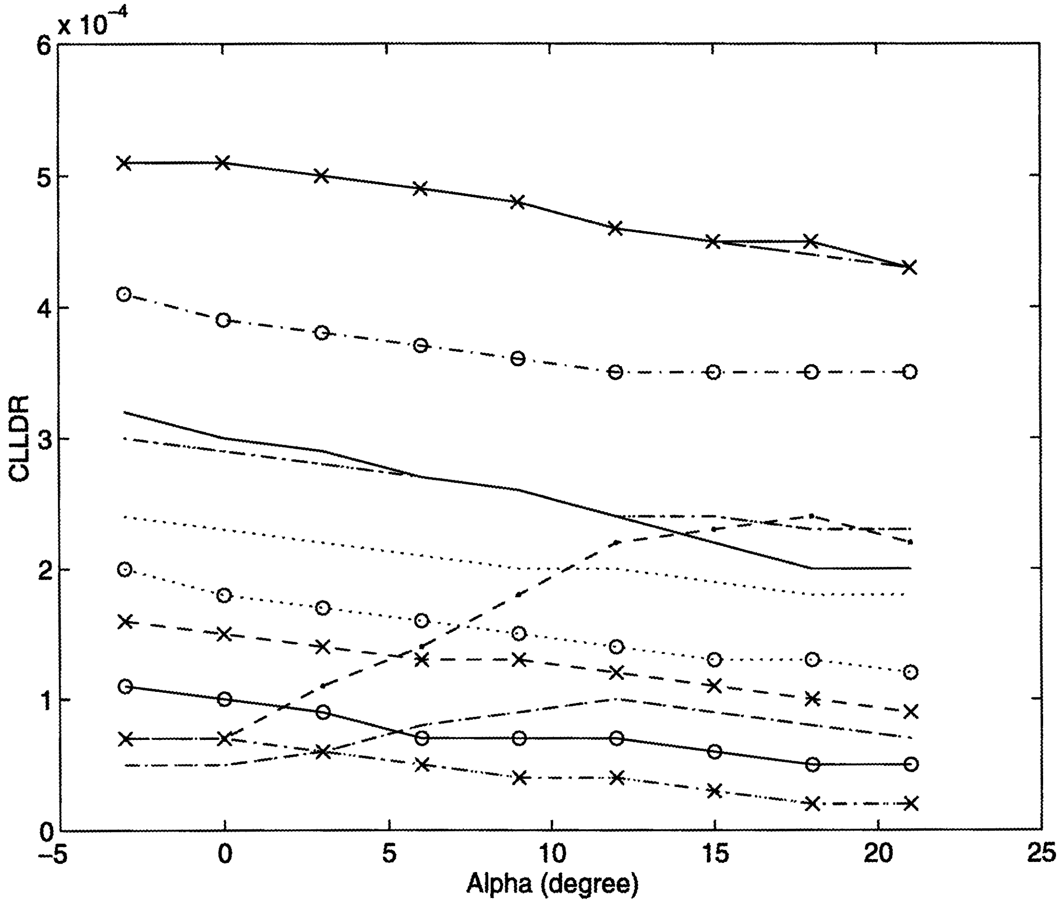


Figure A-17: Aerodynamics coefficient $C_{L_{LDR}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	———	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

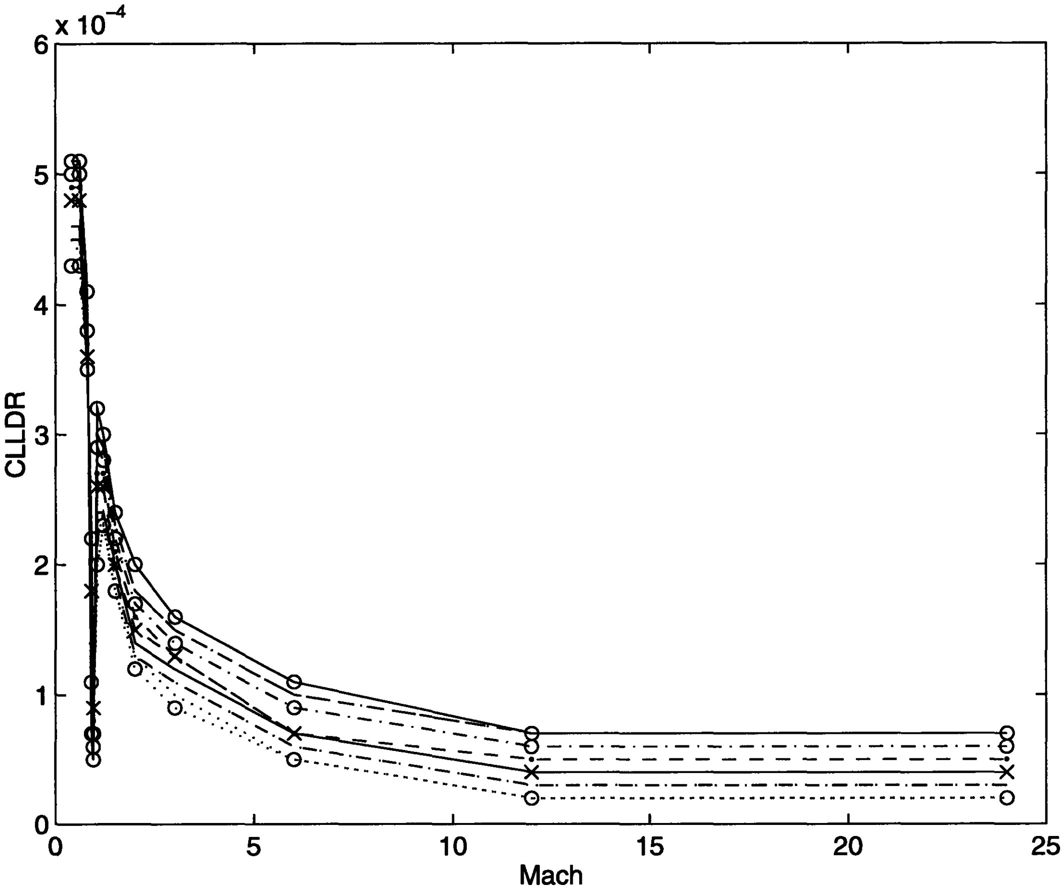


Figure A-18: Aerodynamics coefficient $C_{L_{LDR}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	· · ·	≡	$\alpha = 18^o$
· · · o· · ·	≡	$\alpha = 21^o$			

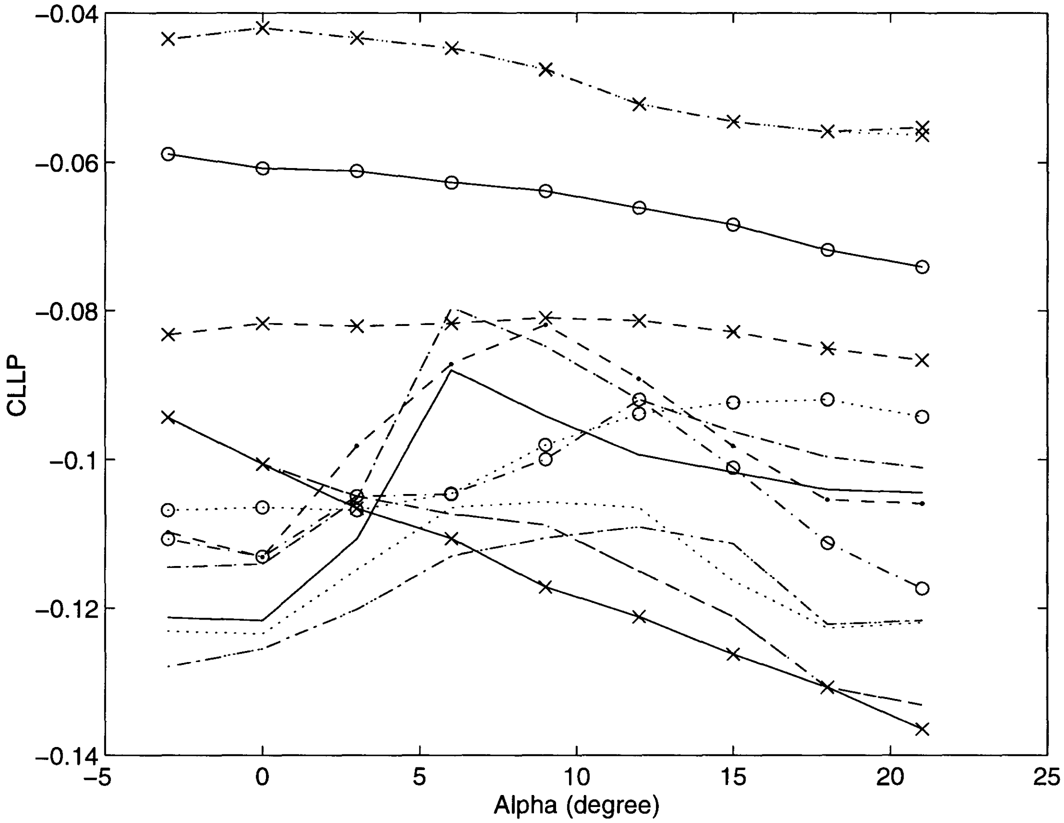


Figure A-19: Aerodynamics coefficient $C_{L_{LP}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

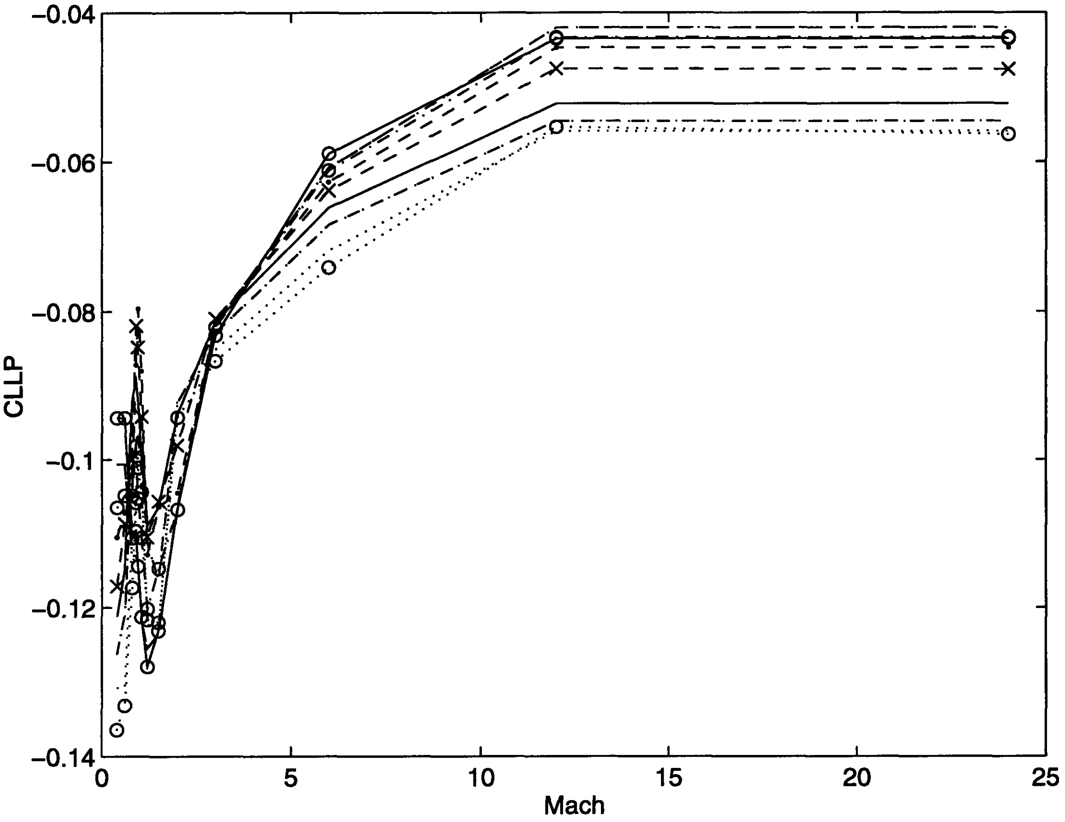


Figure A-20: Aerodynamics coefficient $C_{L_{LP}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

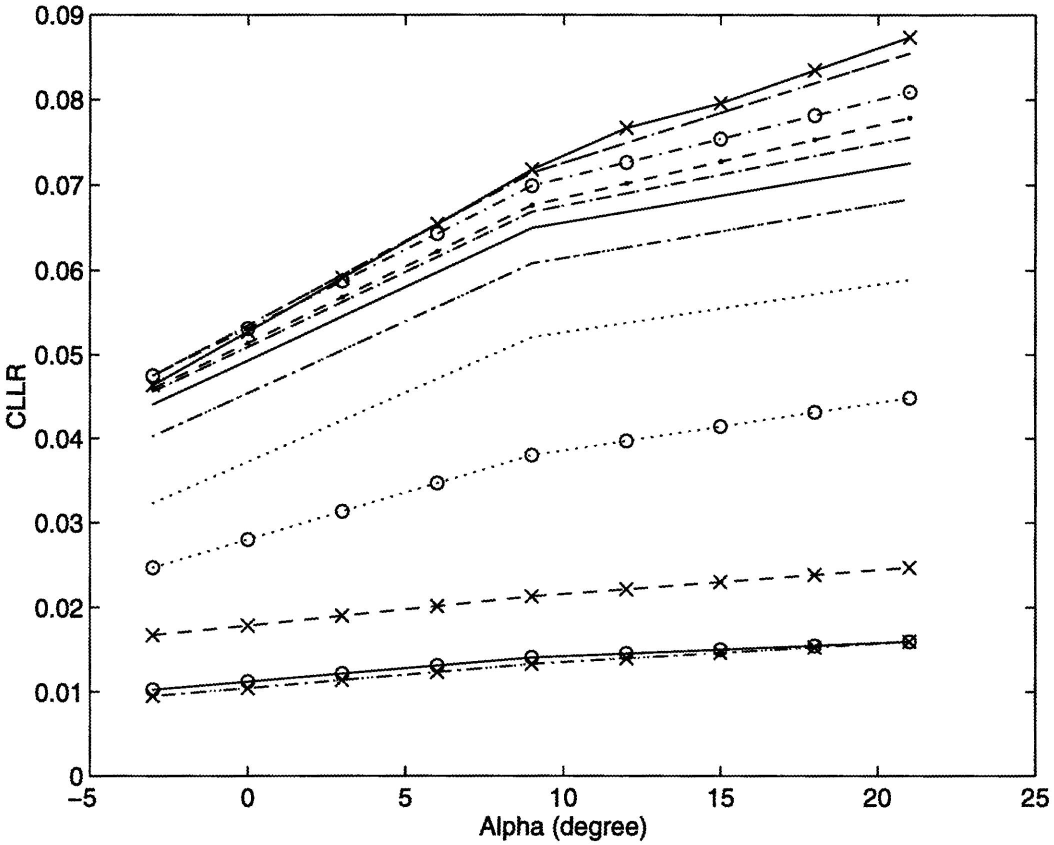


Figure A-21: Aerodynamics coefficient $C_{L_{LR}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.○–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···○···	≡	$M = 2$	–×–×–	≡	$M = 3$
—○—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

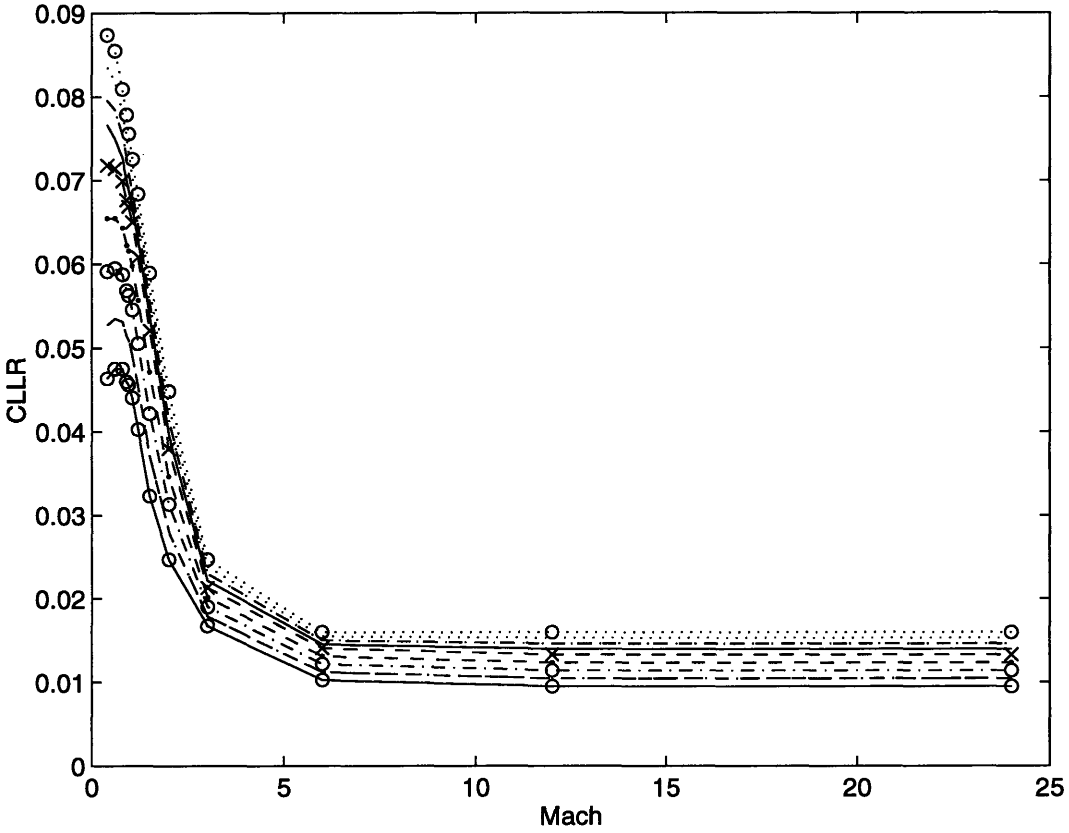


Figure A-22: Aerodynamics coefficient $C_{L_{LR}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	———	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

A.3 Directional Aerodynamics Coefficients

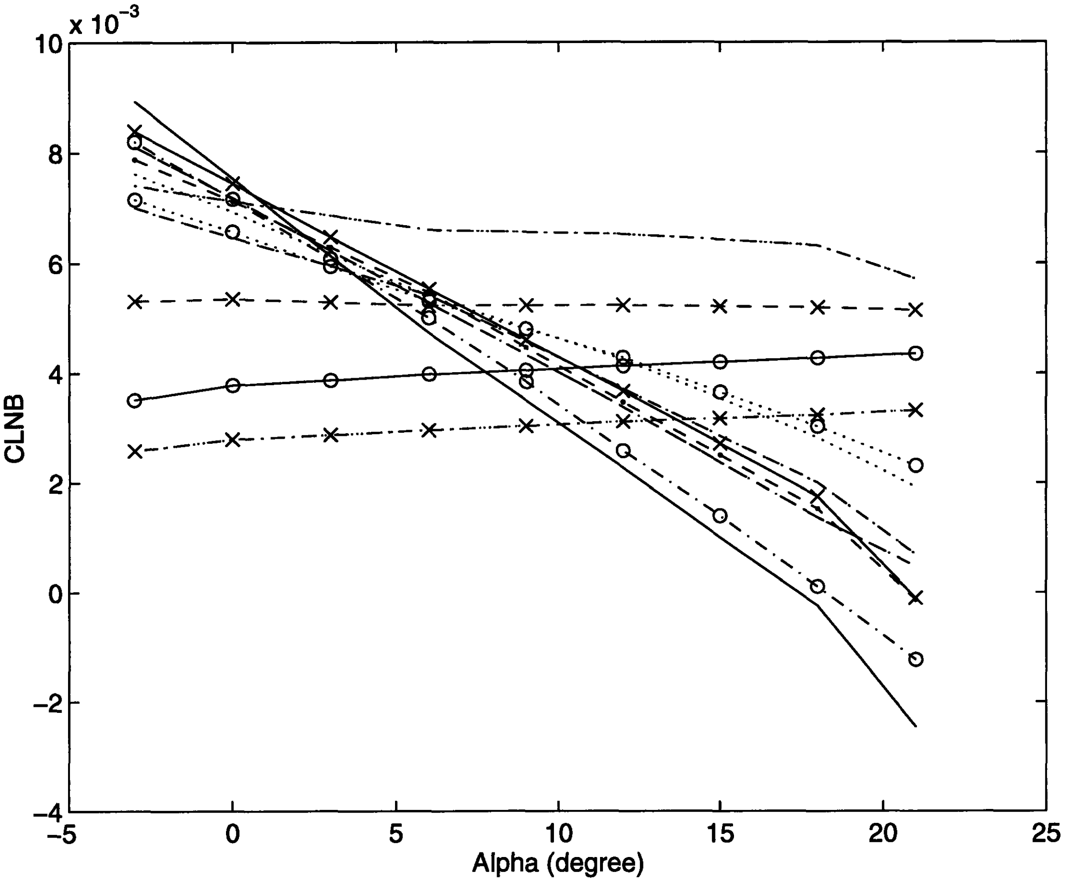


Figure A-23: Aerodynamics coefficient $C_{L_{NB}}$ as a function of angle of attack α

—x—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–x–x–	≡	$M = 3$
—o—	≡	$M = 6$	–.x–.	≡	$M = 12$
···x···	≡	$M = 24$			

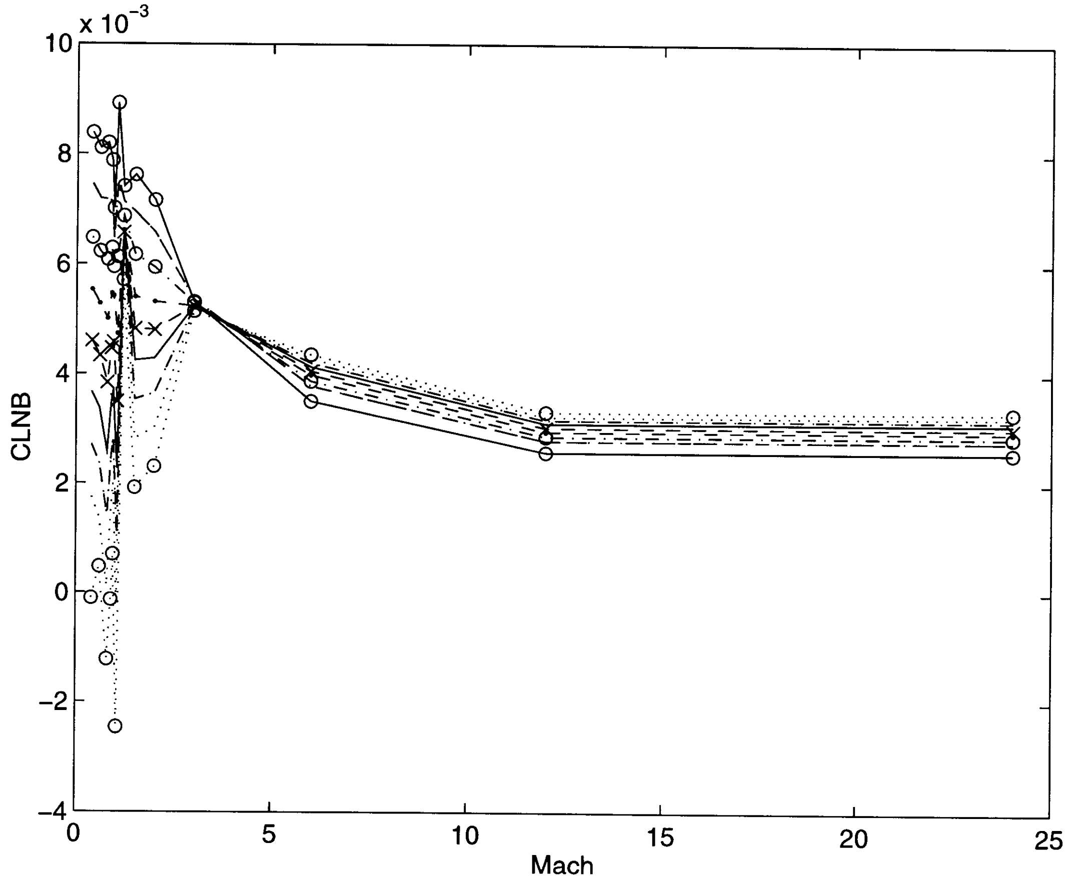


Figure A-24: Aerodynamics coefficient $C_{L_{NB}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	· · ·	≡	$\alpha = 18^o$
· · · o· · ·	≡	$\alpha = 21^o$			

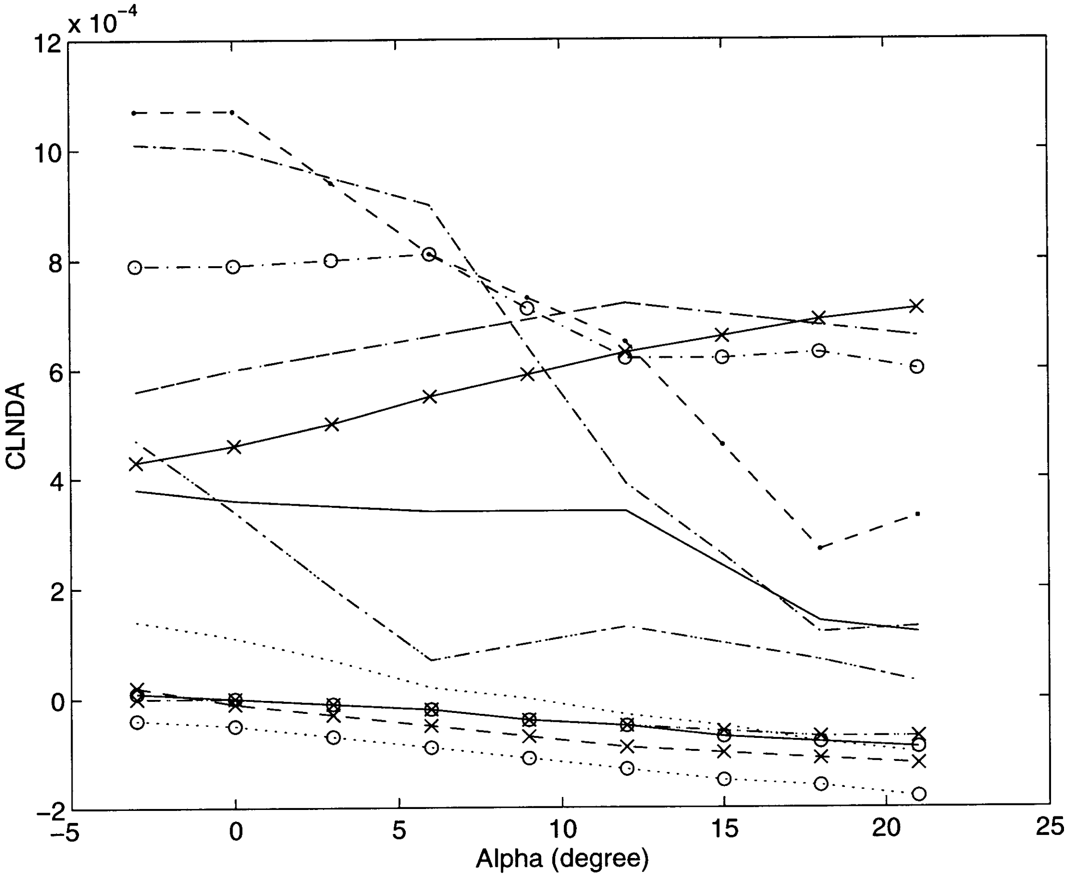


Figure A-25: Aerodynamics coefficient $C_{L_{NDA}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–×–×–	≡	$M = 3$
—o—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

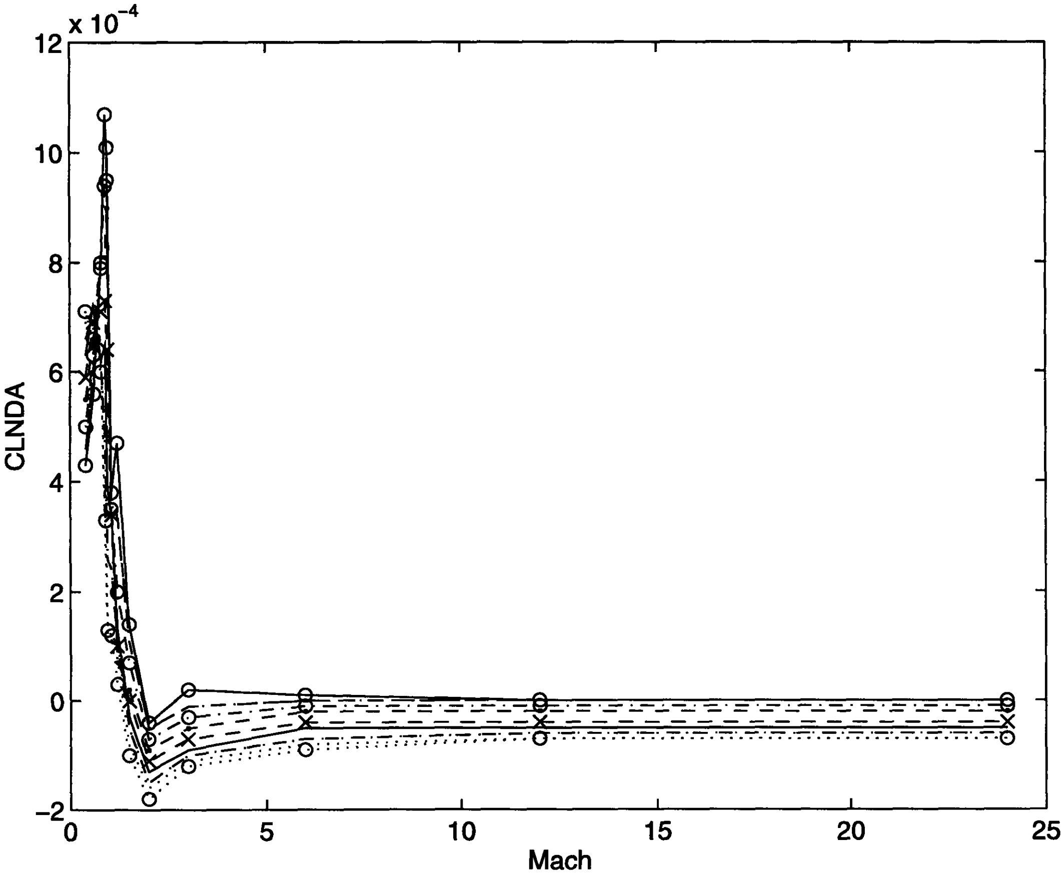


Figure A-26: Aerodynamics coefficient $C_{L_{NDA}}$ as a function Mach number M

—o—	$\equiv \alpha = -3^o$	– –.	$\equiv \alpha = 0^o$
—.o–	$\equiv \alpha = 3^o$	– ● –	$\equiv \alpha = 6^o$
.–×–	$\equiv \alpha = 9^o$	——	$\equiv \alpha = 12^o$
–.–.	$\equiv \alpha = 15^o$	···	$\equiv \alpha = 18^o$
··· o···	$\equiv \alpha = 21^o$		

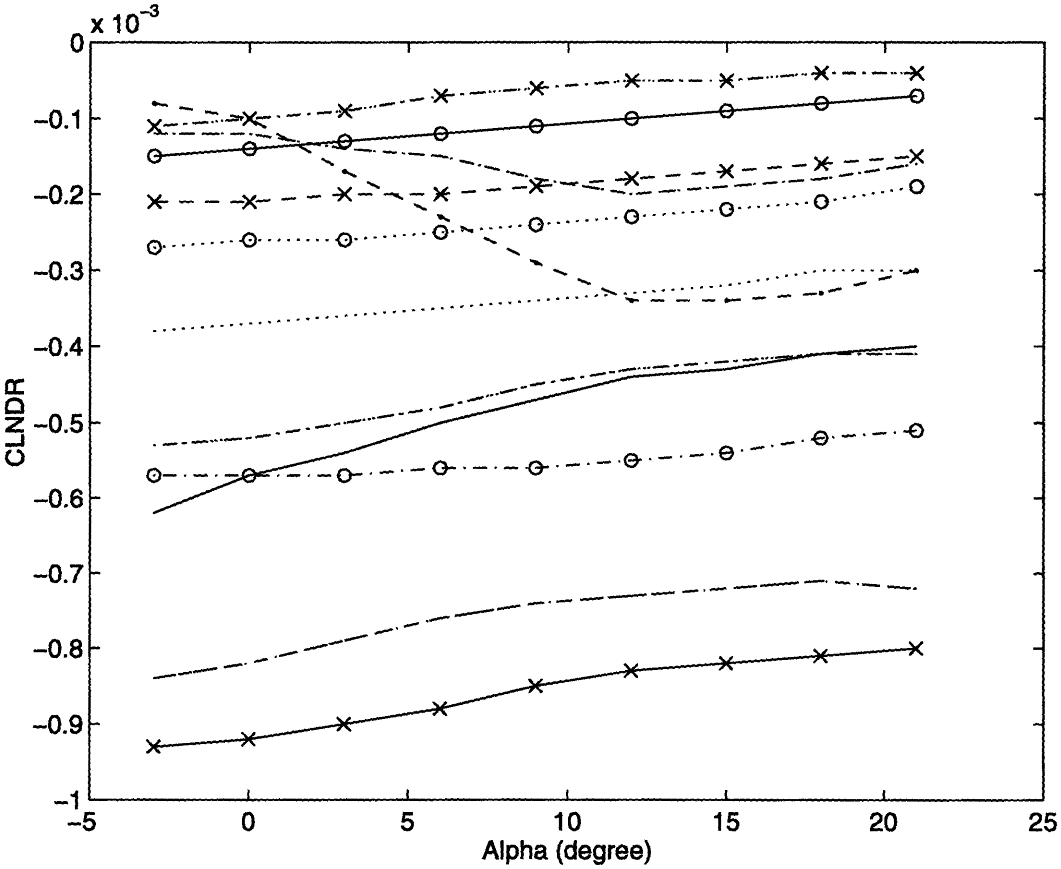


Figure A-27: Aerodynamics coefficient $C_{L_{NDR}}$ as a function of angle of attack α

—x—	≡	$M = .4$	– –.	≡	$M = .6$
—.o–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···o···	≡	$M = 2$	–x–x–	≡	$M = 3$
—o—	≡	$M = 6$	–.x–.	≡	$M = 12$
··· x ···	≡	$M = 24$			

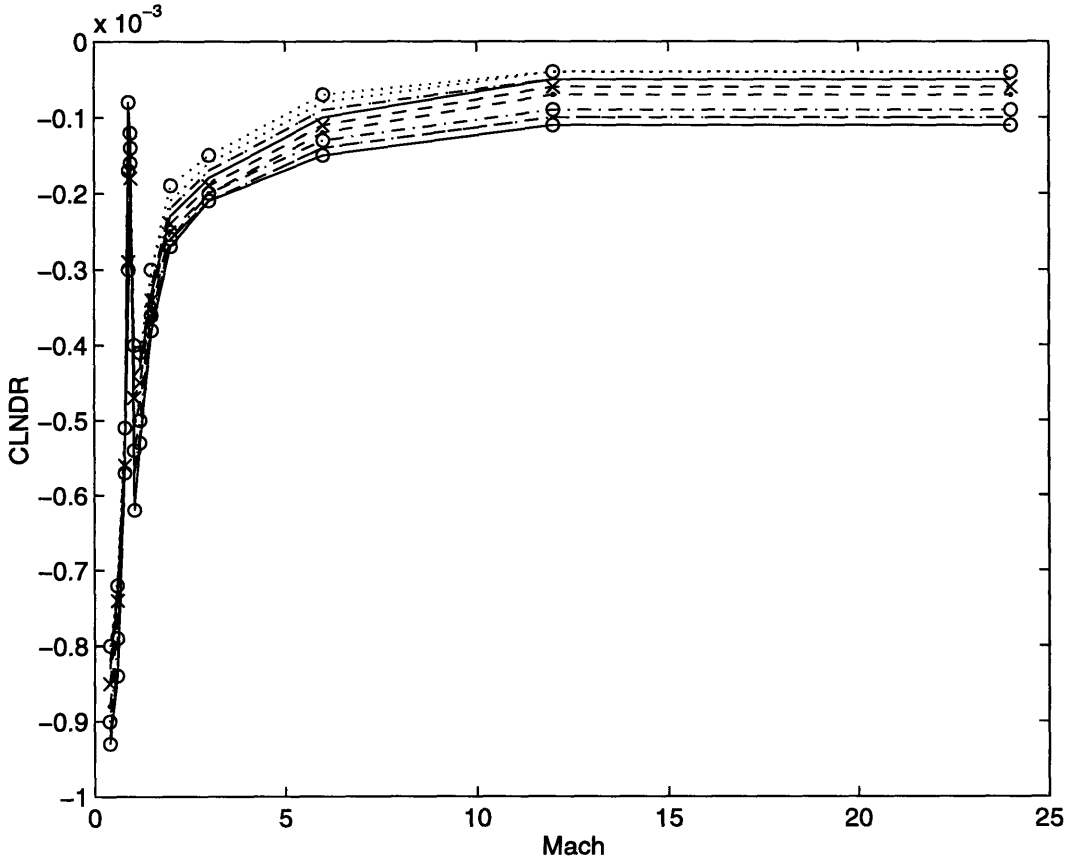


Figure A-28: Aerodynamics coefficient $C_{L_{NDR}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

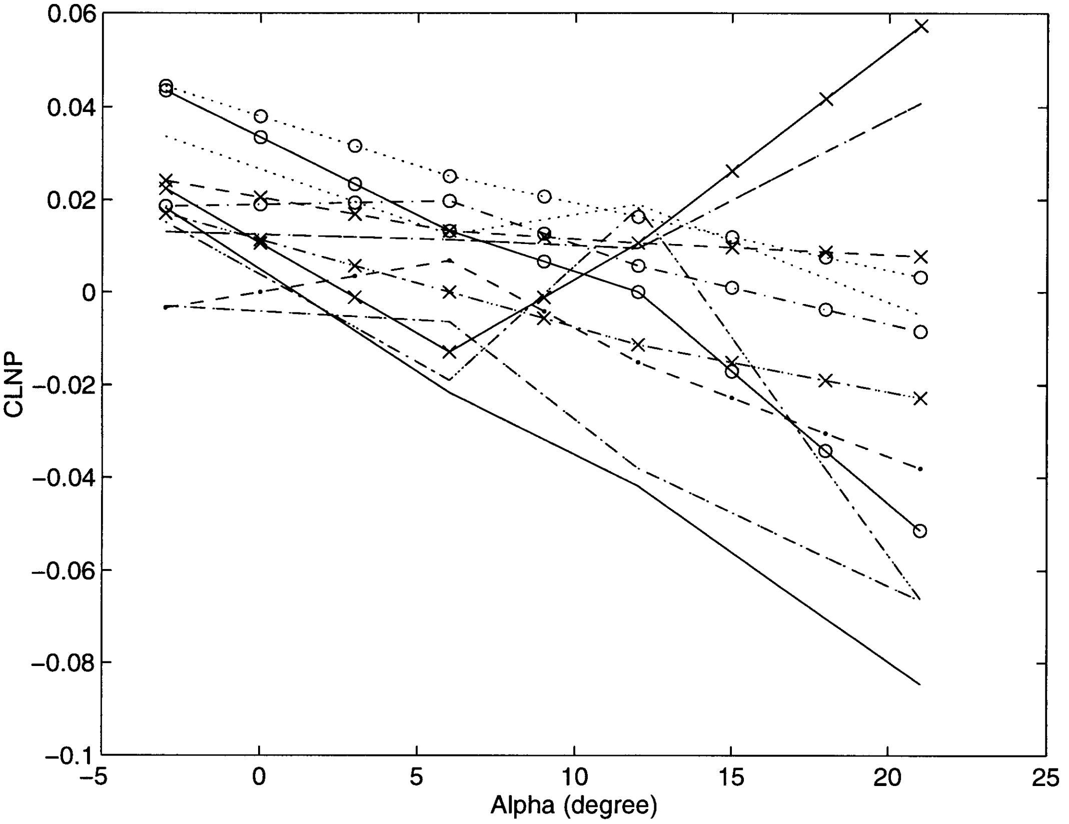


Figure A-29: Aerodynamics coefficient $C_{L_{NP}}$ as a function of angle of attack α

—×—	$\equiv$	$M = .4$	– –.	$\equiv$	$M = .6$
—.o–	$\equiv$	$M = .8$	– ● –	$\equiv$	$M = .9$
–.–.	$\equiv$	$M = .95$	——	$\equiv$	$M = 1.05$
–..–	$\equiv$	$M = 1.2$	······	$\equiv$	$M = 1.5$
··· o ···	$\equiv$	$M = 2$	–×–×–	$\equiv$	$M = 3$
—o—	$\equiv$	$M = 6$	–.×–.	$\equiv$	$M = 12$
··· × ···	$\equiv$	$M = 24$			

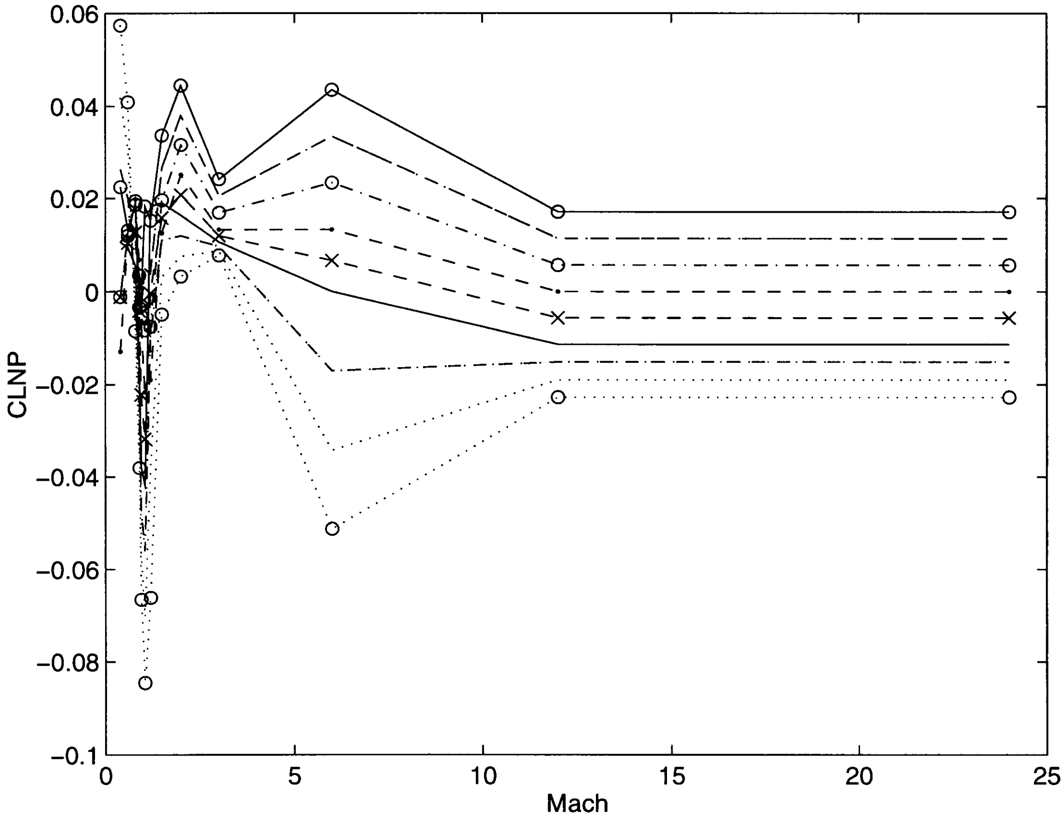


Figure A-30: Aerodynamics coefficient $C_{L_{NP}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	· · ·	≡	$\alpha = 18^o$
· · · o· · ·	≡	$\alpha = 21^o$			

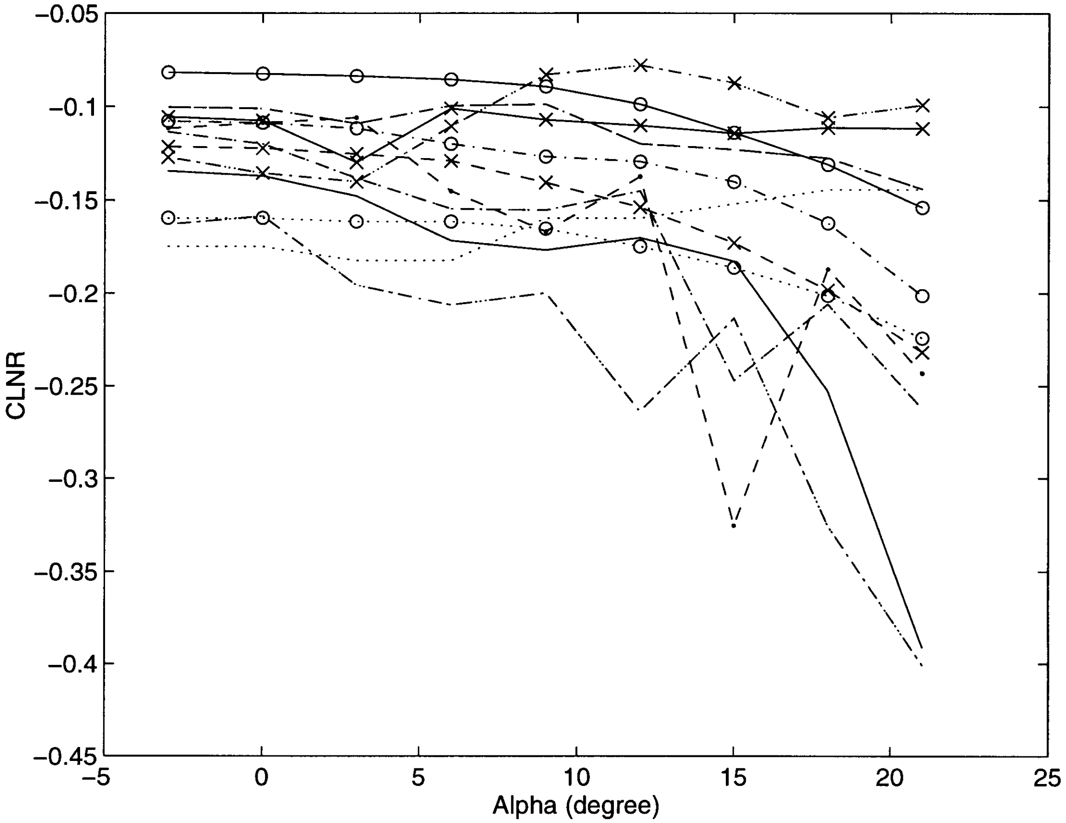


Figure A-31: Aerodynamics coefficient $C_{L_{NR}}$ as a function of angle of attack α

—×—	≡	$M = .4$	– –.	≡	$M = .6$
—.○–	≡	$M = .8$	– ● –	≡	$M = .9$
–.–.	≡	$M = .95$	——	≡	$M = 1.05$
–..–	≡	$M = 1.2$	······	≡	$M = 1.5$
···○···	≡	$M = 2$	–×–×–	≡	$M = 3$
—○—	≡	$M = 6$	–.×–.	≡	$M = 12$
···×···	≡	$M = 24$			

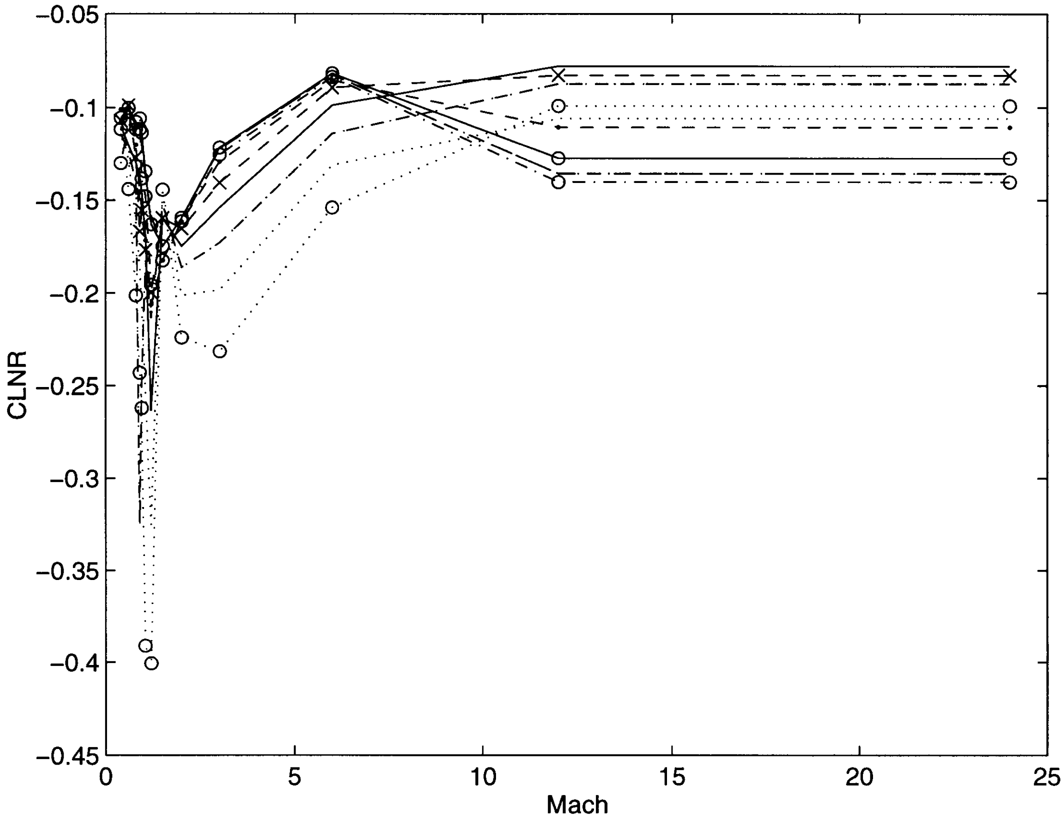


Figure A-32: Aerodynamics coefficient $C_{L_{NR}}$ as a function Mach number M

—o—	≡	$\alpha = -3^o$	– –.	≡	$\alpha = 0^o$
—.o–	≡	$\alpha = 3^o$	– ● –	≡	$\alpha = 6^o$
.–×–	≡	$\alpha = 9^o$	——	≡	$\alpha = 12^o$
–.–.	≡	$\alpha = 15^o$	···	≡	$\alpha = 18^o$
··· o···	≡	$\alpha = 21^o$			

Appendix B

Three-dimensional Aerodynamics Data

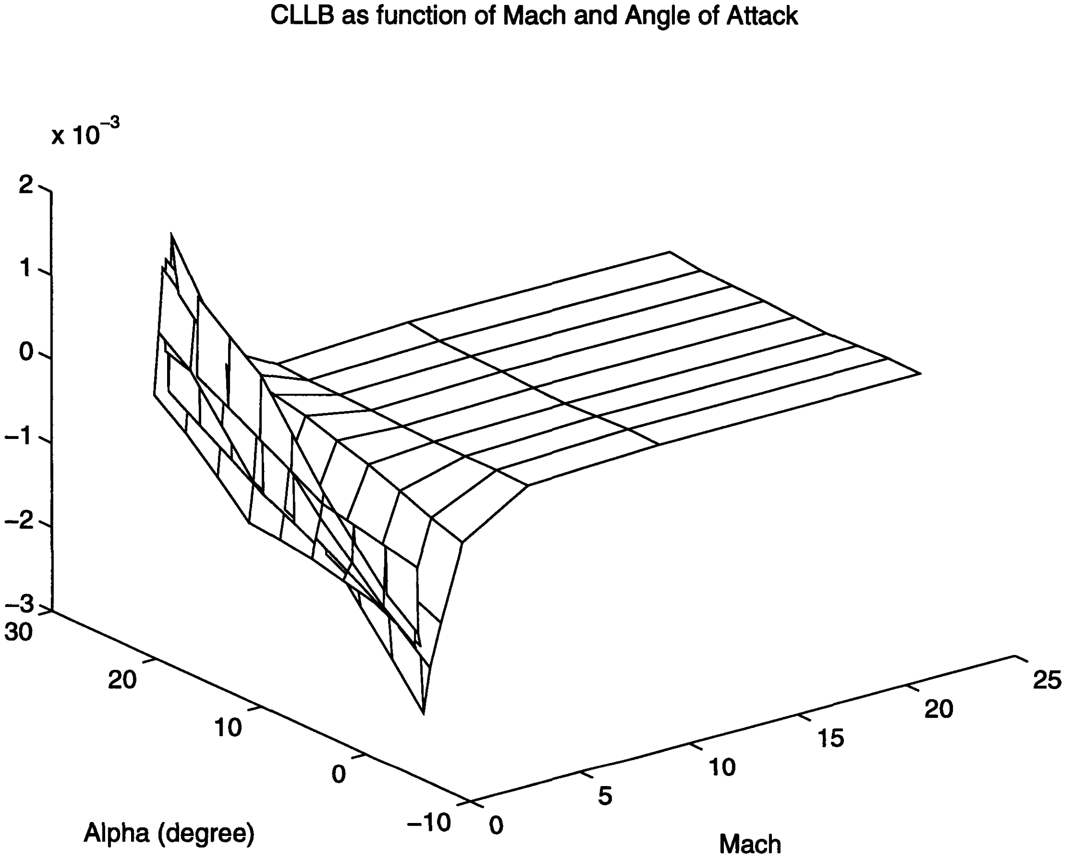


Figure B-1: Aerodynamics coefficient C_{L_LB} as a function of angle of attack α and Mach number M

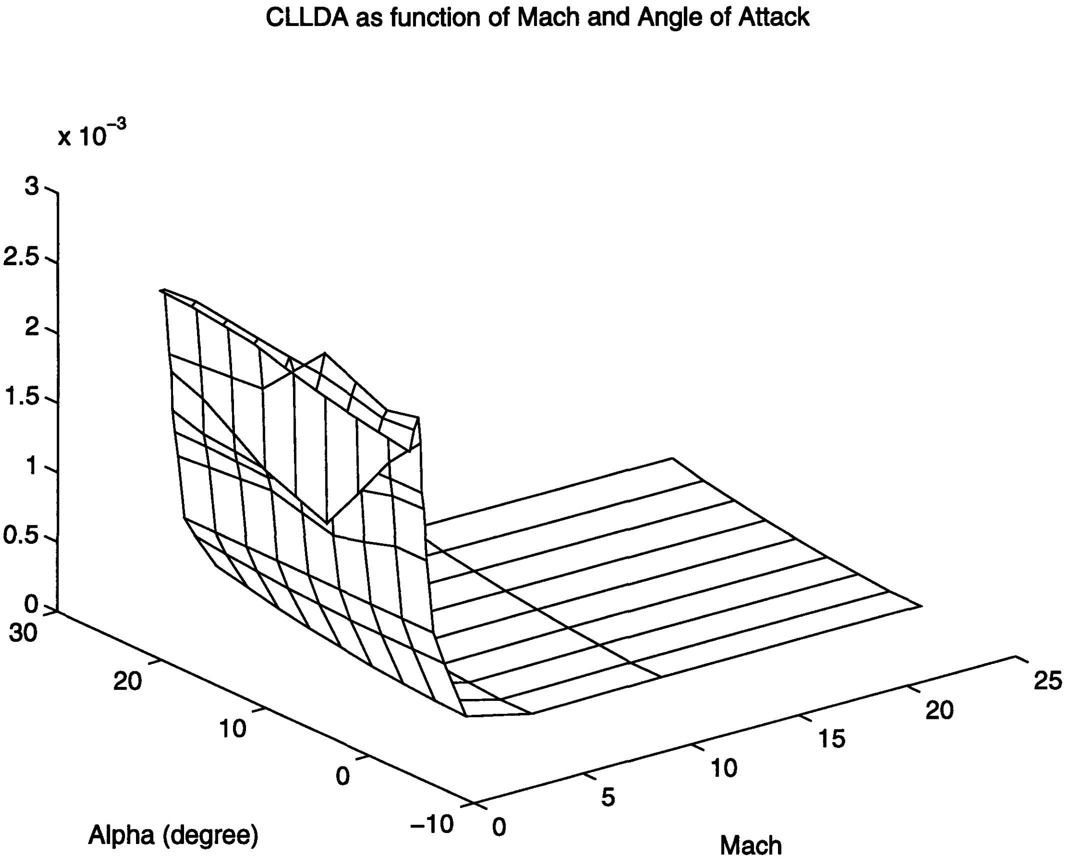


Figure B-2: Aerodynamics coefficient C_{L_LDA} as a function of angle of attack α and Mach number M

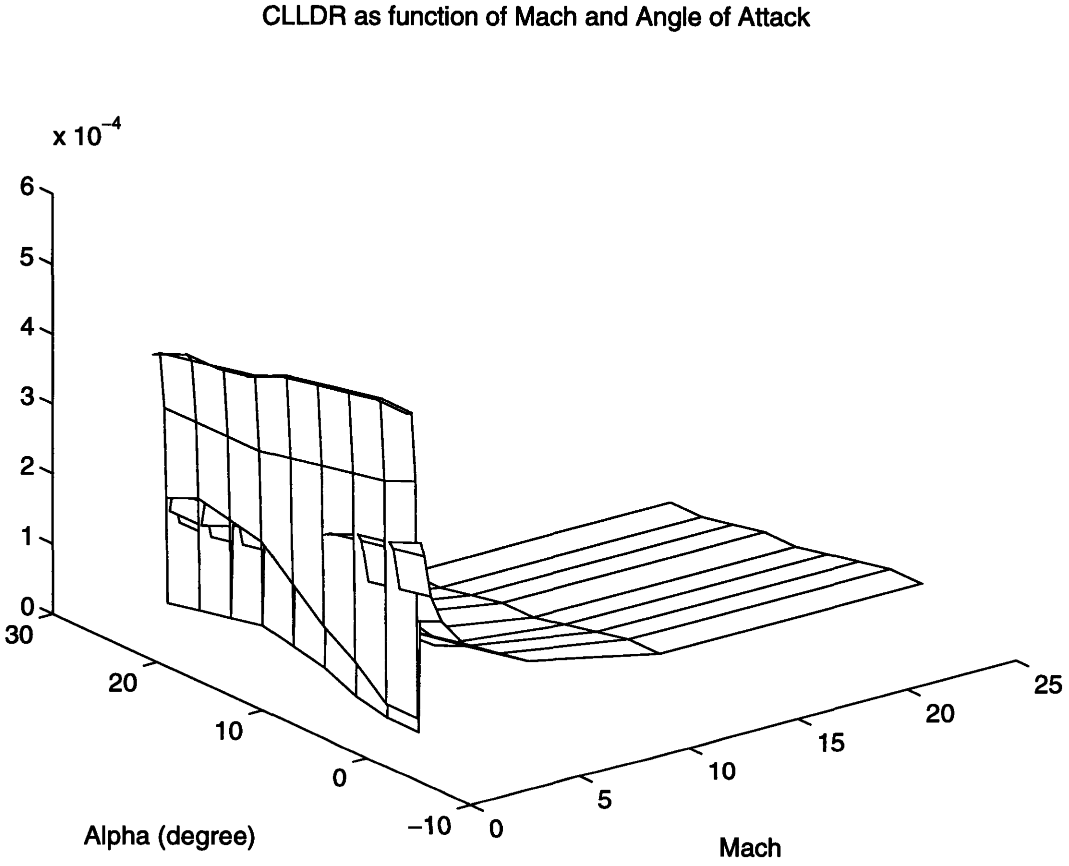


Figure B-3: Aerodynamics coefficient $C_{L_{LDR}}$ as a function of angle of attack α and Mach number M

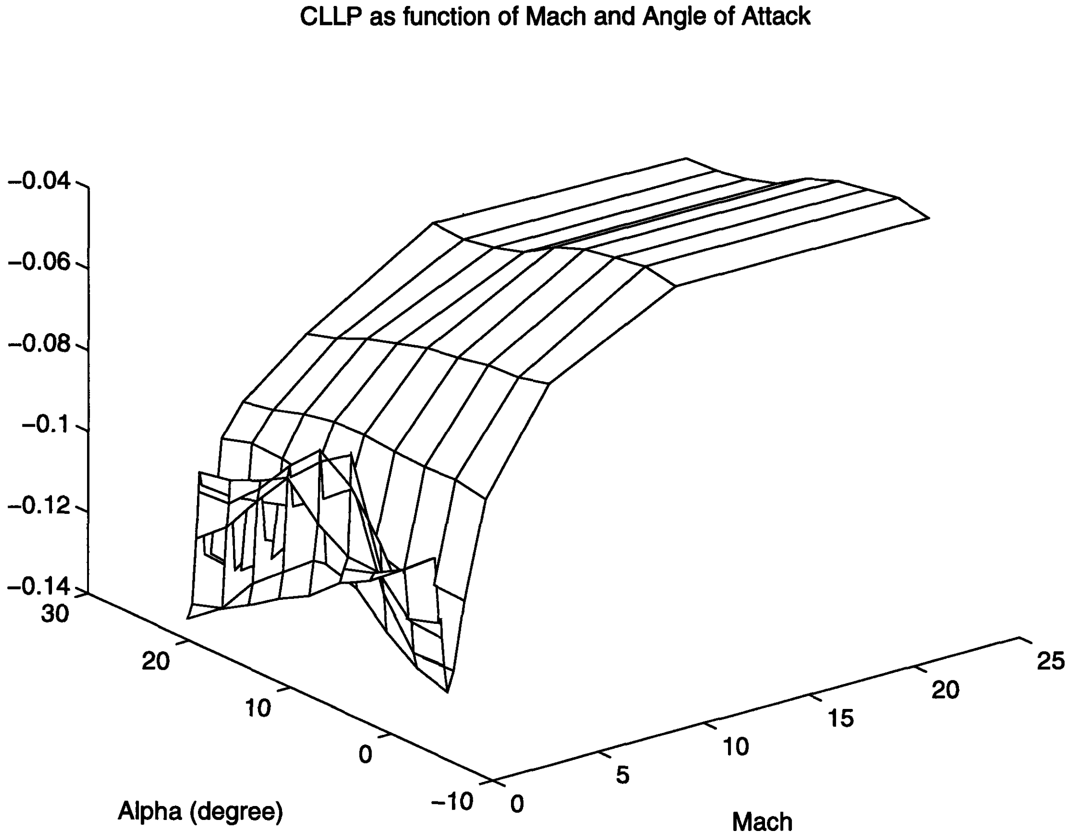


Figure B-4: Aerodynamics coefficient $C_{L_L P}$ as a function of angle of attack α and Mach number M

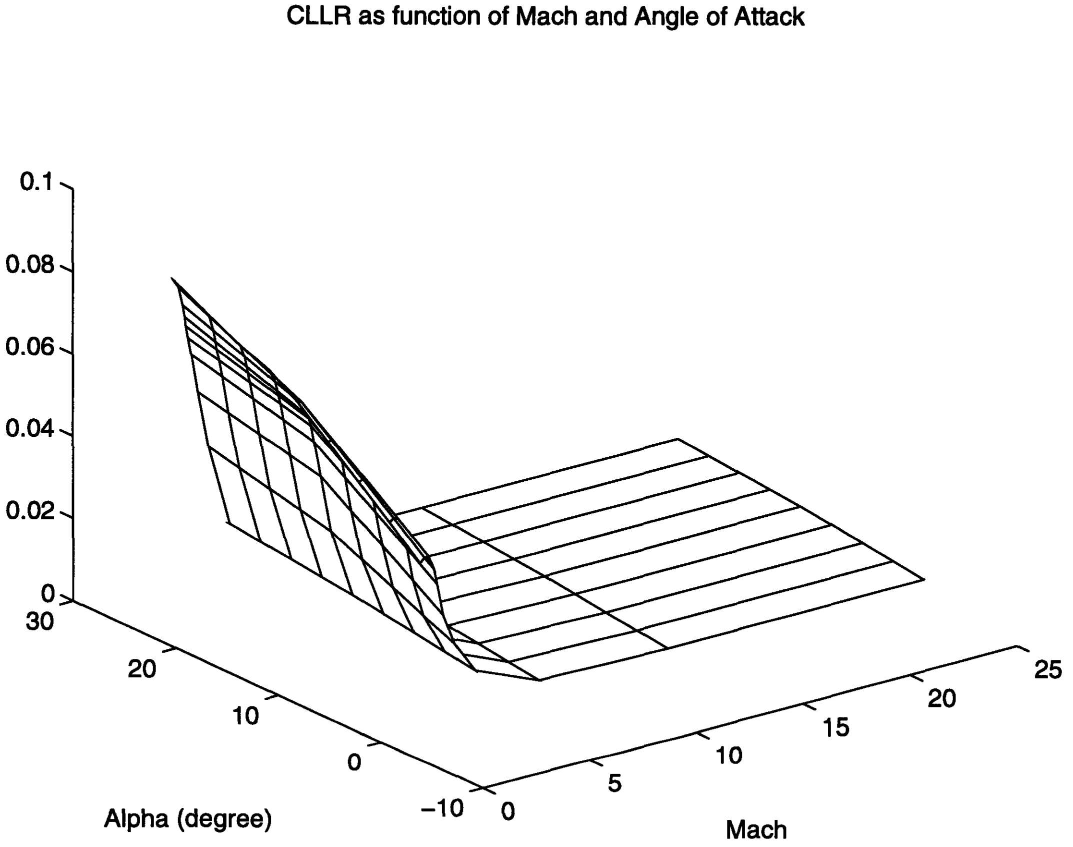


Figure B-5: Aerodynamics coefficient C_{L_LR} as a function of angle of attack α and Mach number M

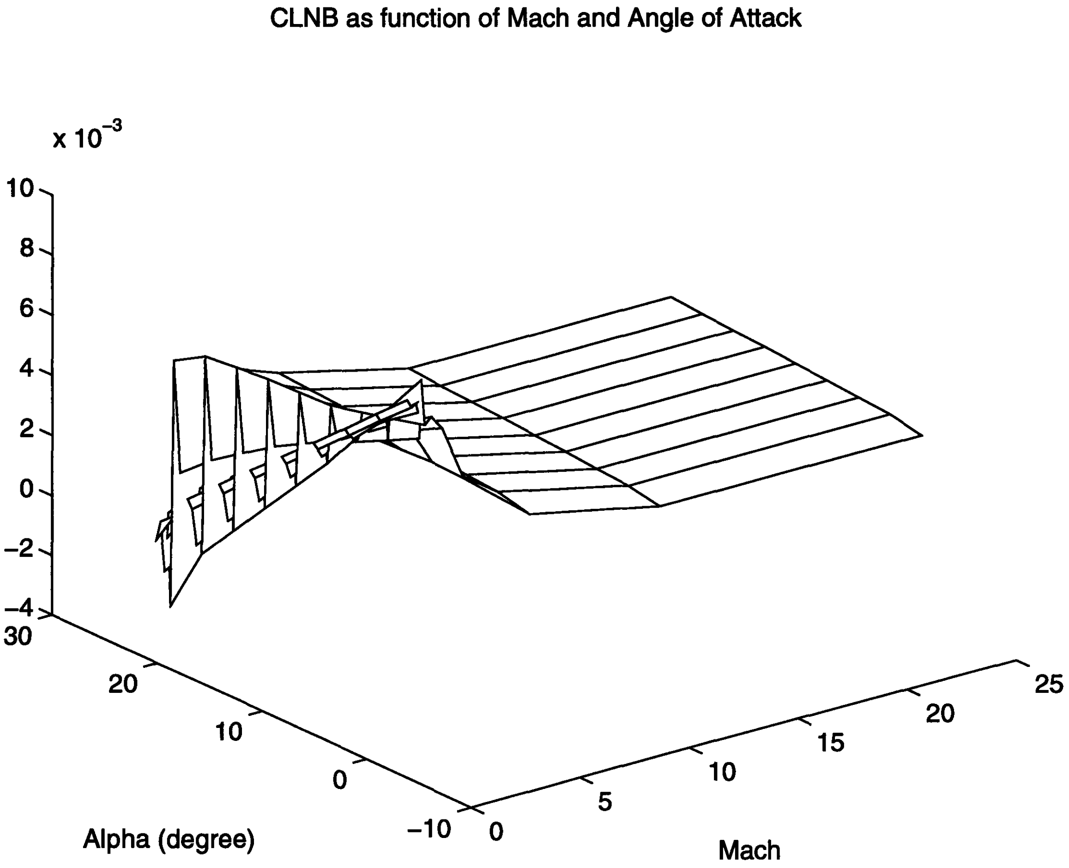


Figure B-6: Aerodynamics coefficient $C_{L_N B}$ as a function of angle of attack α and Mach number M

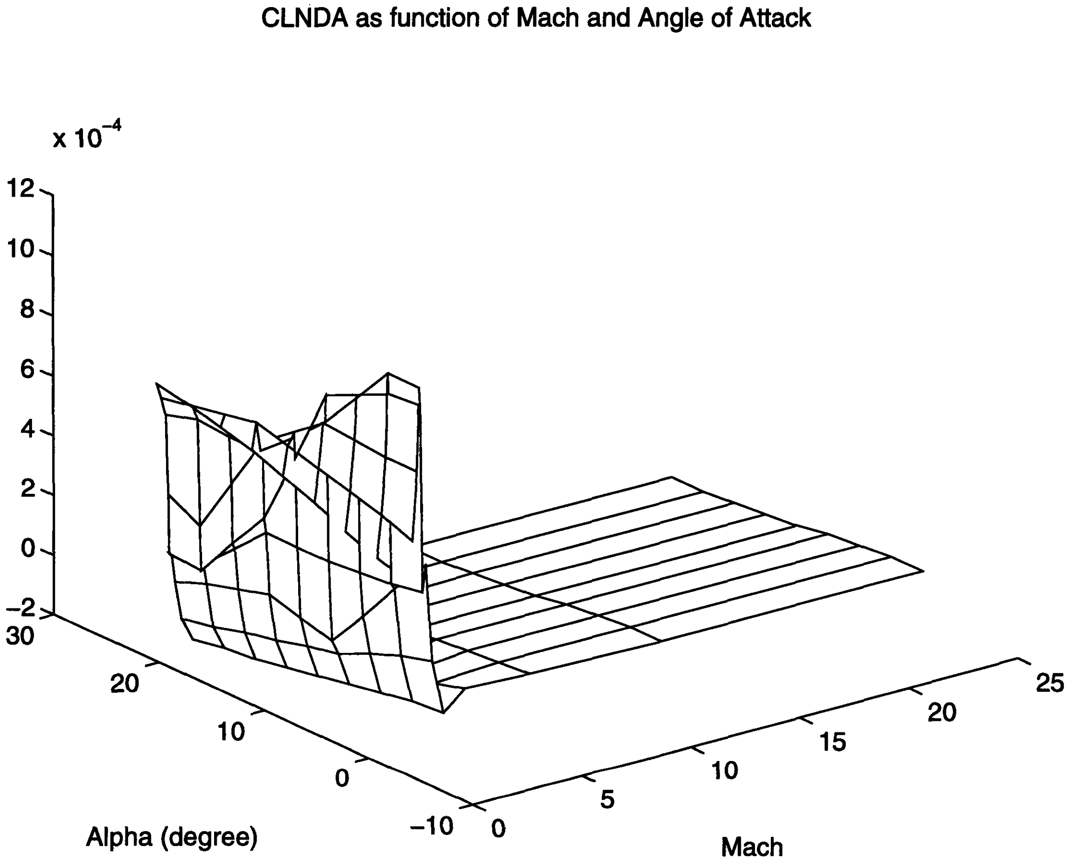


Figure B-7: Aerodynamics coefficient $C_{L_N DA}$ as a function of angle of attack α and Mach number M

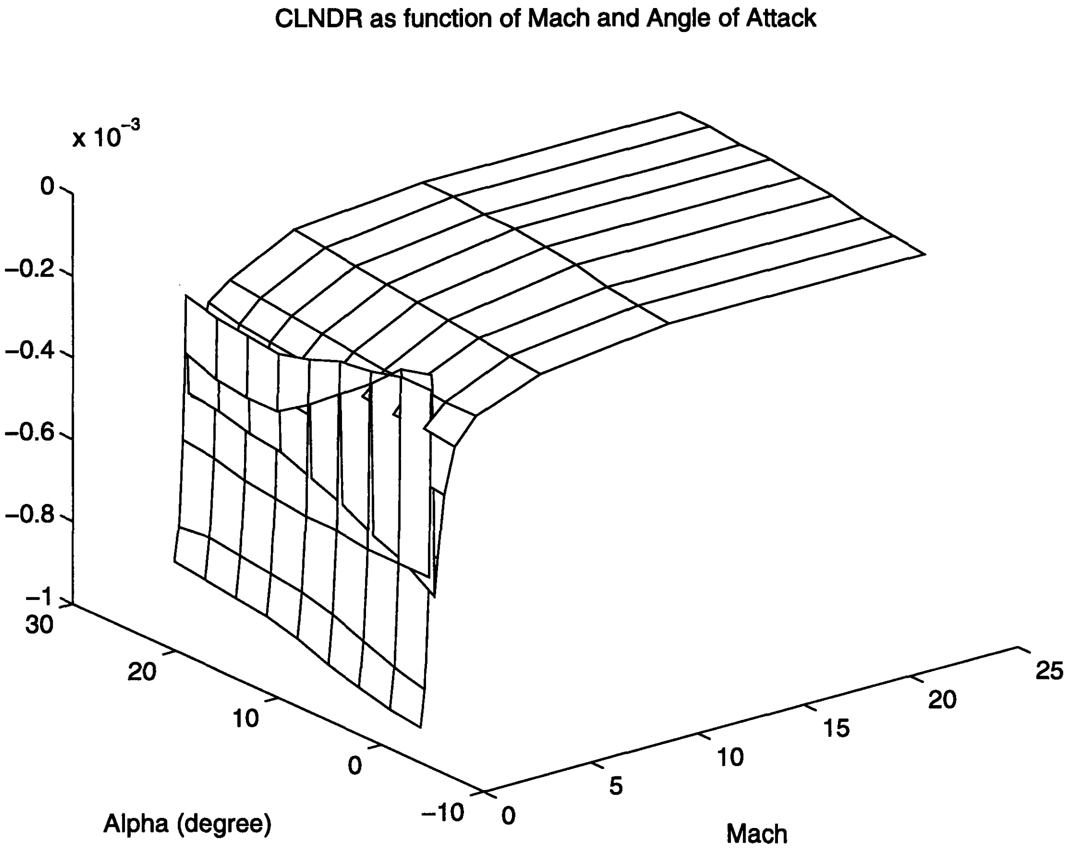


Figure B-8: Aerodynamics coefficient C_{L_NDR} as a function of angle of attack α and Mach number M

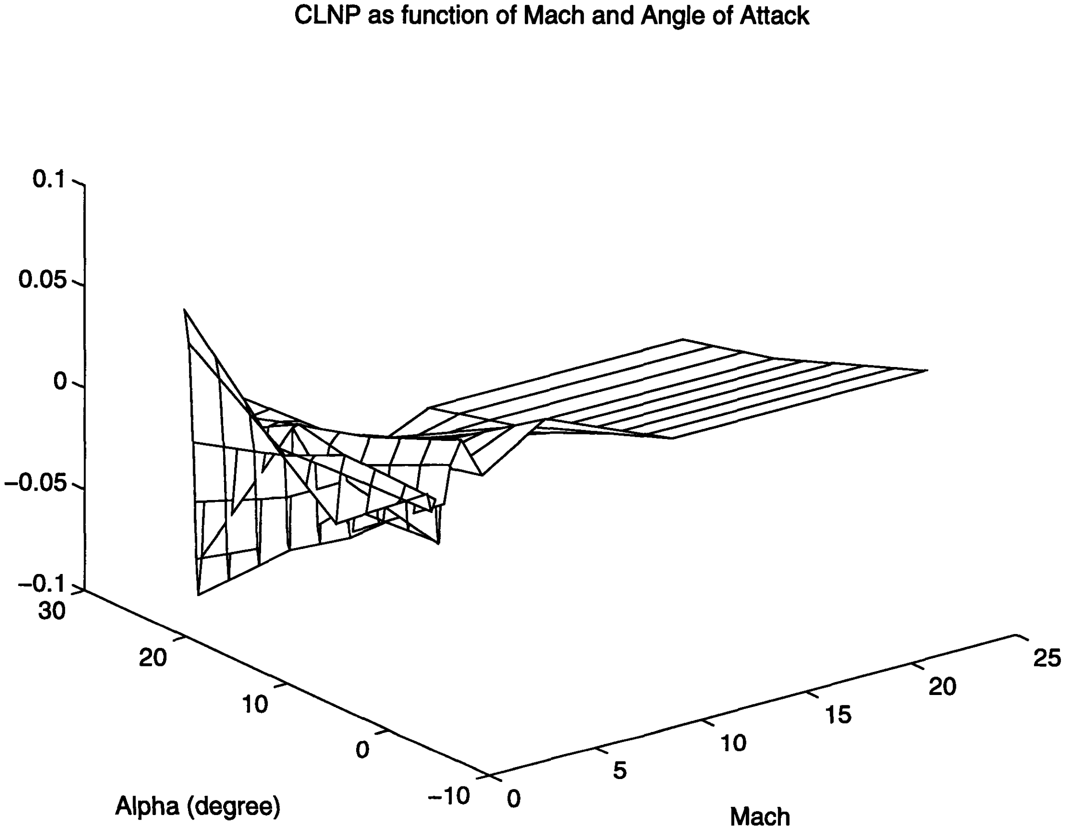


Figure B-9: Aerodynamics coefficient C_{L_NP} as a function of angle of attack α and Mach number M

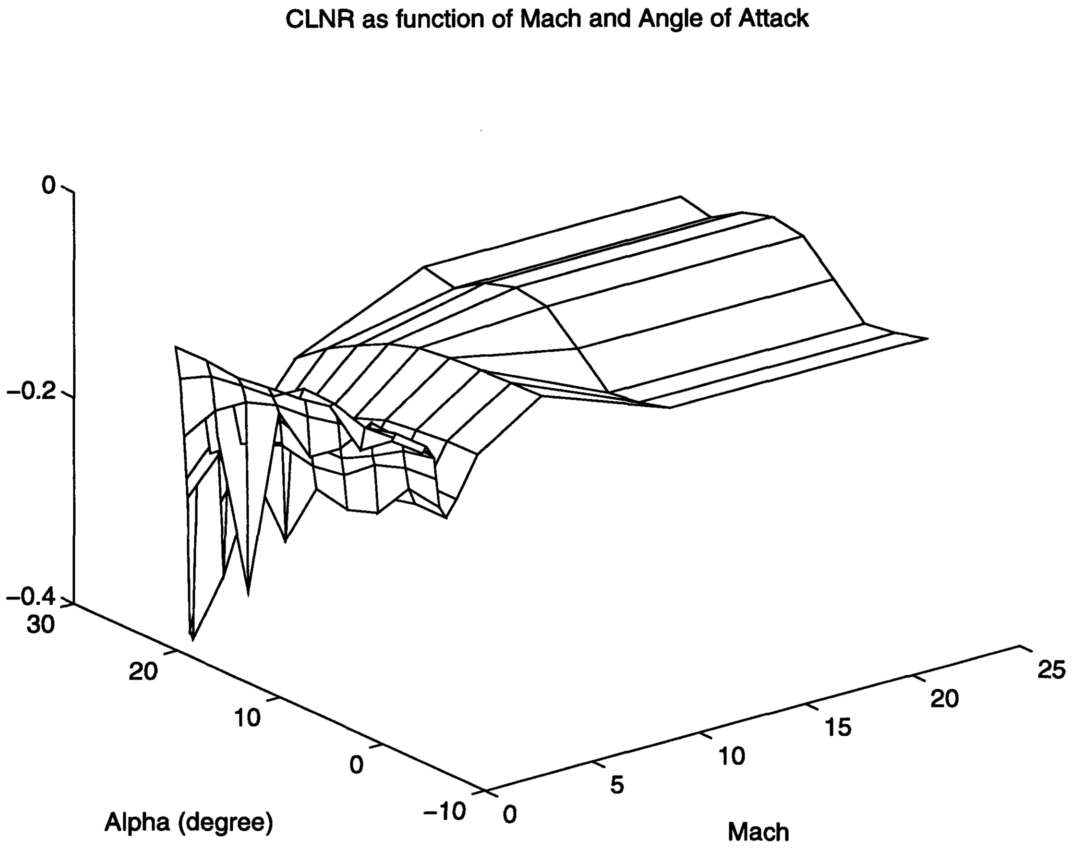


Figure B-10: Aerodynamics coefficient $C_{L_N R}$ as a function of angle of attack α and Mach number M

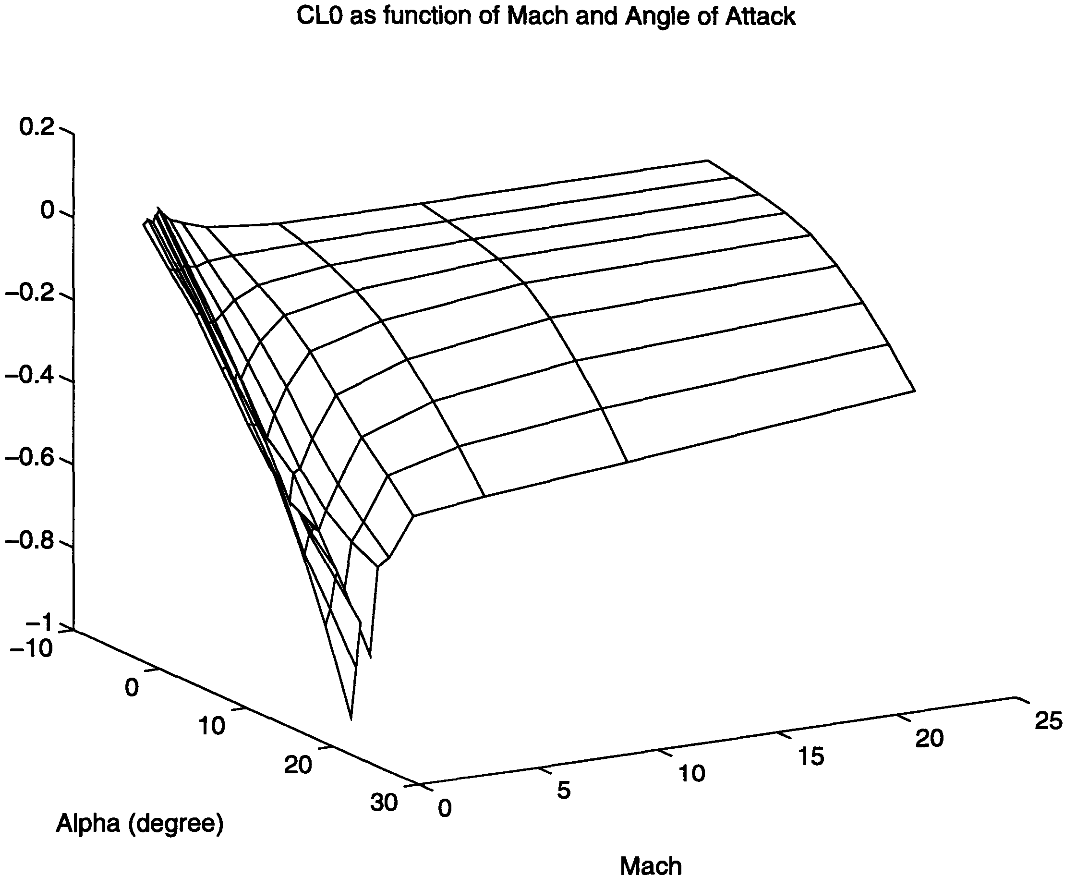


Figure B-11: Aerodynamics coefficient C_{L_0} as a function of angle of attack α and Mach number M

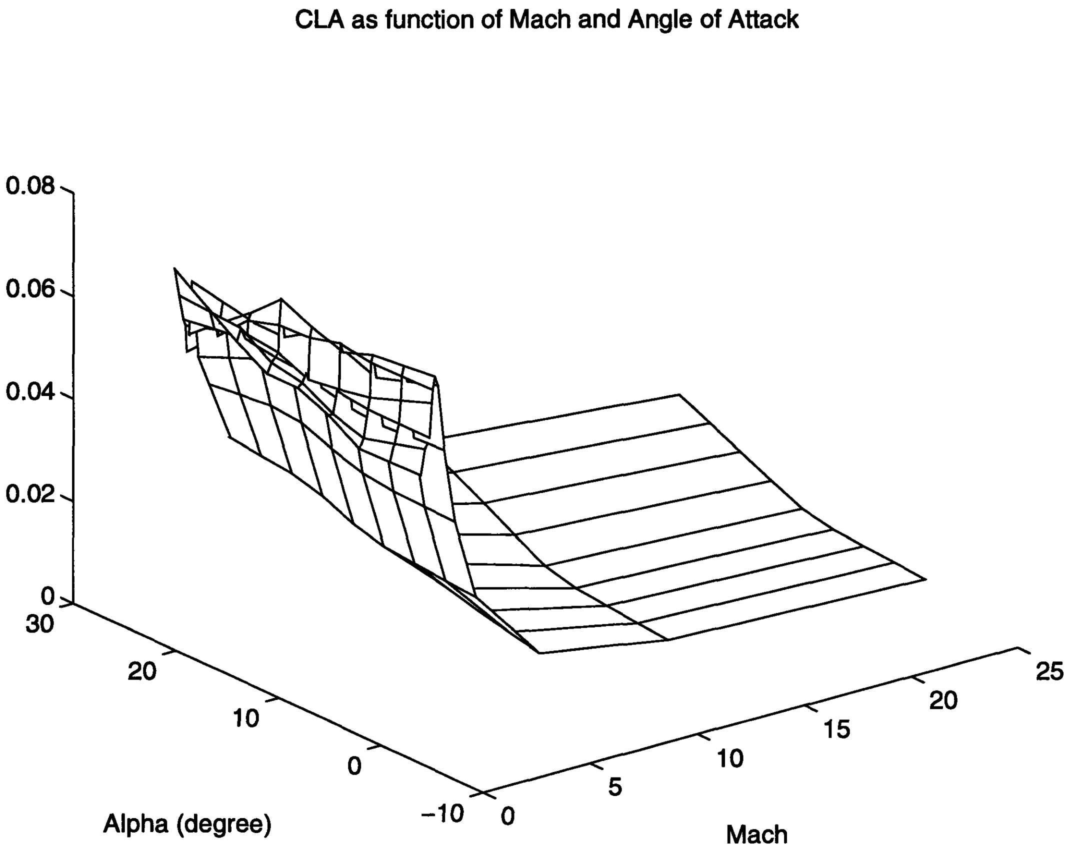


Figure B-12: Aerodynamics coefficient C_{L_A} as a function of angle of attack α and Mach number M

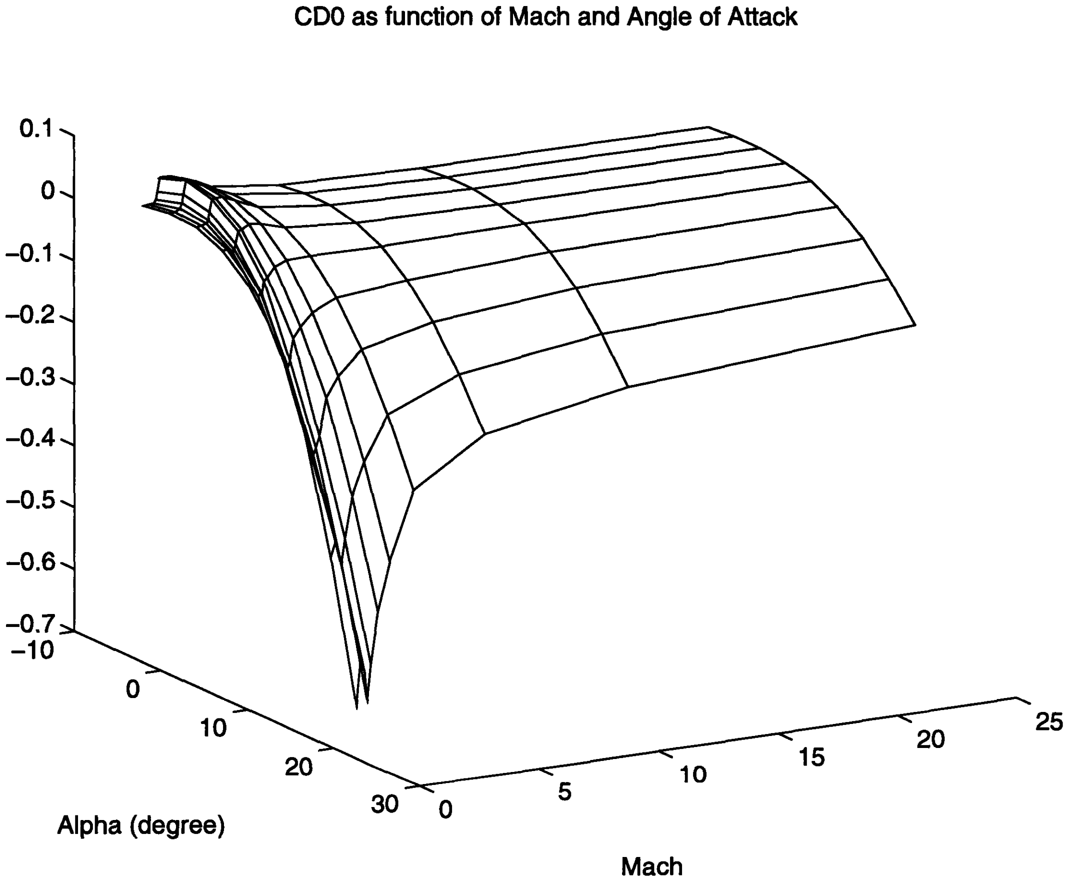


Figure B-13: Aerodynamics coefficient C_{D_0} as a function of angle of attack α and Mach number M

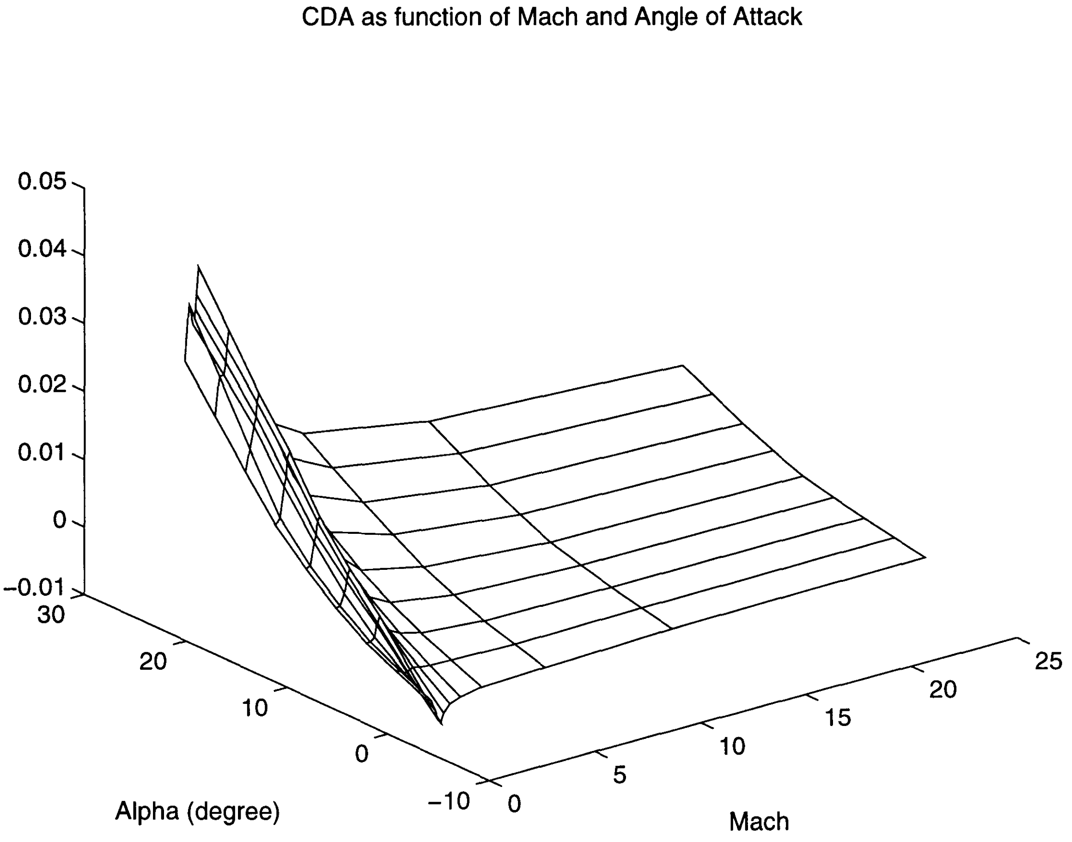


Figure B-14: Aerodynamics coefficient C_{D_A} as a function of angle of attack α and Mach number M

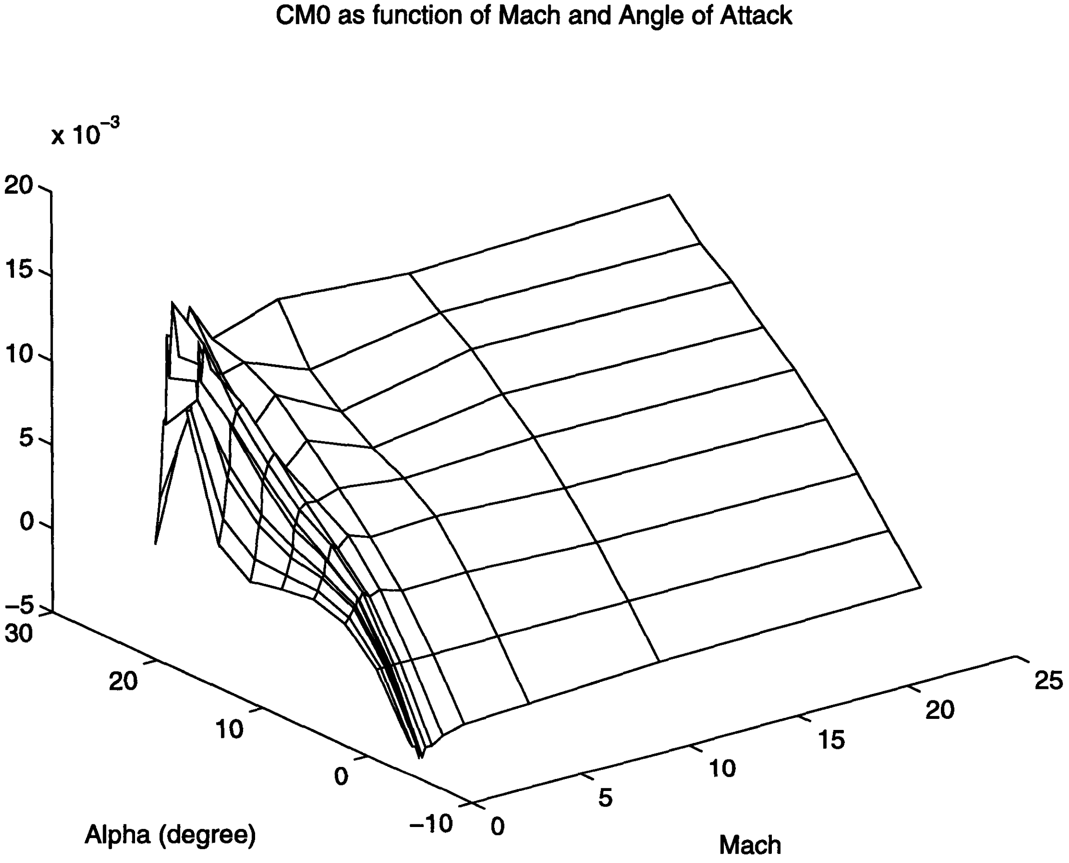


Figure B-15: Aerodynamics coefficient C_{M_0} as a function of angle of attack α and Mach number M

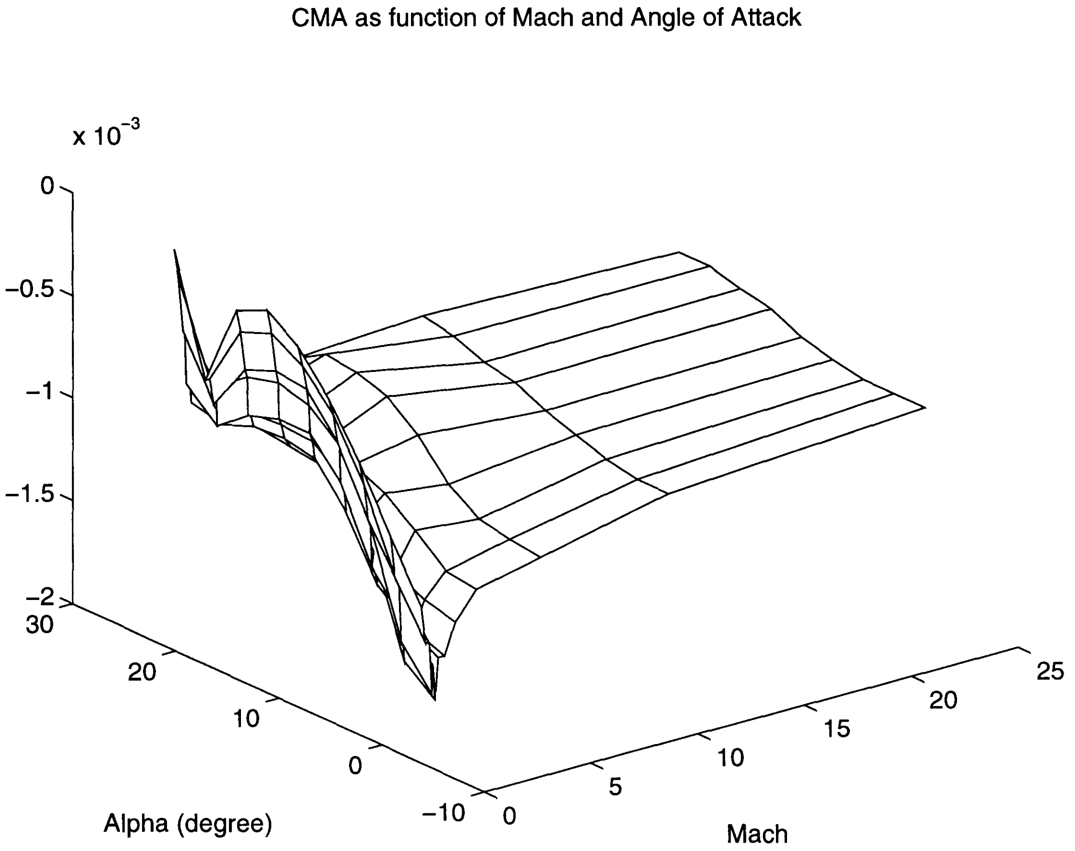


Figure B-16: Aerodynamics coefficient C_{M_A} as a function of angle of attack α and Mach number M

Appendix C

Aerodynamics Data Polynomial Approximation

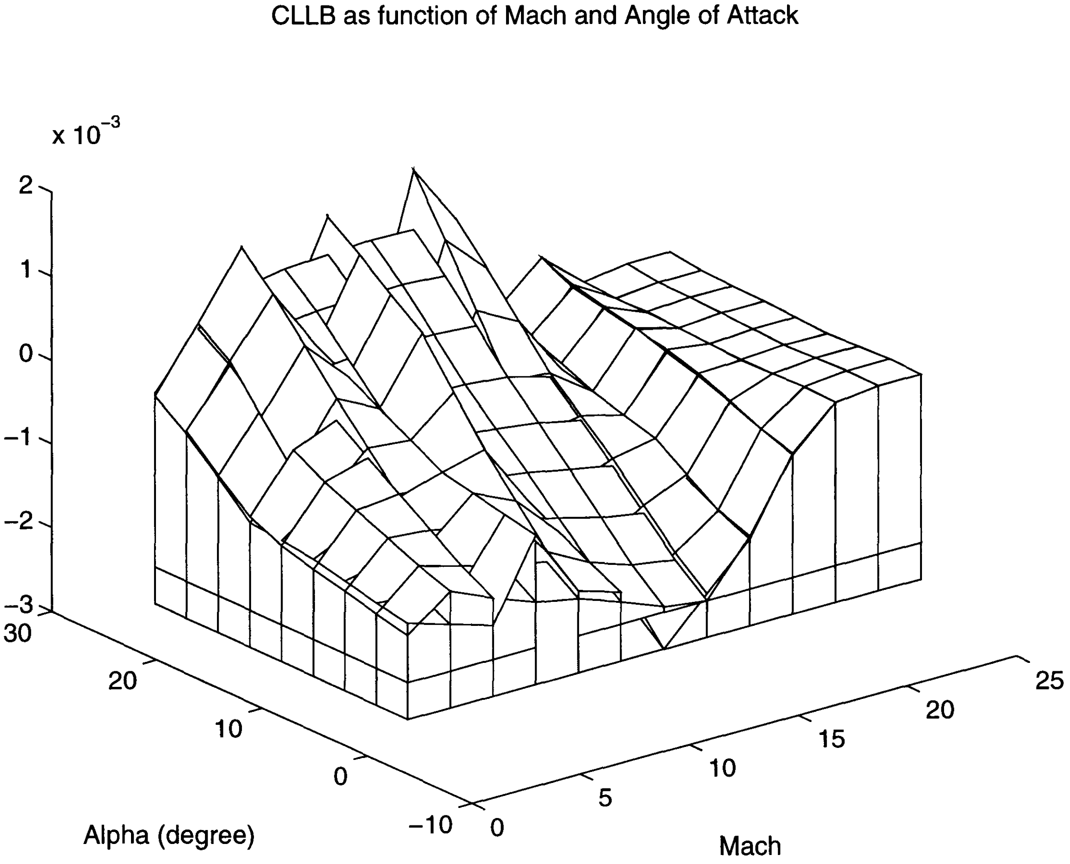


Figure C-1: Aerodynamics coefficient $C_{L_L B}$ as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

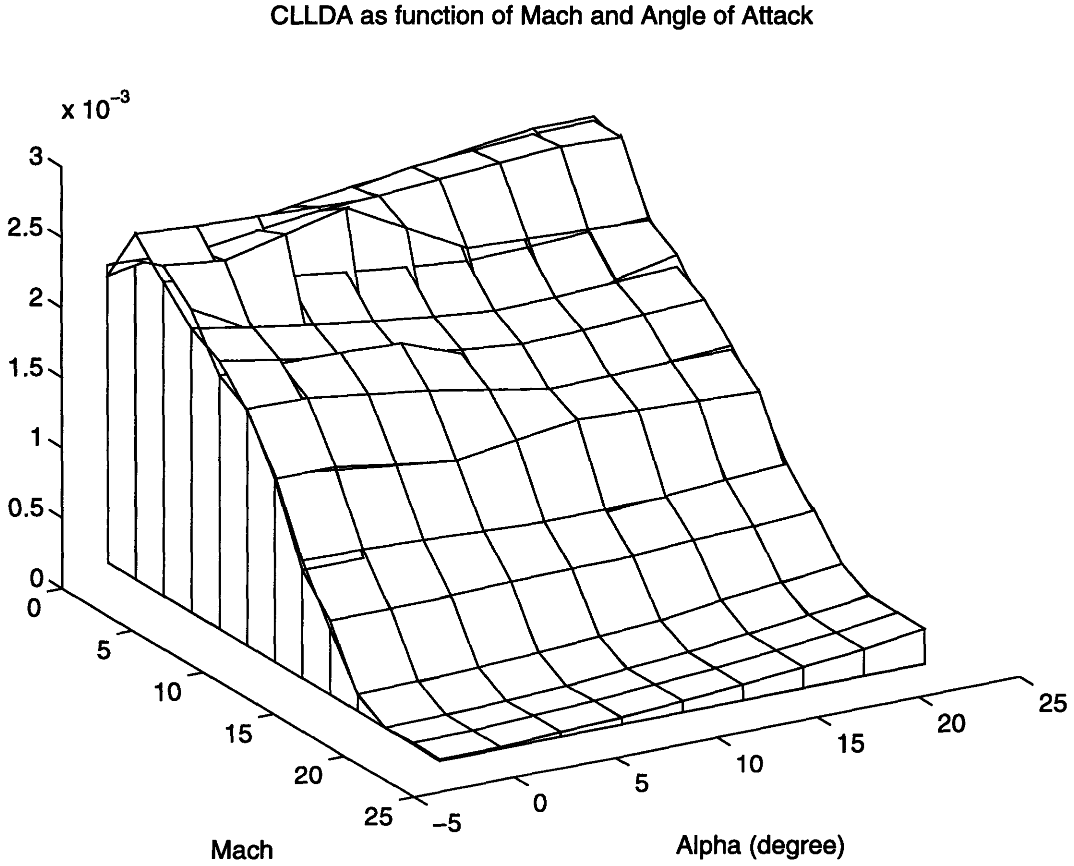


Figure C-2: Aerodynamics coefficient C_{L_LDA} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

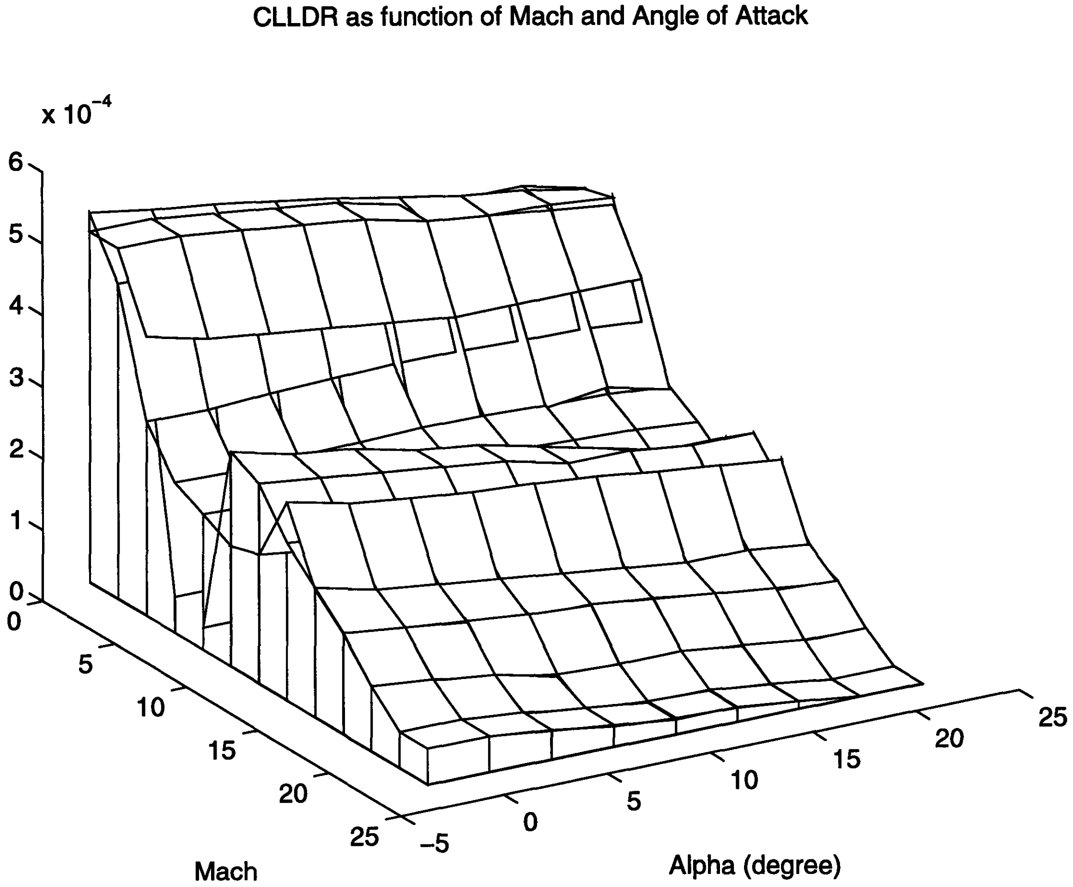


Figure C-3: Aerodynamics coefficient C_{L_LDR} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

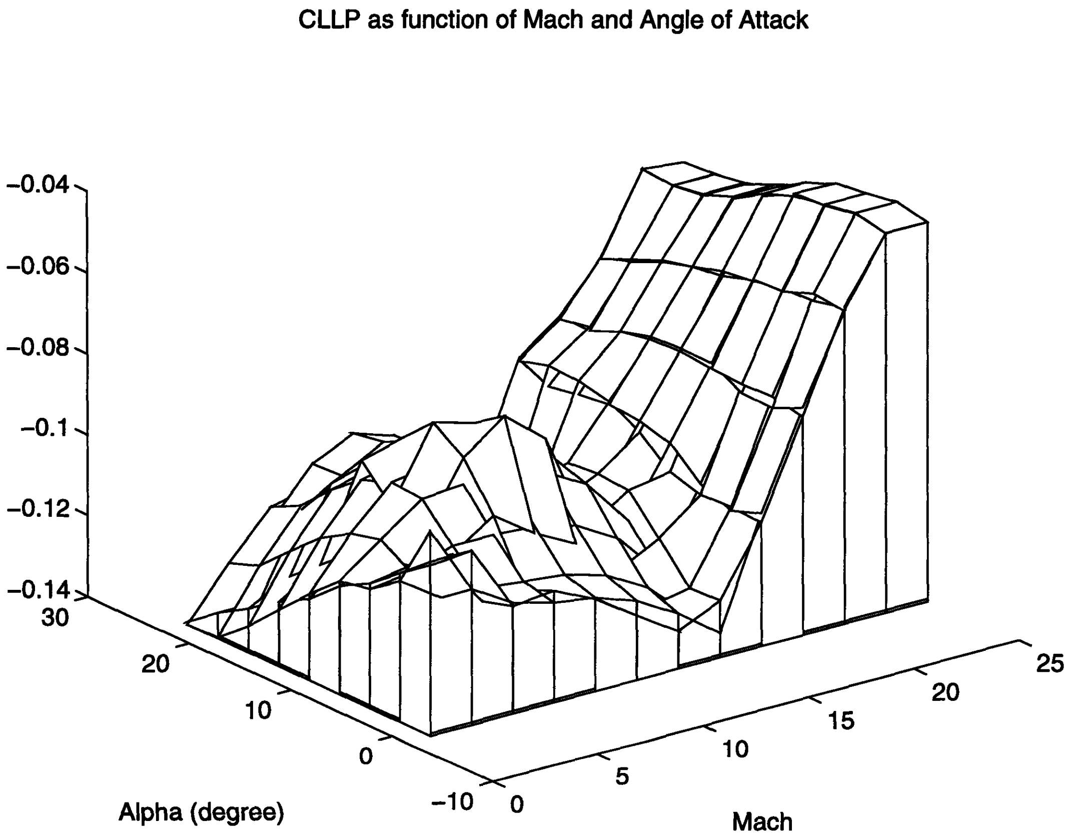


Figure C-4: Aerodynamics coefficient $C_{L_L P}$ as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

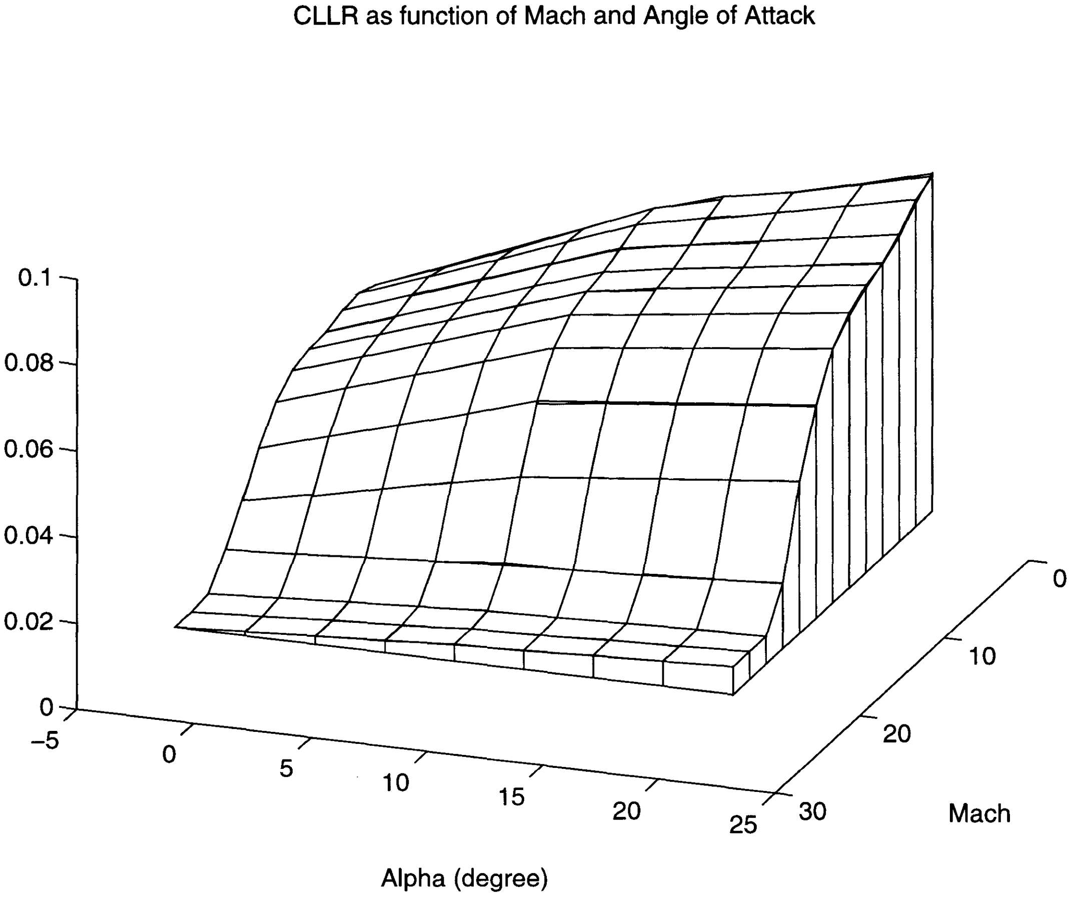


Figure C-5: Aerodynamics coefficient $C_{L_L R}$ as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

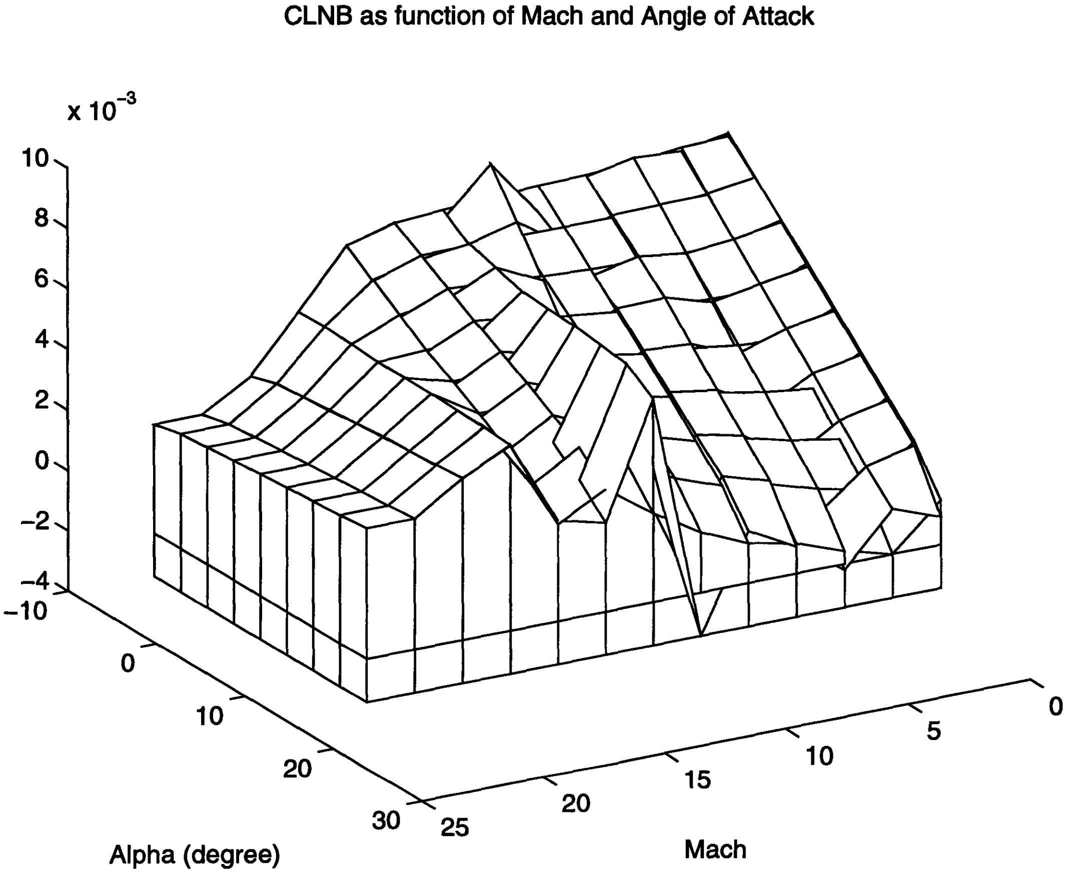


Figure C-6: Aerodynamics coefficient C_{L_NB} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

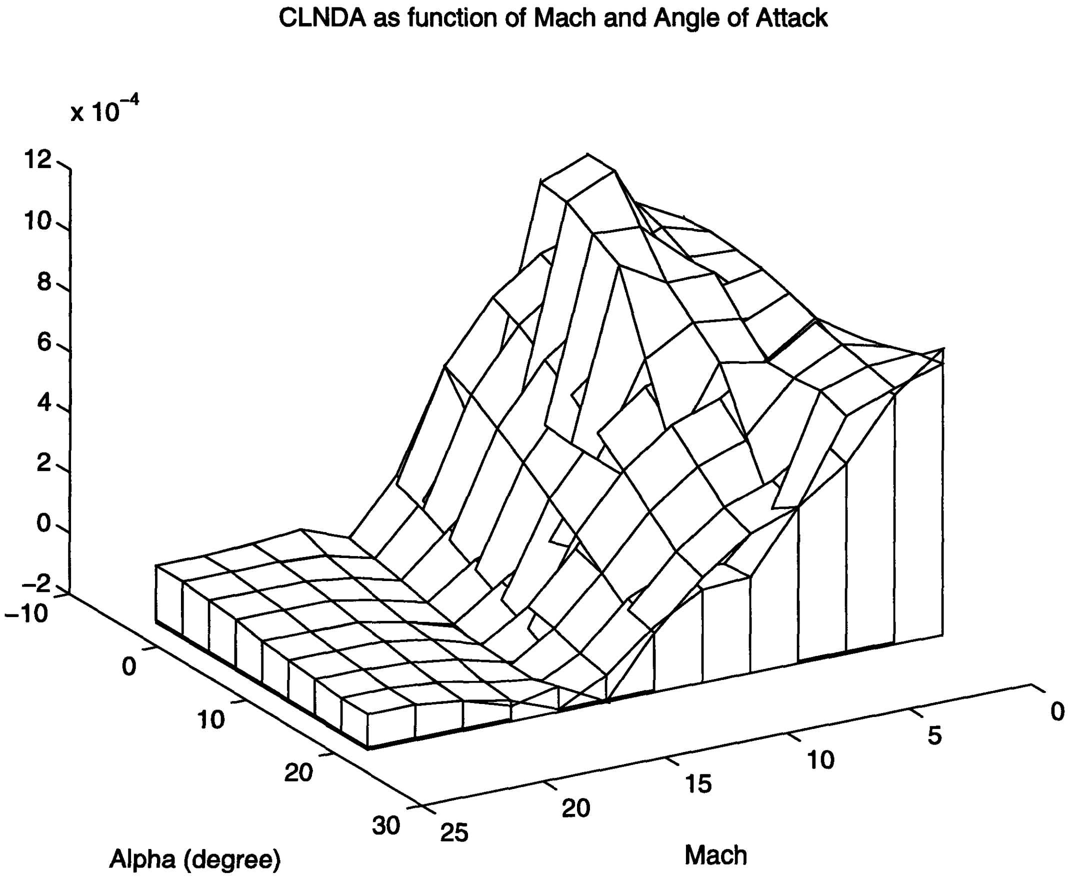


Figure C-7: Aerodynamics coefficient $C_{L_N DA}$ as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

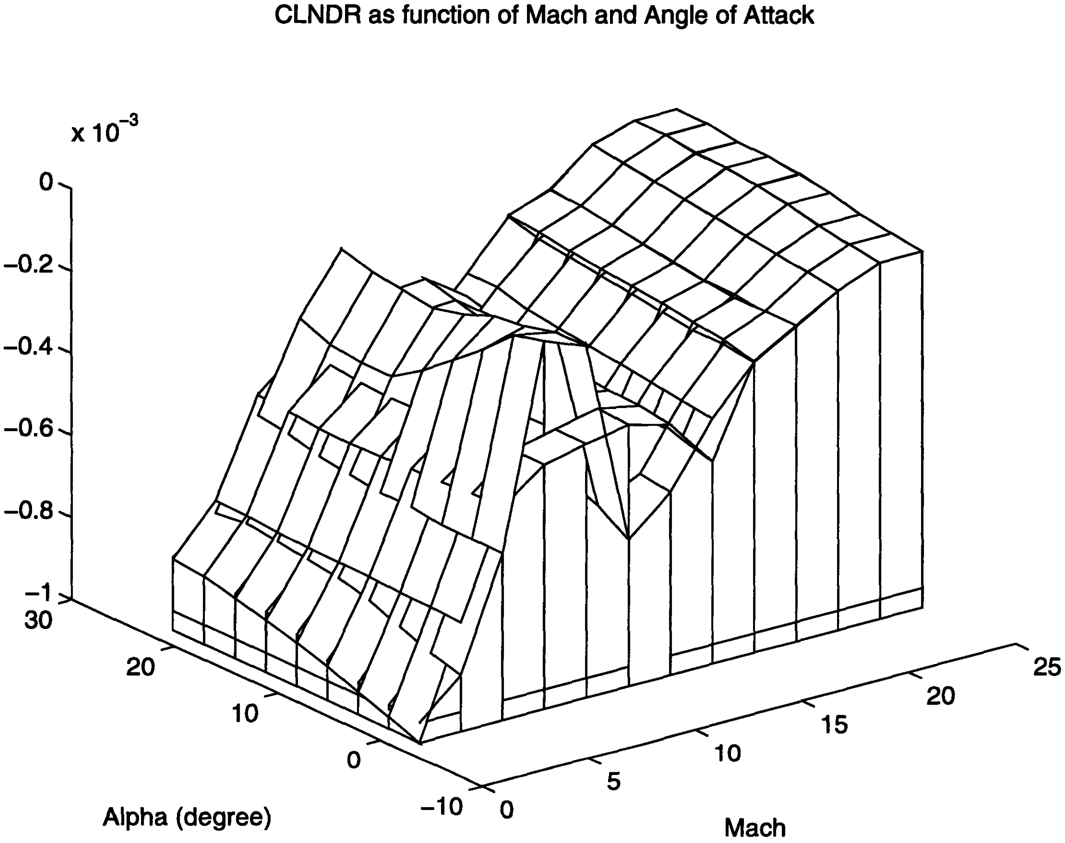


Figure C-8: Aerodynamics coefficient C_{L_NDR} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

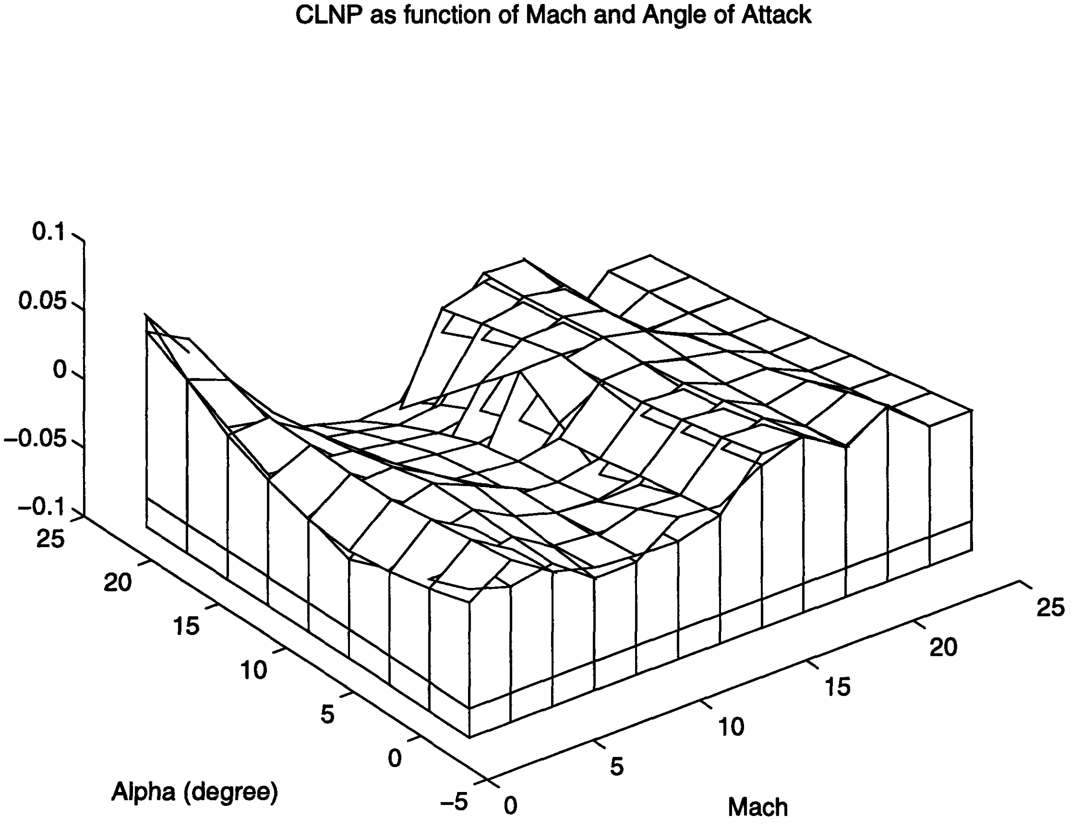


Figure C-9: Aerodynamics coefficient $C_{L_N P}$ as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

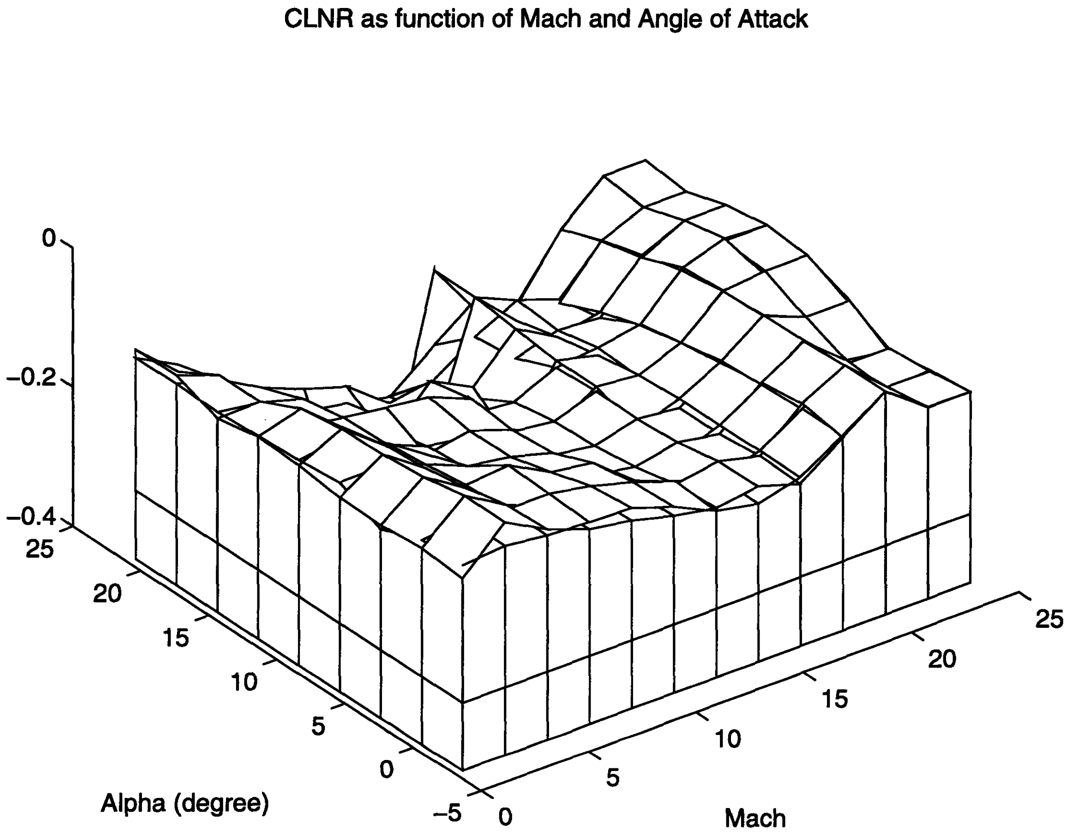


Figure C-10: Aerodynamics coefficient C_{L_NR} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

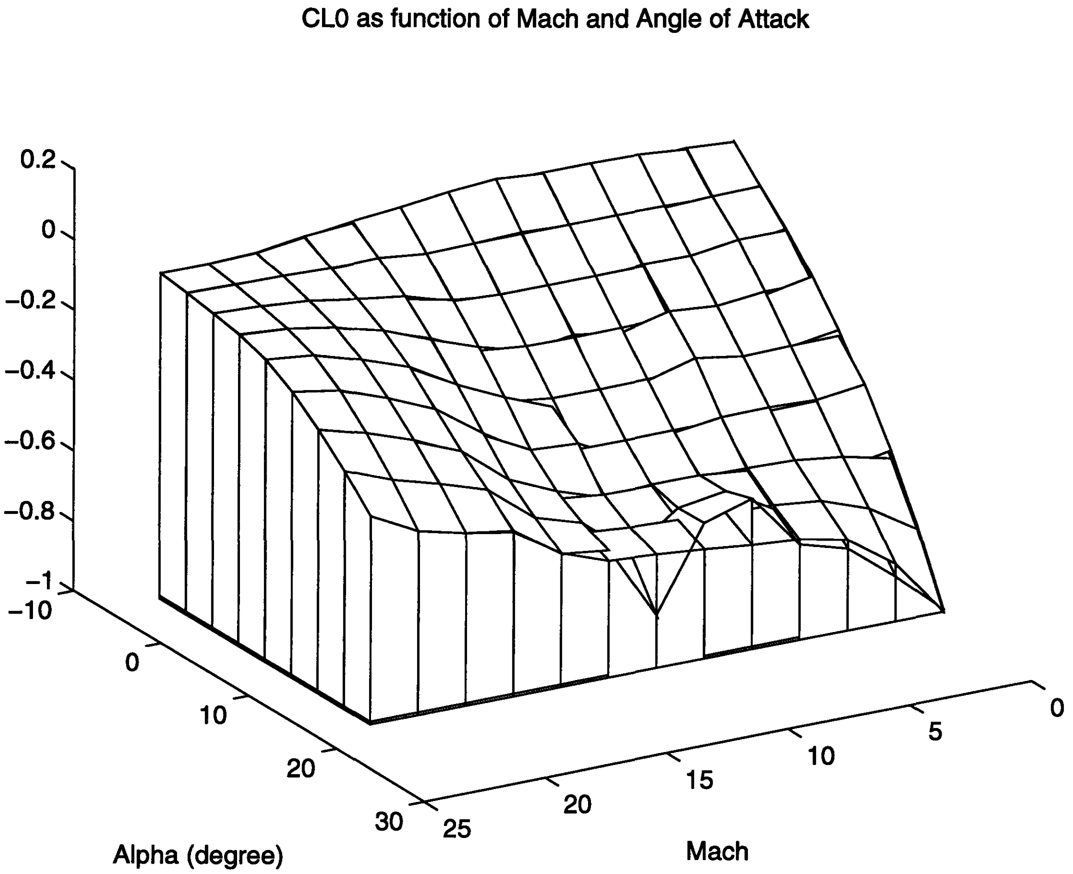


Figure C-11: Aerodynamics coefficient C_{L_0} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

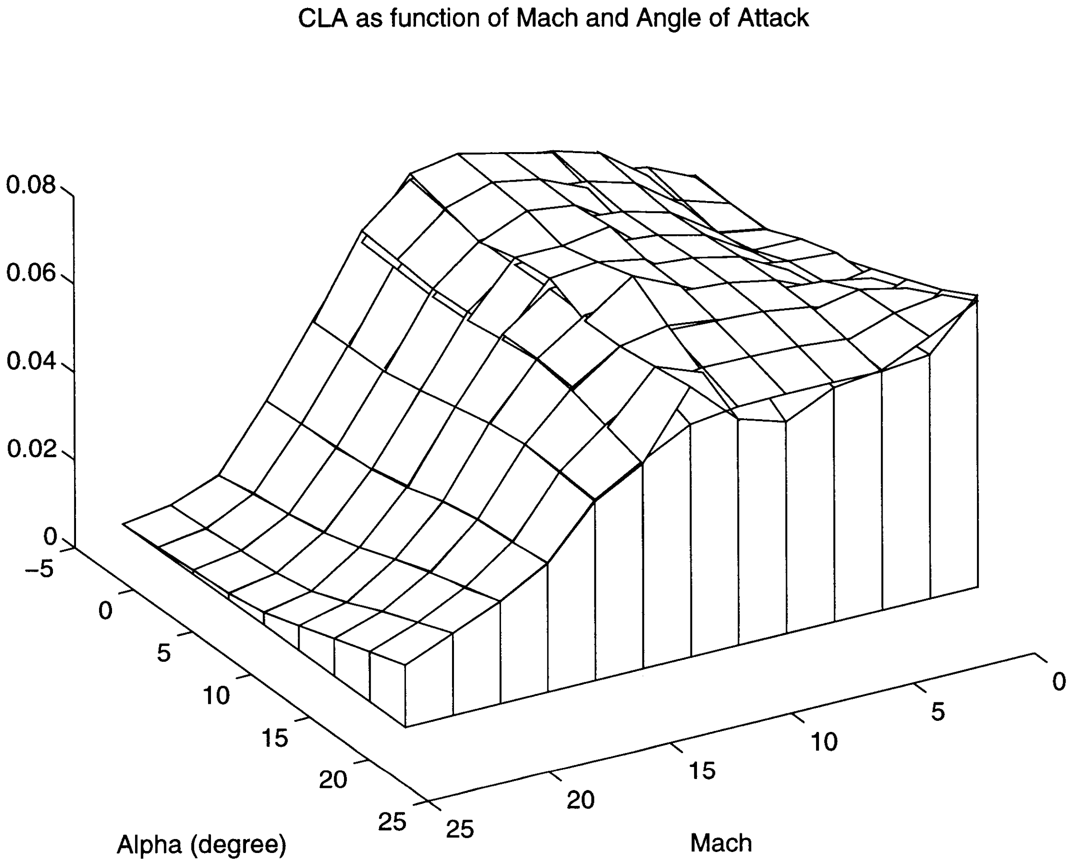


Figure C-12: Aerodynamics coefficient C_{L_A} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

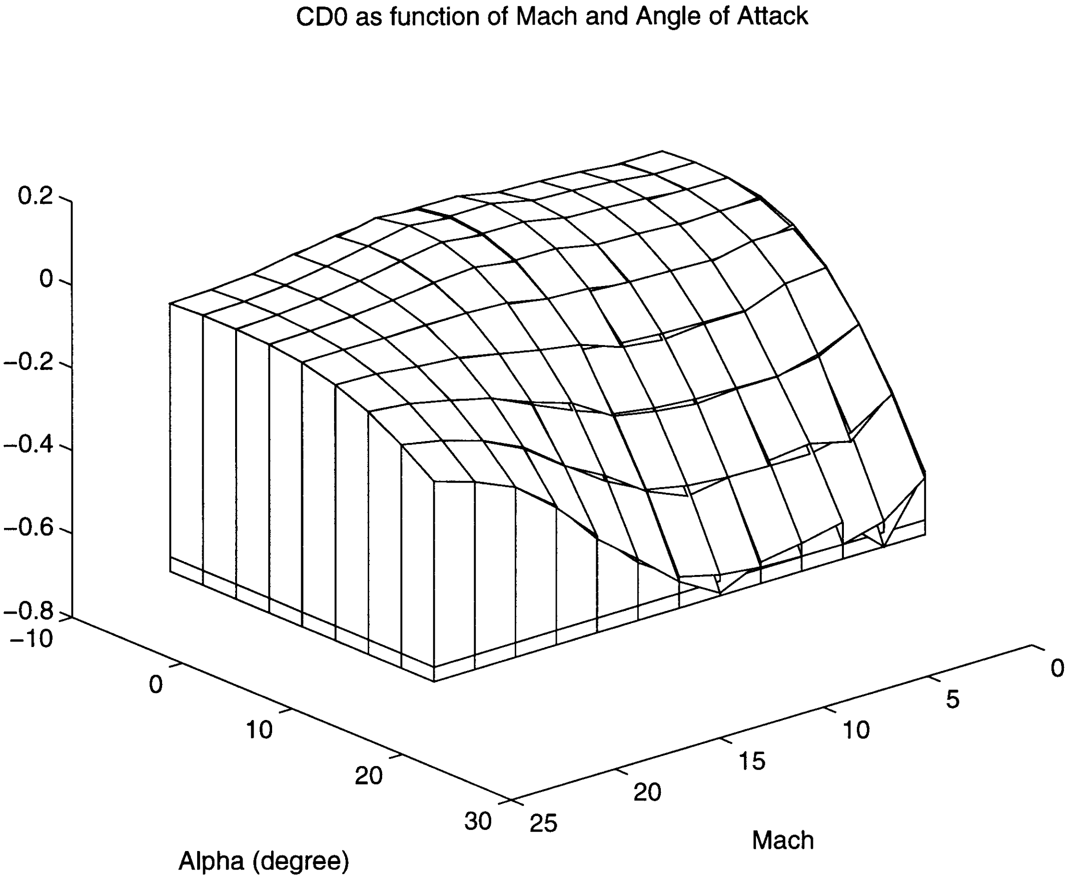


Figure C-13: Aerodynamics coefficient C_{D_0} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

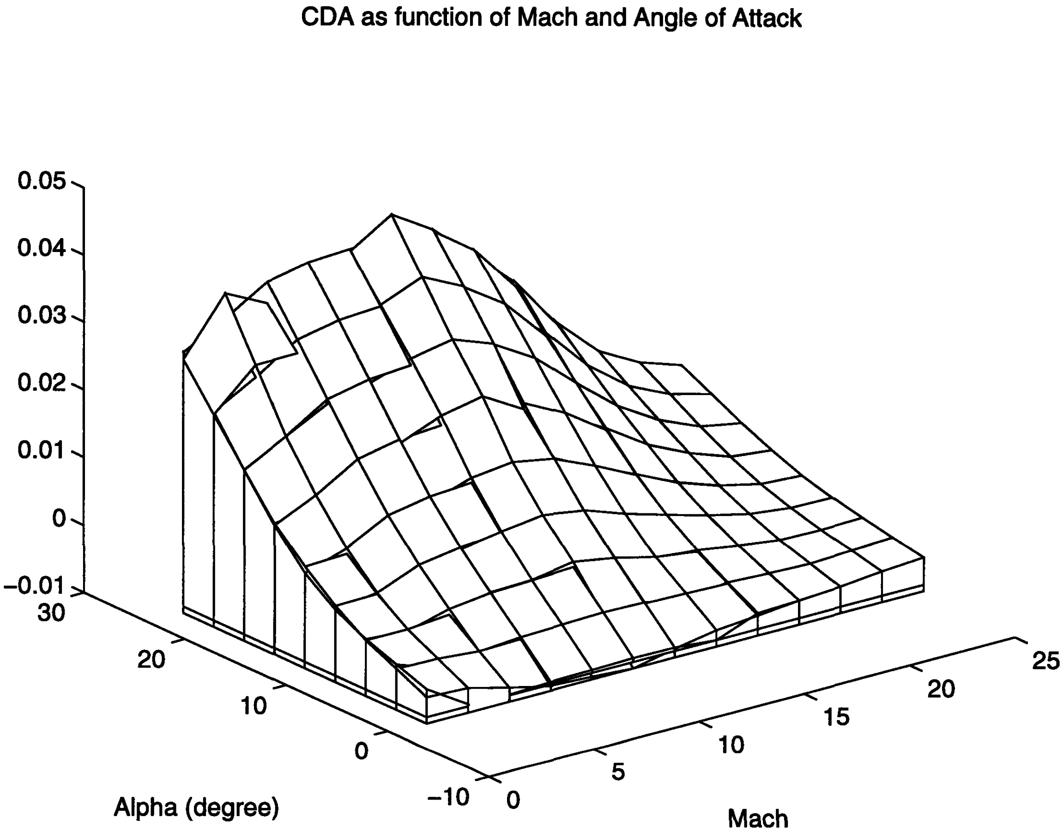


Figure C-14: Aerodynamics coefficient C_{D_A} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

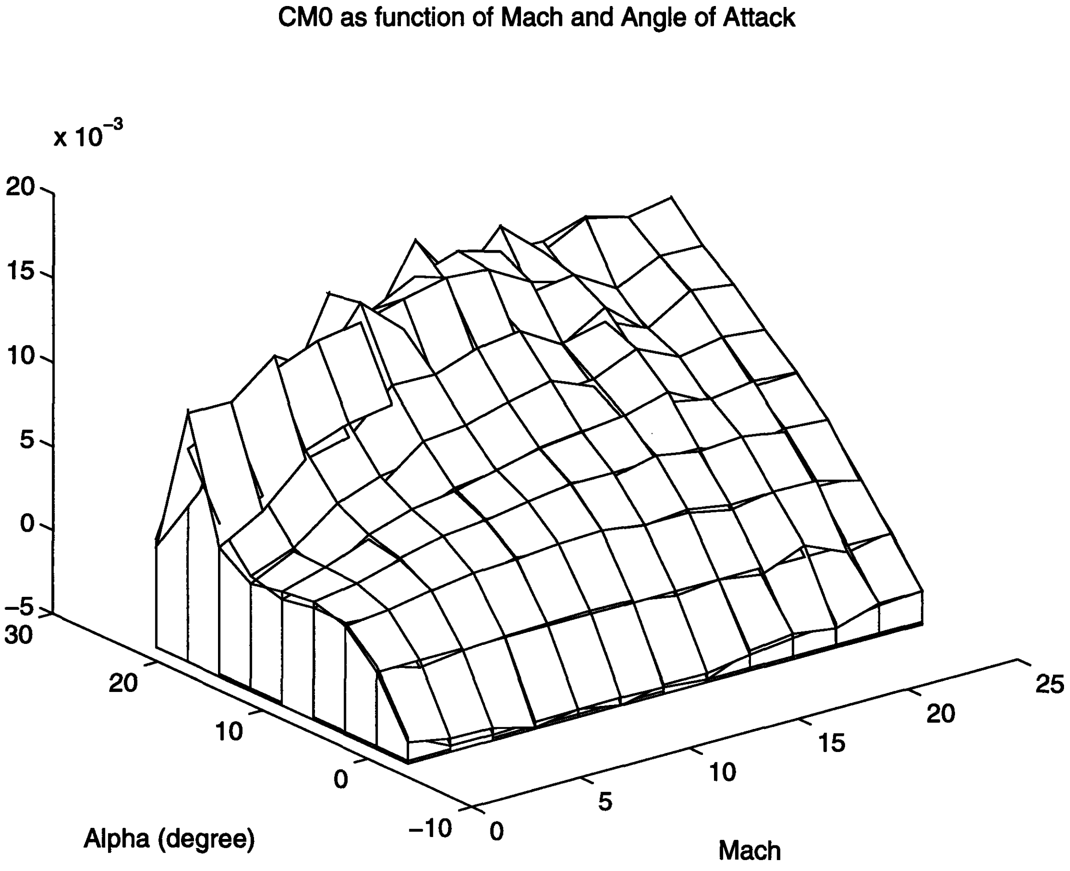


Figure C-15: Aerodynamics coefficient C_{M_0} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

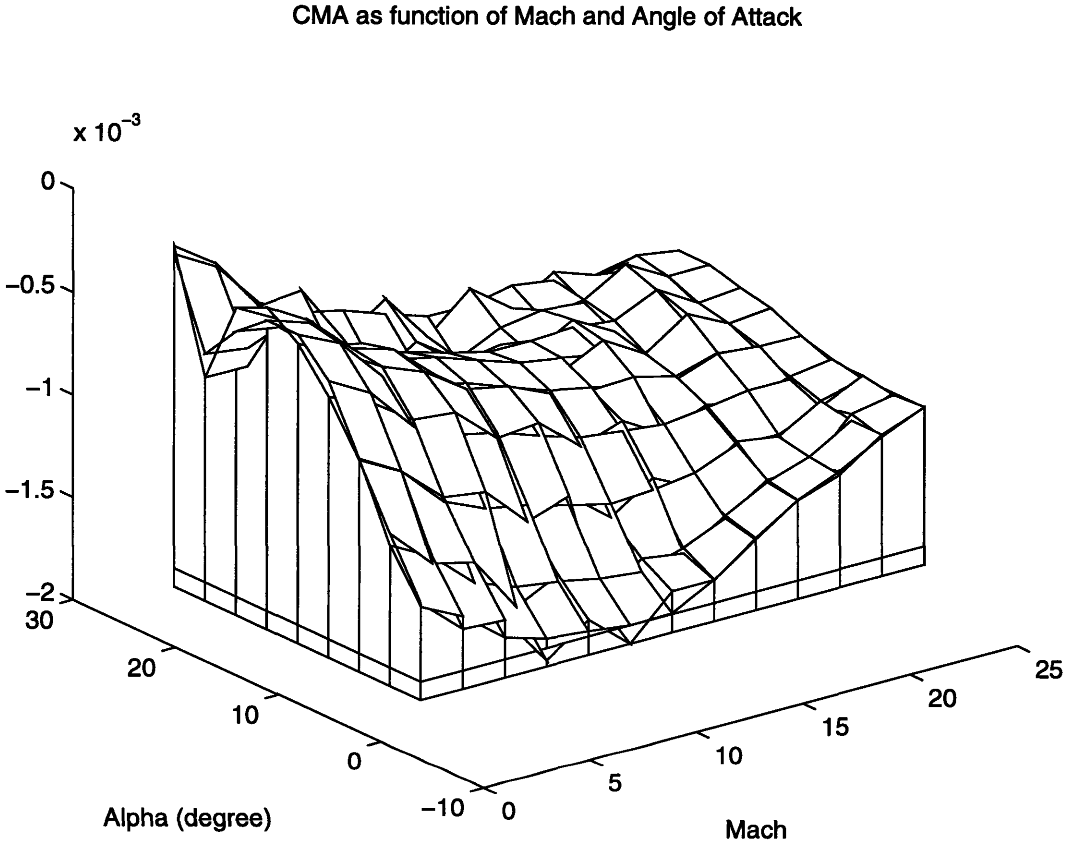


Figure C-16: Aerodynamics coefficient C_{M_A} as a function of angle of attack α and Mach number M : exact versus 2-D polynomial approximation

Appendix D

GHAME numerical data

The aerodynamic data of GHAME were originally presented in an array of 9 by 13. This needed two lines of text per line of the array. For the sake of clarity, here the data are given as 13x9. The data varies as a function of angle of attack $\alpha(-3.0, 0.0, 3.0, 6.0, 9.0, 12.0, 15.0, 18.0$ and $21.0)$ across and Mach number M $(0.4, 0.6, 0.8, 0.9, 0.95, 1.05, 1.2, 1.5, 2.0, 3.0, 6.0, 12.0$ and $24.0)$ respectively.

C_{L_0}

```
0.04508 -0.03575 -0.11693 -0.22054 -0.32599 -0.41960 -0.56105 -0.72859 -0.92320
0.05491 -0.03810 -0.11690 -0.21782 -0.32773 -0.43704 -0.58493 -0.68972 -0.80347
0.04723 -0.04375 -0.13532 -0.23228 -0.33138 -0.43095 -0.53773 -0.61593 -0.69635
0.06222 -0.03800 -0.13317 -0.22108 -0.30955 -0.43224 -0.53748 -0.60137 -0.65836
0.06411 -0.03800 -0.14505 -0.25401 -0.34726 -0.44946 -0.52428 -0.52399 -0.50020
0.07569 -0.03150 -0.13861 -0.23254 -0.35937 -0.49158 -0.52394 -0.54559 -0.54335
0.06492 -0.03675 -0.14068 -0.25132 -0.36570 -0.42414 -0.53246 -0.64640 -0.77975
0.04782 -0.03905 -0.12433 -0.21391 -0.31249 -0.41519 -0.48735 -0.53332 -0.56663
0.03480 -0.02965 -0.09539 -0.16617 -0.24995 -0.34693 -0.42815 -0.49102 -0.54949
0.01583 -0.02475 -0.06776 -0.11392 -0.17259 -0.24957 -0.32392 -0.38885 -0.45683
-0.00175 -0.02075 -0.04593 -0.07910 -0.12465 -0.18941 -0.26223 -0.34431 -0.43640
-0.00547 -0.01871 -0.03483 -0.05409 -0.08445 -0.14139 -0.21567 -0.30273 -0.40437
-0.00468 -0.01562 -0.02893 -0.04481 -0.06985 -0.11688 -0.17836 -0.25036 -0.33438
```

C_{L_α}

0.05483	0.05472	0.05460	0.05827	0.06087	0.06092	0.06440	0.06872	0.07303
0.05972	0.05805	0.05638	0.05833	0.06170	0.06287	0.06678	0.06725	0.06772
0.06170	0.06173	0.06177	0.06255	0.06328	0.06370	0.06455	0.06368	0.06282
0.06832	0.06613	0.06395	0.06275	0.06192	0.06452	0.06533	0.06343	0.06153
0.07085	0.07078	0.07072	0.07170	0.07006	0.06976	0.06792	0.06215	0.05638
0.07335	0.07272	0.07208	0.06993	0.07203	0.07470	0.06927	0.06448	0.05970
0.07145	0.06948	0.06752	0.06987	0.07093	0.06717	0.06722	0.06847	0.06972
0.05858	0.05817	0.05775	0.05830	0.05973	0.06097	0.05978	0.05725	0.05472
0.04392	0.04417	0.04442	0.04527	0.04720	0.04957	0.05017	0.04950	0.04882
0.02805	0.02854	0.02904	0.02967	0.03138	0.03409	0.03582	0.03651	0.03720
0.01379	0.01513	0.01647	0.01815	0.02038	0.02342	0.02613	0.02863	0.03112
0.00956	0.01012	0.01068	0.01139	0.01297	0.01630	0.01986	0.02319	0.02652
0.00794	0.00840	0.00886	0.00946	0.01076	0.01350	0.01645	0.01921	0.02197

C_{m_q}

-2.71900	-2.73700	-2.76900	-2.76400	-2.71100	-2.93800	-1.71700	-1.20300	-1.04100
-2.74800	-2.82400	-2.79800	-2.82300	-2.79800	-2.57200	-1.94100	-1.26100	-1.18500
-3.00000	-3.02500	-3.05100	-3.22700	-3.50300	-3.63000	-3.05100	-2.52100	-2.34400
-3.60000	-3.55400	-3.80600	-4.33600	-4.56500	-4.08400	-3.20200	-2.77300	-3.00000
-3.50300	-3.58100	-4.46600	-5.39600	-4.15700	-2.87400	-2.09300	-1.71400	-1.84100
-3.25100	-3.27700	-3.50500	-3.68100	-3.30300	-2.16700	-1.23500	-0.95700	-1.58800
-3.05000	-3.07600	-3.05100	-3.12500	-3.90700	-3.83200	-1.48800	-1.00800	-1.23500
-3.20000	-3.10000	-3.10000	-3.10000	-3.10000	-3.20000	-3.00000	-1.20000	-1.00000
-2.40000	-2.40000	-2.40000	-2.40000	-2.40000	-2.40000	-2.20000	-1.90000	-1.70000
-1.80000	-1.80000	-1.80000	-1.80000	-1.70000	-1.70000	-1.70000	-1.80000	-1.80000
-1.60000	-1.60000	-1.60000	-1.60000	-1.60000	-1.60000	-1.60000	-1.60000	-1.60000
-0.27800	-0.65300	-0.87600	-0.99800	-1.12200	-1.34500	-1.72200	-1.99300	-2.26700
-0.27800	-0.65300	-0.87600	-0.99800	-1.12200	-1.34500	-1.72200	-1.99300	-2.26700

C_{m_0}

-0.00332	0.00000	0.00187	0.00245	0.00221	0.00182	0.00315	0.01029	0.00149
-0.00379	0.00000	0.00216	0.00288	0.00298	0.00308	0.00463	0.01030	0.00406
-0.00383	0.00000	0.00263	0.00360	0.00398	0.00502	0.00702	0.01232	0.00873
-0.00449	0.00000	0.00270	0.00376	0.00454	0.00562	0.00767	0.01080	0.00846
-0.00411	0.00000	0.00322	0.00468	0.00565	0.00760	0.01008	0.01408	0.01373
-0.00457	0.00000	0.00304	0.00457	0.00599	0.00785	0.01001	0.01182	0.01117
-0.00403	0.00000	0.00347	0.00545	0.00670	0.00857	0.01087	0.01396	0.01555
-0.00403	0.00000	0.00310	0.00512	0.00699	0.00912	0.01121	0.01241	0.01229
-0.00364	0.00000	0.00335	0.00571	0.00691	0.00809	0.00970	0.01193	0.01501
-0.00331	0.00000	0.00292	0.00526	0.00734	0.00932	0.01121	0.01229	0.01280
-0.00312	0.00000	0.00300	0.00546	0.00686	0.00783	0.00915	0.01078	0.01409
-0.00275	0.00000	0.00267	0.00507	0.00721	0.00895	0.01075	0.01203	0.01349
-0.00254	0.00000	0.00254	0.00493	0.00706	0.00869	0.01051	0.01195	0.01395

C_{m_α}

-0.00127	-0.00100	-0.00071	-0.00047	-0.00028	-0.00017	-0.00024	-0.00065	-0.00008
-0.00145	-0.00120	-0.00082	-0.00055	-0.00038	-0.00029	-0.00035	-0.00065	-0.00022
-0.00146	-0.00125	-0.00100	-0.00069	-0.00050	-0.00048	-0.00054	-0.00078	-0.00048
-0.00171	-0.00152	-0.00103	-0.00071	-0.00058	-0.00054	-0.00058	-0.00069	-0.00046
-0.00157	-0.00142	-0.00123	-0.00089	-0.00072	-0.00072	-0.00077	-0.00089	-0.00075
-0.00174	-0.00162	-0.00116	-0.00087	-0.00076	-0.00075	-0.00076	-0.00075	-0.00061
-0.00154	-0.00145	-0.00132	-0.00104	-0.00085	-0.00082	-0.00083	-0.00089	-0.00085
-0.00154	-0.00148	-0.00118	-0.00097	-0.00089	-0.00087	-0.00085	-0.00079	-0.00067
-0.00139	-0.00135	-0.00128	-0.00109	-0.00088	-0.00077	-0.00074	-0.00076	-0.00082
-0.00126	-0.00124	-0.00111	-0.00100	-0.00093	-0.00089	-0.00085	-0.00078	-0.00070
-0.00119	-0.00117	-0.00114	-0.00104	-0.00087	-0.00075	-0.00070	-0.00068	-0.00077
-0.00105	-0.00105	-0.00102	-0.00097	-0.00092	-0.00085	-0.00082	-0.00076	-0.00073
-0.00097	-0.00097	-0.00097	-0.00094	-0.00090	-0.00083	-0.00080	-0.00076	-0.00076

C_{D_0}

0.02941	0.03594	0.03261	0.01436	-0.02306	-0.08415	-0.18718	-0.31882	-0.47897
0.03035	0.03714	0.03394	0.01489	-0.02386	-0.09375	-0.23091	-0.40962	-0.62884
0.03127	0.04141	0.03647	0.01049	-0.04433	-0.13162	-0.25573	-0.38482	-0.54072
0.03633	0.04936	0.04104	0.01345	-0.03845	-0.13204	-0.25708	-0.39617	-0.56164
0.04871	0.06265	0.05359	0.02174	-0.04007	-0.14179	-0.26424	-0.40820	-0.57770
0.06964	0.08447	0.07359	0.04174	-0.02090	-0.12953	-0.25284	-0.42320	-0.62233
0.07106	0.08354	0.07468	0.04369	-0.01598	-0.09461	-0.23458	-0.38219	-0.56587
0.07076	0.08047	0.07480	0.05000	0.00010	-0.07789	-0.17409	-0.31589	-0.48309
0.06228	0.07067	0.06681	0.04731	0.00808	-0.05773	-0.14190	-0.26256	-0.40512
0.04472	0.05161	0.04947	0.03597	0.00925	-0.03957	-0.10402	-0.19187	-0.29579
0.02961	0.03553	0.03502	0.02594	0.00672	-0.02957	-0.07789	-0.14391	-0.22221
0.02122	0.02584	0.02583	0.02021	0.00797	-0.01836	-0.05742	-0.11319	-0.17988
0.01766	0.02150	0.02150	0.01685	0.00672	-0.01507	-0.04744	-0.09366	-0.14894

C_{D_α}

-0.00323	-0.00107	0.00110	0.00415	0.00794	0.01227	0.01816	0.02424	0.03033
-0.00348	-0.00120	0.00109	0.00425	0.00820	0.01323	0.02137	0.02993	0.03849
-0.00507	-0.00173	0.00160	0.00595	0.01150	0.01769	0.02453	0.02995	0.03537
-0.00646	-0.00157	0.00331	0.00764	0.01248	0.01907	0.02588	0.03170	0.03752
-0.00677	-0.00163	0.00352	0.00858	0.01444	0.02153	0.02791	0.03380	0.03969
-0.00716	-0.00132	0.00453	0.00939	0.01514	0.02275	0.02910	0.03633	0.04357
-0.00588	-0.00121	0.00347	0.00838	0.01402	0.01922	0.02683	0.03326	0.03969
-0.00475	-0.00137	0.00201	0.00607	0.01100	0.01645	0.02150	0.02777	0.03404
-0.00419	-0.00151	0.00117	0.00448	0.00841	0.01309	0.01766	0.02305	0.02844
-0.00356	-0.00158	0.00040	0.00281	0.00554	0.00908	0.01269	0.01663	0.02057
-0.00319	-0.00180	-0.00041	0.00139	0.00347	0.00619	0.00896	0.01197	0.01498
-0.00256	-0.00153	-0.00051	0.00068	0.00204	0.00407	0.00640	0.00904	0.01169
-0.00213	-0.00128	-0.00043	0.00056	0.00169	0.00336	0.00529	0.00749	0.00968

$C_{l_{l\beta}}$

-0.00128	-0.00121	-0.00112	-0.00103	-0.00098	-0.00094	-0.00057	-0.00022	0.00006
-0.00143	-0.00125	-0.00102	-0.00080	-0.00063	-0.00047	0.00002	0.00051	0.00079
-0.00158	-0.00129	-0.00093	-0.00057	-0.00015	0.00027	0.00073	0.00117	0.00155
-0.00065	-0.00057	-0.00049	-0.00041	-0.00020	0.00001	0.00022	0.00042	0.00055
-0.00145	-0.00115	-0.00072	-0.00028	0.00024	0.00077	0.00106	0.00136	0.00164
-0.00160	-0.00142	-0.00122	-0.00102	-0.00083	-0.00065	-0.00041	-0.00018	0.00006
-0.00242	-0.00195	-0.00149	-0.00102	-0.00055	-0.00008	0.00059	0.00126	0.00191
-0.00190	-0.00158	-0.00119	-0.00080	-0.00046	-0.00012	0.00032	0.00077	0.00119
-0.00139	-0.00122	-0.00096	-0.00070	-0.00059	-0.00048	-0.00038	-0.00028	-0.00012
-0.00050	-0.00038	-0.00023	-0.00009	0.00002	0.00014	0.00022	0.00031	0.00045
-0.00003	-0.00002	-0.00003	-0.00003	-0.00002	0.00000	0.00001	0.00001	0.00003
0.00003	0.00004	0.00002	0.00000	0.00002	0.00004	0.00005	0.00006	0.00011
0.00003	0.00004	0.00002	0.00000	0.00002	0.00004	0.00005	0.00006	0.00011

$C_{l_{l\delta a}}$

0.00227	0.00231	0.00235	0.00240	0.00246	0.00253	0.00256	0.00259	0.00260
0.00242	0.00240	0.00246	0.00251	0.00253	0.00256	0.00259	0.00262	0.00260
0.00250	0.00245	0.00255	0.00265	0.00242	0.00219	0.00217	0.00215	0.00213
0.00231	0.00207	0.00174	0.00142	0.00151	0.00160	0.00176	0.00191	0.00200
0.00195	0.00195	0.00194	0.00192	0.00176	0.00160	0.00163	0.00165	0.00172
0.00183	0.00182	0.00176	0.00170	0.00162	0.00154	0.00155	0.00156	0.00156
0.00145	0.00145	0.00138	0.00132	0.00138	0.00144	0.00142	0.00140	0.00138
0.00092	0.00092	0.00092	0.00092	0.00092	0.00092	0.00092	0.00092	0.00092
0.00068	0.00068	0.00068	0.00068	0.00069	0.00070	0.00072	0.00074	0.00077
0.00026	0.00027	0.00029	0.00031	0.00034	0.00037	0.00041	0.00046	0.00051
0.00015	0.00015	0.00017	0.00019	0.00022	0.00026	0.00030	0.00035	0.00040
0.00016	0.00016	0.00018	0.00020	0.00023	0.00026	0.00030	0.00034	0.00040
0.00016	0.00016	0.00018	0.00020	0.00023	0.00026	0.00030	0.00034	0.00040

$C_{l_l \delta r}$

0.00051	0.00051	0.00050	0.00049	0.00048	0.00046	0.00045	0.00045	0.00043
0.00051	0.00051	0.00050	0.00049	0.00048	0.00046	0.00045	0.00044	0.00043
0.00041	0.00039	0.00038	0.00037	0.00036	0.00035	0.00035	0.00035	0.00035
0.00007	0.00007	0.00011	0.00014	0.00018	0.00022	0.00023	0.00024	0.00022
0.00005	0.00005	0.00006	0.00008	0.00009	0.00010	0.00009	0.00008	0.00007
0.00032	0.00030	0.00029	0.00027	0.00026	0.00024	0.00022	0.00020	0.00020
0.00030	0.00029	0.00028	0.00027	0.00026	0.00024	0.00024	0.00023	0.00023
0.00024	0.00023	0.00022	0.00021	0.00020	0.00020	0.00019	0.00018	0.00018
0.00020	0.00018	0.00017	0.00016	0.00015	0.00014	0.00013	0.00013	0.00012
0.00016	0.00015	0.00014	0.00013	0.00013	0.00012	0.00011	0.00010	0.00009
0.00011	0.00010	0.00009	0.00007	0.00007	0.00007	0.00006	0.00005	0.00005
0.00007	0.00007	0.00006	0.00005	0.00004	0.00004	0.00003	0.00002	0.00002
0.00007	0.00007	0.00006	0.00005	0.00004	0.00004	0.00003	0.00002	0.00002

$C_{l_l p}$

-0.09435	-0.10059	-0.10651	-0.11054	-0.11712	-0.12118	-0.12627	-0.13076	-0.13642
-0.09435	-0.10059	-0.10488	-0.10727	-0.10872	-0.11495	-0.12118	-0.13076	-0.13315
-0.11066	-0.11301	-0.10488	-0.10460	-0.09994	-0.09196	-0.10108	-0.11111	-0.11734
-0.10971	-0.11305	-0.09819	-0.08717	-0.08189	-0.08911	-0.09819	-0.10537	-0.10587
-0.11446	-0.11400	-0.10526	-0.07960	-0.08478	-0.09196	-0.09625	-0.09964	-0.10108
-0.12126	-0.12168	-0.11058	-0.08797	-0.09416	-0.09937	-0.10169	-0.10401	-0.10442
-0.12795	-0.12551	-0.12016	-0.11290	-0.11047	-0.10895	-0.11126	-0.12221	-0.12168
-0.12312	-0.12350	-0.11476	-0.10640	-0.10564	-0.10640	-0.11628	-0.12274	-0.12198
-0.10678	-0.10640	-0.10678	-0.10450	-0.09804	-0.09386	-0.09234	-0.09196	-0.09424
-0.08322	-0.08170	-0.08208	-0.08170	-0.08094	-0.08132	-0.08284	-0.08512	-0.08664
-0.05890	-0.06080	-0.06118	-0.06270	-0.06384	-0.06612	-0.06840	-0.07182	-0.07410
-0.04347	-0.04199	-0.04332	-0.04469	-0.04750	-0.05221	-0.05453	-0.05590	-0.05533
-0.04347	-0.04199	-0.04332	-0.04469	-0.04760	-0.05221	-0.05453	-0.05590	-0.05633

C_{l_r}

0.04636	0.05272	0.05909	0.06545	0.07182	0.07671	0.07961	0.08350	0.08740
0.04750	0.05348	0.05947	0.06545	0.07144	0.07496	0.07847	0.08198	0.08550
0.04750	0.05311	0.05871	0.06431	0.06992	0.07267	0.07543	0.07818	0.08094
0.04598	0.05139	0.05681	0.06222	0.06764	0.07021	0.07277	0.07534	0.07790
0.04560	0.05092	0.05624	0.06156	0.06688	0.06906	0.07125	0.07344	0.07562
0.04408	0.04930	0.05453	0.05976	0.06498	0.06688	0.06878	0.07068	0.07258
0.04028	0.04541	0.05054	0.05567	0.06080	0.06270	0.06460	0.06650	0.06840
0.03230	0.03724	0.04218	0.04712	0.05206	0.05377	0.05548	0.05719	0.05890
0.02470	0.02802	0.03135	0.03467	0.03800	0.03971	0.04142	0.04313	0.04484
0.01672	0.01786	0.01900	0.02014	0.02128	0.02214	0.02299	0.02385	0.02470
0.01026	0.01121	0.01216	0.01311	0.01406	0.01453	0.01501	0.01548	0.01596
0.00950	0.01045	0.01140	0.01235	0.01330	0.01397	0.01463	0.01529	0.01596
0.00950	0.01045	0.01140	0.01235	0.01330	0.01397	0.01463	0.01529	0.01596

$C_{l_{n\beta}}$

-0.00128	-0.00121	-0.00112	-0.00103	-0.00098	-0.00094	-0.00057	-0.00022	0.00006
-0.00143	-0.00125	-0.00102	-0.00080	-0.00063	-0.00047	0.00002	0.00051	0.00079
-0.00158	-0.00129	-0.00093	-0.00057	-0.00015	0.00027	0.00073	0.00117	0.00155
-0.00065	-0.00057	-0.00049	-0.00041	-0.00020	0.00001	0.00022	0.00042	0.00055
-0.00145	-0.00115	-0.00072	-0.00028	0.00024	0.00077	0.00106	0.00136	0.00164
-0.00160	-0.00142	-0.00122	-0.00102	-0.00083	-0.00065	-0.00041	-0.00018	0.00006
-0.00242	-0.00195	-0.00149	-0.00102	-0.00055	-0.00008	0.00059	0.00126	0.00191
-0.00190	-0.00158	-0.00119	-0.00080	-0.00046	-0.00012	0.00032	0.00077	0.00119
-0.00139	-0.00122	-0.00096	-0.00070	-0.00059	-0.00048	-0.00038	-0.00028	-0.00012
-0.00050	-0.00038	-0.00023	-0.00009	0.00002	0.00014	0.00022	0.00031	0.00045
-0.00003	-0.00002	-0.00003	-0.00003	-0.00002	0.00000	0.00001	0.00001	0.00003
0.00003	0.00004	0.00002	0.00000	0.00002	0.00004	0.00005	0.00006	0.00011
0.00003	0.00004	0.00002	0.00000	0.00002	0.00004	0.00005	0.00006	0.00011

$C_{l_n \delta a}$

```
0.00043 0.00046 0.00050 0.00055 0.00059 0.00063 0.00066 0.00069 0.00071
0.00056 0.00060 0.00063 0.00066 0.00069 0.00072 0.00070 0.00068 0.00066
0.00079 0.00079 0.00080 0.00081 0.00071 0.00062 0.00062 0.00063 0.00060
0.00107 0.00107 0.00094 0.00081 0.00073 0.00065 0.00046 0.00027 0.00033
0.00101 0.00100 0.00095 0.00090 0.00064 0.00039 0.00026 0.00012 0.00013
0.00038 0.00036 0.00035 0.00034 0.00034 0.00034 0.00024 0.00014 0.00012
0.00047 0.00034 0.00020 0.00007 0.00010 0.00013 0.00010 0.00007 0.00003
0.00014 0.00011 0.00007 0.00002 0.00000 -0.00003 -0.00005 -0.00008 -0.00010
-0.00004 -0.00005 -0.00007 -0.00009 -0.00011 -0.00013 -0.00015 -0.00016 -0.00018
0.00002 -0.00001 -0.00003 -0.00005 -0.00007 -0.00009 -0.00010 -0.00011 -0.00012
0.00001 0.00000 -0.00001 -0.00002 -0.00004 -0.00005 -0.00007 -0.00008 -0.00009
0.00000 0.00000 -0.00001 -0.00002 -0.00004 -0.00005 -0.00006 -0.00007 -0.00007
0.00000 0.00000 -0.00001 -0.00002 -0.00004 -0.00005 -0.00006 -0.00007 -0.00007
```

$C_{l_n \delta r}$

```
-0.00093 -0.00092 -0.00090 -0.00088 -0.00085 -0.00083 -0.00082 -0.00081 -0.00080
-0.00084 -0.00082 -0.00079 -0.00076 -0.00074 -0.00073 -0.00072 -0.00071 -0.00072
-0.00057 -0.00057 -0.00057 -0.00056 -0.00056 -0.00055 -0.00054 -0.00052 -0.00051
-0.00008 -0.00010 -0.00017 -0.00023 -0.00029 -0.00034 -0.00034 -0.00033 -0.00030
-0.00012 -0.00012 -0.00014 -0.00015 -0.00018 -0.00020 -0.00019 -0.00018 -0.00016
-0.00062 -0.00057 -0.00054 -0.00050 -0.00047 -0.00044 -0.00043 -0.00041 -0.00040
-0.00053 -0.00052 -0.00050 -0.00048 -0.00045 -0.00043 -0.00042 -0.00041 -0.00041
-0.00038 -0.00037 -0.00036 -0.00035 -0.00034 -0.00033 -0.00032 -0.00030 -0.00030
-0.00027 -0.00026 -0.00026 -0.00025 -0.00024 -0.00023 -0.00022 -0.00021 -0.00019
-0.00021 -0.00021 -0.00020 -0.00020 -0.00019 -0.00018 -0.00017 -0.00016 -0.00015
-0.00015 -0.00014 -0.00013 -0.00012 -0.00011 -0.00010 -0.00009 -0.00008 -0.00007
-0.00011 -0.00010 -0.00009 -0.00007 -0.00006 -0.00005 -0.00005 -0.00004 -0.00004
-0.00011 -0.00010 -0.00009 -0.00007 -0.00006 -0.00005 -0.00005 -0.00004 -0.00004
```

$C_{l_n p}$

```
0.02242 0.01064 -0.00114 -0.01292 -0.00114 0.01064 0.02622 0.04180 0.05738
0.01311 0.01254 0.01197 0.01140 0.01045 0.00950 0.01995 0.03040 0.04085
0.01862 0.01900 0.01938 0.01976 0.01273 0.00570 0.00095 -0.00380 -0.00855
-0.00342 0.00000 0.00342 0.00684 -0.00418 -0.01520 -0.02280 -0.03040 -0.03800
-0.00304 -0.00418 -0.00532 -0.00646 -0.02223 -0.03800 -0.04750 -0.05700 -0.06650
0.01824 0.00494 -0.00836 -0.02166 -0.03173 -0.04180 -0.05605 -0.07030 -0.08455
0.01520 0.00380 -0.00760 -0.01900 -0.00038 0.01824 -0.00988 -0.03800 -0.06612
0.03363 0.02660 0.01957 0.01254 0.01577 0.01900 0.01102 0.00304 -0.00494
0.04446 0.03800 0.03154 0.02508 0.02071 0.01634 0.01197 0.00760 0.00323
0.02413 0.02052 0.01691 0.01330 0.01197 0.01064 0.00969 0.00874 0.00779
0.04351 0.03344 0.02337 0.01330 0.00665 0.00000 -0.01710 -0.03420 -0.05130
0.01710 0.01140 0.00570 0.00000 -0.00570 -0.01140 -0.01520 -0.01900 -0.02280
0.01710 0.01140 0.00570 0.00000 -0.00570 -0.01140 -0.01520 -0.01900 -0.02280
```

$C_{l_n r}$

```
-0.10564 -0.10754 -0.12996 -0.10108 -0.10716 -0.11020 -0.11438 -0.11134 -0.11172
-0.10032 -0.10108 -0.10906 -0.09958 -0.09880 -0.12008 -0.12312 -0.12768 -0.14402
-0.10792 -0.10868 -0.11172 -0.12008 -0.12692 -0.12958 -0.14022 -0.16226 -0.20140
-0.11172 -0.10868 -0.10602 -0.14516 -0.16682 -0.13756 -0.32528 -0.18696 -0.24320
-0.11362 -0.12008 -0.13832 -0.15466 -0.15542 -0.14516 -0.24700 -0.20596 -0.26220
-0.13452 -0.13718 -0.14782 -0.17176 -0.17670 -0.17024 -0.18278 -0.25270 -0.39102
-0.16302 -0.15846 -0.19570 -0.20634 -0.19988 -0.26372 -0.21356 -0.32528 -0.40052
-0.17480 -0.17480 -0.18240 -0.18240 -0.15960 -0.15960 -0.15200 -0.14440 -0.14440
-0.15960 -0.15960 -0.16150 -0.16150 -0.16530 -0.17480 -0.18620 -0.20140 -0.22420
-0.12160 -0.12236 -0.12540 -0.12920 -0.14060 -0.15390 -0.17290 -0.19836 -0.23180
-0.08170 -0.08246 -0.08360 -0.08550 -0.08930 -0.09880 -0.11400 -0.13110 -0.15390
-0.12730 -0.13566 -0.14022 -0.11058 -0.08284 -0.07790 -0.08740 -0.10602 -0.09918
-0.12730 -0.13566 -0.14022 -0.11058 -0.08284 -0.07790 -0.08740 -0.10602 -0.09918
```

C_{y_β}

-0.01868	-0.01866	-0.01844	-0.01822	-0.01810	-0.01799	-0.01760	-0.01697	-0.01697
-0.01815	-0.01790	-0.01790	-0.01790	-0.01790	-0.01790	-0.01790	-0.01805	-0.01820
-0.01870	-0.01870	-0.01865	-0.01860	-0.01870	-0.01880	-0.01885	-0.01890	-0.01825
-0.02040	-0.02020	-0.01995	-0.01970	-0.01935	-0.01900	-0.01810	-0.01720	-0.01560
-0.01946	-0.01850	-0.01800	-0.01750	-0.01750	-0.01750	-0.01620	-0.01490	-0.01317
-0.01667	-0.01700	-0.01675	-0.01650	-0.01630	-0.01610	-0.01600	-0.01590	-0.01532
-0.01785	-0.01715	-0.01645	-0.01575	-0.01525	-0.01475	-0.01470	-0.01465	-0.01469
-0.01782	-0.01654	-0.01584	-0.01515	-0.01475	-0.01435	-0.01429	-0.01423	-0.01333
-0.01938	-0.01638	-0.01563	-0.01488	-0.01446	-0.01405	-0.01358	-0.01311	-0.01231
-0.00895	-0.00792	-0.00757	-0.00723	-0.00691	-0.00659	-0.00641	-0.00623	-0.00592
-0.00601	-0.00552	-0.00527	-0.00503	-0.00488	-0.00473	-0.00465	-0.00414	-0.00403
-0.00562	-0.00515	-0.00477	-0.00440	-0.00430	-0.00420	-0.00410	-0.00400	-0.00338
-0.00562	-0.00515	-0.00477	-0.00440	-0.00430	-0.00420	-0.00410	-0.00400	-0.00338

$C_{y_{\delta a}}$

-0.00320	-0.00330	-0.00355	-0.00380	-0.00390	-0.00400	-0.00406	-0.00412	-0.00416
-0.00321	-0.00333	-0.00354	-0.00375	-0.00387	-0.00400	-0.00403	-0.00405	-0.00410
-0.00428	-0.00415	-0.00403	-0.00390	-0.00385	-0.00380	-0.00357	-0.00335	-0.00314
-0.00510	-0.00500	-0.00445	-0.00390	-0.00323	-0.00255	-0.00215	-0.00175	-0.00216
-0.00400	-0.00400	-0.00370	-0.00340	-0.00230	-0.00120	-0.00075	-0.00030	-0.00062
-0.00214	-0.00203	-0.00203	-0.00202	-0.00180	-0.00158	-0.00149	-0.00140	-0.00134
-0.00155	-0.00110	-0.00105	-0.00100	-0.00112	-0.00125	-0.00119	-0.00113	-0.00104
-0.00060	-0.00054	-0.00046	-0.00038	-0.00027	-0.00015	-0.00015	-0.00015	-0.00015
-0.00003	0.00000	0.00009	0.00018	0.00025	0.00031	0.00034	0.00037	0.00039
0.00011	0.00015	0.00021	0.00026	0.00032	0.00038	0.00041	0.00045	0.00047
-0.00011	-0.00008	-0.00004	0.00001	0.00006	0.00010	0.00013	0.00016	0.00017
-0.00013	-0.00009	-0.00005	-0.00001	0.00002	0.00006	0.00008	0.00010	0.00012
-0.00013	-0.00009	-0.00005	-0.00001	0.00002	0.00006	0.00008	0.00010	0.00011

$C_{y_{\delta r}}$

0.00305	0.00300	0.00295	0.00290	0.00283	0.00275	0.00268	0.00262	0.00261
0.00275	0.00270	0.00268	0.00265	0.00261	0.00258	0.00256	0.00255	0.00253
0.00210	0.00205	0.00203	0.00200	0.00199	0.00198	0.00198	0.00198	0.00199
0.00082	0.00085	0.00090	0.00095	0.00105	0.00115	0.00120	0.00125	0.00118
0.00044	0.00045	0.00050	0.00055	0.00062	0.00070	0.00066	0.00063	0.00056
0.00180	0.00170	0.00166	0.00162	0.00156	0.00150	0.00146	0.00142	0.00138
0.00165	0.00160	0.00155	0.00150	0.00145	0.00140	0.00136	0.00132	0.00132
0.00121	0.00120	0.00116	0.00112	0.00101	0.00102	0.00099	0.00095	0.00092
0.00090	0.00090	0.00087	0.00084	0.00079	0.00075	0.00072	0.00070	0.00072
0.00065	0.00062	0.00059	0.00056	0.00054	0.00052	0.00049	0.00046	0.00044
0.00040	0.00038	0.00035	0.00032	0.00030	0.00028	0.00025	0.00022	0.00020
0.00028	0.00026	0.00023	0.00020	0.00018	0.00016	0.00014	0.00013	0.00010
0.00028	0.00026	0.00023	0.00020	0.00018	0.00016	0.00014	0.00013	0.00010

$C_{m_{\delta e}}$

-0.00019	-0.00019	-0.00019	-0.00019	-0.00019	-0.00019	-0.00019	-0.00018	-0.00017
-0.00020	-0.00020	-0.00020	-0.00020	-0.00020	-0.00020	-0.00020	-0.00019	-0.00017
-0.00021	-0.00021	-0.00021	-0.00021	-0.00021	-0.00021	-0.00021	-0.00020	-0.00018
-0.00023	-0.00023	-0.00023	-0.00023	-0.00023	-0.00023	-0.00023	-0.00022	-0.00020
-0.00017	-0.00017	-0.00017	-0.00017	-0.00017	-0.00017	-0.00017	-0.00016	-0.00014
-0.00014	-0.00014	-0.00014	-0.00014	-0.00014	-0.00014	-0.00014	-0.00013	-0.00012
-0.00014	-0.00014	-0.00014	-0.00014	-0.00014	-0.00014	-0.00014	-0.00012	-0.00011
-0.00016	-0.00016	-0.00016	-0.00016	-0.00016	-0.00016	-0.00016	-0.00015	-0.00013
-0.00013	-0.00013	-0.00013	-0.00013	-0.00013	-0.00013	-0.00013	-0.00012	-0.00011
-0.00009	-0.00009	-0.00009	-0.00009	-0.00009	-0.00009	-0.00009	-0.00008	-0.00007
-0.00015	-0.00015	-0.00015	-0.00014	-0.00014	-0.00013	-0.00012	-0.00012	-0.00012
-0.00014	-0.00014	-0.00014	-0.00014	-0.00013	-0.00012	-0.00012	-0.00011	-0.00011
-0.00014	-0.00014	-0.00014	-0.00014	-0.00013	-0.00012	-0.00012	-0.00011	-0.00011

δ_e

```
2.4936 0.0000 -1.4043 -1.8374 -1.6536 -1.3649 -2.3623 -8.1304 -1.2775
2.7232 0.0000 -1.5534 -2.0657 -2.1362 -2.2112 -3.3234 -7.7868 -3.3318
2.5864 0.0000 -1.7746 -2.4294 -2.6810 -3.3873 -4.7375 -8.7774 -6.7602
2.7449 0.0000 -1.6470 -2.2907 -2.7737 -3.4368 -4.6264 -7.0461 -6.1086
3.3769 0.0000 -2.6411 -3.8401 -4.6392 -6.2352 -8.2696 -12.5045 -13.7870
4.5883 0.0000 -3.0537 -4.5858 -6.0167 -7.8888 -10.0578 -12.8418 -13.7221
4.2608 0.0000 -3.6664 -5.7665 -7.0894 -9.0590 -11.4905 -15.9708 -20.1165
3.5847 0.0000 -2.7536 -4.5499 -6.2123 -8.1109 -9.9647 -11.9311 -13.3703
3.8947 0.0000 -3.5827 -6.0978 -7.3824 -8.6446 -10.3625 -13.7957 -19.6225
5.2793 0.0000 -4.6617 -8.3870 -11.7100 -14.8660 -17.8678 -21.1933 -24.9666
3.0107 0.0000 -2.8951 -5.4283 -7.1408 -8.8443 -10.6945 -13.2967 -17.3718
2.7405 0.0000 -2.6537 -5.1968 -7.7219 -10.3661 -12.8748 -15.1866 -17.0350
2.6006 0.0000 -2.6006 -5.1958 -7.7807 -10.3417 -12.9090 -15.4552 -18.0311
```

L/D

```
-3.01843 -0.99464 1.28370 3.26300 4.56835 4.92715 4.74023 4.28754 3.86251
-3.00868 -1.02577 1.38169 3.24392 4.53213 4.86354 4.62947 4.00031 3.43723
-2.93242 -1.05652 1.18470 3.06295 3.99984 4.10645 3.81081 3.40297 3.06178
-2.52900 -0.76982 1.12896 2.59470 3.32609 3.51035 3.35137 3.07151 2.78377
-2.12455 -0.60653 1.02426 2.37897 3.12431 3.29842 3.17300 2.93662 2.64411
-1.55875 -0.37290 0.87288 1.88397 2.47868 2.79378 2.77785 2.63752 2.40421
-1.66750 -0.43991 0.71169 1.76495 2.45131 2.78378 2.80980 2.68026 2.52971
-1.48946 -0.48525 0.59211 1.55130 2.24781 2.62229 2.73254 2.67760 2.49128
-1.27945 -0.41958 0.52268 1.39679 2.06087 2.46702 2.61159 2.59704 2.44452
-1.21335 -0.47953 0.36328 1.18105 1.81802 2.25449 2.42901 2.45572 2.34340
-1.08469 -0.59414 0.08549 0.83580 1.51044 2.00907 2.25726 2.34988 2.30924
-1.16491 -0.72424 -0.13576 0.54794 1.17278 1.71717 2.07032 2.25354 2.27409
-1.16581 -0.72662 -0.13962 0.54297 1.16751 1.71222 2.06658 2.25113 2.27253
```